

Sonderheft Nr. 6

Schneider

INTERNATIONAL

Sonderheft

6/88

Grundlagen:

Von der Datei zur Datenbank

- relative und sequentielle Dateiverwaltung

Lernen:

**Vokabeltrainer
Funktionsplotter**

Hilfe:

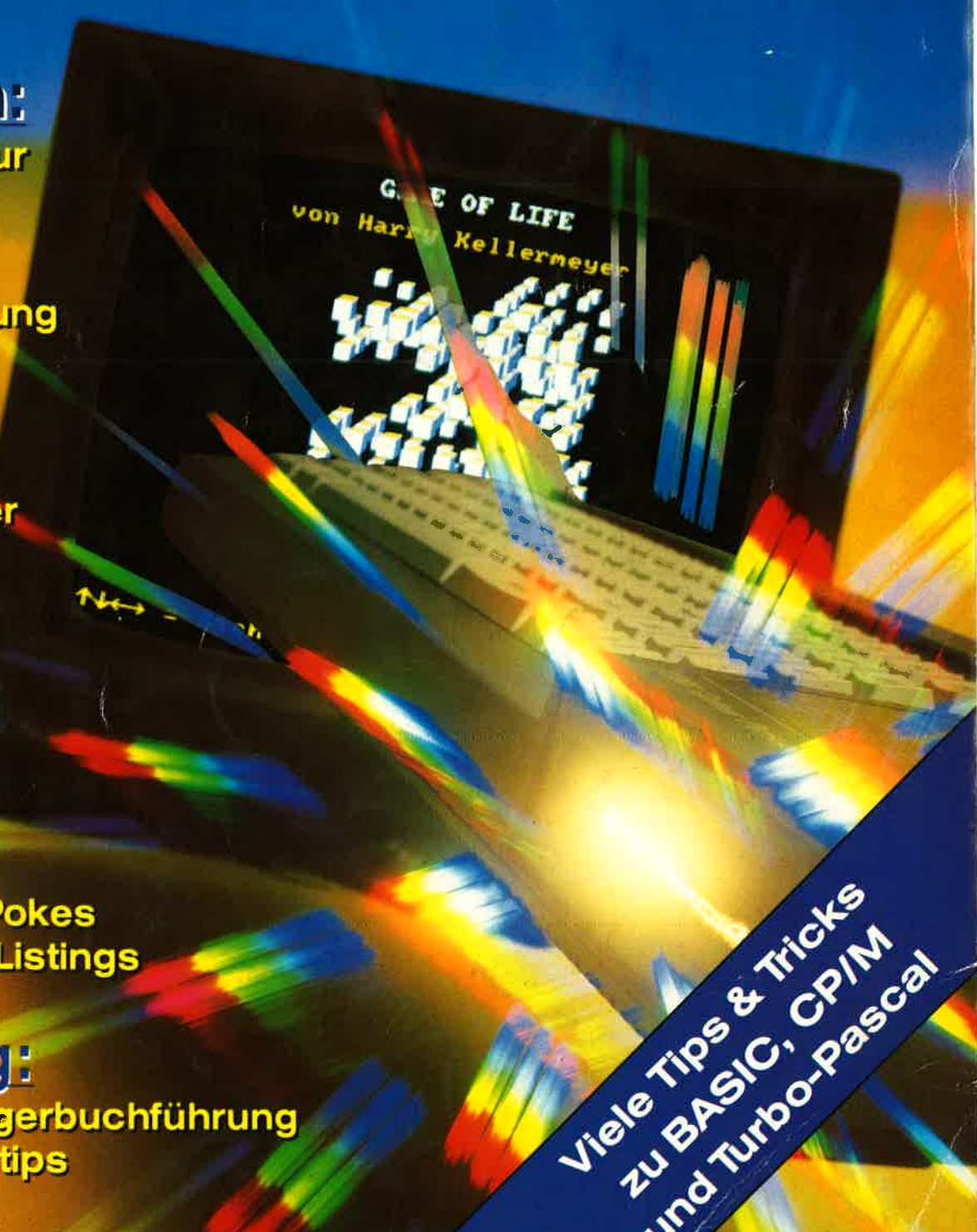
- Umgang mit Disketten
- Utilities

Spiele:

- jede Menge Pokes
- interessante Listings

Anwendung:

- komplette Lagerbuchführung
- viele Druckertips



**Viele Tips & Tricks
zu BASIC, CP/M
und Turbo-Pascal**

SPECIAL OFFERS!

für CPC 464-664-6128, nur auf 3"- Disketten
**Original CPC-Software im Paket
 zu stark herabgesetzten Preisen
 – nur solange Vorrat reicht –**

CPC
 SPECIAL OFFER

No. 1

vier Programme
 Best-Nr. 204 nur

NUR 69,-DM

COPY-STAR II

Ist die ideale Befehlsverweigerung für Druckerbesitzer, denn es stellt für alle gängigen Drucker Hardcopyfunktionen in verschiedenen Größen zur Verfügung. Sogar Farbbilder lassen sich schattiert ausgeben. COPY-STAR II können Sie leicht in eigene Programme einbinden.

MATHE-STAR

Vom Lehrer für Schüler

- lin. Gleichungssysteme
- Gleichungen 4. Grades
- Bruchrechnen
- Primfaktorenzerlegung
- Polynome
- Kurvendiskussion
- Integralrechnung
- Vektorrechnung
- Matrixrechnung
- etc.

STATISTIC-STAR

Eine Grafik sagt mehr als 1000 Zahlen
 Ein professionelles Grafik- und Statisticprogramm zum Auswerten von Daten aller Art (Schule, Studium, Beruf, Hobby, Haushalt ...).

- Linien-, Balken- und Tortengrafik
- Betiteln von Grafiken
- 400 Daten direkt im Speicher
- Umfangreiche Editierfunktionen
- Umfangreiche statistische Berechnungen
- Hardcopyfunktion u.v.m.

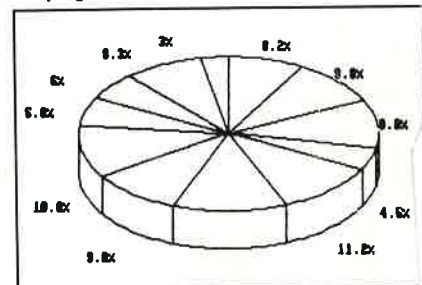
COMPOSER-STAR (664/6128)

Ein Musikprogramm für alle Musikfans

Mit COMPOSER-STAR können Sie komplette Musikstücke oder nur ein paar Taktfolgen auf einfachste Weise erstellen! Natürlich stehen Ihnen auch eine Menge Korrekturhilfen zur Verfügung, um Ihrem Stück den letzten Schliff zu geben. Sie können auch mehrere Stücke verbinden, transponieren, die Tonhöhenkurven und die Lautstärke verändern, den Rauschgenerator benutzen, Notenblätter drucken ... Das Programm ist sehr einfach zu bedienen; das beiliegende Handbuch gibt auch dem Anfänger einen Einblick in die bislang fremde Welt der Musik. Das wichtigste jedoch ist, dass Sie Ihre Werke in ein Basicprogramm umwandeln können, um es in eigenen Programmen zu verwenden.



Beispielgrafik STATISTIC-STAR



CPC
 SPECIAL OFFER

No. 2

vier Programme
 Best-Nr. 205

NUR 69,-DM

STAR-MON

Das Entwicklungssystem für Profis

- Assembler
- Editor
- Disassembler
- Monitor
- vier Breakpoints
- Trace-Funktion
- Bankswitch
- Memory Dump
- Diskettenmonitor
- u.v.m.

CREATOR-STAR

Ein Trickfilmdesigner für alle Hobbyregisseure auf dem CPC!

- Sprite-Designer
- Laufschrift
- Utilities
- Kulissendesigner
- Sprites mit 4 Unterpositionen
- Verbinden von Sprites
- Kulissen auch übereinandergelegt
- Eigene Programmiersprache mit Editor und Compiler

DISKSORT-STAR

Leistungsstarke Diskettenverwaltung, die keinem CPC-Benutzer fehlen sollte. DISKSORT-STAR verwaltet, archiviert, katalogisiert, druckt, ... Ihre Diskettensammlung auf einfachste Weise. Neben der reinen Diskettenverwaltung ist unter anderem noch ein kompletter Diskettenmanager enthalten. Auch in punkto Bedienungskomfort ist DISKSORT-STAR kaum zu schlagen.

DESIGNER-STAR

Grafikprogramm, mit dem man Bildschirmgrafiken komfortabel erstellen kann. Hilfsmenü auf Tastendruck – kein Joystick oder Maus notwendig.

CPC
 SPECIAL OFFER

No. 3 (Games)

neun Programme
 Best-Nr. 107

NUR 49,-DM

9 Spiele für Ihren CPC auf 3" Disketten

- 1) Stan und der Zauberstab – Ein deutsches Grafikadventure besonderer Art
- 2) Pyramide – Eine Mischung aus Textadventure und Actionspiel
- 3) Puzzle – Bringen Sie ein durcheinandergefallenes Bild wieder in Ordnung
- 4) Black Jack
- 5) Orion
- 6) Labyrinth
- 7) Memory
- 8) Zick-Zack
- 9) Slalom

Bitte Bestell-Coupon auf Seite 95 benutzen.

SPECIAL OFFERS sind nur erhältlich bei: **DMV Verlag · Postfach 250 · 3440 Eschwege**

mpressum

erausgeber
Christian Widuch

hefredakteur
Stefan Ritter

Redaktion
Claus Daschner (CD), Heinrich Stiller (HS), Jürgen Borngießer (JB)

Redaktions-Assistenz
Anke Kerstan (Ke)

Produktionsleitung
Gerd Köberich

Satz
Claudia Küllmer, Silvia Führer,
Martina Siebert, Gabriela Joseph

Gestaltung
Petra Biehl, Yvonne Hendricks, Manuela Eska
Mohamed Hawa

Reprografie
Helmut Skoupy, Margarete Schenk

Illustration
Heinrich Stiller

Fotografie
Christian Heckmann

Lektorat
Elvira Domonkos

Anzeigenleitung
Wolfgang Schnell
Es gilt die Anzeigenpreisliste Nr. 2a vom 15.10.86

Anzeigenverkauf
Wolfgang Brill

Freie Mitarbeiter
Jan Schefers, Thomas Kochmann, B. Ott,
Dieter Noack, Dieter Braun, Andreas Martin,
Carsten Hesener, Ekkehard Vosseler,
Heinz Wienhoven, Klaus Kremer, Michael Megelin,
Werner Renziehausen, Helmut Steeb,
Karsten Mohlzahn, Michael Peter,
Wolfgang Otternberg, Holger Kleinert,
Klaus Multerer, Michael Wessel, Gerd Kilian,
Thorsten Schatz, Roland Schäffer, Hans Werner
Fromme, Michael Knaup, Stefan Gründel,
Jan Mirco Maczewski, Jan Wieck, Thorsten Dietl,
Harry Kellermeyer, Oliver Mayer,
Eckehardt Röscheisen, Jan Hennig,
Bernhard Bichmann, Tina Kiene, Matthias Prange,
Günter Lange, Michael Folz, Roland Morlock

Anschrift Verlag/Redaktion:
DMV-Daten & Medien Verlagsges. mbH
Postfach 250, Fuldaer Straße 6
3440 Eschwege
Telefon: (0 56 51) 87 02 · Telex 993 210 dmv d

Vertrieb
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise:
Schneider CPC International Sonderheft
erscheint viermal jährlich.

Einzelpreis DM 14,-/sfr. 14,- /OS 120,-

Für unverlangt eingesandte Manuskripte und Daten-
träger sowie Fotos übernimmt der Verlag keine Haf-
tung.

Honorare nach Vereinbarung (die Zustimmung zum
Abdruck wird vorausgesetzt).

Das Urheberrecht für veröffentlichte Manuskripte
liegt ausschließlich beim Verlag. Nachdruck sowie
Vervielfältigung oder sonstige Verwertung von Tex-
ten, nur mit schriftlicher Genehmigung des Verlages.

Namentlich gekennzeichnete Fremdbeiträge geben
nicht in jedem Fall die Meinung der Redaktion
wieder.

Es gibt nichts, was es nicht gibt...



*Liebe Leser,
seit nunmehr drei Jahren erfreuen sich die Schneider CPCs großer Beliebtheit. Rückblickend kann festgestellt werden, daß es kein anderer Computertyp geschafft hat, diverse Anwendungsbereiche in solch großer Vielfalt abgedeckt zu haben. Der CPC-Einsatz liegt vornehmlich in der Anwendung; das Computerspiel steht hier lediglich an zweiter Stelle. Die ausgeklügelte Konzeption eines Komplettangebotes Tastatur-Speichermedium-Monitor ließ den CPC u.a. im schulischen Bereich eine bedeutende Rolle einnehmen.*

Darüberhinaus werden die CPCs in vielen kleinen Betrieben für Standardaufgaben herangezogen und erfüllen auch dort ihre Anforderungen zur vollsten Zufriedenheit.

Das nun vorliegende CPC-Sonderheft will Ihnen einen Querschnitt aller denkbaren und realisierten Anwendungen bieten, neben allgemeinen Grundlagen erwartet Sie, liebe Leser, eine wahrhaft riesige Software-Fundgrube.

Unsere komplette Lagerbuchhaltung zeigt Ihnen auf eindrucksvolle Weise die Möglichkeiten einer gewerblichen Anwendung, eine wirklich hervorragende Lösung des Problems konnte realisiert werden.

Für schulische Belange haben wir eine Programmsammlung ausgearbeitet, die alle wesentlichen Aufgabenbereiche abdecken kann. Viele Grundlagenartikel zu diesen Themen lassen zudem kaum Fragen offen.

Programmierer können sich über viele nützliche Tips und Tricks freuen, die allesamt die Computerarbeit zum wahren Vergnügen werden lassen und leicht in eigene Programme eingebunden werden können.

Natürlich gibt es auch für die Freunde des Computerspiels viele Informationen in Form von Tips und Lösungshilfen bis hin zu kompletten Lageplänen. Selbstverständlich gibt es auch diesmal wieder viele tolle Spielprogramme zum Abtippen.

Nach insgesamt drei Jahren, in denen auch wir die CPCs immer wieder mit Programmen aller Art versorgt haben, stellt sich natürlich die Frage, ob und wie noch Neues geboten werden kann. An dieser Stelle möchten wir Ihnen für die tolle Unterstützung danken, ohne die eine konsequente Qualitätssteigerung der Artikel und Programme nicht möglich gewesen wäre. Gleichzeitig rufen wir alle Interessierten auch weiterhin zur aktiven Mitarbeit auf, die Redaktion freut sich über jede Resonanz, die auch in Form von Kritik herzlich willkommen ist.

In diesem Sinne entlassen wir Sie nun in die hoffentlich gewinnbringende Lektüre des sechsten CPC-Sonderheftes.

Herzlichst Ihr

Stefan Ritter
(Chefredakteur)



Von der Datei zur Datenbank. Alles Wissenswerte und mehrere Anwendungsbeispiele zu diesem Thema.

S. 6



Das Anwendungsprogramm Faktu verwaltet einen kompletten Geschäftsbetrieb. Von der Lagerbuchführung bis zur Rechnungserstellung reichen die Leistungsdaten.

S.24



Jede Menge Tips und Hilfen zum Umgang mit Ihrer Diskettenstation. U.a. erwartet Sie ein komfortables Diskettenarchiv-Programm.

S. 44

Bericht:

Von der Datei zur Datenbank

- Grundlagen und Anwendungsbeispiele einer relativen und sequenziellen Datei

Anwendungen:

Drucken – aber richtig

- Druckersteuerung per Menue!

Das REF-Programm

- untersucht BASIC-Programme nach Variablen und Zeilennummern.

Sind Ihre Finanzen in Ordnung?

- mit unserer Haushalts-Buchführung kein Problem...

Fakturist

- DIE Super-Anwendung!! Komplette Lagerbuchhaltung mit Rechnungserstellung, Bestandsfortschreibung und und und...

Disketten und Texte codieren mit Coding

- schützen Sie Ihre Daten vor unbefugtem Zugriff.

Lineare Gleichungssysteme

- für die CPCs anschaulich realisiert.

EXODOS

- so erhalten Sie den Disk-Durchblick.

Testme

- ein Vokabeltrainer für jedermann

Menue-Generator

- Ladehilfe für Programme

Farbeneinsteller

- Bildschirmfarben komfortabel einstellen

Schulnoten-Verwaltung

- Zensuren im »Griff«

Comex

- Compressor und Expander für Basic-Programme

Diskordner

- Disketten komfortabel archivieren

Pro-Data

- eine Dateiverwaltung für alle Fälle

Discmanager

- erleichtert den Umgang mit Disketten

Ordnung ist die halbe Musik

- eine außergewöhnliche Verwaltung von Compact-discs(CD)

Plot it

- oder wie erleichtert man sich das Leben mit der Mathematik

Turbo Pascal:

Grafik & Co. unter Turbo Pascal

- Super-Befehlserweiterung

Mit Screen zur Hardcopy

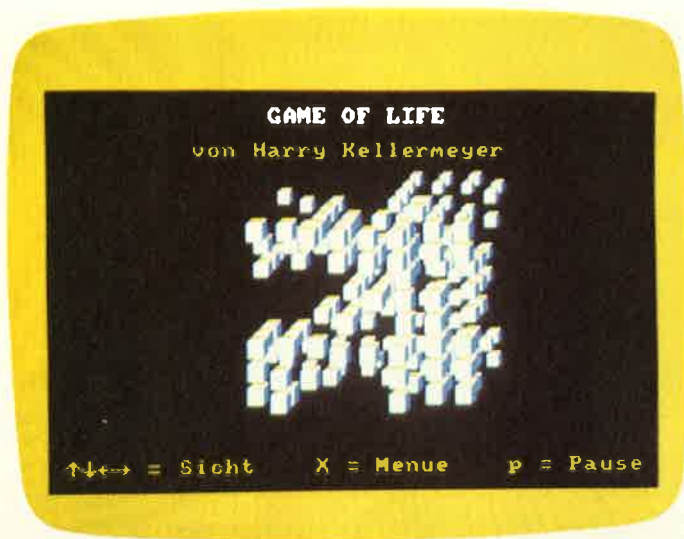
- Routine für Turbo-Anwender

Directory unter Turbo Pascal

- mit DIR zum Ziel.

Spiel:

n Schafen und Wölfen	111
– tolles Spiel für zwei Personen	
absunk	102
– Strategiespiel mit Pfiff...	
uperpokes	110
– für Super-Spiele	
D-Life	115
– das bekannte Spiel jetzt in 3D!!	
Tips und Lösungskarten zu	
– Warlock	118
– Infidel	126
– Bombscare	128
Ball-Cracker	130
O Lala	132
– Software in Frankreich	
Rob der Roboter	140
– ein Spiel für Joystick-Akrobaten	
SHIFT	134
– ein kniffliges Verschiebe-Fix	



Ein Leckerbissen in 3D! Das bekannte Simulationsspiel erlaubt auch trickfilmartige Abläufe...

S.115

Tips & Tricks:

Rasante Textausgabe mit Fastscreen	169
– 100% Geschwindigkeitsvorteil	
Peeks und Pokes zur DDI-1	97
– nützliche Tips zur Schneider-Floppy	
Schnelle Dreiecke	168
– Grafikroutine	
Textausgabe mit Pfiff	174
– animierte Ausgabe von Texten	
Kopfnüsse für BASIC-Freaks	172
– ein BASIC-Raetsel...	
Großbuchstaben mit CHAR	171
– vergrößerte Zeichendarstellung	
Laufschrift-ganz einfach	165
– scrollen von Texten	
Fehlermeldungen im Griff	168
– nie wieder Overflow und division by zero!	
Sort	170
– eine kleine Sortieroutine	
Functional Font Files	157
– neue Zeichensätze für alle Fälle	
Magische Quadrate	177
– per Computer gelöst	
Komplexe Grafiken auf dem CPC	163



Lernprogramme nehmen einen Schwerpunkt dieses Sonderheftes ein. Vom Vokabeltrainer bis zum Funktionsplotter reicht die vielfältige Palette.

S.20

CP/M:

Hardcopies unter CP/M	147
– Text- und Grafikharcopies auf Tastendruck	
Verbesserungen von CP/M 2.2	149
– tolle Ergänzungen für das Betriebssystem	
Deutsches Datum unter CP/M Plus	146

Hardware:

Der Schnittstellen-Umschalter	98
– nutzen von mehreren Geräten	
Der Joystick schießt alleine	96
– Autofeuer- Bauanleitung	



Alle Bildschirmfarben des CPC sind komfortabel über Menue einstellbar.

S. 88



Von der Datei zur Datenbank

Wissenswertes über Datenverwaltung mit Schneider CPC.

Relative und sequentielle Dateiverwaltung, Datenbanksysteme, Direktzugriff, dies sind Schlagworte, die dem Anwender fast täglich begegnen. Wir werden versuchen, anhand von Text, Struktogrammen, Zeichnungen und Programmbeispielen in BASIC, die Möglichkeiten der Dateiverwaltung unter AMSDOS und CP/M und ihre Unterschiede zu beleuchten. Bevor wir jedoch die einzelnen Dateitypen genauer unter die Lupe nehmen, wollen wir zuerst einige Begriffe aus dem Bereich der Dateiverwaltung erläutern.

BEGRIFFE

– DATEI (FILE)

Eine Datei nennt man eine festgelegte Menge von einander zugeordneten Daten. Die Datei hat einen festgelegten Dateinamen, unter dem sie vom Programm aus angesprochen werden kann.

– DATENSATZ (RECORD)

Eine Datei besteht aus einer Anzahl von Datensätzen. Die Länge der Datensätze kann fest oder variabel sein. Bei fester

Datensatzlänge wird unter Umständen Speicherplatz verschwendet. Variable Datensatzlängen erfordern hingegen einen höheren Verwaltungs- sowie Programieraufwand.

– DATENFELD

Der Datensatz ist aus einer Anzahl von Datenfeldern aufgebaut. Das Datenfeld setzt sich wiederum aus einer Anzahl von einzelnen Zeichen zusammen.

In Abb. 1 sehen Sie die drei Begriffe Datei, Datensatz und Datenfeld als Syntaxdiagramme.

SEQUENTIELLE DATEIVERWALTUNG

Nach der etwas trockenen Begriffserklärung wollen wir gleich mit der einfacheren Art der Dateiverwaltung, der sequentiellen, beginnen. Mit ihr lassen sich Files sowohl auf Kassette, als auch auf Diskette oder Festplatte verwalten. Sequentiell bedeutet, daß die einzelnen Dateieinträge hintereinander angeordnet sind. Die Struktur ist dadurch sehr einfach. Wir dürfen jedoch nicht vergessen, daß die sequentielle Dateiverwaltung auch einen großen Nachteil hat: Um z.B. auf den letzten Eintrag einer Datei zuzugreifen zu können, müssen zuerst alle davor liegenden Einträge überlesen werden. Ein Direktzugriff ist nicht möglich. Man kann sich vorstellen, daß die Zugriffszeit bei großen Datenmengen entsprechend

g wird. Zusätzlich muß die ganze Datei in den Speicher geladen werden, will man sie verändern.

Man sieht, daß die sequentielle Verwaltungsmethode nur bei kleinen Dateien sinnvoll ist. Leider ist sie die einzige, die ohne große programmiertechnische Himmzüge mittels Schneider-Basic realisiert werden kann. Vielleicht ist dieser Umstand damit zu erklären, daß die ersten AMSTRAD-Computer nur mit einem Kassettenrekorder als Externspeicher entwickelt worden sind. Die Hardware änderte sich, die Software blieb gleich – schade.

Welche Befehle stellt uns das Schneider-Basic zur Verfügung:

– OPENOUT öffnet eine Datei mit einem definierten Namen zum Beschreiben.

– CLOSEOUT schließt die Datei nach dem Schreibzugriff.

– OPENIN öffnet eine Datei mit einem definierten Namen zum Lesen. Ist die Datei nicht vorhanden, erfolgt eine Fehlermeldung.

– CLOSEIN schließt die Datei nach einem Lesezugriff.

– INPUT #9 liest Daten vom Datenträger in den Speicher.

– PRINT #9 schreibt Daten aus dem Speicher auf den Datenträger. Nach dieser grauen Theorie lassen wir einige Beispiele folgen. In unserem ersten Programmteil wollen wir die 26 Buchstaben des Alphabets hintereinander auf Diskette abspeichern.

Der Dateiname sei "LETTER.SEQ".

```
10 OPENOUT "letter.seq"
20 FOR zähler = 1 TO 26
30 letter$ = chr$(64+zähler)
40 PRINT #9, letter$
50 NEXT zähler
60 CLOSEOUT
70 END
```

Mit CAT sehen wir uns das Directory an und stellen fest, daß unsere Datei »LETTER.SEQ« ordnungsgemäß eingetragen worden ist. Nun wollen wir die Daten wieder einlesen und auf dem Bildschirm sichtbar machen:

```
10 OPENIN "letter.seq"
20 FOR zähler = 1 to 26
30 INPUT #9, letter$
40 PRINT letter$
50 NEXT zähler
60 CLOSEIN
70 END
```

Bei diesen ersten, sehr einfachen, Beispielen haben wir 24 Datensätze, die jeweils nur aus einem Datenfeld bestanden, abgespeichert und wieder eingelesen. Wollen wir jedoch Datensätze mit mehreren Feldern verarbeiten, müssen wir sogenannte Feldvariable definieren. Auch hierzu ein kleines Beispiel:

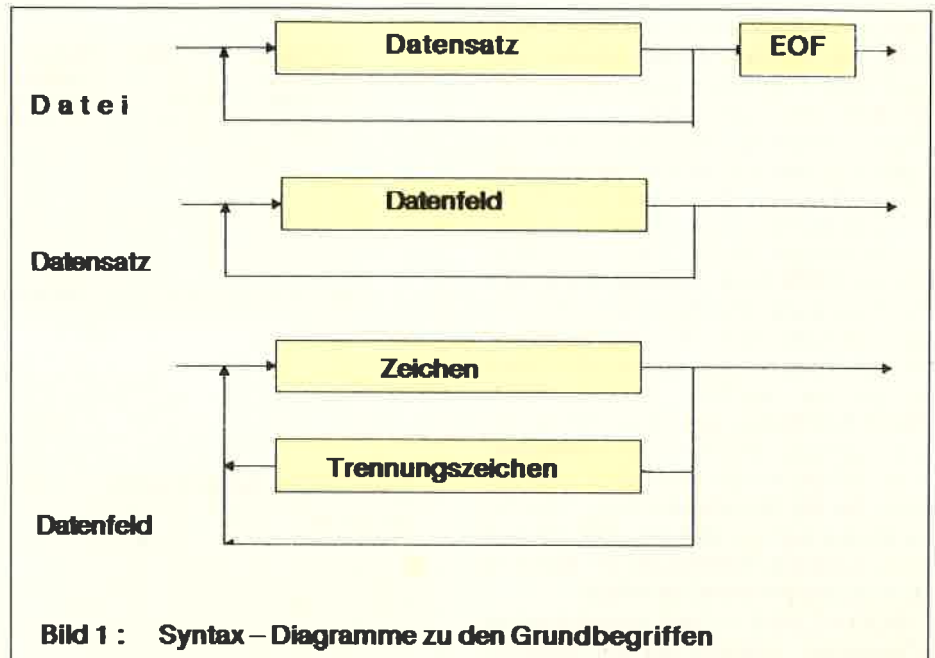


Abb. 1

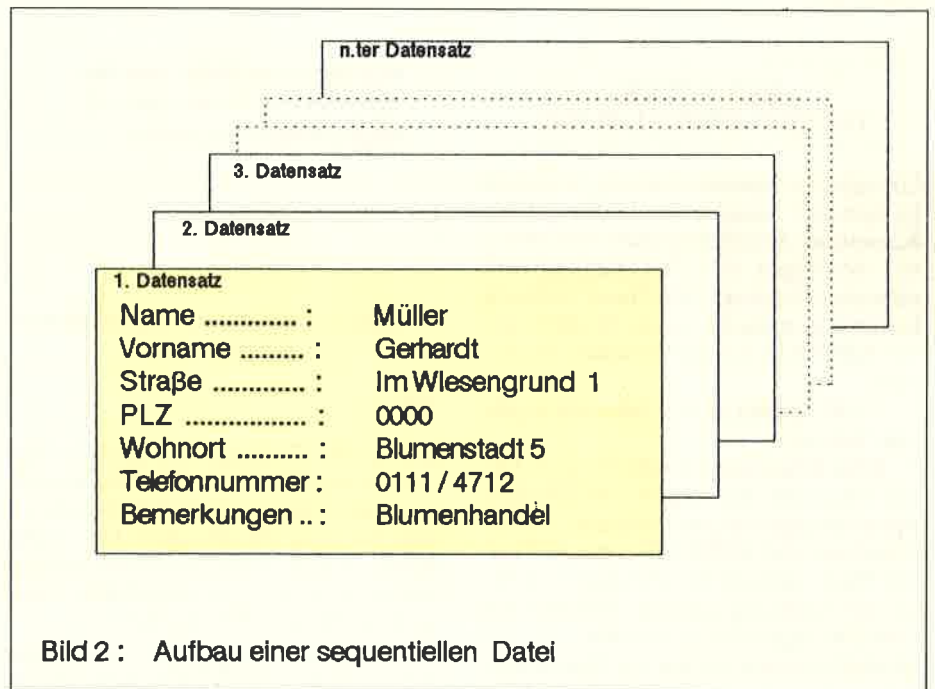


Abb. 2

```
10 DIM fvar$(5,5)
20 OPENIN "namlist.seq"
30 FOR i = 1 TO 5
40 FOR k = 1 TO 5
50 INPUT #9, fvar$(i,k)
60 NEXT k
70 NEXT i
80 CLOSEIN
90 END
```

Mit diesem kleinen Programm werden 5 Datensätze mit je 5 Datenfeldern eingelesen. Die Zählvariable i beinhaltet die Nummer des Datensatzes, k die Nummer des Datenfeldes. Sie sind wichtig für eventuelle Sortier Routinen.

Nach soviel Vorarbeit wollen wir uns endlich mit dem Beispielprogramm einer sequentiellen Dateiverwaltung befassen.

Die nachfolgende Übersicht zeigt den Aufbau des Programms.

Zeile: 230 – Vorbereitung
 250 – Garbage Collection
 310 – Dimensionierungen
 360 – Maskentext
 460 – Sprung zum Menue
 500 – Hilfsroutinen
 520 – Maskenausgabe
 610 – Datei anlegen
 690 – Dateneingabe
 850 – Daten speichern
 1030 – Daten lesen
 1220 – Sortieren
 1480 – Blättern
 1690 – Suchen / Korrigieren
 2090 – Menue
 2280 – Ende

SEQDAT.BAS wurde modular aufgebaut, läßt sich daher leicht verändern und ausbauen. Fehlbedienungen wurden zum größten Teil abgefangen.

Bitte beachten Sie, daß vor der Dateneingabe die Datei eingerichtet werden muß. Ein nochmaliges Einrichten bewirkt die Erstellung einer neuen Datei mit gleichem Namen, die alte Datei wird unter ADRESSEN.BAK abgelegt. Vergessen Sie bitte nicht, die Daten nach der Eingabe abzuspeichern. Vor dem Sortieren, Blättern und Suchen müssen die Daten eingelesen werden. Mittels SEQDAT.BAS lassen sich bis zu 100 Adressen eingeben, speichern und verarbeiten. Eine höhere Anzahl ist möglich, bitte ändern Sie die Dimensionierung. Die Zugriffszeit auf die Diskette und die Zeit zum Sortieren, Blättern und Suchen wird jedoch entsprechend ansteigen. SEQDAT.BAS ist kein ausgeklügeltes Programm, sondern nur ein Beispiel, was Sie zum Ändern und Verbessern anregen soll.

RELATIVE DATEIVERWALTUNG

Um relative Dateiverwaltung betreiben zu können, müssen wir leider auf die Kassette als Speichermedium verzichten, wir benötigen ein Diskettenlaufwerk oder eine Festplatte. Wie Ihnen vielleicht bekannt sein dürfte, ist die Diskette unter AMSDOS folgendermaßen formatiert:

– 2 x 40 Spuren zu je 9 Sektoren ergibt 360 Sektoren.

– Jeder Sektor kann bis zu 512 Byte Daten enthalten, das ergibt 184320 Byte Speicherkapazität pro Diskseite. Leider bietet uns das AMSDOS- oder Schneider-Basic nicht die Möglichkeit, von Basic aus direkt auf einzelne Sektoren der Diskette zugreifen zu können.

Deshalb müssen wir uns ein Maschinenprogramm schreiben, welches auf die AMSDOS-Betriebssystemroutinen zurückgreift. Sie brauchen jedoch über keine MC-Programmierkenntnisse verfügen, denn das Programm ist Bestandteil des Beispiels einer relativen Dateiverwaltung, welches Sie ebenfalls in diesem Heft finden.

Wodurch unterscheidet sich die relative Datei von der sequentiellen? Es ist nicht notwendig, die komplette Datei in den Speicher zu laden, um sie dort zu bearbeiten. Auch muß nicht mehr die ganze Datei vom ersten File an nach einem bestimmten Begriff durchforstet werden, will man einen bestimmten File herausfiltern. Man kann direkt auf jedes File der Datei auf der Diskette zugreifen und zwar relativ zum ersten File der Datei. Anders als bei der sequentiellen Datei ist im Vorfeld der Dateieinrichtung eine

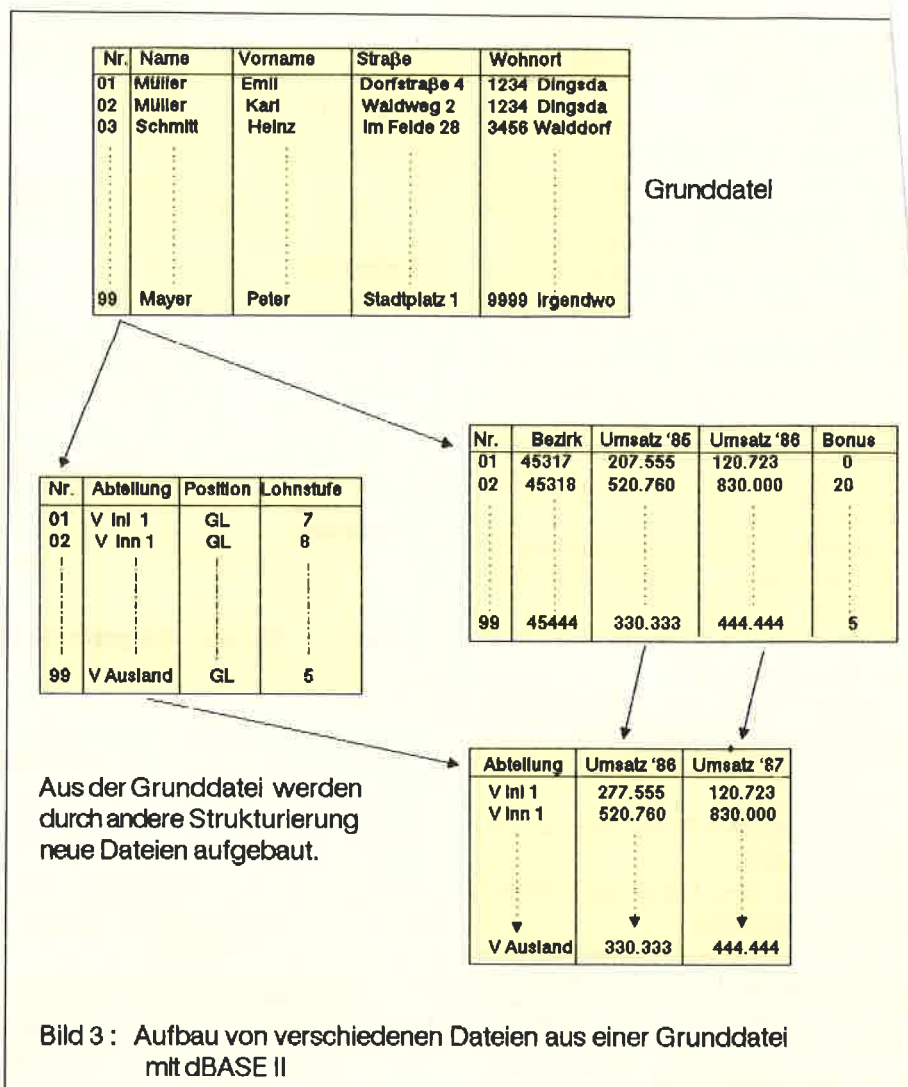


Abb. 3

gründliche Planung erforderlich. Wir müssen uns überlegen, wieviel Datensätze unsere Datei maximal enthalten soll und wie lang die einzelnen Datensätze werden sollen. Diese Parameter sind fest und können im Programmablauf später nicht mehr geändert werden.

Wie weiß ein Programm, in welchem Sektor ein bestimmtes File zu finden ist? Nehmen wir einmal an, unser Datensatz sei exakt 512 Bytes lang, dies ist die einfachste Möglichkeit.

Sektornummer und Datensatznummer sind identisch. Das Programm muß lediglich die Lage des Sektors auf der Diskette errechnen und kann danach gezielt die einzelnen Datensätze verwalten. Wie hoch ist nun die Speicherkapazität pro Diskettenseite? Auch diese Rechnung ist sehr einfach: 40 Spuren x 9 Sektoren ergibt 360 Sektoren, und da Datensatz und Sektor gleich groß sind, stehen uns 360 Datensätze zur Verfügung. Einen Nachteil darf man nicht vergessen. Bei Ausnutzung von weniger als 512 Zeichen pro Datensatz verschenkt man wertvollen Speicherplatz. Deshalb muß man sich gründlich überlegen, ob man nicht ein wenig mehr Rechenaufwand in Kauf

nimmt und kleinere Datensätze mit z.B. 256 oder 128 Byte Kapazität verwendet. Unser Beispielprogramm zur relativen Dateiverwaltung arbeitet mit Datensätzen von maximal 8 Feldern bei einer Gesamtanzahl von 512 Byte. Es ist menuegesteuert und bietet folgende Möglichkeiten:

- Datei einrichten (Dateistruktur festlegen)
- Daten eingeben und editieren
- Daten löschen
- Daten auflisten
- Datei löschen
- Datei laden
- Datei sichern
- Datei sortieren
- Datei auflisten

Das Programm besteht aus einem Programmteil in Maschinensprache und einem Basicteil. Der MC-Teil sorgt für den Diskettenzugriff, während der Basicteil die Benutzeroberfläche darstellt. Treuen Lesern wird das Beispielprogramm eventuell bekannt vorkommen. Es handelt sich in der Tat um die verbesserte Version der relativen Dateiverwaltung aus dem CPC Sonderheft Nr. 2/86.

perimentieren Sie ruhig ein wenig mit dem Programm oder verwenden Sie es für Ihre Schallplatten- oder Kassettenmahlung.

DATENBANKSYSTEM BEISPIEL dBASE II

Bis vor einigen Jahren war es die Regel, daß für die unterschiedlichsten Anwendungen eigene Programme für die Dateiverarbeitung existierten. Die einzelnen Dateien konnten nur von einem Programm verwaltet werden ein Austausch oder eine Verknüpfung von Inhalten mehrerer Dateien war nicht möglich. Diese Forderungen wurden mit der Zeit immer dringender und man machte sich Gedanken über den Aufbau von völlig unabhängigen und universellen Datenbanken. Es entstanden zwei Typen von Datenbanken:

– Hierarchische oder Netzwerk Datenbanken mit Zeigern, um zu einem Begriff weitere Informationen abrufen zu können. Man kann sich die Struktur in Form eines Baumes vorstellen. Der Ausgangsbegriff steckt in der Wurzel des Baumes. Die zusätzlichen Informationen in den Ästen und Zweigen. Bei einem großen Baum wird die Struktur sehr komplex und unübersichtlich. Umfangreiche Berechnungen werden erforderlich. Dies und der enorm große Speicherbedarf sind der Grund dafür, daß solche Datenbanktypen nur in Großrechenanlagen zu finden sind.

– Relationale Datenbanken, wie z.B. dBASE II, arbeiten grundlegend anders: Ausgehend von einer bestimmten Relation, d.h. einer Datei mit gleichartigen Datensätzen lassen sich beliebig neue Dateien generieren, die mit mindestens einem Datenfeld mit der vorherigen Datei verknüpft sind.

Ein Beispiel:
Unsere Datei-Adresse besteht aus Datensätzen mit den Feldern:

- Name
- Vorname
- Strasse
- Wohnort
- Telefon

Wir strukturieren nun eine zweite Datei. Die Datensätze beinhalten die Felder:

- Name
- Mitarbeiternummer
- Vertriebsbezirk
- Eintritt in die Firma
- Stellung
- Gehaltsgruppe
- usw.

Aus dieser Struktur könnte man eine dritte entwickeln usw. Sie sehen, der Aufbau ist sehr einfach und ebenso simpel lassen sich die gewünschten Strukturen verwirklichen.

Wie mächtig dBASE II ist, zeigt die Übersicht des Leistungsumfangs.

Auf dBase II Dateien kann man nicht ohne weiteres von Programmen, die mittels einer höheren Programmiersprache entwickelt worden waren, zugreifen. Deshalb wurde dBASE II von den Entwicklern mit einer eigenen Kommandosprache ausgestattet. Diese verwaltet nicht nur die angelegten Dateien, sondern bietet dem Anwender die Möglichkeit, sich selbst die unterschiedlichsten Anwenderprogramme und Benutzeroberflächen selbst zu programmieren.

Das Erlernen der Kommandosprache ist einfach, obwohl der Sprachumfang sehr umfangreich ist. Ein hervorragendes Beispiel einer benutzeroberfläche mit dBASE II-Kommandodateien möchten wir an dieser Stelle noch einmal erwähnen, obwohl im Heft 9/87 ausführlich darüber berichtet worden ist. Es handelt sich um das Programmpaket VAR-DAT II von SOFTDESIGN. Wer sich mit dBase II selbst intensiver befassen möchte, der sei auf die Serie dBASE II ab Heft 11/86 hingewiesen. Dieser sehr interessante Lehrgang für Anfänger und Fortgeschrittene befaßt sich ausgiebig mit den Besonderheiten des Datenbanksystems und seiner Programmierung. Einen großen Vorteil dürfen wir abschließend nicht vergessen:

Da dBASE II in verschiedenen Versionen sowohl unter unter CP/M, als auch MS-DOS ablauffähig ist, brauchen die Kommandodateien beim Wechsel des Betriebssystems nicht umgeschrieben werden, vorausgesetzt man kann die verschiedenen Diskettenformate unter einen

Hut bringen oder überträgt die Daten z.B. über eine serielle Schnittstelle.

ES MUSS NICHT IMMER dBASE SEIN

Es gibt Anwender, denen ist dBASE II zu umfangreich, zu mächtig. Andere wiederum setzen sich lieber an den gedeckten Tisch und wollen sich nicht vor der eigentlichen Anwendung noch mit der Anpassung des Programms befassen. Nicht nur speziell für diesen Personenkreis, sondern für alle, die sich über Dateiverwaltung informieren wollen, haben wir aus der Vielzahl der Angebote einige Beispiele herausgezogen. In Stichworten erfahren Sie die Besonderheiten der einzelnen Programme und ihre Anwendungsgebiete.

Das war also unser kleiner Streifzug durch das große Gebiet Dateiverwaltung. Obwohl wir nicht tief in die einzelnen Teilgebiete eindringen konnten, hoffen wir, daß es Ihnen Spaß gemacht hat und Sie einiges dazulernen konnten.

(H.-W. Fromme)

LEISTUNGSUMFANG VON dBASE II

Max. Anzahl der Datensätze pro Diskette: 65535

Max. Länge eines Datensatzes: 1000 Zeichen (Byte)

Max. Anzahl der Felder je Satz: 32

Feldtypen und -längen

- alphanumerische Felder: 254 Zeichen
- numerische Felder: 10 Zeichen
- logische Felder: 1 Zeichen

Gleichzeitig eröffnete Dateien: 16
davon Daten-Dateien: 2

Rechengenauigkeit: 10 gültige Ziffern
Anzahl der Indexdateien pro Stammdatei: 7

Max. Anzahl der Speichervariablen: 64

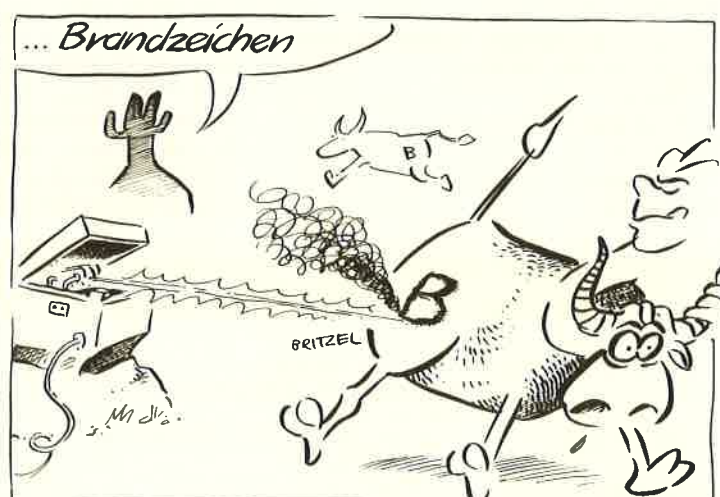
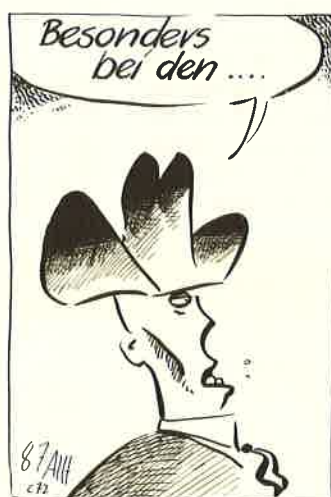
Max. Anzahl der Daten-Dateien pro System: unbegrenzt

Max. Anzahl der Kommando-Dateien: unbegrenzt

WAS DER MARKT SO BIETET:

DATEI STAR V2.0	– Pull Down Menues – Frei definierbare Eingabemaske mit 30 Datenfeldern	QUICKDATA 3.0 JOYCE	– Definition von Datenstrukturen, Bildschirm und Druckformularen
Star Division	– Frei definierbare Such-, Sortier- und Druckmaske	KHS Software	– max. 100 Felder pro Datensatz – max. 99 Zeichen pro Feld
Lüneburg	– 1400 Zeichen pro Datensatz – max. 255 Zeichen pro Feld	Fachhandel	– theoretisch bis zu 32 Mio. Datensätzen pro Datei
DM 98, –	– integrierte Kalkulationsmöglichkeit – Suchen / Selektieren / Etiketten- und Listendruck – Schnittstelle zu STAR WRITER	DM 198, –	– Eingabeformular mit max. je 50 Ein- und Ausgabefeldern – Prüfung der Eingaben – Datenbanksystem

<p>TAS</p> <p>W. Henschke Datentechnik 7034 Gärtringen</p> <p>DM 298, -</p>	<ul style="list-style-type: none"> - relationales Datenbanksystem - eigene Kommandosprache - max. 16 geöffnete Dateien - max. 16 Schlüsselfelder pro Datei - max. 254 Zeichen pro Feld - max. 255 Variable pro Programm - max. 10.254 Zeichen pro Datensatz - Anzahl der Felder pro Datei unbegrenzt - unlimitierte Dateigröße - max. 65.535 Datensätze 	<p>KARTEIKASTEN</p> <p>RSE Software Castrop-Rauxel</p> <p>DM 79, -</p>	<ul style="list-style-type: none"> - Relative Dateiverwaltung - max. 400 Karten pro Diskette - beliebige Suchkriterien - Listenausdruck - komfortabler Editor - Schnelle Sortierroutine - Hardcopyfunktion - DIN Tastaturbelegung
<p>RH-DAT</p> <p>Integral Hydraulik Düsseldorf</p> <p>DM 79, - /89, -</p>	<ul style="list-style-type: none"> - relative Dateiverwaltung unter CP/M - max. 503 bzw. 1003 Datensätze - freie Maskendefinition - max. 21 Felder pro Datensatz - max. 70 Zeichen pro Feld - schnelle Suchroutine 	<p>UNIDATEI</p> <p>RSE Software Castrop-Rauxel</p> <p>DM 79, -</p>	<ul style="list-style-type: none"> - Relative Dateiverwaltung - Maskengenerator für frei definierbare Maske - je nach Maske und Indexfelder mehrere Tausend Datenfelder möglich - Such- und Druckmaske frei erstellbar - schnelle Routinen zum Suchen und Sortieren - DIN Tastaturbelegung - Hardcopyfunktion
<p>PROFIREM JOYCE</p> <p>V.d. Zalm Software Wangerland</p> <p>DM 178, -</p>	<ul style="list-style-type: none"> - bestehend aus Kundendatei, Lagerdatei und Fakturierung - Kundendatei 1000 Adressen - Lagerdatei 1000 Artikel - Fakturierung mit Schnittstelle zu Kunden- und Lagerdatei 	<p>LAGERVERWALTUNG</p> <p>RSE Software Castrop-Rauxel</p> <p>DM 79, -</p>	<ul style="list-style-type: none"> - Relative Dateiverwaltung - Editor - Bestands- / Inventurliste - Artikel VK-Preisliste - Druck auf Etiketten sowie Listendruck - Statistik - Such- und Sortier Routinen - max. 1350 Artikel pro Datei - Dateienanzahl beliebig
<p>DATAMAT</p> <p>DATA Becker Düsseldorf</p> <p>Fachhandel</p> <p>DM 99, -</p>	<ul style="list-style-type: none"> - Menüsteuerung - frei gestaltbare Eingabemaske - max. 80 Zeichen pro Zeile - eingebaute Hardcopyfunktion - max. 50 Felder pro Datensatz - max. 512 Zeichen pro Datensatz - max. 4000 Datensätze - Schnittstelle zu TEXTOMAT - Benutzung von Rechenfeldern - Suchen / Selektieren / Druck, auch von Etiketten 	<p>VEREINSVERWALTUNG</p> <p>RSE Software Castrop-Rauxel</p> <p>DM 79, -</p>	<ul style="list-style-type: none"> - Max 700 Mitglieder pro Datendiskette - integriertes Kassenbuch - umfangreiche Druckroutinen - Jahresabschluss - Komfortabler Editor - Schnelle Such- und Sortierroutine
<p>EASY MEMO</p> <p>MICROLAND</p> <p>Fachhandel</p>	<ul style="list-style-type: none"> - Adressenverwaltung - max. 400 Adressen pro Datendiskette - alphabetisches Sortieren - Selektion nach Teilen der Adresse - Ausdruck, auch von Adressenaufklebern - Schnittstelle zu EASY SCRIPT 	<p>MULTIDATEI CPC</p> <p>Rainbow Arts Fachhandel</p>	<ul style="list-style-type: none"> - Kassetten- und Diskettenversion erhältlich - sequentielle Dateiverarbeitung - frei definierbare Eingabemaske - max. 60 Felder je Datensatz - max. 16 Zeichen pro Feld - Abspeichern der Daten auf Kasette/ Diskette - Suchen und Sortieren - Listen und Ausdrucken - Addition numerischer Felder - Säulengrafiken
<p>ADRESSDATEI</p> <p>RSE Software Castrop-Rauxel</p> <p>DM 59, -</p>	<ul style="list-style-type: none"> - Relative Dateiverwaltung - max. 900 Datensätze pro Diskette - professioneller Editor - schnelles Sortieren und Suchen von Daten - Etikettendruck für verschiedene Formate - Listenausdruck - Hardcopyfunktion eingebaut 		



```

10 '***** [2519]
*****
20 '*' [175]
      *
30 '*      Beispiel einer sequenti [4408]
ellen Dateiverwaltung      *
40 '*' [175]
      *
50 '*      (c) 1987 by Hans-Werner [1409]
Fromme      *
60 '*' [175]
      *
70 '*      PC Schneider Internatio [2355]
nal Sonderheft 6/87      *
80 '*' [175]
      *
90 '*' [175]
      *
100 '*      Programmname: SEQDAT.BA [2892]
S      *
110 '*' [175]
      *
120 '*      Dateiname : ADRESSE.S [1531]
EQ      *
130 '*' [175]
      *
140 '*      Datenfelder : NAME$, VO [4489]
RNAME$, STRASSE$, PLZ$,      *
150 '*      ORT$, TEL [1437]
EFON$, BEMERKUNG$      *
160 '*' [175]
      *
170 '*      Max. Anzahl der Datenfe [1910]
lder: 100      *
180 '*      kann bei Bedarf in der [3220]
Dimensionierung geaendert werden      *
190 '*' [175]
      *
200 '***** [2519]
*****
210 ' [117]
220 ' [117]
230 '*** VORBEREITUNG *** [817]
240 ' [117]
250 '--- GARBAGE COLLECTION --- [1267]
260 ' [117]
270 OPENOUT "muell" [1304]
280 MEMORY HIMEM-1 [259]
290 CLOSEOUT [902]
300 ' [117]
310 '--- DIMENSIONIERUNG DER STRING-ARRAYS [2865]
---
320 ' [117]
330 DIM name$(100), vorname$(100), strasse$( [3953]
100), plz$(100)
340 DIM ort$(100), telefon$(100), bemerkung$ [2030]
(100)
350 ' [117]
360 '--- MASKENTEXT --- [1046]
370 ' [117]
380 mt$(1)="Name      :" [1485]
390 mt$(2)="Vorname    :" [1209]
400 mt$(3)="Strasse   :" [1066]
410 mt$(4)="PLZ      :" [561]
420 mt$(5)="Ort      :" [1555]
430 mt$(6)="Telefon   :" [1155]
440 mt$(7)="Bemerkung  :" [1847]
450 ' [117]
460 '*** SPRUNG ZUM MENUE *** [1312]
470 ' [117]
480 GOTO 2090 [353]
490 ' [117]
500 '*** HILFSROUTINEN *** [1315]
510 ' [117]
520 '--- MASKENAUSGABE --- [614]
530 ' [117]
540 CLS : PRINT ueberschrift$ : PRINT : PR [2209]
INT
550 FOR i = 1 TO 7 [447]
560     PRINT mt$(i) : PRINT [777]
570     NEXT i [375]
580 PRINT : PRINT anmerkung$ [1447]
590 RETURN [555]
600 ' [117]
610 '--- DATEI ANLEGEN --- [1087]
620 ' [117]
630 sz=0 [188]
640 OPENOUT "adressen.seq" [1986]
650 PRINT #9, sz [1056]

```

Listing: Von der Datei zur Datenbank

```

660 CLOSEOUT [902]
670 RETURN [555]
680 ' [117]
690 --- DATENEINGABE --- [1036]
700 ' [117]
710 GOSUB 1070 [829]
720 ueberschrift$ = "EINGABE" : anmerkung$ [2226]
= "BEENDEN MIT LEEREINGABE"
730 GOSUB 540 [937]
740 sz = sz + 1 : LOCATE 1,2 : PRINT "Date [3660]
nsatz : "; sz
750 LOCATE 25,4 : INPUT name$(sz) [1471]
760 IF name$(sz) = "" THEN sz = sz-1 : RET [2732]
URN
770 LOCATE 25,6 : INPUT vorname$(sz) [2256]
780 LOCATE 25,8 : INPUT strasse$(sz) [918]
790 LOCATE 25,10 : INPUT plz$(sz) [1932]
800 LOCATE 25,12 : INPUT ort$(sz) [996]
810 LOCATE 25,14 : INPUT telefon$(sz) [1984]
820 LOCATE 25,16 : INPUT bemerkung$(sz) [1387]
830 CLS : GOTO 730 [983]
840 ' [117]
850 '--- AUF DISKETTE ABSPEICHERN --- [2767]
860 ' [117]
870 CLS : PRINT "SPBICHERN AUF DISKETTE" : [2005]
PRINT
880 IF sz = 0 THEN RETURN [241]
890 OPENOUT "adressen.seq" [1986]
900 PRINT #9, sz [1056]
910 FOR i = 1 TO sz [493]
920     PRINT #9, name$(i) [1352]
930     PRINT #9, vorname$(i) [1140]
940     PRINT #9, strasse$(i) [1538]
950     PRINT #9, plz$(i) [1138]
960     PRINT #9, ort$(i) [1683]
970     PRINT #9, telefon$(i) [1732]
980     PRINT #9, bemerkung$(i) [1548]
990     NEXT i [375]
1000 CLOSEOUT [902]
1010 RETURN [555]
1020 ' [117]
1030 '--- VON DISKETTE LESEN --- [1123]
1040 ' [117]
1050 CLS : PRINT "LADEN VON DISKETTE" : PR [1719]
INT
1060 FOR i = 1 TO 1000 : NEXT [741]
1070 OPENIN "adressen.seq" [2274]
1080 INPUT #9, sz [833]
1090 IF sz=0 THEN CLOSEIN : RETURN [1826]
1100 FOR i = 1 TO sz [493]
1110     INPUT #9, name$(i) [1406]
1120     INPUT #9, vorname$(i) [969]
1130     INPUT #9, strasse$(i) [1137]
1140     INPUT #9, plz$(i) [1492]
1150     INPUT #9, ort$(i) [1264]
1160     INPUT #9, telefon$(i) [1869]
1170     INPUT #9, bemerkung$(i) [1328]
1180     NEXT i [375]
1190 CLOSEIN [752]
1200 RETURN [555]
1210 ' [117]
1220 '--- SORTIEREN --- [1369]
1230 ' [117]
1240 ueberschrift$ = "SORTIEREN DURCH PAAR [2880]
TAUSCH"
1250 anmerkung$ = " ... ich bin beschae [4212]
ftigt, bitte etwas Geduld "
1260 CLS : PRINT ueberschrift$ : PRINT : P [3402]
RINT anmerkung$
1270 IF sz <= 0 THEN RETURN [200]
1280 sortflag = 0 [895]
1290 FOR k = 1 TO sz - 1 [1651]
1300     IF name$(k+1) >= name$(k) THEN 14 [1262]
40
1310     hn$ = name$(k) : hv$ = vorname$( [4070]
k) : hs$ = strasse$(k)
1320     hp$ = plz$(k) : ho$ = ort$(k) : h [1788]
t$ = telefon$(k)
1330     hb$ = bemerkung$(k) [1335]
1340     ' [117]
1350     name$(k) = name$(k+1) : vorname$( [1895]
k) = vorname$(k+1)
1360     strasse$(k) = strasse$(k+1) : plz [1844]
$(k) = plz$(k+1)
1370     ort$(k) = ort$(k+1) : telefon$(k) [4087]
= telefon$(k+1)
1380     bemerkung$(k) = bemerkung$(k+1) [404]
1390     ' [117]
1400     name$(k+1) = hn$ : vorname$(k+1) [3787]

```

Listing: Von der Datei zur Datenbank

```

= hv$ : strasse$(k+1) = hs$
1410   plz$(k+1) = hp$ : ort$(k+1) = ho$ [3283]
      : telefon$(k+1) = ht$
1420   bemerkung$(k+1) = hb$ [180]
1430   sortflag = 1 [894]
1440   NEXT k [373]
1450 IF sortflag > 0 THEN 1280 [1044]
1460 RETURN [555]
1470 ' [117]
1480 '--- BLAETTERN --- [1202]
1490 ' [117]
1500 ueberschrift$ = "BLAETTERN" [1137]
1510 anmerkung$ = "BLAETTERN MIT <-- UND - [3742]
-> ; ENDE MIT <RETURN>"
1520 l = 1 [180]
1530 GOSUB 540 [937]
1540 LOCATE 25,4 : PRINT name$(1) [1002]
1550 LOCATE 25,6 : PRINT vorname$(1) [1493]
1560 LOCATE 25,8 : PRINT strasse$(1) [1911]
1570 LOCATE 25,10 : PRINT plz$(1) [1906]
1580 LOCATE 25,12 : PRINT ort$(1) [1107]
1590 LOCATE 25,14 : PRINT telefon$(1) [1471]
1600 LOCATE 25,16 : PRINT bemerkung$(1) [1577]
1610 taste$ = INKEY$ [1039]
1620 IF taste$ = "" THEN 1610 [1543]
1630 taste = ASC(taste$) [1196]
1640 IF taste = 243 THEN IF l < sz THEN l = [1555]
l+1
1650 IF taste = 242 THEN IF l > 1 THEN l = [2390]
l-1
1660 IF taste = 13 THEN RETURN [2058]
1670 GOTO 1530 [393]
1680 ' [117]
1690 '--- SUCHEN / KORRIGIEREN --- [1031]
1700 ' [117]
1710 IF sz=0 THEN RETURN [241]
1720 ueberschrift$ = "SUCHEN / KORRIGIEREN [2818]
"
1730 anmerkung$ = "" [740]
1740 GOSUB 540 [937]
1750 LOCATE 1,23 : INPUT "SUCHBEGRIFF "; s [2140]
uchbegriff$
1760 i = 1 [423]
1770 IF suchbegriff$ = name$(1) THEN G [3602]
OSUB 1880 : GOTO 1790
1780 i = i + 1 : IF i > sz THEN 1850 E [1584]
LSE 1770
1790 LOCATE 1,23 : PRINT " [2723]
"
1800 LOCATE 1,23 : INPUT "KORREKTUR ( [2636]
J/N) "; antwort$
1810 IF antwort$ = "j" THEN GOSUB 1990 [2243]
1820 LOCATE 1,23 : INPUT "WEITERSUCHEN [2915]
(J/N)"; antwort$
1830 IF antwort$ <>"j" THEN RETURN : [2246]
1840 i = i + 1 : IF i > sz THEN 1850 E [1584]
LSE 1770
1850 LOCATE 1,23 [733]
1860 PRINT " D A T E I - E N D E !!!" : [2989]
FOR i=1 TO 1000 : NEXT i
1870 RETURN [555]
1880 FOR k = 4 TO 16 STEP 2 [1416]
1890 LOCATE 25,1 : PRINT SPACES(50) [1637]
1900 NEXT k [373]
1910 LOCATE 25,4 : PRINT name$(1) [1020]
1920 LOCATE 25,6 : PRINT vorname$(1) [1408]
1930 LOCATE 25,8 : PRINT strasse$(1) [1945]
1940 LOCATE 25,10 : PRINT plz$(1) [1956]
1950 LOCATE 25,12 : PRINT ort$(1) [1277]
1960 LOCATE 25,14 : PRINT telefon$(1) [1473]
1970 LOCATE 25,16 : PRINT bemerkung$(1) [1595]
1980 RETURN [555]
1990 LOCATE 23,4 : INPUT name$(1) [1928]
2000 LOCATE 23,6 : INPUT vorname$(1) [1911]
2010 LOCATE 23,8 : INPUT strasse$(1) [1178]
2020 LOCATE 23,10 : INPUT plz$(1) [2350]
2030 LOCATE 23,12 : INPUT ort$(1) [1941]
2040 LOCATE 23,14 : INPUT telefon$(1) [1951]
2050 LOCATE 23,16 : INPUT bemerkung$(1) [1955]
2060 RETURN [555]
2070 ' [117]
2080 ' [117]
2090 '***** M E N U E ***** [1946]
2100 ' [117]
2110 CLS : PRINT " BEISPIEL EINER S E Q U [2662]
E N T I E L L E N D A T E I "
2120 PRINT : PRINT : PRINT [1082]
2130 PRINT " [ 1 ] DATEI EINRICHTEN" [1889]
: PRINT

```

Listing: Von der Datei zur Datenbank

```

2140 PRINT " [ 2 ] DATENEINGABE" [1969]
: PRINT
2150 PRINT " [ 3 ] DATEI SPEICHERN" [1537]
: PRINT
2160 PRINT " [ 4 ] DATEI LADEN" [1872]
: PRINT
2170 PRINT " [ 5 ] SORTIEREN" [2044]
: PRINT
2180 PRINT " [ 6 ] BLAETTERN" [1664]
: PRINT
2190 PRINT " [ 7 ] SUCHEN / KORRIGIEREN" [2973]
: PRINT
2200 PRINT " [ 8 ] E N D E" [1293]
: PRINT
2210 PRINT : PRINT [743]
2220 PRINT " Ihre Wahl "; [1040]
2230 INPUT wahl [527]
2240 ON WAHL GOSUB 610,710,870,1050,1240,1 [2248]
500,1710,2280
2250 GOTO 2110 [357]
2260 ' [117]
2270 ' [117]
2280 '*** E N D E *** [1135]
2290 CLS [91]
2300 PRINT "E N D E " : FOR i =1 TO 1000 : [2706]
NEXT i : END
2310 ' [117]
2320 '***** [2371]
*****

```

```

100 '***** [345]
110 '* Rel. Datei- * [623]
120 '* verwaltung * [89]
130 '***** [345]
140 '* (c)1986 * [540]
150 '* by Th.Fippl * [723]
180 '***** [345]
190 ' [117]
1000 !DISC:MEMORY &95FF [2812]
1010 LOAD"reldat.bin",&9800 [3080]
1020 CALL &9800 [477]
1030 ON BREAk GOSUB 1830 [561]
1040 SPEED KEY 15,1 [1264]
1050 DEFINT m,n,r:DEFSTR i [1688]
1060 DEF FNdt$(t,n)=RIGHT$("0"+FNst$(t),n) [1434]
1070 DEF FNst$(m)=MID$(STR$(m),2) [1953]
1080 DEF FNlc(m)=UNT(m)MOD 256 [953]
1090 DEF FNhi(m)=UNT(m)\256 [1082]
1100 DEF FNdp(m)=PEEK(m)+256*PEEK(m+1) [2060]
1110 bef1$=CHR$(&F2)+CHR$(&F3)+CHR$(&7F)+C [2542]
HR$(&10)
1120 bef2$=CHR$(&F0)+CHR$(&F1)+CHR$(&7F) [2314]
1130 header$=STRING$(128,CHR$(0)) [912]
1140 blank$=SPACES(255) [679]
1150 GOSUB 1280 [859]
1160 ' [117]
1170 '*** Functions-Auswahl *** [1667]
1180 ' [117]
1190 LOCATE#2,5,13 [899]
1200 PRINT#2,"Ihre Wahl: "CHR$(143)CHR$(8) [1647]
;
1210 i=INKEY$ [672]
1220 IF i<"0"OR i>"9"GOTO 1210 [1993]
1230 PRINT#2,i:CLS:PRINT [1192]
1240 IF i="0"GOTO 6600 [1246]
1250 ON VAL(i)GOSUB 3510,4130,4210,4450,45 [3102]
20,4940,5020,5850,6440
1260 PRINT CHR$(7); [1175]
1270 GOTO 1160 [307]
1280 ' [117]
1290 '*** Bildschirm-Maske *** [2026]
1300 ' [117]
1310 MODE 2: BORDER 2 [1093]
1320 FOR n=1 TO 4 [788]
1330 READ win$,x1,x2,y1,y2,pn [481]
1340 PAPER pn: PEN 1-pn [1791]
1350 x=x2-x1+1 [788]
1360 LOCATE x1-1,y1-1 [974]
1370 PRINT CHR$(150)CHR$(24)win$CHR$(24)ST [3739]
RING$(x-LEN(win$),CHR$(154))CHR$(156)
1380 FOR y=y1 TO y2 [747]
1390 LOCATE x1-1,y [952]
1400 PRINT CHR$(149) [1036]
1410 LOCATE x2+1,y [1175]
1420 PRINT CHR$(149) [1036]
1430 NEXT [350]
1440 LOCATE x1-1,y2+1 [1173]
1450 PRINT CHR$(147)STRING$(x,CHR$(154))CH [2426]
R$(153);

```

Listing: Von der Datei zur Datenbank

```

1460 WINDOW x1,x2,y1,y2 [1674]
1470 WINDOW SWAP n [1093]
1480 CLS#:NEXT [615]
1490 WINDOW SWAP 1 [826]
1500 DATA Console,24,79,2,21,0 [742]
1510 DATA Function,2,21,9,21,0 [1270]
1520 DATA Status,2,21,2,6,1 [838]
1530 DATA Help,2,79,24,24,1 [728]
1540 : [174]
1550 FOR n=1 TO 10 [902]
1560 READ func$ [761]
1570 LOCATE#2,1,n+1 [780]
1580 PRINT#2,CHR$(24)" "RIGHT$(FNst$(n),1) [1508]
" "CHR$(24)" "func$;
1590 y=256-n*16 [593]
1600 MOVE 8,y [667]
1610 DRAW 24,0,0 [550]
1620 NEXT [350]
1630 DATA Datei errichten [1397]
1640 DATA Datei loeschen [1532]
1650 DATA Datei laden [1544]
1660 DATA Datei sichern [1001]
1670 DATA Datei sortieren [1184]
1680 DATA Directory zeigen [1786]
1690 DATA Daten eingeben [1308]
1700 DATA Daten auflisten [729]
1710 DATA Daten loeschen [1197]
1720 DATA Zurueck zu BASIC [1439]
1730 : [174]
1740 INPUT#3,"Datum (Tag,Mon,Jahr)";t,m,j [2566]
1750 IF j<100 THEN j=j+1900 [2009]
1760 dat$=FNdt$(t,2)+"."+FNdt$(m,2)+"."+FN [2965]
dt$(j,4)
1770 CLS#3 [371]
1780 PRINT#3,"Datum: "dat$ [1243]
1790 PRINT#3 [316]
1800 PRINT#3,"Datei : " [1080]
1810 PRINT#3,"belegt:" [1041]
1820 PRINT#3,"frei : "; [1302]
1830 RETURN [555]
1840 ' [117]
1850 '*** INPUT String *** [487]
1860 ' [117]
1870 back$=STRING$(t, ".") [1253]
1880 text$="" [435]
1890 xp=POS(#0):xpo=xp [1579]
1900 yp=VPOS(#0) [1452]
1910 GOSUB 2180 [855]
1920 : [174]
1930 GOSUB 2250 [865]
1940 IF i=CHR$(13)GOTO 1970 [945]
1950 GOSUB 2490 [849]
1960 GOTO 1930 [377]
1970 IF LOWER$(text$)="dt"THEN text$=dat$ [1013]
1980 PRINT:CLS#4 [335]
1990 RETURN [555]
2000 ' [117]
2010 '*** INPUT Zahl *** [463]
2020 ' [117]
2030 t=LEN(STR$(thi))-1 [1007]
2040 back$=STRING$(t, "_") [1396]
2050 xpo=POS(#0) [1047]
2060 yp=VPOS(#0) [1452]
2070 GOSUB 2180 [855]
2080 text$="" : xp=xpo [440]
2090 : [174]
2100 GOSUB 2250 [865]
2110 IF i=CHR$(13)GOTO 2140 [1149]
2120 IF i>="0"AND i<="9"THEN GOSUB 2490 [1693]
2130 GOTO 2100 [359]
2140 tval=VAL(text$) [1421]
2150 IF tval<tlo OR tval>thi GOTO 2080 [2670]
2160 PRINT:CLS#4 [335]
2170 RETURN [555]
2180 ' [117]
2190 '*** Help *** [692]
2200 ' [117]
2210 PRINT#4,["DEL] links,[CLR] rechts,["C [5720]
HR$(&F2)"] Cursor links,["CHR$(&F3)"] Curs
or rechts,[ENTER] beenden"
2220 RETURN [555]
2230 PRINT#4,["CHR$(&F0)"] Cursor nach ob [8687]
en,["CHR$(&F1)"] Cursor nach unten,[DEL] a
endern,[ENTER] weiter"
2240 RETURN [555]
2250 ' [117]
2260 '*** INPUT *** [671]
2270 ' [117]
2280 txt=LEN(text$) [1407]

```

Listing: Von der Datei zur Datenbank

```

2290 LOCATE xpo,yp [676]
2300 PRINT text$+MID$(back$,txt+1) [2174]
2310 LOCATE xp,yp [1028]
2320 CALL &BB8A [558]
2330 i=INKEY$ [672]
2340 IF i=""GOTO 2330 [435]
2350 nn=INSTR(bef1$,i) [576]
2360 CALL &BB8D [555]
2370 ON nn GOTO 2390,2410,2430,2460 [1477]
2380 RETURN [555]
2390 xp=MAX(xp-1,xpo) [1875]
2400 RETURN [555]
2410 xp=MIN(xp+1,xpo+txt) [1396]
2420 RETURN [555]
2430 x=xp-xpo [852]
2440 IF x THEN text$=LEFT$(text$,x-1)+MID$ [5038]
(text$,x+1):xp=xp-1
2450 RETURN [555]
2460 x=xp-xpo [852]
2470 IF x<txt THEN text$=LEFT$(text$,x)+MI [1218]
D$(text$,x+2)
2480 RETURN [555]
2490 IF nn THEN RETURN [241]
2500 x=xp-xpo [852]
2510 IF x<t THEN text$=LEFT$(text$,x)+i+MI [3868]
D$(text$,x+1):xp=xp+1
2520 RETURN [555]
2530 ' [117]
2540 '*** Dateiname *** [870]
2550 ' [117]
2560 PRINT"Dateiname ([ENTER] beenden): "C [3352]
HR$(24);
2570 t=8:GOSUB 1840 [1115]
2580 PRINT CHR$(24) [996]
2590 IF text$="" THEN RETURN [1818]
2600 file$=text$+".RDF" [1005]
2610 !FIND,@file$ [1009]
2620 flag=PEEK(&960E) [460]
2630 RETURN [555]
2640 ' [117]
2650 '*** ERROR *** [916]
2660 ' [117]
2670 ON er GOTO 2690,2710,2730,2750,2770 [1510]
2680 RETURN [555]
2690 PRINT"<<< Datei besteht bereits >>>" [3250]
2700 RETURN [555]
2710 PRINT"<<< Datei ist noch geoeffnet >> [4234]
>"
2720 RETURN [555]
2730 PRINT"<<< Datei nicht gefunden >>>" [3559]
2740 RETURN [555]
2750 PRINT"<<< Datei ist geschlossen >>>" [4361]
2760 RETURN [555]
2770 PRINT"<<< Datei ist voll >>>" [2862]
2780 RETURN [555]
2790 ' [117]
2800 '*** RECORD *** [1043]
2810 ' [117]
2820 bytes=rec*reclen [951]
2830 blk=INT(bytes/512) [1480]
2840 rst=bytes-512*blk [453]
2850 !RECORD,blk,rst [932]
2860 RETURN [555]
2870 ' [117]
2880 '*** GET 1 *** [965]
2890 ' [117]
2900 FOR f=0 TO felder-1 [1642]
2910 !GET,f,@i [1261]
2920 mem1$(f)=1 [484]
2930 NEXT:RETURN [940]
2940 ' [117]
2950 '*** PUT 1 *** [586]
2960 ' [117]
2970 FOR f=0 TO felder-1 [1642]
2980 i=mem1$(f) [412]
2990 !PUT,f,@i [664]
3000 NEXT:RETURN [940]
3010 ' [117]
3020 '*** GET 2 *** [813]
3030 ' [117]
3040 FOR f=0 TO felder-1 [1642]
3050 !GET,f,@i [1261]
3060 mem2$(f)=1 [754]
3070 NEXT:RETURN [940]
3080 ' [117]
3090 '*** PUT 2 *** [562]
3100 ' [117]
3110 FOR f=0 TO felder-1 [1642]
3120 i=mem2$(f) [523]
3130 !PUT,f,@i [664]
3140 NEXT:RETURN [940]
3150 '

```

Listing: Von der Datei zur Datenbank

```

3160 '*** MAX(1,2) *** [1230]
3170 ' [117]
3180 IF LOWER$(mem1$(fld))>LOWER$(mem2$(fld)) GOTO 3260 [2804]
3190 RETURN [555]
3200 ' [117]
3210 '*** MIN(1,2) *** [1232]
3220 ' [117]
3230 IF LOWER$(mem1$(fld))<LOWER$(mem2$(fld)) GOTO 3260 [2249]
3240 RETURN [555]
3250 : [174]
3260 ex=m:rec=ex [1826]
3270 GOSUB 2790 [829]
3280 GOTO 2870 [389]
3290 ' [117]
3300 '*** Maske *** [1364]
3310 ' [117]
3320 CLS [91]
3330 FOR f=1 TO felder [1262]
3340 LOCATE 1,3+ftab(f-1) [1446]
3350 PRINT CHR$(24) " fbez$(f)": "CHR$(24) [2411]
3360 x1=flen(f)\56+1 [1599]
3370 x2=((flen(f)MOD 56)>0) [972]
3380 ftab(f)=ftab(f-1)+x1-x2 [1566]
3390 NEXT:RETURN [940]
3400 ' [117]
3410 '*** Datensatz eingeben *** [2146]
3420 ' [117]
3430 PRINT CHR$(30)"Satznummer:"rec [2432]
3440 GOSUB 2790 [829]
3450 FOR f=0 TO felder-1 [1642]
3460 LOCATE 1,ftab(f)+4 [1822]
3470 t=flen(f+1):GOSUB 1840 [569]
3480 IF text$="" THEN RETURN [1818]
3490 mem1$(f)=text$ [776]
3500 NEXT:RETURN [940]
3510 ' [117]
3520 '*** Datei errichten *** [2043]
3530 ' [117]
3540 IF PEEK(&9600) THEN er=2:GOTO 2640 [2104]
3550 GOSUB 2530 [865]
3560 IF text$="" THEN RETURN [1818]
3570 IF flag THEN er=1:GOTO 2640 [1211]
3580 name$=text$ [243]
3590 PRINT"Anzahl der Felder pro Datensatz (max. 8): "; [3938]
3600 tlo=1:thi=8:GOSUB 2000 [918]
3610 PRINT [361]
3620 PRINT"Nr./Laenge/Feldname" [1602]
3630 maxlen=512:feldertval [1351]
3640 FOR n=1 TO felder [1296]
3650 thi=MIN(255,maxlen) [1085]
3660 PRINT n("FNst$(thi)"); [1149]
3670 GOSUB 2000 [875]
3680 LOCATE 4,yp [719]
3690 PRINT USING" ### ";tval; [1380]
3700 maxlen=maxlen-tval [1073]
3710 flen(n)=tval [1093]
3720 t=40:GOSUB 1840 [749]
3730 fbez$(n)=text$ [267]
3740 NEXT:PRINT [811]
3750 :FIND [992]
3760 bytes=PEEK(&960E)*1024-512 [579]
3770 reclen=512-maxlen [1722]
3780 maxrec=INT(bytes/reclen) [2271]
3790 belrec=0 [73]
3800 PRINT"Anzahl der Datensatze (max. "F [3817]
Nst$(maxrec)"; ";
3810 thi=maxrec:GOSUB 2000 [967]
3820 maxrec=tval [934]
3830 bytes=maxrec*reclen [1753]
3840 blk=INT(bytes/255) [1550]
3850 rst=bytes-255*blk [506]
3860 OPENOUT files [781]
3870 FOR n=1 TO 4 [788]
3880 PRINT#9,headers$; [1833]
3890 NEXT [350]
3900 FOR n=1 TO blk [641]
3910 PRINT#9,blank$; [920]
3920 NEXT [350]
3930 PRINT#9,LEFT$(blank$,rst); [1709]
3940 CLOSEOUT [902]
3950 :OPEN,@files [1094]
3960 POKE &960A,FNlo(reclen) [1273]
3970 POKE &960B,FNhi(reclen) [1577]
3980 POKE &9612,FNlo(maxrec) [1369]
3990 POKE &9613,FNhi(maxrec) [1067]
4000 FOR n=1 TO felder [1296]
4010 :ITEM,@fbez$(n),flen(n) [1821]

```

Listing: Von der Datei zur Datenbank

```

4020 NEXT [350]
4030 ' [117]
4040 '*** Statuszeile *** [1158]
4050 ' [117]
4060 LOCATE#3,9,3 [641]
4070 PRINT#3,UPPER$(name$)TAB(17); [1673]
4080 LOCATE#3,8,4 [648]
4090 PRINT#3,belrec;TAB(14); [861]
4100 LOCATE#3,8,5 [651]
4110 PRINT#3,maxrec-belrec;TAB(14); [1745]
4120 RETURN [555]
4130 ' [117]
4140 '*** Datei loeschen *** [1395]
4150 ' [117]
4160 GOSUB 2530 [865]
4170 IF text$="" THEN RETURN [1818]
4180 IF flag=0 THEN er=3:GOTO 2640 [1147]
4190 :ERA,@files [1000]
4200 RETURN [555]
4210 ' [117]
4220 '*** Datei laden *** [1120]
4230 ' [117]
4240 IF PEEK(&9600) THEN er=2:GOTO 2640 [2104]
4250 GOSUB 2530 [865]
4260 IF text$="" THEN RETURN [1818]
4270 IF flag=0 THEN er=3:GOTO 2640 [1147]
4280 name$=text$ [243]
4290 :OPEN,@files [1094]
4300 felder=PEEK(&9605) [1748]
4310 PRINT"Dateistruktur:" [2145]
4320 PRINT"Nr./Laenge/Feldname" [1602]
4330 p=&961A [412]
4340 FOR n=1 TO felder [1296]
4350 :EMIT,@i [1341]
4360 m=FNdp(p+2)-FNdp(p) [1291]
4370 PRINT n;USING" ### "+i;m [940]
4380 fbez$(n)=1 [84]
4390 flen(n)=m [717]
4400 p=p+2:NEXT [1288]
4410 maxrec=FNdp(&9612) [1647]
4420 belrec=FNdp(&9614) [1470]
4430 reclen=FNdp(&960A) [1114]
4440 GOTO 4030 [539]
4450 ' [117]
4460 '*** Datei sichern *** [1285]
4470 ' [117]
4480 IF PEEK(&9600)=0 THEN er=4:GOTO 2640 [2449]
4490 POKE &9614,FNlo(belrec) [945]
4500 POKE &9615,FNhi(belrec) [846]
4510 :CLOSE:RETURN [1120]
4520 ' [117]
4530 '*** Datei sortieren *** [1241]
4540 ' [117]
4550 IF PEEK(&9600)=0 THEN er=4:GOTO 2640 [2449]
4560 PRINT"[1] aufsteigende" [2486]
4570 PRINT"[2] absteigende Reihenfolge" [3355]
4580 PRINT"[0] beenden" [1491]
4590 tlo=0:thi=2:GOSUB 2000 [1779]
4600 IF tval=0 THEN RETURN [1459]
4610 sort=tval [1202]
4620 PRINT [361]
4630 PRINT"Nach welchem Feld soll sortiert werden ?" [4378]
4640 GOSUB 4850 [921]
4650 FOR n=0 TO belrec-2 [1095]
4660 ex=n:rec=ex [966]
4670 GOSUB 2790 [829]
4680 GOSUB 2870 [901]
4690 FOR m=n+1 TO belrec-1 [1202]
4700 rec=m [441]
4710 GOSUB 2790 [829]
4720 GOSUB 3010 [967]
4730 ON sort GOSUB 3150,3200 [934]
4740 NEXT [350]
4750 rec=n [440]
4760 GOSUB 2790 [829]
4770 GOSUB 3010 [967]
4780 GOSUB 2940 [895]
4790 rec=ex [395]
4800 GOSUB 2790 [829]
4810 GOSUB 3080 [1136]
4820 NEXT [350]
4830 RETURN [555]
4840 : [174]
4850 PRINT [361]
4860 PRINT"Mr./Feldname" [855]
4870 FOR n=1 TO felder [1296]
4880 PRINT n " fbez$(n) [485]
4890 NEXT:PRINT [811]
4900 tlo=1:thi=felder [626]
4910 GOSUB 2000 [875]

```

```

4920 fld=tval-1 [1058]
4930 RETURN [555]
4940 ' [117]
4950 '*** Directory zeigen *** [2305]
4960 ' [117]
4970 WINDOW 24,53,2,21 [1105]
4980 file$="*.RDF" [105]
4990 :DIR,@file$ [1149]
5000 WINDOW 24,79,2,21 [1171]
5010 RETURN [555]
5020 ' [117]
5030 '*** Daten eingeben *** [1253]
5040 ' [117]
5050 IF PEEK(&9600)=0 THEN er=4:GOTO 2640 [2449]
5060 PRINT"[1] Datensaeetze anhaengen" [1985]
5070 PRINT"[2] Datensaeetze einfuegen" [1892]
5080 PRINT"[3] Datensaeetze aendern" [2218]
5090 PRINT"[0] beenden" [1491]
5100 PRINT [361]
5110 tlo=0:thi=4 [695]
5120 GOSUB 2000 [875]
5130 ON tval GOTO 5160,5260,5570 [1142]
5140 RETURN [555]
5150 : [174]
5160 GOSUB 3290 [1198]
5170 WHILE belrec<maxrec [1831]
5180 rec=belrec [961]
5190 GOSUB 3400 [1009]
5200 IF text$=""GOTO 4030 [1516]
5210 GOSUB 2940 [895]
5220 belrec=belrec+1 [1484]
5230 WEND [390]
5240 er=5:GOTO 2640 [478]
5250 : [174]
5260 GOSUB 5460 [871]
5270 IF frerec=0 THEN RETURN [1466]
5280 IF belrec+frerec>maxrec THEN er=5:GOT [2339]
O 2640
5290 FOR n=1 TO belrec-posrec [2107]
5300 rec=belrec-n [1246]
5310 GOSUB 2790 [829]
5320 GOSUB 2870 [901]
5330 rec=rec+frerec [934]
5340 GOSUB 2790 [829]
5350 GOSUB 2940 [895]
5360 NEXT [350]
5370 GOSUB 3290 [1198]
5380 belrec=belrec+frerec [2140]
5390 FOR n=0 TO frerec-1 [1211]
5400 rec=posrec+n [1746]
5410 GOSUB 3400 [1009]
5420 IF text$=""GOTO 4030 [1516]
5430 GOSUB 2940 [895]
5440 NEXT:GOTO 4030 [969]
5450 : [174]
5460 IF belrec=0 THEN frerec=0:RETURN [1358]
5470 PRINT"Satznummer:"; [1705]
5480 tlo=0:thi=belrec-1 [1311]
5490 GOSUB 2000 [875]
5500 posrec=tval [621]
5510 PRINT" Anzahl:"; [1701]
5520 thi=belrec-posrec [1368]
5530 GOSUB 2000 [875]
5540 frerec=tval [1570]
5550 RETURN [555]
5560 : [174]
5570 GOSUB 5460 [871]
5580 GOSUB 3290 [1198]
5590 FOR n=0 TO frerec-1 [1211]
5600 rec=posrec+n [1746]
5610 GOSUB 2790 [829]
5620 GOSUB 6350 [853]
5630 GOSUB 5660 [974]
5640 NEXT:RETURN [940]
5650 : [174]
5660 GOSUB 2230 [821]
5670 y=0 [355]
5680 LOCATE 1,ftab(y)+4 [1595]
5690 CALL &BB8A [558]
5700 i=INKEY$ [672]
5710 IF i=""GOTO 5700 [329]
5720 nn=INSTR(bef2$,i) [719]
5730 CALL &BB8D [555]
5740 IF i=CHR$(13)THEN CLS#4:RETURN [1005]
5750 ON nn GOSUB 5770,5790,5810 [1098]
5760 GOTO 5680 [427]
5770 y=MAX(y-1,0) [1280]
5780 RETURN [555]
5790 y=MIN(y+1,felder-1) [1547]
5800 RETURN [555]
5810 t=flen(y+1):GOSUB 1840 [1010]

```

Listing: Von der Datei zur Datenbank

```

5820 IF text$=""THEN RETURN [1818]
5830 mem1$(y)=text$ [558]
5840 GOTO 2940 [383]
5850 ' [117]
5860 '*** Daten auflisten *** [1970]
5870 ' [117]
5880 IF PEEK(&9600)=0 THEN er=4:GOTO 2640 [2449]
5890 PRINT"[1] Alles auflisten" [2552]
5900 PRINT"[2] bedingte Ausgabe" [1819]
5910 PRINT"[0] beenden":PRINT [2206]
5920 tlo=0:thi=2 [705]
5930 GOSUB 2000 [875]
5940 ON tval GOTO 5970,6050 [309]
5950 RETURN [555]
5960 : [174]
5970 GOSUB 3290 [1198]
5980 FOR rec=0 TO belrec-1 [2277]
5990 GOSUB 2790 [829]
6000 GOSUB 6350 [853]
6010 CALL &BB06 [393]
6020 NEXT [350]
6030 RETURN [555]
6040 : [174]
6050 PRINT [361]
6060 PRINT"Nach welchem Feld soll gesucht [3536]
werden ?"
6070 GOSUB 4850 [921]
6080 PRINT"Suchmaske: "; [2172]
6090 t=flen(fld+1):GOSUB 1840 [1404]
6100 IF text$=""THEN RETURN [1818]
6110 maske$=LOWER$(text$) [1665]
6120 GOSUB 3290 [1198]
6130 FOR rec=0 TO belrec-1 [2277]
6140 GOSUB 2790 [829]
6150 GOSUB 2870 [901]
6160 text$=LOWER$(mem1$(fld)) [1889]
6170 GOSUB 6200 [899]
6180 NEXT:RETURN [940]
6190 : [174]
6200 m=1 [184]
6210 FOR n=1 TO LEN(maske$) [1082]
6220 bst$=MID$(maske$,n,1) [1238]
6230 IF bst$=""GOTO 6320 [1473]
6240 IF bst$="?"GOTO 6260 [803]
6250 IF bst$<>MID$(text$,m,1)THEN RETURN [3059]
6260 m=m+1 [200]
6270 NEXT [350]
6280 GOSUB 6390 [877]
6290 CALL &BB06 [393]
6300 RETURN [555]
6310 : [174]
6320 m=INSTR(m,text$,CHR$(44)) [1094]
6330 IF m=0 GOTO 6280 [907]
6340 GOTO 6270 [397]
6350 ' [117]
6360 '*** PRINT *** [1131]
6370 ' [117]
6380 GOSUB 2870 [901]
6390 PRINT CHR$(30)"Satznummer:"rec [2432]
6400 FOR f=0 TO felder-1 [1642]
6410 LOCATE 1,4+ftab(f) [1206]
6420 PRINT mem1$(f) [925]
6430 NEXT:RETURN [940]
6440 ' [117]
6450 '*** Daten loeschen *** [1785]
6460 ' [117]
6470 IF PEEK(&9600)=0 THEN er=4:GOTO 2640 [2449]
6480 GOSUB 5460 [871]
6490 IF frerec=0 THEN RETURN [1466]
6500 FOR n=0 TO belrec-posrec-1 [2285]
6510 rec=posrec+n+frerec [2354]
6520 GOSUB 2790 [829]
6530 GOSUB 2870 [901]
6540 rec=posrec+n [1746]
6550 GOSUB 2790 [829]
6560 GOSUB 2940 [895]
6570 NEXT [350]
6580 belrec=belrec-frerec [2367]
6590 GOTO 4030 [539]
6600 ' [117]
6610 '*** Zurueck zu BASIC *** [1738]
6620 ' [117]
6630 INPUT"Wirklich (j/n)";i [1959]
6640 IF LOWER$(i)<>"j"GOTO 1260 [2168]
6650 GOSUB 4490 [1023]
6660 MODE 1 [506]
6670 PRINT"Neustart mit 'GOTO 1030'" [3424]
6680 CLEAR [217]

```

Listing: Von der Datei zur Datenbank

Sind Ihre Finanzen in Ordnung?

Wenn nicht, dann haben wir etwas für Sie. Nämlich ein Programm, mit dem Sie Ihr Konto überprüfen können und gleichzeitig Ihre Ein- und Ausgaben im Griff haben.

Die einzelnen Punkte zur Bedienung der Haushalts-Buchführung: Die Menüauswahl erfolgt mit den Cursortasten <rechts> und <links>, während mit den Tasten <oben> und <unten> das Verschieben der Ausgabenliste erfolgt.

Hinweis: 464/664 nur mit Emulator
Zu den Menüepunkten:

INGABE:

Datum: Durch ENTER wird das Datum der vorherigen Eingabe übernommen.

Gegenstand: Es werden alle Zeichen akzeptiert, auch Kommata. ENTER schließt die Eingabe ab, sofern die mögliche Anzahl von Zeichen nicht ausgenutzt wurde.

Art der Bewegung:

- H oder ENTER = Haushaltsbewegung
- B = Bankbewegung
- T = Transfer vom Bankkonto auf Haushaltsbestand

Einnahme, Ausgabe: ENTER setzt die Einnahme bzw. Ausgabe auf Null. Bei Zahleneingabe wird nach der zweiten Zahl hinter dem »Komma« die Eingabe automatisch abgeschlossen.

EDIT: Die zu berichtigende Zeile muß im Ausgabefenster als Letzte erscheinen (Pfeil!), erst dann mit ENTER den Menüpunkt aktivieren.

CLS: Löschen einer Zeile: wie EDIT.

INSERT: Wie EDIT. Die neue Zeile wird unter der letzten angezeigten Zeile eingefügt.

AKTUELL: Es werden die letzten zwölf Zeilen angezeigt.

SPEICHERN: Die Frage nach der Bezeichnung der Datei kann direkt mit ENTER beantwortet werden. Die Datei erhält dann den Namen BUCHDAT.BFG.

LADEN: Wie Speichern.

DRUCKEN: Ausgedruckt werden dabei alle Daten, wobei bei Schönschrift pro DIN A4-Seite 58 Zeilen gedruckt werden und nach dem Wenden der Seite der Lochrand berücksichtigt wird.

HILFE: Erläuterung der Funktionen.

ENDE: Wurden die Daten nicht abgespeichert, erfolgt eine Sicherheitsabfrage. Vor Beendigung wird das Inhaltsverzeichnis mit den *.BFG-Dateien ausgegeben.

Variablenliste zum Haushalts-Buchführungs-Programm

- g\$() = Gegenstand
- hb\$() = Art der Geldbewegung
- ein\$(), einb() = Einnahmen auf Haushaltskonto bzw. Bankkonto.
- aush(), ausb() = Ausgaben vom Haushaltskonto bzw. Bankkonto.
- besth(), bestb() = Bestände der jeweiligen Konten.
- gesaush(), gesausb() = Gesamtausgabe der Konten.
- art\$() = Menüpunkte
- day () = Tag
- mon\$() , monat = Monat
- jahr = Jahr
- zahl = Dimension der Felder (zahl=200 meist ausreichend, kann aber vergrößert werden.)
- disk = Enthält den Wert 1, wenn Daten abgespeichert wurden, ansonsten den Wert 0.
- ab\$, abst\$ = Lochrand bei Druckausgabe.
- zr\$ = Enthält Carriage Return und Line Feed für Drucker.
- name\$ = Name der abzuspeichernden, bzw. zu ladenen Datei.
- taste = Enthält die höchstmögliche Anzahl von Stellen oder Buchstaben für inkey\$-Routine.
- xy = Koordinaten für LOCATE- Befehl bei Ausgabe der durch die inkey\$-Routine eingegebenen Zeichen.
- text\$ = Enthält den durch die inkey\$-Routine eingelesenen Text.
- ps = Wert für den angewählten Menüpunkt.
- unten = Enthält den Wert der zuletzt angezeigten Datenzeile.
- beginn = Enthält den Wert der obersten angezeigten Datenzeile.
- i = Hilfsvariable für FOR-NEXT-Schleifen.

Damit wäre alles erläutert. Jetzt brauchen Sie nur noch das Programm abzutippen und zu starten, und schon werden Ihre Finanzen viel leichter durchschau- und organisierbar.

(M. Megelin/JB)

Buchführung Januar 1999					
Datum	Gegenstand	Art	Einnahmen	Ausgaben	Haush.Best. Bank-Best.
1.	Obertrag				1000.00 2000.00
	Miete	B		845.23	1000.00 1154.77
	Ausgabe 1	H		127.98	872.02 1154.77
	Ausgabe 2	H		65.87	806.15 1154.77
	Ausgabe 3	H		42.96	763.19 1154.77
2.	Einnahme 1	H	100.00		863.19 1154.77
	Ausgabe 4	H		498.00	365.19 1154.77
	Ausgabe 5	H		12.75	352.44 1154.77
	Ausgabe 6	H		68.00	284.44 1154.77
	Euroscheck	T	500.00		784.44 654.77
	Strom/Gas	B		117.87	784.44 536.90
3.	Ausgabe 7	H		22.69	761.75 536.90
	Ausgabe 8	H		299.00	462.75 536.90
	Ausgabe 9	H		94.85	367.90 536.90
4.	Einnahme 2	B	2260.00		367.90 2796.90
	Ausgabe 10	H		200.00	167.90 2796.90
5.	Barabhebung	T	2000.00	2000.00	2167.90 796.90
			Haushalts-Ausgaben:	1432.10	DM
			Bank-Ausgaben:	3463.10	DM

Für 464-664-6128

```

10 ' ***** [1460]
20 ' * [175]
30 ' * Haushalts-Buchfuehrung V 3.1 * [1246]
40 ' * [175]
50 ' * (c) 1986 by Michael Megelin * [1035]
60 ' * [175]
70 ' * Schneider CPC 664/6128 * [1309]
80 ' * [175]
90 ' ***** [1460]
100 ' [117]
110 ON ERROR GOTO 310 [1551]
120 ON BREAK CONT [323]
    
```

Listing HaBu


```

130 ' [117]
140 ' *** Deutscher Zeichensatz *** [1511]
150 ' [117]
160 SYMBOL AFTER 90 [1214]
170 SYMBOL 91,102,24,60,102,126,102,102 [2414]
180 SYMBOL 92,102,60,102,102,102,102,60 [2425]
190 SYMBOL 93,102,68,102,102,102,102,60 [2091]
200 SYMBOL 95,0,0,0,0,0,0,126 [1908]
210 SYMBOL 123,204,0,120,12,124,204,118 [2340]
220 SYMBOL 124,102,0,60,102,102,102,60 [1824]
230 SYMBOL 125,102,0,102,102,102,102,62 [2270]
240 SYMBOL 126,60,102,102,108,102,102,108, [2862]
96
250 KEY DEF 17,1,125,93 [746]
260 KEY DEF 19,1,123,91 [1140]
270 KEY DEF 22,1,124,92 [1264]
280 KEY DEF 26,1,124,92,64 [765]
290 KEY DEF 24,1,126,94,0 [1480]
300 OPENOUT "dummy":MEMORY HIMEM-1:CLOSEOU [2303]
T
310 disk=0:zahl=200 [406]
320 DIM day(zahl),g$(zahl),hb$(zahl),einb( [7689]
zahl),einb(zahl),aush(zahl),ausb(zahl),bes
th(zahl),bestb(zahl),gesaush(zahl),gesausb
(zahl)
330 DIM art$(10),mon$(12) [1283]
340 MODE 2:RESTORE 350:FOR i=1 TO 10:READ [2991]
art$(i):NEXT
350 DATA Eingabe,Edit,Cls,Insert,Aktuell,S [6109]
peichern,Laden,Drucken,Hilfe,Ende
360 RESTORE 370:FOR i=1 TO 12:READ mon$(i) [2986]
:NEXT
370 DATA Januar,Februar,M(rz,Apri,Mai,Jun [5404]
i,Juli,August,September,Oktober,November,D
ezember
380 MOVE 0,376,1:DRAW 640,376:MOVE 0,378:D [2770]
RAW 640,378
390 WINDOW #1,1,80,1,1 [995]
400 ps=1:PRINT #1," ";CHR$(24);art$(1);CHR [3491]
$(24);" ";:FOR i=2 TO 10:PRINT #1," ";art$(
i);" ";:NEXT
410 MOVE 16,360:DRAW 624,360:MOVE 4,358:DR [3936]
AW 636,358:DRAW 636,88:DRAW 4,88:DRAW 4,35
8
420 FOR i=1 TO 17:MOVE 4,357-i,1,0:DRAW 63 [1894]
6,357-i:NEXT
430 MOVE 638,350:DRAW 638,96:MOVE 2,96:DRA [2817]
W 2,350
440 TAG:MOVE 8,356,1,1:PRINT "Datum Gegens [7521]
tand Art Einnahmen Ausgaben
Haush.-Best. Bank-Best. ":TAGOFF
450 LOCATE 2,19:PRINT CHR$(243) [2842]
460 WINDOW #0,3,79,5,19 [1071]
470 MOVE 4,88,1:DRAW 4,4:DRAW 636,4:DRAW 6 [2700]
36,88
480 MOVE 324,4:DRAW 324,88 [1022]
490 MOVE 2,80:DRAW 2,12:MOVE 638,80:DRAW 6 [3427]
38,12:MOVE 16,2:DRAW 308,2:MOVE 340,2:DRAW
624,2
500 WINDOW #2,2,40,21,24:WINDOW #3,42,79,2 [2356]
1,24
510 PRINT #3, " Haushaltsbestand: [2782]
DM "
520 PRINT #3,CHR$(24);" Bankbestand: [3511]
DM ";CHR$(24);
530 PRINT #3, " Haush.-Ausgaben: [2807]
DM "
540 PRINT #3,CHR$(24);" Bank-Ausgaben: [3796]
DM ";CHR$(24);
550 ' [117]
560 ' *** MENUE-STEUERUNG *** [1359]
570 ' [117]
580 IF INKEY(0)=0 THEN GOSUB 690 [1693]
590 IF INKEY(2)=0 THEN GOSUB 760 [924]
600 IF INKEY(1)=0 THEN GOSUB 860 [1712]
610 IF INKEY(8)=0 THEN GOSUB 910 [2022]
620 IF INKEY(9)=0 OR INKEY(18)=0 OR INKEY( [3696]
6)=0 THEN CLEAR INPUT:GOTO 640
630 GOTO 580 [448]
640 ON ps GOSUB 1190,2000,2420,2640,1070,2 [3037]
870,3070,3260,3670,3830
650 GOTO 580 [448]
660 ' [117]
670 ' *** SCROLLUP *** [897]
680 ' [117]
690 IF unten>=anz+15 OR unten=anz THEN RET [1576]
URN

```

Listing HaBu

```

700 LOCATE 1,15:PRINT CHR$(10) [1314]
710 unten=unten+1:IF unten>anz THEN RETURN [2444]
720 nr=unten:LOCATE 1,15:GOSUB 1890:PRINT: [3253]
RETURN
730 ' [117]
740 ' *** SCROLLDOWN *** [1119]
750 ' [117]
760 IF unten<1 THEN RETURN [1304]
770 LOCATE 1,1:PRINT CHR$(11) [832]
780 unten=unten-1:IF unten=14 THEN LOCATE [3768]
1,1:GOSUB 820:RETURN
790 IF unten-14<1 THEN RETURN [1890]
800 nr=unten-14:LOCATE 1,1:GOSUB 1890:PRIN [3377]
T:RETURN
810 : [174]
BUCHF]HRUNG V3.1 [5101]
f|r ";mon$(monat);jahr;TAB(55);CHR$(164);"
":RETURN
830 ' [117]
840 ' *** MENUESTEUERUNG RECHTS *** [1967]
850 ' [117]
860 IF ps<10 THEN ps=ps+1 ELSE RETURN [729]
870 GOSUB 960:RETURN [2243]
880 ' [117]
890 ' *** MENUESTEUERUNG LINKS *** [1686]
900 ' [117]
910 IF ps>1 THEN ps=ps-1 ELSE RETURN [717]
920 GOSUB 960:RETURN [2243]
930 ' [117]
940 ' *** STATUSZEILE *** [1158]
950 ' [117]
960 alt$=art$(ps) [502]
970 art$(ps)=CHR$(24)+art$(ps)+CHR$(24) [2110]
980 LOCATE #1,1,1 [571]
990 FOR i=1 TO 10 [315]
1000 PRINT #1," "art$(i)" "; [1211]
1010 NEXT i [375]
1020 art$(ps)=alt$ [754]
1030 RETURN [555]
1040 ' [117]
1050 ' *** AKTUALISIERUNG *** [1350]
1060 ' [117]
1070 IF anz<15 THEN beginn=1 ELSE beginn=a [2133]
nz-14
1080 CLS #0:IF monat>0 THEN LOCATE 1,14:GO [3196]
SUB 820
1090 LOCATE #2,12,2:PRINT #2,"Aktualisieru [3571]
ng!"
1100 FOR nr=beginn TO anz [1477]
1110 GOSUB 1880:PRINT [858]
1120 NEXT [350]
1130 unten=anz [1070]
1140 GOSUB 1710 [869]
1150 CLS #2:RETURN [616]
1160 ' [117]
1170 ' *** NEUEINGABE *** [817]
1180 ' [117]
1190 anz=anz+1:IF anz>1 THEN 1420 [1995]
1200 LOCATE #2,2,1:PRINT #2," Monat: ___ " [2055]
1210 taste=2:x=9:y=1:GOSUB 3940 [2048]
1220 IF text$="" THEN anz=anz-1:RETURN [1883]
1230 monat=VAL(text$):IF monat<1 OR monat> [3694]
12 THEN CLS #2:GOTO 1200
1240 LOCATE #2,15,1:PRINT #2," Jahr: 19__" [1586]
1250 taste=2:x=23:y=1:GOSUB 3940 [2843]
1260 IF VAL(text$)<86 THEN 1240 ELSE jahr= [3202]
VAL("19"+text$)
1270 LOCATE #2,2,3:PRINT #2," Tag: ___ " [1721]
1280 taste=2:x=9:y=3:GOSUB 3940 [1598]
1290 day(1)=VAL(text$):IF day(1)<1 OR day( [2624]
1)>31 THEN 1270
1300 g$(1)="]bertrag":hb$(1)="" [903]
1310 CLS #2:LOCATE #2,2,1:PRINT #2," Haush [3251]
altsbestand: ___ "
1320 taste=8:x=21:y=1:GOSUB 3940 [1846]
1330 besth(1)=VAL(text$) [790]
1340 LOCATE #2,2,3:PRINT #2," Bankbestand: [2365]
___ "
1350 taste=8:x=16:y=3:GOSUB 3940 [2247]
1360 bestb(1)=VAL(text$) [1330]
1370 LOCATE 1,15:GOSUB 820 [1111]
1380 GOTO 1700 [359]
1390 ' [117]
1400 ' *** EINGABE *** [1674]
1410 ' [117]
1420 CLS #2 [372]
1430 IF unten=anz-1 THEN 1490 [826]

```

Listing HaBu

Anwendungen

```

1440 LOCATE 1,15:IF anz=nr THEN beginn=unt [4257]
en+1 ELSE beginn=unt
1450 FOR nr=beginn TO anz-1 [1488]
1460 GOSUB 1880:PRINT [858]
1470 NEXT [350]
1480 unten=anz-1 [1077]
1490 LOCATE #2,2,1:PRINT #2," Tag: [3737]
Noch";day(anz-1);".";monat;".? (ENTER)"
1500 taste=2:x=9:y=1:GOSUB 3940 [2048]
1510 IF text$=" " THEN anz=anz-1:CLS #2:RE [2277]
TURN
1520 day(anz)=VAL(text$):IF day(anz)<0 OR [3139]
day(anz)>31 THEN 1420
1530 IF day(anz)=0 THEN day(anz)=day(anz-1 [1207]
)
1540 LOCATE #2,2,3:PRINT #2," Gegenstand: [3166]
"
1550 taste=20:x=14:y=3:GOSUB 3940 [2920]
1560 g$(anz)=text$ [571]
1570 PRINT #2:PRINT #2," Bank-oder Hausha [4221]
ltsbewegung: _":PRINT #2
1580 taste=1:x=31:y=3:GOSUB 3940 [2389]
1590 hb$(anz)=UPPER$(text$):IF hb$(anz)=" [2218]
THEN hb$(anz)="H"
1600 IF hb$(anz)<>"H" AND hb$(anz)<>"B" AN [4346]
D hb$(anz)<>"T" THEN 1570
1610 LOCATE #2,2,3:PRINT #2," Einnahmen: _ [2551]
"
1620 taste=8:x=13:y=3:GOSUB 3940 [1954]
1630 IF hb$(anz)="H" THEN einh(anz)=VAL(te [2220]
xt$)
1640 IF hb$(anz)="B" THEN einb(anz)=VAL(te [1876]
xt$)
1650 IF hb$(anz)="T" THEN einh(anz)=VAL(te [6425]
xt$):aush(anz)=einh(anz):aush(anz)=0:einb(
anz)=0:GOTO 1700
1660 LOCATE #2,2,3:PRINT #2," Ausgaben: _ [3095]
"
1670 taste=8:x=13:y=3:GOSUB 3940 [1954]
1680 IF hb$(anz)="H" THEN aush(anz)=VAL(te [1847]
xt$)
1690 IF hb$(anz)="B" THEN ausb(anz)=VAL(te [2122]
xt$)
1700 CLS #2:nr=anz:LOCATE 1,15:PRINT CHR$( [4083]
10);:GOSUB 1790:GOSUB 1880:PRINT:unten=nr
1710 LOCATE #3,24,1:PRINT #3,USING "#####. [2967]
##";besth(anz)
1720 LOCATE #3,24,2:PRINT #3,CHR$(24);:PRI [4571]
NT #3,USING "#####.##";bestb(anz);:PRINT #
3,CHR$(24)
1730 LOCATE #3,24,3:PRINT #3,USING "#####. [2311]
##";gesaush(anz)
1740 LOCATE #3,24,4:PRINT #3,CHR$(24);:PRI [4094]
NT #3,USING "#####.##";gesausb(anz);:PRINT
#3,CHR$(24)
1750 RETURN [555]
1760 ' [117]
1770 ' *** BESTAENDE UND AUSGABEN BERECHNE [3123]
N ***
1780 ' [117]
1790 IF nr=1 THEN RETURN [1065]
1800 besth(nr)=besth(nr-1)+einh(nr)-aush(n [3313]
r)
1810 bestb(nr)=bestb(nr-1)+einb(nr)-ausb(n [2873]
r)
1820 gesaush(nr)=gesaush(nr-1)+aush(nr) [2079]
1830 gesausb(nr)=gesausb(nr-1)+ausb(nr) [2124]
1840 RETURN [555]
1850 ' [117]
1860 ' *** BILDSCHIRMAUSDRUCK *** [1965]
1870 ' [117]
1880 PRINT:LOCATE 1,15 [1402]
1890 IF day(nr)<>day(nr-1) THEN PRINT USIN [3716]
G "##.";day(nr);
1900 PRINT TAB(6);g$(nr); [1188]
1910 PRINT TAB(28);hb$(nr);" "; [1049]
1920 IF einh(nr)>0 OR einb(nr) THEN PRINT [5367]
TAB(32) USING "#####.##";einh(nr)+einb(nr)
;
1930 IF aush(nr)>0 OR ausb(nr) THEN PRINT [3295]
TAB(44) USING "#####.##";aush(nr)+ausb(nr)
;
1940 PRINT TAB(56) USING "#####.##";besth [2138]
(nr);
1950 PRINT TAB(68) USING "#####.##";bestb [1719]
(nr);
1960 RETURN [555]

```

Listing HaBu

```

1970 ' [117]
1980 ' *** BERICHTIGEN *** [1315]
1990 ' [117]
2000 IF unten<2 THEN RETURN [1079]
2010 CLS #2:LOCATE #2,2,1:PRINT #2,"Tag: [3800]
"day(unten)".mon$(monat)
2020 LOCATE #2,2,3:PRINT #2,"ENTER oder Ne [2267]
ueingabe: _"
2030 taste=2:x=24:y=3:GOSUB 3940 [2715]
2040 IF text$=" " THEN CLS #2:RETURN [1422]
2050 IF text$<>" " THEN day(unten)=VAL(text [1839]
$)
2060 CLS #2:LOCATE #2,2,1:PRINT #2,"Gegens [4795]
tand: ";g$(unten)
2070 LOCATE #2,8,3:PRINT #2," [2977]
"
2080 taste=20:x=7:y=3:GOSUB 3940 [3083]
2090 IF text$<>" " THEN g$(unten)=text$ [1690]
2100 CLS #2:PRINT #2," Kontenbewegung: ";h [3283]
b$(unten)
2110 LOCATE #2,2,3:PRINT #2,"(H,B oder T): [2785]
"
2120 hb=0:taste=1:x=16:y=3:GOSUB 3940 [1778]
2130 IF text$=" " THEN 2150 [1165]
2140 text$=UPPER$(text$):IF text$<>" " AND [7220]
(text$="H" OR text$="B" OR text$="T") THEN
hb$(unten)=text$:hb=1: ELSE GOTO 2100
2150 CLS #2:PRINT #2," Einnahme: ";einh(un [4784]
ten)+einb(unten);"DM"
2160 LOCATE #2,8,3:PRINT #2," [1918]
DM"
2170 taste=8:x=7:y=3:GOSUB 3940 [999]
2180 IF text$=" " AND hb$(unten)="T" THEN 2 [2328]
280
2190 un=unten [506]
2200 IF hb$(unten)="H" THEN IF text$<>" " T [9427]
HEN einh(unten)=VAL(text$):einb(unten)=0:E
LSE einh(unten)=einb(unten)+einh(unten):ei
nb(unten)=0
2210 IF hb$(unten)="B" THEN IF text$<>" " T [11218]
HEN einb(unten)=VAL(text$):einb(unten)=0:E
LSE einb(unten)=einb(unten)+einh(unten):ei
nh(unten)=0
2220 IF hb$(unten)="T" THEN IF text$<>" " THEN [9940]
einh(unten)=VAL(text$):ausb(unten)=einh(un):ein
b(unten)=0:aush(unten)=0:ELSE einh(unten)=einh(un)+
einb(unten)+aush(unten)+ausb(unten):ausb(unten)=einh(u
n):einb(unten)=0:aush(unten)=0
2230 CLS #2:PRINT #2," Ausgaben: ";aush(un [4972]
ten)+ausb(unten);"DM"
2240 LOCATE #2,8,3:PRINT #2," [1918]
DM"
2250 taste=8:x=7:y=3:GOSUB 3940 [999]
2260 IF text$<>" " AND hb$(unten)="H" THEN [5819]
aush(unten)=VAL(text$):ausb(unten)=0
2270 IF text$<>" " AND hb$(unten)="B" THEN [4593]
ausb(unten)=VAL(text$):aush(unten)=0
2280 IF hb=0 THEN 2320 [1123]
2290 IF hb$(unten)="H" THEN einh(unten)=ei [8732]
nh(unten)+einb(unten):aush(unten)=aush(un
ten)+ausb(unten):einb(unten)=0:aush(unten)=
0
2300 IF hb$(unten)="B" THEN einb(unten)=ei [10043]
nh(unten)+einb(unten):ausb(unten)=aush(un
ten)+ausb(unten):einb(unten)=0:aush(unten)=
0
2310 IF hb$(unten)="T" THEN IF einh(unten) [5443]
=0 THEN einh(unten)=ausb(unten) ELSE ausb(
unten)=einh(unten)
2320 CLS #2:LOCATE 1,15:PRINT CHR$(11) [2440]
2330 FOR nr=unten TO anz [2026]
2340 GOSUB 1790 [853]
2350 NEXT [350]
2360 nr=unten:GOSUB 1880:PRINT [3372]
2370 GOSUB 1710 [869]
2380 RETURN [555]
2390 ' [117]
2400 ' *** CLS (Loeschen) *** [1965]
2410 ' [117]
2420 CLS #2:LOCATE #2,2,2:PRINT #2,"";g$( [3594]
unten);" lischen?"
2430 a$=INKEY$:IF a$=" " THEN 2430 [1458]
2440 IF a$=" " THEN CLS #2:RETURN [827]
2450 IF un=anz THEN 2550 [1108]
2460 FOR i=unten+1 TO anz [1602]
2470 day(i-1)=day(i) [1294]
2480 g$(i-1)=g$(i) [1039]
2490 hb$(i-1)=hb$(i) [248]
2500 einh(i-1)=einh(i) [1007]

```

Listing HaBu

```

2510 einb(i-1)=einb(i) [1142]
2520 aush(i-1)=aush(i) [1268]
2530 ausb(i-1)=ausb(i) [520]
2540 NEXT [350]
2550 anz=anz-1 [1014]
2560 FOR nr=unten TO anz [2026]
2570 GOSUB 1790 [853]
2580 NEXT [350]
2590 hb$(anz+1)="":einh(anz+1)=0:einb(anz+1)=0:aush(anz+1)=0:ausb(anz+1)=0 [4276]
2600 GOSUB 1710:CLS #2:GOTO 760 [2028]
2610 ' [117]
2620 ' *** INSERT (Einfuegen) *** [1198]
2630 ' [117]
2640 CLS #2:LOCATE #2,2,2:PRINT #2,"Hier (Pfeil!) Zeile einfügen? (ENTER)" [3354]
2650 a$=INKEY$:IF a$="" THEN 2650 [1558]
2660 IF a$=" " THEN CLS #2:RETURN [827]
2670 FOR i=anz TO unten+1 STEP -1 [1818]
2680 day(i+1)=day(i):day(i)=0 [1228]
2690 g$(i+1)=g$(i):g$(i)=" [1269]
2700 hb$(i+1)=hb$(i):hb$(i)=" [1114]
2710 einh(i+1)=einh(i):einb(i)=0 [1747]
2720 einb(i+1)=einb(i):einb(i)=0 [1577]
2730 aush(i+1)=aush(i):aush(i)=0 [1804]
2740 ausb(i+1)=ausb(i):ausb(i)=0 [1021]
2750 NEXT [350]
2760 anz1=anz:anz=unten+1 [1770]
2770 CLS #2:GOSUB 1490 [571]
2780 CLS #2:LOCATE #2,10,2:PRINT #2,"Neube  
rechnung!" [2342]
2790 anz=anz1+1 [956]
2800 FOR nr=unten TO anz [2026]
2810 GOSUB 1790 [853]
2820 NEXT [350]
2830 GOSUB 1710:CLS #2:RETURN [2069]
2840 ' [117]
2850 ' *** DATEN SPEICHERN *** [1438]
2860 ' [117]
2870 IF anz<1 THEN RETURN [972]
2880 CLS #2:PRINT #2," S P E I C H E R N !" [1904]
2890 LOCATE #2,2,3:PRINT #2,"Name der Date  
i: _____ oder ENTER." [5533]
2900 taste=8:x=18:y=3:GOSUB 3940 [2158]
2910 IF text$=" " THEN CLS #2:RETURN [1422]
2920 LOCATE #2,2,3:PRINT #2,"Daten werden  
abgespeichert..." [4540]
2930 IF text$="" THEN name$="BUCHDAT" ELSE  
name$=text$ [2072]
2940 name$="!"+"name$+".bfg" [1031]
2950 disk=1 [152]
2960 SPEED WRITE 1 [1302]
2970 OPENOUT name$ [700]
2980 PRINT #9,anz,monat,jahr,besth(1),best  
b(1) [3231]
2990 FOR i=1 TO anz [856]
3000 PRINT #9,day(i),einh(i),einb(i),aush(i),ausb(i) [1963]
3010 PRINT #9,g$(i):PRINT #9,hb$(i) [2478]
3020 NEXT [350]
3030 CLOSEOUT [902]
3040 CLS #2 [372]
3050 RETURN [555]
3060 ' [117]
3070 ' *** DATEN LADEN *** [1264]
3080 ' [117]
3090 CLS #2:PRINT #2," L A D E N !" [1842]
3100 LOCATE #2,2,3:PRINT #2,"Name der Date  
i: _____ oder ENTER." [5533]
3110 taste=8:x=18:y=3:GOSUB 3940 [2158]
3120 IF text$=" " THEN CLS #2:RETURN [1422]
3130 LOCATE #2,2,3:PRINT #2,"Daten werden  
geladen... Bitte warten." [4635]
3140 IF text$="" THEN name$="BUCHDAT" ELSE  
name$=text$ [2072]
3150 name$="!"+"name$+".bfg" [1031]
3160 OPENIN name$ [1050]
3170 INPUT #9,anz,monat,jahr,besth(1),best  
b(1) [2447]
3180 FOR i=1 TO anz [856]
3190 INPUT #9,day(i),einh(i),einb(i),aush(i),ausb(i) [3945]
3200 LINE INPUT #9,g$(i):LINE INPUT #9,hb$(i) [3150]
3210 NEXT [350]
3220 CLOSEIN [752]

```

Listing HaBu

```

3230 FOR nr=1 TO anz:GOSUB 1790:NEXT [2035]
3240 CLS #2:GOSUB 1070:unten=anz [2167]
3250 RETURN [555]
3260 ' [117]
3270 ' *** DRUCKEN *** [1736]
3280 ' [117]
3290 IF INP(&F500)=90 THEN CLS #2:LOCATE #  
2,6,2:PRINT #2,CHR$(7);"Drucker nicht eing  
eschaltet!":FOR t=1 TO 2500:NEXT:CLS #2:RE  
TURN [7624]
3300 GOTO 4040 [497]
3310 CLS #2:LOCATE #2,2,2:PRINT #2,"Drucke  
r und Papier o.k.? (ENTER)" [4651]
3320 a$=INKEY$:IF a$="" THEN 3320 [1433]
3330 CLS #2:IF a$<>CHR$(13) THEN RETURN [917]
3340 PRINT #8,CHR$(27);"x";CHR$(1); [1737]
3350 ab$=CHR$(9)+CHR$(9):abst$=ab$:zr$=CHR  
$(13)+CHR$(10) [2952]
3360 PRINT #8,CHR$(14);:" Breitschrift [1630]
3370 PRINT #8,CHR$(27);"-";CHR$(1);"  
Buchführung";mon$(monat);jahr;CHR$(27)  
;"-";CHR$(0);zr$; [6411]
3380 PRINT #8,CHR$(27);"8";:" Aufhebung de  
s Papierenddetektors [4094]
3390 PRINT #8,CHR$(19); [1313]
3400 PRINT #8,CHR$(27);"M";zr$;:" Einstell  
ung der Elite-Schrift [3257]
3410 PRINT #8,ab$;"Datum Gegenstand [6040]
Art Einnahmen Ausgaben Haush.Best.  
Bank-Best.";zr$; [975]
3420 PRINT #8,zr$; [3623]
3430 CLS #2:LOCATE #2,2,1:PRINT #2," 0 vo  
n";anz;"Eingaben ausgedruckt." [856]
3440 FOR i=1 TO anz [3718]
3450 PRINT #8,ab$;:IF day(i)<>day(i-1) T  
HEN PRINT #8,TAB(2);USING "###.";day(i); [2047]
3460 PRINT #8,TAB(7);g$(i); [1084]
3470 PRINT #8,TAB(30);hb$(i); [4670]
3480 IF einh(i)+einb(i)>0 THEN PRINT #8,  
TAB(34);USING "#####.###";einh(i)+einb(i); [5260]
3490 IF aush(i)+ausb(i)>0 THEN PRINT #8,  
TAB(44);USING "#####.###";aush(i)+ausb(i); [2611]
3500 PRINT #8,TAB(56);USING "#####.###";  
besth(i); [2854]
3510 PRINT #8,TAB(68);USING "#####.###";  
bestb(i);:PRINT #8,zr$; [1853]
3520 LOCATE #2,2,1:PRINT #2,USING "###";  
i [6168]
3530 IF i/58=INT(i/58) THEN IF i/29=INT(  
i/29) THEN ab$=CHR$(13):GOSUB 3600 ELSE ab  
$=abst$:GOSUB 3600 [350]
3540 NEXT [975]
3550 PRINT #8,zr$; [7609]
3560 PRINT #8,ab$;TAB(20);"Haushalts-Ausga  
ben: ";:PRINT #8,USING "#####.## ";gesaush  
(anz);:PRINT #8," DM";zr$; [5991]
3570 PRINT #8,ab$;TAB(25);"Bank-Ausgaben:  
";:PRINT #8,USING "#####.## ";gesausb(anz)  
;:PRINT #8," DM";zr$; [372]
3580 CLS #2 [555]
3590 RETURN [4359]
3600 LOCATE #2,3,3:PRINT #2,"Bitte Papier  
wenden, anschl. ENTER" [806]
3610 IF INKEY$="" THEN 3610 [2497]
3620 LOCATE #2,1,3:PRINT #2,SPACE$(36) [555]
3630 RETURN [117]
3640 ' [573]
3650 ' *** HILFE *** [117]
3660 ' [356]
3670 CLS:PRINT [6856]
3680 PRINT " Arten der Kontenbewegung: -  
H - Einnahme oder Ausgabe f|r Haushaltsko  
nto" [3717]
3690 PRINT TAB(30);"- B - Einnahme oder Au  
sgabe f|r Bankkonto" [8248]
3700 PRINT TAB(30);"- T - Transfer: Ausgab  
e f|r Bankkonto; Einnahme";TAB(36);"f|r Ha  
ushaltskonto" [5173]
3710 PRINT " Tastenfunktionen:";TAB  
(30);"-DEL- Normale L|schfunktion" [4172]
3720 PRINT TAB(30);"-SPACE- Ausstieg aus a  
ngew{hlter Men}funktion" [3467]
3730 PRINT TAB(30);"- p - Daten nach obe  
n scrollen" [3111]
3740 PRINT TAB(30);"- q - Daten nach unt  
en scrollen" [4336]
3750 PRINT TAB(30);"- r - s - Men|punkte a

```

Listing HaBu

```

nw{hlen"
3760 PRINT TAB(30);"-COPY- oder -ENTER-";T [5209]
AB(36);"Aktivierung des angew{hlten Men|pu
nktes"
3770 LOCATE #2,2,2:PRINT #2,"Zur|ck ins Me [5692]
nl mit beliebiger Taste."
3780 IF INKEY$="" THEN 3780 [588]
3790 CLS:CLS #2:GOTO 1070 [431]
3800 ' [117]
3810 ' *** BEENDEN *** [1421]
3820 ' [117]
3830 IF disk=1 THEN LOCATE #2,4,2:PRINT #2 [4939]
,"Programm wirklich beenden? (J/ )":GOTO 3
890
3840 LOCATE #2,8,1:PRINT #2,"Daten nicht a [3623]
bgespeichert!!!"
3850 LOCATE #2,12,3:PRINT #2,"Abspeichern? [3051]
( /N)"
3860 a$=INKEY$:IF a$="" THEN 3860 [1340]
3870 IF UPPER$(a$)="N" THEN disk=1:CLS #2: [3947]
GOTO 3830 ELSE GOSUB 2870
3880 MODE 2:|DIR,"*.bfg":PRINT "Neustart o [6288]
hne Datenverlust mit 'GOTO 340'.":PRINT:EN
D
3890 a$=INKEY$:IF a$="" THEN 3890 [1534]
3900 IF UPPER$(a$)="J" THEN 3880 ELSE RETU [1577]
RN
3910 ' [117]
3920 ' *** INKEY$-ROUTINE *** [1546]
3930 ' [117]
3940 CLEAR INPUT:text$="" [936]
3950 FOR anzahl=1 TO taste [981]
3960 a$=INKEY$:IF a$="" THEN 3960 [1352]
3970 IF a$=CHR$(13) THEN RETURN [1139]
3980 IF a$=CHR$(127) AND anzahl>1 THEN anz [6271]
ahl=anzahl-1:LOCATE #2,x+anzahl,y:PRINT #2
,"":text$=LEFT$(text$,anzahl-1):GOTO 3960
3990 LOCATE #2,x+anzahl,y:PRINT #2,a$ [1630]
4000 IF anzahl>3 AND taste<10 THEN IF MID$ [3158]

```

Listing HaBu

```

(text$,anzahl-2,1)="." THEN anzahl=taste
4010 text$=text$+a$:IF text$=" " THEN RETU [2625]
RN
4020 NEXT [350]
4030 RETURN [555]
4040 ' Konzeptdruck [1209]
4050 CLS #2:LOCATE #2,2,2:PRINT #2,"Probed [4630]
ruck (P) o. Sch|nschrift (ENTER)"
4060 a$=INKEY$:IF a$="" THEN 4060 [1508]
4070 IF UPPER$(a$)<>"P" THEN 3310 [1479]
4080 CLS #2:LOCATE #2,2,2:PRINT #2,"Drucke [4651]
r und Papier o.k.? (ENTER)"
4090 a$=INKEY$:IF a$="" THEN 4090 [1494]
4100 CLS #2:IF a$<>CHR$(13) THEN RETURN [917]
4110 PRINT #8,CHR$(27)"@" [1408]
4120 zr$=CHR$(13)+CHR$(10) [3237]
4130 PRINT #8,CHR$(27);"-";CHR$(1);" [6227]
Buchf|hrung ";mon$(monat);jahr;CHR$(27)
; "-";CHR$(0);zr$
4140 PRINT #8,ab$;"Datum Gegenstand [6040]
Art Einnahmen Ausgaben Haush.Best.
Bank-Best.";zr$;
4150 PRINT #8,zr$ [881]
4160 FOR i=1 TO anz [856]
4170 PRINT #8,ab$;:IF day(i)<>day(i-1) T [3718]
HEN PRINT #8,TAB(2);USING "##.";day(i);
4180 PRINT #8,TAB(7);g$(i); [2047]
4190 PRINT #8,TAB(30);hb$(i); [1084]
4200 IF einh(i)+einb(i)>0 THEN PRINT #8, [4670]
TAB(34);USING "#####.##";einh(i)+einb(i);
4210 IF aush(i)+ausb(i)>0 THEN PRINT #8, [5260]
TAB(44);USING "#####.##";aush(i)+ausb(i);
4220 PRINT #8,TAB(56);USING "#####.##"; [2611]
besth(i);
4230 PRINT #8,TAB(68);USING "#####.##"; [2854]
bestb(i);:PRINT #8,zr$;
4240 NEXT [350]
4250 CLS #2 [372]
4260 RETURN [555]

```

Listing HaBu

TESTME – Vokabeltrainer

Bei meinem Programm »TESTME« handelt es sich um einen Vokabeltrainer. Nach dem Starten des Programmes erscheint zunächst der Lader. Beim eigentlichen Programm erscheint zuerst das Menue, welches aus 4 Feldern besteht, und einem kleinen Pfeil links unten. Um eine von den 4 Optionen (Vokabeln laden = Vokabeln eingeben = Vokabeln abfragen) anwählen zu können, müssen Sie den Pfeil mit Hilfe des Joysticks auf eines der 4 Felder bewegen und dann den Feuer-Knopf drücken. Wenn Sie die Option »Vokabeln speichern« gewählt haben, dann werden Sie nach dem Namen, bzw. Dateinamen gefragt. Nach Abschluß der Eingabe durch <ENTER> werden die gesamten Vokabeln, die sich im Speicher befinden, als ASCII-Datei abgespeichert. Danach wird automatisch ins Menue zurückgesprungen. Wenn Sie die Option »Vokabeln laden« gewählt haben, dann wird wieder nach dem Dateinamen gefragt, die gesamten Vokabeln, die sich noch im Speicher befinden, werden gelöscht und die Vokabeln aus der Datei werden in den Speicher kopiert und können nun durch die Option »Vokabeln abfragen« abgefragt werden. Wieder wird automatisch ins Menue zurückgesprungen. Wenn Sie die Option »Vokabeln eingeben« gewählt haben, dann erscheint nach kurzer Zeit links oben die Nummer der Vokabel, 50 Vokabeln können maximal eingegeben werden, und 2 Felder. In dem ersten Feld befindet sich ein blinkender Cursor: es wird erst die deutsche Übersetzung der Vokabel abgefragt. Bei Eingabe der Vokabel ist darauf zu achten, daß beim Editie-

ren nur die DEL-Taste funktioniert, da die Abfrage nicht über »Input« verläuft; nach Abschluß der Eingabe durch <ENTER> springt der Cursor in das nächste Feld: es wird nun die Übersetzung der Vokabel verlangt. Nach Abschluß dieser Vokabel durch <ENTER> werden Sie gefragt, ob Sie noch weitere Vokabeln eingeben möchten oder nicht. Wenn Sie »j« für »Ja« drücken geht es mit der Abfrage weiter, wenn Sie aber »n« für »Nein« drücken wird ins Menue zurückgesprungen. Wenn Sie die Option »Vokabeln abfragen« gewählt haben, so erscheinen wieder 2 Felder, jedoch in dem ersten entweder die deutsche Vokabel oder die Übersetzung. Sie werden nun in dem zweiten Feld nach der Übersetzung gefragt. Wieder Abschluß der Eingabe durch <ENTER>. Ist diese Eingabe falsch, so wird ein Fehlerton ausgegeben und die richtige Übersetzung angezeigt, ansonsten wird gefragt, ob weiter abgefragt werden soll oder nicht. Wieder »n« für »Nein« und »j« für »Ja«.

Die Eingabe

Zuerst müssen die Listings 2 und 3 eingegeben, gesichert und gestartet werden. Sie erzeugen die Binärfiles TESTME1.BIN und TESTME2.BIN, die vom Hauptprogramm gebraucht werden. Nachdem Sie dieses (Listing 1) abgetippt und gesichert haben, können Sie das Programm starten.

(Andreas Martin)

ür 464-664-6128



```

10 hoch$=CHR$(10) [1157]
20 GOSUB 720:ON ERROR GOTO 960 [2307]
30 GOSUB 780 [921]
40 GOSUB 800 [820]
50 CALL &BC06,&CO:OUT &BC00,1:OUT &BD00,0: [3125]
PRINT REMAIN (1):MODE 1
60 PRINT [361]
70 PRINT " ";CHR$(150);STRING$(12,CHR$(154 [6059]
));CHR$(156);" ";CHR$(150);STRING$(11
,CHR$(154));CHR$(156);" ";
80 PRINT " ";CHR$(149);CHR$(150);STRING$(1 [7938]
0,CHR$(154));CHR$(156);CHR$(149);" ";
CHR$(149);CHR$(150);STRING$(9,CHR$(154));C
HR$(156);CHR$(149);" ";
90 PRINT " ";CHR$(149);CHR$(149);"VOKABELN [7613]
";CHR$(149);CHR$(149);" ";CHR$(149)
;CHR$(149);"VOKABELN ";CHR$(149);CHR$(149)
;" ";
100 PRINT " ";CHR$(149);CHR$(149);" [6357]
";CHR$(149);CHR$(149);" ";CHR$(149)
);CHR$(149);" ";CHR$(149);CHR$(149)
);" ";
110 PRINT " ";CHR$(149);CHR$(149);" SPEICH [8535]
ERN";CHR$(149);CHR$(149);" ";CHR$(149)
);CHR$(149);" ABFRAGEN";CHR$(149);CHR$(149)
);" ";
120 PRINT " ";CHR$(149);CHR$(147);STRING$( [7868]
10,CHR$(154));CHR$(153);CHR$(149);" "
;CHR$(149);CHR$(147);STRING$(9,CHR$(154));
CHR$(153);CHR$(149);" ";
130 PRINT " ";CHR$(147);STRING$(12,CHR$(15 [4135]
4));CHR$(153);" ";CHR$(147);STRING$(1
1,CHR$(154));CHR$(153);" ";
140 PRINT [361]
150 PRINT [361]
160 PRINT " ";CHR$(150);STRING$(10,CHR$(15 [5371]
4));CHR$(156);" ";CHR$(150);STRING$(
11,CHR$(154));CHR$(156);" ";
170 PRINT " ";CHR$(149);CHR$(150);STRING$( [8953]
8,CHR$(154));CHR$(156);CHR$(149);" "
";CHR$(149);CHR$(150);STRING$(9,CHR$(154))
;CHR$(156);CHR$(149);" ";
180 PRINT " ";CHR$(149);CHR$(149);"VOKABEL [7688]
N";CHR$(149);CHR$(149);" ";CHR$(149)
);CHR$(149);"VOKABELN ";CHR$(149);CHR$(149)
);" ";
190 PRINT " ";CHR$(149);CHR$(149);" [5233]
";CHR$(149);CHR$(149);" ";CHR$(149)
);CHR$(149);" ";CHR$(149);CHR$(149)
);" ";
200 PRINT " ";CHR$(149);CHR$(149);" LADEN [5198]
";CHR$(149);CHR$(149);" ";CHR$(149)
);CHR$(149);" EINGEBEN";CHR$(149);CHR$(149)
);" ";
210 PRINT " ";CHR$(149);CHR$(147);STRING$( [9385]
8,CHR$(154));CHR$(153);CHR$(149);" "
";CHR$(149);CHR$(147);STRING$(9,CHR$(154))
;CHR$(153);CHR$(149);" ";
220 PRINT " ";CHR$(147);STRING$(10,CHR$(15 [5793]
4));CHR$(153);" ";CHR$(147);STRING$(
11,CHR$(154));CHR$(153);" ";
230 PRINT CHR$(10);CHR$(10);CHR$(10);CHR$( [2620]
10)
240 OUT &BC00,1:FOR a=1 TO 40:OUT &BD00,a: [3518]
FOR b=1 TO 10:NEXT b,a
250 CALL &8000 [453]
260 IF FNfeld(64,240,304,368) THEN 310 [1813]

```

Listing TESTME

```

270 IF FNfeld(384,544,304,368) THEN 420 [1023]
280 IF FNfeld(64,208,176,240) THEN 510 [1912]
290 IF FNfeld(384,544,176,240) THEN 620 [1531]
300 GOTO 250 [419]
310 GOSUB 830:CALL &BC06,&40:CLS [2129]
320 LOCATE 17,12:PRINT"THEMA ???" [2157]
330 LOCATE 16,15:maxs=7:lens=0:g$="":GOSUB [3693]
860:them$=g$
340 OPENOUT "!"+them$ [732]
350 PRINT#9,n [810]
360 FOR a=1 TO n [630]
370 PRINT#9,qe$(a,0) [1199]
380 PRINT#9,qe$(a,1) [1193]
390 NEXT [350]
400 CLOSEOUT [902]
410 MODE 1:CALL &BC06,&CO:GOTO 250 [1929]
420 GOSUB 830:CALL &BC06,&40:MODE 1:RANDOM [2984]
IZE TIME
430 INK 1,0 [49]
440 x=INT(RND*n)+1:y=INT(RND*2):IF y=0 THE [15943]
N POKE pok,255:PRINT hoch$;hoch$:PRINT" DE
UTSCH: ":PRINT:POKE pok,0:GOSUB 920:PRINT
qe$(x,y):ELSE POKE pok,255:PRINT hoch$;hoc
h$:PRINT" UEBERSETZUNG: ":PRINT:POKE pok,0
:GOSUB 920:PRINT qe$(x,y)
450 IF y=0 THEN POKE pok,255:PRINT hoch$;h [11594]
och$:PRINT" UEBERSETZUNG: ":PRINT:POKE pok
,0:ELSE POKE pok,255:PRINT hoch$;hoch$:PRI
NT" DEUTSCH: ":PRINT:POKE pok,0
460 maxs=35:lens=0:g$="":GOSUB 920:INK 1,2 [3435]
6:GOSUB 850:TES$=g$
470 IF TES$(<>)qe$(x,1-y) THEN PRINT hoch$;h [5230]
och$:PRINT" FALSCH, ES MUSSTE HEISSEN: ":P
RINT:POKE pok,0:PRINT " ";qe$(x,1-y):POKE
pok,255
480 PRINT hoch$;hoch$:PRINT" WEITER ??? [j [2374]
/n]"
490 a$=INKEY$:IF a$="" THEN 490 [1374]
500 IF a$="j" OR a$="J" THEN FOR f=1 TO 25 [5133]
:PRINT hoch$:NEXT f:GOTO 420 ELSE IF a$="n
" OR a$="N" THEN MODE 1:CALL &BC06,&CO:GOT
O 250 : ELSE 490
510 GOSUB 830:CALL &BC06,&40:MODE 1 [1989]
520 LOCATE 17,12:PRINT"THEMA ???" [2157]
530 LOCATE 16,15:maxs=7:lens=0:g$="":GOSUB [3693]
860:them$=g$
540 ERASE qe$:DIM qe$(50,1) [1706]
550 OPENIN "!"+them$ [583]
560 INPUT #9,n [698]
570 FOR a=1 TO n [630]
580 LINE INPUT #9,qe$(a,0) [1954]
590 LINE INPUT #9,qe$(a,1) [1932]
600 NEXT [350]
610 CLOSEIN:MODE 1:CALL &BC06,&CO:GOTO 250 [2179]
620 GOSUB 830:CALL &BC06,&40:MODE 1:n=1 [2381]
630 ERASE qe$:DIM qe$(50,1) [1706]
640 MODE 1:INK 1,0:PRINT" NUMMER";n;:PRINT [4836]
" ";:PRINT hoch$;hoch$
650 PRINT" DEUTSCH: ":PRINT:GOSUB 920 [2688]
660 PRINT hoch$;hoch$;hoch$:PRINT" UEBERSE [6064]
TZUNG: ":PRINT:GOSUB 920
670 INK 1,26:FOR f=1 TO 8:PRINT CHR$(11);: [12190]
NEXT f:g$="":MAXS=35:LENS=0:GOSUB 850:QE$(
N,0)=G$:PRINT:PRINT CHR$(9);CHR$(9);:FOR f
=1 TO 7:PRINT CHR$(10);:NEXT f:g$="":MAXS=
35:LENS=0:GOSUB 850:QE$(N,1)=G$
680 PRINT hoch$;hoch$:PRINT" WEITER ??? [j [2374]
/n]"
690 a$=INKEY$:IF a$="" THEN 690 [1398]
700 IF (a$="j" OR a$="J") AND n<50 THEN n= [8537]

```

Listing TESTME

```

n+1:FOR f=1 TO 26:PRINT hoch$:NEXT f:GOTO
640 ELSE IF a$="n" OR a$="N" THEN MODE 1:C
ALL &BC06,&CO:GOTO 250 ELSE GOTO 690
710 END [110]
720 SYMBOL AFTER 65:SPEED WRITE 1:INK 0,0: [4835]
BORDER 6:INK 1,26:PEN 1:PAPER 0
730 mem=HIMEM+1:MEMORY &3FFF:LOAD"!TESTME2 [3618]
.BIN",mem-1
740 SYMBOL &9F,&80,&CO,&EO,&FO,&F8,&EO,&B0 [2702]
,&18
750 LOAD"!TESTME1.BIN",&8000 [1173]
760 KEY 140,"":GOSUB 750:POKE pok,255:CALL [5214]
&BC06,&CO:GOTO 40"+CHR$(13)
770 RETURN [555]
780 DEF FNfeld(x,x1,y,y1)=(PEEK(&8066)*256 [9168]
+PEEK(&8065)>=x AND PEEK(&8066)*256+PEEK(&
8065)<=x1 AND PEEK(&8068)*256+PEEK(&8067)>
y AND PEEK(&8068)*256+PEEK(&8067)<=y1)
790 CALL &BB48:IF PEEK(6)=&80 THEN pok=&B2 [2647]
95:RETURN ELSE pok=&B735:RETURN
800 DIM qe$(50,1) [519]
810 OPENOUT "dummy":MEMORY HIMEM-1:CLOSEOU [2303]
T
820 RETURN [555]
830 WHILE INKEY$<>"":WEND [1786]
840 RETURN [555]
850 ' [117]
860 g$="" : a$="" : EVERY 10 GOSUB 910:POKE po [2987]
k,0
870 a$="" : a$=INKEY$:IF a$="" THEN 870 [1989]
880 IF a$<>CHR$(127) THEN 890 ELSE IF lens [9063]
>0 THEN PRINT CHR$(8);CHR$(16);:lens=lens-
1:g$=LEFT$(g$,lens):GOTO 870 ELSE GOTO 870
890 IF a$<>CHR$(13) THEN GOTO 900 ELSE r=R [6066]
EMAIN(0):POKE pok,255:CALL &BB84:RETURN
900 IF NOT lens>MAXs THEN g$=g$+a$:PRINT C [4932]
HR$(1)+a$;:lens=lens+1:GOTO 870:ELSE GOTO
870
910 IF cur=0 THEN CALL &BB84:cur=1:RETURN [2448]
ELSE CALL &BB81:cur=0:RETURN
920 PRINT CHR$(9)CHR$(132)STRING$(36,CHR$( [10294]
9) )CHR$(136)CHR$(9)CHR$(130)CHR$(135)STRIN
G$(36,CHR$(131))CHR$(139)CHR$(129)CHR$(9)C
HR$(133)STRING$(36,CHR$(9))CHR$(138)CHR$(9
)CHR$(136);
930 PRINT CHR$(141)STRING$(36,CHR$(140))CH [8498]
R$(142)CHR$(132)CHR$(9)CHR$(129)STRING$(36
,CHR$(9))CHR$(130)CHR$(9);CHR$(9)CHR$(13);
940 PRINT CHR$(9);CHR$(9);CHR$(11);CHR$(11 [3949]
);CHR$(11);
950 RETURN [555]
960 RESUME NEXT [703]

```

```

1 '***** [1383]
2 '* POINT.DAT - DATA-Lader von 'CPC' * [2010]
3 '***** [1383]
4 ' [117]
5 DATA 8000,E5,3E,01,CD,59,BC,11,22,71C8 [1467]
6 DATA 8008,00,21,22,00,C3,46,80,CD,0ACD [1458]
7 DATA 8010,24,BB,FE,00,28,F9,E1,D1,20B7 [440]
8 DATA 8018,F5,CD,CO,BB,3E,9F,CD,FC,589A [2135]
9 DATA 8020,BB,F1,E1,D1,CB,67,28,02,7766 [1913]
10 DATA 8028,18,2A,CB,47,28,04,23,23,1AA5 [1603]
11 DATA 8030,18,06,CB,4F,28,02,2B,2B,1125 [1209]
12 DATA 8038,CB,57,28,04,1B,1B,18,06,7582 [1522]
13 DATA 8040,CB,5F,28,02,13,13,D5,E5,76FB [1255]
14 DATA 8048,D5,E5,CD,CO,BB,3E,9F,CD,42B3 [1655]
15 DATA 8050,FC,BB,18,BB,22,67,80,ED,5911 [2373]

```

Listing TESTME

```

16 DATA 8058,53,65,80,CD,CO,BB,3E,00,2880 [1460]
17 DATA 8060,CD,59,BC,E1,C9,00,00,00,6F18 [1850]
25 DATA *ENDE* [502]
26 adr=&8000:zeile=5:MEMORY &7FFF [2060]
27 READ d$:IF d$="*ENDE*"THEN 38 [1062]
28 pr=0 [117]
29 FOR i=1 TO 8 [462]
30 READ a$:a=VAL("&"+a$) [883]
31 POKE adr,a:adr=adr+1 [717]
32 pr=pr*2:IF pr>65535 THEN pr=pr-65535 [2485]
33 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553 [1890]
6
34 NEXT i [375]
35 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN [2610]
pr2=pr2+65536
36 IF pr<>pr2 THEN PRINT"Pruefsummenfehler [3615]
in Zeile";zeile:STOP
37 zeile=zeile+1:GOTO 27 [855]
38 SAVE"!TESTME1.BIN",B,&8000,&67 [1535]
39 PRINT d$:END [346]

```

```

1 '***** [1476]
2 '* SYMBOL.DAT - DATA-Lader von 'CPC' * [1125]
3 '***** [1476]
4 ' [117]
5 DATA 9000,00,0C,1E,36,66,7E,66,67,01C3 [2298]
6 DATA 9008,00,3E,63,63,7E,63,63,3E,0754 [1598]
7 DATA 9010,00,3E,63,60,60,60,63,3E,0798 [2070]
8 DATA 9018,00,3C,66,63,63,63,66,3C,0794 [1852]
9 DATA 9020,00,3E,63,60,78,60,63,3E,0758 [1302]
10 DATA 9028,00,3E,63,60,78,60,60,20,0740 [1633]
11 DATA 9030,00,3E,63,60,60,66,63,3E,0780 [1714]
12 DATA 9038,00,22,63,63,7F,63,63,22,0040 [1148]
13 DATA 9040,00,3C,18,18,18,18,18,3C,0D2C [1580]
14 DATA 9048,00,02,03,03,03,03,63,3E,003C [1286]
15 DATA 9050,00,23,66,6C,78,7C,66,23,001F [2060]
16 DATA 9058,00,20,60,60,60,61,63,3F,007D [2343]
17 DATA 9060,00,22,77,6B,6B,63,63,22,02E0 [1331]
18 DATA 9068,00,32,73,6B,6B,6B,67,26,064C [1772]
19 DATA 9070,00,3E,63,63,63,63,63,07BC [1284]
20 DATA 9078,00,3E,63,63,7E,60,60,20,0740 [1616]
21 DATA 9080,00,3E,63,63,63,6B,65,3A,0794 [2448]
22 DATA 9088,00,3E,63,63,7E,6C,66,23,077F [1176]
23 DATA 9090,00,3E,63,60,3E,03,63,3E,04E4 [1812]
24 DATA 9098,00,7E,5A,18,18,18,18,3E,15EC [1782]
25 DATA 90A0,00,22,63,63,63,63,63,3E,00BC [2081]
26 DATA 90A8,00,22,63,63,63,63,36,1C,0034 [1269]
27 DATA 90B0,00,22,63,63,63,6B,6B,36,0084 [2322]
28 DATA 90B8,00,63,36,1C,1C,1C,36,63,1F5F [1583]
29 DATA 90C0,00,24,66,66,3C,18,18,3C,022C [2094]
30 DATA 90C8,00,7F,43,04,1E,08,31,7F,172D [1501]
31 DATA *ENDE* [502]
32 adr=&9000:zeile=5:MEMORY adr-1 [2952]
33 READ d$:IF d$="*ENDE*"THEN 44 [1048]
34 pr=0 [117]
35 FOR i=1 TO 8 [462]
36 READ a$:a=VAL("&"+a$) [883]
37 POKE adr,a:adr=adr+1 [717]
38 pr=pr*2:IF pr>65535 THEN pr=pr-65535 [2485]
39 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553 [1890]
6
40 NEXT i [375]
41 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN [2610]
pr2=pr2+65536
42 IF pr<>pr2 THEN PRINT"Pruefsummenfehler [3615]
in Zeile";zeile:STOP
43 zeile=zeile+1:GOTO 33 [869]
44 SAVE"!TESTME2.BIN",B,&9000,&D0 [1536]
45 PRINT d$:END [346]

```

Listing TESTME

Das haben Ihre Zeitschriften nicht verdient....



Ordnung und Übersicht schaffen die DMV-Sammelmappen



Coupon:

Ausschneiden oder kopieren und senden an

DMV,
Kennwort: Sammelmappe,
Fuldaer Str. 6,
3440 Eschwege

Ich bestelle hiermit

_____ 2 Stck. 15,80 DM
zuzüglich 3,- DM Porto und Verpackung

für die Ausgabe von

DOS PC PASCAL

Name _____

Str., Nr _____

PLZ, Ort _____



Datum, _____

Unterschrift (bei Minderjährigen des gesetzl. Vertreters) _____

Fakturist

Mit diesem Programmpaket ist es möglich, die komplette Lagerverwaltung mit Bestellwesen, Lieferwesen, Inventur und Rechnungslegung zeitsparend mit dem Computer durchzuführen. Insbesondere die Rechnungserstellung ist weitgehend automatisiert, so daß selbst 30 oder 40 Rechnungen mit einem persönlichen Zeitaufwand von nur einigen Minuten (der Rest wird vom Rechner erledigt) geschrieben werden. Es muß nur für genügend Papier im Drucker gesorgt werden.

Kurzbeschreibung

Mit diesem Programmpaket wird das Lager mit allen Ein- und Ausgängen mit dem Rechner verwaltet. Die Lieferung sowie die Bestellung von Waren kann über dieses Programm abgewickelt werden. Der Lagerbestand wird dabei automatisch aktualisiert. Eine zeitsparende Inventur ist ebenso möglich wie die automatische Erstellung von Rechnungen für die Lieferscheine. Da die Diskettenkapazität nur 169 KByte beträgt, kann bei der Installation das Laufwerk B als Datenlaufwerk, falls vorhanden, eingestellt werden. Ebenso ist es möglich, mit der Vortexfloppy FI-X oder FIS/FID zu arbeiten. Speziell für Besitzer von Farbmonitoren wird die optimale Farbeinstellung für Anwendersoftware mit dem Programmteil SYSTEM ermöglicht.

Die genaue Bedienung des Programms ist der nachfolgenden Bedienungsanleitung zu entnehmen. Doch zuvor einige Hinweise zum Abtippen der Programme.

Hinweise zum Abtippen und erstmaligen Benutzen des Programms

Man kann sich sehr viel Arbeit ersparen, wenn man sich diese Programme im DATA-BOX-Service auf Disc bestellt. Sollen die Programme jedoch einzeln abgetippt werden, so sollten die Hinweise beachtet werden.

Die meisten Unterprogramme werden in fast allen Einzelprogrammen benutzt. Diese sollten zuerst abgetippt und unter den verschiedenen Programmnamen auf Disc gespeichert werden. Dadurch müssen diese Teile nicht mehrfach geschrieben werden. Danach kann der Rest eingetippt werden.

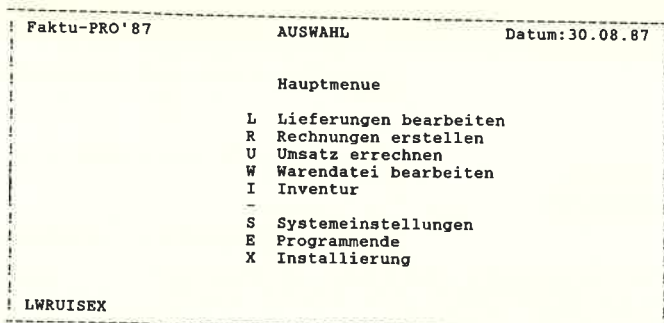


Bild 1

Der Menuegenerator z.B. (ab Zeile 15000) wird in allen Programmen bis START.BAS und RECH.BAS genutzt.

Alle Programme sind eigenständig und können daher direkt aufgerufen werden oder in eigenen Programmen Verwendung finden. Sollten diese jedoch kommerziell vertrieben werden, dann müssen die Urheberrechte des Verlages berücksichtigt werden. Für die eigene private Anwendung können diese oder Teile davon genutzt werden. Sollten dabei technische Probleme auftreten, so stehe ich über den Verlag für Fragen jederzeit zur Verfügung.

Die REMARK-Zeilen müssen mit abgetippt werden, da diese die Unterprogrammeinsprünge für die GOSUB-Befehle darstellen. Es reicht jedoch wenn die Zeile vorhanden ist. Die Unterprogrammnamen müssen nicht abgetippt werden, dienen jedoch der Übersichtlichkeit.

Da in einigen Routinen der Befehl EOF (End of File) benutzt wird, muß bei Verwendung des Betriebssystems VDOS 2.xx (Discbetriebssystem der Firma Vortex) im Startprogramm unter den ersten Zeilen der Befehl DERROR 2 (siehe auch Vortex-Handbuch) eingefügt werden, da sonst der o.g. Befehl zum Programmabbruch führen kann.

In den Programmen LIEF.BAS und RECH.BAS müssen an den entsprechenden Stellen statt des Firmenkopfes=Bäckerei Konditorei ... = der eigene eingesetzt werden. Die Aufteilung des Druckkopfes sollte jedoch weitestgehend beibehalten werden. Insbesondere die Anzahl der Druckzeilen darf nicht verändert werden.

Benutzen Sie einen anderen als einen EPSON kompatiblen Drucker, so müssen die Drucksteuerzeichen im Programm START.BAS entsprechend geändert werden.

Da das Programm schon beim erstmaligen Starten einige Dateien benötigt, muß es vor dem erstmaligen Benutzen installiert werden. Diese Installation wird automatisch von dem Programm INSTALL.BAS durchgeführt. Der erste Start des Programmpaketes muß daher nicht mit RUN "START sondern mit RUN "INSTALL stattfinden.

Das Programmpaket besteht aus den nachfolgenden Einzelprogrammen

START .BAS	3 KB	(Grundeinstellungen und Datum)
AUSWAHL.BAS	4 KB	(Hauptmenue zur Programmauswahl)
ART.BAS	9 KB	(Dateiverwaltung für WAREN.DAT)
LIEF.BAS	13 KB	(Liefer- und Bestellscheinerstellung)
RECH.BAS	5 KB	(Rechnungserstellung)
INV .BAS	6 KB	(Inventur)
INSTALL.BAS	2 KB	(Installierung des Programmpaketes)
SYSTEM.BAS	6 KB	(Systemeinst. und Disc-Routinen)

und wird mit der Befehlssequenz RUN "START gestartet. Das Programm meldet sich mit dem Titel FAKTU-PROG und der Versionsnummer. Darunter wird das Datum der letzten Bearbeitung angezeigt. Mit den CURSOR-Tasten wird das aktuelle Datum eingestellt.

Pfeil rechts → plus ein Tag
 Pfeil links ← minus ein Tag
 Pfeil rauf ↑ plus ein Monat
 Pfeil runter ↓ minus ein Monat

Das Jahr wird automatisch beim Übergang von Dezember auf Januar um eins erhöht bzw. beim Übergang von Januar auf Dezember um eins verringert. Ist das Datum eingestellt, wird es mittels der ENTER-Taste übernommen und auf Disc in der Datei DATUM.DAT gespeichert. Danach wird man aufgefordert, den

ucker einzuschalten und auf den Anfang des ersten Blattes zustellen. Sollte kein Drucker angeschlossen sein, so kann s Programm nicht weiter abgearbeitet werden. Der Drucker rd hierbei auf 79 Zeilen pro 12 Zollblatt mit einem Zeilenab- ind von 11/72 Zoll eingestellt. Danach wird das Programm USWAHL.BAS geladen und gestartet.

auf dem Monitor erscheint nun das in Bild 1 dargestellte Haupt- enue.

n der ersten Zeile wird in allen Programmteilen gleich jeweils er Titel, der Name des Programmteils, in dem man sich be- findet, und das Datum angezeigt. Die Zeilen 2 – 24 dienen der Verarbeitung und in der letzten Zeile werden die jeweils zuläs- sigen Tasten angezeigt und die Eingaben abgefordert. Von die- sem Hauptmenue ausgehend wird nun die Funktionsweise und Bedienung der einzelnen Programmteile beschrieben.

1.00 Waredatei bearbeiten

Zuerst soll die Waredatei bearbeitet werden. Mit der Taste W wird das Programm ART.BAS geladen und es erscheint das Me- nue »Artikeldatei bearbeiten« (Bild 2).

Bei der ersten Benutzung ist die Waredatei noch nicht vorhan- den und muß über die Taste A neu angelegt werden.

1.10 Datei neu anlegen

Auf dem Monitor erscheint die Eingabetabelle. In der Status- zeile werden die einzelnen Daten (Artikelnummer, Bezeich- nung, Einheit, Verkaufspreis, Einkaufspreis und Lagerbestand) abgefragt. Nach der Eingabe müssen diese jeweils mit der ENT- ER-Taste bestätigt werden. Sind für einen Artikel alle Daten ein- gegeben, so wird dieser in der Tabelle angezeigt. Bei Falschein- gaben kann der jeweils letzte Buchstabe mit der DEL-Taste ge- löscht werden. Die einzelnen Datenfeldlängen sind nachfolgend aufgezeigt:

Artikelnummer → 0000 bis 9999
 Artikelbezeichnung → maximal 24
 Zeichen Einheit → maximal 4
 Zeichen Verkaufspreis → 0.01 DM bis 9999.99 DM
 Einkaufspreis → 0.01 DM bis 9999.99 DM
 Menge → 0 bis 9999

Werden größere Daten eingegeben, so werden diese entweder als Fehleingabe angezeigt (Zahlenfelder) oder auf die zuläs- sige Länge gekürzt (Stringfelder). Nachdem alle Daten erfaßt sind, kann die Eingaberoutine durch Eingabe des Schrägstrichs »/« (unter dem Fragezeichen) für die Artikelnummer und der ENTER-Taste beendet werden. Das Menue gemäß Bild 2 wird wieder angezeigt.

1.20 Datei verändern

Soll die Waredatei verändert werden, so gelangt man mit der Taste V zum Menue »Datei verändern« (Bild 3).

1.21 Artikel löschen ändern

Nach Drücken der Taste L erscheint wieder die Eingabetabel- le mit den Artikeln. Da nicht alle Daten (es können 200 Arti- kel verwaltet werden) auf eine Bildschirmseite unterzubringen



Bild 2

sind, kann man mit der Taste »Pfeil rechts« vor- und mit der Taste »Pfeil links« zurückblättern, um die Datei durchzusehen. Mit der Taste L kann ein Artikel unter Eingabe der Artikelnummer gelöscht werden. Mit der Taste A können die Daten nachein- ander für eine Artikelnummer verändert werden. Diese Ände- rungen werden sofort in der Tabelle neu eingetragen. Zum Me- nue gelangt man mit der Taste M zurück.

1.22 Artikel anhängen

Um weitere Artikel in die Datei aufzunehmen, gelangt man mit der Taste A in die Eingaberoutine. Hier können nun, wie un- ter Punkt 1.10 (Datei neu anlegen) beschrieben, weitere Arti- kel eingegeben werden. Ins Menue (Bild 3) gelangt man wie- der mit der Eingabe des Schrägstrichs für die Artikelnummer.

1.23 Preise ändern

Wenn nur die Preise einzelner Artikel verändert werden sollen, so kann dies über die Taste P erreicht werden. In der Statuszeile wird nach der Artikelnummer gefragt. Danach werden die al- ten Preise und die Bezeichnung am Bildschirm angezeigt und die Eingabe der neuen Preise erwartet. Nach der Frage »Wei- ter Preise ändern?« gelangt man mit N ins Menue (Bild 3) zu- rück oder kann mit J weitere Preise ändern.

1.24 Menue

Mit der Taste M gelangt man wieder ins Menue »Artikeldatei bearbeiten« (Bild 2) zurück.

1.30 Datei sortieren

Mit der Auswahl dieses Punktes mittels der Taste S hat man die Möglichkeit, die Waredatei über ein weiteres Menue nach der Artikelnummer oder nach Alphabet zu sortieren. Nach dem Sortiervorgang oder mit der Taste M gelangt man wieder ins Menue gemäß Bild 2.

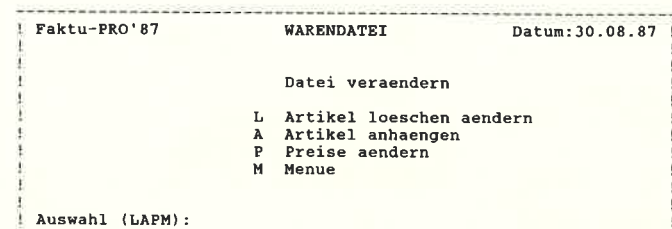


Bild 3

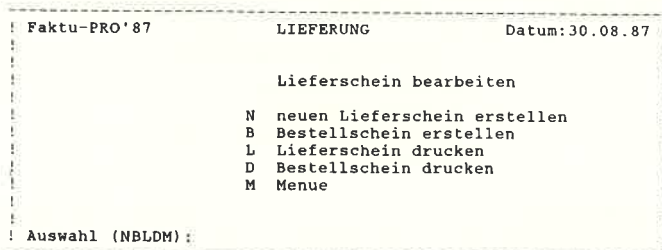


Bild 4

1.40 Datei ausdrucken

Mit der Taste D wird sofort mit der Ausgabe des Inhaltes der Waredatei über den Drucker begonnen. Dieser Ausdruck ist nur für Kontrollzwecke gedacht und daher nicht »schön« gestaltet worden, sondern rein zweckmäßig gehalten. Nach dem Ausdruck wird wieder das Menue (Bild 2) angezeigt.

1.50 Hauptmenue

Mit der Taste H wird die Bearbeitung der Waredatei abgeschlossen und es erscheint die Frage, ob die im Rechner befindlichen Daten gespeichert werden sollen. Hat man im Datenbestand keine Änderungen vorgenommen oder sollen diese nicht gespeichert werden, so kann man die Frage mit N beantworten, damit der bisherige Datenbestand auf der Disc erhalten bleibt. Sollen die durchgeführten Änderungen jedoch gespeichert werden, so muß mit J geantwortet werden. Die Datei WAREN.DAT wird mit dem neuen Inhalt abgespeichert. Danach gelangt man wieder ins Programm AUSWAHL.BAS mit dem Hauptmenue (Bild 1).

2.00 Lieferung bearbeiten

Um Liefer- und Bestellscheine bearbeiten zu können, wird das Programm LIEF.BAS mittels der Taste L vom Hauptmenue (Bild 1) aus aufgerufen. Die Datei WAREN.DAT wird eingelesen und es erscheint das Menue »Lieferschein bearbeiten« (Bild 4).

2.10 neuen Lieferschein erstellen

Mit der Taste N wird die Eingaberoutine zur Eingabe des Lieferscheins aufgerufen. Zuerst erscheint die Abfrage nach dem Empfänger oder Kunden des Lieferscheins. Die später zu erstellende Rechnung wird ebenfalls diesen Empfänger enthalten. Es wird nacheinander die Eingabe des Vor- und Zunamens, der Straße mit der Hausnummer, sowie der Postleitzahl und des Ortes erwartet. Auch hier kann mittels der DEL-Taste die letzte Eingabe gelöscht werden. Wird die Abfrage »Daten o.k. (JN)?« mit N beantwortet, so wird der Empfänger erneut abgefragt. Mit J erscheint nun die Eingabetabelle. In dieser wird in der Statuszeile jeweils nach der Artikelnummer und der zu liefernden Menge gefragt. Diese Daten werden danach in der Tabelle dargestellt. Da nur maximal 45 Artikel mit einem Lieferschein er-

faßt werden können, ist im rechten Bildschirmteil jeweils Menge der bisher eingegebenen Artikel dargestellt.

Bei der Eingabe der Liefermenge wird gleichzeitig geprüft, diese Menge noch auf Lager verfügbar ist. Falls dies nicht der Fall ist, wird die noch vorhandene Menge in der Statuszeile angezeigt und diese Artikeleingabe ignoriert. Sie müßte erneut in der verfügbaren Menge geschehen. Die Eingabe der Menge ist zwischen 0 und 9999 zulässig.

Bei der Abfrage der Artikelnummer müssen die führenden Nullen nicht mit eingegeben werden (für 0005 reicht die Eingabe 5). Sind alle Daten verarbeitet, so kann die Eingaberoutine mit der Eingabe des Schrägstrichs »/« für die Artikelnummer beendet werden. Dies geschieht automatisch, wenn 45 Artikel eingegeben wurden.

Die Abfrage »Daten o. k. (JN)?« ermöglicht mittels der Taste N die Korrektur der Eingaben. Dazu wird der Inhalt des Lieferscheins auf vier Bildschirmseiten angezeigt. Mit der ENTER-Taste kann nun von der ersten bis zur vierten Seite geblättert werden. Ein Zurückblättern ist nicht möglich. Die Daten können durch Eingabe der Artikelnummer und der richtigen Menge korrigiert werden. Mit der Menge 0 werden Artikel aus dem Lieferschein gelöscht. Das Hinzufügen weiterer Artikel ist nicht möglich. Nach der vierten Seite wird der Korrekturteil automatisch beendet.

Jetzt wird der Lagerbestand aktualisiert, der Lieferschein mit einer Nummer versehen (automatisch bis 9999) und gespeichert. Danach wird wieder das Menue (Bild 4) angezeigt.

2.20 Bestellschein erstellen

Die Erstellung eines Bestellscheins geschieht analog zum Lieferschein. Auch hier wird die Waredatei sofort aktualisiert, jedoch wird die bestellte Menge zum Lagerbestand addiert.

2.30 Lieferschein drucken

Durch Drücken der Taste L wird die Druckroutine aufgerufen. Hierbei wird geprüft, ob vorher ein Lieferschein erstellt wurde und dieser noch im Speicher ist. Ist dies der Fall, so kann der Lieferschein ausgedruckt werden. Wurde vorher jedoch kein Lieferschein erstellt oder wurde dieser Programmteil LIEFERUNG seitdem verlassen, so werden die auf Disc gespeicherten Lieferscheine angezeigt und die Eingabe des zu druckenden Lieferscheindateinamens erwartet. Dieser muß ohne Extension (.LIF) eingegeben werden.

Danach erscheint das Menue »Lieferschein drucken«. Hier kann die Druckroutine mittels der Taste M noch verlassen werden oder die Art des Ausdrucks – mit Warenliste oder ohne – ausgewählt werden. Diese Warenliste soll für den zu beliefernden Kunden einen Gesamtüberblick der lieferbaren Waren vermitteln und zugleich als Bestellliste dienen.

Nach dieser Auswahl beginnt die Ausgabe über die Drucker. Danach wird wieder ins Menue (Bild 4) zurückgesprungen.

2.40 Bestellschein drucken

Durch Drücken der Taste D gelangt man in die Druckroutine für den Bestellschein (analog zu Beschreibung zu Punkt 2.30). Dieser dient dazu, bei dem eigenen Lieferanten Waren zu bestellen, um die Lagerbestände wieder aufzufüllen. Nach der Prüfung, ob vorher ein Bestellschein erstellt wurde (mit ggf. Anzeige der gespeicherten Dateien), wird sofort mit dem Ausdruck begonnen. Das Menue (Bild 4) wird danach wieder aufgerufen.

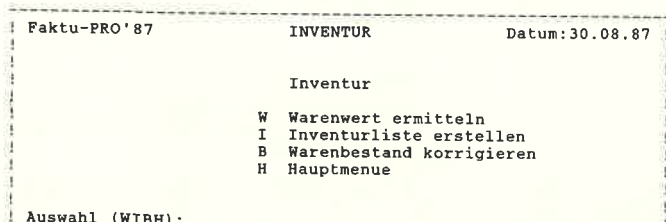


Bild 5

0 Menue

t der Taste M wird das »Hauptmenue« angezeigt. Will man mehrere Lieferscheine oder Bestellscheine nacheinander drucken, so muß immer bis in dieses Hauptmenue zurückgegangen werden. Danach wird mit der Taste L wieder das Menue »Lieferschein bearbeiten« (Bild 4) angezeigt und es können weitere Daten von Disc gedruckt werden. Dieser Weg ist nur notwendig, wenn mehrere Dateien gedruckt werden sollen. Für die Erstellung mehrerer Lieferscheine z.B. ist dies nicht notwendig.

Mit der Taste H wird das LIEFERPROGRAMM beendet und das Auswahl-Menue (Bild 1) geladen.

3.00 Rechnung erstellen

Mit der Taste R (Rechnungserstellung) wird das Programm RECH.BAS geladen. Dieses meldet sich mit der Frage, ob die Rechnungen wirklich erstellt werden sollen. Mit N für Nein gelangt man wieder ins Auswahlmenue zurück. Mit J wird die Rechnungserstellung gestartet. Es werden nun alle gespeicherten Lieferscheine nacheinander geladen und jeweils eine Rechnung erstellt. Am Bildschirm wird dabei angezeigt, wieviele Lieferscheine noch bearbeitet werden müssen und welcher gerade bearbeitet wird. So können auf einmal maximal 45 Lieferscheine (sofern die Disc-Kapazität ausreicht) bearbeitet werden.

Sind alle Rechnungen erstellt, so erscheint die Frage, ob diese noch einmal gedruckt werden sollen. Ein Grund hierfür kann ein schlechtes Farbband oder Papiersalat im Drucker sein. Mit J werden die Rechnungen erneut gedruckt. Mit N erwartet der Rechner die Genehmigung, die Lieferscheine auf der Disc zu löschen, damit für neue Lieferscheine Platz geschaffen wird. Wird die Frage mit N beantwortet, so werden die Lieferscheine bei der nächsten Rechnungserstellung erneut bearbeitet und somit die gleichen Artikel zum zweiten Mal in Rechnung gestellt. Mit J werden die Lieferscheine auf der Disc gelöscht, die Rechnungssumme gespeichert (dies geschieht auch bei N) und wieder das Auswahlmenue geladen.

4.00 Umsatz errechnen

Mit der Taste U wird die Summe aller bearbeiteten Rechnungen und Bestellscheine sowie deren Differenz errechnet und am Bildschirm angezeigt. Mit Tastendruck gelangt man danach wieder ins Hauptmenue zurück.

5.00 Inventur

Mit der Taste I wird das Inventurprogramm INV.BAS geladen und es erscheint das Menue »Inventur«, nachdem die Waren-datei geladen wurde, am Bildschirm (Bild 5).

5.10 Warenwert ermitteln

Mit der Taste W wird eine Routine gestartet, die den Wert aller in der Datei WAREN.DAT und somit der auf Lager befindlichen Artikel errechnet. Der Einkaufs- und Verkaufswert werden am Monitor angezeigt.

5.20 Inventurliste erstellen

Mit der Taste I wird eine Inventurliste erstellt und ausgedruckt. Diese enthält alle Artikel mit der Artikelnummer, Artikelbezeichnung und dem Lagerbestand incl. aller noch nicht gelieferten Bestellscheine und noch nicht ausgelieferten Lieferschei-

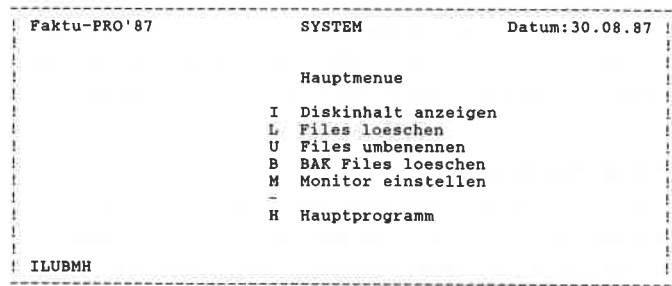


Bild 6

ne. Mit dieser Liste kann der tatsächliche Warenbestand kontrolliert werden. Eine notwendige Korrektur kann mit dem Programmteil »Warenbestand korrigieren« durchgeführt werden.

5.30 Warenbestand korrigieren

Mit der Taste B hat man die Möglichkeit, Differenzen zwischen der Lagerbuchführung und dem tatsächlichen Lagerbestand auszugleichen. Ebenso kann über diese Routine der Warenbestand korrigiert werden, wenn die mit Bestellschein bestellten Mengen nicht oder nur teilweise geliefert wurden, und somit der Lagerbestand korrigiert werden muß.

Es wird sofort nach der Artikelnummer gefragt. Danach wird der Artikel mit dem gespeicherten Bestand angezeigt und der neue Lagerbestand in der Eingabe erwartet. Sollen weitere Daten geändert werden, so muß die nachfolgende Frage »Weitere Daten ändern« mit J beantwortet werden. Mit N wird diese Routine verlassen und wieder das Menue (Bild 5) angezeigt.

5.40 Hauptmenue

Mit der Taste H wird das Inventurprogramm verlassen und wieder das Auswahlmenue (Bild 1) geladen. Zuvor wird jedoch gefragt, ob die im Rechner befindlichen Daten auf Disc gespeichert werden sollen. Wurden mittels der Routine »Warenbestand korrigieren« die Daten verändert, so muß mit J geantwortet werden um diese Änderungen auf der Disc zu speichern und somit die Datei WAREN.DAT zu aktualisieren. Mit der Taste N werden die Daten im Rechner gelöscht, aber auf der Disc nicht verändert.

6.00 Systemeinstellungen

Mit der Taste S wird das Programm SYSTEM.BAS geladen. Damit können Systemeinstellungen durchgeführt oder Disketten bearbeitet werden. Es erscheint ein Menue (Bild 6).

6.10 Diskinhalt anzeigen

Mit der Taste I wird nach der Auswahl des Laufwerks A oder B (C als RAMdisk ist z. Zt. noch nicht möglich) der Diskinhalt am Bildschirm angezeigt. Auf Tastendruck gelangt man zurück zum Hauptmenue.

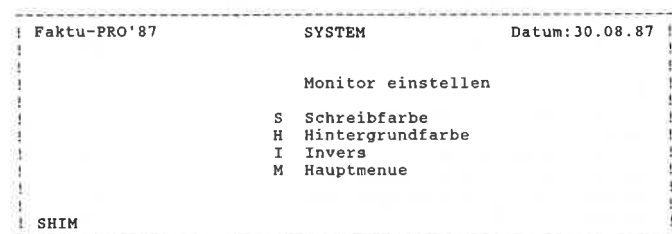


Bild 7

Wird hier ein nicht angeschlossenes Laufwerk angewählt, so wird das Programm mit einer Fehlermeldung abgebrochen und kann mit der Befehlssequenz RUN wieder gestartet werden.

6.20 Files löschen

Mit der Taste L gelangt man nach der Laufwerkswahl in die Löschroutine. Der Diskinhalt wird angezeigt und nach Tastendruck der zu löschende Filename abgefragt. Dieser Name muß mit der Extension wie im Inhaltsverzeichnis dargestellt eingegeben werden. Hierbei ist das Wildcardzeichen * erlaubt. Um z.B. alle Bestellscheine auf der Diskette zu löschen, müßte als Filename B*.LIF eingegeben werden. Nach der Eingabe des Filenamens wird noch eine Sicherheitsabfrage durchgeführt. Mit N kann der Vorgang abgebrochen werden. Mit J wird gelöscht. Am Ende erscheint das Hauptmenue.

6.30 File umbenennen

Mit der Taste U wird nach der Laufwerkswahl wieder das Inhaltsverzeichnis angezeigt. Auf Tastendruck kann zuerst der alte und dann der neue Filename eingegeben werden. Hierbei wird keine Sicherheitsabfrage durchgeführt und die Umbenennung sofort vollzogen. Das Menue bietet dann weitere Bearbeitungsmöglichkeiten.

6.40 BAK-Files löschen

Mit der Taste B wird wieder nach der Laufwerkswahl das Inhaltsverzeichnis angezeigt. Auf Tastendruck werden nach einer Sicherheitsabfrage alle auf der Disc befindlichen Files mit der Extension .BAK gelöscht. Danach erscheint wieder das Hauptmenue.

6.50 Monitor einstellen

Mit der Taste M erscheint ein weiteres Menue (Bild 7). Von diesem können die eingestellten Farben des Monitors verändert werden. Die neuen Farbeinstellungen werden auf Disc gespeichert und zukünftig beim Programmstart immer wieder eingestellt.

6.51 Schreibfarbe

Die Schreibfarbe wird in der Statuszeile abgefragt. Es können die Farben von 0 (Schwarz) bis 26 (Weiß) eingestellt werden. Diese Farbnummern sind entsprechend dem INK-Befehl codiert. Welche Nummer welcher Farbe entspricht, kann im CPC-Handbuch nachgelesen werden.

Zusätzlich wurde eine Sicherheitsroutine installiert, die verhindert, daß die Schreib- und Hintergrundfarbe gleich eingestellt werden.

Nach Eingabe der Nummer wird die Schreibfarbe sofort auf den neuen Wert eingestellt.

6.52 Hintergrundfarbe

Die Einstellung der Hintergrundfarbe geschieht in gleicher Weise wie bei der Schreibfarbe. Der Rand (BORDER) wird immer gleich mit der Hintergrundfarbe eingestellt. Auch diese Einstellung wird sofort durchgeführt.

6.53 Invers

Mit der Taste I werden die Schreib- und Hintergrundfarbe miteinander vertauscht.

6.54 Hauptmenue

Mit der Taste M gelangt man wieder ins Hauptmenue des Systemprogramms (Bild 6) zurück.

6.60 Hauptprogramm

Mit der Taste H wird das Auswahlprogramm geladen. Zuvor werden noch die geänderten Monitoreinstellungen in der Datei FARBE.DAT gespeichert. Danach erscheint das Auswahlmenue (Bild 1).

7.00 Installation

Dieses Programm dient dazu, die benötigten Dateien außer der Waredatei erstmalig auf der Disc anzulegen. Dabei werden die Zähler für die Lieferschein- und Bestellscheinnummer auf Null gesetzt. Ebenso wird die Monitoreinstellung und das Datum auf einen definierten Anfangszustand gesetzt. Zum Schluß der Installation wird noch abgefragt, ob das Programmsystem mit einem oder zwei Diskettenlaufwerken betrieben werden soll. Bei der Auswahl von zwei Laufwerken wird das Laufwerk A als Programmlaufwerk mit den installierten Dateien und das Laufwerk B zur Speicherung der Liefer- und Bestellscheine benutzt. Damit können mehr Dateien verarbeitet werden, da mit einem Laufwerk nur ca. 80 Kb Platz verbleiben.

Wird die Installation gestartet, wenn sich bereits Daten auf der Disc befinden, so besteht die Gefahr, daß diese gelöscht werden.

(H. Wienhoven/cd)

für 464-664-6128

```
1 '***** [1391]
2 '***      INSTALL .BAS V4.0      *** [1512]
3 '***      WIESOFT 1987          *** [1208]
4 '***      Heinz Wienhoven      *** [1817]
8 '***** [1391]
10 CLS [91]
20 MODE 1 [506]
30 PRINT TAB(14);CHR$(24);" I N S T A L L [2927]
   ";CHR$(24)
40 PRINT:PRINT:PRINT [1082]
50 PRINT "Bitte die Diskette zur Auf-" [3275]
60 PRINT "nahme des Programms" [2370]
70 PRINT "F A K T U   einlegen" [1858]
```

Listing Fakturist

```
80 PRINT:PRINT"Mit diesem Programm werden [5451]
die notwen-"
90 PRINT"digen Dateien angelegt. Sollten d [5850]
iese "
100 PRINT"schon vorhanden sein, so wird de [4242]
ren In-"
110 PRINT"halt geloescht!" [2717]
120 PRINT:INPUT "Soll installiert werden ( [3857]
JN) ? ",ant$
130 an$=UPPER$(ant$):IF an$="N" THEN RUN " [2471]
AUSWAHL.BAS"
140 nu=0 [368]
150 PRINT"Die Datei NR.LID wird angelegt": [6714]
OPENOUT "NR.LID":PRINT #9,nu:CLOSEOUT
160 nu$="0000" [744]
170 PRINT"Die Datei SUMM.REC wird angelegt [6632]
":OPENOUT "SUMM.REC":FOR i=0 TO 3:PRINT #9
,nu$:NEXT i:CLOSEOUT
```

Listing Fakturist

```

80 OPENOUT "XXX.LIF":PRINT #9,nu:CLOSEOUT [3862]
.90 OPENOUT "XXXX.PLU":PRINT #9,nu:CLOSEOU [2487]
r
200 PRINT"Die Datei RECH.SUM wird angelegt [6707]
":OPENOUT"RECH.SUM":PRINT#9,nu:PRINT#9,nu:
CLOSEOUT
210 PRINT"Die Datei RNR.LID wird angelegt" [7238]
:OPENOUT "RNR.LID":PRINT #9,nu:CLOSEOUT
220 PRINT"Die Datei DATUM.DAT wird angeleg [7713]
t":OPENOUT "DATUM.DAT":dat$="00.00.00":PRI
NT#9,dat$:PRINT#9,dat$:CLOSEOUT
230 PRINT"Die Datei FARBE.DAT wird angeleg [7503]
t":OPENOUT "FARBE.DAT":f0=0:f1=26:PRINT#9,
f0:PRINT#9,f1:CLOSEOUT
240 PRINT:PRINT "Soll das Programm mit ein [3647]
em (nur LW A)"
250 PRINT "oder mit zwei (A und B) Laufwer [3277]
ken"
260 PRINT "betrieben werden? Das LW B dien [3857]
t "
270 PRINT "dabei als Datenlaufwerk!" [1332]
280 INPUT " E in oder Z wei Laufwerke: ";a [2840]
nt$
290 lw$=UPPER$(ant$) [823]
300 IF lw$="E" THEN lauf$="A":GOTO 330 [2131]
310 IF lw$="Z" THEN lauf$="B":GOTO 330 [1524]
320 GOTO 280 [413]
330 PRINT "Die Datei LW.DAT wird angelegt" [5251]
:OPENOUT "LW.DAT":PRINT#9,lauf$:CLOSEOUT
340 CLS [91]
350 PRINT " Die Dateien sind" [2161]
360 PRINT " installiert!" [2437]
370 RUN "start.bas" [650]

```

```

1 ***** [577]
2 ** START.BAS V5.0 ** [1403]
3 ** Heinz Wienhoven ** [1458]
7 ***** [577]
9 OPENOUT "DUMMY":MEMORY HIMEM-1:CLOSEOUT [3606]
10 in$=CHR$(24):feld$=in$+" "+in$ [857]
15 '!DERROR,2 [367]
16 'Diese Zeile 15 muss nur beim Vortexbet [3811]
riebssystem eingegeben werden
20 MODE 2 [513]
25 OPENIN "DATUM.DAT":INPUT#9,dat$:CLOSEIN [3266]
30 OPENIN "FARBE.DAT":INPUT#9,farbe0:INPUT [4280]
#9,farbe1:CLOSEIN
40 INK 1,farbe1:INK 0,farbe0:BORDER farbe0 [3579]
50 READ y,x [354]
60 WHILE x<>0 [1191]
70 LOCATE x,y:PRINT feld$ [854]
80 READ x [433]
90 WEND [390]
95 READ x:IF x<0 THEN GOTO 50 [795]
100 LOCATE 15,12:PRINT "Version 5.0" [1715]
110 LOCATE 32,12:PRINT "WiesOFT ";CHR$(164 [2525]
);" 1987"
340 LOCATE 10,20:PRINT "Bitte das Datum mi [6174]
t den CURSORTasten eingeben : ";
350 GOSUB 3000 [961]
390 dat1$=dat$ [724]
400 OPENOUT "DATUM.DAT":PRINT#9,dat$:PRINT [2111]
#9,dat1$:CLOSEOUT
430 GOSUB 4000:ORIGIN 0,0:MOVE 0,16:DRAW 6 [2723]
39,16
440 GOSUB 5000 [883]
450 PRINT#1,TAB(20);"Das Programm AUSWAHL [2689]
wird geladen";
460 RUN "AUSWAHL.BAS" [723]
1000 ' DATAS fuer Hauptttitel [1858]
1010 DATA 3,2,3,4,5,6,9,10,11,12,13,16,20, [6094]
23,24,25,26,27,30,34,42,43,44,45,49,50,51,
52,57,58,59,64,67,68,69,73,74,75,76,77,0,-
1
1020 DATA 4,2,9,13,16,19,25,30,34,42,46,49 [3104]
,53,56,60,64,66,70,73,77,0,-1
1030 DATA 5,2,9,13,16,18,25,30,34,42,46,49 [3539]

```

Listing Fakturist

```

,53,56,60,63,66,70,77,0,-1
1040 DATA 6,2,3,4,9,10,11,12,13,16,17,25,3 [5413]
0,34,37,38,39,42,43,44,45,49,50,51,52,56,6
0,66,67,68,69,70,76,0,-1
1050 DATA 7,2,9,13,16,18,25,30,34,42,49,51 [2373]
,56,60,66,70,75,0,-1
1060 DATA 8,2,9,13,16,19,25,30,34,42,49,52 [2822]
,56,60,66,70,74,0,-1
1080 DATA 9,2,9,13,16,20,25,31,32,33,42,49 [4423]
,53,57,58,59,67,68,69,74,0,-1
1090 DATA 14,2,3,6,7,10,11,14,15,18,19,22, [6185]
23,26,27,30,31,34,35,38,39,42,43,46,47,50,
51,54,55,58,59,62,63,66,67,70,71,74,75,0,1
1100 DATA 1,1 [273]
3000 ' Datum abfragen [1272]
3010 'das Datum wird mit den CURSORTasten [4012]
eingestellt
3020 ta=VAL(LEFT$(dat$,2)) [1096]
3030 mo=VAL(MID$(dat$,4,2)) [1434]
3040 ja=VAL(RIGHT$(dat$,2)) [1708]
3050 LOCATE 27,22:PRINT in$;" ";dat$;" ";i [1441]
n$
3060 y$=INKEY$:IF y$="" THEN 3060 [2682]
3070 IF y$=CHR$(13) THEN RETURN [3651]
3080 IF y$=CHR$(241) THEN mo=mo-1 [2476]
3090 IF y$=CHR$(240) THEN mo=mo+1 [2494]
3100 IF y$=CHR$(242) THEN ta=ta-1 [2689]
3110 IF y$=CHR$(243) THEN ta=ta+1 [2844]
3120 IF mo<1 THEN mo=12:ja=ja-1 [1679]
3130 IF mo>12 THEN mo=1:ja=ja+1 [2117]
3140 IF ta>31 THEN ta=31 [1317]
3150 IF ta<1 THEN ta=1 [1233]
3160 dat$=RIGHT$(STR$(ta+100),2)+"."+RIGHT [4322]
$(STR$(mo+100),2)+"."+RIGHT$(STR$(ja+100),
2)
3170 GOTO 3050 [447]
4000 ' Windows [288]
4010 WINDOW#0,1,80,1,23 [1190]
4020 WINDOW#1,1,80,25,25 [1010]
4090 RETURN [555]
5000 '*** Drucker und Papier einstellen EP [3914]
SON FX-80+
5020 CLS#1 [373]
5030 PRINT#1,"Drucker e i n s c h a l t e [4069]
n ! Weiter mit Taste":CALL &BB06
5040 PRINT#1,"Drucker auf Blattanfang stel [5202]
len! Weiter mit Taste";CALL &BB06
5045 PRINT#8,CHR$(27);CHR$(64);' Drucker i [3453]
nitiallisieren
5050 PRINT#8,CHR$(27);"c";CHR$(79);' 79 Ze [2750]
ilen je Blatt
5060 PRINT#8,CHR$(27);"C";CHR$(0);CHR$(12) [3156]
;' 12 Zoll je Blatt
5070 PRINT#8,CHR$(27);"A";CHR$(11);'11/72 [2836]
Zoll pro Zeile
5080 PRINT#8,CHR$(13); [1113]
5090 RETURN [555]
9000 'Verzoegerungsschleife [915]
9010 FOR i=1 TO 500:NEXT i:RETURN [1213]

```

```

2 ***** [1164]
3 *** AUSWAHL .BAS V2.0 *** [1017]
4 *** WiesOFT 1987 *** [1133]
5 *** Heinz Wienhoven *** [1757]
9 ***** [1164]
20 in$=CHR$(24):bi$=CHR$(7) [1073]
180 OPENIN "DATUM.DAT":INPUT#9,dat$:CLOSEI [3266]
N
200 MODE 2 [513]
210 PRINT "Faktu-PRO'87";TAB(24);"AUSWAHL" [3936]
;TAB(60);"Datum: ";dat$
400 '*** Windows [637]
410 WINDOW#0,1,80,2,23 [1198]

```

Listing Fakturist

Anwendungen

```

420 WINDOW#1,1,80,25,25 [1010]
500 ON ERROR GOTO 29000 [1503]
999 '----- [3264]

1000 '*** Hauptmenue [812]
1010 RESTORE 1015:GOSUB 15000:ende=0 [1424]
1015 DATA Hauptmenue,L,Lieferungen bearbei [6200]
ten,R,Rechnungen erstellen
1016 DATA U,Umsatz errechnen,W,Warendatei [3548]
bearbeiten
1017 DATA I,Inventur,-,S,Systemeinstellun [2232]
gen
1018 DATA E,Programmende,X,Installierung,x [2358]
,x,LWRUISEX
1020 ON wa GOSUB 2000,3000,4000,9100,6000, [3588]
8000,1100,7000
1030 IF ende=1 THEN GOTO 1500 [725]
1040 GOTO 1000 [339]
1100 ende=1:RETURN [263]
1499 '----- [3300]

1500 '*** Ende des Programms [1352]
1510 MODE 2 [513]
1520 LOCATE 1,25:PRINT "Das Programm wird [3592]
beendet!";
1530 NEW:END [477]
1999 '----- [3336]

2000 PRINT#1,TAB(20);"Programm LIEFERUNG w [3658]
ird geladen";:RUN "LIEF.BAS"
3000 PRINT#1,TAB(20);"Programm ARTIKEL wir [5153]
d geladen";:RUN "ART.BAS"
4000 PRINT#1,TAB(20);"Programm RECHNUNG wi [2910]
rd geladen";:RUN "RECH.BAS"
6000 PRINT#1,TAB(20);"Programm INVENTUR wi [4343]
rd geladen";:RUN "INV.BAS"
7000 PRINT#1,TAB(20);"Programm INSTALL wir [5107]
d geladen";:RUN "INSTALL.BAS"
8000 PRINT#1,TAB(20);"Programm SYSTEM wird [3486]
geladen";:RUN "SYSTEM.BAS"
8999 '----- [3336]

9000 '*** Rechnungssumme lesen [3596]
9020 OPENIN"RECH.SUM" [1285]
9030 INPUT#9,svk [903]
9040 INPUT#9,sne [1013]
9050 CLOSEIN [752]
9060 RETURN [555]
9099 '----- [3336]

9100 '*** Rechnungssumme anzeigen [3065]
9110 CLS:PRINT t$;"Die Rechnungssumme wird [3451]
errechnet"
9120 GOSUB 9000:sumvk=svk:sumne=sne [3536]
9130 rab=sumvk-sumne [1009]
9140 FOR i=1 TO 5:PRINT:NEXT [1325]
9150 PRINT t$;"Bilanzsumme:" [2456]
9160 PRINT:PRINT USING " Rechnungss [5770]
ummen :#####.## DM";sumvk
9170 PRINT USING " Bestellsommen [4652]
:#####.## DM";sumne
9190 PRINT" [1855]
-----"
9200 PRINT USING " Bilanzsumme [4006]
:#####.## DM";rab
9210 PRINT" [2093]
-----"
9220 PRINT#1,"Weiter mit Taste!":CALL &BBO [2548]
6:CLS#1:RETURN
9999 '----- [3336]

10000 '*** STRING-Eingabe [707]
10010 an$="" [187]
10020 y$=INKEY$:IF y$=""THEN 10020 [2649]
10030 IF ASC(y$)=13 THEN CLS#1:RETURN [1749]
10032 IF ASC(y$)=127 THEN an$=LEFT$(an$,LE [3458]
N(an$)-1):CLS#1:PRINT#1,an$;
10040 IF ASC(y$)<an OR ASC(y$)>en THEN PRI [4028]
NT bi$;:GOTO 10020
10050 PRINT#1,y$;:an$=an$+y$:y$="":GOTO 10 [2289]
020
10099 '----- [3264]

10100 '*** Tastenabfrage [506]
10110 PRINT#1,wa$;bi$; [358]
10120 y$=UPPER$(INKEY$):IF y$=""THEN 10120 [3467]
10130 wa=INSTR(wa$,y$):IF wa=0 THEN GOSUB [5159]
10200:GOTO 10110

```

Listing Fakturist

```

10140 CLS#1:RETURN [1242]
10199 '----- [3264]

10200 '*** FEHLEINGABE [1735]
10210 PRINT bi$;:PRINT#1,TAB(30);in$;" FEH [2451]
LEINGABE ";in$;
10220 FOR kk=1 TO 1000:NEXT kk:PRINT bi$;; [3257]
PRINT#1
10230 RETURN [555]
14999 '----- [3264]

15000 '*** Menugenerator [1386]
15010 CLS:READ a1$ [730]
15020 LOCATE 24,5:PRINT a1$:PRINT [1891]
15030 READ a1$,a2$ [509]
15040 WHILE a1$<>"x" [1728]
15050 PRINT TAB(20);in$;" ";a1$;" ";in$;" [2406]
";a2$
15060 READ a1$,a2$ [509]
15070 WEND [390]
15080 READ wa$:PRINT:PRINT:GOSUB 10100 [2198]
15090 RETURN [555]
19999 '----- [3318]

28999 '----- [3264]

29000 '*** Fehlerbehandlung [1559]
29010 CLS#1:PRINT#1,"Fehler Nr. ";ERR;" in [2140]
Zeile ";ERL
29020 FOR ii=1 TO 2000:NEXT ii [1314]
29030 RESUME NEXT [703]

2 '***** [1391]
3 '*** ARTIKEL .BAS V4.0 [1383]
4 '*** WIESOFT 1987 [1208]
5 '*** Heinz Wienhoven [1817]
9 '***** [1391]
20 in$=CHR$(24):bi$=CHR$(7) [1073]
30 i=0:wa=0:wa$="":y$="":an=0:en=0:dat$="" [2444]
:na$="":an$="":er=0:k=0:j=0
32 t$=STRING$(9,32) [1587]
40 DIM ar$(200,5) [597]
170 '*** Datum einlesen [1681]
180 OPENIN "DATUM.DAT":INPUT#9,dat$:INPUT# [4251]
9,dat1$:CLOSEIN
200 MODE 2 [513]
210 PRINT "Faktu-PRO'87";TAB(24);"WARENDAT [4950]
EI";TAB(60);"Datum: ";dat$
310 ON ERROR GOTO 29000 [1503]
400 '*** Windows [637]
410 WINDOW#0,1,80,2,23 [1198]
420 WINDOW#1,1,80,25,25 [1010]
430 WINDOW#2,1,9,6,21 [1403]
440 WINDOW#3,11,35,6,21 [954]
450 WINDOW#4,37,41,6,21 [1406]
460 WINDOW#5,43,50,6,21 [1236]
470 WINDOW#6,52,59,6,21 [1034]
480 WINDOW#7,61,67,6,21 [1552]
1000 GOSUB 2000:RUN"AUSWAHL.BAS" [1661]
1600 '*** Tabellenaufbau [1249]
1610 CLS:PRINT:PRINT" Datei: ";ar$(0,1) [2166]
1620 PRINT" Art.Nr. Artikel";TAB(37);"Ein [5391]
h Vkpreis Ekpreis Menge"
1630 MOVE 8,336:DRAW 540,336:MOVE 76,350:D [4432]
RAW 76,60:MOVE 284,350:DRAW 284,60
1640 MOVE 332,350:DRAW 332,60:MOVE 404,350 [4057]
:DRAW 404,60:MOVE 476,350:DRAW 476,60
1650 MOVE 540,350:DRAW 540,60 [2280]
1660 RETURN [555]
1700 '*** Windows loeschen [1342]
1710 CLS#2:CLS#3:CLS#4:CLS#5:CLS#6:CLS#7:R [3358]
ETURN
1800 '*** Tabelle fuellen [890]
1810 GOSUB 1700 [871]

```

Listing Fakturist

```

320 FOR ii=er TO er+15 [1071]
330 PRINT#2," ";ar$(ii,0):PRINT#3,ar$( [2229]
i,1):PRINT#4,ar$(ii,2)
840 PRINT#5,USING"#####.###";VAL(ar$(ii,3)) [1096]
850 PRINT#6,USING"#####.###";VAL(ar$(ii,4)) [1903]
860 PRINT#7,USING"#####";VAL(ar$(ii,5)) [1625]
.870 NEXT ii:RETURN [1136]
3000 '*** Artikeldatei [881]
2010 RESTORE 2015:GOSUB 15000:GOSUB 9900 [2270]
2015 DATA Artikeldatei bearbeiten,A,Datei [6971]
neu anlegen,V,Datei veraendern,S,Datei sor
tieren,D,Datei ausdrucken,H,Hauptmenue,x,x
,AVSDH
2020 IF wa=5 THEN GOSUB 4500:RETURN [1560]
2030 ON wa GOSUB 3000,2200,2800,2500 [2345]
2040 GOTO 2010 [305]
2200 '*** Datei veraendern [1556]
2205 IF ar$(0,1)<>" THEN 2210 [1505]
2206 GOSUB 9400:GOSUB 9100 [2087]
2210 RESTORE 2215:GOSUB 15000:GOSUB 9900 [2115]
2215 DATA Datei veraendern,L,Artikel loesc
hen aendern einfuegen,A,Artikel anhaengen,
P,Preise aendern,M,Menue,x,x,LAPM
2220 IF wa=4 THEN RETURN [1275]
2230 ON wa GOSUB 3300,3700,6400 [1980]
2240 GOTO 2210 [361]
2500 '*** Datei ausdrucken [1412]
2510 IF ar$(0,1)<>" THEN GOTO 2530 [2200]
2520 GOSUB 9400:GOSUB 9100 [2087]
2530 PRINT#1,"Die Datei ";ar$(0,1);" wird [2726]
ausgedruckt!";i=1
2540 PRINT#8,t$;" Datei: ";ar$(0,1);t [4835]
$;t$;t$;"Datum: ";dat$:PRINT#8
2550 WHILE ar$(i,1)<>" [1091]
2560 PRINT#8,USING" ### \ \ \ \ [7260]
###.### #####;i;ar$(i,0);ar$(i,1);ar$(i,2)
;VAL(ar$(i,3));VAL(ar$(i,4));VAL(ar$(i,5))
2565 IF i MOD 70=0 THEN PRINT#8,CHR$(12); [846]
2570 i=i+1:WEND [758]
2575 PRINT#8,CHR$(12); [1101]
2580 CLS#1:RETURN [1242]
2800 '*** Datei sortieren [1254]
2805 IF ar$(0,1)=" THEN GOSUB 9400:GOSUB [1645]
9100
2810 RESTORE 2815:GOSUB 15000:GOSUB 9900 [2166]
2815 DATA Datei sortieren,N,nach Artikelnr [4634]
.,A,nach Alphabet,M,Menue,x,x,NAM
2820 IF wa=3 THEN RETURN [2034]
2822 IF wa=1 THEN g=0 ELSE g=1 [1867]
2826 PRINT:PRINT t$;"Die Datei ";ar$(0,1); [4122]
" wird sortiert":PRINT
2830 i=200 [506]
2840 WHILE ar$(i,1)="":i=i-1:WEND [1987]
2860 GOSUB 12000 [998]
2870 PRINT t$;"Datei ist sortiert" [2172]
2880 FOR i=1 TO 200:NEXT:CLS:RETURN [1533]
3000 '*** Datei neu anlegen [1483]
3010 GOSUB 9600:GOSUB 9400 [2109]
3070 GOSUB 1600:GOSUB 1700 [1664]
3071 ab=1 [361]
3080 LOCATE 3,22:PRINT "Bei Ende der Einga [7354]
be fuer Artikelnr. -/- eingeben"
3081 FOR i=ab TO 200 [596]
3090 GOSUB 3800 [993]
3160 NEXT i [375]
3200 CLS:RETURN [832]
3300 '*** Artikel loeschen aendern [1634]
3310 GOSUB 1600 [859]
3320 LOCATE 3,22:PRINT " ";CHR$(242);" Se [7456]
ite zurueck /";CHR$(243);" Seite vor /M Me
nue /L loeschen /A aendern";
3330 er=1:GOSUB 1800:an=48:en=57:wa$=CHR$( [2709]
242)+CHR$(243)+"MLA"
3340 GOSUB 9900:ON wa GOTO 3350,3370,3390, [2247]
3400,3590
3350 er=er-15:IF er<1 THEN er=1 [705]
3360 GOSUB 1800:GOTO 3340 [1056]
3370 er=er+15:IF er>185 THEN er=185 [997]
3380 GOTO 3360 [457]
3390 CLS:RETURN [832]
3400 GOSUB 3900:j=i:IF i=0 THEN GOTO 3360 [4901]
ELSE PRINT#1,"Einen Moment bitte!";
3410 FOR i=j TO 199 [1193]
3420 FOR k=0 TO 5 [1313]
3430 ar$(i,k)=ar$(i+1,k) [1589]
3440 NEXT k,i [376]
3450 FOR k=0 TO 5:ar$(200,k)="":NEXT k [2488]

```

```

3460 GOTO 3360 [457]
3590 GOSUB 3900:IF i=0 THEN GOTO 3360 ELSE [2825]
GOSUB 3800:GOTO 3340
3700 '*** Artikel anhaengen [882]
3710 GOSUB 1600 [859]
3720 PRINT#1,"Einen Moment bitte!";i=0 [1637]
3730 IF ar$(200,1)<>" THEN PRINT#1,"Datei [3764]
voll!";FOR i=1 TO 2000:NEXT i:RETURN
3740 WHILE ar$(i,1)<>"":i=i+1:WEND [1811]
3750 ab=i:er=ab-15:IF er<1 THEN er=1 [2146]
3755 GOSUB 1800 [883]
3760 GOTO 3080 [624]
3800 '*** Daten eingeben [1147]
3810 PRINT#1,"Artikel-Nr.: ";:an=46:en=57: [7466]
GOSUB 10000:IF ASC(an$)=47 THEN i=400:GOTO
3880 ELSE ar$(i,0)=RIGHT$("0000"+an$,4)
3820 PRINT#1,"Artikelbez. ";:an=32:en=122: [4171]
GOSUB 10000:ar$(i,1)=LEFT$(an$,24)
3830 PRINT#1,"Einheit: ";:GOSUB 10000:ar$( [4055]
i,2)=LEFT$(an$,4)
3840 PRINT#1,"Vk-Preis: ";:an=46:en=57:GOS [8337]
UB 10000:IF VAL(an$)>9999.99 OR VAL(an$)<0
.01 THEN GOSUB 4900:GOSUB 9800:GOTO 3840 E
LSE ar$(i,3)=an$
3850 PRINT#1,"Ek-Preis: ";:GOSUB 10000:IF [7517]
VAL(an$)>9999.99 OR VAL(an$)<0.01 THEN GOS
UB 4900:GOSUB 9800:GOTO 3850 ELSE ar$(i,4)
=an$
3860 PRINT#1,"Menge: ";:an=48:GOSUB 10000: [5797]
IF VAL(an$)>9999 THEN GOSUB 4900:GOTO 3850
ELSE ar$(i,5)=an$
3865 er=i-10:IF er<1 THEN er=1 [1808]
3870 GOSUB 1800 [883]
3880 RETURN [555]
3900 '*** Datensatz nach Artikelnr. suchen [2230]
3910 i=1:PRINT#1,"Artikelnr.: ";:an=46:en= [6987]
57:GOSUB 10000:IF an$="/" THEN GOTO 3940 E
LSE al$=RIGHT$("0000"+an$,4)
3920 WHILE al$<>ar$(i,0) AND i<200:i=i+1:W [3359]
END
3930 IF i>199 THEN PRINT#1,"Artikel nicht [6438]
gespeichert! Weiter mit Taste!";i=0:CALL &
BB06
3940 RETURN [555]
4500 '*** Abfrage Speichern [2246]
4510 PRINT t$;"Sollen die Daten gespeicher [7779]
t werden? Bei Nein werden Sie geloescht!"
4520 wa$="JN":GOSUB 9900 [1186]
4530 IF wa=1 THEN GOSUB 9000 [2041]
4540 GOSUB 9600 [1048]
4550 RETURN [555]
4900 '*** Fehlermeldung bei zu grosser Zah [2773]
l
4905 CLS#1 [373]
4910 PRINT#1,"Die Zahl ist in der Tabelle [4916]
nicht darstellbar!";
4920 FOR kk=1 TO 2000:NEXT kk [1366]
4930 CLS#1:RETURN [1242]
6400 '*** Preise einzeln aendern [1560]
6405 CLS:LOCATE 1,10 [846]
6410 GOSUB 3900:nr=i [904]
6420 PRINT USING "\ [4675]
alter VK-Preis: #####.## DM";ar$(nr,1);VAL
(ar$(nr,3))
6425 PRINT USING " [4718]
alter Ek-Preis: #####.## DM";VAL(ar$(nr,4)
)
6430 PRINT "neuer Vk-Preis?":GOSUB 10000:I [6780]
F VAL(an$)>9999.99 OR VAL(an$)<0.01 THEN G
OSUB 4900:GOTO 6430 ELSE ar$(nr,3)=an$
6435 PRINT "neuer Ek-Preis?":GOSUB 10000:I [5899]
F VAL(an$)>9999.99 OR VAL(an$)<0.01 THEN G
OSUB 4900:GOTO 6430 ELSE ar$(nr,4)=an$
6440 PRINT:PRINT "Weitere Preise aendern?" [4572]
:wa$="JN":GOSUB 9900
6450 IF wa=2 THEN CLS:RETURN [2594]
6460 GOTO 6400 [442]
9000 '*** Datei speichern [907]
9010 GOSUB 11000 [1038]
9020 CLS#1:PRINT#1,in$;"Die Datei ";na$;" [3950]
wird gespeichert!";in$;
9030 OPENOUT na$ [633]
9040 i=0 [420]
9050 WHILE ar$(i,1)<>" AND i<201 [1078]
9060 FOR j=0 TO 5:PRINT#9,ar$(i,j):NEXT j [1120]
9070 i=i+1 [444]
9080 WEND:CLOSEOUT [1782]
9090 CLS#1:CLS:RETURN [712]

```

Listing Fakturist

Listing Fakturist

```

9100 '*** Datei lesen [1144]
9120 GOSUB 11000:PRINT"Die Datei ";na$;" w [5047]
ird gelesen!";:FOR i=1 TO 1999:NEXT i
9125 i=0 [420]
9130 OPENIN na$ [589]
9140 WHILE NOT EOF [1840]
9150 FOR j=0 TO 5:INPUT#9,ar$(i,j):NEXT j [1264]
9160 i=i+1 [444]
9170 WEND:CLOSEIN [1209]
9175 ar$(0,1)=na$ [838]
9180 CLS:RETURN [832]
9200 '*** Dateinamen einlesen [1807]
9210 CLS:GOSUB 11000 [576]
9220 a$="*.DAT":|DIR,@a$ [805]
9230 PRINT"Name der Datei ohne Extension e [5374]
ingeben!"
9240 an=45:en=122:GOSUB 10000 [1666]
9250 IF LEN(an$)>8 THEN GOSUB 9800:GOTO 92 [3464]
40
9260 IF LEN(an$)<8 THEN an$=an$+" ":GOTO 9 [1751]
260
9270 na$=UPPER$(an$)+".LIF":ar$(0,1)=na$ [1865]
9280 RETURN [555]
9400 '*** Dateinamen vorgeben [1239]
9410 na$="WAREN.DAT" [1103]
9420 ar$(0,1)=na$ [838]
9430 RETURN [555]
9600 '*** Datei loeschen [1117]
9610 CLS#1:PRINT#1,"Einen Moment bitte!";: [2205]
i=0
9620 WHILE ar$(i,1)<>" "AND i<201 [1078]
9630 FOR j=0 TO 5:ar$(i,j)="":NEXT j [1746]
9640 i=i+1:WEND [758]
9650 CLS#1:RETURN [1242]
9800 '*** FEHLEINGABE [1735]
9810 PRINT bi$;:PRINT#1,TAB(30);in$;" FEHL [2451]
EINGABE ";in$;
9820 FOR kk=1 TO 1000:NEXT kk:PRINT bi$;:P [3257]
RINT#1
9830 RETURN [555]
9900 '*** AUSWAHL [946]
9910 CLS#1:PRINT#1,"Auswahl (";wa$;");";bi [2953]
$;
9920 y$=UPPER$(INKEY$):IF y$=""THEN 9920 [3456]
9930 wa=INSTR(wa$,y$):IF wa=0 THEN GOSUB 9 [4460]
800:GOTO 9910
9940 CLS#1:RETURN [1242]
10000 '*** Abfrage [1291]
10010 PRINT#1,"Eingabe: ";:an$="" [1383]
10020 y$=INKEY$:IF y$=""THEN 10020 [2649]
10030 IF ASC(y$)=13 THEN CLS#1:RETURN [1749]
10032 IF ASC(y$)=127 THEN an$=LEFT$(an$,LE [3771]
N(an$)-1):CLS#1:PRINT#1,"Eingabe: ";an$;
10040 IF ASC(y$)<an OR ASC(y$)>en THEN PRI [4028]
NT bi$;:GOTO 10020
10050 PRINT#1,y$;:an$=an$+y$:y$=""GOTO 10 [2289]
020
11000 '*** Laufwerkstatus abfragen (aus CP [3846]
C International 01.86)
11005 RETURN [555]
11010 OUT(&FA7E),1:FOR i=1 TO 1000:NEXT i [1828]
11020 OUT(&FB7F),4:OUT(&FB7F),(-PEEK(&A700 [1904]
)+2):
11030 st=INP(&FB7F):OUT(&FA7E),0 [1514]
11040 rd=st AND 32 [781]
11050 wp=st AND 64 [319]
11060 IF rd=0 THEN PRINT#1,"Disk richtig e [5212]
inlegen! Weiter mit Taste":CALL &BB06:GOTO
11000
11070 IF wp=64 THEN PRINT#1,"Schreibschutz [7566]
auf Disk entfernen! Weiter mit Taste":CAL
L &BB06:GOTO 11000
11080 CLS#1:FOR ii=1 TO 1000:NEXT ii:RETUR [1953]
N
12000 '*** Quick Sort [1260]
12010 DIM sta(20,2):DIM help$(5) [1705]
12020 ti=1 [243]
12030 sta(ti,1)=1:sta(ti,2)=i [1353]
12040 WHILE ti<>0 [1208]
12050 li=sta(ti,1):re=sta(ti,2) [836]
12060 ti=ti-1 [226]
12070 WHILE li<re [953]
12080 i=li:j=re:pr$=ar$((li+re)/2,g) [2709]
12090 WHILE i<=j [891]
12100 WHILE ar$(i,g)<pr$:i=i+1:WEND [1595]
12110 WHILE ar$(j,g)>pr$:j=j-1:WEND [1798]
12120 IF i<=j THEN GOSUB 13000:i=i+1:j=j-1 [2034]
12130 WEND [390]
12140 IF i<re THEN ti=ti+1:sta(ti,1)=i:sta [2146]

```

Listing Fakturist

```

(ti,2)=re
12150 re=j [158]
12160 WEND [390]
12170 WEND [390]
12180 ERASE sta:ERASE help$ [1404]
12190 RETURN [555]
13000 '*** Tausch fuer Quicksort [871]
13010 FOR k=0 TO 5 [1313]
13020 help$(k)=ar$(i,k) [884]
13030 ar$(i,k)=ar$(j,k) [1473]
13040 ar$(j,k)=help$(k) [990]
13050 NEXT k [373]
13060 RETURN [555]
15000 '*** Menugenerator [1386]
15010 CLS:READ a1$ [730]
15020 LOCATE 24,5:PRINT a1$:PRINT [1891]
15030 READ a1$,a2$ [509]
15040 WHILE a1$<>"x" [1728]
15050 PRINT TAB(20);in$;" ";a1$;" ";in$;" [2406]
";a2$
15060 READ a1$,a2$ [509]
15070 WEND [390]
15080 READ wa$:PRINT:PRINT [1707]
15090 RETURN [555]
29000 '*** Fehlerbehandlung [1559]
29010 CLS#1:PRINT#1,"Fehler Nr. ";ERR;" in [2140]
Zeile ";ERL
29020 FOR ii=1 TO 2000:NEXT ii [1314]
29030 RESUME NEXT [703]

```



```

2 '***** [1391]
3 '*** LIEFERUN.BAS V4.0 **** [2036]
4 '*** WIESOFT 1987 **** [1208]
5 '*** Heinz Wienhoven **** [1817]
9 '***** [1391]
10 zeis="" [459]
20 in$=CHR$(24):bi$=CHR$(7) [1073]
30 i=0:wa=0:wa$="":y$="":an=0:en=0:dat$="" [2144]
:na$="":an$="":k=0:j=0
32 z$="!":t$=STRING$(9,32) [1437]
40 DIM ar$(200,5):DIM lif$(50,5):DIM rech$ [2243]
(50,3)
170 '*** Datum einlesen [1681]
180 OPENIN "DATUM.DAT":INPUT#9,dat$:INPUT# [4251]
9,dat1$:CLOSEIN
190 OPENIN "LW.DAT":INPUT#9,lw$:CLOSEIN [2160]
200 MODE 2 [513]
210 PRINT "Faktu-PRO'87";TAB(24);"LIEFERUN [3927]
G";TAB(60);"Datum: ";dat$
310 ON ERROR GOTO 29000 [1503]
400 '*** Windows [637]
410 WINDOW#0,1,80,2,23 [1198]
420 WINDOW#1,1,80,25,25 [1010]
430 WINDOW#2,1,9,6,21 [1403]
440 WINDOW#3,11,35,6,21 [954]
450 WINDOW#4,37,41,6,21 [1406]
460 WINDOW#5,43,50,6,21 [1236]
500 wa=1:GOTO 1020 [851]
1000 '*** Hauptmenue [812]
1005 CLS#1 [373]
1010 RESTORE 1015:GOSUB 15000:GOSUB 9900 [1940]
1015 DATA Hauptmenue,L,Liefer- Bestellsche [5028]
in bearbeiten,H,Hauptprogramm,x,x,LH
1020 ON wa GOSUB 4000,1500 [1942]
1030 GOTO 1010 [345]
1500 '*** Programmende [1990]
1520 RUN "AUSWAHL.BAS" [723]
1600 '*** Tabellenaufbau [1249]
1610 CLS:PRINT:PRINT" Datei: ";ar$(0,1) [2166]
1620 PRINT" Art.Nr. Artikel";TAB(37);"Ein [4000]
h Menge"
1630 MOVE 8,336:DRAW 404,336:MOVE 76,350:D [3413]
RAW 76,60:MOVE 284,350:DRAW 284,60
1640 MOVE 332,350:DRAW 332,60:MOVE 404,350 [2968]
:DRAW 404,60
1660 RETURN [555]

```

Listing Fakturist


```
'00 '*** Windows loeschen [1342]
710 CLS#2:CLS#3:CLS#4:CLS#5:RETURN [1645]
300 '*** Datensatz nach Artikelnr. suchen [2230]
910 i=1:PRINT#1,"Artikelnr. ";:an=46:en= [6987]
7:GOSUB 10000:IF an$="/ THEN GOTO 3940 E
SE al$=RIGHT$( "0000"+an$,4)
920 WHILE al$<>ar$(i,0) AND i<200:i=i+1:W [3359]
JND
3930 IF i>199 THEN PRINT#1,"Artikel nicht [6438]
gespeichert! Weiter mit Taste!":i=0:CALL &
BB06
3940 RETURN [555]
4000 '*** Lieferschein bearbeiten [2426]
4001 CLS:GOSUB 9400:GOSUB 9100 [1077]
4002 IF zeis$<>"X" THEN 4010 [712]
4003 LOCATE 10,10:PRINT "Es sind bereits m [6676]
ehr als 40 Daten in der Summendatei"
4004 PRINT t$;"gespeichert. Es koennen nur [8039]
noch 5 Lieferscheine erfasst werden. Sie
sollten jetzt eine Rechnung ers
tellen"
4005 PRINT t$;"Weiter mit Taste!":CALL &BB [2113]
06
4006 CLS [91]
4010 RESTORE 4015:GOSUB 15000:GOSUB 9900 [2434]
4015 DATA Lieferschein bearbeiten,N,neuen [11626]
Lieferschein erstellen,B,Bestellschein ers
tellen,L,Lieferschein drucken,D,Bestellsch
ein drucken,M,Menue,x,x,NBLDM
4020 an=46:en=57:IF wa=5 THEN lif$(0,1)=" [1446]
:RETURN
4030 ON wa GOSUB 4100,6700,5000,5200 [2664]
4040 GOTO 4002 [543]
4100 '*** neuen Lieferschein erstellen [3985]
4105 GOSUB 4300:fak=-1 [1622]
4110 OPENIN"NR.LID":INPUT#9,nu:CLOSEIN [3094]
4130 nu=nu+1:n$=STR$(nu):nu$=RIGHT$(n$,LEN [5065]
(n$)-1):num$=RIGHT$( "000"+nu$,4)
4140 na$="L"+num$+"-"+RIGHT$(dat$,2)+".LIF [3461]
":lif$(0,1)=na$:lif$(0,0)=ar$(0,1):lif$(0,
2)=dat1$
4150 GOSUB 4400:IF lif$(1,0)=" THEN CLS:G [3130]
OTO 4230
4200 GOSUB 1700:CLS:PRINT#1,"Dieser Liefer [7437]
schein erhaelt die Nummer: ";num$;:GOSUB 9
000
4210 GOSUB 8800:OPENOUT"NR.LID":PRINT#9,nu [4557]
:CLOSEOUT:a$="*.BAK":|ERA,@a$
4220 GOSUB 4800:GOSUB 8600 [1972]
4230 RETURN [555]
4300 '*** Felder lif$ leer [1469]
4310 FOR j=0 TO 50 [575]
4320 FOR i=0 TO 5:lif$(j,i)="":NEXT i [1465]
4330 NEXT j:RETURN [844]
4400 '*** Dateneigabe fuer Bestell- Liefer [3855]
schein
4403 GOSUB 6000 [907]
4405 GOSUB 1600:GOSUB 1700:LOCATE 70,10:PR [4651]
INT "lfd. Nr."
4410 LOCATE 3,22:PRINT"Bei Ende der Eingab [7618]
e fuer Artikelnr. -/- eingeben!":nr=1
4420 LOCATE 72,12:PRINT in$;nr;in$ [1418]
4425 GOSUB 3900:IF an$="/ THEN GOTO 4460 [2346]
4430 IF i=0 THEN 4420 [1029]
4435 PRINT#1,ar$(i,1); [1008]
4440 PRINT#1," Menge ";:GOSUB 10000:IF VAL [11069]
(an$)>9999 THEN GOSUB 4900:GOTO 4420 ELSE
IF an$="0" THEN 4420 ELSE FOR j=0 TO 4:lif
$(nr,j)=ar$(i,j):NEXT j:GOSUB 7000:IF an$=
"" THEN GOTO 4425 ELSE lif$(nr,5)=an$
4450 GOSUB 4600:nr=nr+1:IF nr>45 THEN GOTO [2947]
4460 ELSE GOTO 4420
4460 LOCATE 3,22:PRINT"Daten O.K.?" ;t$;t$; [4692]
t$;t$;t$:wa$="JN":GOSUB 9900:IF wa=2 THEN
GOSUB 4700
4470 RETURN [555]
4600 '*** Lieferschein in Tabelle je ein D [5510]
atensatz
4610 PRINT#2," ";lif$(nr,0):PRINT#3,lif$( [3753]
nr,1):PRINT#4,lif$(nr,2)
4640 PRINT#5,USING"#####";VAL(lif$(nr,5)) [1629]
4650 RETURN [555]
4700 '*** Lieferschein Datensaeetze aendern [4934]
und loeschen
4710 j=1:LOCATE 3,22:PRINT"Zur Aenderung A [4875]
rt. Nr. eingeben. Naechstes Bild mit ENTER
!";
4720 GOSUB 1700:FOR nr=j TO j+14:GOSUB 460 [2071]
```

Listing Fakturist

```
0:NEXT nr
4740 GOSUB 6500:IF an$=""AND j=35 THEN RET [3560]
URN
4750 IF an$=""THEN j=j+15:IF j>35 THEN j=3 [2763]
5:GOTO 4720 ELSE GOTO 4720
4760 PRINT#1,"Neue Menge fuer Art. Nr. ";1 [3964]
if$(i,0);:GOSUB 10000
4770 IF an$="0"THEN GOSUB 6600 ELSE lif$(i [1986]
,5)=an$
4780 GOTO 4720 [537]
4800 '*** Berechnung [872]
4810 j=1:PRINT#1,"Berechnung! Einen Moment [4300]
bitte!";:sum1=0:sum2=0
4820 WHILE lif$(j,1)<>" AND j<51 [1685]
4830 sum1=sum1+(VAL(lif$(j,3))*VAL(lif$(j, [2743]
5)))
4840 sum2=sum2+(VAL(lif$(j,4))*VAL(lif$(j, [1187]
5)))
4850 j=j+1:WEND [664]
4860 RETURN [555]
4900 '*** Fehlermeldung bei zu grosser Zah [2773]
l
4905 CLS#1 [373]
4910 PRINT#1,"Die Zahl ist in der Tabelle [4916]
nicht darstellbar!";
4920 FOR kk=1 TO 2000:NEXT kk [1366]
4930 CLS#1:RETURN [1242]
5000 '*** Lieferschein ausdrucken [3070]
5004 zz$="Lieferschein ":fak=-1 [2775]
5006 IF lif$(0,1)=" THEN a$=Lw$+"L*.lieF": [1743]
GOSUB 8900
5008 a$=LEFT$(lif$(0,1),1):IF a$<>"L"THEN [8326]
PRINT#1,"Im Speicher befindet sich kein Li
eferschein! Taste";:CALL &BB06:lif$(0,1)="
":GOTO 5006
5010 RESTORE 5015:GOSUB 15000:GOSUB 9900 [2766]
5015 DATA Lieferschein drucken,B,Druck mit [7875]
Warenliste,L,nur Lieferschein drucken,M,M
enue,x,x,BLM
5016 IF wa=3 THEN RETURN [2034]
5017 IF wa=1 THEN GOSUB 5400:GOSUB 5900 EL [3270]
SE GOSUB 5920
5018 jj=3' 3=kein Doppel-, 2=ein Doppel-, [3459]
1=zwei Doppeldrucke
5020 CLS:LOCATE 3,22:PRINT"Die Daten werde [4967]
n ausgedruckt!":GOSUB 5800
5030 PRINT#8,t$;CHR$(14);zz$;LEFT$(lif$(0, [4916]
1),5);CHR$(20);TAB(45);"Datum: ";lif$(0,2)
5040 PRINT#8 [343]
5050 PRINT#8,t$;"!Art.Nr. !";TAB(21);"Arti [5362]
kelbezeichnung";TAB(48);"! Preis !"
;TAB(67);"Menge !"
5060 PRINT#8,t$;STRING$(68,"-"):i=1 [1797]
5070 FOR i=1 TO 45 [548]
5075 IF lif$(i,1)<>" THEN 5079 [1397]
5077 PRINT#8," ! ! ! [3963]

! ! !
!":GOTO 5090
5079 IF fak=1 THEN GOTO 5085 [338]
5080 PRINT#8,USING" ! \ \ \ \ [7740]
! ! ! ! !
#### \ \ !";z$;lif$(i,0);z$;lif$(i,1);z$
;z$;VAL(lif$(i,5));lif$(i,2);z$:GOTO 5090
5085 PRINT#8,USING" ! \ \ \ \ [7459]
! ! ! ! !
#### \ \ !";z$;lif$(i,0);z$;lif$(i,1);z$
;VAL(lif$(i,3));z$;VAL(lif$(i,5));lif$(i,2
);z$
5090 NEXT i [375]
5100 PRINT#8,t$;STRING$(68,"-") [1531]
5110 PRINT#8:PRINT#8 [1314]
5120 PRINT#8,t$;CHR$(15);"Wir danken f]r d [5500]
as uns entgegengebrachte Vertrauen";CHR$(1
8)
5130 IF jj=3 THEN PRINT#8,CHR$(12);:GOSUB [3101]
4300:RETURN
5140 IF jj=1 THEN GOSUB 5900:GOSUB 5800:jj [1908]
=2:GOTO 5030
5150 GOSUB 5900:GOSUB 5800:jj=3:GOTO 5030 [2818]
5200 '*** Bestellschein drucken [2836]
5210 zz$="Bestellschein ":fak=1:IF lif$(0, [5823]
1)=" THEN a$="B*.LIF":GOSUB 8900
5220 a$=LEFT$(lif$(0,1),1):IF a$<>"B"THEN [10417]
PRINT#1,"Im Speicher befindet sich kein Be
stellschein! Taste";:CALL &BB06:CLS#1:lif$
(0,1)="":GOTO 5200
5230 GOSUB 5920:GOTO 5018 [1340]
5400 '*** Warenliste drucken [1418]
```

Listing Fakturist

```

5410 na$=lif$(0,0):GOSUB 9100 [1175]
5420 LOCATE 3,22:PRINT"Die Warenliste wird gedru [4505]
gedruckt!":bl=1:GOSUB 5920:ze=10
5430 bla$="!-----! [3342]
-----!"
5440 GOSUB 5720 [915]
5450 FOR ll=1 TO i [1085]
5460 PRINT#8,USING" !###!\ [8399]
\!\ \!\ ####.## !
.....!";z$;ll;z$;ar$(ll,1);z$;ar$(1
1,0);z$;VAL(ar$(ll,3));z$;z$
5480 ze=ze+1 [935]
5490 IF ze>60 THEN bl=bl+1:GOSUB 5600:GOSU [3952]
B 5700
5500 NEXT ll [452]
5510 GOSUB 5600:CLS:CLS#1 [2141]
5520 RETURN [555]
5600 '*** Unterschrift [1693]
5610 PRINT#8:PRINT#8:PRINT#8,t$;"Unterschr [4899]
ift : . . . . . "
5620 RETURN [555]
5700 '*** Bestellkopf [1719]
5710 GOSUB 5900:ze=10 [1382]
5720 PRINT#8,t$;CHR$(14);"Warenliste";CHR$ [3688]
(20);TAB(45);"Datum: ";lif$(0,2)
5730 PRINT#8,t$;"Blatt: ";bl [2011]
5740 PRINT#8,t$;"! Nr. ! Artikelbezeichnun [6075]
g ! Art.Nr. ! Preis ! Bestellung
!"
5750 PRINT#8,t$;bla$ [1287]
5760 RETURN [555]
5800 '*** Empfaengerfirma eindrucken [2142]
5810 PRINT#8,t$;lif$(0,3) [861]
5820 PRINT#8,t$;lif$(0,4) [1023]
5830 PRINT#8,t$;lif$(0,5):PRINT#8:PRINT#8 [1743]
5840 RETURN [555]
5900 '*** Druckkopf fuer Lieferschein [1513]
5910 PRINT#8,CHR$(12); [1101]
5920 PRINT#8," B[ckerei Konditorei [4985]
";TAB(60);"4240 Emmerich-Elten"
5930 PRINT#8," Pusemockel ";TA [3289]
B(65);"Kloberstr. 8"
5940 PRINT#8," Liefer.Nr.: 841 000 [4627]
000";TAB(65);"Tel.:02828/4711"
5950 FOR zz=1 TO 3:PRINT#8:NEXT zz [2823]
5960 RETURN [555]
6000 '*** Empfaenger eingeben [2552]
6010 CLS:GOSUB 1700:LOCATE 2,5:PRINT"Bitte [4212]
den Empfaenger eingeben!"
6020 LOCATE 2,7:PRINT"Bitte den Vor- und N [3059]
achnamen eingeben!"
6030 an=32:en=122:GOSUB 10000:lif$(0,3)=an [1052]
$
6040 LOCATE 2,7:PRINT"Name : ";lif$(0,3) [2752]
;"
6050 LOCATE 2,8:PRINT"Bitte Strasse und Ha [5125]
usnr. eingeben!":GOSUB 10000
6060 lif$(0,4)=an$:LOCATE 2,8:PRINT"Strass [4363]
e: ";lif$(0,4);"
6070 LOCATE 2,9:PRINT"Bitte PLZ und Ort ei [3108]
ngeben!":GOSUB 10000
6080 lif$(0,5)=an$:LOCATE 2,9:PRINT"Ort [3040]
: ";lif$(0,5);"
6090 LOCATE 2,11:PRINT"Daten o.k. (JN)? ": [5315]
wa$="JN":GOSUB 9900
6100 IF wa=2 THEN GOTO 6010 [1378]
6110 RETURN [555]
6500 '*** Lieferschein auf Artikelnr durch [4029]
suchen
6510 i=0:PRINT#1,"Art. Nr. : ";an=46:en=57 [3123]
:GOSUB 10000:IF an$=" THEN RETURN
6520 al$=RIGHT$("0000"+an$,4) [2116]
6530 WHILE al$<>lif$(i,0)AND i<46:i=i+1:WE [2993]
ND
6540 IF i>44 THEN PRINT#1,"Artikel nicht g [4041]
efunden! Weiter mit Taste!":i=0:CALL &BB06
6550 RETURN [555]
6600 '*** Datensatz aus Lieferschein loesc [1741]
hen
6610 FOR k=i TO 49 [929]
6620 FOR kk=0 TO 5 [1038]
6630 lif$(k,kk)=lif$(k+1,kk) [907]
6640 NEXT kk:NEXT k [880]
6650 FOR k=0 TO 5:lif$(50,k)="" :NEXT k [1546]
6660 RETURN [555]
6700 '*** Bestellschein bearbeiten [1901]
6710 GOSUB 4300:OPENIN"RNR.LID":INPUT#9,nu [3647]
:CLOSEIN

```

Listing Fakturist

```

6720 fak=1 [128]
6730 nu=nu+1:n$=STR$(nu):nu$=RIGHT$(n$,LEN [5065]
(n$)-1):num$=RIGHT$("000"+nu$,4)
6740 na$="B"+num$+"-"+RIGHT$(dat$,2)+" .LIF [3460]
":lif$(0,1)=na$:lif$(0,0)=ar$(0,1):lif$(0,
2)=dat$
6750 GOSUB 4400:IF lif$(1,0)="" THEN CLS:G [3378]
OTO 6790
6760 GOSUB 1700:CLS:PRINT#1,"Dieser Bestel [7666]
lschein erhaelt die Nummer: ";num$;:GOSUB
9000
6770 GOSUB 8800:OPENOUT"RNR.LID":PRINT#9,n [5251]
u:CLOSEOUT:a$="*.BAK":ERA,@a$
6780 GOSUB 4800:GOSUB 8000:sne=sne+sum1:GO [3313]
SUB 8100
6790 RETURN [555]
7000 '*** Lagerbestand anpassen [2580]
7010 IF VAL(an$)>VAL(ar$(i,5)) AND fak=-1 [9846]
THEN PRINT#1,"Es sind nur noch ";ar$(i,5);
" auf Lager! Weiter mit Taste";an$="":CAL
L &BB06:GOTO 7040
7020 wert=VAL(an$):wert=wert*fak:wert=wert [3823]
+VAL(ar$(i,5))
7030 wert$=STR$(wert):ar$(i,5)=RIGHT$(STR$ [3651]
(wert),LEN(wert$)-1)
7040 CLS#1 [373]
7050 RETURN [555]
7800 '*** Berechnung fuer Rechnung [1724]
7810 sumvk=0:sumne=0:PRINT#1,in$;"BERECHNU [2333]
NG";in$
7820 FOR kk=1 TO ii [959]
7830 vk=VAL(rech$(kk,2)):ne=VAL(rech$(kk,3 [3034]
)):a$=LEFT$(rech$(kk,1),1)
7840 IF a$="L"THEN sumvk=sumvk+vk:sumne=su [5788]
mne+ne ELSE sumvk=sumvk-vk:sumne=sumne-ne
7845 IF a$="R"THEN rech$(kk,2)="-"+rech$(k [4490]
k,2):rech$(kk,3)="-"+rech$(kk,3)
7850 NEXT kk [455]
7860 steu=sumne*0.07:rab=sumvk-sumne:pro=( [3743]
100/sumvk)*rab
7870 CLS#1:RETURN [1242]
8000 '*** Rechnungssumme lesen [3596]
8010 OPENIN "RECH.SUM" [1285]
8020 INPUT#9,svk [903]
8030 INPUT#9,sne [1013]
8040 CLOSEIN [752]
8050 RETURN [555]
8100 '*** Rechnungssumme speichern [3626]
8110 OPENOUT "RECH.SUM" [1315]
8120 PRINT#9,svk [1353]
8130 PRINT#9,sne [1335]
8140 CLOSEOUT [902]
8150 RETURN [555]
8600 '*** Lieferscheinsumme in Summendatei [3353]
speichern
8620 OPENIN"SUMM.REC" [925]
8630 i=1 [423]
8640 WHILE NOT EOF [1840]
8650 INPUT#9,rech$(i,0),rech$(i,1),rech$(i [2463]
,2),rech$(i,3)
8660 i=i+1:WEND [758]
8670 CLOSEIN [752]
8680 rech$(i,1)=na$:rech$(i,2)=STR$(sum1): [6762]
rech$(i,3)=STR$(sum2):rech$(i,0)=lif$(0,2)
:IF i>40 THEN zei$="X"
8690 OPENOUT"SUMM.REC":FOR j=1 TO i:FOR k= [8114]
0 TO 3:PRINT#9,rech$(j,k):NEXT k:NEXT j:CL
OSEOUT:a$="*.BAK":ERA,@a$:RETURN
8800 '*** Lieferschein speichern [2825]
8810 GOSUB 11000:CLS#1:na$=lw$+na$ [1489]
8820 PRINT#1,in$;"Die Datei ";na$;" wird g [5128]
espeichert!";in$;
8830 OPENOUT na$ [633]
8840 i=0 [420]
8850 WHILE lif$(i,1)<>""AND i<51 [1178]
8860 FOR j=0 TO 5:PRINT#9,lif$(i,j):NEXT j [1943]
8870 i=i+1:WEND [758]
8880 CLOSEOUT:CLS#1:CLS:RETURN [2201]
8900 '*** Lieferschein lesen [2858]
8910 GOSUB 11000 [1038]
8920 GOSUB 9200:CLS#1 [1482]
8930 PRINT#1,"Der Lieferschein ";na$;" wir [4859]
d gelesen!";i=0
8940 OPENIN na$ [589]
8950 WHILE NOT EOF [1840]
8960 FOR j=0 TO 5:INPUT#9,lif$(i,j):NEXT j [1894]
8970 i=i+1:WEND [758]
8980 CLOSEIN:CLS#1 [931]

```

Listing Fakturist

```

:990 RETURN [555]
9000 '*** Warendatei speichern [1871]
9010 GOSUB 11000 [1038]
9020 CLS#1:PRINT#1,in$;"Die Datei ";ar$(0, [4471]
1);" wird aktualisiert!";in$;
9030 OPENOUT ar$(0,1) [731]
9040 i=0 [420]
9050 WHILE ar$(i,1)<>" " AND i<201 [1078]
9060 FOR j=0 TO 5:PRINT#9,ar$(i,j):NEXT j [1120]
9070 i=i+1 [444]
9080 WEND [390]
9090 CLOSEOUT:CLS#1:RETURN [1953]
9100 '*** Datei lesen [1144]
9120 GOSUB 11000:PRINT#1,"Die Datei ";na$; [5905]
" wird gelesen!";FOR i=1 TO 1999:NEXT i
9125 i=0 [420]
9130 OPENIN na$ [589]
9140 WHILE NOT EOF [1840]
9150 FOR j=0 TO 5:INPUT#9,ar$(i,j):NEXT j [1264]
9160 i=i+1 [444]
9170 WEND:CLOSEIN [1209]
9175 ar$(0,1)=na$ [838]
9180 CLS:RETURN [832]
9200 '*** Dateinamen einlesen [1807]
9210 CLS:GOSUB 11000 [576]
9220 |DIR,@a$ [883]
9230 PRINT"Name der Datei ohne Extension e [5374]
ingeben!"
9240 an=45:en=122:GOSUB 10000 [1666]
9250 IF LEN(an$)>8 THEN GOSUB 9800:GOTO 92 [3464]
40
9260 IF LEN(an$)<8 THEN an$=an$+" ":GOTO 9 [1751]
260
9270 na$=UPPER$(an$)+".LIF":ar$(0,1)=na$ [1865]
9280 RETURN [555]
9400 '*** Dateinamen vorgeben [1239]
9410 na$="WAREN.DAT" [1103]
9420 ar$(0,1)=na$ [838]
9430 RETURN [555]
9800 '*** FEHLEINGABE [1735]
9810 PRINT bi$;:PRINT#1,TAB(30);in$;" FEHL [2451]
EINGABE ";in$;
9820 FOR kk=1 TO 1000:NEXT kk:PRINT bi$;:P [3257]
RINT#1
9830 RETURN [555]
9900 '*** AUSWAHL [946]
9910 CLS#1:PRINT#1,"Auswahl (";wa$;"):";bi [2953]
$;
9920 y$=UPPER$(INKEY$):IF y$=""THEN 9920 [3456]
9930 wa=INSTR(wa$,y$):IF wa=0 THEN GOSUB 9 [4460]
800:GOTO 9910
9940 CLS#1:RETURN [1242]
10000 '*** Abfrage [1291]
10010 PRINT#1,"Eingabe: ";:an$="" [1383]
10020 y$=INKEY$:IF y$=""THEN 10020 [2649]
10030 IF ASC(y$)=13 THEN CLS#1:RETURN [1749]
10032 IF ASC(y$)=127 THEN an$=LEFT$(an$,LE [3771]
N(an$)-1):CLS#1:PRINT#1,"Eingabe: ";an$;
10040 IF ASC(y$)<an OR ASC(y$)>en THEN PRI [4028]
NT bi$;:GOTO 10020
10050 PRINT#1,y$;:an$=an$+y$:y$="" :GOTO 10 [2289]
020
11000 '*** Laufwerkstatus abfragen (aus CP [3846]
C International 01.86)
11005 RETURN [555]
11010 OUT(&FA7E),1:FOR i=1 TO 1000:NEXT i [1828]
11020 OUT(&FB7F),4:OUT(&FB7F),(-PEEK(&A700 [1904]
)+2):
11030 st=INP(&FB7F):OUT(&FA7E),0 [1514]
11040 rd=st AND 32 [781]
11050 wp=st AND 64 [319]
11060 IF rd=0 THEN PRINT#1,"Disk richtig e [5212]
inlegen! Weiter mit Taste":CALL &BB06:GOTO
11000
11070 IF wp=64 THEN PRINT#1,"Schreibschutz [7566]
auf Disk entfernen! Weiter mit Taste":CAL
L &BB06:GOTO 11000
11080 CLS#1:FOR ii=1 TO 1000:NEXT ii:RETUR [1953]
N
15000 '*** Menuegenerator [1386]
15010 CLS:READ a1$ [730]
15020 LOCATE 24,5:PRINT a1$:PRINT [1891]
15030 READ a1$,a2$ [509]
15040 WHILE a1$<>"x" [1728]
15050 PRINT TAB(20);in$;" ";a1$;" ";in$;" [2406]
";a2$
15060 READ a1$,a2$ [509]
15070 WEND [390]

```

Listing Fakturist

```

15080 READ wa$:PRINT:PRINT [1707]
15090 RETURN [555]
29000 '*** Fehlerbehandlung [1559]
29010 CLS#1:PRINT#1,"Fehler Nr. ";ERR;" in [2140]
Zeile ";ERL
29020 FOR ii=1 TO 2000:NEXT ii [1314]
29030 RESUME NEXT [703]

```



```

2 '***** [1391]
3 '*** RECHNUNG.BAS V5.0 *** [1201]
4 '*** WIESOFT 1987 *** [1208]
5 '*** Heinz Wienhoven *** [1817]
9 '***** [1391]
20 in$=CHR$(24):bi$=CHR$(7) [1073]
30 i=0:dat$="":an$="":j=0 [1522]
32 z$="!":t$=STRING$(9,32):ba$="*.BAK" [1621]
40 DIM lif$(50,5):DIM rech$(50,3) [1182]
170 '*** Datum einlesen [1681]
180 OPENIN "DATUM.DAT":INPUT#9,dat$:INPUT# [4251]
9,dat1$:CLOSEIN
190 OPENIN "LW.DAT":INPUT#9,lw$:CLOSEIN [2160]
200 MODE 2 [513]
210 PRINT "Faktu-PRO'87";TAB(24);"RECHNUNG [5108]
";TAB(60);"Datum: ";dat$
310 ON ERROR GOTO 29000 [1503]
400 '*** Windows [637]
410 WINDOW#0,1,80,2,23 [1198]
420 WINDOW#1,1,80,25,25 [1010]
1000 CLS:PRINT#1,"Sollen die Rechnungen er [4934]
stellt werden (JN)? ";bi$;
1010 an$=UPPER$(INKEY$):IF an$="" THEN 101 [2293]
0
1020 IF an$="N" THEN 1050 [1324]
1030 IF an$="J" THEN CLS#1:GOSUB 7000:GOTO [2775]
1050
1040 PRINT#1,TAB(10);in$;" Fehleingabe ";i [2900]
n$;FOR i=1 TO 2000:NEXT i:GOTO 1000
1050 RUN "AUSWAHL.BAS" [723]
2000 '*** Anzahl der Lieferscheine einlese [4108]
n
2010 ii=0 [215]
2020 OPENIN "SUMM.REC" [925]
2030 WHILE NOT EOF:FOR jj=0 TO 3:INPUT#9,r [6370]
ech$(ii,jj):NEXT jj:ii=ii+1:WEND
2040 CLOSEIN [752]
2050 RETURN [555]
4300 '*** Felder lif$ leer [1469]
4310 FOR x=0 TO 50:FOR y=0 TO 5:lif$(x,y)= [3576]
"":NEXT y:NEXT x
4320 RETURN [555]
5900 '*** Druckkopf fuer Rechnung [1806]
5910 PRINT#8,CHR$(12); [1101]
5920 PRINT#8," B{ckerei Konditorei [4985]
";TAB(60);"4240 Emmerich-Elten"
5930 PRINT#8," Pusemoekel ";TA [3289]
B(65);"Kloberstr. 8"
5940 PRINT#8," Liefer.Nr.: 841 000 [4627]
000";TAB(65);"Tel.:02828/4711"
5950 FOR zz=1 TO 3:PRINT#8:NEXT zz [2823]
5960 RETURN [555]
7000 '*** Rechnungen erstellen [1223]
7010 CLS:LOCATE 10,10:PRINT in$;" Die Rech [5816]
nungen werden erstellt ";in$
7020 GOSUB 2000:kk=1:sumvk=0 [1710]
7030 WHILE kk<ii [1018]
7040 num$=MID$(rech$(kk,1),3,5)+" vom "+re [2867]
ch$(kk,0)
7050 LOCATE 10,12:PRINT USING"Es muessen n [5327]
och ## Lieferscheine bearbeitet werden!";i
i-kk
7060 GOSUB 8900:anz=1' anz= Anzahl der Aus [3882]
drucke je Rechnung (2= ein Doppel)

```

Listing Fakturist

Anwendungen

```

7070 GOSUB 5900:GOSUB 7700:sum=0 [2386]
7080 PRINT#8,t$;CHR$(14);"Rechnung";CHR$(2 [3922]
0);TAB(55);"Datum: ";dat$
7090 PRINT#8,t$;"zu Lieferschein ";num$:PR [2852]
INT#8
7100 PRINT#8,t$;"!Art.Nr.! Artikelbez. [5656]
! Preis ! Menge ! Gesamtpreis
!"
7110 PRINT#8,t$;"!-----!----- [4154]
!-----!-----!-----
--!"
7120 FOR ll=1 TO 45 [1059]
7130 IF lif$(ll,0)=" THEN PRINT#8,t$;"! [5180]
!
! ! !:GOTO 7160
7140 ges=VAL(lif$(ll,5))*VAL(lif$(ll,3)):s [3401]
um=sum+ges
7150 PRINT#8, USING " ! \ \ ! \ [7774]
\ ! ####.## ! #### !
#####.## DM!";z$;lif$(ll,0);z$;lif$(ll,
1);z$;VAL(lif$(ll,3));z$;VAL(lif$(ll,5));z
$;ges;z$
7160 NEXT ll [452]
7170 PRINT#8,t$;"!-----!----- [4154]
!-----!-----!-----
--!"
7180 PRINT#8, USING " ! [6097]
! Rechnungssumme : DM ###
#####.## DM!";z$;sum;z$
7190 PRINT#8,t$;"!-----!----- [4624]
!-----!-----
--!"
7200 PRINT#8:PRINT#8 [1314]
7210 PRINT#8,t$;CHR$(15);"Bankverbindung: [7542]
Stadtsparkasse Emmerich-Rees BLZ 300 123 7
9 Konto Nr.: 123456";CHR$(18)
7220 anz=anz-1:IF anz>0 THEN GOTO 7070 [2718]
7230 kk=kk+1:sumvk=sumvk+sum:WEND [2266]
7240 CLS:CLS#1 [236]
7250 PRINT#1,"Sollen die Rechnungen erneut [5718]
ausgedruckt werden (J/N)? ";
7260 a$=UPPER$(INKEY$):IF a$=" THEN 7260 [1949]
7270 IF a$="J" THEN GOTO 7000 [1674]
7280 CLS#1:PRINT#1,"Koennen die Liefersche [5435]
ine geloeschet werden (J/N)? ";
7290 a$=UPPER$(INKEY$):IF a$=" THEN 7290 [1951]
7300 IF a$="J" THEN GOSUB 7900:GOTO 7325 [2357]
7310 IF a$="N" THEN CLS#1:PRINT#1,"Bei der [6879]
naechsten Rechnungerst. werden diese erne
ut gedruckt!";GOTO 7330
7320 PRINT#1," Fehleingabe Weiter mit Ta [3577]
ste!";CALL &BB06:GOTO 7280
7325 GOSUB 8000:svk=svk+sumvk:GOSUB 8100 [3269]
7330 PRINT#8,CHR$(12):RETURN [1194]
7700 *** Empfaenger fuer Rechnung [2486]
7705 PRINT#8 [343]
7710 PRINT#8,t$;lif$(0,3) [861]
7720 PRINT#8,t$;lif$(0,4) [1023]
7730 PRINT#8,t$;lif$(0,5) [1017]
7750 PRINT#8:PRINT#8:PRINT#8 [1827]
7760 RETURN [555]
7900 *** alte Lieferscheine loeschen [2573]
7910 a$="L*.LIF":|ERA,@a$ [1167]
7930 OPENOUT"SUMM.REC":FOR i=0 TO 3:PRINT# [5470]
9,"0000":NEXT i:CLOSEOUT
7940 |ERA,@ba$ [1132]
7960 RETURN [555]
8000 *** Rechnungssumme lesen [3596]
8020 OPENIN"RECH.SUM" [1285]
8030 INPUT#9,svk [903]
8040 INPUT#9,sne [1013]
8050 CLOSEIN [752]
8060 RETURN [555]
8100 *** Rechnungssumme speichern [3626]
8120 OPENOUT"RECH.SUM" [1315]
8130 PRINT#9,svk [1353]
8140 PRINT#9,sne [1335]
8150 CLOSEOUT [902]
8160 RETURN [555]
8900 *** Lieferschein lesen [2858]
8905 GOSUB 4300:na$=rech$(kk,1) [1794]
8910 CLS#1:PRINT#1,in$;" Der Lieferschein [6534]
";num$;" wird bearbeitet! ";in$;i=0
8920 OPENIN na$ [589]
8930 WHILE NOT EOF:FOR j=0 TO 5:INPUT#9,li [4412]
f$(i,j):NEXT j:i=i+1:WEND
8940 CLOSEIN [752]
8950 RETURN [555]

```

Listing Fakturist

```

29000 *** Fehlerbehandlung [1559]
29010 CLS#1:PRINT#1,"Fehler Nr. ";ERR;" in [2140]
Zeile ";ERL
29020 FOR ii=1 TO 2000:NEXT ii [1314]
29030 RESUME NEXT [703]

2 ***** [1391]
3 *** INVENTUR.BAS V1.0 **** [965]
4 *** WIESOFT 1987 **** [1208]
5 *** Heinz Wienhoven **** [1817]
9 ***** [1391]
20 in$=CHR$(24):bi$=CHR$(7):z$="!" [1287]
30 i=0:wa=0:wa$="":y$="":an=0:en=0:dat$=" [2444]
:na$="":an$="":er=0:k=0:j=0
32 t$=STRING$(9,32) [1587]
40 DIM ar$(200,5) [597]
170 *** Datum einlesen [1681]
180 OPENIN "DATUM.DAT":INPUT#9,dat$:INPUT# [4251]
9,dat1$:CLOSEIN
200 MODE 2 [513]
210 PRINT "Faktu-PRO'87";TAB(24);"INVENTUR [4752]
";TAB(60);"Datum: ";dat$
310 ON ERROR GOTO 29000 [1503]
400 *** Windows [637]
410 WINDOW#0,1,80,2,23 [1198]
420 WINDOW#1,1,80,25,25 [1010]
500 *** Warendatei lesen [1469]
510 GOSUB 9400:GOSUB 9100 [2087]
1000 GOSUB 2000:RUN"AUSWAHL.BAS" [1661]
2000 *** Artikeldatei [881]
2010 RESTORE 2015:GOSUB 15000:GOSUB 9900 [2270]
2015 DATA Inventur,W,Warenwert ermitteln,I
,Inventurliste erstellen,B,Warenbestand ko
rrigieren,H,Hauptmenue,x,x,WIBH
2020 IF wa=4 THEN GOSUB 4500:RETURN [2106]
2030 ON wa GOSUB 3000,2500,6400 [1743]
2040 GOTO 2010 [305]
2500 *** Inventurliste erstellen [1862]
2510 IF ar$(0,1)<>" THEN GOTO 2530 [2200]
2520 GOSUB 9400:GOSUB 9100 [2087]
2530 PRINT#1,"Die Datei ";ar$(0,1);" wird [2726]
ausgedruckt!":i=1
2540 PRINT#8,t$;"Inventurliste : ";ar$(0,1 [5557]
);t$;t$;t$;"Datum: ";dat$:PRINT#8
2548 PRINT#8,"!A-Nr.! Artikelbez. [7291]
! Bestand !A-Nr.! Artikelbez.
Bestand !"
2549 PRINT#8,STRING$(80,"-") [1487]
2550 WHILE ar$(i,1)<>" [1091]
2560 PRINT#8,USING"! \ \ ! \ [11094]
\ ! ##### ! \ \ ! \
\ ! ##### !";z$;ar$(i,0);z$;ar$(i,1);z
$;VAL(ar$(i,5));z$;z$;ar$(i+1,0);z$;ar$(i+
1,1);z$;VAL(ar$(i+1,5));z$
2565 IF i MOD 141=0 THEN PRINT#8,CHR$(12); [1364]
2570 i=i+2:WEND [406]
2575 PRINT#8,CHR$(12); [1101]
2580 CLS#1:RETURN [1242]
3000 *** Warenwert ermitteln [2211]
3010 CLS:CLS#1:PRINT#1,in$;" Berechnung! E [4412]
inen Moment bitte ";in$;
3020 i=1:sumvk=0:sunek=0 [792]
3030 WHILE ar$(i,1)<>" [1091]
3040 sumvk=sumvk+VAL(ar$(i,3))*VAL(ar$(i,5 [2630]
))
3050 sunek=sunek+VAL(ar$(i,4))*VAL(ar$(i,5 [1959]
))
3060 i=i+1:WEND [758]
3070 CLS#1:LOCATE 10,10:PRINT"Der Wert des [5187]
Warenbestandes betraegt:"
3080 LOCATE 15,13:PRINT USING"Einkaufsprei [2975]
s : #####.## DM";sumek
3090 LOCATE 15,14:PRINT USING"Verkaufsprei [3602]
s : #####.## DM";sumvk
3100 LOCATE 10,20:PRINT"Weiter mit TASTE!" [3911]

```

Listing Fakturist

```

:CALL &BB06:RETURN
3900 '*** Datensatz nach Artikelnr. suchen [2230]
3910 i=1:PRINT#1,"Artikelnr.: ";an=46:en= [6987]
57:GOSUB 10000:IF an$="/" THEN GOTO 3940 E
LSE a$=RIGHT$("0000"+an$,4)
3920 WHILE a$<>ar$(i,0) AND i<200:i=i+1:W [3359]
END
3930 IF i>199 THEN PRINT#1,"Artikel nicht [6438]
gespeichert! Weiter mit Taste!":i=0:CALL &
BB06
3940 RETURN [555]
4500 '*** Abfrage Speichern [2246]
4510 PRINT t$;"Sollen die Daten gespeicher [7779]
t werden? Bei Nein werden Sie geloescht!"
4520 wa$="JN":GOSUB 9900 [1186]
4530 IF wa=1 THEN GOSUB 9000 [2041]
4540 GOSUB 9600 [1048]
4550 RETURN [555]
4900 '*** Fehlermeldung bei zu grosser Zah [2773]
l
4905 CLS#1 [373]
4910 PRINT#1,"Die Zahl ist in der Tabelle [4916]
nicht darstellbar!";
4920 FOR kk=1 TO 2000:NEXT kk [1366]
4930 CLS#1:RETURN [1242]
6400 ' *** Warenbestand korrigieren [2584]
6405 CLS:LOCATE 1,10 [846]
6410 GOSUB 3900:nr=i [904]
6420 PRINT USING "\ [5035]
\
alter Bestand: #####";ar$(nr,1);VAL(ar$(
nr,5))
6430 PRINT "neuer Bestand?":GOSUB 10000:IF [6984]
VAL(an$)>9999 OR VAL(an$)<0.01 THEN GOSUB
4900:GOTO 6430 ELSE ar$(nr,5)=an$
6440 PRINT:PRINT "Weiteren Bestand aendern [3151]
?":wa$="JN":GOSUB 9900
6450 IF wa=2 THEN CLS:RETURN [2594]
6460 GOTO 6400 [442]
9000 '*** Datei speichern [907]
9010 GOSUB 11000 [1038]
9020 CLS#1:PRINT#1,in$;"Die Datei ";na$;" [3950]
wird gespeichert!";in$;
9030 OPENOUT na$ [633]
9040 i=0 [420]
9050 WHILE ar$(i,1)<>"AND i<201 [1078]
9060 FOR j=0 TO 5:PRINT#9,ar$(i,j):NEXT j [1120]
9070 i=i+1 [444]
9080 WEND:CLOSEOUT [1782]
9090 CLS#1:CLS:RETURN [712]
9100 '*** Datei lesen [1144]
9120 GOSUB 11000:PRINT#1,"Die Datei ";na$; [5008]
" wird gelesen!";
9125 i=0 [420]
9130 OPENIN na$ [589]
9140 WHILE NOT EOF [1840]
9150 FOR j=0 TO 5:INPUT#9,ar$(i,j):NEXT j [1264]
9160 i=i+1 [444]
9170 WEND:CLOSEIN [1209]
9175 ar$(0,1)=na$ [838]
9180 CLS#1:RETURN [1242]
9200 '*** Dateinamen einlesen [1807]
9210 CLS:GOSUB 11000 [576]
9220 a$="*.DAT":|DIR,@a$ [805]
9230 PRINT>Name der Datei ohne Extension e [5374]
ingeben!"
9240 an=45:en=122:GOSUB 10000 [1666]
9250 IF LEN(an$)>8 THEN GOSUB 9800:GOTO 92 [3464]
40
9260 IF LEN(an$)<8 THEN an$=an$+" " :GOTO 9 [1751]
260
9270 na$=UPPER$(an$)+" .LIF":ar$(0,1)=na$ [1865]
9280 RETURN [555]
9400 '*** Dateinamen vorgeben [1239]
9410 na$="WAREN.DAT" [1103]
9420 ar$(0,1)=na$ [838]
9430 RETURN [555]
9600 '*** Datei loeschen [1117]
9610 CLS#1:PRINT#1,"Einen Moment bitte!";: [2205]
i=0
9620 WHILE ar$(i,1)<>"AND i<201 [1078]
9630 FOR j=0 TO 5:ar$(i,j)="":NEXT j [1746]
9640 i=i+1:WEND [758]
9650 CLS#1:RETURN [1242]
9800 '*** FEHLEINGABE [1735]
9810 PRINT bi$;:PRINT#1,TAB(30);in$;" FEHL [2451]
EINGABE ";in$;
9820 FOR kk=1 TO 1000:NEXT kk:PRINT bi$;:P [3257]
RINT#1

```

Listing Fakturist

```

9830 RETURN [555]
9900 '*** AUSWAHL [946]
9910 CLS#1:PRINT#1,"Auswahl (";wa$;"):";bi [2953]
$;
9920 y$=UPPER$(INKEY$):IF y$=""THEN 9920 [3456]
9930 wa=INSTR(wa$,y$):IF wa=0 THEN GOSUB 9 [4460]
800:GOTO 9910
9940 CLS#1:RETURN [1242]
10000 '*** Abfrage [1291]
10010 PRINT#1,"Eingabe: ";an$="" [1383]
10020 y$=INKEY$:IF y$=""THEN 10020 [2649]
10030 IF ASC(y$)=13 THEN CLS#1:RETURN [1749]
10032 IF ASC(y$)=127 THEN an$=LEFT$(an$,LE [3771]
N(an$)-1):CLS#1:PRINT#1,"Eingabe: ";an$;
10040 IF ASC(y$)<an OR ASC(y$)>en THEN PRI [4028]
NT bi$;:GOTO 10020
10050 PRINT#1,y$;an$=an$+y$:y$="":GOTO 10 [2289]
020
11000 '*** Laufwerkstatus abfragen (aus CP [3846]
C International 01.86)
11005 RETURN [555]
11010 OUT(&FA7E),1:FOR i=1 TO 1000:NEXT i [1828]
11020 OUT(&FB7F),4:OUT(&FB7F),(-PEEK(&A700 [1904]
)+2):
11030 st=INP(&FB7F):OUT(&FA7E),0 [1514]
11040 rd=st AND 32 [781]
11050 wp=st AND 64 [319]
11060 IF rd=0 THEN PRINT#1,"Disk richtig e [5212]
inlegen! Weiter mit Taste":CALL &BB06:GOTO
11000
11070 IF wp=64 THEN PRINT#1,"Schreibschutz [7566]
auf Disk entfernen! Weiter mit Taste":CAL
L &BB06:GOTO 11000
11080 CLS#1:FOR ii=1 TO 1000:NEXT ii:RETUR [1953]
N
15000 '*** Menuegenerator [1386]
15010 CLS:READ a1$ [730]
15020 LOCATE 24,5:PRINT a1$:PRINT [1891]
15030 READ a1$,a2$ [509]
15040 WHILE a1$<>"x" [1728]
15050 PRINT TAB(20);in$;" ";a1$;" ";in$;" [2406]
";a2$
15060 READ a1$,a2$ [509]
15070 WEND [390]
15080 READ wa$:PRINT:PRINT [1707]
15090 RETURN [555]
29000 '*** Fehlerbehandlung [1559]
29010 CLS#1:PRINT#1,"Fehler Nr. ";ERR;" in [2140]
Zeile ";ERL
29020 FOR ii=1 TO 2000:NEXT ii [1314]
29030 RESUME NEXT [703]

```

```

2 ***** [1164]
3 *** SYSTEM .BAS V1.0 *** [919]
4 *** WiesOFT 1987 *** [1133]
5 *** Heinz Wienhoven *** [1757]
9 ***** [1164]
10 MODE 2 [513]
20 in$=CHR$(24):bi$=CHR$(7) [1073]
100 WINDOW#1,1,80,25,25 [1010]
110 OPENIN "DATUM.DAT":INPUT#9,datum$:CLOS [3401]
EIN
120 OPENIN "FARBE.DAT":INPUT#9,farbe0:INPU [4280]
T#9,farbe1:CLOSEIN
140 PRINT "Faktu-PRO'87";TAB(24);"SYSTEM"; [4328]
TAB(60);"Datum: ";datum$
150 WINDOW#0,1,80,2,23 [1198]
500 ON ERROR GOTO 29000 [1503]
999 '----- [3264]

```

```

1000 '*** Hauptmenue [812]
1010 RESTORE 1015:GOSUB 15000:ende=0 [1424]
1015 DATA Hauptmenue,I,Diskinhalt anzeigen [6317]
,L,Files loeschen,U,File umbenennen
1016 DATA B,BAK-Files loeschen,M,Monitor e [3855]
instellen,-,

```

Listing Fakturist

Anwendungen

```

1017 DATA H,Hauptprogramm,x,x,ILUBMH [3284]
1020 ON wa GOSUB 2000,2020,2040,2030,2 [3221]
0500,1100
1030 IF ende=1 THEN GOTO 1500 [725]
1040 GOTO 1000 [339]
1100 ende=1:RETURN [263]
1499 '----- [3300]

1500 '*** Ende des Programms [1352]
1510 CLS:PRINT#1,TAB(20);"Programm AUSWAHL [3042]
wird geladen";
1520 OPENOUT "FARBE.DAT":PRINT#9,farbe0:PR [3273]
INT#9,farbe1:CLOSEOUT
1530 x.$="FARBE.BAK":!ERA,@x.$ [1125]
1540 RUN "AUSWAHL.BAS" [723]
9999 '----- [3336]

10000 '*** STRING-Eingabe [707]
10010 an$="" [187]
10020 y$=INKEY$:IF y$=""THEN 10020 [2649]
10030 IF ASC(y$)=13 THEN CLS#1:RETURN [1749]
10032 IF ASC(y$)=127 THEN an$=LEFT$(an$,LE [3458]
N(an$)-1):CLS#1:PRINT#1,an$;
10040 IF ASC(y$)<an OR ASC(y$)>en THEN PRI [4028]
NT bi$;:GOTO 10020
10050 PRINT#1,y$;:an$=an$+y$:y$="":GOTO 10 [2289]
020
10099 '----- [3264]

10100 '*** Tastenabfrage [506]
10110 PRINT#1,wa$;bi$; [358]
10120 y$=UPPER$(INKEY$):IF y$=""THEN 10120 [3467]
10130 wa=INSTR(wa$,y$):IF wa=0 THEN GOSUB [5159]
10200:GOTO 10110
10140 CLS#1:RETURN [1242]
10199 '----- [3264]

10200 '*** FEHLEINGABE [1735]
10210 PRINT bi$;:PRINT#1,TAB(30);in$;" FEH [2451]
LEINGABE ";in$;
10220 FOR kk=1 TO 1000:NEXT kk:PRINT bi$;: [3257]
PRINT#1
10230 RETURN [555]
10999 '----- [3264]

11000 '*** Laufwerkstatus abfragen (aus CP [3846]
C International 01.86)
11010 OUT(&FA7E),1:FOR i=1 TO 1000:NEXT i [1828]
11020 OUT(&FB7F),4:OUT(&FB7F),(-PEEK(&A700 [1904]
)+2):
11030 st=INP(&FB7F):OUT(&FA7E),0 [1514]
11040 rd=st AND 32 [781]
11050 wp=st AND 64 [319]
11060 IF rd=0 THEN PRINT#1,"Disk richtig e [5212]
inlegen! Weiter mit Taste":CALL &BB06:GOTO
11000
11070 IF wp=64 THEN PRINT#1,"Schreibschutz [7566]
auf Disk entfernen! Weiter mit Taste":CAL
L &BB06:GOTO 11000
11080 CLS#1:FOR ii=1 TO 1000:NEXT ii:RETUR [1953]
N
14999 '----- [3264]

15000 '*** Menuegenerator [1386]
15010 CLS:READ a1$ [730]
15020 LOCATE 24,5:PRINT a1$:PRINT [1891]
15030 READ a1$,a2$ [509]
15040 WHILE a1$<>"x" [1728]
15050 PRINT TAB(20);in$;" ";a1$;" ";in$;" [2406]
";a2$
15060 READ a1$,a2$ [509]
15070 WEND [390]
15080 READ wa$:PRINT:PRINT:GOSUB 10100 [2198]
15090 RETURN [555]
19999 '----- [3318]

20000 '*** Diskinhalt anzeigen [1927]
20010 GOSUB 20100:CLS:inh.$=di.$+"*.*":!DI [1998]
R,@inh.$
20020 PRINT#1,"Weiter mit Taste!":CALL &BB [2083]
06:CLS#1
20030 RETURN [555]
20099 '----- [3241]

20100 '*** Laufwerk einstellen [3396]
20110 RESTORE 20115:GOSUB 15000 [976]
20115 DATA Laufwerkawahl,A,Laufwerk A,B, [4190]
Laufwerk B,-,Ramdisk,x,x,AB
20120 ON wa GOTO 20140,20150,20160 [1119]

```

Listing Fakturist

```

20140 di.$="A":RETURN [983]
20150 di.$="B":RETURN [1557]
20160 di.$="C":RETURN [1922]
20199 '----- [3241]

20200 '*** Files loeschen [1911]
20210 GOSUB 20000:PRINT:PRINT "Files loesc [2662]
hen"
20220 PRINT#1,"Filename: ";:an=32:en=126:G [3960]
OSUB 10000
20230 pu=INSTR(an$,"."):IF pu=0 OR pu>9 TH [3483]
EN GOSUB 10200:GOTO 20220
20240 IF LEN(an$)>pu+3 THEN GOSUB 10200:GO [4288]
TO 20220
20250 PRINT#1,"Wirklich loeschen? ";:wa$=" [4715]
JN":GOSUB 10100
20260 IF wa=2 THEN 20290 [1038]
20270 name$=di.$+an$ [789]
20280 !ERA,@name$ [1132]
20290 RETURN [555]
20299 '----- [3241]

20300 '*** BAK Files loeschen [1493]
20310 GOSUB 20000:PRINT:PRINT "BAK-Files 1 [2847]
oeschen"
20320 an$="*.BAK":GOTO 20250 [592]
20399 '----- [3091]

20400 '*** Files umbenennen [1904]
20410 GOSUB 20000:PRINT:PRINT "Files umben [2526]
ennen"
20420 PRINT#1,"Filename: ";:an=32:en=126:G [3960]
OSUB 10000
20430 pu=INSTR(an$,"."):IF pu=0 OR pu>9 TH [3491]
EN GOSUB 10200:GOTO 20420
20440 IF LEN(an$)>pu+3 THEN GOSUB 10200:GO [4233]
TO 20420
20450 alna$=di.$+an$ [807]
20460 PRINT#1,"neuer Filename: ";:GOSUB 10 [2382]
000
20470 pu=INSTR(an$,"."):IF pu=0 OR pu>9 TH [3451]
EN GOSUB 10200:GOTO 20460
20480 IF LEN(an$)>pu+3 THEN GOSUB 10200:GO [4225]
TO 20460
20490 nena$=di.$+an$:!REN,@nena$,@alna$:RE [912]
TURN
20499 '----- [3318]

20500 '*** Monitor einstellen [1981]
20510 CLS:RESTORE 20515:GOSUB 15000 [1716]
20515 DATA Monitor einstellen,S,Schreibfar [4459]
be,H,Hintergrundfarbe,I,Invers
20516 DATA M,Hauptmenue,x,x,SHIM [1334]
20520 IF wa=4 THEN RETURN [1275]
20530 ON wa GOSUB 21000,21300,21600 [2080]
20540 GOTO 20500 [482]
20999 '----- [3318]

21000 '*** Schreibfarbe aendern [896]
21010 PRINT#1,"Schreibfarbe (0-26): ";:an= [5401]
48:en=57:GOSUB 10000
21020 farbel=VAL(an$):IF farbel<0 OR farbe [4055]
1>26 THEN CLS#1:GOTO 21010
21030 IF farbel=farbe0 THEN GOTO 21010 [1567]
21040 INK 1,farbe1:RETURN [1781]
21299 '----- [3318]

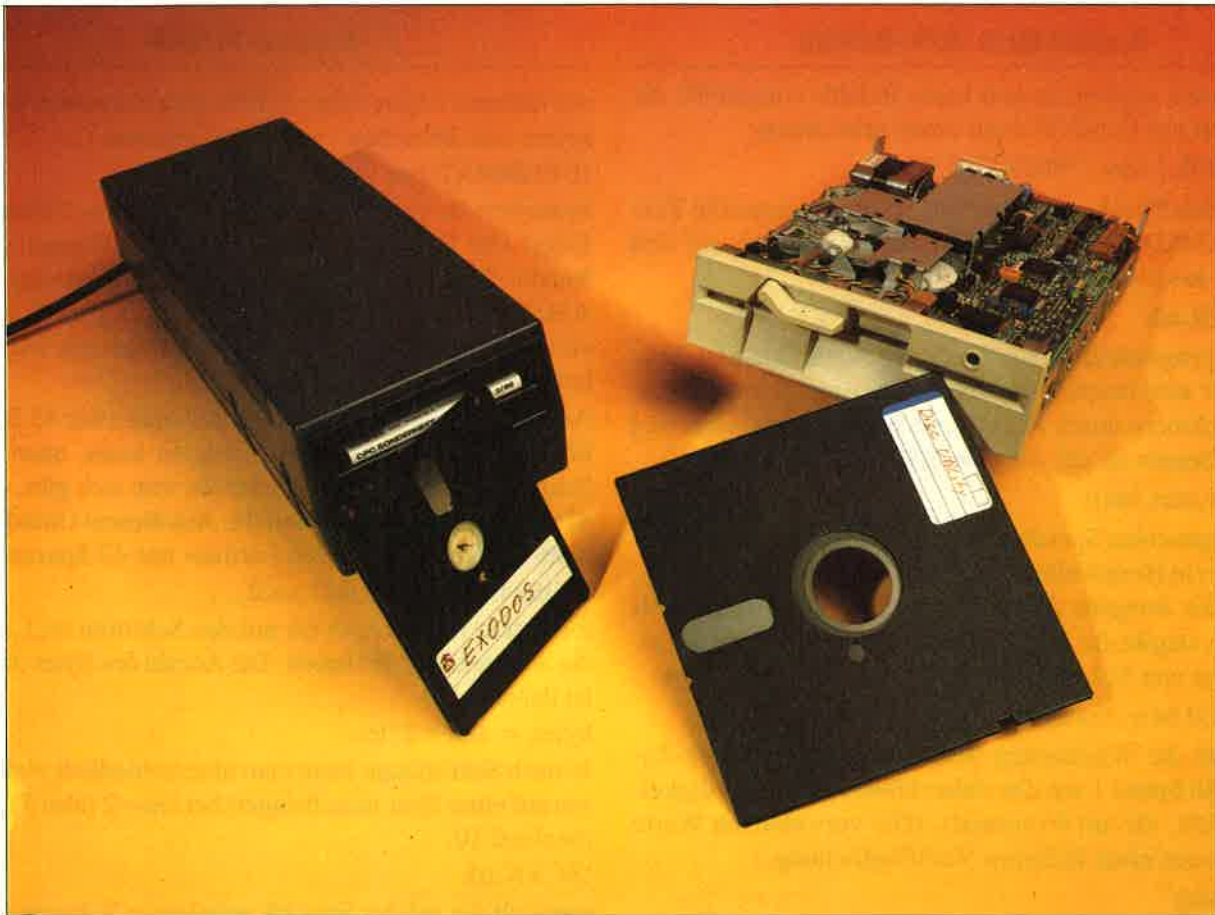
21300 '*** Hintergrundfarbe aendern [2434]
21310 PRINT#1,"Hintergrundfarbe (0-26): "; [4287]
:an=48:en=57:GOSUB 10000
21320 farbe0=VAL(an$):IF farbe0<0 OR farbe [2994]
0>26 THEN CLS#1:GOTO 21310
21330 IF farbe0=farbe1 THEN GOTO 21310 [2302]
21340 INK 0,farbe0:BORDER farbe0 [1436]
21350 RETURN [555]
21599 '----- [3318]

21600 '*** Inversdarstellung [1619]
21610 fa.=farbe1:farbe1=farbe0:farbe0=fa. [2883]
21620 INK 1,farbe1:INK 0,farbe0:BORDER far [3579]
be0
21630 RETURN [555]
28999 '----- [3264]

29000 '*** Fehlerbehandlung [1559]
29010 CLS#1:PRINT#1,"Fehler Nr. ";ERR;" in [2140]
Zeile ";ERL
29020 FOR ii=1 TO 2000:NEXT ii [1314]
29030 RESUME NEXT [703]

```

Listing Fakturist



Disc-Durchblick mit ExoDOS

Dieses Programm ist weniger einer der üblichen Disketten-Monitore, als eine hervorragende RSX-Erweiterung des doch etwas mageren AMSDOS. Daß es besser geht, zeigt ja selbst das CP/M 2.2! Sehen Sie sich einmal die Befehlsliste in Ruhe an; bestimmt kommen Sie dann sofort zum Abtippen des Programms.

Zuerst lassen Sie den BASIC-Lader »EXODOS.LAD« ablaufen, der dann ein File namens »EXODOS.BIN« auf Ihrer Diskette erzeugt. Dann brauchen Sie nur noch den kleinen »EXODOS.BAS«-Starter ablaufen zu lassen, und es kann sofort losgehen.

Was kann ExoDOS?

Die erste erfreuliche Nachricht ist die, daß CPC 464-Besitzer endlich auch z.B. !ERA, "FILE.BAS" eingeben können, um so die bekannte, äußerst umständliche String-Handhabung zu umgehen. (Das »!« soll hier und im ganzen Artikel den RSX-Strich darstellen.) Aber auch sonst ist ExoDOS ganz auf die Praxis abgestimmt:

!LIST(,maske)

ist eine Erweiterung des bekannten Befehls DIR(,maske) und gibt alle Dateien der Diskette aus: Usernummer, Dateiname, Extension und den Status der Datei. Normale Dateien sind R/W (read and write) und DIR (directory files). R/O-(read only)-Files sind nicht löschar; SYS-(system)-Files werden bei DIR und CAT nicht aufgelistet. Der Status einer Datei kann mit STAT (s.u.) gesetzt werden.

!ERASE,maske

arbeitet wie ERA,maske, allerdings fragt der Rechner bei jeder einzelnen Datei vorher um Erlaubnis. Mit ERASE, "*" können Sie also recht komfortabel mehrere Files hintereinander löschen.

!RENAME,maskel,maske2

ist mit REN,maskel,maske2 zu vergleichen, wobei aber auch Wildcards zugelassen sind! Dadurch können Sie z.B. unter RE-NAME, "*.INC", "*.PAS" mit einem Befehl alle »PAS-Dateien« in »INC-Dateien« umbenennen!

!USER,nr

ist nur insofern neu, als daß hier alle Nummern von 0 bis 255 zulässig sind.

!USER,"filename",nr

ändert die Usernummer einer Datei, ohne daß diese erst in den Rechner geladen werden muß.

Gelöschte Dateien werden immer mit der Usernummer 229 belegt, so daß Sie mit

!USER,229:!USER,"filename",0

eine versehentlich gelöschte Datei sehr schnell wieder »zum Leben erwecken« können.

!STAT,"filename",par

legt den bei LIST erwähnten Status der Datei fest. Der Parameter par bedeutet folgendes: par = 0 : normal, par = 1 : read only, par = 2 : system file, par = 3 : read only system file.

!PARAMS,"filename"

schließlich gibt die Parameter, die im Fileheader abgelegt sind, aus: Dateiname mit Extension, Dateityp, Start, Länge und Autostart.

Komfortable RSX-Befehle

Als nächstes werden ein paar kurze Befehle vorgestellt, die die Arbeit mit ExoDOS noch etwas erleichtern:

!PRINTER,1 bzw. **!PRINTER,0**

schaltet den Drucker ein bzw. aus. Damit können alle Texte aus AMSDOS und ExoDOS zusätzlich auch auf den Drucker ausgegeben werden.

!BUFFER,adr

legt den programminternen Buffer ab Adresse adr an. Dieser Buffer wird hauptsächlich beim Lesen und Schreiben einzelner Sektoren durch READ und WRITE verwendet und liegt zu Beginn ab der Adresse &9000.

!DUMP(,adr(,len))

gibt den gesamten Speicherinhalt ab der Adresse adr auf der Länge len in Hexadezimal- und ASCII-Form aus. Fehlt len, so muß die Ausgabe durch ESC angehalten werden, fehlt auch adr, so gibt der Rechner den Inhalt des Buffers auf einer Länge von 512 Bytes (= normale Sektorgröße) aus.

!SPEED,0 bzw. **!SPEED,1**

beeinflusst die Wartezeiten während eines Disketten-Zugriffs. Mit Speed 1 werden daher höhere Geschwindigkeiten erreicht, als mit 0 (normal). (Die verwendeten Werte stammen aus einer früheren Veröffentlichung.)

!TRY(,anz)

zeigt dem Rechner an, wieviele Versuche er maximal unternehmen soll, Daten von Diskette zu lesen. Ohne Parameter wird der Standardwert 16 angenommen. Bei schwer zu lesenden Disketten sollte man die Zahl etwas erhöhen.

Klassische Monitorbefehle

Auch die »klassischen« Befehle des Disketten-Monitors dürfen natürlich nicht fehlen:

!SEEK,trk

bringt den Lesekopf des Laufwerks auf die Spur trk.

!READ,trk,sec

liest den Sektor sec aus der Spur trk in den Buffer ein.

!WRITE,trk,sec

schreibt den Inhalt des Buffers in den Sektor sec der Spur trk. Bei READ bzw. WRITE ist auch noch eine andere Eingabe möglich:

!READ,blk bzw. **!WRITE,blk**

In dieser Form wird der jeweils 1. Sektor des Blocks blk gelesen bzw. geschrieben. Ein Block enthält 1024 Bytes und ist der kleinste Bereich, der unter AMSDOS noch getrennt verwaltet werden kann. Im Directory sind die Blöcke angegeben, die eine bestimmte Datei belegt. Block 0 ist immer der 1. Sektor des Directory, unabhängig vom Format.

!BLOCK,blk

rechnet die Blocknummer blk in Spur und Sektor um, was bei READ,blk und WRITE,blk automatisch durchgeführt wird.

Format-Befehle

Als nächstes folgen einige Befehle zum Formatieren von Disketten und Betreiben von Fremdformaten:

!DFORMAT,par

formatiert die gesamte Diskette und hat keine Sicherheitsabfrage! Der Parameter par bestimmt das Format: par=0 : Vendor-Format (CP/M-Format ohne Systemspuren, par=1: AMSDOS-Format (»Data-Only-Format«), par=2: ergibt ein »Extended Format«, das 42 Spuren mit jeweils 10 Sektoren besitzt.

An dieser Stelle sei angemerkt, daß man zwar 43 Spuren (0 bis 42) auf der Diskette formatieren kann, ohne daß der Schritt-Motor unschöne Geräusche von sich gibt, die Spur 42 aber die Spur 41 überschreibt. Aus diesem Grunde hat das oben erwähnte »Extended Format« nur 42 Spuren.

!FORMAT,trk,len,sec1,sec2,...

formatiert nur die Spur trk mit den Sektoren sec1,sec2,..., die alle die Länge len haben. Die Anzahl der Bytes pro Sektor ist dabei

$\text{bytes} = 128 * 2 \uparrow \text{len}$

Je nach Sektorlänge kann man unterschiedlich viele Sektoren auf einer Spur unterbringen, bei len=2 (also 512 Bytes) maximal 10.

!SCAN,trk

ermittelt die auf der Spur trk angelegten Sektoren und ihre Längen. Damit können Sie z.B. Fremdformate auf Disketten identifizieren.

Fremdformate unter AMSDOS

Entgegen allen früheren Berichten lassen sich bestimmte Fremdformate (so z.B. das o.g. »Extended Format«) unter AMSDOS ohne Einschränkungen betreiben. Das beinhaltet auch das Laden und Abspeichern von Dateien. Dazu müssen lediglich einige Parameter übergeben werden:

!LOGIN,trk,sec,len,off,sys

vereinbart ein solches Format und schaltet den Auto-Login ab (s.u.). Die Anzahl der Spuren (40) wird mit trk bezeichnet, sec ist die Anzahl der Sektoren pro Spur (9), len die Länge der Sektoren (2), off die Nummer des ersten Sektors einer Spur (&41) und sys die Anzahl der Systemspuren (2). (Die eingeklammerten Zahlen geben als Beispiel die Werte für das CP/M-Format an.)

!EXTEND

übergibt die Werte 42, 10, 2, &C1, 0 an den obigen Befehl und vereinbart so das Extended Format.

!LOGIN

ohne Parameter, ermittelt das Format der Diskette über AMSDOS und legt dieses ab. Der Auto-Login wird aber nicht eingeschaltet.

!AUTOLOG,0 bzw. **!AUTOLOG,1**

dient dazu, den Auto-Login ab- bzw. einzuschalten. Bei eingeschaltetem Auto-Login wird vor jedem Zugriff auf Diskette (bei CAT,LOAD,etc.) das Disketten-Format bestimmt.

Wenn Sie ein Fremdformat verwenden, dann müssen Sie diesen abschalten, damit Ihre Format-Vereinbarung erhalten bleibt.

Zum Schluß noch zwei recht schnelle Kopierbefehle, die Sie aber nur benutzen sollten, wenn Sie einen CPC 6128 oder eine Speichererweiterung besitzen, da diese Routinen auf die zweite RAM-Bank zurückgreifen:

```
!COPY(,trk1(,trk2))
```

kopiert die Spuren trk1 bis trk2 von einer Diskette zur anderen. Zu Beginn legen Sie die Quelldiskette ein und folgen dann

den Anweisungen des Programms. Geben Sie nur trk1 an, so wird auch nur diese kopiert, ohne Parameter wird Spur 0 bis 39 kopiert.

```
!FILECOPY, "filename"
```

kopiert eine Datei unabhängig vom Typ und vom Diskettenformat. Dieser Befehl wird genauso wie der letzte gehandhabt. Das war's, und wie Sie sehen, haben wir Ihnen nicht zuviel versprochen. Wahrscheinlich wird auch bei Ihnen ExoDOS bald zu den Standard-Tools gehören.

(T. Kochmann/JB)

Wie funktioniert ExoDOS?

Dieser kurze Artikel ist für alle diejenigen geschrieben, die sich beim Disassemblieren von ExoDOS an manchen Stellen verständlicherweise die Zähne ausgebissen haben, weil etliche Routinen zwischendurch geheimnisvolle Abstecker ins Floppy-ROM machen; aber natürlich auch für die Leser, die sich anderweitig für das »Interieur« von ExoDOS interessieren.

Insbesondere die Befehle, die das Directory der Diskette bearbeiten, strotzen geradezu von Routinen, mit denen der Nichteingeweihte absolut nichts anzufangen weiß. Am Beispiel des Befehls !USER, "filename", usernr sollen einige der Routinen erklärt werden:

Der für uns interessante Einsprung befindet sich an der Adresse &83E6. An dieser Stelle ist das Floppy-ROM bereits selektiert und IX zeigt auf den letzten Wert, der mit dem RSX-Befehl übergeben wurde, hier also die neue Usernummer.

In dem angesprochenen Unterprogramm ab &841E wird zunächst IX zweimal inkrementiert, so daß es jetzt auf den eingegebenen Dateinamen zeigt. Der nachfolgende CALL &CD73 rettet den Stackpointer und ist wichtig, falls während der weiteren Befehle eine Fehlermeldung auftritt. Weiter geht es mit einem CALL &CDC7, der die Stringparameter des Filenamens holt, in HL (Start) und B (Länge) ablegt, und IX um 2 erhöht. Nun folgt der erste interessantere CALL: Die Routine ab &DA8D legt den eingegebenen Filenamens in der Form ab, in der er auch im Directory erscheint: Der erste Teil wird auf 8 Buchstaben mit Blanks aufgefüllt, der zweite Teil (Extension) auf 3. Es gibt eine ganze Reihe weiterer solcher Routinen, z.B. CALL &DA5B und CALL &DAA6, die sich alle nur dadurch unterscheiden, ob sie Wildcards akzeptieren, welche Extents sie zulassen, usw. BC enthält immer das erste Byte dieses »normierten« Filenamens, welches die Drivenummer darstellt, danach folgen der User- und ab dem 3. Byte der eigentliche Filename. Der nachfolgende CALL &CE14 nimmt nun endlich das Laufwerk in Betrieb und führt ggf. einen Login durch. BC wird um 1 erhöht.

Der CALL &D683 löscht die gesamte Belegungs- (Allocation-) Tabelle und trägt die Blocks, die das Directory belegt, ein. In dieser Tabelle kann das AMSDOS jederzeit ablesen, wo es auf der Diskette noch freie Blocks gibt, um Daten abzulegen. Sie wird erst bei den folgenden CALLs weiter aufgefüllt.

Der letzte CALL des Unterprogramms führt nach &D698 und macht endlich Ernst: Der erste Directory-Eintrag, der zu dem eingegebenen Filenamens paßt, wird gesucht. Beim Aussprung zeigt DE auf das erste Byte dieses Eintrags; also die Usernummer, HL enthält die Nummer des Eintrags und BC zeigt (immer noch) auf den »normierten« Filenamens. Für den Fall, daß nichts gefunden wurde, wird das Carry gelöscht, was hier mit einem JP &D50C (Error 6, »File not found«) quittiert wird. Zurück im Hauptprogramm finden wir als nächsten interessan-

ten Befehl die Kombination aus »LD A,(IX-4)« und »LD (DE),A«. Hier wird die gewünschte Usernummer aus der Eingabe geholt und in den Directory-Eintrag gesetzt. Um die restlichen 31 Bytes des Eintrags zu ändern, müßten Sie DE entsprechend erhöhen. Die beiden Register HL und BC sollten Sie bei derartigen Manövern aber immer unberührt lassen!

Nun steht der geänderte Directory-Eintrag zwar im RAM, nicht aber auf Diskette. Also muß dieser mit CALL &D97A erst wieder zurückgeschrieben werden.

Der anschließende CALL &D698 sollte Ihnen bekannt vorkommen. Er sucht den nächsten passenden Eintrag, das ist hier der nächste Extent. Falls noch einer gefunden wurde, wird das Programm bei &83EA fortgesetzt.

Mit diesen Informationen haben Sie schon ein solides Grundwissen, um auch die anderen RSX-Befehle dieser Art zu verstehen, nämlich RENAME, ERASE, LIST und STAT.

Nun zu einem anderen interessanten Teil von ExoDOS:

Wie kann ich AMSDOS beibringen, von Disketten mit meinem eigenen Format sogar Dateien zu lesen? — Auf diese Frage kann man am besten mit einer Tabelle antworten (Tabelle 1). Sie enthält alle Adressen des AMSDOS-RAM-Bereichs, die Sie ändern können, bzw. müssen, um ein anderes Format zu vereinbaren. Wichtig ist, in die Adresse &A8A8 einen Wert ungleich 0 zu setzen, damit durch das nächste »CAT« o.ä. Ihr vereinbartes Format nicht wieder zerstört wird.

Zum Schluß sei noch kurz erwähnt, was CALL &D392 und CALL &D3AF bei »FILECOPY« bewirken: Mit ihnen kann man den jeweils nächsten Record aus dem aktuellen File lesen, bzw. beschreiben, und zwar unabhängig vom Filetyp! HL zeigt auf den Beginn des Buffers; das Carry-Flag zeigt auf übliche Weise Erfolg oder Mißerfolg der Routine an.

Tabelle 1:

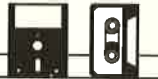
Wichtige Speicheradressen für die Verwendung alternativer Disketten-Formate unter AMSDOS:

&A890/1	Records pro Track (36) (= Records pro Sektor *Sektoren pro Track)
&A895/6	max. Blocknummer (170) (= (Anz. der Tracks -Anzahl Systemspuren) *Records pro Track / 8 -Anz. Directory-Blocks +1)
&A897/8	max. Anzahl der Directory- Einträge - 1 (63)
&A899/A	Bitmasken für die belegten Directory-Blocks (&C0,&00) (1 Block = &80,&00; 2 Blocks = &C0,&00; etc.)
&A89B/C	max. Anzahl der Blocks pro Extent (16)
&A89D/E	Zahl der Systemspuren (2)
&A89F	Sektoroffset (&41)
&A8A0	Sektoren pro Track (9)
&A8A1	Gap3 bei Read/Write (42) (kleinerer Wert möglich)
&A8A2	Gap3 bei Formatieren (82) (kleinerer Wert möglich)
	&A8A4 Sektorgröße (2) (Bytes pro Sektor = = 128*2↑Sektorgröße)
&A8A5	Records pro Sektor (4)
&A8A8	Flag für Auto-Login (0) (0: ein, > <0: aus)

(In Klammern sind die Standard- Werte für CP/M-Format angegeben.)

(T. Kochmann/JB)

für 464-664-6128



```

10 '***** BASIC-Lader fuer ExoDOS 2.1 *****
20 '
30 z=100:FOR i=&8000 TO &89C9 STEP 16:s=0
40 FOR j=0 TO 15:READ x:POKE i+j,x:s=s+x:NEXT
50 READ cs:IF s=cs THEN z=z+10:NEXT:GOTO 70
60 PRINT"Fehler in Zeile";z:END
70 PRINT"Allles O.K.!"
80 SAVE"exodos",b,&8000,&9CA:END
90 '
100 DATA 1,62,128,33,58,128,205,209,188,62,195,50,
48,0,33,253,1653
110 DATA 128,34,49,0,33,90,187,17,30,129,1,3,0,237
,176,33,1147
120 DATA 25,129,50,90,187,34,91,187,58,4,172,254,2
01,192,62,195,1931
130 DATA 33,62,129,50,4,172,34,5,172,201,0,0,0,0,1
,33,128,1123
140 DATA 247,75,129,247,85,129,247,96,129,247,218,
129,247,252,129,247,2853
150 DATA 9,130,247,22,130,247,31,130,247,66,130,24
7,116,130,247,218,2347
160 DATA 130,247,4,131,247,212,131,247,247,131,247
,58,132,247,121,132,2664
170 DATA 247,213,132,247,226,132,247,248,132,247,1
25,133,247,69,134,247,3026
180 DATA 142,134,247,136,135,80,82,73,78,84,69,210
,66,85,70,70,1761
190 DATA 69,210,68,85,77,208,83,80,69,69,196,84,82
,217,83,69,1749
200 DATA 69,203,82,69,65,196,87,82,73,84,197,66,76
,79,67,203,1698
210 DATA 76,73,83,212,69,82,65,83,197,82,69,78,65,
77,197,85,1593
220 DATA 83,69,210,83,84,65,212,80,65,82,65,77,211
,83,67,65,1601
230 DATA 206,65,85,84,79,76,79,199,69,88,84,69,78,
196,76,79,1612
240 DATA 71,73,206,68,70,79,82,77,65,212,70,79,82,
77,65,212,1588
250 DATA 67,79,80,217,70,73,76,69,67,79,80,217,0,2
43,217,33,1667
260 DATA 250,255,57,34,23,129,225,94,35,86,237,83,
20,129,217,251,2125
270 DATA 223,20,129,201,0,0,7,0,0,245,205,33,129,2
41,0,0,1433
280 DATA 0,197,237,75,84,129,13,193,248,205,43,189
,56,7,71,205,1952
290 DATA 189,136,120,24,244,205,189,136,205,46,189
,56,248,201,254,3,2445
300 DATA 192,123,254,13,192,225,42,194,176,55,201,
61,194,237,136,123,2418
310 DATA 50,84,129,201,0,61,194,237,136,237,83,94,
129,201,0,144,1980
320 DATA 61,1,0,0,40,23,75,66,221,94,2,221,86,3,61
,40,994
330 DATA 12,237,91,94,129,1,0,2,60,60,194,237,136,
205,233,202,1893
340 DATA 11,203,56,203,25,203,56,203,25,203,56,203
,25,203,56,203,1934
350 DATA 25,3,235,197,76,205,166,136,77,205,166,13
6,205,196,219,229,2476
360 DATA 6,16,78,35,205,166,136,62,32,205,90,187,1
6,244,62,32,1572
370 DATA 205,90,187,205,90,187,225,6,16,126,14,46,
254,32,56,4,1743
380 DATA 254,127,56,1,121,205,90,187,35,16,238,205
,233,202,193,11,2174
390 DATA 120,177,202,233,202,205,189,136,24,185,61
,194,237,136,33,234,2568
400 DATA 129,29,32,3,33,243,129,195,13,198,50,0,25
0,0,175,15,1494
410 DATA 12,1,3,35,0,200,0,1,1,10,0,3,61,40,6,60,4
33
420 DATA 30,16,194,237,136,123,195,3,198,61,194,23
7,136,83,30,0,1873
430 DATA 205,99,199,195,231,136,205,40,130,205,102
,198,195,231,136,205,2712
440 DATA 40,130,205,78,198,195,231,136,61,40,16,61
,225,194,237,136,2183
450 DATA 229,75,30,0,221,86,2,42,94,129,201,205,88
,130,42,94,1668

```

Listing ExoDOS

```

460 DATA 129,201,61,194,237,136,205,88,130,90,22,0
,197,205,57,203,2155
470 DATA 193,205,196,219,121,195,160,136,213,14,0,
205,251,196,209,203,2716
480 DATA 35,203,18,203,35,203,18,203,35,203,18,205
,13,218,205,128,1943
490 DATA 200,195,162,200,205,115,205,183,6,0,40,6,
205,194,205,205,2326
500 DATA 199,205,205,166,218,205,20,206,197,14,0,6
2,21,205,235,202,2360
510 DATA 193,205,233,202,205,233,202,205,131,214,2
05,162,214,210,196,219,3229
520 DATA 26,2,205,216,215,48,243,229,197,213,26,95
,22,0,205,57,1999
530 DATA 203,209,205,196,219,205,200,219,205,196,2
19,33,9,0,25,126,2469
540 DATA 230,128,7,205,242,136,35,126,230,128,7,20
3,207,205,242,136,2467
550 DATA 193,225,205,233,202,205,189,136,24,192,20
5,139,131,237,91,94,2701
560 DATA 129,26,254,9,40,76,19,62,10,205,211,136,2
54,89,40,2,1562
570 DATA 62,78,205,90,187,27,213,66,75,204,150,212
,209,33,16,0,1827
580 DATA 25,235,24,221,61,205,139,131,205,199,205,
205,141,218,33,16,2263
590 DATA 0,9,80,89,68,77,42,94,129,26,190,194,175,
205,126,254,1758
600 DATA 9,40,15,213,197,229,205,56,131,225,17,16,
0,25,193,209,1780
610 DATA 24,236,205,233,202,195,233,202,229,213,19
7,62,16,245,26,254,2772
620 DATA 63,32,1,126,2,35,19,3,241,61,32,241,193,2
09,225,35,1518
630 DATA 3,235,62,11,205,242,136,235,80,89,62,12,2
05,242,136,11,1966
640 DATA 205,20,206,229,205,131,214,205,152,214,22
5,62,5,218,235,202,2728
650 DATA 197,68,77,205,131,214,205,152,214,210,12,
213,205,217,217,221,2758
660 DATA 225,62,10,218,235,202,221,229,195,247,212
,42,94,129,34,210,2565
670 DATA 131,225,205,115,205,229,205,194,205,205,1
99,205,205,166,218,205,3117
680 DATA 20,206,205,131,214,205,152,214,210,12,213
,229,197,42,210,131,2591
690 DATA 10,119,35,235,1,12,0,237,176,62,255,18,19
,19,18,19,1235
700 DATA 237,83,210,131,235,193,225,205,152,214,56
,223,42,210,131,54,2601
710 DATA 9,201,0,0,254,2,40,14,61,194,237,136,205,
115,205,205,1878
720 DATA 207,205,253,117,1,201,205,30,132,183,221,
126,252,18,205,122,2478
730 DATA 217,205,152,214,56,244,201,254,2,194,237,
136,205,30,132,229,2708
740 DATA 197,33,9,0,25,221,86,252,6,2,126,23,203,2
6,31,119,1359
750 DATA 35,16,247,193,225,205,122,217,205,152,214
,56,226,201,221,35,2570
760 DATA 221,35,225,205,115,205,229,205,199,205,20
5,141,218,205,20,206,2839
770 DATA 205,131,214,205,152,214,210,12,213,201,61
,194,237,136,205,199,2789
780 DATA 205,205,119,188,208,205,233,202,235,205,2
00,219,205,196,219,33,3077
790 DATA 18,0,25,126,205,160,136,205,196,219,17,3,
0,25,205,151,1691
800 DATA 136,25,205,196,219,205,151,136,205,196,21
9,43,25,205,151,136,2453
810 DATA 205,233,202,205,233,202,195,122,188,61,19
4,237,136,83,30,0,2526
820 DATA 205,99,199,210,231,136,42,94,129,54,255,8
4,93,19,229,213,2292
830 DATA 30,0,205,108,197,225,237,91,81,190,115,12
3,35,114,35,235,2021
840 DATA 225,210,231,136,190,40,8,48,229,98,107,43
,43,24,223,235,2090
850 DATA 237,82,205,233,202,69,203,56,5,26,19,205,
160,136,62,45,1945
860 DATA 205,90,187,26,19,198,48,205,90,187,205,19
6,219,16,234,205,2330
870 DATA 233,202,195,233,202,61,194,237,136,253,42
,66,190,29,253,115,2641
880 DATA 24,201,183,194,237,136,221,33,238,132,62,
5,24,21,0,0,1711

```

Listing ExoDOS

```

890 DATA 193,0,2,0,10,0,42,0,183,32,8,30,0,205,108
,197,1010
900 DATA 195,231,136,254,5,194,237,136,33,67,202,2
37,91,66,190,1,2275
910 DATA 22,0,213,253,225,237,176,253,54,24,255,25
3,54,17,14,253,2303
920 DATA 54,18,28,221,126,0,253,119,13,221,126,2,2
53,119,15,221,1789
930 DATA 126,4,253,119,20,71,62,128,4,7,16,253,253
,119,21,221,1677
940 DATA 70,6,253,112,16,111,38,0,205,109,133,253,
117,0,221,126,1770
950 DATA 8,253,78,13,145,71,205,109,133,203,60,203
,29,203,60,203,1976
960 DATA 29,203,60,203,29,43,253,117,5,253,116,6,2
01,120,183,32,1853
970 DATA 4,33,0,0,201,5,200,84,93,25,16,253,201,61
,194,237,1607
980 DATA 136,17,0,0,205,99,199,210,231,136,33,235,
133,1,39,9,1683
990 DATA 22,82,221,53,0,250,162,133,33,20,134,40,5
,1,41,10,1207
1000 DATA 22,28,253,42,66,190,253,112,16,253,114,1
8,205,233,202,6,2013
1010 DATA 0,62,5,205,242,136,88,22,0,229,197,205,5
7,203,193,225,2069
1020 DATA 17,4,0,229,229,62,10,112,25,61,32,251,80
,30,0,126,1268
1030 DATA 225,197,79,205,82,198,193,225,210,231,13
6,62,13,205,90,187,2538
1040 DATA 4,13,242,177,133,205,233,202,195,233,202
,0,0,65,2,0,1906
1050 DATA 0,70,2,0,0,66,2,0,0,71,2,0,0,67,2,0,282
1060 DATA 0,72,2,0,0,68,2,0,0,73,2,0,0,69,2,0,290
1070 DATA 0,74,2,64,0,0,193,2,0,0,198,2,0,0,194,2,
731
1080 DATA 0,0,199,2,0,0,195,2,0,0,200,2,0,0,196,2,
798
1090 DATA 0,0,201,2,0,0,197,2,0,0,202,2,0,0,198,2,
806
1100 DATA 0,0,203,2,192,254,3,218,237,136,221,33,2
48,191,221,86,2245
1110 DATA 4,221,78,2,214,2,71,33,2,9,237,66,30,28,
32,2,1031
1120 DATA 30,82,253,42,66,190,253,112,16,253,115,1
8,253,113,20,42,1858
1130 DATA 94,129,229,114,35,54,0,221,126,0,221,43,
221,43,35,119,1684
1140 DATA 35,113,35,16,238,225,30,0,205,82,198,195
,231,136,33,102,1874
1150 DATA 198,17,78,198,6,1,214,3,210,237,136,34,9
2,135,237,83,1879
1160 DATA 95,135,33,128,190,112,221,86,2,221,94,0,
60,40,7,83,1507
1170 DATA 60,40,3,17,39,0,221,42,94,129,221,115,0,
30,0,213,1224
1180 DATA 205,99,199,14,0,205,251,196,209,253,42,6
6,190,253,78,15,2275
1190 DATA 58,128,190,183,196,233,202,58,128,190,18
3,62,6,196,242,136,2391
1200 DATA 213,197,33,0,0,205,52,135,56,5,193,209,1
95,231,136,40,1900
1210 DATA 5,205,119,135,48,239,58,128,190,183,40,1
0,62,9,205,211,1847
1220 DATA 136,62,7,205,242,136,193,209,33,0,0,205,
61,135,210,231,2065
1230 DATA 136,40,17,205,119,135,48,243,58,128,190,
183,196,233,202,196,2329
1240 DATA 233,202,55,201,205,119,135,58,128,190,18
3,40,170,62,8,205,2194
1250 DATA 211,136,24,163,229,205,97,135,205,91,135
,24,7,229,205,97,2193
1260 DATA 135,205,94,135,197,1,192,127,237,73,193,
225,208,213,253,86,2574
1270 DATA 21,203,58,203,27,25,209,124,183,55,201,1
95,102,198,195,78,2077
1280 DATA 198,197,1,0,127,151,203,36,23,203,36,23,
55,203,28,203,1687
1290 DATA 60,198,196,237,121,193,201,12,253,126,15
,71,253,134,16,185,2271
1300 DATA 192,72,20,221,126,0,146,201,61,194,237,1
36,50,128,190,42,2016
1310 DATA 66,190,1,25,0,9,237,91,94,129,1,119,0,23
7,176,205,1580
1320 DATA 199,205,221,43,221,43,205,119,188,208,25
3,42,125,190,253,54,2569

```

```

1330 DATA 41,0,205,233,202,62,6,205,242,136,253,54
,44,255,33,0,1971
1340 DATA 0,17,128,0,1,0,2,221,229,221,33,146,211,
205,122,136,1672
1350 DATA 221,225,143,50,129,190,33,0,2,237,66,32,
6,205,125,188,1852
1360 DATA 195,146,188,68,77,62,9,205,211,136,62,7,
205,242,136,205,2154
1370 DATA 93,136,253,54,8,255,58,128,190,183,32,34
,60,50,128,190,1852
1380 DATA 197,205,199,205,205,140,188,48,19,237,91
,125,190,33,159,0,2241
1390 DATA 25,235,1,85,0,9,1,69,0,237,176,55,193,20
8,197,205,1696
1400 DATA 131,214,193,33,0,0,17,128,0,221,33,175,2
11,205,122,136,1819
1410 DATA 208,205,93,136,253,54,44,255,58,129,190,
183,40,8,62,8,1926
1420 DATA 205,211,136,195,181,135,205,233,202,205,
233,202,253,54,8,0,2658
1430 DATA 253,54,44,0,205,122,188,205,93,136,195,1
43,188,229,213,197,2465
1440 DATA 42,66,190,1,25,0,9,237,91,94,129,6,119,7
8,26,119,1232
1450 DATA 121,18,35,19,16,247,193,209,225,201,197,
229,205,97,135,205,2352
1460 DATA 148,136,1,1,192,127,237,73,225,193,62,0,20
8,11,25,120,177,1935
1470 DATA 32,232,55,201,221,229,201,35,126,205,160
,136,43,78,24,6,1984
1480 DATA 79,62,38,205,90,187,121,15,15,15,15,205,
175,136,121,230,1709
1490 DATA 15,198,48,254,58,218,90,187,198,7,195,90
,187,205,9,187,2146
1500 DATA 208,254,252,192,151,50,84,129,205,125,18
8,205,146,188,237,123,2737
1510 DATA 23,129,201,205,242,136,205,129,187,205,2
4,187,205,132,187,205,2602
1520 DATA 166,202,254,252,192,24,221,216,62,4,195,
235,202,62,4,195,2486
1530 DATA 242,136,229,197,245,33,251,136,195,243,2
02,82,47,87,32,32,2389
1540 DATA 32,255,82,47,45,32,32,32,255,68,73,82,32
,32,32,255,1386
1550 DATA 83,89,83,32,32,32,255,80,97,114,97,109,1
01,116,101,114,1535
1560 DATA 32,101,114,114,111,114,255,70,111,114,10
9,97,116,116,105,110,1789
1570 DATA 103,32,116,114,97,99,107,255,13,18,82,10
1,97,100,105,110,1549
1580 DATA 103,32,102,114,111,109,32,83,79,85,82,67
,69,32,100,105,1305
1590 DATA 115,99,255,13,18,87,114,105,116,105,110,
103,32,111,110,32,1525
1600 DATA 68,69,83,84,73,78,65,84,73,79,78,32,100,
105,115,99,1285
1610 DATA 255,13,73,110,115,101,114,116,32,83,79,8
5,82,67,69,32,1426
1620 DATA 100,105,115,99,58,32,18,7,255,13,73,110,
115,101,114,116,1431
1630 DATA 32,68,69,83,84,73,78,65,84,73,79,78,32,1
00,105,115,1218
1640 DATA 99,58,32,18,7,255,128,68,101,108,101,116
,101,32,139,32,1395
1650 DATA 40,89,47,78,41,63,32,255,128,82,101,110,
97,109,105,110,1487
1660 DATA 103,32,139,255,32,116,111,32,139,255,0,0
,0,0,0,1214

```



```

10 ***** ExoDOS 2.1 *****
20
30 MEMORY &7FFF
40 LOAD"exodos.bin",&8000
50 CALL &8000
60 MODE 2
70 PRINT" ExoDOS 2.1":PRINT
80 NEW

```

Listing ExoDOS

Der Disketten-Ordner

Im Laufe der Zeit sammeln sich eine ganze Menge von Programmen und Datensätzen auf einer größeren Anzahl von Disketten an. Wer hat nicht schon einmal verzweifelt nach einem bestimmten Programm auf seinen Disketten gefahndet. Das hier vorgestellte Disketten-Ordner-Programm soll helfen, das Chaos zu begrenzen.

Bis zu 2000 Einträge werden verwaltet. Die 3“, 5,25“-Formate von Schneider sowie das 5,25“-Format von Vortex mit 720 kB werden gelesen. Die alphabetische Sortierung erfolgt schnell durch eine Maschinenroutine.

Das Programm läuft unter AMS-DOS, VDOS 2.0 und VDOS 2.1. Die Speichererweiterung von Vortex ist mit eingebunden, aber nicht notwendig. Das neue EPROM, BOS 2.0, von Vortex für die Speichererweiterung muß jedoch mit DISBOS abgeschaltet werden, da der Disketten-Ordner sonst abstürzt.

Eingabe des Programms

Nach Eingabe des Hauptprogramms und Abspeicherung unter dem Namen CATLIST.BAS muß noch die Maschinenroutine CATSORT.BIN generiert werden.

In Bild 1 ist ein Hexdump der Maschinenroutine aufgelistet. Wer kein komfortables Programm zur Eingabe von Hexlisten hat, sollte den Lader in Listing 2 verwenden. Der Lader enthält eine zeilenweise Überprüfung der DATA- Zeilen.

Falls nur unter AMS-DOS oder nur unter VDOS gearbeitet wird, können die Zeilen 130 und 140 geändert werden. In Zeile 130 wird die Variable »dos« auf den entsprechenden Wert gesetzt (1 für AMS-DOS und 2 für VDOS).

z.B.: 130 dos=1

Die Zeile 140 entfällt.

Bei weiteren Änderungen bzw. Erweiterungen sollte beachtet werden, daß nur noch wenige Bytes bis zu dem Datenfeld frei sind.

Anwendung des Programms

Nach Start des Programms meldet sich das Hauptmenue mit den folgenden Möglichkeiten:

- 1 Neueingabe der Disketten
- 2 Aktualisierung der Diskettenliste
- 3 Einladen der Liste
- 4 Abspeicherung der Liste
- 5 Ausdruck der Liste
- 6 Anzeige der Liste
- 7 Liste der Diskettenbezeichnungen
- 8 Laufwerk/User

Neueingabe der Disketten

Bis zu 255 Disketten(seiten) sind in einer Liste archivierbar; soweit die obere Grenze von 2000 Programmnamen-Einträgen nicht überschritten wird. In den 2000 Einträgen sind auch die Diskettennamen mitgezählt. Jede Diskette kann mit einem 9 Zeichen langen Namen versehen werden.

Um die Eingabe abzurechnen, wird statt einer Diskettenbezeichnung das Wort »Ende« eingegeben. Erweiterungen sind später jederzeit möglich.

Aktualisierung der Diskettenliste

Nach Wahl des Menüepunkts 2 wird ein Untermenue aufgelistet.

- 1 Disketteninhalt ändern
- 2 Diskettenliste erweitern
- 3 Disketteneintrag löschen

4 Liste neu sortieren

5 Anzeige der Diskettenbezeichnungen

0 Rücksprung ins Hauptmenue

Bei Änderungen bzw. Löschungen behalten die Disketten ihren Platz in der Liste der Diskettenamen, so daß die Platznummer nicht verloren geht, und die restlichen Disketten keine Verschiebung der Benummerung erfahren.

Zum Sortieren muß Punkt 4 verwendet werden. Vom Programm wird die Sortieroutine nicht selbst aufgerufen.

Über Menüpunkt 5 kann man eine Liste aller Diskettenbezeichnungen anzeigen lassen. Ein Ausdruck ist nur vom Hauptmenue aus möglich. Außer den Bezeichnungen werden noch die Nummer, die Anzahl der Dateien, der freie Platz und die User-Nummer der einzelnen Disketten(seiten) aufgeführt. Ein automatischer Seitenstopp ist vorhanden.

Der freie Platz springt in 4k-Intervallen (Vortex-Format). Falls nur unter AMS-DOS gearbeitet wird, kann das Programm auf 1k-Intervalle geändert werden. Hierzu muß in Zeile 740 die Division durch 4 und in Zeile 1900 die Multiplikation mit 4 entfallen.

Einladen/Abspeichern der Liste

Die gesamte Liste wird als Binärfile gespeichert und eingeladen. Hierdurch werden selbst lange Datenfelder in wenigen Sekunden von der Diskette gelesen. Außerdem entfällt auf diese Weise die lästige Garbage Collection.

Grundsätzlich beträgt die Länge der Daten 26 Kilobyte. Beim Abspeichern wird zuerst der alte Datensatz gelöscht. Hierdurch kann es selbst bei randvollen Datendisketten keinen »disk full« Error geben.

Bis zu zehn verschiedene Datensätze (Bezeichnung: 0 bis 9) werden vom Disketten-Ordner verwaltet, so daß man seine Programme in Gebiete einsortieren kann; z.B. CPM-, Dienstprogramme, Sprachen, Text, Spiele u.s.w.

Ich empfehle für den Disketten-Ordner mit seinen Datenfiles eine extra Diskette anzulegen.

Ausdruck/Anzeige der Liste

Der Ausdruck erfolgt dreispaltig in ganz kleiner Schrift (Indexmodus/Schmalschrift), so daß auf einer DIN-A4 Seite 405 Programmnamen Platz finden. Alphabetisch sortiert findet so die komplette Programmsammlung auf wenigen Seiten eine übersichtliche Darstellung. Die jeweils letzte Zahl in den drei Spalten gibt die Diskettennummer an, auf der das Programm gespeichert ist.

Die Druckersteuer codes in Zeile 980 sind für einen Epson

Hexdump von CATSORT.BIN

```

-----
12500 30D4    2A 00 01 22 02 01 AF 32    305
12508 30DC    08 01 2A 04 01 22 06 01    97
12516 30E4    2A 06 01 54 5D 01 0D 00    240
12524 30EC    09 22 06 01 01 0B 00 1A    88
12532 30F4    BE 28 04 30 20 18 05 23    378
12540 30FC    13 0D 20 F3 2A 02 01 2B    395
12548 3104    22 02 01 7C FE 00 20 D8    663
12556 310C    7D FE 00 20 D3 3A 08 01    689
12564 3114    FE 00 20 BC C9 3E 01 32    788
12572 311C    08 01 01 0D 00 2A 06 01    72
12580 3124    54 5D ED 42 46 1A 77 78    815
12588 312C    12 23 13 0D 20 F6 18 CC    591
-----

```

5121

X-80 geschrieben. Eine Angleichung an andere Drucker dürfte, aufgrund der wenigen Steuercodes, keine Schwierigkeiten bereiten.

Die Anzeige der Programmliste wird zweispaltig auf den Bildschirm geschrieben. Zum Suchen kann man vorher den Anfangsbuchstaben des gewünschten Programms vorgeben. Dies ist jedoch nur sinnvoll, wenn die Liste alphabetisch sortiert ist. Soll die Programmliste von Anfang an gezeigt werden, nur auf ENTER drücken. Da die Liste Mehrfachabspeicherungen auf verschiedenen Disketten jetzt übersichtlich aufzeigt, werden sicherlich so einige Kilobyte frei werden.

Nun wünsche ich viel Spaß beim Sortieren.

(Werner Renziehausen)

FÜR 464-664-6128



```

10 ' Sortierte CAT-Liste aller Disketten [2617]
   (10611 Bytes)
20 ' [117]
30 ' (C) Werner Renziehausen [2805]
   Juli 1987
40 ' [117]
50 ' Max. 2000 Eintraege (Files und Disket [3709]
   tenbezeichnungen)
60 ' Max. 256 Disketten [1877]
70 ' Lauffaehig unter VDOS 2.0, VDOS 2.1 u [2080]
   nd AmsDOS
80 ' [117]
90 x=42619-HIMEM:IF x<256 THEN POKE &FF,x: [3302]
   'Speichererw.-Anpassung!
100 CLEAR:OPENOUT"Dummy":MEMORY 11499:CLOS [4989]
   EOUT:DEFINT b-z:adr=12500:adr=38487
110 LOAD"catsort.bin",12500:|A:la$="A" [3961]
120 FOR i=12500 TO 12500+&5F:POKE &110+i-1 [2922]
   2500,PEEK(i):NEXT
130 MODE 1:LOCATE 8,13:INPUT"AmsDOS/VDOS [3813]
   1/2 - ",dos
140 IF dos<>1 AND dos<>2 THEN 130 [1410]
150 MODE 1:CALL &BC02:PEN 1:PAPER 0:BORDER [4001]
   1:INK 3,18:INK 1,26:INK 0,0:f=1
160 PRINT STRING$(40,154)" "CHR$(164)" Wer [5323]
   ner Renziehausen Juli 1987
170 PRINT STRING$(40,154) [1464]
180 LOCATE 12,6:PRINT CHR$(24)SPC(18):LOCA [3835]
   TE 12,7:PRINT" DISKETTEN-ORDNER "
190 LOCATE 12,8:PRINT SPC(18)CHR$(24) [1269]
200 PRINT" 1 Neueingabe der Disketten [2405]
210 PRINT" 2 Aktualisierung der Diskette [5012]
   nliste
220 PRINT" 3 Einladen der Liste [2056]
230 PRINT" 4 Abspeicherung der Liste [3301]
240 PRINT" 5 Ausdruck der Liste [2365]
250 PRINT" 6 Anzeige der Liste [1654]
260 PRINT" 7 Liste der Diskettenbezeichn [3750]
   ungen
270 PRINT" 8 Laufwerk/User [2520]
280 z$=INKEY$: IF z$="" THEN 280 [694]
290 z=ASC(z$)-48 [948]
300 IF z<1 OR z>8 THEN 280 [1075]
310 ON z GOSUB 360,1640,1530,1270,950,940, [2769]
   2030,2740
320 GOTO 150 [415]
330 ' [117]
340 ' Neueingabe der Disketten [986]
350 ' [117]
360 CLS [91]
370 WINDOW 1,40,2,24:WINDOW#1,1,40,25,25:W [2250]
   INDOW#2,1,40,1,1
380 PRINT#2,CHR$(24)" NEUEINGABE - Lauf [4046]
   werk: "la$" User:";:PRINT#2,USING"##";us
   r;:PRINT#2," "CHR$(24);
390 IF zz<>0 THEN GOSUB 2300:IF a$<>"J" TH [4186]
   EN RETURN
400 PRINT#1,CHR$(24)SPC(5)"Abbruch der Ein [5844]
   gabe mit "CHR$(34)"Ende"CHR$(34)SPC(5)CHR$
   (24);
410 zz=0:disc=0 [1986]

```

Listing Catlist

```

420 'Schleife Einlesen [766]
430 z=0 [343]
440 PRINT"Bezeichnung der eingelegten":PR [7566]
   INT USING"####";disc+1;:PRINT". Diskette (
   max. 9 Z.): .....";
450 LOCATE POS(#0)-9,VPOS(#0):CALL &BB03:I [7389]
   NPUT",file$:IF LEN(file$)>9 THEN 440 ELSE
   file$=file$+SPACE$(9-LEN(file$))
460 IF UPPER$(file$)="ENDE " THEN MODE [1716]
   1:RETURN
470 disc=disc+1 [1253]
480 GOSUB 490:IF adrz>=adrz THEN RETURN [1534]
   ELSE 780
490 PEN 3:CAT:PEN 1:bloecke=0:adrz=adr+zz [4302]
   *13:adrz=adr-disc*13+13
500 FOR j=&9E7C-PEEK(&FF) TO &A67A-PEEK(& [2845]
   FF) STEP 14
510 IF PEEK(j+1)<32 THEN 590 [1415]
520 z=z+1 [702]
530 FOR k=1 TO 12:i=j+k [1622]
540 POKE adrz+k,PEEK(i) [939]
550 NEXT k:POKE adrz+13,disc:adrz=adrz+1 [2546]
   3
560 IF adrz>=adrz THEN CLS:LOCATE 10,10 [7145]
   :PRINT"Speicher voll!":SOUND 1,100,100
   :FOR i=1 TO 3000:NEXT:disc=disc-1:RETURN
570 bloecke=bloecke+PEEK(j+12) [1899]
580 NEXT j [370]
590 i=0:FOR j=&9E7C-PEEK(&FF) TO &A67A-PE [4730]
   EK(&FF) STEP 14:POKE j+1,0
600 i=i+1:IF i=128 THEN j=&A67A-PEEK(&FF [1996]
   )
610 NEXT [350]
620 FOR j=1 TO 9:POKE adrz+j,ASC(MID$(fi [2801]
   le$,j,1)):NEXT
630 POKE adrz+11,z:POKE adrz+13,disc:PO [3607]
   KE adrz+10,usr
640 fr=0:f=0:IF dos=1 THEN maxblocks=PEEK [6304]
   (&A895+(PEEK(&A700)*64))-1:fr=maxblocks-bl
   oecke:GOTO 740
650 ON ERROR GOTO 790 [1583]
660 FOR ff=23 TO 1 STEP -1 [1755]
670 |GCHAR,6,ff,@f:IF f<>32 THEN fp=ff:f [2584]
   f=1
680 NEXT [350]
690 IF f=105 THEN ps=22:' VDOS 2.1 [690]
700 IF f=98 THEN ps=29:' VDOS 2.0 [1447]
710 IF f=102 THEN ps=3:' AmsDOS [1944]
720 |GCHAR,ps,fp,@f:fr=f-48:|GCHAR,ps-1,f [4812]
   p,@f:fr=fr+VAL(CHR$(f))*10
730 |GCHAR,ps-2,fp,@f:fr=fr+VAL(CHR$(f))* [3174]
   100
740 ON ERROR GOTO 0:POKE adrz+12,fr/4 [2661]
750 zz=zz+z [1067]
760 RETURN [555]
770 'Ende - Schleife Einlesen [1237]
780 GOTO 430 [462]
790 INPUT"frei (Wert ablesen) = ",fr [1693]
800 IF fr<0 OR fr>720 THEN 790 ELSE RESUME [2166]
   740
810 ' [117]
820 ' Unterprogramm - Sortieren [1863]
830 ' [117]
840 CLS:PRINT zz;" Files sind gespeichert" [1376]
850 PRINT" CPC sortiert Programme - bitte [2345]
   warten
860 PRINT" Zeitbedarf je nach Vorsortierun [3314]
   g
870 PRINT" bis zu ca."ROUND((zz/1500)^2*40 [2197]
   0,0)"Sekunden
880 POKE &101,INT((zz-1)/256):POKE &100,(( [2568]
   zz-1) MOD 256)
890 POKE &105,INT((adr+1)/256):POKE &104,( [4001]
   (adr+1) MOD 256)
900 CALL &110:RETURN [991]
910 ' [117]
920 ' Ausgaberroutinen der sortierten File [5119]
   s - Bildschirm und Drucker
930 ' [117]
940 aus=0:CLS:IF zz=0 THEN GOSUB 1530:GOTO [3178]
   990 ELSE GOTO 990
950 aus=8:IF zz=0 THEN GOSUB 1530 [2240]
960 MODE 1:PRINT" Datum: ";:INPUT datum$ [4551]
   :IF datum$="" THEN 960
970 PRINT" Drucker einschalten - Taste dru [3465]
   ecken":CALL &BB06
980 PRINT#aus,CHR$(27)CHR$(64)SPC(10)"Prog [7631]

```

Listing Catlist

```

rammliste vom ";datum$;CHR$(15)CHR$(27)"3"
CHR$(16)CHR$(27)"S1"CHR$(27)"1"CHR$(5):PRI
NT#8
990 IF aus=0 THEN MODE 1:LOCATE 5,10:PRINT [9374]
"Liste ab Anfangsbuchstabe = ";:ELSE jj=0:
GOTO 1040
1000 a$=INKEY$:IF a$="" THEN 1000 ELSE a=A [3162]
SC(UPPER$(a$)):PRINT CHR$(a)
1010 FOR j=0 TO zz-1: IF PEEK(adr+13*j+1) > [3677]
=a THEN jj=j:j=zz
1020 NEXT [350]
1030 IF aus=0 THEN MODE 2:WINDOW 1,80,1,23 [10260]
:WINDOW#1,1,80,25,25:PRINT#1,SPC(14)"Abbru
ch mit "CHR$(24)" A "CHR$(24)" weit
er "CHR$(24)" W "CHR$(24)" zurueck
"CHR$(24)" Z "CHR$(24);
1040 lbrnt=PEEK(&B08B):hbprnt=PEEK(&B08C) [3306]
:adrz=adr
1050 FOR j1=jj+1 TO zz [1542]
1060 FOR j2=0 TO aus*1/8+1 [1112]
1070 IF aus=8 THEN j=j1+j2*135 ELSE j=j [1881]
1+j2*23
1080 adrz=adr+13*(j-1):IF j>zz THEN 114 [1427]
0
1090 PRINT#aus,SPACE$(10);:PRINT#aus,US [4063]
ING"####";j;:PRINT#aus," ";
1100 FOR k=1 TO 11:IF k=9 THEN PRINT#au [2488]
s,".";
1110 PRINT#aus,CHR$(PEEK(adrz+k)AND 12 [2112]
7);
1120 NEXT:PRINT#aus,SPC(2) [1742]
1130 PRINT#aus,USING"###";PEEK(adrz+12); [6420]
:PRINT#aus,"K";SPACE$(3);:PRINT#aus,
USING"###";PEEK(adrz+13);
1140 NEXT:PRINT#aus:IF j1 MOD 135 = 0 AN [5649]
D aus=8 THEN PRINT#aus,CHR$(12);:j1=j1
+270
1150 IF aus=0 AND (j-jj) MOD 46=0 THEN 1 [1827]
180
1160 IF aus=0 AND j1>=zz THEN i=(zz-jj)M [2462]
OD 23:j1=j1+23-i:GOTO 1180
1170 NEXT:IF aus=8 THEN RETURN ELSE jj=0: [3040]
GOTO 1050
1180 CALL &BB03 [396]
1190 a$=INKEY$:IF a$="" THEN 1190 [1443]
1200 IF UPPER$(a$)="A" THEN RETURN [1375]
1210 IF UPPER$(a$)="W" THEN j1=j1+23:CLS:G [1720]
OTO 1170
1220 IF UPPER$(a$)="Z" THEN j1=j1-69:CLS:I [5711]
F j1<1 THEN j1=0:jj=0:GOTO 1170 ELSE 1170
1230 GOTO 1180 [399]
1240 ' [117]
1250 ' Programmliste auf Diskette speicher [2541]
n
1260 ' [117]
1270 MODE 1:IF zz=0 THEN LOCATE 9,13:PRINT [7532]
"Keine Daten gespeichert!";SOUND 1,100,100
:FOR i=1 TO 3000:NEXT:RETURN
1280 WINDOW 1,40,2,24:WINDOW#1,1,40,25,25: [2250]
WINDOW#2,1,40,1,1
1290 PRINT#2,CHR$(24)" SPEICHERN - Lau [6229]
fwerk: "la$" User:";:PRINT#2,USING"###";u
sr;:PRINT#2," "CHR$(24);:GOSUB 1300:GOTO
1420
1300 LOCATE 9,11:PRINT"Datendiskette einle [2865]
gen
1310 PRINT" Taste druecken [1671]
1320 CALL &BB06:CLS:INK 2,0:PEN 2:ON ERROR [5228]
GOTO 1330:|DIR,"catlst-?.dat":GOTO 1340
1330 CAT:RESUME 1340 [578]
1340 ON ERROR GOTO 0 [1381]
1350 y=VPOS(#0):LOCATE 1,2:PRINT SPACE$(40 [6226]
):LOCATE 1,y-2:PRINT SPACE$(40)
1360 INK 2,18:PEN 1:PRINT" Liste (0...9) - [2641]
Nr.:";
1370 PRINT#1,CHR$(24)SPC(12)"Abbruch mit [4390]
"CHR$(34)"A"CHR$(34)SPC(12)CHR$(24);
1380 a$=INKEY$:IF a$="" THEN 1380 [1405]
1390 IF UPPER$(a$)="A" THEN RETURN [1375]
1400 i=ASC(a$):IF i<48 OR i>57 THEN 1380 [1630]
1410 PRINT a$:b$="catlst-"+a$+".dat":RETUR [3217]
N
1420 IF UPPER$(a$)="A" THEN RETURN [1375]
1430 POKE 38510,disc-1 [371]
1440 POKE 38508,INT(zz/256) [290]
1450 POKE 38509,(zz MOD 256) [1235]
1460 PEN 0:ON ERROR GOTO 1470:|ERA,b$:GOTO [2373]

```

Listing Catlist

```

1480
1470 |ERA,@b$:RESUME 1480 [1756]
1480 ON ERROR GOTO 0:SAVE b$,b,12501,26010 [2786]
1490 RETURN [555]
1500 ' [117]
1510 ' Programmliste von Diskette einlesen [2716]
1520 ' [117]
1530 neudisc=0 [1221]
1540 MODE 1:WINDOW 1,40,2,24:WINDOW#1,1,40 [4184]
,25,25:WINDOW#2,1,40,1,1
1550 PRINT#2,CHR$(24)" LADEN - Lau [7373]
fwerk: "la$" User:";:PRINT#2,USING"###";u
sr;:PRINT#2," "CHR$(24);
1560 IF zz<>0 THEN GOSUB 2300:IF a$<>"J" T [4186]
HEN RETURN
1570 CLS:GOSUB 1300:IF UPPER$(a$)="A" THEN [2505]
RETURN ELSE LOAD b$
1580 disc=PEEK(38510)+1 [783]
1590 zz=PEEK(38508)*256+PEEK(38509) [2326]
1600 RETURN [555]
1610 ' [117]
1620 ' Aktualisieren der Liste - Hauptteil [3126]
1630 ' [117]
1640 IF zz=0 THEN GOSUB 1530 [1980]
1650 MODE 1:CLS:PRINT CHR$(24)" * Aktuali [5373]
sierung der Diskettenliste * "CHR$(24)
1660 PRINT" 1 Disketteninhalt aendern [2681]
1670 PRINT" 2 Diskettenliste erweitern [3830]
1680 PRINT" 3 Disketteneintrag loeschen [1919]
1690 PRINT" 4 Liste neu sortieren [2893]
1700 PRINT" 5 Anzeige der Diskettenbezei [3862]
chnungen
1710 PRINT" 0 Ruecksprung ins Hauptmenue [3909]
1720 a$=INKEY$:IF a$="" THEN 1720 [1528]
1730 a=ASC(a$)-48:IF a=0 THEN RETURN ELSE [2734]
IF a<1 OR a>5 THEN 1720
1740 ON a GOSUB 2550,1970,2380,1760,1790 [1647]
1750 GOTO 1640 [307]
1760 IF zz=0 THEN CLS:PEN 3:PRINT" Liste n [9317]
icht geladen";PEN 1:SOUND 1,100,100:FOR i=
1 TO 3000:NEXT:RETURN
1770 GOSUB 820: RETURN [1457]
1780 ' [117]
1790 ' Diskettenliste anzeigen [847]
1800 ' [117]
1810 aus=0: menue=2 [163]
1820 IF zz=0 THEN GOSUB 1530 [1980]
1830 CLS:az=adrd:PRINT#aus, [1550]
1840 IF aus=8 THEN PRINT#8,SPACE$(10); [2023]
1850 PRINT#aus,SPC(10)"** Diskettenliste * [3796]
**":PRINT#aus
1860 PRINT#aus:IF aus=8 THEN PRINT#8,SPAC [2855]
E$(10);
1870 PRINT#aus," Nr Datei frei User [4939]
Diskette":PRINT#aus
1880 FOR i=1 TO disc [695]
1890 IF aus=8 THEN PRINT#8,SPACE$(10); [2023]
1900 PRINT#aus,USING"#####";PEEK(az+13); [4235]
PEEK(az+11);4*PEEK(az+12);
1910 PRINT#aus,"K "":PRINT#aus,USING"## [5278]
";PEEK(az+10);:PRINT#aus," ";
1920 FOR j=1 TO 9:PRINT#aus,CHR$(PEEK(az+ [3316]
j));:NEXT:PRINT#aus:az=az-13
1930 IF aus=0 AND i MOD 20=0 THEN CALL &B [2369]
B03:CALL &BB06
1940 NEXT:IF aus=0 THEN PRINT" Weiter [4348]
- Taste druecken":CALL &BB06
1950 RETURN [555]
1960 ' [117]
1970 ' Erweiterung der Diskettenliste [2823]
1980 ' [117]
1990 MODE 1:WINDOW 1,40,2,24:WINDOW#1,1,40 [4184]
,25,25:WINDOW#2,1,40,1,1
2000 PRINT#2,CHR$(24)" ERWEITERUNG - Lau [5167]
fwerk: "la$" User:";:PRINT#2,USING"###";u
sr;:PRINT#2," "CHR$(24);
2010 PRINT#1,CHR$(24)SPC(5)"Abbruch der Ei [5844]
ngabe mit "CHR$(34)"Ende"CHR$(34)SPC(5)CHR
$(24);
2020 GOTO 430 [462]
2030 ' [117]
2040 ' Auswahl Bildschirm oder Drucker [1762]
2050 ' [117]
2060 CLS:PRINT" Ausgabe auf Bildschir [3259]
m/Drucker
2070 LOCATE 17,13:PRINT CHR$(24)" B / D "C [2417]
HR$(24)

```

Listing Catlist

```

2080 a$=UPPER$(INKEY$):IF a$="D" OR a$="B" [3661]
    THEN 2090 ELSE 2080
2090 IF a$="B" THEN aus=0 ELSE aus=8 [1896]
2100 GOSUB 1820: RETURN [1499]
2110 ' [117]
2120 'Loeschen einer Disketteneintragung m [4298]
    it den gespeicherten Files
2130 ' Uebergabe der Diskette in der Varia [3624]
    blen "zd"
2140 ' [117]
2150 POKE &9670,1:POKE &9673,17:POKE &9676 [4850]
    ,33:POKE &9679,&ED:POKE &967A,&B0:POKE &96
    7B,&C9
2160 z=0:'at1=TIME [766]
2170 FOR i=1 TO zz [583]
2180 IF i>zz THEN 2240 [1121]
2190 IF PEEK(adr+i*13)<>zd THEN 2240 [1199]
2200 a=(zz-i+1)*13:POKE &9671,a MOD 256: [3086]
    POKE &9672,INT(a/256)
2210 a=adr+i*13-12:POKE &9674,a MOD 256: [4844]
    POKE &9675,INT(a/256)
2220 a=a+13:POKE &9677,a MOD 256:POKE &9 [3076]
    678,INT(a/256)
2230 CALL &9670:z=z+1:zz=zz-1:GOTO 2180 [2404]
2240 NEXT [350]
2250 FOR a=38501-zd*13 TO 38512-zd*13 [1741]
2260 POKE a,0 [143]
2270 NEXT [350]
2280 RETURN [555]
2290 ' [117]
2300 'Neueigabe? [728]
2310 ' [117]
2320 PRINT [361]
2330 PRINT"Achtung! [1252]
2340 PRINT"Vorhandene Liste wird geloescht [3133]
    !
2350 PRINT"Wirklich neu einlesen [2225]
2360 PRINT" J / - [482]
2370 a$=UPPER$(INKEY$):IF a$="" THEN 2370 [3128]
    ELSE RETURN
2380 ' [117]
2390 'Disketteninhalt loeschen [1830]
2400 ' [117]
2410 CLS:PRINT"* DISKETTENINHALT LOESCHEN [2005]
    *
2420 PRINT:PRINT"(Ruecksprung mit Eingabe [4089]
    0)
2430 PRINT:PRINT"Es sind";disc;"Diskette(n [3600]
    ) gespeichert
2440 INPUT"Welche loeschen? Nr.: ";zd [2883]
2450 IF zd=0 THEN RETURN [1022]
2460 IF zd>disc THEN 2440 ELSE GOSUB 2470: [3179]
    GOTO 2520
2470 PRINT:PRINT"Diskette: "; [1822]
2480 az=adr-d-zd*13+13 [774]
2490 FOR j=1 TO 10:PRINT CHR$(PEEK(az+j)); [3027]
    :NEXT
2500 PRINT:PRINT"Diese Diskette loeschen/a [5046]
    endern ?
2510 PRINT" J / -":RETURN [353]
2520 a$=INKEY$:IF a$="" THEN 2520 [1512]
2530 IF UPPER$(a$)="J" THEN PRINT:PRINT"Bi [4373]
    tte warten ...":GOSUB 2110:RETURN
2540 GOTO 2430 [349]
2550 ' [117]
2560 'Disketteninhalt aendern [1142]
2570 ' [117]
2580 MODE 1:WINDOW 1,40,2,24:WINDOW#1,1,40 [4184]
    ,25,25:WINDOW#2,1,40,1,1
2590 PRINT#2,CHR$(24)" AENDERUNG - Lau [6307]
    fwerk: "la$" User:":PRINT#2,USING"##";u
    sr:PRINT#2," "CHR$(24);
2600 PRINT#1,CHR$(24)SPC(12)"Abbruch mit [3950]
    "CHR$(34)"0"CHR$(34)SPC(12)CHR$(24);
2610 PRINT:PRINT"Es sind";disc;"Diskette(n [3600]
    ) gespeichert
2620 INPUT"Welche aendern? Nr.: ";zd [2574]
2630 IF zd=0 THEN RETURN [1022]
2640 IF zd>disc THEN 2620 ELSE GOSUB 2470 [1956]
2650 a$=INKEY$:IF a$="" THEN 2650 [1558]
2660 IF UPPER$(a$)<>"J" THEN 2620 [1797]
2670 CALL &BB03:PRINT:PRINT"Diskettenbezei [6261]
    chung: .....";
2680 LOCATE POS(#0)-9,VPOS(#0):INPUT"";fil [4154]
    e$:IF LEN(file$)>9 THEN 2670 ELSE file$=fi
    le$+SPACE$(9-LEN(file$))
2690 PRINT:PRINT"Bitte warten ...":GOSUB 2 [2225]

```

```

110
2700 z=0:zdd=disc:disc=zd:GOSUB 490:disc=z [2838]
dd:RETURN
2710 ' [117]
2720 'Laufwerk/User [1729]
2730 ' [117]
2740 CLS:PRINT"Laufwerk (A/B): "; [1700]
2750 la$=UPPER$(INKEY$):IF la$="A" THEN !A [2675]
    :GOTO 2770
2760 IF la$="B" THEN !B:GOTO 2770 ELSE 275 [1497]
    0
2770 PRINT la$:PRINT:INPUT"User (0...15): [2845]
    ",i
2780 IF i<0 OR i>15 THEN 2770 [1581]
2790 ON ERROR GOTO 2810:|SELECT,STR$(i) [1944]
2800 usr=i:FOR i=1 TO 2000:NEXT:ON ERROR G [2566]
    OTO 0:RETURN
2810 !USER,i:RESUME 2800 [1191]

```

```

////////////////////
100 'Lader fuer CATSORT.BIN - (C) W.Renz [3261]
    iehausen 07.87
110 ' [117]
120 MEMORY 12499:d=250 [752]
130 FOR i=0 TO 11 [508]
140 x=0:FOR j=0 TO 7 [1063]
150 READ a$:b=VAL("&"+a$):x=x+b:POKE 125 [1601]
    00+i*8+j,b
160 NEXT:y=y+x [1219]
170 READ a$:s=VAL(a$) [1183]
180 IF s<>x THEN z=i*10+d:PRINT:PRINT"Feh [4649]
    ler in Data-Zeile"z:PRINT:END
190 NEXT [350]
200 READ a$:s=VAL(a$) [1183]
210 IF s<>y THEN PRINT:PRINT"Fehler in der [7166]
    Pruefsumme":PRINT:END
220 PRINT:PRINT"CATSORT.BIN wird gespeiche [4988]
    rt":PRINT
230 SAVE"catsort.bin",b,12500,&60 [1679]
240 ' [117]
250 DATA 2a,00,01,22,02,01,af,32, 305 [1344]
260 DATA 08,01,2a,04,01,22,06,01, 97 [1462]
270 DATA 2a,06,01,54,5d,01,0d,00, 240 [1812]
280 DATA 09,22,06,01,01,0b,00,1a, 88 [1108]
290 DATA be,28,04,30,20,18,05,23, 378 [1583]
300 DATA 13,0d,20,f3,2a,02,01,2b, 395 [1293]
310 DATA 22,02,01,7c,fe,00,20,d8, 663 [1969]
320 DATA 7d,fe,00,20,d3,3a,08,01, 689 [1221]
330 DATA fe,00,20,bc,c9,3e,01,32, 788 [1147]
340 DATA 08,01,01,0d,00,2a,06,01, 72 [1304]
350 DATA 54,5d,ed,42,46,1a,77,78, 815 [2213]
360 DATA 12,23,13,0d,20,f6,18,cc, 591 [1941]
370 DATA 5121 [287]

```

```

////////////////////
100 ' Hexdump von CATSORT.BIN - (C) Wern [3505]
    er Renziehausen 07.87
110 ' [117]
120 MODE 2:OPENOUT"Dummy":MEMORY 12499:CLO [3767]
    SEOUT:DEFINT b-z
130 LOAD"catsort.bin",12500:a=0 [3509]
140 z=0:y=0:PRINT#a:PRINT#a," Hexdump von [5521]
    CATSORT.BIN
150 PRINT#a," -----":PRI [2080]

```

```

NT#a
160 FOR i=12500 TO 12500+&5F [1126]
165 IF (i-12500) MOD 8 = 0 AND i<>12500 T [3203]
HEN PRINT#a, " ";:PRINT#a,USING"###";z-y:
y=z
170 IF (i-12500) MOD 8 = 0 THEN PRINT#a,i [2146]
;HEX$(i) " ";

```

Listing Catlist

```

180 x=PEEK(i):z=z+x:PRINT#a, "HEX$(x,2); [3274]
200 NEXT:PRINT#a, " ";:PRINT#a,USING"###" [2779]
;z-y
210 PRINT#a,SPC(38)"-----":PRINT#a,SPC(38 [2642]
);z:PRINT#a
220 INPUT"Ausdruck j/n ";a$:a=0 [1958]
230 IF UPPER$(a$)="J" THEN a=8:GOTO 140 [1422]

```

Listing Catlist

Discmanager

Ein Disketten-Utilitie mit den wichtigsten Hilfen bei dem Umgang mit Disketten. Kurz und hilfreich und in kompakter Form.

ANLEITUNG ZU DISCMANAGER

Der Discmanager nimmt Ihnen fast alle Arbeit beim Bearbeiten von Disketten ab. Z.B. brauchen Sie nicht immer CP/M zu booten, um zu formatieren oder zu kopieren. Sie können weiterhin das Format einer Diskette auslesen oder sogar den Fileheader auslesen, was ja normalerweise nur auf Kassette möglich ist.

Stichpunkte der Möglichkeiten

1. Diskette formatieren
CPM - Data - IBM
2. Diskette kopieren
Von A-A, A-B, B-A, B-B
3. Files löschen
4. Files umbenennen

5. Programme laden

6. Auslesen der Fileheader

Erkennen Sie wo das File beginnt und aufhört. Wichtig beim Übertragen von Maschinencode von »Hand«

7. Format ermitteln

8. Laden von CP/M

ACHTUNG!!

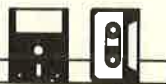
Sie können nur DATA, CP/M (VENDOR) und IBM formatierte Disketten kopieren. Außerdem darf der Menüpunkt – KOPIEREN nur zum Erstellen von Sicherheitskopien für den eigenen Gebrauch benutzt werden. Sie machen sich sonst im Sinne des Urheberrechtsgesetzes strafbar!!!

Sie können auch mit zwei Laufwerken kopieren und zwar von A – A, A – B, B – A und B – B.

Beim Kopieren mit zwei Laufwerken brauchen Sie auch nicht vor jedem Schreib-/Lese-Wechsel eine Taste zu drücken, es läuft automatisch ab.

(Oliver Mayer/CD)

für 464-664-6128



```

10 MODE 2 [513]
20 INK 0,26:INK 1,0:BORDER 26 [1616]
30 POKE &BE78,255 42960 [339]
40 MEMORY &3FFF 16383 [758]
50 RESTORE 2520 [894]
60 FOR a=&A000 TO &A015 40960 40930 [1451]
70 READ a$:POKE a,VAL("&" +a$) [1258]
80 NEXT [350]
90 RESTORE 2530 [900]
100 FOR a=&A03A TO &A065 41408/41064 [1063]
110 READ a$:POKE a,VAL("&" +a$) [1258]
120 NEXT [350]
130 RESTORE 2540 [906]
140 FOR a=&A070 TO &A07C 41072/41034 [1167]
150 READ a$:POKE a,VAL("&" +a$) [1258]
160 NEXT [350]
170 GOSUB 2560 [827]
180 m$=INKEY$ [180]
190 m$=UPPER$(m$) [256]
200 IF m$="" THEN 180 [937]
210 IF m$="A" THEN GOSUB 310 [1063]
220 IF m$="B" THEN GOSUB 1030 [570]
230 IF m$="C" THEN GOSUB 1640 [817]
240 IF m$="D" THEN GOSUB 1710 [703]
250 IF m$="E" THEN GOSUB 1790 [1218]
260 IF m$="F" THEN GOSUB 1850 [1077]
270 IF m$="G" THEN GOSUB 2190 [1518]
280 IF m$="H" THEN GOSUB 2360 [1209]
290 IF m$="X" THEN GOSUB 2450 [1535]
300 GOTO 180 [401]
310 CLS [91]
320 PRINT"Menuepunkt - Diskette formatiere [4902]
n"
330 LOCATE 1,3:INPUT"Laufwerk A/B : ",dr$ [1349]

```

Listing Discmanager

```

340 dr$=UPPER$(dr$) [941]
350 IF dr$="" THEN 310 [633]
360 dr=ASC(dr$)-65 [1708]
370 IF dr<0 OR dr>1 THEN 310 [1440]
380 PRINT:PRINT"welches Format : " [2627]
390 PRINT:PRINT"Data ... 1" [1752]
400 PRINT"CPM ... 2" [1438]
410 PRINT"IBM ... 3" [1110]
420 a$=INKEY$ [278]
430 IF a$="1" THEN sc=&C1 [981]
440 IF a$="2" THEN sc=&41 [820]
450 IF a$="3" THEN sc=1 [1309]
460 IF a$<"1" OR a$>"3" THEN 420 [2291]
470 s=sc:POKE &A00C,sc 40932 [526]
480 POKE &A006,dr 40966 [434]
490 ad=&A016 40932 [730]
500 RESTORE 2550 [848]
510 FOR a=1 TO 9 [767]
520 READ s:POKE ad,0:POKE ad+1,0:POKE ad+2 [2259]
,s+sc:POKE ad+3,2
530 ad=ad+4 [308]
540 NEXT [350]
550 IF sc=&41 THEN GOSUB 690 [904]
560 LOCATE 1,11:PRINT"Bitte Legen sie die [7739]
zu formatierende Disc in Laufwerk "dr$SPC
(20)
570 CALL &BB06 [393]
580 LOCATE 1,13:PRINT"formatiere Spur" [3391]
590 FOR t=0 TO 41 [813]
600 LOCATE 16,13:PRINT t [359]
610 FOR a=&A016 TO &A038 STEP 4:POKE a,t:M [3550]
EXT 40981 41016
620 POKE &A007,t 40964 [382]
630 CALL &A000 40960 [637]
640 NEXT [350]
650 IF sc=&41 THEN GOSUB 870 [900]
660 PRINT:PRINT"Fertig.Bitte eine Taste d [3147]
ruecken"

```

Listing Discmanager


```

670 CALL &BB06 [393]
680 GOSUB 2560:RETURN [1840]
690 LOCATE 1,11:PRINT"Bitte legen sie ein [7218]
e CPM Disc in Drive "dr$" und druecken ein
e Taste
700 CALL &BB06 [393]
710 LOCATE 1,13:PRINT"lese CPM Spuren" [3384]
720 POKE &A043,dr [491]
730 buf=&2000 [460]
740 FOR t=0 TO 1 [578]
750 FOR s=65 TO 73 [1115]
760 b$=HEX$(buf,4) [959]
770 l$=LEFT$(b$,2) [636]
780 r$=RIGHT$(b$,2) [1779]
790 POKE &A040,VAL("&"+r$):POKE &A041,VAL( [1768]
"&"+l$)
800 POKE &A044,t [554]
810 POKE &A046,s [547]
820 CALL &A03A [742]
830 buf=buf+512 [512]
840 NEXT s,t [585]
850 LOCATE 1,13:PRINT SPC(20) [1687]
860 RETURN [555]
870 POKE &A059,dr [394]
880 buf=&2000 [460]
890 LOCATE 1,13:PRINT"schreibe CPM Spuren" [3882]
900 FOR t=0 TO 1 [578]
910 FOR s=&41 TO &49 [1219]
920 b$=HEX$(buf,4) [959]
930 l$=LEFT$(b$,2) [636]
940 r$=RIGHT$(b$,2) [1779]
950 POKE &A056,VAL("&"+r$):POKE &A057,VAL( [1827]
"&"+l$)
960 POKE &A05A,t [734]
970 POKE &A05C,s [783]
980 CALL &A050 [595]
990 buf=buf+512 [512]
1000 NEXT s,t [585]
1010 LOCATE 1,13:PRINT SPC(20) [1687]
1020 RETURN [555]
1030 CLS [91]
1040 PRINT"Menuepunkt - Diskette kopieren" [3371]
1050 LOCATE 1,3:INPUT"Quelldiskette in Lau [4964]
fwerk A/B : ",quell$
1060 q$=UPPER$(quell$) [1220]
1070 IF q$="" THEN 1030 [853]
1080 q=ASC(q$)-65 [876]
1090 IF q>1 OR q<0 THEN 1030 [1188]
1100 LOCATE 1,3:INPUT"Zieldiskette in Lauf [2928]
werk A/B : ",ziel$
1110 z$=UPPER$(ziel$) [1342]
1120 IF z$="" THEN 1100 [1314]
1130 z=ASC(z$)-65 [935]
1140 IF z>1 OR z<0 THEN 1100 [450]
1150 anft=0:endt=6:d=1 [568]
1160 IF q<>z THEN LOCATE 1,3:PRINT"Bitte Q [13536]
uelldiskette in Laufwerk "q$" und Zieldisk
ette in Laufwerk "z$" einlegen":LOCATE 1,5
:PRINT"Dann eine Taste druecken":CALL &BB0
6
1170 ;DRIVE,@q$;:DRIVE,@z$ [2137]
1180 IF q=0 AND z=0 THEN qs=PEEK(&A89F):zs [2809]
=PEEK(&A89F)
1190 IF q=0 AND z=1 THEN qs=PEEK(&A89F):zs [3587]
=PEEK(&A8DF)
1200 IF q=1 AND z=1 THEN qs=PEEK(&A8DF):zs [2816]
=PEEK(&A8DF)
1210 IF q=1 AND z=0 THEN qs=PEEK(&A8DF):zs [3584]
=PEEK(&A89F)
1220 IF qs<>zs THEN LOCATE 1,5:PRINT"Quel [5954]
l und Zieldisc ungleich formatiert.Bitte e
ine Taste druecken.":CALL &BB06:GOSUB 310
1230 buf=&2000 [460]
1240 IF q<>z THEN 1270 [1283]
1250 LOCATE 1,5:PRINT"Bitte Quelldisc in [6563]
Drive "q$" legen dann eine Taste druecken"
1260 CALL &BB06 [393]
1270 POKE &A043,q [449]
1280 sc=qs [244]
1290 FOR t=anft TO endt [1699]
1300 POKE &A044,t [554]
1310 LOCATE 1,7:PRINT"lese Spur "t [1908]
1320 FOR s=sc TO sc+8 [808]
1330 b$=HEX$(buf,4) [959]
1340 POKE &A040,VAL("&"+RIGHT$(b$,2)):POKE [2578]
&A041,VAL("&"+LEFT$(b$,2))

```

```

1350 POKE &A046,s [547]
1360 CALL &A03A [742]
1370 buf=buf+512 [512]
1380 NEXT s,t [585]
1390 LOCATE 1,7:PRINT SPC(20) [1793]
1400 buf=&2000 [460]
1410 IF q<>z THEN 1440 [1241]
1420 LOCATE 1,5:PRINT"Bitte Zieldisc in D [6701]
rive "CHR$(65+z)" legen dann eine Taste dr
uecken "
1430 CALL &BB06 [393]
1440 POKE &A059,dr [394]
1450 sc=zs [173]
1460 FOR t=anft TO endt [1699]
1470 POKE &A05A,t [734]
1480 LOCATE 1,7:PRINT"schreibe Spur "t [3028]
1490 FOR s=sc TO sc+8 [808]
1500 b$=HEX$(buf,4) [959]
1510 POKE &A056,VAL("&"+RIGHT$(b$,2)):POKE [2735]
&A057,VAL("&"+LEFT$(b$,2))
1520 POKE &A05C,s [783]
1530 CALL &A050 [595]
1540 buf=buf+512 [512]
1550 NEXT s,t [585]
1560 LOCATE 1,7:PRINT SPC(20) [1793]
1570 anft=anft+7:endt=endt+7 [1789]
1580 IF anft=35 THEN endt=39 [826]
1590 d=d+1:IF d=7 THEN 1610 [2484]
1600 GOTO 1230 [365]
1610 PRINT"Fertig.Bitte eine Taste druecke [3836]
n"
1620 CALL &BB06 [393]
1630 GOSUB 2560:RETURN [1840]
1640 CLS [91]
1650 PRINT"Menuepunkt - Programm loeschen" [3576]
1660 LOCATE 1,3:CAT [621]
1670 INPUT"Name des Programms : ",les$ [1801]
1680 IF les$="" THEN GOSUB 2560:RETURN [1761]
1690 ;ERA,@les$ [960]
1700 GOSUB 2560:RETURN [1840]
1710 CLS [91]
1720 PRINT"Menuepunkt - Programm umbenenne [4174]
n"
1730 LOCATE 1,3:CAT [621]
1740 INPUT"alter Name : ",alt$ [1195]
1750 INPUT"neuer Name : ",neu$ [1579]
1760 IF alt$="" OR neu$="" THEN GOSUB 2560 [2512]
:RETURN
1770 ;REN,@neu$,@alt$ [1141]
1780 GOSUB 2560:RETURN [1840]
1790 CLS [91]
1800 PRINT"Menuepunkt - Programm laden" [2853]
1810 CAT [96]
1820 INPUT"Welches File : ",l$ [1495]
1830 IF l$="" THEN GOSUB 2560:RETURN [2000]
1840 RUN l$ [259]
1850 CLS [91]
1860 PRINT"Menuepunkt - Fileheader ausgabe [3904]
n"
1870 LOCATE 1,3:INPUT"Ausgabe auf Drucker [4885]
J/N : ",dru$
1880 dru$=UPPER$(dru$) [1335]
1890 IF dru$="J" THEN pr=8 ELSE pr=0 [2054]
1900 CAT [96]
1910 PRINT:INPUT"Welches Programm (mit Ext [5770]
ension zb. BAS oder BIN) : ",he$
1920 IF he$="" THEN GOSUB 2560:RETURN [2288]
1930 he$=UPPER$(he$) [955]
1940 IF MID$(he$,LEN(he$)-3,1)<>". " THEN 1 [2694]
910
1950 FOR a=1 TO LEN(he$) [878]
1960 POKE &A0FF+a,ASC(MID$(he$,a,1)) [1961]
1970 NEXT [350]
1980 POKE &A071,LEN(he$) [752]
1990 CALL &A070 [655]
2000 anf=PEEK(&A76B)*256+PEEK(&A76A) [2252]
2010 lang=PEEK(&A76E)*256+PEEK(&A76D) [1291]
2020 strt=PEEK(&A770)*256+PEEK(&A76F) [1378]
2030 typ=PEEK(&A767) [991]
2040 IF typ=0 THEN t$="Basic":GOTO 2100 [2229]
2050 IF typ=1 THEN t$="Basic geschuetzt":G [2706]
OTO 2100
2060 IF typ=2 THEN t$="Binaer":GOTO 2100 [1951]
2070 IF typ=3 THEN t$="Binaer geschuetzt": [2762]
GOTO 2100
2080 IF typ=22 THEN t$="ASCII Datei":GOTO [1880]

```

Listing Discmanager

```

2100
2090 t$="unbekannt" [1514]
2100 PRINT#pr [748]
2110 PRINT#pr,"Programmname ..... ";he$ [1702]
2120 PRINT#pr,"Ladeadresse ..... &";HEX$(a [1664]
nf,4)
2130 PRINT#pr,"Laenge ..... &";HEX$(l [2627]
ang,4)
2140 PRINT#pr,"Einsprung ..... &";HEX$(s [3790]
trt,4)
2150 PRINT#pr,"Programmart ..... ";t$ [1967]
2160 PRINT:PRINT"Bitte eine Taste druecken [3746]
"
2170 CALL &BB06 [393]
2180 GOSUB 2560:RETURN [1840]
2190 CLS [91]
2200 PRINT"Menuepunkt - Format ermitteln" [4524]
2210 LOCATE 1,3:INPUT"Laufwerk A/B : ",dr$ [1349]
2220 dr$=UPPER$(dr$) [941]
2230 IF dr$="" THEN 2210 [661]
2240 dr=ASC(dr$)-65 [1708]
2250 IF dr<0 OR dr>1 THEN 2210 [1317]
2260 IF dr=0 THEN :A:form=PEEK(&A89F) [1255]
2270 IF dr=1 THEN :B:form=PEEK(&A8DF) [1967]
2280 IF PEEK(&A89F)=&C1 OR PEEK(&A8DF)=&C1 [5081]
THEN frm$="Data":GOTO 2320
2290 IF PEEK(&A89F)=&41 OR PEEK(&A8DF)=&41 [3624]
THEN frm$="CPM":GOTO 2320
2300 IF PEEK(&A89F)=1 OR PEEK(&A8DF)=1 THE [4369]
N frm$="IBM":GOTO 2320
2310 frm$="unbekannt" [389]
2320 LOCATE 1,5:PRINT"Die Disc in Laufwerk [4876]
"dr$" ist "frm$" formatiert"
2330 LOCATE 1,7:PRINT"Bitte eine Taste dru [3904]
ecken"
2340 CALL &BB06 [393]
2350 GOSUB 2560:RETURN [1840]
2360 CLS [91]
2370 PRINT"Menuepunkt - CPM Starten" [3538]
2380 LOCATE 1,3:INPUT"Starten von Laufwerk [3092]
: ",dr$
2390 dr$=UPPER$(dr$) [941]
2400 IF dr$="" THEN GOSUB 2560:RETURN [1986]
2410 dr=ASC(dr$)-65 [1708]
2420 IF dr<0 OR dr>1 THEN 2380 [1299]
2430 :DRIVE,@dr$ [1127]
2440 :CPM [777]
2450 CLS [91]

```

Listing Discmanager

```

2460 PRINT"Menuepunkt - Programm beenden" [3473]
2470 LOCATE 1,3:INPUT"Wirklich J/N : ",en$ [2607]
2480 en$=UPPER$(en$) [1071]
2490 IF en$="J" THEN CALL &BC00:MODE 2:NEW [2018]
2500 IF en$="N" THEN GOSUB 2560:RETURN [1399]
2510 GOTO 2470 [309]
2520 DATA 0E,07,CD,0F,B9,11,00,20,21,16,A0 [1952]
,0E,C1,CD,52,C6,0E,FF,CD,0F,B9,C9
2530 DATA 0E,07,CD,0F,B9,21,00,00,11,00,00 [8785]
,0E,00,CD,66,C6,0E,FF,CD,0F,B9,C9,0E,07,CD
,0F,B9,21,00,00,11,00,00,0E,00,CD,4E,C6,0E
,FF,CD,0F,B9,C9
2540 DATA 06,09,21,00,A1,CD,77,BC,CD,7A,BC [1672]
,C9
2550 DATA 0,5,1,6,2,7,3,8,4 [1008]
2560 MODE 2 [513]
2570 PRINT TAB(18);"Discman V1.0 - Copyrig [3037]
ht 1987 by Oliver Mayer"
2580 PRINT TAB(15);CHR$(150);STRING$(50,15 [2766]
4);CHR$(156)
2590 FOR a=3 TO 15 [762]
2600 LOCATE 15,a:PRINT CHR$(149) [1572]
2610 LOCATE 66,a:PRINT CHR$(149) [1465]
2620 NEXT [350]
2630 PRINT TAB(15);CHR$(147);STRING$(50,15 [3510]
4);CHR$(153)
2640 x=28 [514]
2650 LOCATE 35,4:PRINT"Hauptmenue" [2780]
2660 LOCATE x,6:PRINT"Diskette Formatieren [3671]
... A"
2670 LOCATE x,7:PRINT"Diskette Kopieren .. [3094]
.... B"
2680 LOCATE x,8:PRINT"File loeschen ..... [3448]
.... C"
2690 LOCATE x,9:PRINT"File umbenennen ..... [3650]
.... D"
2700 LOCATE x,10:PRINT"Programm laden .... [3049]
.... E"
2710 LOCATE x,11:PRINT"Fileheader Ausgeben [3724]
.... F"
2720 LOCATE x,12:PRINT"Format ermitteln .. [3606]
.... G"
2730 LOCATE x,13:PRINT"CPM Starten ..... [3384]
.... H"
2740 LOCATE x,14:PRINT"ENDE ..... [2370]
.... X"
2750 RETURN [555]
2760 RUN [243]

```

Listing Discmanager

Disketten und Texte codieren mit CODING

CODING ist ein Codierprogramm, mit dem man Text-Dateien oder ganze Diskettenseiten codieren kann. Es läuft auf allen CPC-Rechnern, benötigt allerdings ein DDI-1 Laufwerk. Will man das Programm auf einem CPC 664/6128 starten, muß es zuvor mit dem CODING.ANP-Programm angepaßt werden. Mit diesem Programm können auch Farben, Drucker-codes und die Größe des Textspeichers angepaßt werden.

Das Programm CODING.LDR enthält den Basic-Lader, der die Datei CODING.BIN erzeugt, die im weiteren mit dem Programm CODING.LOA geladen und gestartet werden kann.

Arbeiten mit CODING

Angenommen, Sie wollen eine Diskettenseite codieren, dann drücken Sie bei der ersten Frage auf »D«. Nun können Sie die Spur, bei der mit dem Codiervorgang begonnen werden soll, eingeben. Die voreingestellte Startspur ist 0, sie wird nach »Start

with track« ausgegeben. Dahinter folgt das Wort »Change?«. Wenn Sie die Startspur nicht verändern wollen, drücken Sie die ENTER-Taste, sonst geben Sie die neue Startspur ein. Das Gleiche wird auch für die Endspur ausgeführt. Daraufhin werden Sie aufgefordert, ein Codewort einzugeben. Als Codewort können Sie einen max. 60 Zeichen langen Text eingeben, in dem auch Control-Zeichen vorkommen können.

VERGESSEN SIE ABER NICHT, SICH DAS CODEWORT GUT ZU MERKEN, BZW. AUFZUSCHREIBEN!!!!

Nach der Eingabe des Codewortes werden Sie aufgefordert, die entsprechende Diskette einzulegen. Legen Sie eine Diskette im DATA-ONLY, SYSTEM/VENDOR oder IBM-Format in Laufwerk A und drücken Sie eine Taste. Die Diskette wird jetzt eingeloggt und das Programm beginnt mit dem Codierungsvorgang. Nach welchem Verfahren hier codiert wird, wird etwas später im Text erläutert. Während des Codierens wird die Nummer des Tracks, der gerade codiert wird, ausgegeben. Das Codieren einer ganzen Diskettenseite dauert etwas länger als das Formatieren einer Diskette unter CP/M. Wenn alles in Ordnung ist, gibt das Programm eine Fertig-Meldung aus, und springt wieder zum Programmanfang. Wenn Sie einen TEXT codieren wol-

1, so muß dieser die Extension .TXT haben. Haben Sie schon vor einen Text codiert, den Sie nun decodieren wollen, so muß die Datei die Extension .COD haben. Nachdem Sie gewählt haben, ob Sie codieren oder decodieren wollen, müssen Sie den Namen der Quelldatei eingeben (nur die ersten 8 Zeichen). Wenn Sie Codieren gewählt haben, müssen Sie als nächstes den Namen der Zieldatei (ebenfalls nur die ersten 8 Zeichen) und das Codewort eingeben (hier gilt das Gleiche wie oben). Die Text-Datei wird nun eingelesen, codiert und als BIN-Datei weggeschrieben. Wollen Sie eine Datei decodieren, so ist als nächstes das Codewort (s.o.) einzugeben. Bei diesem Programmteil können Sie auch noch das Ausgabegerät auswählen. Zur Verfügung stehen: Bildschirm, Drucker und Floppy-Laufwerk.

Wenn Sie das Floppy-Laufwerk wählen, müssen Sie noch den Namen einer Zieldatei angeben. Auf den Drucker können auch Zeichen >CHR\$(127) ausgegeben werden; alle Drucker-codes sind an den NLQ 401 angepaßt. Wenn Sie die Codes ändern wollen, können Sie das mit dem CODING.ANP-Programm tun. Bei der Ausgabe auf den Bildschirm wird immer eine Bildschirmseite komplett beschrieben, danach wartet das Programm auf einen Tastendruck und es geht weiter. Damit die Schlußmeldung den Bildschirminhalt unten nicht zerstört, wird nach Ende des Decodierungsvorgangs noch einmal auf einen Tastendruck gewartet, der Bildschirm um zwei Zeilen nach oben geschoben und dann erst die Meldung ausgegeben. Das Programm springt danach wieder an den Anfang.

Der Codierungsvorgang

Beispiel: Sie wollen den Text »Ich heiße Otto, und wie heißt du?« mit dem Codewort »HALLO DU!« codieren. Die Meldung: ICH HEISSE OTTO, UND WIE HEISST DU? und das Codewort:

HALLO DU!HALLO DU!HALLO DU!HALLO DU

Die beiden ASCII-Werte der beiden übereinanderstehenden Buchstaben werden mit XOR codiert und das Ergebnis kommt wieder in den Speicher. Wenn der codierte Text wieder mit dem Codewort codiert wird, kommt der ursprüngliche Text wieder heraus.

Das Anpassungs-Programm

Bei allen Eingaben im CODING.ANP-Programm muß unbedingt auf die Richtigkeit der Eingaben geachtet werden, da insbesondere die Zahleneingaben nicht auf ihre Richtigkeit überprüft werden. Wenn Sie die alten Werte beibehalten wollen, so drücken Sie nur die ENTER-Taste. Alte Werte stehen ggf. in Klammern. Zuerst geben Sie die von Ihnen gewünschten Farben ein. Als nächstes folgt die Eingabe für Start- und Endtrack. Dieses erleichtert ihnen die

Arbeit, wenn Sie mehrmals die gleichen Tracks codieren möchten und nicht jedesmal Ihre Eingaben neu machen wollen. Die nächste Frage werden Sie wohl sicher mit ENTER bestätigen, es sei denn, Sie hätten das Floppy-ROM z.B. in ein EPROM gebrannt und vorher die ROM-Nummer verändert. Dann müßten Sie diese neue Nummer angeben. Die letzte Frage auf dieser Bildschirmseite bezieht sich auf die Größe des Textspeichers. Wählen Sie aber keine kleineren Werte als 4608, denn 4608 ist die Länge eines Tracks im DATA-ONLY und im SYSTEM/VENDOR-Format, und bei der Routine SEKTOR lesen / schreiben findet keine Abfrage statt, ob genügend Speicherplatz vorhanden ist. Zwischen 4608 und dem angegebenen Höchstwert können Sie aber frei wählen. Falls Sie beispielsweise noch einen Zeichensatz einbauen wollen, so entfernen Sie das SYMBOL AFTER 256 aus dem CODING.LOA-Programm und ersetzen es entsprechend. Achten Sie aber darauf, daß Sie Ihrem Zeichensatz genügend Platz beschaffen. Er wird sonst überschrieben!

Die Verkleinerung des Textspeichers kann auch für ROM-Erweiterungen nützlich sein, die RAM-Speicher benötigen. Auf der nächsten Bildschirmseite können Sie die Drucker-codes anpassen. Falls Sie die Codes ändern wollen, drücken Sie die »A«-Taste. Das Programm fragt dann die einzelnen Codes ab. Danach können Sie auch noch die 30 freien Speicherstellen mit Drucker-codes belegen. Achten Sie aber darauf, daß keine 255 vorkommt, weil die Zahl 255 das Kennzeichen für das Tabellenende ist. ENTER bedeutet auch hier, daß der alte Wert beibehalten wird. Falls Sie einen Code in 0 umändern wollen, müssen Sie auch eine »0« drücken und nicht wie üblich ENTER! Danach folgt die Anpassung für die CPC-Computer. Drücken Sie die entsprechende Taste für Ihren Rechner. Falls Sie einen Buchstaben drücken, wird CPC-464 angenommen! Letztendlich folgt noch die Speicheraufteilung und die Abspeicherung der neuen Version.

Zum Schluß dieser Datei sollen noch einige Bemerkungen zur Disketten-Codierung kommen: Vermeiden Sie es, falls die Diskette noch relativ neu ist, ALLE Spuren zu codieren, denn auf den letzten Spuren befinden sich meistens noch E5-Bytes. Durch diesen Wert ist es ziemlich einfach, das Codewort herauszufinden. Man braucht nur alle Bytes eines Sektors mit XOR E5 zu verknüpfen, um das Codewort herauszufinden. Wenn Sie dies verhindern wollen, codieren Sie die Diskette zweimal hintereinander mit verschiedenen Codewörtern, vergessen Sie aber die Reihenfolge nicht! Die falsche Reihenfolge könnte tödlich für Ihre Daten sein!!!

Denken Sie auch daran, daß das Codierungsprogramm nicht auf der Diskette sein sollte, die Sie gerade codieren. Kopieren Sie das Programm vorher auf eine andere Diskette, denn wenn Sie einmal alle Tracks einer Diskette codiert haben und sich auf dieser Diskette das Codierungsprogramm befand, von dem Sie kein BACKUP auf einer anderen Diskette hatten, dürfte es für Ihre Dateien und dem Codierprogramm zu spät sein!

(J. Schefers/JB)

für 464-664-6128



```

1 ' BASIC-Loader fuer CODING V2.0           [2799]
2 ' Copyright (C) by J.S.S. - Software       [902]
3 '                                           [117]
4 adr=&4000:n$="CODING.BIN"                 [2628]
5 IF HIMEM<adr-1 THEN PRINT"Computer vor d [6105]
em Start zuruecksetzen !":NEW
6 MEMORY adr-1                             [160]
7 MODE 1:LOCATE 1,3:PRINT>Loading ... "+n$ [3638]
+" ... Please wait"
8 LOAD n$,adr                               [711]
9 CALL adr                                  [311]

```

Listing Coding

```

1 ' BASIC-Loader fuer CODING V2.0           [2799]
2 ' Copyright (C) Juni 1987 by              [889]
3 ' J.S.S. - Software                       [1196]
4 '                                           [117]
10 DATA 0E,07,CD,15,B9,FE,80,20,08,21,A5,4 [4852]
7,CD,8F,43,F3,76,21,F1,BD, 2362
11 DATA 11,9D,48,01,03,00,C5,E5,ED,B0,E1,C [4602]
1,EB,ED,B0,21,0D,45,06,4F, 2355
12 DATA 7E,CD,5A,BB,23,10,F9,06,1D,3E,2D,C [3870]

```

Listing Coding

Anwendungen

D,14,44,3E,51,32,8B,B2,11, 1870
 13 DATA FD,47,0E,08,DF,94,48,13,01,03,00,2 [4391]
 1,15,48,C5,ED,B0,E5,0E,08, 1799
 14 DATA DF,94,48,13,E1,C1,ED,B0,21,27,47,C [2493]
 D,8F,43,CD,1A,44,FE,54,28, 2528
 15 DATA 06,FE,44,28,2E,18,F3,CD,5A,BB,21,5 [4408]
 C,45,CD,8F,43,06,08,21,FD, 2072
 16 DATA 47,CD,9C,43,21,C2,45,CD,8F,43,21,C [3635]
 6,45,CD,8F,43,CD,1A,44,FE, 2478
 17 DATA 44,CA,7D,42,FE,43,CA,70,41,18,F1,C [2315]
 D,5A,BB,21,F7,46,CD,8F,43, 2673
 18 DATA 3A,84,48,CD,21,44,3E,23,CD,6F,BB,2 [3899]
 1,40,47,CD,8F,43,CD,4E,44, 2102
 19 DATA FE,FF,20,05,3A,84,48,18,09,FE,28,3 [3853]
 8,05,CD,3D,44,18,EB,F5,21, 2067
 20 DATA 0F,47,CD,8F,43,3A,85,48,CD,21,44,3 [3413]
 E,23,CD,6F,BB,21,40,47,CD, 2043
 21 DATA 8F,43,CD,4E,44,FE,FF,20,05,3A,85,4 [2664]
 8,18,09,FE,28,38,05,CD,3D, 2024
 22 DATA 44,18,EB,47,F1,B8,28,0A,38,08,21,4 [3952]
 A,47,CD,8F,43,18,AF,32,84, 1911
 23 DATA 48,78,32,85,48,CD,DF,43,21,57,47,C [3480]
 D,8F,43,CD,FF,44,3A,9F,A8, 2461
 24 DATA E6,FE,32,88,48,1E,00,21,A3,58,B7,2 [2190]
 8,03,21,A3,5A,22,89,48,3A, 1869
 25 DATA 84,48,57,47,3A,85,48,90,3C,47,21,8 [3220]
 F,47,CD,8F,43,7A,CD,21,44, 1990
 26 DATA D5,C5,3E,66,32,8B,48,CD,DA,44,2A,8 [4107]
 9,48,22,81,48,CD,B7,44,C1, 2461
 27 DATA D1,D5,C5,3E,4E,32,8B,48,CD,DA,44,C [3485]
 1,D1,14,10,D2,C3,F6,41,21, 2698
 28 DATA 08,46,CD,8F,43,C3,02,42,CD,5A,BB,2 [3408]
 1,75,45,CD,8F,43,06,08,21, 1919
 29 DATA 09,48,CD,9C,43,CD,DF,43,CD,F9,44,2 [2429]
 1,95,45,CD,8F,43,06,0F,CD, 2418
 30 DATA 12,44,21,C5,46,CD,8F,43,21,FD,47,0 [4623]
 6,0C,11,00,38,CD,77,BC,D2, 1971
 31 DATA 02,42,FE,16,C2,14,42,21,A3,48,7D,F [2363]
 E,FB,20,05,7C,FE,A6,28,AF, 2318
 32 DATA CD,80,BC,FE,1A,28,04,77,23,18,EB,3 [3061]
 6,1A,23,22,81,48,CD,7A,BC, 2123
 33 DATA CD,B7,44,AE,12,21,09,48,06,0C,11,0 [2513]
 0,38,CD,8C,BC,30,24,11,A3, 1650
 34 DATA 48,B7,2A,81,48,ED,52,EB,21,A3,48,0 [2403]
 1,00,00,3E,02,CD,98,BC,CD, 2135
 35 DATA 8F,BC,21,9A,45,CD,8F,43,CD,06,BB,C [5003]
 3,23,40,CD,8F,BC,CD,7A,BC, 2745
 36 DATA 21,D8,47,CD,8F,43,CD,06,BB,C3,23,4 [4561]
 0,21,B2,45,CD,8F,43,18,E6, 2376
 37 DATA 3A,83,48,FE,01,20,0B,06,0C,3E,0A,C [3895]
 D,2B,BD,10,FB,18,C8,CD,8F, 1925
 38 DATA BC,0F,38,C2,CD,06,BB,21,F7,47,CD,8 [3596]
 F,43,18,B7,21,15,48,11,12, 1985
 39 DATA 48,01,03,00,ED,B0,21,75,45,CD,8F,4 [2528]
 3,06,08,21,09,48,CD,9C,43, 1679
 40 DATA 21,C2,45,CD,8F,43,06,0F,CD,12,44,2 [4200]
 1,C5,46,CD,8F,43,21,09,48, 1852
 41 DATA 06,0C,11,00,38,CD,8C,BC,30,8C,3E,0 [4274]
 2,32,83,48,18,32,CD,5A,BB, 1685
 42 DATA 21,12,48,11,06,48,01,03,00,ED,B0,C [2776]
 D,DF,43,CD,F9,44,21,18,46, 1779
 43 DATA CD,8F,43,CD,1A,44,FE,53,28,0A,FE,5 [2434]
 0,28,72,FE,46,28,99,18,EF, 2369
 44 DATA AF,32,83,48,CD,6C,BB,21,FD,47,06,0 [3171]
 C,11,00,38,CD,77,BC,D2,02, 2100
 45 DATA 42,FE,02,20,4A,69,60,01,A3,48,C5,0 [4517]
 9,7C,FE,A6,38,0D,FE,A7,30, 2153
 46 DATA 05,7D,FE,FB,38,04,C1,C3,67,41,E1,C [3031]
 D,83,BC,CD,7A,BC,11,A3,48, 2767
 47 DATA 0E,00,21,43,48,06,00,3A,80,48,B8,2 [4765]
 8,F5,1A,AE,FE,1A,CA,1C,42, 1695
 48 DATA F5,3A,83,48,B7,28,58,3D,28,4D,F1,C [3536]
 D,95,BC,23,13,04,18,E0,21, 2117
 49 DATA F6,45,CD,8F,43,C3,02,42,3E,01,32,8 [3139]
 3,48,21,58,46,CD,8F,43,CD, 2120
 50 DATA FF,44,CD,2E,BD,38,FB,21,1B,48,7E,F [2834]
 E,FF,28,06,CD,2B,BD,23,18, 2379
 51 DATA F5,21,C2,45,CD,8F,43,06,13,CD,12,4 [3234]
 4,21,76,46,CD,8F,43,21,D9, 2158

Listing Coding

52 DATA 46,CD,8F,43,C3,AF,42,F1,CD,2B,BD,3 [3449]
 0,FB,18,AF,F1,FE,0A,CC,76, 2924
 53 DATA 43,B7,28,A6,FE,0A,38,07,FE,0E,DC,5 [3770]
 A,BB,38,9B,D5,E5,C5,CD,5D, 2696
 54 DATA BB,C1,E1,D1,18,90,0C,79,FE,19,20,1 [2435]
 0,CD,06,BB,D5,E5,C5,CD,6C, 2792
 55 DATA BB,C1,E1,D1,0E,00,AF,C9,3E,0A,C9,F [4098]
 5,7E,B7,28,06,CD,5A,BB,23, 2594
 56 DATA 18,F6,F1,C9,C5,CD,81,BB,CD,06,BB,F [4491]
 E,0D,28,16,FE,7F,28,1B,E5, 2834
 57 DATA C5,F5,CD,5D,BB,F1,C1,E1,77,23,10,E [4410]
 5,CD,84,BB,C1,C9,CD,84,BB, 3427
 58 DATA 78,C1,B8,28,D7,C9,78,C1,B8,28,D1,F [4077]
 5,2B,36,20,3E,08,CD,5A,BB, 2625
 59 DATA B7,17,CD,5A,BB,F1,C5,3C,47,18,BE,2 [3556]
 1,E0,45,CD,8F,43,06,3C,21, 2311
 60 DATA 43,48,E5,CD,9C,43,36,00,D1,B7,ED,5 [3643]
 2,7D,3C,32,80,48,C9,FE,81, 2580
 61 DATA DA,9D,48,F5,3E,1B,CD,9D,48,3E,3D,C [2039]
 D,9D,48,F1,CD,9D,48,AF,C3, 2817
 62 DATA 9D,48,3E,20,CD,5A,BB,10,FB,C9,CD,0 [3254]
 6,BB,DF,91,48,C9,E5,D5,C5, 2951
 63 DATA 26,00,6F,11,0A,00,DF,9A,48,7D,C6,3 [3883]
 0,CD,5A,BB,7B,C6,30,CD,5A, 2142
 64 DATA BB,C1,D1,E1,C9,3E,08,CD,5A,BB,17,C [3482]
 D,5A,BB,1F,CD,5A,BB,17,C3, 2803
 65 DATA 5A,BB,AF,21,86,48,77,23,77,2B,06,0 [3407]
 0,04,78,FE,03,28,31,CD,81, 1817
 66 DATA BB,CD,06,BB,FE,0D,28,27,FE,7F,28,1 [3469]
 1,FE,30,38,F1,FE,3A,30,ED, 2565
 67 DATA CD,5A,BB,D6,30,77,23,18,DB,78,FE,0 [3843]
 1,28,DF,3E,08,CD,5A,BB,17, 2354
 68 DATA CD,5A,BB,05,2B,18,D2,CD,84,BB,78,F [3810]
 E,01,28,1D,2A,86,48,11,0A, 2007
 69 DATA 00,E5,FE,03,28,09,2E,00,DF,97,48,C [2885]
 1,79,85,C9,26,00,DF,97,48, 2159
 70 DATA C1,78,85,C9,3E,FF,C9,11,A2,48,06,0 [2254]
 0,21,42,48,13,23,04,3A,80, 1837
 71 DATA 48,B8,28,F2,DD,2A,81,48,DD,7C,BA,2 [3034]
 0,04,DD,7D,BB,C8,1A,AE,12, 2520
 72 DATA 18,E5,3A,90,A8,B7,1F,1F,ED,4B,88,4 [3811]
 8,47,21,A3,48,0C,C5,D5,E5, 2378
 73 DATA DF,8B,48,E1,D1,01,00,02,09,C1,10,F [4130]
 0,C9,21,86,46,CD,8F,43,CD, 2387
 74 DATA 8A,BB,CD,06,BB,CD,8D,BB,AF,DF,8E,4 [3305]
 8,C9,04,02,1D,02,02,1C,00, 2136
 75 DATA 00,00,1C,01,1A,1A,1F,1C,03,43,20,4 [2742]
 F,20,44,20,49,20,4E,20,47, 739
 76 DATA 20,20,20,56,45,52,53,49,4F,4E,20,3 [2593]
 2,2E,30,1F,18,06,A4,20,4A, 1153
 77 DATA 55,4E,49,20,31,39,38,37,20,42,59,2 [2277]
 0,4A,2E,53,2E,53,2E,20,2D, 1159
 78 DATA 20,53,4F,46,54,57,41,52,45,1F,1A,0 [3000]
 4,1F,03,0A,4E,61,6D,65,20, 1173
 79 DATA 6F,66,20,73,6F,75,72,63,65,20,66,6 [3748]
 9,6C,65,3A,20,00,0A,0D,0A, 1473
 80 DATA 20,20,4E,61,6D,65,20,6F,66,20,64,6 [2504]
 5,73,74,69,6E,61,74,69,6F, 1802
 81 DATA 6E,20,66,69,6C,65,3A,20,00,0A,0D,0 [3012]
 A,0A,00,1F,1E,18,28,44,65, 985
 82 DATA 2D,29,43,6F,64,69,6E,67,20,63,6F,6 [4221]
 D,70,6C,65,74,65,00,0A,0D, 1594
 83 DATA 0A,4E,6F,20,41,53,43,49,49,2D,46,6 [2491]
 9,6C,65,0A,0D,0A,00,20,20, 1118
 84 DATA 43,27,6F,64,69,6E,67,20,6F,72,20,4 [3718]
 4,27,65,63,6F,64,69,6E,67, 1760
 85 DATA 20,3F,20,00,0A,0D,0A,20,20,45,6E,7 [3310]
 4,65,72,20,63,6F,64,65,77, 1296
 86 DATA 6F,72,64,3A,20,00,0A,0D,0A,4E,6F,2 [3252]
 0,62,69,6E,61,72,79,20,66, 1448
 87 DATA 69,6C,65,00,0A,0D,0A,46,69,6C,65,2 [3837]
 0,74,6F,6F,20,62,69,67,00, 1439
 88 DATA 0A,0D,0A,20,20,53,65,6C,65,63,74,2 [3058]
 0,6F,75,74,70,75,74,20,64, 1558
 89 DATA 65,76,69,63,65,3A,20,53,27,63,72,6 [3164]
 5,65,6E,2C,20,50,27,72,69, 1675
 90 DATA 6E,74,65,72,20,6F,72,20,46,27,6C,6 [4190]
 F,70,70,79,2D,64,69,73,63, 1867
 91 DATA 20,3F,20,00,0A,0D,0A,20,20,50,6C,6 [2956]

Listing Coding

```

5,61,73,65,20,74,75,72,6E, 1315
92 DATA 20,70,72,69,6E,74,65,72,20,6F,6E,3 [2685]
A,5F,00,50,20,52,20,49,20, 1541
93 DATA 4E,20,54,20,49,20,4E,20,47,00,0A,0 [2960]
D,0A,20,20,49,6E,73,65,72, 1122
94 DATA 74,20,64,69,73,63,20,77,69,74,68,2 [3513]
0,74,68,65,20,66,69,6C,65, 1844
95 DATA 20,74,6F,20,62,65,20,28,64,65,2D,2 [2495]
9,63,6F,64,65,64,20,69,6E, 1607
96 DATA 74,6F,20,64,72,69,76,65,20,41,3A,5 [3725]
F,00,28,44,20,45,20,2D,20, 1365
97 DATA 29,20,43,20,4F,20,44,20,49,20,4E,2 [3506]
0,47,20,20,2E,20,2E,20,2E, 935
98 DATA 20,20,50,20,4C,20,45,20,41,20,53,2 [2839]
0,45,20,20,57,20,41,20,49, 1019
99 DATA 20,54,00,0A,0D,0A,20,20,53,74,61,7 [2992]
2,74,20,77,69,74,68,20,74, 1363
100 DATA 72,61,63,6B,3A,20,00,0A,0D,0A,20, [3744]
20,53,74,6F,70,20,20,77,69, 1314
101 DATA 74,68,20,74,72,61,63,6B,3A,20,00, [4980]
0A,0D,0A,0A,0A,20,20,54,27, 1115
102 DATA 65,78,74,20,6F,72,20,44,27,69,73, [3647]
63,20,3F,20,00,43,68,61,6E, 1557
103 DATA 67,65,20,3F,20,00,1F,2D,0C,10,08, [3555]
10,0B,0B,10,09,10,08,00,0A, 540
104 DATA 0D,0A,20,20,50,6C,65,61,73,65,20, [2996]
69,6E,73,65,72,74,20,64,69, 1619
105 DATA 73,63,20,74,6F,20,62,65,20,28,64, [4086]
65,2D,29,63,6F,64,65,64,20, 1606
106 DATA 69,6E,74,6F,20,64,72,69,76,65,20, [2790]
41,3A,5F,00,1F,03,12,28,44, 1422
107 DATA 65,2D,29,43,6F,64,69,6E,67,20,74, [3229]
72,61,63,6B,20,00,04,02,50, 1466
108 DATA 72,6F,67,72,61,6D,20,73,74,61,72, [3185]
74,73,20,6F,6E,6C,79,20,77, 1986
109 DATA 69,74,68,20,63,6F,6E,6E,65,63,74, [3370]
65,64,20,46,6C,6F,70,70,79, 1970
110 DATA 2D,52,4F,4D,20,21,07,00,1F,1D,18, [3286]
45,72,72,6F,72,20,77,68,69, 1321

```

Listing Coding

```

111 DATA 6C,65,20,64,69,73,63,2D,6F,70,65, [3420]
72,61,74,69,6F,6E,07,00,1F, 1720
112 DATA 01,19,0A,0A,00,20,20,20,20,20, [2103]
20,20,2E,54,58,54,20,20,20, 700
113 DATA 20,20,20,20,20,2E,43,4F,44,54,58, [2852]
54,43,4F,44,1B,49,01,1B,36, 1072
114 DATA 1B,38,1B,3C,00,00,00,00,00,00, [4098]
00,00,00,00,00,00,00,00, 170
115 DATA 00,00,00,00,00,00,00,00,00,00, [2241]
00,00,00,00,00,00,00,00, 255
116 DATA 00,00,00,00,00,00,00,00,00,00, [1858]
00,00,00,00,00,00,00,00, 0
117 DATA 00,00,00,00,00,00,00,00,00,00, [1858]
00,00,00,00,00,00,00,00, 0
118 DATA 00,00,00,00,00,00,00,00,00,00, [1858]
00,00,00,00,00,00,00,00, 0
119 DATA 00,27,00,00,00,00,00,66,C6,07,DA, [3173]
CD,07,A6,CA,07,85,DB,07,50, 1590
120 DATA 37,FE,8C,37,FE,00,00,00,C3,FA,00, [2216]
1203
121 MEMORY &3FFF [758]
122 dat=0:sz=0:dz=10 [319]
123 FOR adr=&4000 TO &48A2 [1021]
124 READ byte$:byte=VAL("&"+byte$):dat=dat [2593]
+1
125 sz=sz+byte [619]
126 POKE adr,byte [84]
127 IF dat<20 AND adr<&48A2 THEN 131 [1517]
128 READ chksum [1222]
129 IF chksum<>sz THEN PRINT "Fehler in Ze [3056]
ile :";dz
130 dz=dz+1:sz=0:dat=0 [1666]
131 NEXT adr [547]
132 PRINT [361]
133 PRINT"Aufruf mit 'CALL &4000'" [1469]
134 PRINT"Zum Abspeichern bitte 'CONT' ein [5405]
geben !"
135 STOP [464]
136 SAVE"coding",b,&4000,&8A2 [1047]
137 END [110]

```

Listing Coding

Sie wollen uns ein Programm zusenden! Folgende Regeln sollten Sie beachten:

- 1) Schicken Sie Ihre Programme auf Kasette oder Diskette. Dabei mehrmals hintereinander abspeichern.
- 2) Legen Sie eine ausführliche Programmbeschreibung bei, aus der eindeutig der Sinn und Zweck des Programmes hervorgeht. Eine Variablenliste ist sehr wichtig, in ihr sollten alle im Programm definierten Variablen mit ihrer Verwendung aufgeführt sein.
Beispiel: n\$ = Nachname
g = glücklicher usw.
- 3) Wenn Sie gleichzeitig Besitzer eines Druckers sind, schicken Sie auf jeden Fall ein Programmlisting mit.
- 4) Vergessen Sie Ihren Absender nicht! Vollständige Anschrift, wenn möglich auch die Telefonnummer, ersparen unnötige Verzögerungen unsererseits.
- 5) Behalten Sie auf jeden Fall eine Kopie Ihres Programms, unvorhersehbare Umstände, wie z.B. Brandriß der Kasette werden somit nicht zur Katastrophe.
- 6) Wenn Sie sogar mehrere Programme geschrieben haben und diese veröffentlichen wollen, nehmen Sie bitte für jedes Programm eine separate Kasette (mehrmals abspeichern!).
Damit erleichtern Sie uns die Arbeit wesentlich und eine schnellere Bearbeitung wird möglich
- 7) Versehen Sie das Programm mit den sehr wichtigen REM-Statements. Das fördert nicht nur die Übersicht für uns und unsere Leser, sondern unterstützt zusätzlich das strukturierte Programmieren. Merke: Auch erfahrene Programmierer verwenden REM-Zeilen, das Programm sollte schließlich jeder verstehen und anwenden können.
- 8) Vermeiden Sie möglichst Variablennamen, wo leicht I und i oder O und o verwechselt werden können. Variablen sollten grundsätzlich in Kleinbuchstaben geschrieben werden und möglichst verständlich sein. Die Variable butter sagt mehr aus als nur bu.
- 9) Bieten Sie das Programm niemals verschiedenen Verlagen gleichzeitig an! Mit der Veröffentlichung und dem daraus resultierenden Honorar, gehen die Urheberrechte an den Verlag über! Sollte Ihr Programm noch in einer anderen Zeitschrift abgedruckt werden, liegt ein Verstoß gegen das Wettbewerbsrecht vor.
- 10) Jedes bei uns ordnungsgemäß eingegangene Programm wird ausführlich begutachtet und getestet. Sie erhalten von uns dann umgehend Bescheid, ob Ihr Programm veröffentlicht wird oder nicht. Wenn Sie das Programm nach unserer Begutachtung zurückhaben wollen, legen Sie bitte der Einsendung einen frankierten Rückumschlag bei. Die Kosten einer eventuellen Rücksendung sind in jedem Fall vom Einsender zu tragen

Falls wir uns für den Abdruck Ihres Programmes entschieden haben, setzen wir uns unverzüglich mit Ihnen in Verbindung und besprechen Ihr verdientes Honorar!

Übrigens wäre es nett, wenn Sie uns auch ein paar persönliche Daten schreiben würden (z.B. Ihr Alter, Tätigkeit, wie und wann kam die Idee zu dem Programm, was alles machen Sie mit dem CPC usw.?). So, jetzt aber genug der Theorie, mit fast unerträglicher Spannung und Neugier wartet Ihre »PC Schneider International« auf Ihre Beiträge.

Einsendung bitte an: DMV – Daten & Medien Verlagsges.mbH, Fuldaer Straße 6, 3440 Eschwege

PRO-DATA

– Eine Dateiverwaltung für alle Fälle!

Brauchen Sie ein Programm, mit dem Sie Ihre Adressen, Schallplatten, CD's, Videos, Disketten oder was Sie wollen, verwalten können? Hier haben Sie es!

Das Beste zuerst: Um Ihrer individuellen Datei auch das besondere Aussehen zu verleihen, befindet sich im Hauptmenue von PRO-DATA ein Masken-Generator, mit dem Sie die Datei-Masken problemlos selbst erstellen können. Aber jetzt zur Bedienung des Programms. Starten Sie das Programm durch Eingabe von RUN »PRO-DATA«. Zunächst müssen Sie die aktuelle Uhrzeit (St, Min, Sec) eingeben. Anschließend erscheint folgendes Hauptmenue:

- Maske definieren
- Eingabe/Blättern
- Daten speichern
- Maske speichern
- Daten sortieren
- Daten laden
- Maske laden
- Daten löschen
- Directory
- Ende
- Daten drucken

Vor dem Menüpunkt »Maske definieren« befindet sich ein kleiner Pfeil. Diesen Pfeil können Sie nun mit den Cursortasten steuern. Setzen Sie durch Drücken von <Cursor hoch>, <Cursor ab>, <Cursor links> und <Cursor rechts> den Pfeil so, daß er auf den gewünschten Menüpunkt zeigt.

Durch Drücken von COPY oder ENTER aktivieren Sie nun diesen Menüpunkt. Unter dem Menue wird Ihnen noch mitgeteilt, ob sich eine Maske und ob sich Daten im Speicher befinden.

Die einzelnen Menüpunkte

1.) Maske definieren

Um überhaupt anfangen zu können, müssen Sie zuerst eine Maske definieren. Als erstes werden Sie gefragt, wieviel Felder ein Datensatz haben soll. Geben Sie eine Zahl zwischen 1 und 10 ein (ENTER = Rücksprung). Danach werden die Namen für die einzelnen Felder abgefragt. Geben Sie die gewünschten Namen ein. Am Ende werden Sie gefragt, ob alles richtig ist. Drücken Sie <N>, können Sie die Maske nochmal neu erstellen, drücken Sie <J>, wird ins Hauptmenue zurückgesprungen. Befinden sich Daten im Speicher, so können Sie die Maske nicht ändern.

2.) Maske speichern

Um eine Maske nicht immer wieder neu eingeben zu müssen, können Sie mit diesem Menüpunkt die Maske auf Dis-

kette (Kassette) speichern. Nach Eingabe des Dateinamens wird die Maske gespeichert.

3.) Maske laden

Mit diesem Menüpunkt können Sie eine Maske laden. Vorher müssen Sie den Dateinamen eingeben. Drücken Sie nur ENTER, wird ins Hauptmenue zurückgesprungen (bei »Maske speichern« übrigens auch), andernfalls wird die Maske geladen.

4.) Eingabe/Blättern

Dieser Menüpunkt ist der Kern des Programms. Sie können den Menüpunkt nur aktivieren, wenn sich eine Maske im Speicher befindet. Es erscheint die Maske und, falls vorhanden, der erste Datensatz.

Nun können Sie Daten eingeben und eingegebene Daten komfortabel ändern. Sie können den Cursor mit den Cursortasten innerhalb der Maske frei bewegen und Änderungen vornehmen. Auch die ENTER-Taste hat ihre gewohnte Funktion. Probieren Sie es doch einfach aus. Sie werden schnell feststellen, wie bequem diese Dateneingabe ist.

Zeichen löschen können Sie wie gewohnt mit <CLR> oder . Drücken Sie <Cursor hoch> oder <Cursor ab>, so plaziert sich der Cursor immer auf das erste Zeichen des neuen Datenfeldes. Die Länge eines Datenfeldes ist allerdings auf eine Zeile begrenzt. Durch gleichzeitiges Drücken von SHIFT und <Cursor ab> bzw. <Cursor hoch> gelangen Sie zum nächsten bzw. zum vorhergehenden Datensatz. Der Cursor befindet sich dann auf dem ersten Zeichen des ersten Datenfeldes. Sie kommen allerdings nur zum nächsten Datensatz, wenn im vorhergehenden auch wirklich (zumindest im ersten Datenfeld) Daten stehen.

Durch Drücken von CTRL S können Sie nach einem bestimmten Ausdruck suchen lassen. In der letzten Zeile werden Sie gefragt: »Gesuchter Ausdruck:«. Hier geben Sie nun Ihren Suchstring ein. Anschließend werden Sie gefragt: »In allen Feldern oder nur in einem bestimmten Feld suchen (1/2)?«. Wissen Sie nicht, in welchem Datenfeld der Ausdruck stehen muß, so drücken Sie <1>, andernfalls drücken Sie <2> und geben anschließend die Datenfeld-Nummer ein.

Es beginnt die Suche nach dem Ausdruck. Findet der Rechner den gesuchten Ausdruck, so wird der Datensatz, in dem der Ausdruck steht, ausgegeben. Der Cursor befindet sich dann am Anfang des Feldes, wo der Ausdruck gefunden wurde. Findet der Rechner den Ausdruck nicht, so wird Ihnen dies mitgeteilt und nach Tastendruck zurückgesprungen. Durch Drücken von CTRL G können Sie einen bestimmten Datensatz anwählen. In der letzten Zeile werden Sie gefragt: »Datensatz-Nummer:«. Geben Sie die gewünschte Nummer ein. Anschließend wird der Datensatz ausgegeben und der Cursor auf das erste Zeichen des ersten Datenfeldes gesetzt. Allerdings nur, wenn die eingegebene Nummer auch im

erlaubten Bereich liegt, nämlich in dem Bereich, in dem auch wirklich Daten stehen, andernfalls wird nochmal abgefragt. Weiterhin können Sie durch Drücken von CTRL L den momentan sichtbaren Datensatz löschen. Schließlich können Sie durch Drücken von ESC zurück ins Hauptmenue springen. Übrigens werden diese Funktionen in der letzten Zeile als Gedächtnishilfe aufgeführt.

5.) Daten sortieren

Haben Sie diesen Menüpunkt angewählt, so werden Sie gefragt, nach welchem Feld sortiert werden soll.

Geben Sie die Nummer des gewünschten Feldes ein. Anschließend werden die Daten nach dem eingegebenen Feld alphabetisch sortiert. Dies kann eine Weile dauern. Hier müssen Sie also etwas Geduld haben, vor allem, wenn Sie viele Datensätze im Speicher haben. Dauert es Ihnen zu lang, können Sie das Sortieren durch Drücken von ESC vorzeitig abbrechen.

6.) Daten löschen

Wählen Sie diesen Menüpunkt an, so werden Sie, vorausgesetzt es befinden sich Daten im Speicher, in der letzten Zeile gefragt, ob Sie wirklich alle Daten löschen wollen. Drücken Sie <J>, werden die Daten gelöscht.

7.) Ende

Mit diesem Menüpunkt verlassen Sie das Programm.

8.) Daten speichern

Hier können Sie Daten auf Diskette (Kassette) speichern. Nach Eingabe des Dateinamens werden die Daten gespeichert. Drücken Sie bei der Eingabe des Dateinamens nur ENTER, so wird sofort wieder ins Hauptmenue zurückgesprungen.

9.) Daten laden

Unter dieser Rubrik können Sie Daten von Diskette (Kassette) laden. Nach Eingabe des Dateinamens werden die Daten geladen. Drücken Sie bei der Eingabe des Dateinamens nur ENTER, so wird ebenfalls wieder ins Hauptmenue zurückgesprungen.

10.) Directory

Hier können Sie sich das Inhaltsverzeichnis (Directory) einer Diskette zeigen lassen (entspricht CAT). Nach Tastendruck wird wieder ins Hauptmenue gesprungen.

11.) Daten drucken

Mit diesem Menüpunkt können Sie Daten auf dem Drucker ausgeben. Vorher können Sie eine Überschrift des Datenausdrucks eingeben (z.B. »Adressen meiner Freunde in NRW«). Schließlich werden Sie noch gefragt, ob die Maske mit ausgedruckt werden soll. Drücken Sie nach Wunsch <J> oder <N>. Anschließend wird die Überschrift auf den Drucker ausgegeben und unterstrichen. Dann folgen die Datensätze. Zwischen jedem Datensatz wird eine Leerzeile frei gelassen. Wieviel Datensätze auf eine Seite passen, wird vom Programm berechnet. Ist eine Seite vollständig bedruckt, werden Sie aufgefordert, eine neue Seite einzulegen und eine Taste zu drücken. Dann geht's mit dem Datenausdruck weiter.

Der Ausdruck erfolgt in NLQ-Qualität und natürlich auch mit deutschen Sonderzeichen. Die Steuer codes gelten für den Matrixdrucker Schneider DMP 2000.

So, damit wäre die Bedienung des Programms erklärt. Zum Schluß noch ein Hinweis: Wie Sie sicherlich nach Start des Pro-

gramms bemerkt haben, läuft im Hintergrund eine Uhr mit. Diese Uhr ist wohl für die meisten sehr nützlich. Allerdings hat Sie auch einen kleinen Nachteil: Sie verlangsamt die Programmausführung etwas. Wenn Sie die Uhr nicht benötigen, so können Sie die Uhrsteuerung folgendermaßen ausschalten:

– Laden Sie das Programm durch Eingabe von LOAD »PRO-DATA.BAS«. – Jetzt geben Sie ein: DELETE 5970 – 6000.

– Zum Schluß sichern Sie diese Version noch durch Eingabe von SAVE »PRODATA2.BAS«.

In Zukunft steht Ihnen dann eine Programmversion ohne Uhr zur Verfügung, die Sie durch Eingabe von RUN »PRODATA2.BAS« starten können.

Die Programm-Struktur

Das Hauptprogramm beginnt in Zeile 5640. Von Zeile 5650 – 5950 werden Vorbereitungen, wie Variablen-Besetzung, Menüpunkte einlesen und ähnliches getroffen. In Zeile 5970 bis 5990 wird die aktuelle Uhrzeit eingelesen und ein entsprechender Interrupt eingerichtet. Nun folgt die Hauptschleife (Zeile 6010 – 6100). Diese ruft ein Unterprogramm zur Ausgabe des Hauptmenues auf, dann steuert es die Wahl des Menüpunktes und schließlich ruft es nach Anwahl eines Menüpunktes das entsprechende Unterprogramm auf. Diese Hauptschleife wird erst verlassen, wenn der Benutzer den Menüpunkt »Ende« im Hauptmenue angewählt hat.

Fast alle Eingaben des Benutzers werden von einer eigenen Eingabe-Routine entgegengenommen (Unterprogramm Zeile 1170 – 1420). Diese hat im Vergleich zum INPUT-Befehl des BASIC-Interpreters den Vorteil, daß man nicht ziellos mit dem Cursor auf dem Bildschirm »herumfahren« kann und daß man die maximal mögliche Länge der Eingabe bestimmen kann (Variable laenge). Diese Routine wird auch bei Dateneingaben des Benutzers im Menüpunkt »Blättern/Editieren« benutzt. Damit aber dann auch die Sonderfunktionen dieses Menüpunktes zur Verfügung stehen (Wechseln des Datenfeldes durch <Cursor hoch> und <Cursor ab>, Suchen durch CTRL S, usw.), muß das Unterprogramm wissen, ob einfach nur ein String eingelesen werden soll oder ob man sich derzeit im Menüpunkt »Blättern/Editieren« befindet. Dazu wird die Variable e benutzt. Ist e=1, so befindet man sich im Menüpunkt »Blättern/Editieren« und es wird in Zeile 1300 geprüft, ob eine Sondertaste gedrückt wurde und gegebenenfalls entsprechend reagiert.

Übrigens kann man dem Programm eine eigene Überschrift des Hauptmenues beibringen. Dazu ändert man in Zeile 5710 den Inhalt der Variable titel\$. In der auf Diskette abgespeicherten Version sind 100 Datensätze möglich. Da die mögliche Anzahl der Datenfelder zwischen 1 und 10 schwanken kann, ist es möglich, daß die vom Speicherplatz her mögliche Anzahl der Datensätze größer oder kleiner als 100 ist. Deswegen kann man in Zeile 5690 die maximal mögliche Anzahl von Datensätzen bestimmen. Dazu dient die Variable maxzahl. Das soll zur Erklärung des Programms genügen.

Sind noch Fragen offen geblieben, so wenden Sie sich an die Redaktion. Diese wird Ihre Fragen an mich weiterleiten. Somit wünschen wir Ihnen viel Spaß mit dem Programm und hoffen, daß Sie Ihre Daten damit gut unterbringen können.

(Th. Dietl/JB)

für 464-664-6128



```

1000 'Pro-Data [1063]
1010 'Datenverwaltung [1227]
1020 'copyright by Torsten Dietl [1288]
1030 'im Juni 1987 [531]
1040 ' [117]
1050 ' [117]
1060 GOTO 5640 [403]
1070 : [174]
1080 : [174]
1090 'Uhr [500]
1100 sec=sec+1 [561]
1110 IF sec=60 THEN sec=0:mi=mi+1:IF mi=60 [5397]
    THEN mi=0:st=st+1:IF st=24 THEN st=0
1120 LOCATE #1,72,3 [745]
1130 PRINT #1,USING"###:###:##";st;mi;sec; [1042]
1140 RETURN [555]
1150 : [174]
1160 : [174]
1170 'Prozedur,zur Eingabe eines Strings [2419]
1180 'Eingabeparameter: laenge,beep [1783]
1190 IF laenge=0 THEN laenge=80 [2518]
1200 IF e<>1 THEN eingabe$="" [2030]
1210 a$="T" [419]
1220 WHILE NOT a$="" :a$=INKEY$:WEND [2025]
1230 zeiger=1 [487]
1240 x=POS(#0):y=VPOS(#0) [1340]
1250 CALL &BBB1 [318]
1260 : [174]
1270 WHILE NOT a$=CHR$(13) [2119]
1280 a$=INKEY$ [278]
1290 IF a$="" THEN 1390 [904]
1300 IF e=1 THEN IF a$=CHR$(240) OR a$=C [8226]
    HR$(241) OR a$=CHR$(244) OR
    a$=CHR$(245) OR a$=CHR$(252) OR a$=CHR$(
    19) OR a$=CHR$(7) OR a$=CHR$(12) T
    HEN GOSUB 3320
1310 IF a$=CHR$(242) THEN IF zeiger>1 TH [2328]
    EN zeiger=zeiger-1
1320 IF a$=CHR$(243) THEN IF LEN(eingabe [3976]
    $)-zeiger>-1 THEN zeiger=zeiger+1
1330 IF a$=CHR$(16) THEN IF LEN(eingabe$ [6977]
    )-zeiger>-1 THEN
    eingabe$=LEFT$(eingabe$,zeiger-1)+MID$(
    eingabe$,zeiger+1)
1340 IF a$=CHR$(127) THEN IF zeiger>1 TH [7763]
    EN
    eingabe$=LEFT$(eingabe$,zeiger-2)+MID$(
    eingabe$,zeiger):zeiger=zeiger-1
1350 IF a$=CHR$(250) THEN zeiger=1 [1003]
1360 IF a$=CHR$(251) THEN zeiger=LEN(ein [3928]
    gabe$)+1
1370 IF ASC(a$)>31 AND ASC(a$)<127 THEN [11489]
    IF LEN(eingabe$)<laenge THEN
    eingabe$=LEFT$(eingabe$,zeiger-1)+a$+MI
    D$(eingabe$,zeiger):
    eiger=zeiger+1
1380 CALL &BBB4:LOCATE x,y:PRINT eingabe [4130]
    $" " ;:LOCATE x+zeiger-1,y:CALL &BBB1
1390 WEND [390]
1400 CALL &BBB4 [315]
1410 PRINT [361]
1420 RETURN [555]
1430 : [174]
1440 : [174]
1450 'Deutscher Zeichensatz [1352]
1460 SYMBOL AFTER 91 [1213]
1470 SYMBOL 91,&66,&18,&3C,&66,&7E,&66,&66 [2047]
    ,&0 'AE
1480 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C [2360]
    ,0 'OE
1490 SYMBOL 93,&66,&44,&66,&66,&66,&66,&3C [2599]
    ,0 'UE
1500 SYMBOL 123,&CC,0,&78,&C,&7C,&CC,&76,0 [2451]
    ,ae
1510 SYMBOL 124,&66,0,&3C,&66,&66,&66,&3C, [2403]
    0 'oe
1520 SYMBOL 125,&66,0,&66,&66,&66,&66,&3E, [2014]

```

Listing Prodata

```

0 'ue
1530 SYMBOL 126,&3C,&66,&66,&6C,&66,&66,&6 [2854]
    C,&60 'scharfes s
1540 KEY DEF 17,1,125,93 [746]
1550 KEY DEF 19,1,123,91 [1140]
1560 KEY DEF 26,1,124,92,64 [765]
1570 KEY DEF 24,1,126,94,0 [1480]
1580 RETURN [555]
1590 : [174]
1600 : [174]
1610 'Menue aufbauen [1861]
1620 CLS #1 [373]
1630 CLS [91]
1640 DI [84]
1650 LOCATE #1,INT((80-LEN(titel$))/2),2 [2275]
1660 PRINT #1,titel$ [725]
1670 EI [86]
1680 FOR i=1 TO 11 [314]
1690 LOCATE mx(i),my(i) [1695]
1700 PRINT menue$(i) [533]
1710 NEXT i [375]
1720 LOCATE mx(menuezeiger)-2,my(menuezeig [4324]
    er)
1730 PRINT CHR$(243); [1201]
1740 MOVE 0,160:DRAW 639,160:LOCATE 6,16 [1415]
1750 IF m=0 THEN PRINT "Keine "; [1800]
1760 PRINT "Maske im Speicher ";:IF m=1 TH [4549]
    EN PRINT CHR$(1);CHR$(6)
1770 LOCATE 6,18 [520]
1780 IF d=0 THEN PRINT "Keine "; [1781]
1790 PRINT "Daten im Speicher ";:IF d=1 TH [4573]
    EN PRINT CHR$(1);CHR$(6)
1800 RETURN [555]
1810 : [174]
1820 : [174]
1830 'Menue-Steuerung [2164]
1840 LOCATE mx(menuezeiger)-2,my(menuezeig [4324]
    er)
1850 PRINT " "; [673]
1860 IF a$=CHR$(240) THEN IF (menuezeiger> [10269]
    1 AND menuezeiger<4) OR (menuezeiger>4 AND
    menuezeiger<8) OR menuezeiger>8 THEN menu
    ezeiger=menuezeiger-1
1870 IF a$=CHR$(241) THEN IF menuezeiger<3 [7799]
    OR (menuezeiger>3 AND menuezeiger<7)
    OR (menuezeiger>7 AND menuezeiger<11) THE
    N menuezeiger=menuezeiger+1
1880 IF a$=CHR$(242) THEN IF menuezeiger>3 [5233]
    THEN IF menuezeiger>7 THEN
    menuezeiger=4 ELSE menuezeiger=1
1890 IF a$=CHR$(243) THEN IF menuezeiger<8 [5354]
    THEN IF menuezeiger<4 THEN
    menuezeiger=4 ELSE menuezeiger=8
1900 LOCATE mx(menuezeiger)-2,my(menuezeig [4324]
    er)
1910 PRINT CHR$(243); [1201]
1920 RETURN [555]
1930 : [174]
1940 : [174]
1950 'Warte auf Tastendruck [1863]
1960 CLS #2 [372]
1970 PRINT #2,TAB(30)"Bitte Taste dr]cken" [2160]
1980 WHILE INKEY$="" :WEND [1607]
1990 CLS #2 [372]
2000 RETURN [555]
2010 : [174]
2020 : [174]
2030 'Maske definieren [1777]
2040 IF d=1 THEN RETURN [876]
2050 CLS #1 [373]
2060 DI [84]
2070 LOCATE #1,32,2 [902]
2080 PRINT #1,"Maske definieren" [2028]
2090 EI [86]
2100 hi=azdf:azdf=-1:laenge=2 [1314]
2110 WHILE azdf<0 OR azdf>10 [1950]
2120 CLS [91]
2130 LOCATE 1,2 [614]
2140 PRINT "Anzahl Datenfelder ? "; [2654]
2150 GOSUB 1170 [817]
2160 azdf=VAL(eingabe$) [749]
2170 WEND [390]
2180 IF azdf=0 THEN azdf=hi:RETURN [1303]
2190 : [174]
2200 laenge=40 [564]
2210 FOR i=1 TO azdf [694]
2220 PRINT [361]
2230 PRINT USING"Name des ##.Datenfeldes [2497]
    ? ";i;

```

Listing Prodata


```

2240 GOSUB 1170 [817]
2250 feldname$(i)=eingabe$ [1608]
2260 IF i=1 THEN maxlaenge=LEN(eingabe$) [4339]
ELSE maxlaenge=MAX(maxlaenge,LEN
(eingabe$))
2270 NEXT i [375]
2280 : [174]
2290 PRINT [361]
2300 PRINT "Alles richtig (j/n) ?" [1946]
2310 a$="" [388]
2320 WHILE NOT(a$="J" OR a$="N") [1955]
2330 a$=UPPER$(INKEY$) [789]
2340 WEND [390]
2350 IF a$="N" THEN GOTO 2030 [952]
2360 datenlaenge=77-maxlaenge [1375]
2370 m=1 [184]
2380 WINDOW #3,maxlaenge+3,80,4,24 [1931]
2390 PRINT CHR$(7); [1175]
2400 RETURN [555]
2410 : [174]
2420 : [174]
2430 'Eingabe Dateiname [1177]
2440 CLS:PRINT [356]
2450 PRINT "Dateiname:"; [2206]
2460 laenge=12 [608]
2470 GOSUB 1170 [817]
2480 posi=INSTR(eingabe$,".") [1775]
2490 IF (posi=0 AND LEN(eingabe$)>8) OR po [3228]
si=1 THEN 2430
2500 IF posi>9 THEN 2430 [1111]
2510 IF posi>0 AND LEN(MID$(eingabe$,posi+ [3272]
1))>3 THEN 2430
2520 name$=eingabe$ [440]
2530 RETURN [555]
2540 : [174]
2550 : [174]
2560 'Maske speichern [2498]
2570 IF m=0 THEN RETURN [729]
2580 CLS #1 [373]
2590 DI [84]
2600 LOCATE #1,32,2 [902]
2610 PRINT #1,"Maske speichern" [2131]
2620 EI [86]
2630 GOSUB 2430 [861]
2640 IF name$="" THEN RETURN [716]
2650 OPENOUT name$ [700]
2660 PRINT #9,azdf [1131]
2670 FOR i=1 TO azdf [694]
2680 PRINT #9,feldname$(i) [1371]
2690 NEXT i [375]
2700 PRINT #9,maxlaenge [1743]
2710 PRINT #9,datenlaenge [1595]
2720 CLOSEOUT [902]
2730 RETURN [555]
2740 : [174]
2750 : [174]
2760 'Maske laden [1530]
2770 IF d=1 THEN RETURN [876]
2780 CLS #1 [373]
2790 DI [84]
2800 LOCATE #1,34,2 [894]
2810 PRINT #1,"Maske laden" [2026]
2820 EI [86]
2830 GOSUB 2430 [861]
2840 IF name$="" THEN RETURN [716]
2850 OPENIN name$ [1050]
2860 INPUT #9,azdf [1254]
2870 FOR i=1 TO azdf [694]
2880 LINE INPUT #9,feldname$(i) [2193]
2890 NEXT i [375]
2900 INPUT #9,maxlaenge [1282]
2910 INPUT #9,datenlaenge [2055]
2920 WINDOW #3,maxlaenge+3,80,4,24 [1931]
2930 CLOSEIN [752]
2940 m=1 [184]
2950 RETURN [555]
2960 : [174]
2970 : [174]
2980 'Eingaben / Bl{ttern [1128]
2990 IF m=0 THEN RETURN [729]
3000 e=1:beendet=false:laenge=datenlaenge [2822]
3010 CLS #1 [373]
3020 CLS [91]
3030 DI [84]
3040 LOCATE #1,30,2 [942]
3050 PRINT #1,"Eingaben / Bl{ttern" [2043]
3060 EI [86]

```

Listing Prodata

```

3070 FOR i=1 TO azdf [694]
3080 PRINT [361]
3090 PRINT STRING$(maxlaenge-LEN(feldnam [4965]
e$(i)),32);feldname$(i);":
3100 NEXT i [375]
3110 : [174]
3120 satz=1:feld=1 [1387]
3130 eingabe$=daten$(1,1) [1311]
3140 GOSUB 3470 [987]
3150 IF anzahl=0 THEN anzahl=1 [584]
3160 f$=" SHI "+CHR$(240)+" SHI "+CHR$ [7505]
(241)+" Suchen-CTRL S Gehezu-CTRL G
L:schen-CTRL L ENDE-ESC"
3170 PRINT #2,f$ [520]
3180 LOCATE maxlaenge+3,2 [1277]
3190 : [174]
3200 WHILE NOT beendet [2874]
3210 GOSUB 1170 [817]
3220 IF NOT beendet THEN IF feld<azdf TH [9722]
EN daten$(satz,feld)=eingabe$:
feld=feld+1:eingabe$=daten$(satz,feld):
LOCATE maxlaenge+3,feld*2 E
LSE LOCATE maxlaenge+3,feld*2
3230 WEND [390]
3240 : [174]
3250 e=0 [340]
3260 IF daten$(anzahl,1)="" THEN anzahl=an [1683]
zahl-1
3270 IF anzahl>0 THEN d=1 [1106]
3280 CLS #2 [372]
3290 RETURN [555]
3300 : [174]
3310 : [174]
3320 'Neues Feld oder neuer Satz angew{hit [1409]
3330 CALL &BB84:vd=0 [958]
3340 daten$(satz,feld)=eingabe$ [1653]
3350 IF a$=CHR$(240) THEN IF feld>1 THEN f [9007]
eld=feld-1:eingabe$=daten$(satz,feld):
zeiger=1:x=maxlaenge+3:y=2:LOCATE x,fel
d*2
3360 IF a$=CHR$(241) THEN IF feld<azdf THE [6564]
N feld=feld+1:eingabe$=daten$(satz,
feld):zeiger=1:x=maxlaenge+3:y=y+2:LOCATE
x,feld*2
3370 IF a$=CHR$(244) THEN IF satz>1 THEN s [5752]
atz=satz-1:GOSUB 3470:feld=1:eingabe$=
daten$(satz,feld):zeiger=1:x=maxlaenge+3:
y=2:LOCATE x,y
3380 IF a$=CHR$(245) THEN IF daten$(satz,1 [15803]
)<>" AND satz<maxzahl THEN satz=satz
+1:GOSUB 3470:feld=1:eingabe$=daten$(satz
,feld):x=maxlaenge+3:y=2:LOCATE x,y:z
eiger=1:IF daten$(satz,1)="" THEN anzahl=s
atz
3390 IF a$=CHR$(252) THEN a$=CHR$(13):been [2990]
det=true
3400 IF a$=CHR$(19) THEN GOSUB 3600 [1342]
3410 IF a$=CHR$(7) THEN IF anzahl>0 THEN n [12495]
=0:WHILE n<1 OR n>anzahl:INPUT #2,
"Datensatz-Nummer:",n:WEND:satz=n:GOSUB 3
470:feld=1:eingabe$=daten$(satz,fel
d):x=maxlaenge+3:y=2:PRINT #2,f$:zeiger=1:
LOCATE x,y
3420 IF a$=CHR$(12) THEN IF daten$(satz,1) [13221]
<>" THEN FOR i=satz TO anzahl-1:
FOR n=1 TO azdf:daten$(i,n)=daten$(i+1,n)
:NEXT n,i:FOR i=1 TO azdf:daten$ (an
zahl,i)="" :NEXT i:GOSUB 3470:zeiger=1:feld
=1:x=maxlaenge+3:y=2:vd=1
3430 IF a$=CHR$(12) THEN IF vd=1 THEN eing [5710]
abe$=daten$(satz,1):LOCATE x,y:IF
anzahl>1 THEN anzahl=anzahl-1
3440 RETURN [555]
3450 : [174]
3460 : [174]
3470 'Datensatz printen [1754]
3480 CLS #3 [371]
3490 FOR i=1 TO azdf [694]
3500 LOCATE maxlaenge+3,i*2 [1173]
3510 PRINT daten$(satz,i) [1972]
3520 NEXT i [375]
3530 DI [84]
3540 LOCATE #1,2,3 [805]
3550 PRINT #1,"Datensatz Nr. ";satz; [1807]
3560 EI [86]
3570 RETURN [555]
3580 : [174]
3590 :

```

Listing Prodata

Anwendungen

```

[174]
3600 'Suchen [357]
3610 INPUT #2,"Gesuchter Ausdruck:",text$ [2710]
3620 IF text$="" THEN PRINT #2,f$:RETURN [2149]
3630 IF LEN(text$)>datenlaenge THEN 3610 [2972]
3640 PRINT #2,"In allen Feldern suchen ode [5794]
r nur in einem bestimmten (1/2) ?"
3650 a$="" [388]
3660 WHILE NOT (a$="1" OR a$="2") [1748]
3670 a$=INKEY$ [278]
3680 WEND [390]
3690 suchart=VAL(a$) [868]
3700 a$=CHR$(19) [527]
3710 IF suchart=2 THEN PRINT #2,USING"Welc [6541]
hes Feld (1-##) ? ";azdf;:INPUT #2,"",
feldnummer:IF feldnummer<1 OR feldnummer>
azdf THEN 3710
3720 : [174]
3730 'Jetzt wird gesucht [2302]
3740 gefunden=false [1817]
3750 i=1:s=0 [405]
3760 WHILE i<=anzahl AND gefunden=false [2625]
3770 IF suchart=1 THEN n=1:WHILE n<=azdf [6819]
AND s=0:s=INSTR(daten$(i,n),text$):
n=n+1:WEND ELSE s=INSTR(daten$(i,feldnu
mmer),text$)
3780 IF s<>0 THEN gefunden=true [2655]
3790 i=i+1 [444]
3800 WEND [390]
3810 : [174]
3820 IF s=0 THEN PRINT #2,"";text$;"' nic [10157]
ht gefunden! Taste dr)cken!":
WHILE INKEY$="" :WEND:PRINT #2,f$:RETURN
3830 i=i-1:n=n-1 [676]
3840 satz=i:IF suchart=1 THEN feld=n ELSE [4410]
feld=feldnummer
3850 zeiger=1 [487]
3860 eingabe$=daten$(satz,feld) [1784]
3870 GOSUB 3470 [987]
3880 x=maxlaenge+3:y=feld*2 [2490]
3890 LOCATE x,y [672]
3900 PRINT #2,f$ [520]
3910 RETURN [555]
3920 : [174]
3930 : [174]
3940 'Sortieren [223]
3950 IF d=0 THEN RETURN [669]
3960 CLS #1 [373]
3970 CLS [91]
3980 DI [84]
3990 LOCATE #1,32,2 [902]
4000 PRINT #1,"Daten sortieren" [1499]
4010 EI [86]
4020 PRINT [361]
4030 PRINT USING"Nach welchem Feld sortier [3780]
en (1-##) ? ";azdf;
4040 laenge=2 [548]
4050 GOSUB 1170 [817]
4060 feldnummer=VAL(eingabe$) [1510]
4070 IF feldnummer<1 OR feldnummer>azdf TH [2670]
EN 4030
4080 CLS [91]
4090 PRINT [361]
4100 PRINT "Sortieren nach ";feldname$(fel [2852]
dnummer)
4110 PRINT [361]
4120 PRINT "Bitte warten - Daten werden so [4764]
rtiert !"
4130 PRINT [361]
4140 PRINT "Abbruch mit ESC m'glich !" [2763]
4150 : [174]
4160 getauscht=true [1545]
4170 i=1 [423]
4180 WHILE i<=anzahl AND getauscht=true [2960]
4190 getauscht=false [1781]
4200 FOR n=1 TO anzahl-1 [1177]
4210 tauschen=false [1169]
4220 text1$=LOWER$(daten$(n,feldnummer [1475]
))
4230 text2$=LOWER$(daten$(n+1,feldnumm [2977]
er))
4240 IF text1$="" AND text2$<>"" THEN [3936]
tauschen=true:GOTO 4390
4250 IF text2$="" THEN 4390 [1311]
4260 l1=LEN(text1$) [809]
4270 l2=LEN(text2$) [944]
4280 unterschied=false [1244]
4290 q=1 [264]
4300 WHILE unterschied=false AND q<=MI [3656]

```

Listing Prodata

```

N(11,12)
4310 IF INKEY$=CHR$(252) THEN RETURN [1391]
4320 w1=ASC(MID$(text1$,q,1)) [588]
4330 w2=ASC(MID$(text2$,q,1)) [1953]
4340 IF w1<w2 THEN unterschied=true [1487]
4350 IF w2<w1 THEN unterschied=true: [4905]
tauschen=true
4360 q=q+1 [50]
4370 WEND [390]
4380 IF unterschied=false THEN IF l2<l [5659]
1 THEN tauschen=true
4390 IF tauschen=true THEN FOR k=1 TO [8853]
azdf:hilfe$(k)=daten$(n,k):daten$(n,k)
=daten$(n+1,k):daten$(n+1,k)=hilfe$(k
):NEXT k:getauscht=true
4400 NEXT n [366]
4410 i=i+1 [444]
4420 WEND [390]
4430 PRINT CHR$(7); [1175]
4440 RETURN [555]
4450 : [174]
4460 : [174]
4470 'Daten loeschen [1405]
4480 IF d=0 THEN RETURN [669]
4490 PRINT #2,TAB(22)"Wirklich alle Daten [6589]
l'schen (j/n) ?"
4500 a$="" [388]
4510 WHILE NOT(a$="N" OR a$="J") [1988]
4520 a$=UPPER$(INKEY$) [789]
4530 WEND [390]
4540 IF a$="J" THEN PRINT #2,TAB(21)"Bitte [9519]
warten - Daten werden gel'scht !":
FOR i=1 TO anzahl:FOR n=1 TO azdf:daten$(
i,n)="" :NEXT n,i:anzahl=0:d=0
4550 CLS #2 [372]
4560 RETURN [555]
4570 : [174]
4580 : [174]
4590 'Ende [715]
4600 ende=true [431]
4610 RETURN [555]
4620 : [174]
4630 : [174]
4640 'Daten speichern [1379]
4650 IF d=0 THEN RETURN [669]
4660 CLS #1 [373]
4670 DI [84]
4680 LOCATE #1,32,2 [902]
4690 PRINT #1,"Daten speichern" [1463]
4700 EI [86]
4710 GOSUB 2430 [861]
4730 IF name$="" THEN RETURN [716]
4740 OPENOUT name$ [700]
4750 PRINT #9,anzahl [1596]
4760 FOR i=1 TO anzahl [1073]
4770 FOR n=1 TO azdf [811]
4780 PRINT #9,daten$(i,n) [1299]
4790 NEXT n [366]
4800 NEXT i [375]
4810 CLOSEOUT [902]
4820 RETURN [555]
4830 : [174]
4840 : [174]
4850 'Daten laden [1528]
4860 IF m=0 THEN RETURN [729]
4870 CLS #1 [373]
4880 DI [84]
4890 LOCATE #1,34,2 [894]
4900 PRINT #1,"Daten laden" [988]
4910 EI [86]
4920 GOSUB 2430 [861]
4940 IF name$="" THEN RETURN [716]
4950 OPENIN name$ [1050]
4960 INPUT #9,anzahl [934]
4970 FOR i=1 TO anzahl [1073]
4980 FOR n=1 TO azdf [811]
4990 LINE INPUT #9,daten$(i,n) [1732]
5000 NEXT n [366]
5010 NEXT i [375]
5020 CLOSEIN [752]
5030 d=1 [339]
5040 RETURN [555]
5050 : [174]
5060 : [174]
5070 'Directory [889]
5080 CLS #1 [373]
5090 CLS [91]
5100 DI [84]
5110 LOCATE #1,26,2 [799]

```

Listing Prodata

```

5120 PRINT #1,"Inhaltsverzeichnis Diskette [2759]
"
5130 EI [86]
5140 CAT [96]
5150 GOSUB 1950 [853]
5160 RETURN [373]
5170 : [174]
5180 : [174]
5190 'Drucken [1548]
5200 IF d=0 THEN RETURN [669]
5210 CLS #1 [373]
5220 CLS [91]
5230 DI [84]
5240 LOCATE #1,33,2 [938]
5250 PRINT #1,"Daten drucken" [962]
5260 EI [86]
5270 PRINT [361]
5280 PRINT "Schalte Drucker ein und dr}cke
Taste !" [3282]
5290 WHILE INKEY$="" :WEND [1607]
5300 PRINT [361]
5310 PRINT "]berschrift des Datenausdrucks [2649]
:"
5320 laenge=78:GOSUB 1170 [1184]
5330 ueberschrift$=eingabe$ [1013]
5340 PRINT:PRINT "Soll die Maske mit ausgabe
druckt werden (J/N) ? "; [4144]
5350 a$="" [388]
5360 WHILE NOT (a$="N" OR a$="J"):a$=UPPER
$(INKEY$):WEND [4104]
5370 IF a$="N" THEN PRINT "Nein":druckart= [4568]
0 ELSE PRINT "Ja":druckart=1
5380 : [174]
5390 WIDTH 80 [858]
5400 az=62 [157]
5410 azs=INT(az/(azdf+1)) [1646]
5420 PRINT #8,CHR$(27);"R";CHR$(2); [1898]
5430 PRINT #8,CHR$(27);"x";CHR$(1); [1737]
5440 PRINT #8,CHR$(27);"-";CHR$(1); [2015]
5450 PRINT #8,ueberschrift$ [1257]
5460 PRINT #8,CHR$(27);"-";CHR$(0); [2019]
5470 PRINT #8 [343]
5480 : [174]
5490 n=1:s=1 [275]
5500 FOR i=1 TO anzahl [1073]
5510 FOR q=1 TO azdf [1358]
5520 IF druckart=1 THEN PRINT #8,STRIN [3752]
G$(maxlaenge-LEN(feldname$(q)),32);
feldname$(q);": ";
5530 PRINT #8,daten$(i,q) [1363]
5540 NEXT q [367]
5550 PRINT #8 [343]
5560 IF n=azs THEN PRINT:PRINT "Wenn der
Drucker fertig ist, lege neues Blatt ein
und dr}cke Taste!":WHILE INKEY$="" :WEND:n=
0:s=s+1:PRINT #8,TAB(70);"Seite";s:PRINT #
8
5570 n=n+1 [140]
5580 NEXT i [375]
5590 : [174]
5600 PRINT CHR$(7); [1175]
5610 RETURN [555]
5620 : [174]
5630 : [174]
5640 'Hauptprogramm [1580]
5650 MODE 2 [513]
5660 CALL &BB48 [383]
5670 IF HIMEM>42618 THEN GOSUB 1450:OPENOU [4278]
T "d":MEMORY HIMEM-1:CLOSEOUT
5680 BORDER 0:INK 0,0:INK 1,24 [1350]
5690 maxxahl=100:mx=32:menuezeiger=1 [1867]
5700 true=-1:false=0:ende=false:anzahl=0 [2492]
5710 titel$="Datenverwaltung":m=0:d=0:e=0 [1897]
5720 DIM daten$(maxzahl,10) [1606]
5730 DIM feldname$(10) [506]
5740 DIM mx(11),my(11),menue$(11) [1245]
5750 DATA Maske definieren,6,3 [1075]
5760 DATA Maske speichern,6,5 [1648]
5770 DATA Maske laden,6,7 [513]
5780 DATA Eingabe / Bl{ttern,32,3 [1443]
5790 DATA Daten sortieren,32,5 [1451]
5800 DATA "Daten l{schen",32,7 [1121]
5810 DATA Ende,32,9 [713]
5820 DATA Daten speichern,60,3 [1154]
5830 DATA Daten laden,60,5 [1111]
5840 DATA Directory,60,7 [1453]
5850 DATA Daten drucken,60,9 [1168]
5860 : [174]
5870 FOR i=1 TO 11 [314]

```

```

5880 READ menue$(i) [929]
5890 READ mx(i),my(i) [714]
5900 NEXT i [375]
5910 : [174]
5920 WINDOW #0,1,80,4,24 [1114]
5930 WINDOW #1,1,80,1,3 [1001]
5940 WINDOW #2,1,80,25,25 [1376]
5950 PAPER #1,1:PEN #1,0 [1741]
5960 : [174]
5970 INPUT "Bevor Sie ins Hauptmenue gelan [9356]
gen, geben Sie bitte die aktuelle Uhrzeit
ein: ",st,mi,sec
5980 IF st<0 OR st>23 OR mi<0 OR mi>59 OR [3927]
sec<0 OR sec>59 THEN 5970
5990 EVERY 50,3 GOSUB 1090 [1046]
6000 : [174]
6010 'Hauptschleife [866]
6020 WHILE NOT ende [2055]
6030 GOSUB 1610 'Menue [1391]
6040 a$="" [388]
6050 WHILE NOT (a$=CHR$(224) OR a$=CHR$( [3169]
13))
6060 a$=INKEY$ [278]
6070 IF a$=CHR$(240) OR a$=CHR$(241) O [3562]
R a$=CHR$(242) OR a$=CHR$(243) THEN
GOSUB 1830
6080 WEND [390]
6090 ON menuezeiger GOSUB 2030,2560,2760 [4568]
,2980,3940,4470,4590,4640,4850,5070
,5190
6100 WEND [390]
6110 : [174]
6120 a=REMAIN(3) [1302]
6130 MODE 2 [513]
6140 END [110]

```



Drucken - aber richtig!

Dieses Programm ist für all diejenigen gedacht, die einen Riteman F+ oder Riteman Super F+ besitzen und Ihren Drucker mit menuegesteuerter Software einstellen möchten.

Menuepunkt 1 erlaubt es Ihnen, einen der insgesamt 8 internationalen Zeichensätze anzuwählen. Das Programm zeigt Ihnen an, welche DIP-Schalter an der Rückseite auf welche Position zu bringen sind. Dies geschieht, indem Sie den Drucker ausschalten, die DIP-Schalter nach Anweisung einstellen und danach den Drucker wieder einschalten. Dann drücken Sie ein-

fach eine Taste, damit der Drucker den von Ihnen eingestellte Zeichensatz anerkennt.

Mit Menuepunkt 2 verändern Sie den Zeilenabstand. Wählen Sie in diesem Untermenue die Punkte 4 oder 5 an, so müssen Sie eine Zahl eingeben. Diese darf bei Punkt 4 nicht größer als 255 und bei Punkt 5 nicht größer als 85 sein.

In Menuepunkt 3 wird ein Untermenue aufgerufen, mit dem Sie verschiedene Schriftarten ausprobieren können. Diese lassen sich teilweise auch miteinander kombinieren.

Wählen Sie Menuepunkt 4, dann wird der Drucker initialisiert, d.h. er befindet sich wieder im gleichen Zustand wie nach dem Einschalten.

(K. Molzahn/JB)

Für 464-664-6128



```

10 REM ***** [1287]
20 REM * PRINTFIX * [1013]
30 REM * by Karsten Molzahn * [1222]
40 REM ***** [1287]
50 REM Aufforderung zum Druckeranschalten [2566]
60 REM ----- [2076]
70 MODE 1:BORDER 0:INK 0,0:INK 1,26 [1688]
80 PEN 2:LOCATE 12,12:PRINT"Drucker einsch [3379]
alten !"
90 PEN 1:LOCATE 18,25:PRINT"Taste":CALL &B [1693]
B18
100 REM Hauptmenue [1328]
110 REM ----- [847]
120 MODE 1:BORDER 7 [1433]
130 LOCATE 15,2:PRINT"Hauptmenue" [1876]
140 LOCATE 15,3:PRINT"-----" [1632]
150 LOCATE 4,6:PRINT"1. Zeichensatz aender [2666]
n"
160 LOCATE 4,8:PRINT"2. Zeilenabstand aend [4320]
ern"
170 LOCATE 4,10:PRINT"3. Schriftart aender [2894]
n"
180 LOCATE 4,12:PRINT"4. Drucker initialis [4725]
ieren"
190 LOCATE 4,14:PRINT"5. Programmende" [4004]
200 LOCATE 7,20:PEN 3:PRINT"Auswahl ?":PEN [3787]
1
210 a$=INKEY$:IF a$="" THEN 210 ELSE a$=UP [2281]
PER$(a$)
220 IF a$="1" THEN GOTO 280 [1270]
230 IF a$="2" THEN GOTO 1250 [1140]
240 IF a$="3" THEN GOTO 1580 [1692]
250 IF a$="4" THEN CLS:PRINT#8,CHR$(27);"● [2206]
";:GOTO 100
260 IF a$="5" THEN GOTO 2000 [1424]
270 GOTO 210 [427]
280 REM Zeichensatz aendern [1469]
290 REM ----- [1255]
300 MODE 1:LOCATE 10,2:PRINT"Zeichensatz a [4734]
endern"
310 LOCATE 10,3:PRINT"-----" [2329]
320 LOCATE 4,6:PRINT"1. USA" [1308]
330 LOCATE 4,7:PRINT"2. Frankreich" [1328]
340 LOCATE 4,8:PRINT"3. Deutschland" [2584]
350 LOCATE 4,9:PRINT"4. Grossbritannien" [3186]

360 LOCATE 4,10:PRINT"5. Daenemark" [1447]
370 LOCATE 4,11:PRINT"6. Schweden" [1597]
380 LOCATE 4,12:PRINT"7. Italien" [2304]
390 LOCATE 4,13:PRINT"8. Spanien" [1628]
400 LOCATE 4,14:PRINT"9. Hauptmenue" [2768]
410 LOCATE 7,20:PEN 3:PRINT"Auswahl ?":PEN [3787]
1
420 b$=INKEY$:IF b$="" THEN 420 ELSE b$=UP [3432]
PER$(b$)
430 IF b$="1" THEN 530 [959]
440 IF b$="2" THEN 620 [940]
450 IF b$="3" THEN 710 [693]
460 IF b$="4" THEN 800 [787]
470 IF b$="5" THEN 890 [322]

```

Listing: Drucken - aber richtig

```

480 IF b$="6" THEN 980 [529]
490 IF b$="7" THEN 1070 [1143]
500 IF b$="8" THEN 1160 [1139]
510 IF b$="9" THEN 100 [1404]
520 GOTO 420 [401]
530 REM Zeichensatz USA [1443]
540 CLS:LOCATE 2,3:PRINT"Zeichensatz USA" [2300]
550 LOCATE 2,4:PRINT"-----" [2312]
560 LOCATE 4,6:PRINT"DIP-Schalter umstelle [3277]
n!"
570 LOCATE 4,8:PRINT"DIP-Schalter 1-1 O [3810]
N"
580 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1770]
ON"
590 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1907]
ON"
600 LOCATE 18,25:PRINT"Taste !":CALL &BB18 [2951]
610 PRINT#8,CHR$(27);"R";CHR$(0);:GOTO 280 [2455]
620 REM Zeichensatz Frankreich [2080]
630 CLS:LOCATE 2,3:PRINT"Zeichensatz Frank [3387]
reich"
640 LOCATE 2,4:PRINT"-----" [2588]
--"
650 LOCATE 4,6:PRINT"DIP-Schalter umstelle [3277]
n!"
660 LOCATE 4,8:PRINT"DIP-Schalter 1-1 O [3994]
FF"
670 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1770]
ON"
680 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1907]
ON"
690 LOCATE 18,25:PRINT"Taste !":CALL &BB18 [2951]
700 PRINT#8,CHR$(27);"R";CHR$(1);:GOTO 280 [2451]
710 REM Zeichensatz Deutschland [1432]
720 CLS:LOCATE 2,3:PRINT"Zeichensatz Deuts [3660]
chland"
730 LOCATE 2,4:PRINT"-----" [2696]
--"
740 LOCATE 4,6:PRINT"DIP-Schalter umstelle [3277]
n!"
750 LOCATE 4,8:PRINT"DIP-Schalter 1-1 O [3810]
N"
760 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1930]
OFF"
770 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1907]
ON"
780 LOCATE 18,25:PRINT"Taste !":CALL &BB18 [2951]
790 PRINT#8,CHR$(27);"R";CHR$(2);:GOTO 280 [2014]
800 REM Zeichensatz Grossbritannien [2941]
810 CLS:LOCATE 2,3:PRINT"Zeichensatz Gross [3761]
britannien"
820 LOCATE 2,4:PRINT"-----" [2820]
--"
830 LOCATE 4,6:PRINT"DIP-Schalter umstelle [3277]
n!"
840 LOCATE 4,8:PRINT"DIP-Schalter 1-1 O [3994]
FF"
850 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1930]
OFF"
860 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1907]
ON"
870 LOCATE 18,25:PRINT"Taste !":CALL &BB18 [2951]
880 PRINT#8,CHR$(27);"R";CHR$(3);:GOTO 280 [2490]
890 REM Zeichensatz Daenemark [2214]
900 CLS:LOCATE 2,3:PRINT"Zeichensatz Daene [3624]
mark"
910 LOCATE 2,4:PRINT"-----" [2382]
--"

```

Listing: Drucken - aber richtig

```

920 LOCATE 4,6:PRINT"DIP-Schalter umstelle [3277]
n!"
930 LOCATE 4,8:PRINT"DIP-Schalter 1-1 O [3810]
N"
940 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1770]
ON"
950 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1851]
OFF"
960 LOCATE 18,25:PRINT"Taste !":CALL &BB18 [2951]
970 PRINT#8,CHR$(27);"R";CHR$(4);:GOTO 280 [3078]
980 REM Zeichensatz Schweden [1438]
990 CLS:LOCATE 2,3:PRINT"Zeichensatz Schwe [3543]
den"
1000 LOCATE 2,4:PRINT"----- [2363]
"
1010 LOCATE 4,6:PRINT"DIP-Schalter umstell [3277]
en!"
1020 LOCATE 4,8:PRINT"DIP-Schalter 1-1 [3994]
OFF"
1030 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1770]
ON"
1040 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1851]
OFF"
1050 LOCATE 18,25:PRINT"Taste !":CALL &BB1 [2951]
8
1060 PRINT#8,CHR$(27);"R";CHR$(5);:GOTO 28 [3330]
0
1070 REM Zeichensatz Italien [2106]
1080 CLS:LOCATE 2,3:PRINT"Zeichensatz Ital [2191]
ien"
1090 LOCATE 2,4:PRINT"-----" [2497]
1100 LOCATE 4,6:PRINT"DIP-Schalter umstell [3277]
en!"
1110 LOCATE 4,8:PRINT"DIP-Schalter 1-1 [3810]
ON"
1120 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1930]
OFF"
1130 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1851]
OFF"
1140 LOCATE 18,25:PRINT"Taste !":CALL &BB1 [2951]
8
1150 PRINT#8,CHR$(27);"R";CHR$(6);:GOTO 28 [2474]
0
1160 REM Zeichensatz Spanien [1632]
1170 CLS:LOCATE 2,3:PRINT"Zeichensatz Span [2468]
ien"
1180 LOCATE 2,4:PRINT"-----" [2497]
1190 LOCATE 4,6:PRINT"DIP-Schalter umstell [3277]
en!"
1200 LOCATE 4,8:PRINT"DIP-Schalter 1-1 [3994]
OFF"
1210 LOCATE 4,10:PRINT"DIP-Schalter 1-2 [1930]
OFF"
1220 LOCATE 4,12:PRINT"DIP-Schalter 1-3 [1851]
OFF"
1230 LOCATE 18,25:PRINT"Taste !":CALL &BB1 [2951]
8
1240 PRINT#8,CHR$(27);"R";CHR$(7);:GOTO 28 [2863]
0
1250 REM Zeilenabstand aendern [2523]
1260 REM ----- [1292]
1270 MODE 1:LOCATE 10,2:PRINT"Zeilenabstan [4558]
d aendern"
1280 LOCATE 10,3:PRINT"----- [2341]
--"
1290 LOCATE 4,6:PRINT"1. 1/8 Zoll" [1467]
1300 LOCATE 4,8:PRINT"2. 7/72 Zoll" [3175]
1310 LOCATE 4,10:PRINT"3. 1/6 Zoll" [1602]
1320 LOCATE 4,12:PRINT"4. n/216 Zoll" [2798]
1330 LOCATE 4,14:PRINT"5. n/72 Zoll" [2451]
1340 LOCATE 4,16:PRINT"6. Hauptmenue" [2139]
1350 LOCATE 7,20:PEN 3:PRINT"Auswahl ?":PE [3787]
N 1
1360 c$=INKEY$:IF c$="" THEN 1360 ELSE c$= [2485]
UPPER$(c$)
1370 IF c$="1" THEN PRINT#8,CHR$(27);"0"; [1759]
1380 IF c$="2" THEN PRINT#8,CHR$(27);"1"; [1482]
1390 IF c$="3" THEN PRINT#8,CHR$(27);"2"; [1840]
1400 IF c$="4" THEN 1440 [815]
1410 IF c$="5" THEN 1510 [877]
1420 IF c$="6" THEN 100 [1196]
1430 GOTO 1250 [361]
1440 REM Zeilenabstand n/216 Zoll [2631]
1450 MODE 1:LOCATE 4,2:PRINT"Zeilenabstand [4270]
n/216 Zoll"
1460 LOCATE 4,3:PRINT"----- [2277]

```

Listing: Drucken - aber richtig

```

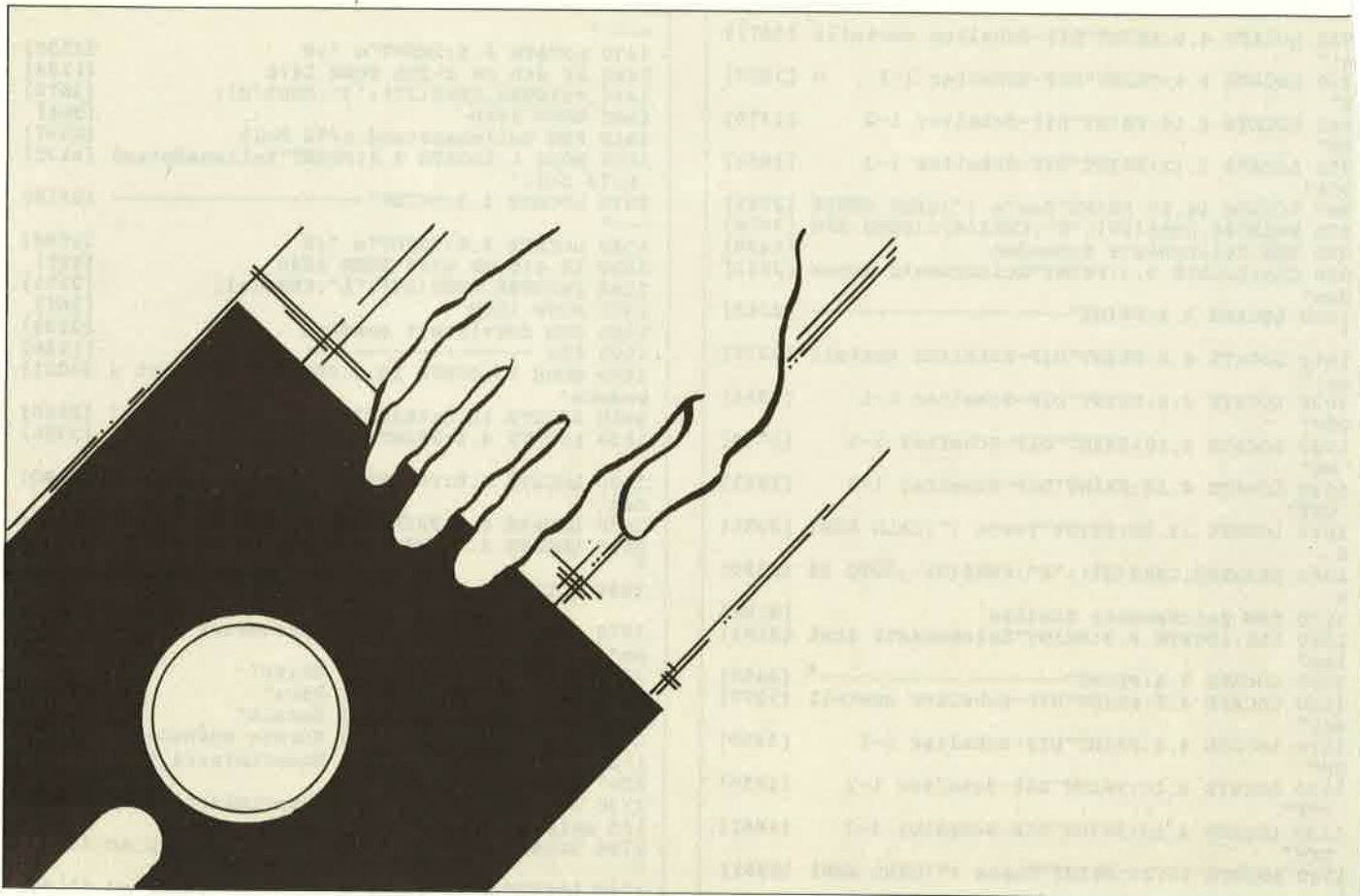
----"
1470 LOCATE 4,6:INPUT"n ";d [1559]
1480 IF d<0 OR d>255 THEN 1470 [1184]
1490 PRINT#8,CHR$(27);"3";CHR$(d); [2075]
1500 GOTO 1250 [361]
1510 REM Zeilenabstand n/72 Zoll [2557]
1520 MODE 1:LOCATE 4,2:PRINT"Zeilenabstand [4135]
n/72 Zoll"
1530 LOCATE 4,3:PRINT"----- [2218]
----"
1540 LOCATE 4,6:INPUT"n ";e [1558]
1550 IF e<0 OR e>85 THEN 1540 [841]
1560 PRINT#8,CHR$(27);"A";CHR$(e); [2111]
1570 GOTO 1250 [361]
1580 REM Schriftart aendern [2122]
1590 REM ----- [1136]
1600 MODE 1:LOCATE 10,2:PRINT"Schriftart a [4021]
endern"
1610 LOCATE 10,3:PRINT"-----" [2260]
1620 LOCATE 4,4:PRINT"A Breitschrift" [2396]

1630 LOCATE 4,5:PRINT"B Breitschrift aufhe [5140]
ben"
1640 LOCATE 4,6:PRINT"C Fettdruck" [1480]
1650 LOCATE 4,7:PRINT"D Fettdruck aufheben [2663]
"
1660 LOCATE 4,8:PRINT"E Doppeldruck" [2643]

1670 LOCATE 4,9:PRINT"F Doppeldruck aufheb [2813]
en"
1680 LOCATE 4,10:PRINT"G Elite" [1119]
1690 LOCATE 4,11:PRINT"H Pica" [1493]
1700 LOCATE 4,12:PRINT"I Kursiv" [2036]
1710 LOCATE 4,13:PRINT"J Kursiv aufheben" [3033]
1720 LOCATE 4,14:PRINT"K Komprimierte Schr [4374]
ift"
1730 LOCATE 4,15:PRINT"L Komprimierte Schr [5222]
ift aufheben"
1740 LOCATE 4,16:PRINT"M Unterstreichen an [3340]
"
1750 LOCATE 4,17:PRINT"N Unterstreichen au [2680]
s"
1760 LOCATE 4,18:PRINT"O Schoenschrift" [2970]
1770 LOCATE 4,19:PRINT"P Schoenschrift auf [3565]
heben"
1780 LOCATE 4,20:PRINT"Q Hauptmenue" [2594]
1790 LOCATE 7,24:PEN 3:PRINT"Auswahl ?":PE [2862]
N 1
1800 f$=INKEY$:IF f$="" THEN 1800 ELSE f$= [3194]
UPPER$(f$)
1810 IF f$="A" THEN PRINT#8,CHR$(27);"W";C [3251]
HR$(1);:GOTO 1580
1820 IF f$="B" THEN PRINT#8,CHR$(27);"W";C [2768]
HR$(0);:GOTO 1580
1830 IF f$="C" THEN PRINT#8,CHR$(27);"E";: [3812]
GOTO 1580
1840 IF f$="D" THEN PRINT#8,CHR$(27);"F";: [2648]
GOTO 1580
1850 IF f$="E" THEN PRINT#8,CHR$(27);"G";: [2570]
GOTO 1580
1860 IF f$="F" THEN PRINT#8,CHR$(27);"H";: [2725]
GOTO 1580
1870 IF f$="G" THEN PRINT#8,CHR$(27);"M";: [2572]
GOTO 1580
1880 IF f$="H" THEN PRINT#8,CHR$(27);"P";: [2272]
GOTO 1580
1890 IF f$="I" THEN PRINT#8,CHR$(27);"4";: [2034]
GOTO 1580
1900 IF f$="J" THEN PRINT#8,CHR$(27);"5";: [2838]
GOTO 1580
1910 IF f$="K" THEN PRINT#8,CHR$(27);CHR$( [4188]
15);:GOTO 1580
1920 IF f$="L" THEN PRINT#8,CHR$(18);:GOTO [2855]
1580
1930 IF f$="M" THEN PRINT#8,CHR$(27);"-";C [2796]
HR$(1);:GOTO 1580
1940 IF f$="N" THEN PRINT#8,CHR$(27);"-";C [1807]
HR$(0);:GOTO 1580
1950 IF f$="O" THEN PRINT#8,CHR$(27);"x";C [2600]
HR$(1);:GOTO 1580
1960 IF f$="P" THEN PRINT#8,CHR$(27);"x";C [2479]
HR$(0);:GOTO 1580
1970 IF f$="Q" THEN 100 [672]
1980 GOTO 1800 [371]
1990 REM Programmende [984]
2000 CLS:END [578]

```

Listing: Drucken - aber richtig



Lineare Gleichungssysteme auf dem CPC

In mehr als 90% aller Programme, die man in einschlägigen Zeitschriften findet, wird entweder gespielt oder gezeigt, wie man noch besser mit einem Computer umgehen kann. Das erinnert durchaus an den Liedersänger, der ein Lied darüber singt, wie es einem Liedersänger so geht; an die Theatergruppe, die ein Stück über das Leben einer Theatergruppe aufführt, oder an den Maler, der auf seinen Bildern einen Maler verewigt. *l'art pour l'art*, sagen die Franzosen; *computing for Computer* sagen die Freaks, oder handeln jedenfalls so.

Interessiert es Sie, was so ein Knopfkasten sonst noch kann? Mit diesem Programm können Sie einen Teil der Mathematik, die "Linearen Gleichungssysteme", nachvollziehen.

Aber Sie brauchen vor dem Wort "Mathematik" nicht zu erschrecken. Wenn Sie in der Lage sind Windows zu erstellen, geschachtelte Gosubs zu verwalten oder sogar ein "Softscrolling" zu programmieren, dann haben Sie keinen Grund anzunehmen, dieses Thema sei zu schwer für Sie!

Einige "lockere" Beispiele

Als erstes möchte ich Ihnen an einigen Beispielen zeigen, mit welchen Problemen wir uns als erstes beschäftigen wollen. Damit nicht alles so tierisch ernst wirkt, sollen uns nun ein paar "lockere" Gedankengänge den Einstieg in dieses trockene Thema erleichtern.

1) Franz züchtet Papageien und Goldhamster. Auf meine Frage, wie viele Tiere er jeweils habe, antwortet er grinsend: "Meine Tiere haben zusammen 20 Köpfe und 66 Füße!" Wissen Sie, wie viele Papageien und Goldhamster Franz besitzt?

2) Fritzchen sagt: "Ich habe genauso viele Schwestern wie Brüder". Seine Schwester Erna hingegen behauptet: "Ich habe doppelt so viele Brüder wie Schwestern". Hat da jemand gelogen oder geht das doch? Und wenn ja, wie viele Jungen und Mädchen gibt es dann in dieser Familie?

3) Weil er einen besonders preisgünstigen Großhändler kennt, hat sich Albert für eine Party einen Gag ausgedacht: Er will für genau 100. – DM 100 Flaschen folgender Getränke kaufen: ,Sekt für DM 10. – pro Flasche, Wein für DM 3. – pro Flasche, Bier für DM –.50 pro Flasche (Wo gibts das noch?). Dabei soll jedes Getränk mindestens einmal vertreten sein. Falls Sie diese Aufgabe bereits kennen oder sogar lösen können: Eine exakte Lösung, die Sie auch auf ähnliche Aufgabenstellungen anwenden können, folgt.

4) Vater und Sohn tragen Pakete. Als der Sohn zu stöhnen beginnt, tröstet ihn der Vater, mit den Worten: "Wenn ich dir noch eins von meinen Paketen gäbe, dann erst hätten wir beide gleichviele zu tragen. Nähme ich aber stattdessen noch ein Paket von dir, dann hätte ich schon doppelt so viele wie du!" Können Sie berechnen, wie viele Pakete der Vater bzw. der Sohn trägt?

5) Ein Hersteller von elektronischen Blumenvasen hat drei Maschinen, die er mit A, B und C bezeichnet und die unterschiedlich schnell arbeiten. Für eine feste Produktionsmenge von 10.000 Stück brauchen die Maschinen A und B 18 Stunden, die Maschinen A und C 20 Stunden und die Maschinen B und C

5 Stunden. Der Fabrikant möchte nun wissen, wie lange jede Maschine einzeln brauchen würde und wieviel Zeit nötig wäre, wenn alle drei Maschinen zugleich arbeiten.

Das Lösungsverfahren

Diese Beispiele stammen aus dem Bereich "Lineare Gleichungssysteme". Schüler lernen davon die einfachste Form etwa in Klasse 9 unter dem Titel "Gleichungen mit 2 Unbekannten", die sie mit drei oder vier verschiedenen Lösungsverfahren zu bearbeiten haben (meist Gleichsetzungs-, Einsetzungs- und Additionsverfahren, dazu ein graphisches Verfahren).

Wir wollen hier ein Verfahren anwenden, das sich besser als die oben genannten für die Auswertung mit dem Computer eignet und das sich auch ohne Änderung leicht für wesentlich größere Systeme benutzen läßt.

Beginnen wir mit Aufgabe 1. Natürlich kann man diese Aufgabe im Kopf lösen, am bequemsten so: Zählen wir doch vorerst von den Goldhamstern nur die Hinterfüße! Dann haben wir 20 Köpfe = 20 Tiere und 40 (Hinter-)füße. Die überzähligen Füße (von 40 bis 66) können nur die Vorderfüße der Hamster sein. Also: 26 Vorderfüße gehören zu 13 Hamstern. Damit bleiben noch 7 Papageien. Fertig! Wozu noch ein Computerprogramm? Nun, können Sie die Aufgaben 2 bis 5 auch so rasch lösen? Selbst wenn es so ist, wird es Ihnen nicht schwerfallen einzusehen, daß es in der Praxis Aufgaben von erheblich höherem Schwierigkeitsgrad gibt. Deshalb jetzt die Lösungsmethode, die sich auf (fast) beliebig große Systeme anwenden läßt.

Zuerst legen wir fest, daß die Anzahl der Papageien mit p bezeichnet werden soll, die der Goldhamster mit g . Das ist gleichbedeutend mit der Festlegung von Variablennamen. Da jeder Papagei natürlich einen Kopf und zwei Füße hat, gibt es also p Papageienköpfe und $2 \cdot p$ Papageienfüße. Entsprechend ergeben sich g Hamsterköpfe und $4 \cdot g$ Hamsterfüße. Also muß $p + g$ zusammen 20 ergeben, und $2 \cdot p + 4 \cdot g$ muß 66 liefern. Daraus machen wir 2 Gleichungen:

- (A) $p + g = 20$
- (B) $2 \cdot p + 4 \cdot g = 66$

Der Aufbau dieser Gleichungen ist typisch für unsere Probleme: Immer stehen die Variablen links vom Gleichheitszeichen, alles andere (die Konstanten) rechts. Wir wollen uns noch im Klaren darüber sein, daß p das gleiche bedeutet wie $1 \cdot p$, g das gleiche wie $1 \cdot g$. Man bezeichnet die Zahlen 2 und 4 in der Gleichung (B) als "Vorzahlen" oder als "Koeffizienten" von p bzw. von g . Ist kein Koeffizient mitgeschrieben, dann ist also immer die "1" gemeint. Schreiben wir sie einmal mit auf:

- (A) $1 \cdot p + 1 \cdot g = 20$
- (B) $2 \cdot p + 4 \cdot g = 66$

Solche Gleichungen wollen wir in Zukunft als "Lineares Gleichungssystem" bezeichnen und mit LGS abkürzen. Das Wesentliche dieses LGS steckt in den Zahlen: Die Variablennamen sind willkürlich gewählt, und das Gleichheitszeichen sowie die Pluszeichen kann man der Übersichtlichkeit wegen während der Rechenphase weglassen, wenn man sie hinterher wieder an der richtigen Stelle anbringt. Dann entsteht ein "nacktes" Zahlenschema, bei dem die Position der Zahlen zueinander aber unverändert bleibt:

```
1 1 20
2 4 66
```

Schließt man dieses Schema noch in große runde Klammern ein, dann nennt man das Ganze in der Mathematik eine "Matrix" (Mehrzahl: Matrizen). Und mit solchen Matrizen läßt sich gut rechnen!

Unsere Matrix besteht aus 2 Zeilen, die den 2 Gleichungen entnommen sind, und 3 Spalten. In unserem Programm werden wir die jeweiligen Zeilen mit i , die Spalten mit k nummerieren. Der folgende Programmteil spricht also jedes "Element" der Matrix an:

```
for i = 1 to 2
for k = 1 to 3
.....
next k:next i
```

Mit $i=2$, $k=3$ erwischen wir also die Zahl 66. Dementsprechend können wir diese Zahl auch verschlüsseln mit $a(2,3)$, und wir werden deshalb mit der DIM-Anweisung ein Feld bestimmen, das unsere Zahlen aufnehmen soll:

```
DIM a(2,3).
```

In größeren Problemen ergeben sich durchaus mehr Zeilen und Spalten als in unserem Beispiel. Deshalb wollen wir flexibel sein und generell die Anzahl der Zeilen mit m , die der Spalten mit n bezeichnen, so daß der o.a. Programmteil sich ein wenig verändert:

```
for i = 1 to m
for k = 1 to n
.....
next k:next i
```

und unsere DIM-Anweisung heißt dann:

```
DIM a(m, n)
```

Wie können wir nun unsere Matrix überreden, uns die Lösung des LGS zu verraten? Folgende Schritte müssen mehrfach durchlaufen werden:

Wir ernennen zuerst eine Zahl der Matrix zum "King", d.h. wir suchen ein sogenanntes Pivot-Element PE. Dieses muß folgende Bedingungen erfüllen:

- I) Es darf nicht Null sein,
- II) in keiner Zeile und in keiner Spalte darf mehr als ein Pivot-Element gewählt werden,
- III) es darf nicht in der letzten Spalte stehen. Nun, das ist meist einfach zu erreichen. Wir könnten z.B. die erste Zahl oben links in unserer Matrix zum 1. Pivot-Element (PE) ernennen. Und was dann?

Drei Schritte sind jetzt notwendig, die man (falls die Zahlen nicht zu groß sind) leicht im Kopf ausführen kann, die aber der Computer selbständig durchführt und die eine neue Matrix erzeugen:

- A) Die Pivotzeile bleibt unverändert (Das ist natürlich die Zeile, in der das PE steht).
- B) Die Pivotspalte wird Null außer dem PE selbst (über und unter dem PE stehen also nur noch Nullen in der neuen Matrix).
- C) Man verbindet im Gedanken jedes der noch nicht erfaßten Elemente der alten (!) Matrix mit dem PE und stellt sich diese Linie als Diagonale eines Rechtecks vor. Man sucht die andere Diagonale und stellt fest, welche Zahlen auf ihrem Ende stehen. Ich nenne sie hier einfach K_1 und K_2 . Sie stellen den 3. und 4. "Punkt" dieses gedachten Rechtecks dar. Jetzt ersetzt man das fragliche Element der alten Matrix durch die Zahl: altes El. \cdot PE $-$ $K_1 \cdot K_2$.

Das ist ein Hammer, werden Sie sagen! Aber haben Sie nicht auch schon mal mit einem Grafik-Programm ein Rechteck gezeichnet? Bei den mir bekannten Programmen wird ein solches Rechteck festgelegt, indem man zwei diagonal gegenüberliegende Eckpunkte auswählt; die zwei anderen Eckpunkte findet das Programm selbständig. Man weiß aber auch vorher schon, wo sie sein werden! Nun, genau so ist es in der Matrix: Wenn sie "zu Fuß" rechnen wollen, findet man bei einem bißchen Übung die Zahlen, die hier mit K_1 und K_2 bezeichnet wurden, sofort. In einer schematischen Übersicht sieht das so aus:

Anwendungen

```

.....
.. PE ... K1 ..
.....
.. K2 ... ? .....
.....

```

Dabei bedeutet das Fragezeichen das Element, das neu berechnet werden muß. In unserem Beispiel wählen wir als Pivot-Element die "1" oben links aus, was keinen Konflikt mit den Regeln I), II) und III) beschwört, und nach Anwendung der Regeln A) und B) entsteht:

```

  1   1   20
  0   ?   ?

```

Die Fragezeichen ergeben sich jetzt nach Regel C) so:

$$4 * 1 - 2 * 1 = 2 \text{ und}$$

$$66 * 1 - 2 * 20 = 26$$

Also sieht unsere Matrix nach diesem Vorgang jetzt so aus:

```

  1   1   20
  0   2   26

```

Nun machen wir das Ganze noch ein zweites Mal: Als Pivot-Element können wir nach den Regeln I), II) und III) nur die "2" wählen. Also wieder nach den Regeln A) und B):

```

  ?   0   ?
  0   2   26

```

und nach Regel C):

```

  2   0   14
  0   2   26

```

Dabei entstand die obere "2" durch $2 * 1 - 0 * 0 = 2$ und die "14" durch $2 * 20 - 1 * 26 = 14$.

Ein weiteres Pivot-Element läßt sich nun nicht mehr bestimmen, wir sind fertig. Was sind also die Lösungen? So, wie wir vorhin aus dem LGS eine Matrix hergestellt haben, indem wir die Variablenbezeichnungen und das Gleichheitszeichen weggelassen haben, so erhalten wir aus der Matrix wieder ein LGS, indem wir beides wieder einfügen:

$$2 * p + 0 * g = 14, \text{ also } 2 * p = 14,$$

$$\text{mithin } p = 7$$

$$0 * p + 2 * g = 26, \text{ also } 2 * g = 26, g = 13$$

Zugegeben, das alles sieht nach sehr viel Aufwand aus, aber wer diese Umformungen einübt, kann die Lösung mit wenigen Zeilen errechnen. Das sieht dann in Kurzform etwa so aus:

$$1 \ 1 \ 20 \rightarrow 1 \ 1 \ 20 \rightarrow 2 \ 0 \ 14$$

$$\rightarrow p = 14/2 = 7$$

$$2 \ 4 \ 66 \quad 0 \ 2 \ 26 \quad 0 \ 2 \ 26$$

$$g = 26/2 = 13$$

Falls Ihnen das alles aber doch zuviel ist, oder falls Sie ein LGS von höherem Grad (mit mehr als 2 Unbekannten) lösen wollen, benutzen Sie einfach das Programm. Es rechnet nach der oben angegebenen Methode, die ein modifizierter "Gaußscher Algorithmus" ist. Was also sollte unser Programm leisten können?

Was das Programm kann

Nach dem Start fragt es erst nach der Anzahl der Gleichungen und der Anzahl der Variablen. Das ist nicht immer die gleiche Zahl, wie wir noch sehen werden. Bis zu maximal 4 Variablen erscheint der Bildschirm in Mode 1, sonst in Mode 2. Danach wird eine Matrix den Umrissen nach mit Numerierung aufgebaut, so daß man einfach die Zahlen zeilenweise eingeben kann. Nach jeder Zeile fragt der Kasten, ob alles korrekt ist, wenn man mit "n" antwortet, hat man Gelegenheit, die jeweils letzte

Zeile zu wiederholen. Als nächstes wird man gefragt, ob man alle Zwischenmatrizen sehen will. Hat man mit "j" oder mit "n" geantwortet, will das Programm noch "freie Variable Nr. wissen. Was das ist, lesen Sie im nachfolgenden Text.

Drücken Sie einfach die RETURN-Taste (bzw. die ENTER-Taste). In wenigen Sekunden sieht man das Ergebnis vor sich in der Form "X1 = ... X2 = ...". Zusätzlich wird die Endmatrix angezeigt. Die Variablen heißen im Programm immer X1, X2. Daß damit in unserem ersten Problem p und g gemeint sind, werden Sie sicher leicht einsehen.

Für eine weitere Aufgabe müssen Sie einfach nur RUN eingeben. Eine Besonderheit des Programms ist, daß man ganze Zahlen und Brüche eingeben kann. Als Bruchstrich benutzen Sie den Schrägstrich, der für die Division vorgesehen ist, also z.B. 2/3 oder -4/17. Alle Zwischenergebnisse werden mit ganzen Zahlen berechnet, was den sonst unvermeidlichen Rundungsfehlern keine Chance läßt. Auch die Ergebnisse werden ganzzahlig oder in Form von Brüchen ausgegeben. Deshalb sind sie völlig exakt, sofern der Rechenbereich die Grenzen von $+ - 10$ Milliarden nicht überschreitet. Damit das auch nicht während einer Rechnung allzu oft unsichtbar passiert, werden die neu berechneten Zeilen jeder Matrix daraufhin überprüft, ob man sie "kürzen" kann und gegebenenfalls durch einen gemeinsamen Teiler dividiert.

Trotzdem ist es nicht zweckmäßig, alle Variablen im Programm als "INT" zu definieren, weil der Rechenbereich $+ - 32767$ doch zu klein wäre.

Ein Nachteil dieser Programmauslegung soll nicht verschwiegen werden: Falls ein Problem Dezimalzahlen enthält, z.B. Meßwerte, dann sollten besser die Ganzzahlrechnungen gestrichen werden (für Kenner: und durch eine Pivot-Vorwahl ergänzt werden, um Rundungsfehler klein zu halten).

Um die Regel II) zum Aufsuchen eines Pivot-Elementes einzuhalten, muß das Programm sich "merken", welche Zeilen und Spalten nicht mehr in Frage kommen. Ist z.B. ein PE in der Zeile mit der Nummer i und in der Spalte mit der Nummer k gefunden worden, so schreibt das Programm eine "2" in das Element a(0,k) und die Nummer k in das Element a(i,0), und bei einer späteren Suche scheiden diese Zeile und diese Spalte sofort aus.

Auf ähnliche Weise werden auch die freien Variablen verwaltet: Ihre Kennzeichnung ist eine "1" im Element a(0,k). In dieser Spalte wird vorerst kein PE gesucht. Hat man aber Variablen zu Unrecht als frei erklärt, so bemerkt das Programm den Fehler, gibt eine Nachricht aus und rechnet nun richtig zu Ende.

Natürlich brauchen Sie als Benutzer des Programms sich darum überhaupt nicht zu kümmern. Auch in anderer Hinsicht ist das Programm gegen "fiese Situationen" geschützt. So erkennt es ohne Absturz unlösbare Probleme und gibt eine entsprechende Nachricht aus. Der gewählte Algorithmus garantiert aber, daß keine Lösung "übersehen" wird.

Auch die Reihenfolge, in der Sie die Zeilen eingeben, ist ohne Bedeutung. Beginnen wir mit Aufgabe 2. Zuerst legen wir fest, daß die Anzahl der Jungen in der Familie mit j, die der Mädchen mit m bezeichnet wird. Als nächstes machen wir uns klar, daß Fritzchen damit m Schwestern, aber nur j-1 Brüder hat, denn er ist zwar bei den j Jungen mitgezählt, aber er ist nicht sein eigener Bruder! Anzahl der Schwestern gleich Anzahl der Brüder bedeutet nun einfach $j - 1 = m$ (A). Erna hingegen hat j Brüder, aber nur m-1 Schwestern, weil auch sie sich nicht als ihre eigene Schwester zählen mag.

Nun ist, laut Text der Aufgabe, j doppelt so groß wie m-1, man kann aber beide Zahlen auf gleiche Größe bringen, wenn man die kleinere mit 2 malnimmt: $2 * (m-1)$, das ergibt $2m-2$. Also: $j = 2m - 2$ (B)

un muß sortiert werden: Die Variablen j und m bzw. $2m$ sollen links vom Gleichheitszeichen, alles andere rechts davon stehen. Das geht aber nur richtig aus, wenn man zugleich die Vorzeichen ändert: Aus $-$ wird $+$, und aus dem oft nicht mitgeschriebenen $+$ wird $-$. Also:

$$(A) j - m = +1$$

$$(B) j - 2m = -2$$

Die Matrix dazu hat folgendes Aussehen:

$$\begin{array}{ccc} 1 & -1 & 1 \\ 1 & -2 & -2 \end{array}$$

Starten Sie das Programm. Es handelt sich um 2 Gleichungen mit 2 Variablen. Geben Sie die Matrix ein. Das Ergebnis sehen Sie fast sofort vor sich! Bedenken Sie: $X1$ bedeutet jetzt j (also die Anzahl der Jungen in der Familie), $X2$ bedeutet m (=Anzahl der Mädchen).

Die Aufgabe 3 kann man so lösen: Nehmen wir an, Albert kauft s Flaschen Sekt, w Flaschen Wein und b Flaschen Bier. Dann hat er insgesamt $s + w + b$ Flaschen und das sollen hundert sein:

$$(A) s + w + b = 100$$

Eine Flasche Sekt kostet DM 10.-, also kosten 3 Flaschen $3 \cdot 10$ Mark, und s Flaschen kosten $s \cdot 10$ Mark. Genauso ergibt sich $w \cdot 3$ Mark für w Flaschen Wein und $0.5 \cdot b$ Mark oder $1/2 \cdot b$ Mark für b Flaschen Bier. Da Albert genau 100 Mark ausgeben will, gilt die Gleichung (B) $10s + 3w + 1/2b = 100$. Die Informationen des Textes reichen für keine weitere Gleichung. Wir haben hier also zwar 3 Variable, aber nur 2 Gleichungen! Eigentlich ist ein solches System nicht eindeutig lösbar, aber es gibt noch eine Zusatzbedingung: Alle Variablen müssen ganzzahlig sein! Und dann geht's doch! Die Mathematiker nennen Systeme mit dieser Zusatzforderung "diophantisch". Mal sehen, was das Programm damit macht.

Geben Sie ein: 2 Gleichungen, 3 Variable, Matrix:

$$\begin{array}{cccc} 1 & 1 & 1 & 100 \\ 10 & 3 & 1/2 & 100 \end{array}$$

Lösung: $X1 = -200/7 + 5/14 \cdot X3$

$$X2 = 900/7 - 19/14 \cdot X3$$

Wie man aus diesem Wust von Zahlen eine Lösung erkennen kann, weiß keiner. Aber sie sehen, daß eine Variable, nämlich $X3$, auf der rechten Seite der Gleichungen erscheint. Für $X3$ wird keine Lösung mehr angegeben. Das bedeutet aber, daß $X3$ nicht berechenbar ist und somit frei gewählt werden kann! Solche Variablen heißen "freie Variable". Noch ein Versuch: Geben Sie die Werte noch einmal ein, aber tippen Sie bei der Frage nach der Nummer der freien Variablen eine "1" ein, bei der Wiederholung der Frage RETURN! Der Computer gibt aus:

$$X2 = 20 - 19/5 \cdot X1$$

$$X3 = 80 + 14/5 \cdot X1$$

Wie Sie sehen, ist hier $X1$ zur freien Variablen geworden. Jetzt ist es eine Kleinigkeit, die Lösung zu finden:

Die erste Zeile besagt nämlich, daß man von 20 das 19/5-fache von $X1$ abziehen muß, um $X2$ zu erhalten. Man wird sofort vermuten, daß dann $X2$ nicht mehr ganzzahlig ist. Oder doch? Wenn man 19/5 malnimmt mit einer 5 oder 10 oder 15 usw., dann entsteht eine ganze Zahl! Probieren wir es aus:

$$\text{Mit } X1 = 5 \text{ wird } X2 = 20 - 19/5 \cdot 5 = 20 - 19 = 1$$

$$\text{Mit } X1 = 10 \text{ wird } X2 = 20 - 19/5 \cdot 10 = 20 - 38 = -18$$

Das geht natürlich nicht: $X2$ kann nicht negativ werden. Machen wir $X1$ noch größer, wird $X2$ noch kleiner. Es bleibt also nur eine Möglichkeit: $X1 = 5$ und $X2 = 1$. Die zweite Lösungszeile liefert uns jetzt noch $X3$:

$$X3 = 80 + 14/5 \cdot 5 = 80 + 14 = 94.$$

Hier bedeutet $X1 = s$, $X2 = w$, $X3 = b$, und die komplette Lösung heißt

$$s = 5, w = 1, b = 94.$$

Also: Albert kauft 5 Flaschen Sekt, eine Flasche Wein und 94 Flaschen Bier. Nun ja, bei dem Preis! Kontrolle, ob alles aufgeht:

$$5 + 1 + 94 = 100 \text{ Flaschen,}$$

$$50 + 3 + 47 = 100 \text{ Mark.}$$

Sie haben sicher alles verstanden, und so macht Ihnen die Aufgabe 4 kaum Kopfzerbrechen. Vater trägt v Pakete, der Sohn $s + 1$. Es muß also gelten:

$$v - 1 = s + 1,$$

und nach Umstellung wie in der zweiten Aufgabe:

$$A) v - s = 2$$

Nimmt er aber noch eines an, dann hat er $v + 1$, und das sind doppelt so viele wie die $s - 1$ Pakete, die dem Sohn bleiben. Also verdoppeln wir die kleinere Zahl, das ergibt $2s - 2$, und jetzt sind beide Zahlen gleich groß. Also heißt die Gleichung:

$$v + 1 = 2s - 2,$$

und nach Umstellung:

$$B) v - 2s = -3$$

Die Matrix heißt jetzt:

$$\begin{array}{ccc} 1 & -1 & 2 \\ 1 & -2 & -3 \end{array}$$

Die Lösung finden Sie sicher selbst. Bei der Aufgabe 5 hilft ein kleiner Trick: Wenn die Maschinen A und B 18 Stunden zur Fertigung der Vasen brauchen, dann schaffen sie in einer Stunde $1/18$ der Menge. Genauso schaffen A und C in einer Stunde $1/20$, B und C $1/45$ der Gesamtmenge. Also heißen die Gleichungen:

$$A + B = 1/18$$

$$A + C = 1/20$$

$$B + C = 1/45$$

und die zugehörige Matrix:

$$\begin{array}{cccc} 1 & 1 & 0 & 1/18 \\ 1 & 0 & 1 & 1/20 \\ 0 & 1 & 1 & 1/45 \end{array}$$

Das sind nun 3 Gleichungen mit 3 Unbekannten (oder ein LGS vom Grad 3). Ihr Computer schafft die Lösung augenblicklich:

$$X1 = A = 1/24,$$

$$X2 = B = 1/72,$$

$$X3 = C = 1/120$$

Das bedeutet natürlich, daß die Maschinen diese Bruchteile der ganzen Arbeit in jeweils einer Stunde schaffen, also braucht A 24 Stunden, B 72 Stunden und C 120 Stunden, wenn sie allein arbeiten.

Und die Zusatzfrage? Wir zählen A, B und C zusammen:

$$1/24 + 1/72 + 1/120 = 15/360 + 5/360 + 3/360 = 23/360.$$

Wenn also alle Maschinen zusammen $23/360$ der Arbeit in einer Stunde schaffen, dann brauchen sie für die ganze Arbeit eben $360/23$ Stunden, das sind 15, 65 Stunden (oder, falls Sie es noch genauer wissen wollen, 15 Stunden, 39 Minuten und knapp 8 Sekunden!).

Lineare Gleichungssysteme, wie sie hier vorgestellt werden, treten in unüberschaubar vielen Anwendungen auf, und sie haben keineswegs immer den Geruch von Denksportaufgaben, wie bei unserer kleinen Auswahl. Ein Problem mit ernsthafterem Hintergrund soll Ihnen zur eigenen Bearbeitung überlassen werden:

Das E-Werk bietet Ihnen zwei Tarife an. Sie sollen sich entscheiden, ob Sie DM 30.- Grundgebühr und DM -.12 pro Kilowattstunde zahlen wollen oder lieber DM 12.- Grundgebühr und DM -.18 pro Kilowattstunde.

Der erste Tarif ist offenbar günstiger, wenn Sie viel verbrauchen. Aber wo ist die Grenze? Nehmen Sie an, Sie verbrauchen x kWh und zahlen dafür y DM. Die Tarife liefern

$$1. y = 12x + 3000$$

$$2. y = 18x + 1200$$

Anwendungen

(in Pfennigen, um Dezimalzahlen zu vermeiden). Nach dem Umsortieren sieht man die Matrix:

```
-12  1  3000
-18  1  1200
```

Die Lösung finden Sie nach Lesen des Artikels bestimmt selbst. Vielleicht sollten Sie jetzt eine Pause machen und die Tarife Ihres E-Werkes heraussuchen?

(D. Noack/JB)

FÜR 464-664-6128



```
1000 '***** [894]
1010 '*** Gleichungssysteme *** [1917]
1020 '*** D. Noack *** [1143]
1030 '***** [894]
1040 ' [117]
1050 ON ERROR GOTO 3600 [1483]
1060 INPUT"wieviele Gleichungen";m [2786]
1070 INPUT"wieviele Variable ";n [1660]
1080 IF n<5 THEN md=1 ELSE md=2 [1409]
1090 MODE md [440]
1100 n=n+1:DIM a(m,n) [1062]
1110 up=1:o=n-1:GOSUB 4000' Matrix [3302]
1120 ef=0:FOR i=1 TO m [1067]
1130 IF a(i,0)=0 AND a(i,n)<>0 THEN GOSUB [5574]
3800:IF ef=0 THEN up=0:PRINT "unloesbar!":
GOSUB 10000:END
1140 NEXT i [375]
1150 ' [117]
1160 FOR i=1 TO m [457]
1170 v=a(i,0) [839]
1180 IF v THEN GOSUB 2000 [1251]
1190 NEXT i [375]
1200 IF ef THEN PRINT"keine freie Variable [4499]
!":ef=0:GOSUB 4100:GOTO 1150
1210 PRINT [361]
1220 IF m$="n" THEN GOSUB 10000 [526]
1230 up=0:END [503]
1240 ' [117]
2000 '***** [345]
2010 '*** Ausgabe *** [829]
2020 '***** [345]
2030 y=a(i,v):x=a(i,n)*SGN(y):y=ABS(y) [2270]
2040 GOSUB 3000' kuerzen [927]
2050 d$="X"+MID$(STR$(v),2)+" "+STR$(x) [1119]
2060 IF y>1 THEN d$=d$+"/"+MID$(STR$(y),2) [2751]
2070 PRINT d$; [384]
2080 FOR k=1 TO n-1 [1693]
2090 IF a(0,k)<2 THEN GOSUB 2130:PRINT a [2317]
$;
2100 NEXT k [373]
2110 PRINT :RETURN [914]
2120 ' [117]
2130 a$="" [388]
2140 y=a(i,v):x=a(i,k)*SGN(y):y=ABS(y):IF [2816]
x=0 THEN RETURN
2150 GOSUB 3000' kuerzen [927]
2160 IF x<0 THEN a$="+" ELSE a$="" [851]
2170 a$=a$+STR$(x) [835]
2180 IF y>1 THEN a$=a$+"/"+MID$(STR$(y),2) [2891]
2190 a$=a$+"X"+MID$(STR$(k),2) [1457]
2200 RETURN [555]
2210 ' [117]
3000 '***** [345]
3010 '*** kuerzen *** [1437]
3020 '***** [345]
3030 p=ABS(x):q=ABS(y) [400]
3040 GOSUB 3300' ggT [945]
3050 x=x/q:y=y/q [1028]
3060 RETURN [555]
3070 ' [117]
3300 '***** [332]
3310 '*** ggT *** [568]
3320 '***** [332]
3330 r=p-q*INT(p/q) [1039]
3340 IF r THEN p=q:q=r:GOTO 3330 [829]
3350 RETURN [555]
3360 ' [117]
3600 '***** [445]
3610 '*** Fehler *** [1262]
3620 '***** [445]
3630 PRINT"Fehler ";ERR;" in Zeile ";ERL:E [2382]
ND
```

Listing Lineare Gleichungssysteme

```
3640 ' [117]
3800 '***** [332]
3810 '*** Sub *** [321]
3820 '***** [332]
3830 FOR k=1 TO n-1 [1693]
3840 IF a(0,k)=1 THEN a(0,k)=0:ef=1 [1363]
3850 NEXT k [373]
3860 RETURN [555]
3870 ' [117]
4000 '***** [445]
4010 '*** Matrix *** [1240]
4020 '***** [445]
4030 IF e=0 THEN e=n [441]
4040 GOSUB 6000' Eingabe [1751]
4050 PRINT"Zwischenmatrix j/n" [1913]
4060 GOSUB 5000 [883]
4070 m$=i$ [123]
4080 INPUT"freie Variable Nr.":v$ [2022]
4090 IF LEN(v$) THEN a(0,VAL(v$))=1:GOTO 4 [2046]
080
4100 ' [117]
4110 IF o=0 THEN o=MAX(m,n) [370]
4120 FOR j=1 TO o [798]
4130 PRINT"j=";j:PRINT [1070]
4140 ip=0 [179]
4150 GOSUB 8000' Pivot suchen [2252]
4160 ' [117]
4170 'falls gefunden,dann tauschen [1895]
4180 ' [117]
4190 IF ip>0 THEN GOSUB 9000:NEXT j [1587]
4200 IF up THEN RETURN [808]
4210 GOSUB 10000'*** Display *** [2243]
4220 END [110]
4230 ' [117]
5000 '***** [445]
5010 '*** Tasten *** [579]
5020 '***** [445]
5030 i$="" : WHILE i$="" : i$=INKEY$:WEND [2275]
5040 i$=LOWERS$(i$) [773]
5050 IF INSTR("jn",i$)=0 THEN 5030 [1607]
5060 RETURN [555]
5070 ' [117]
6000 '***** [345]
6010 '*** Eingabe *** [1674]
6020 '***** [345]
6030 FOR i=1 TO m [457]
6040 LOCATE 1,i+2:PRINT i [1200]
6050 NEXT i [375]
6060 FOR k=1 TO e [1321]
6070 LOCATE k*6,1:PRINT k [1619]
6080 NEXT k [373]
6090 MOVE 50-10*md,390-10*md:DRAWR 0,-m*20 [2148]
6100 MOVE 50-10*md,390-10*md:DRAWR e*50*(3 [1994]
-md),0
6110 ma=0 [89]
6120 FOR i=1 TO m:p=1 [1029]
6130 FOR k=1 TO e [1321]
6140 LOCATE k*6,i+2 [1198]
6150 GOSUB 7000 [867]
6160 a(i,k)=a(a(0,k)=b [90]
6170 p=p*b [71]
6180 NEXT k [373]
6190 ' [117]
6200 LOCATE 1,m+5:PRINT"alles richtig?" [2345]
6210 GOSUB 5000' Tasten [1009]
6220 LOCATE 1,m+5:PRINT SPACES(14) [1952]
6230 IF i$="n" THEN LOCATE 4,i+2:PRINT SPA [2746]
CES(e*5+2):i=i-1:GOTO 6300
6240 FOR k=1 TO e [1321]
6250 IF p<>1 THEN a(i,k)=a(i,k)*p/a(0, [2836]
k)
6260 ma=MAX(ABS(a(i,k)),ma) [1803]
6270 a(0,k)=0 [430]
6280 NEXT k [373]
6290 a(i,0)=0 [446]
6300 NEXT i [375]
6310 RETURN [555]
```

Listing Lineare Gleichungssysteme

```

7000 '*****' [577]
7010 '*** Eingabe Brueche ***' [2491]
7020 '*****' [577]
7030 INPUT "",i$ [440]
7040 a=VAL(i$):IF a<0 THEN a$=STR$(a) ELSE [3498]
      a$=MID$(STR$(a),2)
7050 IF a$<>i$ THEN b$=MID$(i$,LEN(a$)+2): [2395]
b=MAX(VAL(b$),1) ELSE b=1
7060 RETURN [555]
7070 ' [117]
8000 '*****' [665]
8010 '*** Pivot suchen ***' [1213]
8020 '*****' [665]
8030 FOR i=1 TO m [457]
8040 IF a(i,0)=0 THEN GOSUB 8080:IF ip T [1537]
HEN RETURN
8050 NEXT i [375]
8060 RETURN [555]
8070 ' [117]
8080 FOR k=1 TO o [1343]
8090 IF a(0,k)=0 THEN GOSUB 8130:IF ip T [2416]
HEN RETURN
8100 NEXT k [373]
8110 RETURN [555]
8120 ' [117]
8130 IF a(i,k)<>0 THEN ip=i:kp=k [2031]
8140 RETURN [555]
8150 ' [117]
9000 '*****' [528]
9010 '*** tauschen ***' [646]
9020 '*****' [528]
9030 a(ip,0)=kp:a(0,kp)=2' notieren [1619]
9040 ' [117]
9050 FOR i=1 TO m [457]
9060 IF i<ip THEN GOSUB 9180:IF q>1 THE [3199]
N GOSUB 9130
9070 FOR k=1 TO n: [1582]
9080 ma=MAX(ABS(a(i,k)),ma) [1803]
9090 NEXT k [373]

```

Listing Lineare Gleichungssysteme

```

9100 NEXT i [375]
9110 IF m$="j" THEN GOSUB 10000'Matrix [1153]
9120 RETURN [555]
9130 FOR k=1 TO n [1344]
9140 a(i,k)=a(i,k)/q [747]
9150 NEXT k [373]
9160 RETURN [555]
9170 ' [117]
9180 FOR k=1 TO n [1344]
9190 IF k<>kp THEN a(i,k)=a(i,k)*a(ip,kp [2303]
)-a(ip,k)*a(i,kp)
9200 NEXT k [373]
9210 a(i,kp)=0' Pivotspalte=0 [1196]
9220 FOR k=1 TO n-1 [1693]
9230 q=ABS(a(i,k)) [778]
9240 IF q=1 THEN RETURN [1532]
9250 IF q=0 THEN NEXT k:RETURN [2037]
9260 ' [117]
9270 FOR k=k+1 TO n [617]
9280 p=ABS(a(i,k)) [640]
9290 IF p=1 THEN q=1:RETURN [1301]
9300 IF p THEN GOSUB 3300 [1242]
9310 IF q>1 THEN NEXT k [667]
9320 RETURN [555]
9330 ' [117]
10000 '*****' [673]
10010 '*** Display Matrix ***' [2281]
10020 '*****' [673]
10030 ma$=STR$(ma):u$="#" [246]
10040 FOR x=1 TO LEN(ma$) [910]
10050 u$=u$+"#" [655]
10060 NEXT x [356]
10070 FOR x=1 TO m [685]
10080 FOR y=1 TO n [1240]
10090 PRINT USING u$;a(x,y); [986]
10100 NEXT y [359]
10110 PRINT [361]
10120 NEXT x [356]
10130 RETURN [555]

```

Listing Lineare Gleichungssysteme

Das REF-PROGRAMM

Bei der Untersuchung, Straffung oder Änderung von fremden Programmen wie auch bei der Arbeit an eigenen Produktionen, wäre es sehr vorteilhaft, sich mal schnell einen Überblick über die verwendeten Variablen zu verschaffen, oder nachzusehen, wo denn überall hin und her gesprungen wird, oder kurz auszudrucken, in welchen Zeilen jener bestimmte Befehl auftritt, der im eigenen Basic ganz anders heißt.

Tja, das sind schöne Träume, oder etwa nicht?

»NEIN«, heißt hier die Antwort, denn es gibt ja REF. REF untersucht Basic-Programme auf das Vorkommen von Variablen, Zeilennummern in Befehlen (Sprünge, Restore, Resume) oder beliebige Befehle und gibt die entsprechenden Zeilennummern auf den Monitor oder auf einen Drucker aus. Dabei bestehen folgende Wahlmöglichkeiten:

A) Variablen

- zeilenweise
- einmal/mehrfach je Zeile
- ohne/mit Typkennzeichen
- alphabetisch geordnet, nach Typen getrennt.

B) Zeilennummern in Befehlen

- zeilenweise
- geordnet nach den angesprochenen Zielnummern.

C) jegliche Befehle.

Programm-Beschreibung

REF.BAS verwendet die Datei REF.DAT (Bildschirmgestaltung, diverse Strings), welche durch RUN: »LAYOUT.REF« erstellt wird. REF wird gestartet

- wenn nur das zu untersuchende Programm im Speicher steht mit : CHAIN MERGE "REF",62000
- wenn beide Programme vorhanden sind : RUN 62000
- wenn nur REF oder keines geladen ist : RUN / RUN"REF.

Vorab werden abgefragt:

- die Ausführungsart (A, B, C)
- der Name des zu ladenden Programms
- ob das Programm schon geladen ist
- erste und letzte zu untersuchende Zeile
- der Ausgabe-Stream (beim Drucker zusätzlich: das Datum)
- die jeweiligen Details zur Ausführungsart.

Danach wird das Programm, wenn notwendig, geladen und in einem Durchgang untersucht. Alle Zeilen bis zur Startzeile werden überlesen. Sobald die Endzeile erreicht ist, wird die Untersuchung abgebrochen. Bei der zeilenweisen Ausführung und den Basic-Befehlen erfolgt die Ausgabe laufend während der Bearbeitung. Bei geordneter Ausführung werden die Variablennamen bzw. Zielzeilennummern mit den Zeilennummern ihres Auftretens in Arrays abgelegt und nach beendeter Untersuchung sortiert und ausgegeben. Danach ist mit RUN 63000 ein weiterer Durchgang möglich.

```

Basic-Programm: REF          (Zeilen 62000 - 64650)      28.08.1987
Variablen, alphabetisch

INTBOGR...
  ANZ 62290 63210 63230
  INDEX
  a 62310 62460 62480 62490 62500 62560 62720 62850 62860 62890 62900
    62910 62920 62930 62960 63000 63010 63040 63050 63060 63070 63090
    63120 63320 63400 63490 63530 63640 63830 63840 64150 64650
  ab 63340 63420 63710 64080
  afg 62210 62290 62700 62840 63180 63810 63840 63860 63880 63890 64070
    64190 64210
  altart 62920 63520 63680 64320
  anzeig 63260 64290
  art 62460 62490 62500 62920 62970 63120 63210 63230 63270 63310 63620
    63640 63680
  aus 62070 62080 62090 62230 62740 62750 62800 62930 62940 63070 63240
    63250 63260 63270 63350 63560 63580 63630 63640 63650 63660 64000
    64010
  b 62460 62470 62480 62490 62500 62520 62860 62870 62880 63010 63020
    63030 63210 63840 64190 64200
  c 63540 63550 63560 63570 63620 63630 63640
  change 62850 62870 62890 62900 62910 63000 63020 63040 63050 63060
  da 63930 64090
  dimax 62040 63340 63420 64140 64150 64320 64410
  ein 63810 64380 64650
  einmal 63220 64280
  emax 63800 63910 63950 64020 64520
  endezei 62300 62340 62370 62430 62520
  enc 62850 62860 63000 63010 63320 63340 63350 63400 63420 63430 63640
    64160
  fo 64450 64460 64520 64580 64590 64610
  foflag 63800 63910 63930 63950 64000 64020 64260 64390 64480 64500 64510
    64540
  geord 62210 62220 62290 62730 63190 63340 63420 64260 64270 64390 64400
  hi 63470 63480
  inline 62280 62740 63240 63250
  inter 62910 63060

STRING...
  TYP 63630 64330
  VAR 63210 63230 63490 63530 64140 64300
  VARIA 62870 62900 62920 62930 63320 63350 64320
  prog 62070 63910 64110
  quest 64450 64470
  rahmen 63780
  e 62310 62320 62930 62940 63480 63490 64010 64190 64650
  such 63840
  c 63800 63810 63820 63840 63910 63930 64020 64490 64520 64540 64590
    64600 64610
  text 64250
  text0 63910 63950 63980 64000 64260 64280 64290 64390 64610
  text1 63930 63950 64000 64260 64280 64290 64390 64610
  tokens 62320 64040 64050 64060 64190 64200
  tx 62090 64210 64270 64300 64340 64400 64410 64420
  typen 62490 62500 63120 64080
  v 62570 62650 63130 63140 63210 63230 63260 63310 63320 63350 63530
    63540 63560
  wort 63820 63880 64210

  zeit 62080 64020
  zwi 62900

REAL...
  LN 63020 63050 63070 63400 63430 64410
  lazei 62140 62160 62370
  lnadr 62280 62370
  lnanf 62080 62160 63960
  lnend 62080 62150 62220 63340 63420 63980
  lninter 62790 62800 63050
  lnle1 62710 62720 62730 63400 63430
  m 62140 62160 62280 62310 62370 62390 62400 62410 62420 62430 62460
    62480 62500 62550 62560 62630 62640 62650 62700 62710 63120 63130
    63140 63180 63960 63980 64000 64080 64260 64280 64290 64390 64490
    64500 64540
  nozei 62130 62140 62150 62160 62740 62790 62800 63250 63310 63340 63390
    63420 63470 63480 64080
  
```

Abbl: Die Variablenliste von REF

REF Untersuchung von BASIC-PROGRAMMEN
VARIABLEN

(RETURN)=Variablen / (1)=Befehle mit Zeilennummern / (Befehl):

NAME des Programms	(RETURN)=vorhanden / (1)=laden	ref vorhanden
STARTZEILE	(RETURN)=erste	erste
ENDZEILE	(RETURN)=letzte	65000
AUSGABE auf	(RETURN)=Monitor / (1)=Drucker:	Monitor
ORDNUNG	(RETURN)=nach Zeilen / (1)=alphabetisch ..	zeilenweise
MEHRFACH-AUSGABE je Zeile	(RETURN)=ja / (1)=nein ..	mehrfach
TYP-ANZEIGE	(RETURN)=ja / (1)=nein ..	keine

- Felder werden in GROSSBUCHSTABEN angezeigt
 - DEFINT/DEFSTR sollten am Programm-ANFANG stehen
 - FUNKTIONS-NAMEN (z.B. FNuhr) werden als Variablen behandelt

REF Untersuchung von BASIC-PROGRAMMEN
Befehl PRINT (Token ABF)

(RETURN)=Variablen / (1)=Befehle mit Zeilennummern / (Befehl): **PRINT**

NAME des Programms	(RETURN)=vorhanden / (1)=laden	ref vorhanden
STARTZEILE	(RETURN)=erste	62000
ENDZEILE	(RETURN)=letzte	64650
AUSGABE auf	(RETURN)=Monitor / (1)=Drucker:	Drucker
DATUM		28.08.87

Abb. 2: Das Hauptmenue mit den verschiedenen Eingabemöglichkeiten.

Einzelheiten des Programmes

- Die Typdeklarationen (DEFINT...) werden verarbeitet sollten aber am Anfang des zu untersuchenden Programms stehen (wenn es noch nicht abgearbeitet wurde, sind alle Variablen ohne Typkennzeichen als Realvariablen abgelegt).
- Bei REM und ' wird der Rest der Zeile übersprungen.
- Nach " werden alle Zeichen überlesen, bis ein abschließendes " auftritt, oder das Zeilenende erreicht ist (somit führen weder Steuerzeichen in Texten noch Zeilen in der Form >100 LOAD"MC.OBJ< zu Fehlinterpretationen).
- Arrays werden in Großbuchstaben ausgegeben. REF erkennt sie bei der Untersuchung daran, daß nach dem Variablennamen eine Klammer kommt. Der Befehl ERASE wird gesondert behandelt, da auf ihn immer Feldvariablen folgen. Bei alphabetischer Variablenausgabe erscheinen die Arrays als Großbuchstaben vor den einfachen Variablen in Kleinbuchstaben.
- Um einen Basic-Befehl suchen zu lassen, ist der entsprechende, zu einem Token gehörende Befehl im Klartext einzugeben (z.B. > print <). Funktionen (z.B. chr\$) werden davon nicht erfaßt, ebenso wurde ON ERROR GOTO 0 (&B4) aus Platzgründen ausgelassen.
- Mit DEF FN selbstdefinierte Funktionen werden ebenfalls als Variablen interpretiert (schließlich müssen ihre Namen ja auch den Typdeklarationen entsprechen!).
- Fehleingaben werden weitestgehend aufgefangen!
- Begrenzungen:
 Zeilenweise können beliebig viele Variablen, Zeilennummern oder Befehle vorkommen. Bei geordneter Ausführung werden längere Variablennamen gekürzt und ein * angehängt. Ebenfalls ist die Zahl der Variablen/Zeilennummern durch die Arrays auf 101 begrenzt. Die Nummer jeder Zeile, in der REF ein Objekt findet, wird in 2 Byte kodiert und im Array VAR abgelegt. Somit darf jede Variable/Zielzeilennummer in 127 verschiedenen Zeilen auftreten (begrenzt durch die maximale Stringlänge). Damit dürfte es keine Schwierigkeiten geben. Lediglich bei sprungfreudigen Programmen besteht die Gefahr, daß zu viele verschiedene Zeilennummern angesprochen werden. In diesem Fall wird die Bearbeitung mit einer akustischen Meldung abgebrochen und die ermittelten Daten werden ausgegeben (auch bei zu vielen Variablen).

Bei größeren zu untersuchenden Programmen (in Richtung 30K) neigt der Computer zu der Meldung MEMORY FULL. In diesem Fall ist es am sinnvollsten, das Programm in zwei Etappen zu untersuchen.

```

10 REM layout.ref      28.8.87      [1862]
20 DEFSTR a,d-z        [497]
30 '                   [117]
40 x=CHR$(24)          [938]
50 '                   [117]
60 ' --- Kopf Bildschirmbild ----- [1923]
70 '                   [117]
80 t=x+SPACE$(5)+x+STRING$(35,131) [727]
90 t=t+CHR$(139)+CHR$(135)+STRING$(33,131) [841]
100 t=t+CHR$(139)+x+SPACE$(4)      [1318]
110 t=t+" REF "+x+" Untersuchung von BASIC [4777]
    -PROGRAMMEN "
120 t=t+CHR$(138)+CHR$(133)+CHR$(31) [614]
130 t=t+CHR$(76)+CHR$(2)+CHR$(138)+x+SPACE [3186]
    $(4)
140 t=t+SPACE$(5)+x+STRING$(35,140) [1008]
150 t=t+CHR$(142)+CHR$(141)+STRING$(33,140 [1966]
    )
160 t=t+CHR$(142)+x+SPACE$(4)+x    [1294]
170 CLS:PRINT t                    [436]
180 kopf=t                          [459]
190 '                                [117]
200 ' --- Text fuer Variablen ----- [1979]
210 '                                [117]
220 t=CHR$(31)+CHR$(2)+CHR$(21)+STRING$(7 [3292]
    7,45)
230 t=t+" - Felder werden in GROSSBUCHS [3313]
    TABEN angezeigt"
240 t=t+CHR$(31)+CHR$(3)+CHR$(23)    [1009]
250 t=t+"- DEFINT/DEFSTR sollten am Progra [3708]
    mm-ANFANG stehen"
260 t=t+CHR$(18)+CHR$(31)+CHR$(3)+CHR$(24) [2468]
270 t=t+"- FUNKTIONS-NAMEN (zB. FNuhr) wer [4849]
    den als Variablen behandelt"
280 PRINT t                          [462]
290 text=t                           [437]
300 '                                [117]
310 ' --- Ausfuehrung bestimmen ----- [2759]
320 '                                [117]
330 afg=CHR$(31)+CHR$(1)+CHR$(4)+x   [1327]
340 afg=afg+CHR$(18)+CHR$(10)+x+STRING$(80 [2085]
    ,131)
350 afg=afg+CHR$(31)+CHR$(6)+CHR$(4)+x [1952]
360 afg=afg+"(RETURN)=Variablen / (1)=Befe [6735]
    hie mit Zeilennummern / (Befehl): "+x
370 PRINT afg                         [428]
380 '                                [117]
390 '                                [117]
400 prog=CHR$(31)+CHR$(3)+CHR$(7)    [1751]
410 prog=prog+"NAME des Programms ..... [6594]
    ..... "
420 lad=CHR$(31)+CHR$(3)+CHR$(8)      [1434]
430 lad=lad+" (RETURN)=vorhan [3618]
    den / (1)=laden ..... "
440 start=CHR$(31)+CHR$(3)+CHR$(10)   [1601]
450 start=start+"STARTZEILE ..... [5219]
    (RETURN)=erste ..... "
460 ende=CHR$(31)+CHR$(3)+CHR$(11)   [1241]
470 ende=ende+"ENDZEILE ..... [3901]
    (RETURN)=letzte ..... "
480 aus=CHR$(31)+CHR$(3)+CHR$(12)+CHR$(10) [2193]
490 aus=aus+"AUSGABE auf ..... (R [5646]
    ETURN)=Monitor / (1)=Drucker: "
500 zeit=CHR$(31)+CHR$(24)+CHR$(14)+"DATUM [4726]
    ..... "
510 geord1=CHR$(31)+CHR$(3)+CHR$(15)  [2028]
520 geord1=geord1+"ORDNUNG ..... (RETURN)= [3980]
    nach Zeilen / (1)=alphabetisch .. "
530 einmal=CHR$(31)+CHR$(3)+CHR$(17)  [2420]
540 einmal=einmal+"MEHRFACH-AUSGABE je Zei [4210]
    le (RETURN)=ja / (1)=nein .. "
550 typanz=CHR$(31)+CHR$(3)+CHR$(18)  [2809]
560 typanz=typanz+"TYP-ANZEIGE ..... [5751]
    ... (RETURN)=ja ... / (1)=nein .. "
570 geord2=CHR$(31)+CHR$(3)+CHR$(15)  [2961]
580 geord2=geord2+"ORDNUNG ..... (RETURN)= [7981]
    nach Zeilen- / (1)=Zielnummern ... "
590 '                                [117]
600 ' --- String mit Token erstellen --- [2816]
610 su=STRING$(63,32)                [1002]
620 MID$(su,7)=CHR$(8C)+CHR$(85)+CHR$(8C [1983]
    0):'line
630 MID$(su,12)=CHR$(2)+CHR$(3)+CHR$(4)+CH [2598]
    R$(8B)+CHR$(8C):'Var.
640 MID$(su,18)=CHR$(8D)+CHR$(8E):'Zeile [3295]
    nnr.
650 MID$(su,24)=CHR$(89)+CHR$(8F)+CHR$(8 [2369]
    7C):'m+2

```

Listing REF

```

660 MID$(su,31)=CHR$(8B)+CHR$(8C):'m+3 [2850]
670 MID$(su,36)=CHR$(8F):'m+5 [1449]
680 MID$(su,48)=CHR$(8E)+CHR$(8F)+CHR$(8 [2059]
    90):'defint etc
690 MID$(su,54)=CHR$(89B):' ERASE [1430]
700 '                                [117]
710 GOSUB 970:' Liste mit Basic-Befehlen - [4029]
    --
720 '                                [117]
730 ' --- Erstellen der Datei --- [1940]
740 OPENOUT"ref.dat" [1492]
750 WRITE#9,kopf:' kopf [1328]
760 WRITE#9,afg:' Art [1518]
770 FOR b=0 TO 4:WRITE#9,kom(b):NEXT:' Bef [4129]
    ehls=Strings
780 '                                [117]
790 WRITE#9,prog:' Programm-Name [2155]
800 WRITE#9,lad:' Programm laden? [2409]
810 WRITE#9,start:' Startzeile [2199]
820 WRITE#9,ende:' Endzeile [1636]
830 WRITE#9,aus:' Stream [1774]
840 WRITE#9,zeit:' Datums-Eingabe [2035]
850 WRITE#9,su:' Such-Maske [2625]
860 '                                [117]
870 WRITE#9,text:' Text fuer Variablen [1484]
880 WRITE#9,geord1:' Ordnung Variablen [3064]
890 WRITE#9,einmal:' Mehrfach-Ausgabe [2310]
900 WRITE#9,typanz:' Typ-Anzeige [1531]
910 WRITE#9,geord2:' Ordnung Zeilennummern [3502]
920 '                                [117]
930 CLOSEOUT [902]
940 PRINT:PRINT" Datei REF.DAT erstellt [3226]
    ."
950 END [110]
960 '                                [117]
970 RESTORE 1040 [727]
980 FOR b=0 TO 4 [951]
990 FOR c=1 TO 28 [794]
1000 READ s:IF s="E" THEN RETURN [1647]
1010 kom(b)=kom(b)+s+STRING$(9-LEN(s),32 [2607]
    )
1020 NEXT c [381]
1030 NEXT b [378]
1040 DATA AFTER, AUTO, BORDER, CALL, CAT, CHAIN [5368]
    , CLEAR, CLG, CLOSEIN, CLOSEOUT, CLS, CONT
1050 DATA DATA, DEF, DEFINT, DEFREAL, DEFSTR, D [5163]
    EG, DELETE, DIM, DRAW, DRAWR, EDIT, ELSE
1060 DATA END, ENT, ENV, ERASE, ERROR, EVERY, FO [5589]
    R, GOSUB, GOTO, IF, INK, INPUT, KEY, LET
1070 DATA LINE, LIST, LOAD, LOCATE, MEMORY, MER [4021]
    GE, MIDS, MODE, MOVE, MOVER, NEXT, NEW, ON
1080 DATA ON BREAK, XX, ON SQ, OPENIN: 'XX sta [2811]
    tt ON ERROR GOTO 0
1090 DATA OPENOUT, ORIGIN, OUT, PAPER, PEN, PLO [2724]
    T, PLOT, POKE, PRINT, "", RAD, RANDOMIZE
1100 DATA READ, RELEASE, REM, RENUM, RESTORE, R [5146]
    BSUME, RETURN, RUN, SAVE, SOUND, SPEED
1110 DATA STOP, SYMBOL, TAG, TAGOFF, TROM, TROF [5043]
    F, WAIT, WEND, WHILE, WIDTH, WINDOW, ZONE
1120 DATA WRITE, DI, EI, FILL, GRAPHICS, MASK, F [5752]
    RAME, CURSOR, XX, ERL, FN, SPC, STEP, SWAP
1130 DATA XX, XX, TAB, THEN, TO, USING, >, =, < [3524]
    , <>, <=, +, -, *, /, ^, \
1140 DATA AND, MOD, OR, XOR, NOT [1275]
1150 DATA E [226]

```



```

62000 ' REF: Untersuchung von Basic-Progra [2640]
    mmen - Helmut Steeb 28.8.1987
62010 '                                [117]
62020 ' --- Vereinbarungen --- [1316]
62030 '                                [117]
62040 DEFINT a-k:DEFSTR p-z:dimmax=100 [2937]
62050 GOSUB 63770:' INIT [1854]

```

Listing REF

Anwendungen

```

62060 CLS:PRINT [356]
62070 PRINT#aus,"Basic-Programm: "prog; [3883]
62080 PRINT#aus,TAB(30)"(Zeilen"lnanf"--"ln [3781]
end;CHR$(8)""TAB(60)zeit
62090 PRINT#aus,tx:PRINT#aus, [1441]
62100 ' [117]
62110 ' --- Zeilenanfang --- [1111]
62120 ' [117]
62130 WHILE nozei [1209]
62140 lazei=PEEK(m)+256*PEEK(m+1):nozei=PE [4554]
EK(m+2)+256*PEEK(m+3):m=m+4
62150 IF nozei>lnend THEN nozei=0:' zum En [2987]
de
62160 IF nozei<lnanf THEN m=m+lazei ELSE G [2971]
OSUB 62280
62170 WEND [390]
62180 ' [117]
62190 ' --- ENDE --- [803]
62200 ' [117]
62210 GOSUB 63710:IF afg<2 AND geord THEN [3503]
GOSUB 62840
62220 IF geord=2 THEN PRINT:PRINT"Array VO [3738]
LL in Zeile"lnend
62230 PRINT#aus,:ZONE 13:WIDTH 132:END:' P [3320]
rogramm-ENDE
62240 ' [117]
62250 ' === UNTERPROGRAMME === [2446]
62260 ' [117]
62270 ' --- Zeilen-Interpretation --- [1523]
62280 inline=1:lnadr=m-4 [1278]
62290 IF (afg OR geord)=0 THEN anz(0)=0:an [4227]
z(1)=0:anz(2)=0
62300 WHILE NOT endezei [2470]
62310 a=PEEK(m):s=CHR$(a):m=m+1 [1825]
62320 ON INSTR(tokens,s)\6+1 GOSUB 62380,6 [4745]
2370,62630,62700,62390,62400,62410,62420,6
2460,62550,62790
62330 WEND [390]
62340 endezei=0:RETURN [955]
62350 ' [117]
62360 ' --- Bytes-Interpretation --- [591]
62370 m=lnadr+lazei:endezei=-1:RETURN:' nae [4338]
chste Zeile
62380 RETURN:' (m+1) [1274]
62390 m=m+1:IF a<>124 THEN RETURN:' (m+2) [3505]
1-Byte, Funktion
62395 WHILE PEEK(m-1)<128:m=m+1:WEND:RETUR [2798]
N:'!sx
62400 m=m+2:RETURN:' (m+3) 2-Byte [1372]
62410 m=m+5:RETURN:' (m+6) Fliesskomma [1716]
62420 WHILE PEEK(m)<>34 AND PEEK(m):m=m+1: [3466]
WEND:'Text in"
62430 endezei=NOT PEEK(m):m=m+1:RETURN [2471]
62440 ' [117]
62450 ' --- Defint/Defstr/Defreal --- [3067]
62460 m=m+1:b=44:art=a-&8E:IF art THEN art [3153]
=3-art
62470 WHILE (b OR 1)=45 [1387]
62480 a=(PEEK(m)OR 32)-96:b=PEEK(m+1):m=m+ [2905]
2:'a=Buchstabe,b=, - 0 1
62490 IF b=44 THEN MID$(typen,a)=CHR$(art) [2596]
:'komma
62500 IF b=45 THEN MID$(typen,a)=STRING$( [2744]
PEEK(m)OR 32)-95-a,art):' -
62510 WEND [390]
62520 endezei=NOT b:RETURN [1840]
62530 ' [117]
62540 ' ---- Erase --- [492]
62550 WHILE PEEK(m)>1 [1421]
62560 a=PEEK(m+1):m=m+4:GOSUB 63120:'Name [2930]
lesen
62570 v=UPPER$(v) [1002]
62580 GOSUB 63180:'verarbeiten [1772]
62590 WEND:' Komma:nochmal; 0/1:weiter [1823]
62600 RETURN [555]
62610 ' [117]
62620 ' --- Variable --- [1455]
62630 m=m+2:GOSUB 63120:'Name lesen [1893]
62640 WHILE PEEK(m+1)=32:m=m+1:WEND:'Space [1951]
ueberlesen
62650 IF PEEK(m+1)<>82 THEN v=LOWER$(v) E [2680]
LSE v=UPPER$(v):m=m+1:' ( bei Feld
62660 GOSUB 63180:'verarbeiten [1772]
62670 RETURN [555]
62680 ' [117]
62690 ' ---Zeilennummer --- [1283]
62700 IF afg<>1 THEN m=m+2:RETURN:' ueberle [2744]
sen
62710 lnziel=PEEK(m)+256*PEEK(m+1):m=m+2:' [2585]

```

Listing REF

```

Zeilennr. lesen
62720 IF a=&1D THEN lnziel=PEEK(lnziel+3)+ [4189]
256*(PEEK(lnziel+4))
62730 IF geord THEN GOSUB 63390:RETURN:'Li [2207]
ste durchsuchen+ablegen
62740 IF inline THEN PRINT#aus,:PRINT#au [3732]
s,USING " #####: ";nozei,:inline=0
62750 PRINT#aus,USING" #####";lnziel; [1682]
62760 RETURN [555]
62770 ' [117]
62780 ' --- Befehl --- [1159]
62790 IF lninter=nozei THEN RETURN [1700]
62800 lninter=nozei:PRINT#aus,USING " [3395]
#####";nozei;
62810 RETURN [555]
62820 ' [117]
62830 ' --- Ausgabe bei geord --- [2095]
62840 CLS#2:IF afg THEN 63000 [593]
62850 FOR a=0 TO eno:change=a:' Variablen [2338]
---
62860 FOR b=a+1 TO eno [849]
62870 IF varia(change)>varia(b) THEN cha [2104]
nge=b
62880 NEXT b [378]
62890 IF change=a THEN 62920 [1107]
62900 zwi=varia(a):varia(a)=varia(change [3787]
):varia(change)=zwi
62910 inter=index(a):index(a)=index(chan [3783]
ge):index(change)=inter
62920 art=ASC(varia(a)):IF art<>altart THE [4506]
N GOSUB 63620
62930 s=MID$(varia(a),2):PRINT#aus,SPC(9-L [2627]
EN(s))LEFT$(s,9)::'spc(-1)=""
62940 IF LEN(s)>9 THEN PRINT#aus,"*"; ELSE [3956]
PRINT#aus," ";
62950 GOSUB 63530 [1020]
62960 NEXT a [383]
62970 art=2:GOSUB 63620 [74]
62980 RETURN [555]
62990 ' [117]
63000 FOR a=0 TO eno:change=a:' Zeilennr. [2719]
in Befehlen ---
63010 FOR b=a+1 TO eno [849]
63020 IF ln(change)>ln(b) THEN change=b [3178]
63030 NEXT b [378]
63040 IF change=a THEN 63070 [1227]
63050 lninter=ln(a):ln(a)=ln(change):ln( [3056]
change)=lninter
63060 inter=index(a):index(a)=index(chan [3783]
ge):index(change)=inter
63070 PRINT#aus,USING" ##### in ";ln(a); [1965]
63080 GOSUB 63530 [1020]
63090 NEXT a:RETURN [987]
63100 ' [117]
63110 ' --- Variablen-Name lesen --- [2270]
63120 IF a>10 THEN art=ASC(MID$(typen,(PEE [4389]
K(m)OR &EO)-&EO) ELSE art=a-2
63130 v="":WHILE PEEK(m)<128:v=v+CHR$(PEEK [3722]
(m)):m=m+1:WEND:'lesen
63140 v=v+CHR$(PEEK(m)AND 127) [2594]
63150 RETURN [555]
63160 ' [117]
63170 ' --- Variablen-Name verarbeiten --- [3581]
63180 m=m+1:IF afg THEN RETURN [1569]
63190 IF geord THEN GOSUB 63310:RETURN:'Li [3346]
ste durchsuchen+ablegen
63200 ' [117]
63210 FOR b=0 TO anz(art)-1:IF v<>var(b,ar [3619]
t) THEN NEXT:GOTO 63230:'nicht vorh.
63220 IF einmal THEN RETURN ELSE 63240 [2826]
:'
vorhanden
63230 var(anz(art),art)=v:anz(art)=anz(a [3135]
rt)+1:'ablegen
63240 IF inline=0 THEN PRINT#aus," ";:GOT [1993]
O 63260
63250 PRINT#aus,:PRINT#aus,USING" #####: [1975]
";nozei,:inline=0
63260 PRINT#aus,v;:IF anzeig THEN RETURN [3595]
63270 IF art=0 THEN PRINT#aus,"%"; ELSE IF [3140]
art=1 THEN PRINT#aus,"$";
63280 RETURN [555]
63290 ' [117]
63300 ' --- Liste durchsuchen+ablegen (Var [2699]
iablen) ---
63310 PRINT#2,CHR$(13)nozei;:v=CHR$(art)+v [939]
63320 FOR a=0 TO eno:IF v=varia(a) THEN 63 [4289]
360:'Variable schon vorhanden
63330 NEXT [350]
63340 IF eno=dimax THEN lnend=nozei:noze [5252]

```

Listing REF

```

i=0:endezei=-1:geord=2:ab=1:RETURN
63350 eno=eno+1:varia(eno)=v [2524]
63360 GOSUB 63470:RETURN [1531]
63370 ' [117]
63380 ' --- Liste durchsuchen+ablegen (Zei [3021]
lennummern) ---
63390 PRINT#2,CHR$(13)nozei; [606]
63400 FOR a=0 TO eno:IF lnziel=ln(a) THEN [4940]
63440:'Zielnr. schon vorhanden
63410 NEXT [350]
63420 IF eno=dimmax THEN lnend=nozei:noze [5252]
i=0:endezei=-1:geord=2:ab=1:RETURN
63430 eno=eno+1:ln(eno)=lnziel [1734]
63440 GOSUB 63470:RETURN [1531]
63450 ' [117]
63460 ' --- Zeilennr.zum String --- [2394]
63470 hi=INT(nozei/256) [1348]
63480 s=CHR$(nozei-256*hi)+CHR$(hi) [1172]
63490 IF s<>RIGHT$(var(a),2) THEN var(a)=v [3086]
ar(a)+s:' nur 1x ablegen
63500 RETURN [555]
63510 ' [117]
63520 ' --- Zeilennummern aus bei geord -- [2303]
-
63530 v=var(index(a)) [1511]
63540 FOR c=1 TO LEN(v) STEP 2 [2026]
63550 IF (c-(c>1)) MOD 2=2 THEN PRINT#a [4307]
us,:PRINT#aus,SPACE$(10);
63560 PRINT#aus,STRING"#####";ASC(MID$(v [3135]
,c))+256*ASC(MID$(v,c+1));
63570 NEXT c [381]
63580 PRINT#aus, [907]
63590 RETURN [555]
63600 ' [117]
63610 ' --- Typ-Ueberschrift --- [2726]
63620 FOR c=altart+1 TO art [1253]
63630 PRINT#aus,:PRINT#aus,typ(c); [1570]
63640 IF c<art OR a>eno THEN PRINT#aus, "- [4298]
":GOTO 63660
63650 PRINT#aus,TAB(35)STRING$(10,45) [2227]
63660 PRINT#aus, [907]
63670 NEXT [350]
63680 altart=art:RETURN [1000]
63690 ' [117]
63700 'Sound -- [761]
63710 ENT 1,3,ab*50,24,1,ab*74,75 [1343]
63720 SOUND 1,450,150,15,1 [1411]
63730 RETURN [555]
63740 ' [117]
63750 ' --- INIT --- [1083]
63760 ' [117]
63770 MODE 2:WINDOW#1,43,74,2,2:WINDOW#2,1 [2829]
1,37,5,5
63780 OPENIN"ref.dat":INPUT#9,rahmen:PRINT [3732]
rahmen:rahmen=""
63790 ' [117]
63800 feflag=0:emax=9:GOSUB 64450:IF t<>"" [2644]
AND t<>"1" THEN 63820
63810 afg=VAL(t):ein=5:GOSUB 64650:GOTO [2764]
63880:'Var./Zeilenr.
63820 t=UPPER$(t):wort=t:' Basic-Befehl [2604]
63830 FOR a=0 TO 4 [728]
63840 INPUT#9,such:b=INSTR(such,t):IF b AN [4685]
D afg=0 THEN afg=80+b*9+a*28
63850 NEXT [350]
63860 IF afg=0 THEN PRINT#2,"Befehl unguel [4956]
tig! (TASTE)":CALL &BB06:RUN 62000
63870 ' [117]
63880 IF afg>127 THEN PRINT#1,"Befehl "wor [3814]
t" (Token &"HEX$(afg))":GOTO 63910
63890 IF afg THEN PRINT#1,"ZEILENUMMERN [6119]
in Befehlen" ELSE PRINT#1,"VARIABLEN"
63900 ' [117]
63910 feflag=1:emax=8:GOSUB 64450:prog=UPP [2325]
ER$(t)
63920 ' [117]
63930 feflag=2:text0="vorhanden":text1="la [4283]
den":GOSUB 64450:da=(t="")
63940 ' [117]
63950 feflag=3:emax=5:text0="erste":text1= [4507]
"":GOSUB 64450
63960 IF m=0 THEN lnanf=1 ELSE lnanf=m [2050]
63970 ' [117]
63980 text0="letzte":GOSUB 64450:IF m=0 TH [3391]
EN lnend=60000 ELSE lnend=m
63990 ' [117]
64000 feflag=2:text0="Monitor":text1="Druc [2962]
ker":GOSUB 64450:aus=8*m
64010 IF aus=0 THEN INPUT#9,s:GOTO 64040 [1360]

```

Listing REF

```

64020 feflag=1:emax=10:GOSUB 64450:zeit= [1203]
t:'Datum
64030 ' [117]
64040 INPUT#9,tokens [1215]
64050 MID$(tokens,6)=CHR$(0):MID$(tokens,1 [2559]
7)=CHR$(13)
64060 MID$(tokens,30)=CHR$(26):MID$(tokens [3543]
,42)=CHR$(34)
64070 IF afg>127 THEN GOSUB 64190 ELSE ON [3845]
afg+1 GOSUB 64250,64380:'jew. INIT
64080 m=368:nozei=1:ab=-1:typen=STRING$(26 [3911]
,2):ZONE 8:WIDTH 80
64090 IF da THEN CLOSEIN:RETURN [2019]
64100 LOCATE 37,20:PRINT CHR$(24)"(TASTE) [2658]
= Programm laden"CHR$(24):CALL &BB06
64110 CHAIN MERGE prog,62060,DELETE -6199 [1064]
9
64120 ' [117]
64130 ' --- INIT geord --- [1847]
64140 DIM index(dimmax),var(dimmax) [1828]
64150 FOR a=0 TO dimmax:index(a)=a:NEXT [2544]
64160 eno=-1:RETURN [760]
64170 ' [117]
64180 ' --- INIT BEFEHL --- [1280]
64190 s=CHR$(afg):b=INSTR(tokens,s):tokens [2641]
=token+s
64200 IF b THEN MID$(tokens,b)=CHR$(0) [610]
64210 tx="Befehl "+wort+" (Token &"HEX$(a [2086]
fg)+"")"
64220 RETURN [555]
64230 ' [117]
64240 ' --- INIT VARIABLE --- [1272]
64250 INPUT#9,text:PRINT text:text="":' E [2912]
rklaerung
64260 feflag=2:text0="zeilenweise":text1=" [3909]
":GOSUB 64450:geord=m
64270 tx="Variablen, ":IF geord THEN 64320 [2678]
64280 text0="mehrfach":text1="einmal":GO [4522]
SUB 64450:einmal=m
64290 text0="Anzeige":text1="keine":GOSU [4072]
B 64450:anzeig=m
64300 tx=tx+"zeilenweise":DIM var(20,2) [2268]
64310 RETURN [555]
64320 GOSUB 64140:DIM varia(dimmax):altart [3029]
=-1:' geord
64330 typ(0)="INTEGER..":typ(1)="STRING.. [2383]
..":typ(2)="REAL...."
64340 tx=tx+"alphabetisch" [2727]
64350 RETURN [555]
64360 ' [117]
64370 ' --- INIT ZEILENUMMER --- [2467]
64380 ein=4:GOSUB 64650 [1175]
64390 feflag=2:text0="zeilenweise":text1=" [4993]
geordnet":GOSUB 64450:geord=m
64400 IF geord=0 THEN tx=" Zeile: Ziel [2088]
":GOTO 64420
64410 tx=" Ziel in Zeile":GOSUB 64140: [1885]
DIM ln(dimmax)
64420 tx=CHR$(10)+tx:RETURN [1598]
64430 ' [117]
64440 ' --- Abfrage --- [1068]
64450 fe=1:INPUT#9,quest [1212]
64460 WHILE fe [856]
64470 PRINT quest; [857]
64480 IF feflag THEN PRINT CHR$(18); [1314]
64490 INPUT,"",t:IF t<>"" THEN m=ASC(t) EL [2315]
SE m=48
64500 IF feflag>1 AND(m<48 OR m>57) THEN 6 [1684]
4530
64510 ON feflag+1 GOSUB 64610,64590,64600, [2194]
64610
64520 IF LEN(t)>emax THEN fe=1 [1470]
64530 WEND [390]
64540 IF feflag>1 THEN m=VAL(t) [1617]
64550 RETURN [555]
64560 ' [117]
64570 ' --- Fehlerbehandlung bei Eingabe - [4130]
--
64580 fe=0:RETURN:' feflag=0 [2032]
64590 fe=(t=""):RETURN:' feflag=1 - kein " [2376]
"
64600 IF t<>"" AND t<>"1" THEN RETURN:' fe [3081]
flag=2 - nur () / (1)
64610 fe=0:IF t="" THEN PRINT text0 ELSE I [3181]
F t="1" THEN PRINT CHR$(8)text1
64620 RETURN [555]
64630 ' [117]
64640 ' --- INPUT#9, zur Korrektur --- [2325]
64650 FOR a=1 TO ein:INPUT#9,s:NEXT:RETURN [2576]

```

Listing REF

COMEX - Compressor/Expander

In einer der letzten Ausgaben von PC-Schneider International war ein Programm abgedruckt, welches BASIC-Programme verdichtet. Es ist selbst in BASIC geschrieben und daher dauert es sehr lange, bis ein Programm komprimiert ist. Bei langen Programmen kann dies schon an die 20 Minuten laufen.

In diesem Zusammenhang möchte ich Ihnen COMEX vorstellen. Hierbei handelt es sich um einen BASIC-COMPRESSOR und BASIC-EXPANDER. Diese Routinen sind in Maschinensprache geschrieben und daher sehr schnell. Eine genaue Beschreibung erscheint im Text weiter unten. Dieses Anwendungsprogramm verdichtet Basicprogramme auf die kleinstmögliche Version unter Berücksichtigung der vom Anwender gewünschten maximalen Zeilenlänge. Der Anwender kann, um zwei Beispiele zu nennen, sich entscheiden, ob er nur REM-Zeilen entfernen oder nur Datazeilen verdichten möchte, oder auch beides; es ist ganz ihm selbst überlassen. Er kann auch ein verdichtetes Basicprogramm wieder vollständig listbar machen, um es eventuell abzuändern, zu überprüfen oder zu erweitern.

Die Vorteile von COMEX

COMEX bietet dem Benutzer folgende Vorteile:

- größtenteils Menuesteuerung
- sekundenschnelle Abarbeitung
- komprimiertes Programm sofort lauffähig.

Des weiteren sind kürzere Ladezeiten (besonders für Cassettenbenutzer interessant), mehr Speicherplatz auf Diskette und Kassette und eine schnellere Abarbeitung des komprimierten Programmes vorgesehen.

Handhabung von COMEX

Durch RUN "COMEX" wird das Maschinenprogramm geladen und nach Eingabe des Rechnertyps das Basicprogramm automatisch gelöscht. Das Maschinenprogramm ist ab sofort auf Ihren Rechner eingestellt und nun kann ein beliebiges BASIC-Programm, welches Sie verdichten wollen, geladen werden. Durch Druck auf die kleine Entertaste wird die Maschinencode-Routine angesprungen und es erscheint ein Menue. Hier wird die maximale Zeilenlänge, die das verdichtete Programm haben soll, ermittelt.

Drücken Sie z.B. die Taste »1«, so beträgt die maximale Zeilenlänge 180 Bytes und das Programm läßt sich noch vollständig auflisten. Durch Drücken der Taste »5« (maximale Komprimierung) kann natürlich nicht mehr das ganze Programm gelistet werden, ein Teil des Programmes ist unsichtbar. Deswegen wurde auch ein BASIC-EXPANDER eingebaut, der im Anschluß erklärt wird. Nach Drücken der gewünschten Taste erscheint ein neues Menue.

SYNTAX-CHECK:

Dieser Programmpunkt dient zur Überprüfung der Anführungszeichen im Programm. Steht am Ende einer Basiczeile ein String

und wurde das Anführungszeichen ausgelassen, so erkennt die zwar der Interpreter bei der Ausführung des Programmes nicht mehr aber, wenn das Programm komprimiert wurde. SYNTAX-CHECK listet diese fehlerhaften Programmzeilen auf, und der Anwender kann diese Zeilen auf die übliche Art und Weise editieren. Danach kann COMEX wieder mit der kleinen Entertaste gestartet werden.

REMARKS entfernen: Hier werden alle Zeilen entfernt, die mit "REM", ":" oder ";" beginnen. Sollten REM-Zeilen durch GOTO etc. angesprungen werden, so wird die Zeilennummer in der betreffenden Programmzeile berichtigt. Die Zeilennummer nach GOTO etc. zeigt nun auf die nächsthöhere Programmzeile nach der vorerst angesprungenen REM-Zeile.

SPACE entfernen: Die Adresse &AC00 beherbergt ein Flag für Spaces bei Token-Wandlung. Im Normalfall enthält diese Adresse eine Null, das heißt, alle Spaces werden übernommen. Durch POKE &AC00,1 vor der Eingabe eines Programmes werden die überflüssigen Spaces unterdrückt und man gewinnt dadurch Speicherplatz. Diese Spaceunterdrückung wirkt aber nur, wenn VOR der Eingabe ein POKE durchgeführt wurde. Eine nachträgliche Veränderung dieses Flags hat keine Wirkung. COMEX simuliert diese Routine und dadurch können auch nachträglich alle überflüssigen Spaces entfernt werden.

BASICZEILEN komprimieren: In diesem Teil des Programmes werden Multistatementzeilen erzeugt, d.h., die der aktuellen Programmzeile folgende Basiczeile wird der aktuellen Zeile angehängt. Dabei werden die Verwaltungsbytes der zweiten Zeile entfernt und der Rest des Programmes durch die Blockverschieberoutinen an das Ende der aktuellen Programmzeile geschoben. Hierbei werden die maximale Zeilenlänge und solche Zeilen berücksichtigt, die durch GOTO etc. angesprungen werden.


DATAZEILEN komprimieren: In diesem Programmabschnitt werden nur Datazeilen, natürlich auch unter Berücksichtigung der gewünschten maximalen Zeilenlänge, verdichtet.

BASIC-EXPANDER: Um das verdichtete Programm wieder vollständig sichtbar zu machen, wird das Programm in kurze Statements zerpfückt. Beim Listen erkennt man diese Zeilen durch eine »1« als Zeilennummer. Durch RENUM werden diese Zeilen angepaßt.

HINWEIS: Nach jedem Programmpunkt erscheint ein Hinweis auf die Anzahl der eingesparten Bytes. Um eine optimale Verdichtung des Basicprogrammes zu erzielen, sollten die Menüpunkte chronologisch abgearbeitet werden. Das Programm ist nach der jeweiligen Abarbeitung sofort lauffähig. Sollte ein bearbeitetes Programm bereits durch RUN gestartet worden sein und wollen Sie noch Änderungen vornehmen, so muß das Programm mit NEW gelöscht und neu geladen werden, da sich sonst Fehler bei der Komprimierung einschleichen und das Programm nicht lauffähig wird.

(R. Schäffer/JB)

für 464-664-6128



```

10 REM ***** [2887]
*****
15 REM *** B A S I C --- C O M P R E [2062]
   S O R   ( C )           ***
17 REM *** [417]
           ***
20 REM ***   written by SCHAEFFER ROLAND [2400]
Listing: Comex
    
```

```

A-5020 SALZBURG           ***
22 REM *** [417]
           ***
25 REM ***** [2887]
*****
60 ' [117]
70 REM INITIALISIERUNG [2397]
80 ' [117]
90 SYMBOL AFTER 256:MEMORY &8FFF:SYMBOL AF [2228]
TER 32
100 KEY 139,"call &9000"+CHR$(13):LOAD "co [8869]
mex.bin":MODE 1:a$="B A S I C - C O M P
Listing: Comex
    
```



```

R E S S O R (C)":bs="C P C 464 --->
1":cs="C P C 664 ---> 2"
200 ds="C P C 6128 ---> 3":ts="BITTE [5920]
RECHNERTYP EINGEBEN":LOCATE 21-LEN(as)/2,2
:PRINT as:LOCATE 21-LEN(bs)/2,8:PRINT bs:L
OCATE 21-LEN(cs)/2,11:PRINT cs:LOCATE 21-L
EN(ds)/2,14:PRINT ds
300 LOCATE 21-LEN(ts)/2,22:PRINT ts [917]
320 ' [117]
325 REM TATATURABFRAGE [1584]
330 ' [117]
400 CALL &BB06:CALL &BB09:IF INKEY(64)=0 T
HEN 900 ELSE IF INKEY(65)=0 THEN 500 ELSE
IF INKEY(57)=0 THEN 600 ELSE 400
480 ' [117]
485 REM POKES FUER CPC 664 [967]
487 ' [117]
500 POKE &9529,&E8:GOTO 700 [784]
580 ' [117]
585 REM POKES FUER CPC 6128 [1071]
590 ' [117]
600 POKE &9529,&E3 [674]
680 ' [117]
690 'POKES FUER CPC 464,664,6128 [1104]
695 ' [117]
700 POKE &9508,&64:POKE &9532,&64:POKE &95
59,&64:POKE &95A5,&64:POKE &963D,&64:POKE
&96C1,&64:POKE &97A7,&64:POKE &98A8,&64
800 POKE &9001,&66:POKE &9213,&66:POKE &96
C4,&66:POKE &9750,&66:POKE &9766,&66:POKE
&97A1,&66:POKE &987E,&66:POKE &98A2,&66:PO
KE &9925,&66
850 ' [117]
900 MODE 2:a$="Bitte gewünschtes Programm [16107]
laden":bs="danach":cs="Start mit KLEINER
ENTER-TASTE":LOCATE 41-LEN(as)/2,5:PRINT a
$:LOCATE 41-LEN(bs)/2,9:PRINT bs:LOCATE 41
-LEN(cs)/2,13:PRINT cs:NEW

```

```

100 'REM DATA - LADER COMEX.BIN [2044]
110 ' [117]
120 'Ladeadresse: &9000 [645]
130 'Laenge: &095F [363]
140 'Startadresse: &0000 [1303]
150 ' [117]
160 MEMORY &9000-1:zei=270 [1594]
170 FOR i=&9000 TO &995F STEP 8 [2448]
180 sum=0:FOR j=0 TO 7 [1044]
190 READ a$:a=VAL("&"+a$):sum=sum+a:PO
KE j+1,a [2315]
200 NEXT j [370]
210 READ a$:a=VAL("&"+a$) [883]
220 IF a<>sum THEN PRINT"DATAFEHLER in Z [3034]
eile"zei:END
230 zeiz=zei+10 [393]
240 NEXT i [375]
250 SAVE"COMEX .BIN",B,&9000,&95F,&0 [2040]
260 ' [117]
270 DATA 2A,83,AE,22,5D,99,AF,32,0354 [1246]
280 DATA 47,99,3E,02,CD,0E,BC,21,02D8 [1999]
290 DATA 01,0F,CD,75,BB,21,66,92,0326 [1310]
300 DATA CD,56,92,21,05,05,CD,75,0322 [860]
310 DATA BB,21,C9,92,CD,56,92,21,040D [1513]
320 DATA 08,19,CD,75,BB,21,E1,92,03B2 [888]
330 DATA CD,56,92,21,0A,19,CD,75,033B [1734]
340 DATA BB,21,F0,92,CD,56,92,21,0434 [1403]
350 DATA 0C,19,CD,75,BB,21,FF,92,03D4 [2003]
360 DATA CD,56,92,21,0E,19,CD,75,033F [1741]
370 DATA BB,21,0E,93,CD,56,92,21,0353 [1895]
380 DATA 10,19,CD,75,BB,21,1D,93,02F7 [1215]
390 DATA CD,56,92,21,14,19,CD,75,0345 [2031]
400 DATA BB,21,2C,93,CD,56,92,CD,041D [1375]
410 DATA 60,92,CD,06,BB,FE,31,28,03D7 [981]
420 DATA 16,FE,32,28,17,FE,33,28,02DE [2335]
430 DATA 16,FE,34,28,19,FE,35,28,02E6 [865]
440 DATA 1A,FE,39,28,1B,1E,E3,21,02B0 [1764]
450 DATA B4,00,18,5E,21,BE,00,18,0221 [1407]
460 DATA 59,21,C8,00,18,54,21,D2,02A1 [1260]
470 DATA 00,18,4F,21,60,EA,18,4A,0234 [1601]
480 DATA 3E,02,CD,0E,BC,21,01,0F,0208 [2449]
490 DATA CD,75,BB,21,99,92,CD,56,046C [1495]
500 DATA 92,21,08,19,CD,75,BB,21,02F2 [1234]
510 DATA FB,93,CD,56,92,21,0A,19,0387 [1392]
520 DATA CD,75,BB,21,1C,94,CD,56,03F1 [1822]
530 DATA 92,21,0C,19,CD,75,BB,21,02F6 [1139]
540 DATA E0,93,CD,56,92,CD,60,92,04E7 [1703]

```

Listing: Comex

```

550 DATA CD,06,BB,FE,31,CA,B4,91,04CC [1584]
560 DATA FE,32,CA,D7,91,FE,36,C8,055E [1348]
570 DATA 18,EE,22,59,99,3E,02,CD,0327 [1723]
580 DATA 0E,BC,21,01,0F,CD,75,BB,02F8 [1698]
590 DATA 21,66,92,CD,56,92,21,07,02F6 [1820]
600 DATA 0A,CD,75,BB,21,4E,93,CD,03D6 [1476]
610 DATA 56,92,21,09,0A,CD,75,BB,0319 [854]
620 DATA 21,65,93,CD,56,92,21,0B,02FA [1888]
630 DATA 0A,CD,75,BB,21,81,93,CD,0409 [1510]
640 DATA 56,92,21,0D,0A,CD,75,BB,031D [1463]
650 DATA 21,9B,93,CD,56,92,21,0F,0334 [1970]
660 DATA 0A,CD,75,BB,21,BE,93,CD,0446 [1837]
670 DATA 56,92,21,11,0A,CD,75,BB,0321 [1374]
680 DATA 21,E0,93,CD,56,92,CD,60,0476 [2020]
690 DATA 92,CD,06,BB,FE,31,28,17,038E [1740]
700 DATA FE,32,28,27,FE,33,28,2B,0303 [1511]
710 DATA FE,34,28,2F,FE,35,28,33,0317 [1434]
720 DATA FE,36,CA,FB,91,18,E2,21,04A5 [788]
730 DATA 12,01,CD,75,BB,CD,87,94,03F8 [1706]
740 DATA 3A,47,99,FE,01,C8,21,07,0309 [2168]
750 DATA 3C,18,1E,CD,58,92,21,09,0256 [1502]
760 DATA 3C,18,16,CD,31,95,21,0B,0229 [1600]
770 DATA 3C,18,0E,CD,A4,95,21,0D,0296 [1160]
780 DATA 3C,18,06,CD,26,96,21,0F,0213 [1865]
790 DATA 3C,CD,75,BB,21,F7,93,CD,04B1 [1597]
800 DATA 56,92,3E,07,CD,5A,BB,CD,03DC [1827]
810 DATA FB,91,18,9A,21,0F,19,CD,0354 [338]
820 DATA 75,BB,21,69,94,CD,56,92,0403 [1834]
830 DATA CD,99,97,21,08,3C,CD,75,03A4 [1930]
840 DATA BB,21,F7,93,CD,56,92,3E,0459 [1970]
850 DATA 07,CD,5A,BB,C3,E0,90,21,043D [2048]
860 DATA 0F,19,CD,75,BB,21,69,94,0343 [1435]
870 DATA CD,56,92,CD,9C,98,21,0A,03E1 [1534]
880 DATA 3C,CD,75,BB,21,F7,93,CD,04B1 [1597]
890 DATA 56,92,3E,07,CD,5A,BB,C3,03D2 [1811]
900 DATA E0,90,C9,21,15,03,CD,75,03B4 [1195]
910 DATA BB,21,3D,94,CD,56,92,21,0383 [1308]
920 DATA 15,1A,CD,75,BB,AF,2A,5D,0362 [2194]
930 DATA 99,ED,5B,83,AE,ED,52,F5,0546 [1480]
940 DATA D5,E5,C5,CB,80,11,10,27,0412 [1493]
950 DATA CD,40,92,11,E8,03,CD,40,03A8 [1427]
960 DATA 92,11,64,00,CD,40,92,1E,02C4 [1206]
970 DATA 0A,CD,40,92,CB,C0,1E,01,0353 [1714]
980 DATA CD,40,92,C1,E1,D1,F1,C9,05CC [1188]
990 DATA AF,ED,52,38,03,3C,18,F9,0376 [1690]
1000 DATA 19,B7,20,03,CB,40,C8,CB,0391 [1144]
1010 DATA C0,C6,30,C3,5A,BB,7E,23,042F [1521]
1020 DATA FE,00,C4,5A,BB,20,F7,C9,04B7 [1196]
1030 DATA CD,09,BB,38,FB,C9,42,41,0410 [1596]
1040 DATA 53,49,43,20,2D,20,43,4F,01DE [2086]
1050 DATA 4D,50,52,45,53,53,4F,52,027B [1793]
1060 DATA 20,20,2D,20,20,77,72,69,01FF [1645]
1070 DATA 74,74,65,6E,20,62,79,20,02D6 [1733]
1080 DATA 53,43,48,41,45,46,46,45,0235 [956]
1090 DATA 52,20,52,4F,4C,41,4E,44,0232 [1101]
1100 DATA 00,42,41,53,49,43,20,2D,01AF [1576]
1110 DATA 20,45,58,50,41,4E,44,45,0225 [1563]
1120 DATA 52,20,2D,20,77,72,69,74,0285 [1448]
1130 DATA 74,65,6E,20,62,79,20,53,02B5 [2052]
1140 DATA 43,48,41,45,46,46,45,52,0234 [1466]
1150 DATA 20,20,52,4F,4C,41,4E,44,0200 [1819]
1160 DATA 00,4D,41,58,49,4D,41,4C,0209 [1895]
1170 DATA 45,20,5A,45,49,4C,45,4E,022C [2332]
1180 DATA 4C,41,45,4E,47,45,20,3A,0206 [1402]
1190 DATA 00,31,38,30,20,20,2D,0133 [880]
1200 DATA 2D,2D,2D,3E,20,20,31,00,0136 [1299]
1210 DATA 31,39,30,20,20,2D,2D,0161 [1780]
1220 DATA 2D,2D,3E,20,20,32,00,32,013C [1558]
1230 DATA 30,30,20,20,2D,2D,2D,0154 [1614]
1240 DATA 2D,3E,20,20,33,00,32,31,0141 [1368]
1250 DATA 30,20,20,2D,2D,2D,0151 [1107]
1260 DATA 3E,20,20,34,00,4D,41,5E,0198 [1717]
1270 DATA 2E,20,2D,2D,2D,2D,2D,016D [1003]
1280 DATA 20,20,35,00,39,20,20,2D,011B [1771]
1290 DATA 2D,2D,2D,2D,20,20,42,41,0177 [1109]
1300 DATA 53,49,43,20,2D,20,45,20,01B1 [2016]
1310 DATA 58,20,50,20,41,20,4E,20,01B7 [1549]
1320 DATA 44,20,45,20,52,00,31,20,016C [2088]
1330 DATA 20,2D,2D,2D,2D,2D,20,20,0141 [2129]
1340 DATA 53,59,4E,54,41,58,2D,43,0257 [1319]
1350 DATA 48,45,43,4B,00,32,20,20,018D [1501]
1360 DATA 2D,2D,2D,2D,2D,20,20,52,0173 [1101]
1370 DATA 45,4D,41,52,4B,53,20,65,0248 [1682]
1380 DATA 6E,74,66,65,72,6E,65,6E,0360 [2048]
1390 DATA 00,33,20,20,2D,2D,2D,0127 [1058]
1400 DATA 2D,20,20,52,50,41,43,45,01D9 [1816]
1410 DATA 20,65,6E,74,66,65,72,6E,0312 [1692]
1420 DATA 65,6E,00,34,20,20,2D,2D,01A1 [1673]

```

Listing: Comex

Anwendungen

1430 DATA 2D,2D,2D,20,20,42,41,53,019D
 1440 DATA 49,43,5A,45,49,4C,45,4B,0253
 1450 DATA 20,6B,6F,6D,70,72,69,6D,031F
 1460 DATA 69,65,72,65,6E,00,35,20,0268
 1470 DATA 20,2D,2D,2D,2D,2D,20,20,0141
 1480 DATA 44,41,54,41,5A,45,49,4C,024E
 1490 DATA 45,4E,20,6B,6F,6D,70,72,02DC
 1500 DATA 69,6D,69,65,72,65,6E,00,02E9
 1510 DATA 36,20,20,2D,2D,2D,2D,0157
 1520 DATA 20,20,50,52,4F,47,52,41,020B
 1530 DATA 4D,4D,45,4E,44,45,00,4F,0205
 1540 DATA 4B,2E,00,31,20,20,2D,2D,0144
 1550 DATA 2D,2D,2D,20,20,42,41,53,019D
 1560 DATA 49,43,5A,45,49,4C,45,4E,0253
 1570 DATA 20,2D,20,45,58,50,41,4E,01E9
 1580 DATA 44,45,52,00,32,20,20,2D,017A
 1590 DATA 2D,2D,2D,2D,20,20,44,41,0179
 1600 DATA 54,41,5A,45,49,4C,45,4E,025C
 1610 DATA 20,20,2D,20,45,58,50,41,01BB
 1620 DATA 4E,44,45,52,00,44,41,53,0201
 1630 DATA 20,50,52,4F,47,52,41,4D,0238
 1640 DATA 4D,20,48,41,54,20,4A,45,01F9
 1650 DATA 54,5A,54,20,20,20,20,01A2
 1660 DATA 20,20,20,42,59,54,45,53,01E7
 1670 DATA 20,57,45,4E,49,47,45,52,0231
 1680 DATA 00,42,20,49,20,54,20,54,0193
 1690 DATA 20,45,20,20,20,20,57,20,015C
 1700 DATA 41,20,52,20,54,20,45,20,01AC
 1710 DATA 4E,20,20,21,21,21,00,CD,01BE
 1720 DATA BF,96,3E,22,ED,B1,CC,94,04B3
 1730 DATA 94,E0,16,F8,F5,C5,E5,2B,054E
 1740 DATA 2B,7E,FE,20,CC,E8,94,7E,048D
 1750 DATA FE,BF,28,46,FE,2C,28,42,03BF
 1760 DATA FE,EF,28,3E,FE,F4,28,3A,04A7
 1770 DATA FE,3B,28,36,FE,B6,28,32,03A5
 1780 DATA FE,B7,28,2E,FE,ED,28,2A,0448
 1790 DATA FE,A8,28,26,FE,AB,28,22,03B7
 1800 DATA FE,CA,28,1E,FE,CB,28,1A,0419
 1810 DATA FE,EE,28,16,FE,F1,28,12,0453
 1820 DATA FE,F2,28,0E,FE,F0,28,0A,0446
 1830 DATA FE,F3,28,06,E1,C1,F1,C9,057B
 1840 DATA 2B,C9,E1,C1,2B,23,0B,7E,036D
 1850 DATA FE,22,28,0C,B7,20,F6,5E,0406
 1860 DATA C5,CD,04,95,C1,E1,F1,C9,0587
 1870 DATA 23,0B,F1,C9,22,45,99,2A,0312
 1880 DATA 81,AE,23,4E,23,46,2B,09,023D
 1890 DATA E5,B7,ED,5B,45,99,ED,52,0501
 1900 DATA E1,38,F0,B7,ED,42,23,23,0435
 1910 DATA 4E,23,46,C5,D1,CD,00,B9,03D3
 1920 DATA CD,0D,E1,3E,01,32,47,99,030C
 1930 DATA C9,2A,81,AE,23,CD,CE,96,0476
 1940 DATA C8,E5,23,23,23,3E,8C,0303
 1950 DATA BE,E1,28,06,CD,06,97,EA,0421
 1960 DATA 4D,95,09,18,E8,E5,0B,71,034C
 1970 DATA 23,70,CD,4F,97,E1,18,DD,041C
 1980 DATA 2A,81,AE,23,CD,CE,96,C8,0475
 1990 DATA E5,C5,23,23,5E,23,56,ED,03B4
 2000 DATA 53,4D,99,23,7E,FE,C5,28,03C5
 2010 DATA 09,FE,01,28,05,C1,E1,09,02E0
 2020 DATA 18,E2,C1,E1,22,4B,99,09,03AB
 2030 DATA 22,49,99,23,23,5E,23,56,0221
 2040 DATA ED,53,51,99,ED,5B,4D,99,0458
 2050 DATA CD,D5,96,EA,9B,95,CD,4F,056E
 2060 DATA 97,18,ED,ED,5B,51,99,73,0411
 2070 DATA 23,72,18,B4,2A,81,AE,23,02DD
 2080 DATA 22,55,99,18,03,2A,55,99,0243
 2090 DATA CD,CE,96,C8,C5,CD,71,97,0593
 2100 DATA C1,30,06,09,22,55,99,18,0228
 2110 DATA BF,CD,7A,97,28,F5,ED,43,051A
 2120 DATA 4F,99,22,57,99,3E,A1,CD,03A6
 2130 DATA F8,96,EA,BE,95,3E,B2,CD,0585
 2140 DATA F8,96,EA,BE,95,09,CD,CE,056C
 2150 DATA 96,C8,CD,71,97,38,D4,ED,052C
 2160 DATA 43,53,99,CD,7A,97,28,CB,0400
 2170 DATA E5,C5,CD,36,97,C1,E1,30,0516
 2180 DATA C3,E5,C5,23,23,5E,23,56,038A
 2190 DATA ED,53,51,99,CD,D5,96,C1,0523
 2200 DATA E1,EA,BC,95,CD,84,97,2A,052E
 2210 DATA 57,99,ED,4B,4F,99,09,22,033B
 2220 DATA 4B,99,23,23,23,23,22,49,01DB
 2230 DATA 99,CD,4F,97,18,87,21,59,0365
 2240 DATA 99,23,AF,BE,20,08,21,FF,0371
 2250 DATA 00,22,59,99,18,06,21,60,01B3
 2260 DATA EA,22,59,99,2A,81,AE,23,037A
 2270 DATA 22,55,99,18,03,2A,55,99,0243
 2280 DATA CD,CE,96,C8,E5,C5,CD,7A,05EA
 2290 DATA 97,C1,E1,28,06,09,22,55,02E7
 2300 DATA 99,18,ED,ED,43,4F,99,22,03D8
 2310 DATA 57,99,09,CD,CE,96,C8,CD,04BF
 2320 DATA 7A,97,20,E9,ED,43,53,99,0436
 2330 DATA E5,C5,CD,36,97,C1,E1,30,0516
 2340 DATA DD,E5,C5,23,23,5E,23,56,03A4

[1569]
 [1338]
 [1672]
 [1148]
 [2129]
 [1356]
 [2305]
 [1590]
 [1414]
 [1571]
 [1448]
 [2403]
 [1569]
 [1338]
 [2026]
 [1966]
 [1093]
 [1446]
 [1390]
 [1664]
 [2067]
 [1780]
 [1839]
 [742]
 [1533]
 [1677]
 [1656]
 [2045]
 [1193]
 [1713]
 [1803]
 [1461]
 [1425]
 [2096]
 [1434]
 [1214]
 [855]
 [1967]
 [1361]
 [836]
 [1405]
 [1455]
 [1107]
 [692]
 [1135]
 [1074]
 [1132]
 [1643]
 [2437]
 [1826]
 [2397]
 [2042]
 [1616]
 [1990]
 [1954]
 [1660]
 [1833]
 [1673]
 [1960]
 [1724]
 [1701]
 [2001]
 [1704]
 [1767]
 [1348]
 [1011]
 [2153]
 [1383]
 [1755]
 [1878]
 [1498]
 [1130]
 [2785]
 [1849]
 [1409]
 [1398]
 [2066]
 [2327]
 [1740]
 [1911]
 [1656]
 [1729]
 [1386]
 [1731]
 [1011]
 [1121]
 [1453]
 [1338]
 [1802]
 [1575]
 [1409]
 [837]

2350 DATA ED,53,51,99,CD,D5,96,C1,0523
 2360 DATA E1,EA,56,96,CD,A8,96,2A,04BC
 2370 DATA 57,99,ED,4B,4F,99,09,22,033B
 2380 DATA 4B,99,23,23,23,23,23,01B6
 2390 DATA 22,49,99,CD,4F,97,18,9D,036C
 2400 DATA 2A,57,99,E5,ED,4B,4F,99,041F
 2410 DATA 09,2B,36,2C,E1,ED,5B,5B,031A
 2420 DATA 99,1B,1B,73,23,72,C9,ED,038D
 2430 DATA 4B,81,AE,2A,83,AE,C5,B7,0451
 2440 DATA ED,42,E5,C1,E1,C9,4E,23,04F0
 2450 DATA 46,2B,78,B1,C9,CD,BF,96,0485
 2460 DATA 3E,1E,ED,B1,CC,E2,96,E0,051E
 2470 DATA 18,F8,F5,C5,E5,7B,EE,20,0508
 2480 DATA 0B,23,7A,BE,20,06,E1,C1,032E
 2490 DATA F1,FD,E1,C9,E1,C1,F1,C9,06F4
 2500 DATA E5,C5,ED,B1,28,05,E2,03,045A
 2510 DATA 97,18,F7,C1,E1,C9,E5,C5,05BB
 2520 DATA 3E,20,ED,B1,CC,14,97,E2,0455
 2530 DATA 03,97,18,F6,F5,C5,E5,2B,0472
 2540 DATA 2B,7E,FE,80,38,12,23,ED,02BE
 2550 DATA 4B,99,23,22,49,99,DD,E1,03C9
 2560 DATA DD,E1,F1,DD,E1,C1,E1,C9,06D8
 2570 DATA FE,29,28,EA,18,BE,2A,4F,0388
 2580 DATA 99,ED,4B,53,99,09,2B,2B,031C
 2590 DATA 2B,2B,22,5B,99,ED,5B,59,030D
 2600 DATA 99,7C,92,C0,7D,93,C9,2A,046A
 2610 DATA 83,AE,ED,5B,49,99,B7,ED,04FF
 2620 DATA 52,E5,C1,2A,49,99,ED,5B,044C
 2630 DATA 4B,99,ED,B0,ED,53,83,AE,04F2
 2640 DATA 1B,06,0A,AF,12,13,10,FC,020B
 2650 DATA C9,E5,B7,2A,59,99,ED,42,04B0
 2660 DATA E1,C9,E5,23,23,23,23,3E,0359
 2670 DATA 8C,BE,E1,C9,2A,57,99,E5,04F3
 2680 DATA ED,4B,4F,99,09,2B,36,01,028B
 2690 DATA E1,ED,5B,5B,99,73,23,72,0425
 2700 DATA C9,21,4D,99,AF,77,23,77,0390
 2710 DATA 2A,83,AE,22,5D,99,2A,81,031E
 2720 DATA AE,23,CD,CE,96,C8,18,03,03E5
 2730 DATA 09,18,F7,22,57,99,ED,43,035A
 2740 DATA 4F,99,CD,97,98,3E,8C,BE,046C
 2750 DATA 28,05,3E,A1,BE,20,05,2A,0219
 2760 DATA 57,99,18,B4,11,05,00,7E,0280
 2770 DATA FE,02,20,07,CD,2C,98,23,02DB
 2780 DATA 13,18,F4,FE,03,28,F5,FE,043B
 2790 DATA 04,28,F1,FE,OD,28,ED,FE,043B
 2800 DATA 0B,28,E9,FE,OC,28,E5,FE,0431
 2810 DATA 19,20,04,23,13,18,0E,FE,0269
 2820 DATA FF,28,F8,FE,1A,20,04,23,037E
 2830 DATA 13,18,F0,FE,1B,28,F8,FE,0452
 2840 DATA 1C,28,F4,FE,1E,28,F0,FE,046A
 2850 DATA 1F,20,08,23,23,23,13,13,00D6
 2860 DATA 13,18,E4,FE,01,28,1A,FE,034E
 2870 DATA 00,28,04,23,13,18,AB,2A,014C
 2880 DATA 57,99,18,84,23,23,13,13,01F8
 2890 DATA F5,23,13,7E,17,30,FA,F1,03DB
 2900 DATA C9,23,7E,FE,97,28,E5,2B,0437
 2910 DATA AF,77,23,22,45,99,2A,57,02CA
 2920 DATA 99,73,23,72,2A,4F,99,B7,036A
 2930 DATA ED,52,CD,97,98,22,53,99,0449
 2940 DATA 2A,5D,99,ED,5B,45,99,B7,03FD
 2950 DATA ED,52,CD,97,98,22,55,99,044B
 2960 DATA 2A,5D,99,E5,CD,97,98,EB,04EC
 2970 DATA E1,ED,4B,55,99,ED,B8,2A,04D6
 2980 DATA 5D,99,CD,97,98,22,83,AE,0445
 2990 DATA 2A,45,99,ED,4B,53,99,71,039D
 3000 DATA 23,70,23,ED,4B,4D,99,03,02D7
 3010 DATA 71,23,70,C3,A0,97,23,23,0344
 3020 DATA 23,23,23,C9,3E,DC,32,59,02D7
 3030 DATA 99,2A,83,AE,22,5D,99,2A,0336
 3040 DATA 81,AE,23,CD,CE,96,C8,E5,0530
 3050 DATA CD,97,98,3E,8C,BE,E1,28,048D
 3060 DATA 03,09,18,BF,CD,71,97,30,0318
 3070 DATA F8,22,57,99,ED,43,4F,99,0422
 3080 DATA 11,B4,00,19,23,13,7E,FE,0290
 3090 DATA 2C,28,0E,FE,00,28,02,18,019F
 3100 DATA F3,2A,57,99,18,DB,AF,77,0426
 3110 DATA 23,22,45,99,2A,57,99,13,0250
 3120 DATA 73,23,72,2A,4F,99,B7,ED,03BE
 3130 DATA 52,23,23,CD,97,98,22,53,0309
 3140 DATA 99,2A,5D,99,ED,5B,45,99,03DF
 3150 DATA B7,ED,52,23,23,CD,97,98,0438
 3160 DATA 22,55,99,2A,5D,99,E5,23,0338
 3170 DATA 23,CD,97,98,EB,E1,ED,4B,0523
 3180 DATA 55,99,ED,B8,2A,5D,99,23,03D6
 3190 DATA 23,CD,97,98,22,83,AE,2A,039C
 3200 DATA 45,99,ED,4B,53,99,71,23,0396
 3210 DATA 70,23,ED,4B,4D,99,03,71,0325
 3220 DATA 23,70,23,3E,8C,77,23,3E,0258
 3230 DATA 20,77,C3,A1,98,00,00,00,0293
 3240 DATA 00,00,00,00,00,00,00,00,0000
 3250 DATA 00,00,00,00,00,00,00,00,0000
 3260 DATA 00,00,00,00,00,00,00,00,0000

[2066]
 [1462]
 [1740]
 [2004]
 [1460]
 [1450]
 [921]
 [1810]
 [1861]
 [1469]
 [1185]
 [1198]
 [1671]
 [2557]
 [2174]
 [1373]
 [1522]
 [1381]
 [1438]
 [1643]
 [2159]
 [2053]
 [1608]
 [1361]
 [1760]
 [1614]
 [1758]
 [1418]
 [2326]
 [2016]
 [2066]
 [1451]
 [1585]
 [842]
 [2300]
 [2075]
 [1348]
 [1734]
 [1347]
 [1977]
 [1463]
 [1526]
 [1671]
 [1560]
 [1123]
 [1560]
 [1123]
 [1543]
 [1333]
 [1593]
 [1942]
 [1850]
 [1180]
 [1434]
 [866]
 [1558]
 [897]
 [1030]
 [1036]
 [2306]
 [1044]
 [1756]
 [1231]
 [1411]
 [1915]
 [1311]
 [1922]
 [1067]
 [820]
 [1375]
 [1750]
 [1956]
 [1583]
 [1442]
 [1616]
 [1879]
 [1114]
 [1654]
 [1831]
 [833]
 [767]
 [2048]
 [736]
 [1453]
 [899]
 [951]
 [957]
 [1025]
 [1037]
 [1003]
 [1003]
 [1003]

KNOW



für CPC 464 · 664 · 6128

Kassettensversion (nur 464) 24,- DM
Diskette 3" 29,- DM

- kann eine **unbegrenzte** Anzahl von Fragen und Antworten verwalten!
- hat ständig ca. **400 Fragen plus** Antworten im Speicher
- bietet die Möglichkeit, **eigene** Fragen einzugeben, und zwar mit einem **komfortablen** Editor!
- besitzt eine **Supergrafik** mit Window-Technik!
- ist vollkommen **menuegesteuert!**
- zeigt bis jetzt noch nie dagewesene Tricks mit dem **Videocontroller!**
- stellt zu jeder Frage **5 mögliche** Antworten vor!
- kann man mit der **ganzen Familie** spielen, da jeder eine Mindest-Chance von 1-5 hat!
- muß man mit **Strategie** spielen, da der Beste in jedem Wissensgebiet am Ende noch Zusatzpunkte bekommt!
- wurde mit **äußerst** schnellen **Suchroutinen** ausgestattet, die verhindern, daß eine schon richtig beantwortete Frage nochmals erscheint!
- ist spielbar mit **1-4** Einzelspielern oder in Gruppen mit einem Vielfachen davon!
- erkennt **automatisch**, welche Fragenblöcke auf der Diskette noch unbeantwortet sind!
- zeichnet sich nicht nur durch die og. Punkte als höchst **zukunftsicher** aus!



In den Tiefen der Galaxis wartet die letzte Herausforderung auf Sie

Der Zeltherr, ein Wesen aus den Tiefen des Weltraums, hat einen kosmischen Wettbewerb ausgeschrieben um etwas Zerstreung in sein unendlich langes Leben zu bringen: Den **STARTEST**.
 Der Preis sind zusätzliche Lebensjahre. Jede Intelligenz, ob Spinnwesen von Astus, ob Mensch von der Erde oder Schläferide Steine von Murka, können teilnehmen. Doch vor dem ersahnten Preis 6 Prüfungen, die sowohl die Intelligenz als auch die Geschicklichkeit testen.

Nur wenige können sie alle bestehen, aber wer es schafft, dem winkt eine Verlängerung seiner Lebensspanne.

für CPC 464 · 664 · 6128

Kassette 24,- DM
Diskette 3" 29,- DM

CPC Power-Spiele-Paket!

18 tolle Spiele für Ihren CPC 464, 664 oder 6128 auf 4 Kassetten bzw. 3"-Disketten zum Knüllerpreis von nur 50,- DM (4 Kassetten) 70,- DM (4 Disketten)

nur solange Vorrat reicht



Platinenservice

Für Ihren CPC

Die CPC-Schneiderware ist ein universelles Peripheriesystem für die Schneider CPC's auf der Basis des bekannten ECB-Bussystems. Um die Schneiderware an Ihren CPC anzuschließen, benötigen Sie:

1. Das Verbindungskabel vom Expansionsport des Rechners zur Basisplatine (Rechnertyp beachten, da Anschlüsse bei 464/664 verschieden von 6128 sind).
2. Die Basisplatine, welche die Pinbelegung der CPC-Ports auf die des ECB-Systems umsetzt. Die Karte enthält fünf Steckplätze zur Aufnahme und gleichzeitigen Ansteuerung der Schneiderware-Erweiterungskarten.

Wollen Sie nur eine Karte betreiben, so können Sie diese über ein selbstgefertigtes Kabel an den CPC anschließen. Die Anschlußbelegung dieses Kabels sehen Sie in Heft 7/86, S. 61.

Das verwendete Platinenmaterial ist glasfaserverstärktes Epoxydharz; die beidseitig beschichteten Platinen sind chemisch durchkontaktiert. Für die Fertigbausteine kommen Bauteile erster Wahl zum Einsatz.

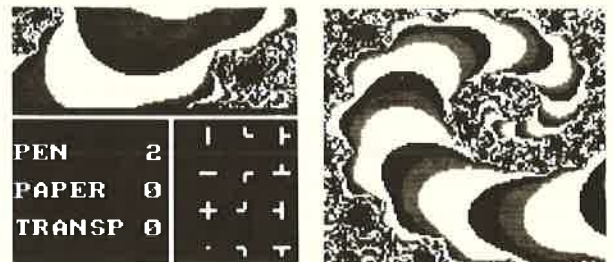
Die Preise

Basisplatine, unbestückt	24,90 DM
dto., bestückt	62,90 DM
Kabel 464/664	35,90 DM
Kabel 6128	45,90 DM
Centronics, unbestückt	17,90 DM
dto., bestückt	79,90 DM
V/24, bestückt	139,90 DM
Netzteil, unbestückt	17,90 DM
dto., bestückt	119,90 DM
Trafo	79,90 DM
Karte und Trafo	184,90 DM
Hardware-Uhr, unbest.	29,80 DM
Hardware-Uhr, bestck.	99,90 DM
PIO-Karte, Karte bestck.	198,90 DM
MIDI-Interf., Plat. unbest.	39,90 DM
MIDI-Interf., kompl. best.	198,00 DM
A/D-D/A Wandl., funktionsf.	169,90 DM
EPR0M/RAM-Karte	
funktionsfertig	229,90 DM
Eprommer-Karten	
(2 Karten, bestückt)	198,90 DM

Copyshop

Das universelle Hardcopy-Programm für Schneider CPC 464/664/6128

Autor: Matthias Uphoff



* COPYSHOP im Detail:

- Hardcopy in 4 (!) Formaten: DIN A4, DIN A5, 13,5 x 8,5 cm und 21,5 x 13,5 cm
- superschnelle Hardcopy-Routine: DIN A4 in ca. 4 Minuten (DMP 4000)
- arbeitet in allen 3 Modos
- Anpaßmenue für JEDEN Epson-kompatiblen Drucker
- läuft ebenfalls mit den Seikosha-Druckern GP-500 CPC, GP-550 CPC und GP-1000 CPC
- Anpassung an Drucker möglich, die mit 1280 Punkten pro Zeile arbeiten, z. B. CPA-80 GS
- Okimate ML 182 - Anpassung kann beim Verlag angefordert werden
- Anpassung auch für Drucker, die die Bitbild-Bytes verkehrt herum drucken (das MSB unten statt oben), z. B. NEC P2-Pinwriter.
- 32 Farbraster über Menue wählbar
- Grafikeditor
- komfortable Pull-Down-Menues
- schnelle Fill-Routine
- beliebige Ausschnittvergrößerungen
- Bildschirm invertieren
- selbstrelozierbare Hardcopy-Routinen für eigene Programme
- neue Save- und Load-Routinen erkennen automatisch Mode und Farbwerte
- Freezer - saved auf Tastendruck Screenshots aus laufenden BASIC-Programmen, die anschließend ausgedruckt werden können

* Das auf dem Datenträger mitgelieferte Programm »Screen Save«, welches beliebige Screens auf Disk abspeichert, arbeitet nur nach Entfernen des Vortex-Controllers

Und die Weltneuheit: Hardcopy-Simulator auf dem Bildschirm !!

Sie können sich Ihre Hardcopy vor dem endgültigen Ausdruck auf dem Bildschirm ansehen!

COPYSHOP ist das ultimative Hardcopy-Programm für alle Schneider Computer. Erhältlich auf Kassette (DM 59,-) und 3"-Diskette (DM 69,-) inkl. ausführlicher Bedienungsanleitung. (unverbindliche Preisempfehlung)

Bitte Bestell-Coupon auf Seite 95 benutzen.
DMV Daten- und Medien-Verlagsgesellschaft mbH
 Postfach 250 · Fuldaer Str. 6
 3440 Eschwege · Tel. (0 56 51) 87 02

Schulnoten-Verwaltung

Schulnoten können Ärger bereiten, nicht nur den Schülern, die diese kleinen Ziffern hinter den Lehrfächern auf den Zeugnissen fürchten wie die Windpocken, sondern auch (man sollte es nicht meinen) den Lehrern, die am Ende eines Schuljahres vor einem Wust von Zensuren stehen und diese in eine, dem Schüler gerecht werdende Zeugnisnote umwandeln müssen. Was bietet sich in so einem Falle mehr an, als ein Computer, der für eine solche Problematik geradezu prädestiniert ist?

Das Programm Notenverwaltung unterteilt die Noten in zwei einzelne Gruppen:

- 1) Noten, die sich aus mündlichen Abfragen, unangekündigten Arbeiten (Extemporalen), etc. bilden (diese werden im Programm und in der Beschreibung immer mit der Abkürzung EX bezeichnet) und
- 2) Noten, die sich aus vorher angekündigten Arbeiten, Probenarbeiten, etc. zusammensetzen (im Programm und der Beschreibung immer mit der Abkürzung SchA bezeichnet).

Die Anzahl und die Namen der Fächer können einfach auf die jeweiligen Gegebenheiten zugeschnitten werden:

- 1) In Zeile 180 können Sie in der Variablen "fachanz" die Anzahl der Fächer ändern, die Sie verwalten wollen.
- 2) In den Datazeilen ab Zeile 4990 ist es möglich, die Namen der Fächer zu ändern.

Die Durchschnittsnote (DS) eines Faches wird aus der DS-Note der EX-Noten und der DS-Note der SchA-Noten gebildet. In einigen Fächern zählt die DS-Note der SchA-Noten doppelt, in anderen nur einfach, in manchen Fächern werden keine Arbeiten der Art SchA geschrieben. Daher muß in den Datazeilen hinter jedem Fach, durch Komma getrennt, eine Zahl eingegeben werden. Dabei gilt folgendes:

- 0 = In diesem Fach gibt es keine SchA-Noten.
- 1 = In diesem Fach wird die DS-Note der SchA-Noten einfach gezählt.
- 2 = In diesem Fach wird die DS-Note der SchA-Noten doppelt gezählt.

Nach dem Starten des Programmes erscheint das Hauptmenue mit folgenden Punkten:

- 1) Noten eingeben:
 - a) Art der Arbeit (1 = Note der Gruppe SchA; 2 = Note der Gruppe EX)
 - b) Fach (Eingabe der Kennziffer des Faches aus der linken Bildschirmhälfte)

- c) Datum (Tag (ENTER), Monat (ENTER), Jahr (ENTER))
 - d) Note
- 2) Noten ausgeben:
 - a) Durchschnittsnoten: Berechnung der DS-Noten aller Fächer und übersichtliche Ausgabe. Die Berechnung erfolgt mit folgender Formel: DS-Note = (Summe der SchA-Noten geteilt durch die Anzahl der SchA-Noten mal 1 (bzw. 2) plus Summe der EX-Noten geteilt durch die Anzahl der EX-Noten) geteilt durch 2 (bzw. 3)
 - b) Nach Datum: Ausgabe der Noten mit einem bestimmten Datum. Es werden auch unvollständige Daten angenommen. Einige Beispiele:
.05 (Ausgabe aller Noten vom Monat Mai).
.86 (Ausgabe aller Noten aus dem Jahr 1986)
. . (Ausgabe aller Noten)
 - c) Nach Fächern: Ausgabe aller Noten eines bestimmten Faches.
 - d) Grafik: Ein Kurvendiagramm der Noten eines Faches wird ausgegeben. Mit ENTER kommen sie zurück ins Menue.
 - 3) laden von Noten vom Datenträger.
 - 4) abspeichern von Noten auf einen Datenträger.
 - 5) Noten ändern: Hier können Falscheingaben rückgängig gemacht werden.

Die Variablen des Programmes

Hier nun die Variablenliste zum Noten-Verwaltungsprogramm:

art	=	Art der Note (EX oder SchA)
ex	=	Summe der EX-Noten
exanz	=	Anzahl der EX-Noten
fach	=	Fach, für das die Note gezählt wird
fach\$	=	Namen der Fächer
fachanz	=	Anzahl der Fächer
gnote	=	Durchschnittsnote
j	=	Jahr
m	=	Monat
t	=	Tag
n	=	Note
note	=	Anzahl aller Noten
sch	=	SchA - DS-Note doppelt oder einfach?
schul	=	Summe der SchA-Noten
schulanz	=	Anzahl der SchA-Noten

Um die Noten zu verbessern, ist dieses Programm wohl nicht zu gebrauchen, da hilft wohl doch nur lernen. Aber um einen guten Überblick über die eigenen Zensuren oder die seiner Schüler zu erhalten, kann dieses Programm eine gute Hilfe sein.
(Th. Schatz/JB)

```

für 464-664-6128
10 '*****
20 '* *
30 '* Notenverwaltung *
40 '* *
60 '* *
70 '* (c) by T.Schatz *
80 '* *
90 '*****
100 '
110 ' *** Initialisierung ***
120 '
130 RESTORE 4950
Listing: Schulnoten-Verwaltung
    
```

```

140 DIM mo$(12)
150 FOR i=1 TO 12
160 READ mo$(i)
170 NEXT i
180 fachanz=15
190 DIM fach$(fachanz),sch(fachanz)
200 DIM art(500),fach(500),n(500),m(500),t
(500),j(500)
210 DIM ex(fachanz),schul(fachanz),schulanz
z(fachanz),exanz(fachanz),gnote(fachanz)
220 DIM merke2(fachanz),merke3(fachanz),me
rke4(fachanz),merke5(fachanz)
230 RESTORE 4960
240 FOR i=1 TO fachanz
250 READ fach$(i),sch(i)
260 NEXT i
Listing: Schulnoten-Verwaltung
    
```

```

270 MODE 1 [506]
280 ' [117]
290 '*** Hauptmenue *** [954]
300 ' [117]
310 CLS [91]
320 PRINT [361]
330 PRINT [361]
340 PRINT TAB(14); CHR$(24);" Hauptmenue " [2922]
;CHR$(24)
350 PRINT:PRINT [1082]
360 PRINT TAB(9);"<1> Noten eingeben" [1474]
370 PRINT [361]
380 PRINT TAB(9);"<2> Noten ausgeben" [2193]
390 PRINT [361]
400 PRINT TAB(9);"<3> Noten speichern" [1860]
410 PRINT [361]
420 PRINT TAB(9);"<4> Noten laden" [1391]
430 PRINT [361]
440 PRINT TAB(9);"<5> Noten aendern" [2236]
450 PRINT [361]
460 PRINT TAB(9);"<E> Ende" [1235]
470 PRINT:PRINT [1082]
480 PRINT TAB(14);CHR$(24);" Ihre Wahl! "; [2635]
CHR$(24)
490 a$=LOWERS$(INKEY$) [960]
500 IF a$="" THEN 490 [769]
510 IF a$="e" THEN CLS:END [1437]
520 a=VAL(a$) [956]
530 IF a<1 OR a>5 THEN 490 [883]
540 ON a GOTO 550,1520,1250,990,3130 [1525]
550 ' [117]
560 '*** Noten eingeben *** [1454]
570 ' [117]
580 WINDOW #1,21,40,2,25 [1324]
590 CLS [91]
600 PRINT TAB(11);CHR$(24);" Noten eingebe [2496]
n ";CHR$(24)
610 FOR i=1 TO fachanz [1308]
620 LOCATE 1,2+1 [610]
630 PRINT USING"###";i [708]
640 LOCATE 3,2+1 [642]
650 PRINT ". ";fach$(1) [969]
660 NEXT 1 [375]
670 CLS #1 [373]
680 note=note+1 [566]
690 PRINT #1 [318]
700 PRINT #1,"<1> Scha" [1399]
710 PRINT #1,"<2> Ex" [1526]
720 PRINT #1 [318]
730 INPUT #1,"Art: ",art(note) [2213]
740 PRINT #1 [318]
750 INPUT #1,"Fach: ",fach(note) [1755]
760 IF art(note)=2 THEN exanz(fach(note))= [3016]
exanz(fach(note))+1
770 IF art(note)=1 THEN schulanz(fach(note) [2823]
)=schulanz(fach(note))+1
780 PRINT #1 [318]
790 PRINT #1,"Datum:" [704]
800 LOCATE 21,11 [596]
810 PRINT" . ." [780]
820 LOCATE 21,11 [596]
830 INPUT "",t(note) [698]
840 LOCATE 24,11 [566]
850 INPUT "",m(note) [512]
860 LOCATE 27,11 [739]
870 INPUT "",j(note) [574]
880 LOCATE 21,13 [602]
890 INPUT "Note: ",n(note) [1777]
900 IF art(note)=2 THEN ex(fach(note))=ex( [3201]
fach(note))+n(note)
910 IF art(note)=1 THEN schul(fach(note))= [4084]
schul(fach(note))+n(note)
920 LOCATE 21,15 [608]
930 PRINT "Noch eine Eingabe" [2233]
940 LOCATE 21,16 [611]
950 INPUT "(j/n)? ",jn$ [1056]
960 IF jn$="j" OR jn$="J" THEN 670 [1859]
970 IF jn$="N" OR jn$="n" THEN 290 [1099]
980 GOTO 920 [332]
990 ' [117]
1000 '*** laden *** [1116]
1010 ' [117]
1020 CLS [91]
1030 PRINT TAB(14);CHR$(24);" Noten laden [1990]
";CHR$(24)
1040 PRINT [361]
1050 PRINT"Halten Sie bitte die Datei bere [3786]
it!"

```

Listing: Schulnoten-Verwaltung

```

1060 PRINT [361]
1070 OPENIN"note.dat" [2210]
1080 INPUT #9,note [1327]
1090 FOR i=1 TO note [415]
1100 INPUT #9,art(i) [1337]
1110 INPUT #9,fach(i) [931]
1120 INPUT #9,n(i) [793]
1130 INPUT #9,m(i) [578]
1140 INPUT #9,t(i) [841]
1150 INPUT #9,j(i) [1017]
1160 NEXT i [375]
1170 FOR i=1 TO fachanz [1308]
1180 INPUT #9,exanz(i) [1068]
1190 INPUT #9,schulanz(i) [916]
1200 INPUT #9,schul(i) [1048]
1210 INPUT #9,ex(i) [965]
1220 NEXT i [375]
1230 CLOSEIN [752]
1240 GOTO 290 [411]
1250 ' [117]
1260 '*** speichern *** [1322]
1270 ' [117]
1280 SPEED WRITE 1 [1302]
1290 CLS [91]
1300 PRINT TAB(12);CHR$(24);" Noten speich [2722]
ern ";CHR$(24)
1310 PRINT [361]
1320 PRINT"Halten Sie bitte einen Datentra [4677]
eger bereit!"
1330 PRINT [361]
1340 OPENOUT"note.dat" [1648]
1350 PRINT #9,note [1180]
1360 FOR i=1 TO note [415]
1370 PRINT #9,art(i) [1175]
1380 PRINT #9,fach(i) [1202]
1390 PRINT #9,n(i) [1181]
1400 PRINT #9,m(i) [965]
1410 PRINT #9,t(i) [973]
1420 PRINT #9,j(i) [1155]
1430 NEXT i [375]
1440 FOR i=1 TO fachanz [1308]
1450 PRINT #9,exanz(i) [1203]
1460 PRINT #9,schulanz(i) [2340]
1470 PRINT #9,schul(i) [1561]
1480 PRINT #9,ex(i) [1220]
1490 NEXT i [375]
1500 CLOSEOUT [902]
1510 GOTO 290 [411]
1520 ' [117]
1530 '*** Noten ausgeben *** [1477]
1540 ' [117]
1550 ' Untermenue [640]
1560 ' [117]
1570 CLS [91]
1580 PRINT [361]
1590 PRINT [361]
1600 PRINT TAB(14); CHR$(24);" Untermenue [2605]
";CHR$(24)
1610 PRINT TAB(14);CHR$(24);" ausgeben " [2745]
;CHR$(24)
1620 PRINT:PRINT:PRINT [1082]
1630 PRINT TAB(9);"<1> Durchschnittsnote [2492]
n"
1640 PRINT [361]
1650 PRINT TAB(9);"<2> nach Datum" [1863]
1660 PRINT [361]
1670 PRINT TAB(9);"<3> nach Faechern" [3014]
1680 PRINT [361]
1690 PRINT TAB(9);"<4> Grafik" [1414]
1700 PRINT [361]
1710 PRINT TAB(9);"<H> Hauptmenue" [2129]
1720 PRINT:PRINT:PRINT [1082]
1730 PRINT TAB(14);CHR$(24);" Ihre Wahl! " [2635]
;CHR$(24)
1740 a$=LOWERS$(INKEY$) [960]
1750 IF a$="" THEN 1740 [699]
1760 IF a$="h" THEN 290 [1790]
1770 a=VAL(a$) [956]
1780 IF a<1 OR a>4 THEN 1740 [1638]
1790 ON a GOTO 1800,1940,2460,2680 [1383]
1800 ' [117]
1810 ' Durchschnittsnoten [1751]
1820 ' [117]
1830 CLS [91]
1840 PRINT [361]
1850 PRINT TAB(10);CHR$(24);" Durchschnitt [3204]
snoten ";CHR$(24)
1860 PRINT:PRINT [743]

```

Listing: Schulnoten-Verwaltung

Anwendungen

```

1870 GOSUB 4590 [1011]
1880 FOR i=1 TO fachanz [1308]
1890 PRINT fach$(i);TAB(15);". . . . ." [2018]
. . .
1900 PRINT USING"#.##";gnote(i) [1283]
1910 NEXT [350]
1920 GOSUB 4740 [1013]
1930 GOTO 1570 [385]
1940 ' [117]
1950 ' nach Datum [534]
1960 ' [117]
1970 CLS [91]
1980 PRINT TAB(10);CHR$(24);" Ausgabe nach [4075]
Datum ";CHR$(24)
1990 PRINT:PRINT [743]
2000 PRINT"Datum:" [1107]
2010 LOCATE 1,6 [626]
2020 PRINT" . . ." [780]
2030 LOCATE 1,6 [626]
2040 INPUT "",ta [421]
2050 LOCATE 4,6 [638]
2060 INPUT "",monat [1070]
2070 LOCATE 7,6 [331]
2080 INPUT "",jahr [453]
2090 CLS [91]
2100 PRINT TAB(10);CHR$(24);" Ausgabe nach [4723]
Datum ";CHR$(24):PRINT:PRINT
2110 FOR i=1 TO note [415]
2120 IF ta=0 AND monat=0 AND jahr=0 THEN G [1204]
OSUB 2290:GOTO 2270
2130 IF ta=0 AND jahr=0 THEN 2200 [1084]
2140 IF ta=0 AND monat=0 THEN 2210 [441]
2150 IF monat=0 AND jahr=0 THEN 2220 [2385]
2160 IF ta=0 THEN 2230 [319]
2170 IF monat=0 THEN 2240 [1548]
2180 IF jahr=0 THEN 2250 [1206]
2190 GOTO 2260 [359]
2200 IF m(i)=monat THEN GOSUB 2290:GOTO 22 [3331]
70 ELSE GOTO 2270
2210 IF j(i)=jahr THEN GOSUB 2290:GOTO 227 [3107]
0 ELSE GOTO 2270
2220 IF t(i)=ta THEN GOSUB 2290:GOTO 2270 [2027]
ELSE GOTO 2270
2230 IF m(i)=monat AND j(i)=jahr THEN GOSU [4994]
B 2290:GOTO 2270 ELSE GOTO 2270
2240 IF j(i)=jahr AND t(i)=ta THEN GOSUB 2 [4575]
90:GOTO 2270 ELSE GOTO 2270
2250 IF m(i)=monat AND t(i)=ta THEN GOSUB [3596]
2290:GOTO 2270 ELSE GOTO 2270
2260 IF m(i)=monat AND t(i)=ta AND j(i)=ja [3774]
hr THEN GOSUB 2290:GOTO 2270 ELSE GOTO 227
0
2270 NEXT [350]
2280 GOTO 2450 [313]
2290 IF art(i)=2 THEN art$="Ex" ELSE art$= [3382]
"SchA"
2300 z=z+1 [702]
2310 v=v+1 [781]
2320 LOCATE 1,v+3 [1167]
2330 PRINT" . . ." [780]
2340 LOCATE 1,v+3 [1167]
2350 PRINT USING"###"; t(i) [919]
2360 LOCATE 4,v+3 [893]
2370 PRINT USING"###";m(i) [560]
2380 LOCATE 7,v+3 [611]
2390 PRINT USING"###";j(i) [962]
2400 LOCATE 12,v+3:PRINT fach$(fach(i)); [1310]
2410 PRINT TAB(30);art$; [1154]
2420 PRINT TAB(37);n(i) [868]
2430 IF z>19 THEN z=0:v=0:GOSUB 4740:CLS:P [6529]
RINT TAB(10);CHR$(24);" Ausgabe nach Datum
";CHR$(24):PRINT:PRINT
2440 RETURN [555]
2450 v=0:z=0:GOSUB 4740:GOTO 1570 [1578]
2460 ' [117]
2470 ' nach Faechern [709]
2480 ' [117]
2490 CLS [91]
2500 PRINT TAB(9);CHR$(24);" Ausgabe nach [3622]
Faechern ";CHR$(24)
2510 PRINT:PRINT [743]
2520 FOR i=1 TO fachanz [1308]
2530 PRINT USING"###";i;:PRINT". ";fach$(i) [2170]
2540 NEXT [350]
2550 LOCATE 25,5 [747]
2560 PRINT"Welches Fach?"; [2441]
2570 INPUT "",fa [450]
2580 CLS [91]

```

Listing: Schulnoten-Verwaltung

```

2590 PRINT TAB(9);CHR$(24);" Ausgabe nach [3622]
Faechern ";CHR$(24)
2600 PRINT [361]
2610 PRINT [361]
2620 FOR i=1 TO note [415]
2630 IF fach(i)=fa THEN GOSUB 2290 [1794]
2640 NEXT [350]
2650 v=0:z=0 [1014]
2660 GOSUB 4740 [1013]
2670 GOTO 1570 [385]
2680 ' [117]
2690 ' Grafik [540]
2700 ' [117]
2710 CLS [91]
2720 PRINT TAB(24);CHR$(24);" Grafik ";CHR [2138]
$(24)
2730 f=0 [392]
2740 FOR i=1 TO fachanz [1308]
2750 LOCATE 1,i+2 [996]
2760 PRINT USING"###";i;:PRINT". "; [2249]
2770 PRINT LEFT$(fach$(i),10) [1710]
2780 NEXT [350]
2790 PLOT 240,80,1 [604]
2800 DRAW 600,80,1 [180]
2810 PLOT 240,80,1 [604]
2820 DRAW 240,320,1 [510]
2830 FOR i=240 TO 600 STEP 30 [1915]
2840 PLOT 1,75,1:DRAW 1,85,1 [1164]
2850 NEXT [350]
2860 FOR i=80 TO 320 STEP 48 [1764]
2870 PLOT 235,1,1:DRAW 245,1,1 [1951]
2880 NEXT [350]
2890 FOR i=1 TO 6 [448]
2900 LOCATE 13,2+i*3 [935]
2910 PRINT i [393]
2920 NEXT i [375]
2930 TAG:FOR i=1 TO 12 [485]
2940 MOVE 220+i*30,72:PRINT mo$(i);:NEXT [1329]
2950 TAGOFF [1066]
2960 LOCATE 17,25 [658]
2970 INPUT"Fach : ",fa [1259]
2980 IF fa=0 THEN GOSUB 4740:GOTO 1570 [2166]
2990 f=f+1:IF f>3 THEN f=1 [2117]
3000 anfang=1 [518]
3010 FOR i=1 TO note [415]
3020 IF fach(i)=fa THEN PEN f:LOCATE 4,fa+ [4347]
2:PRINT fach$(fa):PEN 1:GOTO 3040
3030 GOTO 3110 [459]
3040 tage=(m(i)*30)+t(i) [1088]
3050 IF anfang=1 THEN anfang=0:GOSUB 4820: [3090]
xalt=210+tage:yalt=ypos1
3060 x=210+tage:GOSUB 4820:y=ypos1 [2458]
3070 IF m(i)<merke Monat THEN DRAW 604,yalt [3260]
,f:PLOT 242,yalt,f:DRAW x,y,f:GOTO 3100
3080 PLOT xalt,yalt,f [1605]
3090 DRAW x,y,f [201]
3100 xalt=x:yalt=y:merke Monat=m(i) [2050]
3110 NEXT [350]
3120 GOTO 2960 [395]
3130 ' [117]
3140 '*** Noten aendern *** [1594]
3150 ' [117]
3160 CLS [91]
3170 WINDOW #2,1,20,2,24 [1467]
3180 WINDOW #3,21,40,2,24 [1373]
3190 LOCATE 13,1 [526]
3200 PRINT CHR$(24);" Noten aendern ";CHR$ [2990]
(24)
3210 menue1$=CHR$(24)+CHR$(242)+" "+CHR$(2 [6200]
43)+"=blaettern [ENTER]=aendern H=Hauptm."
+CHR$(24)
3220 menue2$=CHR$(24)+CHR$(240)+" "+CHR$(2 [7019]
41)+"=waehlen [ENTER]=aendern Z=Zurueck"
+CHR$(24)
3230 LOCATE 1,25 [731]
3240 PRINT menue1$ [532]
3250 GOSUB 3320 [1168]
3260 a$=INKEY$ [278]
3270 IF a$="h" OR a$="H" THEN 290 [1557]
3280 IF a$=CHR$(242) THEN d=d-1:GOSUB 3320 [2080]
:GOTO 3260
3290 IF a$=CHR$(243) THEN d=d+1:GOSUB 3320 [1522]
:GOTO 3260
3300 IF a$=CHR$(13) THEN GOTO 3540 [1389]
3310 GOTO 3260 [461]
3320 ' [117]
3330 CLS #2:CLS #3 [637]

```

Listing: Schulnoten-Verwaltung

```

3340 IF d<1 THEN d=1 [332]
3350 IF d>note THEN d=note [871]
3360 LOCATE 3,3 [641]
3370 IF art(d)=1 THEN art$="SchA" [1479]
3380 IF art(d)=2 THEN art$="Ex" [1149]
3390 PRINT "Art: ";art$ [953]
3400 LOCATE 3,5 [631]
3410 PRINT "Fach: ";LEFT$(fach$(fach(d)),1 [2471]
1)
3420 LOCATE 3,7 [637]
3430 PRINT "Datum:" [1107]
3440 LOCATE 3,9:PRINT" . ." [1898]
3450 LOCATE 3,9 [659]
3460 PRINT USING"###";t(d) [925]
3470 LOCATE 6,9 [328]
3480 PRINT USING"###";m(d) [598]
3490 LOCATE 9,9 [316]
3500 PRINT USING"###";j(d) [1008]
3510 LOCATE 3,11 [709]
3520 PRINT"Note: ";n(d) [1940]
3530 RETURN [555]
3540 ' [117]
3550 CLS #3 [371]
3560 LOCATE 1,25 [731]
3570 PRINT menue2$ [538]
3580 lo=3:loalt=3 [805]
3590 LOCATE 1,lo [1029]
3600 PRINT CHR$(243) [1024]
3610 a$=INKEY$ [278]
3620 IF a$="z" OR a$="Z" THEN GOTO 3230 [1640]
3630 IF a$=CHR$(241) THEN lo=lo+2:GOSUB 36 [2576]
70:GOTO 3610
3640 IF a$=CHR$(240) THEN lo=lo-2:GOSUB 36 [1699]
70:GOTO 3610
3650 IF a$=CHR$(13) THEN 3740 [728]
3660 GOTO 3610 [686]
3670 IF lo>11 THEN lo=3 [1424]
3680 IF lo<3 THEN lo=11 [1305]
3690 LOCATE 1,loalt:PRINT" " [1370]
3700 LOCATE 1,lo [1029]
3710 PRINT CHR$(243) [1024]
3720 loalt=lo [740]
3730 RETURN [555]
3740 IF lo=3 THEN nehme=1 [943]
3750 IF lo=5 THEN nehme=2 [1120]
3760 IF lo=7 OR lo=9 THEN nehme=3 [1304]
3770 IF lo=11 THEN nehme=4 [549]
3780 GOSUB 4410 [847]
3790 ON nehme GOTO 3800,3950,4090,4300 [1865]
3800 ' [117]
3810 CLS #3 [371]
3820 PRINT #3 [316]
3830 PRINT #3,"<1> SchA" [1430]
3840 PRINT #3,"<2> Ex" [868]
3850 PRINT #3 [316]
3860 PRINT #3,"Art: "; [653]
3870 INPUT #3,"",art(d) [840]
3880 LOCATE 8,3:PRINT" ":LOCATE 8,3 [1558]
3890 IF art(d)=1 THEN art$="SchA" [1479]
3900 IF art(d)=2 THEN art$="Ex" [1149]
3910 PRINT art$ [442]
3920 CLS #3 [371]
3930 GOSUB 4500 [837]
3940 GOTO 3610 [686]
3950 ' [117]
3960 CLS #3 [371]
3970 FOR i=1 TO fachanz [1308]
3980 LOCATE 21,3+1 [776]
3990 PRINT USING"###";i:PRINT ". ";fach$(i) [2170]
4000 NEXT [350]
4010 LOCATE 3,15 [721]
4020 INPUT"Fach ";fach(d) [1030]
4030 LOCATE 3,15:PRINT" " [1639]
4040 LOCATE 9,5:PRINT" ":LOCATE [2344]
9,5
4050 PRINT fach$(fach(d)) [1792]
4060 CLS #3 [371]
4070 GOSUB 4500 [837]
4080 GOTO 3610 [686]
4090 ' [117]
4100 CLS #3 [371]
4110 LOCATE 21,5 [668]
4120 PRINT" . ." [780]
4130 LOCATE 21,5 [668]
4140 INPUT"",t(d) [457]
4150 LOCATE 24,5 [703]
4160 INPUT"",m(d) [545]
4170 LOCATE 27,5 [628]
4180 INPUT"",j(d) [601]

```

Listing: Schulnoten-Verwaltung

```

4190 LOCATE 3,9 [659]
4200 PRINT" . ." [780]
4210 LOCATE 3,9 [659]
4220 PRINT USING"###";t(d) [925]
4230 LOCATE 6,9 [328]
4240 PRINT USING"###";m(d) [598]
4250 LOCATE 9,9 [316]
4260 PRINT USING"###";j(d) [1008]
4270 CLS #3 [371]
4280 GOSUB 4500 [837]
4290 GOTO 3610 [686]
4300 ' [117]
4310 CLS #3 [371]
4320 LOCATE 21,5 [668]
4330 INPUT "Note: ",nn [1124]
4340 GOSUB 4410 [847]
4350 n(d)=nn [463]
4360 LOCATE 9,11 [539]
4370 PRINT n(d) [438]
4380 CLS #3 [371]
4390 GOSUB 4500 [837]
4400 GOTO 3610 [686]
4410 ' [117]
4420 IF art(d)=1 THEN GOTO 4440 [1966]
4430 IF art(d)=2 THEN GOTO 4470 [1677]
4440 schulanz(fach(d))=schulanz(fach(d))-1 [1630]
4450 schul(fach(d))=schul(fach(d))-n(d) [1217]
4460 GOTO 4490 [511]
4470 exanz(fach(d))=exanz(fach(d))-1 [1108]
4480 ex(fach(d))=ex(fach(d))-n(d) [1917]
4490 RETURN [555]
4500 ' [117]
4510 IF art(d)=1 THEN GOTO 4530 [1908]
4520 IF art(d)=2 THEN GOTO 4560 [1667]
4530 schulanz(fach(d))=schulanz(fach(d))+1 [1618]
4540 schul(fach(d))=schul(fach(d))+n(d) [1192]
4550 GOTO 4580 [501]
4560 exanz(fach(d))=exanz(fach(d))+1 [1128]
4570 ex(fach(d))=ex(fach(d))+n(d) [1759]
4580 RETURN [555]
4590 ' [117]
4600 ' UP Gesamtnoten errechnen [2048]
4610 ' [117]
4620 FOR i=1 TO fachanz [1308]
4630 IF exanz(i)=0 AND schulanz(i)=0 THEN [2918]
gnote(i)=0 :GOTO 4720
4640 IF exanz(i)=0 THEN 4680 [478]
4650 IF schulanz(i)=0 THEN 4700 [2069]
4660 gnote(i)=(ex(i)/exanz(i))+(schul(i)/ [3788]
schulanz(i)*sch(i))/(1+sch(i))
4670 GOTO 4720 [537]
4680 gnote(i)=schul(i)/schulanz(i) [1896]
4690 GOTO 4720 [537]
4700 gnote(i)=ex(i)/exanz(i) [1484]
4710 GOTO 4720 [537]
4720 NEXT [350]
4730 RETURN [555]
4740 ' [117]
4750 ' UP Taste druecken [1295]
4760 ' [117]
4770 LOCATE 6,25 [561]
4780 PRINT CHR$(24);" Bitte eine Taste dru [2938]
ecken! ";CHR$(24)
4790 a$=INKEY$ [278]
4800 IF a$="" THEN 4790 [784]
4810 RETURN [555]
4820 ' [117]
4830 ' UP yposi [156]
4840 ' [117]
4850 IF n(i)=6 THEN yposi=80 [1445]
4860 IF n(i)=5 THEN yposi=128 [1470]
4870 IF n(i)=4 THEN yposi=176 [1666]
4880 IF n(i)=3 THEN yposi=224 [1620]
4890 IF n(i)=2 THEN yposi=272 [1651]
4900 IF n(i)=1 THEN yposi=320 [1789]
4910 RETURN [555]
4920 ' [117]
4930 ' Data's Monate [664]
4940 ' [117]
4950 DATA J,F,M,A,M,J,J,A,S,O,N,D [1158]
4960 ' [117]
4970 ' Data's Faecher [1250]
4980 ' [117]
4990 DATA Mathe,2,Deutsch,2,Englisch,2 [2392]
5000 DATA Bio,0,Wirtschaft,0,Sozialkunde,0 [2015]
5010 DATA Musik,0,Chemie,1,Steno,0 [2380]
5020 DATA Rechnungs.,2,Physik,1,Religion,0 [2231]
5030 DATA Geschichte,0,Erzieh.,0,Maschin., [2130]
0

```

Listing: Schulnoten-Verwaltung

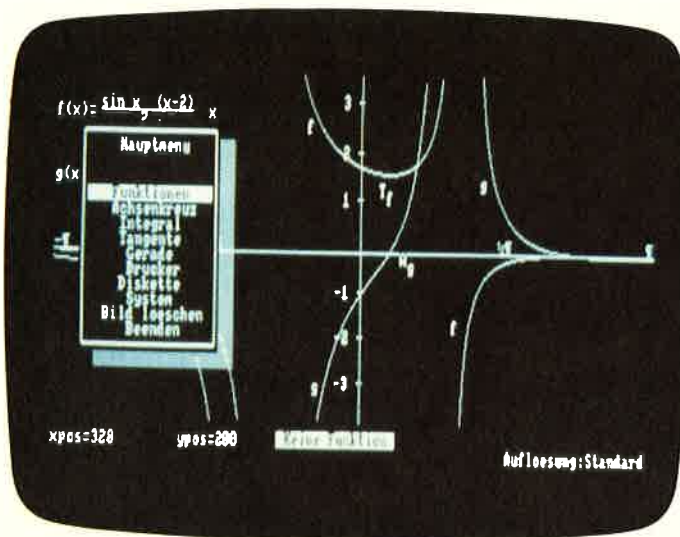


Bild 1. Dies ist ein Funktionsplot mit Hauptmenue

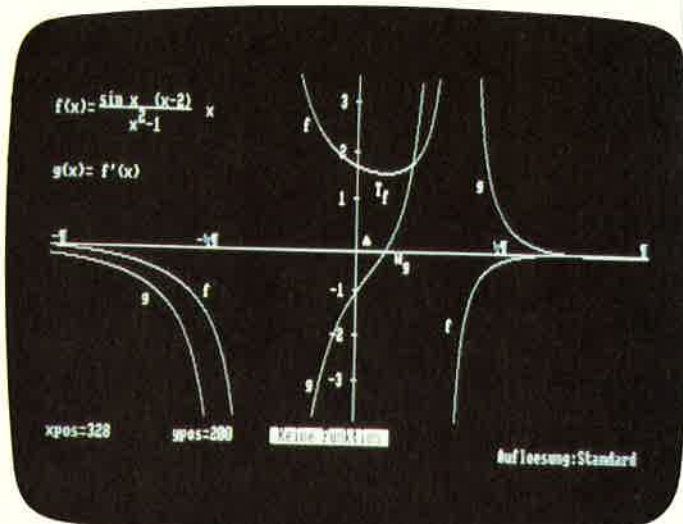


Bild 2. ... und dies ohne.

PLOT IT!

– oder wie erleichtert man sich das Leben mit der Mathematik!

PLOT IT! ist genau das, was sich ein Oberstufenschüler wünscht, der es leid ist, die dritte Ableitung von $f(x) = \sin(x) * ((x-2) / (x^2-1)) / x$ von Hand zu zeichnen und der sich die Fläche unter der Kurve $f(x) = \sqrt{4-x^2}$ von 0 bis 1 lieber vom Computer berechnen läßt, anstatt sich selbst die kleinen grauen Zellen zu ruinieren.

PLOT IT! überzeugt von Anfang an durch eine Vielzahl von Funktionen und eine konsequente Benutzerführung über alle Menue-Ebenen hinweg. Neu dagegen ist die Offenheit, mit der das System konzipiert worden ist: Der Benutzer kann das Programm problemlos an eigene Bedürfnisse anpassen, sämtliche Menuestrukturen sind fast beliebig erweiterbar (auch wenn eine Erweiterung wohl kaum nötig sein wird)!

Neu ist auch die UNDO-Funktion, die sonst fast nur bei größeren Rechnern zu finden ist: Der Benutzer kann die zuvor ausgeführte Funktion durch Tastendruck einfach rückgängig machen! Funktionen, bei denen das nicht der Fall ist, verfügen über vorherige Sicherheitsabfragen.

Nicht neu hingegen ist, daß die Systemwerte, so z.B. das Cursorzeichen oder die Bildschirmfarben, ebenso wie auch die Tastaturbelegung einfach den eigenen Vorstellungen angepaßt werden können, teilweise sogar über die Standardmenueführung. Normal in diesem Programm ist der überlegte Einsatz von DROP-DOWN- WINDOWS, mit denen man immer vor Augen hat, woran man gerade arbeitet, denn Übersichtlichkeit hilft Fehler vermeiden.

Dazu kommt noch die Möglichkeit, die fertigen Funktionsplots zu beschriften und zu kommentieren, auszudrucken oder auf Diskette zu speichern.

I. Das Worksheet

Worksheet ist der Bildschirm, auf dem die Funktionen geplottet und bearbeitet werden können und somit der Programmteil, auf dem sich ein großer Bereich der Arbeit abspielt.

a. Aufbau des Worksheets und Bearbeitungshilfen:

Das Worksheet, oder zu deutsch, das Arbeitsblatt, ist der wichtigste Teil des gesamten Programms, so nimmt es auch 23 der insgesamt 25 Bildschirmzeilen in Anspruch, 23 Zeilen, auf denen die Funktionen in das Achsenkreuz geplottet werden. Die unteren beiden Zeilen sind mit Statuszeigern belegt (deren Funktion vgl. I.b.).

Das gesamte Worksheet ist frei beschriftbar, d.h., man kann an jede beliebige Stelle jeden beliebigen Kommentar schreiben (die Tastaturbeschreibung siehe unter I.c.). Erreichen kann man die einzelnen Positionen mit Hilfe der CURSOR-Tasten, die jeweilige Position wird durch ein Icon (Standard: Pfeilspitze nach oben) angezeigt.

Um die einzelnen Bildschirmstellen besser ansteuern zu können, kann man mit der Taste <TAB> verschiedene Auflösungen wählen, deren Spektrum von "FEIN" (0.1 Pixel) bis "GROB" (64 Pixel) reicht.

Ganz genaue Cursorpositionierung läßt sich mit der Taste <ENTER> erreichen: hierbei kann die gewünschte Position als Zahlenwert eingegeben werden.

Eine gewisse Doppelfunktion hat die Taste <CAPS>. Zeichnet der Benutzer gerade eine Linie (anwählbar über das Hauptmenue, vgl. II.d.), so wird mit <CAPS> Anfangs- oder Endpunkt der Linie markiert.

Wird keine Linie gezeichnet und ist eine mathematische Funktion geplottet und angewählt (II.b.) so schaltet <CAPS> auf "TRACING" um, d.h. der Cursor wird nicht mehr frei bewegt, sondern fährt die aktuelle Kurve ab. Hierbei wird im Statusfeld Funktionswert und Steigung der Kurve angezeigt (I.c.). Nochmaliger Druck auf <CAPS> schaltet den "TRACING"-Modus wieder aus.

Mit der Taste <ESC> wird im Normalfall die UNDO-Funktion aktiviert, d.h., die zuletzt ausgeführte Aktion wird rückgängig gemacht. Zeichnet der Benutzer allerdings eine Linie, wird mit <ESC> nur die Linienziehungsfunktion abgeschaltet. Erst nochmaliger Druck auf <ESC> führt dann zum UNDO. Last but not least wird mit der <COPY>-Taste das Hauptmenue aufgerufen. Mit dem Druck auf <COPY> werden alle Aktionen in den Speicher übernommen, ein UNDO führt dann also immer zu dem Stand vor dem Menueaufruf.

b. Statusanzeigen im Worksheet

Die unteren beiden Zeilen des Bildschirms werden für Statusanzeigen genutzt.

In der oberen linken Ecke steht die aktuelle Cursorposition in Bildschirmkoordinaten, oder, falls der »TRACING«-Modus an-

gewählt ist, der aktuelle x-Wert, sowie der zugeordnete Funktionswert $f(x)$. In der Mitte der Zeile 24 kann die aktuell angeählte Funktion abgelesen werden. Sollte der Text der Funktion zu lang sein, wird er auf 56 Stellen gekürzt. Ist der »TRACING«-Modus angewählt, wird unter dem Funktionswert noch die Näherung der aktuellen Tangentensteigung $f'(x)$ angegeben. Am rechten Rand der Zeile 25 kann die aktuelle Cursorauflösung abgelesen werden, angegeben von »FEIN« bis »GROB«.

c. Tastaturbelegung im Worksheet

1.) Funktionstasten

<ESC>.....UNDO-Funktion
 <TAB>.....Cursorauflösung ändern
 <CAPS>.....- »TRACING«-Modus ein- / ausschalten.
 - Anfangs- / Endpunkt von Linien setzen.
 <ENTER>.....Cursor exakt positionieren
 <COPY>.....Hauptmenue aufrufen
 <JOYSTICK>.....Cursortasten

2.) Sonderzeichen

<CTRL> <1>Zeichen <1/4>
 <CTRL> <2>Zeichen <1/2>
 <CTRL> <3>Zeichen <3/4>
 <CTRL> <0>unendlich
 <CTRL> <Q>»Ä«
 <CTRL> <A>»Ö«
 <CTRL> <Z>»Ü«
 <CTRL> <W>»ä«
 <CTRL> <S>»ö«
 <CTRL> <X>»ü«
 <CTRL> <E>Copyright-Zeichen
 <CTRL> <R>Pi
 <CTRL> <T>Grafiksonderzeichen
 <CTRL> <eckige Klammer auf> .Integralzeichen oben
 <CTRL> <eckige Klammer zu> ...Integralzeichen mitte
 <CTRL> <Schrägstrich rechts> ...Integralzeichen unten

II. Die Menueauswahl

Die Auswahl der Menueoptionen erfolgt über die Pfeiltasten und die <COPY>-Taste. Die Menueauswahl kann mit <ESC> abgebrochen werden.

a. Hauptmenue

- 1.) Funktionen: siehe II.b.
- 2.) Achsenkreuz: Die Intervalle, mit denen das Achsenkreuz bestimmt wird, werden neu gesetzt. Diese Funktion kann nicht rückgängig gemacht werden. Es erfolgt eine Sicherheitsabfrage. Der Bildschirm wird bei dieser Aktion vollständig gelöscht.
- 3.) Integral: siehe II.c.
- 4.) Tangente: Falls der »TRACING«-Modus aktiviert ist, wird an die Funktion an der momentanen x-Position eine Tangente gezeichnet. Die Funktion läßt sich rückgängig machen. Die Länge der Tangente beträgt immer 100 Bildschirmpunkte äquivalent.
- 5.) Gerade: Schaltet die Bearbeitungsweise im Worksheet auf »Linien ziehen« (vergl. II.a.). Die Einstellung wird im Statusfeld angezeigt. Es wird keine unmittelbare Aktion ausgeführt.
- 6.) Drucker: siehe II.d.
- 7.) Diskette: siehe II.e.
- 8.) System : siehe II.f.
- 9.) Bild löschen: Das gesamte Worksheet wird bis auf das Achsenkreuz gelöscht. Die Funktion läßt sich nicht rückgängig machen. Es erfolgt eine Sicherheitsabfrage.

10.) Programm beenden: Das Programm wird beendet. Alle Variablen sind damit verloren, die Funktion läßt sich nicht rückgängig machen. Es erfolgt eine Sicherheitsabfrage. Neustart von BASIC mit <CTRL> <ENTER>.

b. Untermenue »FUNKTIONEN«

Es erscheint eine Liste aller im Speicher befindlichen Funktionen. Das Menue kann mit <ESC> abgebrochen werden. Nach Auswahl einer Funktion erscheint ein »AKTIONS«-Menue, welches ebenfalls abgebrochen werden kann. Die Punkte dieses Menues im einzelnen:

- 1.) Anwählen: Die ausgewählte Funktion wird zur aktuellen Funktion erklärt. Der »TRACING«-Modus verfolgt nun diese Funktion. Die Auswahl ist ungültig, falls die ausgewählte Funktion nicht definiert ist, und muß wiederholt werden.
- 2.) Eingeben: Die ausgewählte Funktion kann verändert werden.
- 3.) Plotten: Die ausgewählte Funktion wird geplottet. Die Auswahl muß wiederholt werden, falls die ausgewählte Funktion nicht definiert ist.
- 4.) Ableitung: Es wird die Ableitungsfunktion zur ausgewählten Funktion erstellt. Die Ableitung muß auf einen nicht belegten Funktionsspeicher (1-10) gelegt werden. Im Anschluß wird die Ableitung wie eine normale Funktion behandelt und kann auch gelöscht (II.b.7), geplottet oder auch wieder abgeleitet werden. Eine Editierung ist allerdings ausgeschlossen. Die Stammfunktion zur Ableitung ist solange vorm löschen geschützt, bis die Ableitung gelöscht wurde.
- 5.) Aufleitung: Es wird die Aufleitungsfunktion zur ausgewählten Funktion erstellt. Es gelten die gleichen Regeln wie bei der Ableitung (II.b.4).
- 6.) Umkehrung: Es wird die Umkehrfunktion zur ausgewählten Funktion erstellt. Es gelten die gleichen Regeln wie bei der Ableitung (II.b.4).
- 7.) Löschen: Die angewählte Funktion wird aus dem Funktionsspeicher gelöscht. Die Funktion kann anschließend nicht wiederhergestellt werden. Es erfolgt keine Sicherheitsabfrage.

c. Untermenue »INTEGRAL«

In diesem Untermenue können Integrale, Flächen und Volumina berechnet werden. Im Gegensatz zur Aufleitung von Funktionen werden hier reine Zahlenwerte berechnet. Die Berechnung bezieht sich auf die jeweils aktuelle Funktion. Ist keine Funktion ausgewählt, kann dieser Menuepunkt nicht angewählt werden. Das Menue läßt sich mit <ESC> abbrechen.

- 1.) Integralberechnung: Es wird das Integral über $f(x)$ im einzugebenden Intervall x_1/x_2 berechnet. Die Berechnung dauert etwa 30 Sekunden.
- 2.) Flächenberechnung: Wie II.c.1, jedoch wird die Fläche berechnet.
- 3.) Drehkörperberechnung: Wie II.c.1, jedoch wird das Drehkörper-Volumen berechnet.

Untermenue »DRUCKER«

Ein Diamant ist unvergänglich, Daten im Computer leider nicht. In diesem Menuepunkt können Daten, bzw. das Worksheet ausgedruckt werden. Das Menue ist mit <ESC> abbrechbar.

- 1.) Hardcopy: Das Worksheet wird ausgedruckt. Ist kein Drucker angeschlossen oder ist der angeschlossene Drucker nicht bereit, so wird ein Fehler ausgegeben und der Ausdruck unterbrochen. Der Druckvorgang an sich dauert etwa vier Minuten und läßt sich mit <ESC> abbrechen.
- 2.) Funktionsliste: Eine Liste aller im Speicher befindlichen Funktionen wird ausgedruckt. Ansonsten gilt das Gleiche wie bei II.d.1.

e. Untermenue »DISKETTE«

Mit Hilfe dieses Menuepunktes lassen sich die Funktionsplots auf Diskette sichern. Funktionsplots haben die Standard-Extension ».P«. Das Menue läßt sich mit <ESC> abbrechen.

- 1.) Bild laden: Ein zuvor gespeichertes Bild kann geladen werden. Hierzu muß der Name der zu ladenden Datei eingegeben werden. Das Achsenkreuz sollte mit dem Gespeicherten übereinstimmen.
- 2.) Bild speichern: Das aktuelle Worksheet wird als Bild gespeichert. Es kann unter BASIC oder mit Malprogrammen weiterverarbeitet werden. Die Eingabe eines gültigen Dateinamens ist zwingend.
- 3.) Disc-Katalog: Das aktuelle Inhaltsverzeichnis wird angezeigt.
- 4.) Laufwerk wechseln: Standard-Laufwerk ändern.
- 5.) Benutzer wechseln: Neuen Benutzerbereich anwählen. Entspricht dem BASIC-Befehl !USER, n. Bei Eingabe von E5 als Benutzerbereich können gelöschte Dateien geladen werden. Die Zahleneingabe erfolgt hexadezimal.

Untermenue »SYSTEM«

In diesem Menue können zwei System-Parameter geändert werden: das Cursorzeichen und die Farben. Das Menue läßt sich mit <ESC> abbrechen.

- 1.) Farben ändern: Der Benutzer wird aufgefordert, die neuen Bildschirmfarben einzugeben. Wird keine Eingabe vorgenommen, werden die aktuellen Bildschirmfarben beibehalten.
- 2.) Icon anpassen: Der Benutzer kann ein neues Cursorzeichen auswählen. Gültig sind alle Zeichen von 0 - 255. Eingegeben wird der ASCII-Code des gewünschten Zeichens (Standard = 244). Wird keine Eingabe vorgenommen, wird das aktuelle Zeichen übernommen.

III. Änderungen am Programm

a. Zeichen-(SYMBOL-) definitionen

Standardgemäß werden im Vorprogramm (PLOTTER.BAS) elf neue Zeichen definiert. Der Programmierer kann nun weitere Zeichen definieren, indem er zwischen der letzten Symbol-Definitionszeile und der Endzeile (DATA -1) weitere DATA-Zeilen der Form DATA symbol-nr.,N1,N2,N3,N4,N5,N6,N7,N8 einfügt.

b. Tastaturdefinitionen.

Ebenfalls standardgemäß werden im Vorprogramm (PLOTTER.BAS) 23 Tastaturdefinitionen vorgenommen. Der Programmierer kann nun weitere Definitionen vornehmen, indem er zwischen der letzten Tastatur-Definitionszeile und der Endzeile (DATA -1) weitere DATA-Zeilen der Form DATA tastennummer, standard, mit shift, mit ctrl einfügt.

c. Binärprogramm-Erweiterungen.

Es wird ein etwa 480 Byte langes Binärprogramm in den Speicher ab &A000 gelesen. Hinter diesem Binärteil sind noch etwa 500 Byte für private Binärprogramme frei. Der Programmierer sollte darauf achten, daß seine Binärprogramme nicht mit dem PLOT IT!-Programm kollidieren.

d. Menustruktur.

Die Menues sind alle, bis auf das »AKTIONEN«-Menue, offen konzipiert, d.h., der Programmierer kann einfach eigene Menuepunkte hinzufügen. Hierzu muß er in der entsprechenden DATA-Zeile am Ende des Hauptprogramms (PLOTTER.001) den entsprechenden Menuepunkt am Ende der Zeile hinzufügen und die Zahl am Anfang der DATA-Zeile um eins erhöhen. Die Größe der Menue-Windows wird automatisch an die Ver-

änderungen angepaßt. Um die Veränderungen nun auch anwählbar zu machen, muß der Benutzer die entsprechende Verteilerzeile finden, und an das »ON GOSUB«, bzw. »ON GOTO« die Zeilennummer seiner eigenen Routine anhängen.

IV. Adressen, Speicherbelegung, RSX-Befehle

a. Verwendete RSX-Befehle sowie Calls und deren Funktion.

WINDOW,x1,x2,y1,y2 definiert ein Window und begrenzt es mit Rahmen und Schatten.
SHIFT,n1,n2,n3 schiebt n3 Byte von n1 nach n2 (Z80-Befehle LDIR bzw. LDDR).
HC druckt eine Mode-2 Hardcopy.
BAR,n1 invertiert n1 Zeichen ab der aktuellen Cursorposition.
INSERT,s1,n1 fügt den String s1 als BASIC- Programmzeile n1 ein.
CALL &BB8A invertiert aktuelle Cursorposition.
CALL &BD19 wartet auf Strahlrücklauf.

b. Speicheraufteilung (alle Angaben in hexadezimaler Darstellung)

0000 - 016F: vom System belegt
0170 - 5FFF: frei für BASIC-Programm und Variablen
6000 - 9FFF: zweiter Bildschirmspeicher / UNDO- Buffer.
A000 - A1DF: Binärteil / RSX-Befehle
A1E0 - A5EB: frei für private Binärteile.
A5EC - A5FC: Symbolmatrizen (bei SYMBOL AFTER 234).
A5FD - BFFF: Disk-RAM, BASIC-RAM, System-RAM, Z80- Stack.
C000 - FFFF: Bildschirmspeicher.

Wichtige Programmzeilen im Hauptprogramm (PLOTTER.001).

```
10 ... Kaltstart
60 ... Hauptschleife !
70 ... Untermenue "FUNKTIONEN"
90 ... Untermenue "AKTIONEN"
200 ... Funktionsliste ausgeben / Funktion auswählen
220 ... Achsenkreuz ändern
330 ... Untermenue "INTEGRAL"
370 ... Tangente zeichnen
380 ... Modus "LINIE ZIEHEN" aktivieren
390 ... Untermenue "DRUCKER"
430 ... Untermenue "DISKETTE"
500 ... Untermenue "SYSTEM"
530 ... Bild löschen
540 ... Programm beenden
550 ... Unterprogramm "AUF TASTE WARTEN".
590 ... Input-Routine
650 ... Menue-Auswahl
740 ... Unterprogramm "WORKSHEET"
1050 ... Unterprogramm "FUNKTION PLOTTEN"
1100 ... Fehler-Abfangroutine
```

(M. Folz/JB)

```
1 REM --- [523]
2 REM --- PLOT IT! - Mehr als ein Funkt [2510]
  ionsplotter
3 REM --- [523]
4 REM --- [523]
5 REM --- (C) ESCape! Software 1987, gesc [4579]
  hrieben von Michael Folz
6 REM --- [523]
7 REM --- (G) russ an alle (Fast alle!) [2562]
8 REM --- [523]
9 REM [272]
10 CALL &BB00:CALL &BB4E:CALL &BBFF:CALL & [11619]
  BC65:CALL &BCA7:DEFINT b-z:DEFSTR a:POKE &
Listing: Plot it
```

```

BDEE,&C9:POKE &AC00,255:POKE &BE78,255:ON
ERROR GOTO 800:GOSUB 810:SYMBOL AFTER 234:
MEMORY &5FFF
20 GOSUB 30:GOSUB 170:GOSUB 460:PLOT-8,-8, [4357]
1:RUN"!plotter.001"
30 READ i:WHILE i<>-1:FOR j=0 TO 7:READ a: [8018]
z(j)=VAL("&"+a):NEXT:SYMBOL i,z(0),z(1),z(
2),z(3),z(4),z(5),z(6),z(7):READ i:WEND:RE
TURN
40 REM *** SYMBOL DEFINITIONEN (ab 234) [1843]
50 DATA 234,66,18,3C,66,66,7E,66,0 [1678]
60 DATA 235,66,3C,66,66,66,66,3C,0 [1396]
70 DATA 236,66,00,66,66,66,66,3C,00 [1658]
80 DATA 237,66,00,78,0C,7C,CC,76,00 [1561]
90 DATA 238,66,00,3C,66,66,66,3C,00 [1166]
100 DATA 239,66,00,66,66,66,66,3E,00 [1420]
110 DATA 245,00,00,66,99,99,66,00,00 [1048]
120 DATA 246,04,7E,04,00,00,00,00,00 [1015]
130 DATA 247,7C,82,82,84,40,40,40,40 [1077]
140 DATA 248,20,20,10,10,08,08,04,04 [1514]
150 DATA 249,02,02,02,02,21,41,41,3E [1725]
160 DATA -1 [199]
170 READ i:WHILE i<>-1:READ j1,j2,j3:KEY D [7138]
EF i,1,j1,j2,j3:READ i:WEND:RETURN
180 REM *** TASTATUR-DEFINITIONEN [2065]
190 DATA 72,&f4,&f0,&f8 [1130]
200 DATA 73,&f5,&f1,&f9 [1004]
210 DATA 74,&f6,&f2,&fa [788]
220 DATA 75,&f7,&f3,&fb [434]
230 DATA 76,224,224,224 [966]
240 DATA 77,13,13,13 [784]
250 DATA 66,1,1,1 [639]
260 DATA 68,9,3,3 [372]
270 DATA 70,2,2,2 [448]
280 DATA 64,&31,&21,168 [740]
290 DATA 65,&32,&22,169 [919]
300 DATA 57,&33,&23,170 [937]
310 DATA 24,&5e,160,245 [791]
320 DATA 67,&71,&51,234 [950]
330 DATA 59,&77,&57,237 [975]
340 DATA 17,&5b,&7b,247 [638]
350 DATA 69,&61,&41,235 [898]
360 DATA 60,&73,&53,238 [441]
370 DATA 19,&5d,&7d,248 [1085]
380 DATA 71,&7a,&5a,236 [856]
390 DATA 63,&78,&58,239 [571]
400 DATA 62,&63,&43,241 [943]
410 DATA 22,&5c,&60,249 [512]
420 DATA 58,&65,&45,164 [684]
430 DATA 50,&72,&52,165 [862]
440 DATA 51,&74,&54,144 [743]
450 DATA -1 [199]
460 FOR I=0 TO &1DD STEP &10:Z=0:FOR J=I T [6150]
O I+&F:READ A$:POKE J+&A000,VAL("&"+A$):Z=
Z+VAL("&"+A$):NEXT J:READ A$:IF VAL("&"+A$
)<>Z THEN GOTO 480
470 NEXT:CALL &A000:IF PEEK(&AC01)<>&C9 TH [5265]
EN POKE &A13F,&97:POKE &A14B,&AB:POKE &A14
C,&E7:RETURN ELSE RETURN
480 MODE 2:PRINT"Checksum error in Zeile"; [4816]
500+(I/&10)*10;" !!!":STOP
490 REM *** BINAER-TEIL (ab #A000) [2822]
500 DATA 21,12,A0,01,16,A0,C3,D1,BC,7E,B7, [2227]
C8,CD,5A,BB,23,07DC
510 DATA 18,F7,00,00,00,00,2A,A0,C3,6C,A0, [1889]
C3,45,A0,C3,8A,069D
520 DATA BB,C3,24,A1,C3,33,A1,C3,50,A1,57, [2438]
49,4E,44,4F,D7,07E6
530 DATA 53,48,49,46,D4,43,49,4E,D6,42,41, [2042]
D2,49,4E,53,45,0632
540 DATA 52,D4,48,C3,00,DD,6E,04,DD,66,05, [2759]
E5,DD,5E,02,DD,07C7
550 DATA 56,03,DD,4E,00,DD,46,01,B7,ED,52, [2163]
E1,30,0B,B7,ED,075E
560 DATA 4A,EB,B7,ED,4A,EB,ED,B8,C9,ED,B0, [2959]
C9,CD,13,A1,E5,0B48
570 DATA D5,1C,1C,14,14,14,25,2D,E5,D5,21, [2473]
00,00,11,18,4F,03EE
580 DATA CD,66,BB,D1,E1,15,7A,94,47,05,05, [2380]
48,24,2C,E5,CD,075E
590 DATA 75,BB,E1,3E,96,CD,5A,BB,3E,9A,CD, [3758]
5A,BB,10,FB,3E,08CA
600 DATA 9C,CD,5A,BB,7B,95,47,18,10,3E,95, [2786]
CD,5A,BB,7A,CD,07F9
610 DATA 6F,BB,21,0B,A1,CD,09,A0,E1,2C,E5, [3378]
CD,75,BB,10,E9,0855

```

Listing: Plot it

```

620 DATA 3E,93,CD,5A,BB,41,3E,9A,CD,5A,BB, [2791]
10,FB,21,0B,A1,0786
630 DATA CD,09,A0,7A,CD,6F,BB,CD,5A,BB,3E, [2922]
99,CD,5A,BB,E1,0963
640 DATA 2C,24,24,CD,75,BB,3E,16,CD,5A,BB, [2443]
3E,01,CD,5A,BB,06C8
650 DATA 3E,CF,41,04,04,CD,5A,BB,10,FB,3E, [3502]
16,CD,5A,BB,AF,0728
660 DATA CD,5A,BB,D1,E1,CD,66,BB,C3,6C,BB, [2400]
95,16,01,CF,CF,09B6
670 DATA 16,02,00,DD,5E,00,DD,6E,02,DD,56, [2781]
04,DD,66,06,25,0545
680 DATA 2D,15,1D,C9,DD,7E,00,47,CD,8A,BB, [2934]
3E,09,CD,5A,BB,0705
690 DATA 10,F6,C9,DD,66,03,DD,6E,02,23,7E, [2045]
23,66,6F,CD,A6,076E
700 DATA BD,DD,56,01,DD,5E,00,CD,00,B9,CD, [2618]
D2,E6,C3,03,B9,08B6
710 DATA 21,CA,A1,CD,BE,A1,11,00,00,21,DO, [2493]
A1,CD,BE,A1,06,078D
720 DATA 3C,AF,CD,2B,BD,30,FA,10,F8,3E,42, [4455]
CD,1E,BB,C0,21,07D9
730 DATA D7,A1,CD,BE,A1,21,00,00,E5,D5,CD, [2973]
FO,BB,47,87,80,0945
740 DATA CB,27,CB,27,32,DC,A1,D1,E1,13,E5, [2588]
D5,CD,FO,BB,47,09D1
750 DATA 87,80,47,3A,DC,A1,B0,4F,D1,E1,CD, [3508]
2B,BD,30,FB,79,090F
760 DATA CD,2B,BD,30,FB,1B,23,23,7C,B7,28, [2478]
CC,7D,FE,68,20,076B
770 DATA C7,13,13,7A,FE,02,20,A1,7B,FE,80, [3132]
20,9C,C9,7E,FE,0822
780 DATA FF,C8,CD,2B,BD,30,F7,23,18,F4,1B, [2662]
40,1B,33,07,FF,0781
790 DATA OD,OA,1B,4B,3C,00,FF,1B,4B,68,01, [3802]
FF,00,00,00,00,0386
800 WINDOW 5,36,8,15:PEN 3:PAPER 1:CLS:PRI [13263]
NT TAB(9)**** FEHLER ****:PRINT:PRINT" Com
puter zuruecksetzen und den Ladevorgan
g wiederholen!":LOCATE 3,7:PRINT" (C) ESCa
pe! Software 1987";:WHILE-1:WEND
810 MODE 1:TAG:FOR i=550 TO 260 STEP-5:PLO [10867]
T-2,-2,3:MOVE 600-i,i/1.5:PRINT"IT!";:PLOT
-2,-2,2:MOVE i,i/1.5:PRINT"PLOT";:NEXT:TAG
OFF:FOR i=6 TO 20:INK 2,i:INK 3,26-i:FOR j
=1 TO 50:NEXT:NEXT
820 PEN 1:a$=" PLOT IT! Mehr als [14475]
ein Funktionsplotter... (C) ESCape
! Software 1987, geschrieben von Michael F
olz fuer CPC International Gruss an Ev
a & Jochen "
830 b$=a$:c$="":WHILE c$="":i=1:WHILE LEN( [10421]
b$)-i>26 AND c$="":LOCATE 8,25:CALL &BD19:
PRINT MID$(b$,i,26);:i=i+1:c$=INKEY$:WEND:
b$=RIGHT$(b$,26)+a$:WEND:LOCATE 6,25:PRINT
"Lade PLOT IT! - bitte warten.":RETURN

```

```

//////////////////////////////////////////////////

```

```

1 REM --- [523]
2 REM --- PLOT IT! - Mehr als ein Funk [2413]
tionsplotter...
3 REM --- [523]
4 REM --- [523]
5 REM --- (C) ESCape! Software 1987, g [4579]
eschrieben von Michael Folz
6 REM --- [523]
7 REM --- (G) russ an alle (Fast all [2562]
e!)
8 REM --- [523]
9 REM [272]
10 DEFINT b-z:DEFSTR a:DEFREAL x,y:DIM am( [6653]
15,15),zm(15),af(10),xil(10),xi2(10):ON ER
ROR GOTO 1030:ON BREAK GOSUB 870
20 xrl=-4:xr2=4:yr1=-4:yr2=4:xs=1:xp=320:y [4291]
p=200:fs=1
30 i=1:WHILE zm(i-1)<>-1:READ zm(i):FOR j= [9062]
0 TO zm(i):READ am(i,j):NEXT:i=i+1:WEND:FO
R i=0 TO 3:READ as(i):NEXT:FOR i=0 TO 2:RE

```

Listing: Plot it

Anwendungen

```

AD f$(i),f(i):INK i,f(i):NEXT:BORDER f(2):
READ ic
40 MODE 2:WINDOW#1,1,80,24,25:WINDOW#2,1,8 [4940]
0,1,23:ORIGIN 0,40,0,640,40,400
50 GOSUB 840:GOSUB 950:GOSUB 910:GOSUB 960 [2117]
:GOSUB 880
60 GOSUB 690:z=1:GOSUB 650:IF zy=-1 THEN G [7945]
OSUB 890:GOTO 60 ELSE ON zy GOSUB 70,220,2
80,320,330,340,380,450,480,490:GOTO 60
70 GOSUB 890:az="Funktion anwaehlen":GOSUB [3684]
200
80 IF zy=-1 THEN 890 ELSE fs=zy [2477]
90 z=2:GOSUB 650:zy=zy-1:IF zy=0 THEN IF a [10354]
f(fs)=" THEN 90 ELSE GOSUB 970:GOTO 890 EL
SE IF zy=-2 THEN 70 ELSE ON zy GOTO 100,19
0,120,140,160,180
100 IF RIGHT$(af(fs),1)<>" AND af(fs)<>" [12345]
THEN GOSUB 1050:GOTO 70 ELSE az=RIGHT$(af(
fs),68):zm=67:WINDOW,3,77,23,23:PRINT"f(x
)":GOSUB 540:IF az=SPACE$(zm)THEN af(fs)
=" ELSE af(fs)="f(x)+"az+" "
110 az="DEF Fny"+CHR$(64+fs)+"(x)+"az+":D [4779]
EF Fny(x)=Fny"+CHR$(64+fs)+"(x):RETURN"+CH
R$(0):INSERT,@az,5999+fs:GOTO 70
120 IF af(fs)=" THEN 90 ELSE az="Ablage de [13323]
r Ableitung zu "+LEFT$(af(fs),31):GOSUB 20
0:IF zy=-1 THEN 70 ELSE IF RIGHT$(af(zy),1
)>"W"OR zy=fs THEN GOSUB 1050:GOTO 120 ELS
E af(zy)="f(x)=Ableitung zu "+LEFT$(af(fs)
,54)+CHR$(64+fs)
130 af(fs)=LEFT$(af(fs),72)+"X":az="GOSUB" [12353]
+STR$(5999+fs)+"":DEF Fny"+CHR$(64+zy)+"(x)
=(-Fny"+CHR$(64+fs)+"(x)+Fny"+CHR$(64+fs)+
"(x+.01))*100:DEF Fny(x)=Fny"+CHR$(64+zy)+
"(x):RETURN"+CHR$(0):INSERT,@az,5999+zy:f
s=zy:GOTO 70
140 IF af(fs)=" THEN 90 ELSE az="Ablage de [14153]
r Aufleitung von "+LEFT$(af(fs),29):GOSUB
200:IF zy=-1 THEN 70 ELSE IF RIGHT$(af(zy)
,1)>"W"OR zy=fs THEN GOSUB 1050:GOTO 140 E
LSE af(zy)="f(x)=Aufleitung von "+LEFT$(af(
fs),52)+CHR$(128+fs)
150 af(fs)=LEFT$(af(fs),72)+"Y":az="GOSUB" [10205]
+STR$(5999+fs)+"":DEF Fny"+CHR$(64+zy)+"(x)
=Fny"+CHR$(64+fs)+"(x):DEF Fny(x)=Fny"+CHR
$(64+zy)+"(x):RETURN"+CHR$(0):INSERT,@az,
5999+zy:fs=zy:GOTO 70
160 IF af(fs)=" THEN 90 ELSE az="Ablage de [13897]
r Umkehrung von "+LEFT$(af(fs),30):GOSUB
200:IF zy=-1 THEN 70 ELSE IF RIGHT$(af(zy)
,1)>"W"OR zy=fs THEN GOSUB 1050:GOTO 140 E
LSE af(zy)="f(x)=Umkehrung von "+LEFT$(af(
fs),53)+CHR$(196+fs)
170 af(fs)=LEFT$(af(fs),72)+"Z":az="GOSUB" [8819]
+STR$(5999+fs)+"":DEF Fny"+CHR$(64+zy)+"(x)
=Fny"+CHR$(64+fs)+"(x):RETURN"+CHR$(0):IN
SERT,@az,5999+zy:fs=zy:GOTO 70
180 IF af(fs)=" THEN 90 ELSE IF RIGHT$(af( [11440]
fs),1)>"W" THEN GOSUB 1050:GOTO 70 ELSE zy=
ASC(RIGHT$(af(fs),1))AND &3F:af(fs)=":x11(
fs)=xrl:xi2(fs)=xr2:IF zy<>32 THEN af(zy)
=LEFT$(af(zy),72)+" ":af(fs)=":x11(fs)=xr
1:xi2(fs)=xr2:GOTO 70 ELSE 70
190 IF af(fs)=" THEN 90 ELSE GOSUB 850:GOS [2944]
UB 890:GOSUB 970:GOTO 980
200 zz=0:GOSUB 680:WINDOW,10,70,5,16:FOR [8190]
i=1 TO 10:PRINT USING"##: ";i:IF af(i)="
THEN PRINT"nicht definiert."ELSE PRINT LEF
T$(af(i),57);
210 NEXT:PRINT:PRINT"*** az:;zz=4:zo=0:zm [4583]
=10:zy=fs:GOSUB 670:GOTO 610
220 az=" Achsenkreuz aendern":GOSUB 530:az [3639]
=UPPER$(az):IF az<>"J" THEN 890
230 WINDOW,7,25,16,21:PRINT"Achsenkreuz-G [11068]
renzen":LOCATE 1,3:PRINT"Xmin: ";az=STR$(
xrl):zm=8:GOSUB 540:xrl=VAL(az):LOCATE 1,4
:PRINT"Xmax: ";az=STR$(xr2):GOSUB 540:xr2
=VAL(az):IF xr2<xrl THEN 230
240 LOCATE 1,5:PRINT"Ymin: ";az=STR$(yrl) [9271]
:GOSUB 540:yrl=VAL(az):LOCATE 1,6:PRINT"Ym
ax: ";az=STR$(yr2):GOSUB 540:yr2=VAL(az):
IF yr2<yrl THEN 240 ELSE CLS#2:GOSUB 910:
GOSUB 880
250 az=" Funktionen zeichnen":GOSUB 530:GO [5472]
SUB 890:IF az<>"J" THEN RETURN
260 i=fs:FOR fs=1 TO 10:IF af(fs)<>" THEN [3484]

```

Listing: Plot it

```

GOSUB 970:GOSUB 980
270 NEXT:fs=i:RETURN [1185]
280 IF af(fs)=" THEN 890 ELSE z=3:GOSUB 65 [7621]
0:za=zy:GOSUB 850:GOSUB 890:zz=2:GOSUB 680
:IF zy=-1 THEN RETURN ELSE y=0:x1=(ABS(xi1
(fs))+ABS(xi2(fs)))/500:ON za GOTO 290,300
,310
290 FOR x=xi1(fs)TO xi2(fs)STEP x1:y=y+Fny [16226]
(x)*x1:NEXT:WINDOW,25,55,12,18:PRINT xi2(
fs):PRINT" CHR$(247):PRINT" CHR$(248)"
f(x) dx ="y:PRINT" CHR$(249):PRINT xi1(
fs):GOSUB 520:GOTO 890
300 FOR x=xi1(fs)TO xi2(fs)STEP x1:y=y+ABS [10251]
(Fny(x)*x1):NEXT:WINDOW,25,55,13,16:PRINT
:PRINT TAB(3)"A ="y"FE.":GOSUB 520:GOTO 89
0
310 FOR x=xi1(fs)TO xi2(fs)STEP x1:y=y+Fny [8960]
(x)^2*x1:NEXT:WINDOW,25,55,13,16:PRINT:PR
INT TAB(3)"V ="y*PI"VE.":GOSUB 520:GOTO 89
0
320 GOSUB 890:IF af(fs)=" OR zf=0 THEN RET [9758]
URN ELSE zz=0:GOSUB 680:ym=Fny1((xp-x0%)/x
f)*yf:MOVE xp-80,-ym*(80/xf)+yp:DRAW xp+80
,ym*(80/xf)+yp,1:GOTO 670
330 z1=-1:zf=0:GOSUB 890:zz=3:GOTO 680 [2327]
340 z=4:GOSUB 650:IF zy=-1 THEN 890 ELSE I [6576]
F(INP(&F500)AND 64)=0 THEN zz=7:GOSUB 680:
ON zy GOSUB 350,360:GOTO 340 ELSE GOSUB 10
60:GOSUB 650:GOTO 340
350 GOSUB 890:WIDTH 255:HC:PRINT#8,CHR$(6 [2865]
4):RETURN
360 WIDTH 80:PRINT#8,CHR$(64):PRINT#8,"Lis [10618]
te der aktuellen Funktionen":PRINT#8,STRIN
G$(80,"-"):FOR i=1 TO 10:PRINT#8,USING"##:
";i:PRINT#8,LEFT$(af(i),72):IF MID$(af(
i),72,1)<>" THEN PRINT#8,"#"ELSE PRINT#8
370 NEXT:RETURN [940]
380 GOSUB 890:z=5:GOSUB 650:IF zy=-1 THEN [5209]
890 ELSE ON zy GOSUB 390,400,410,420,430:G
OTO 380
390 GOSUB 440:LOCATE 1,9:PRINT":Lade Bilda [8476]
tei (*.P) ";az=":zm=8:GOSUB 540:GOSUB 89
0:zz=5:GOSUB 680:IF az=SPACE$(8)THEN RETUR
N ELSE LOAD"!"+az+".P",&6000:RETURN
400 GOSUB 440:LOCATE 1,9:PRINT":Speichere [7613]
Bilddatei (*.P) ";az=":zm=8:GOSUB 540:GO
SUB 890:zz=6:GOSUB 680:SAVE"!"+az+".P",b,&
6000,&4000:RETURN
410 GOSUB 440:LOCATE 1,8:GOTO 520 [2384]
420 IF PEEK(&A702)=0 THEN|B:RETURN ELSE|A: [2579]
RETURN
430 WINDOW,10,35,16,18:PRINT TAB(5)am(z,z [7628]
y):PRINT:PRINT" Neue Benutzernummer: ";az=
HEX$(PEEK(&A701),2):zm=2:GOSUB 540:POKE &A
701,VAL("&"+az):RETURN
440 WINDOW,10,75,15,23:a$="*.p":|DIR,@a$: [2529]
RETURN
450 GOSUB 890:z=6:GOSUB 650:IF zy=-1 THEN [3228]
890 ELSE ON zy GOTO 460,470
460 WINDOW,10,40,15,19:PRINT TAB(10)am(z, [10932]
zy):FOR i=0 TO 2:PRINT:PRINT" f$(i)TAB(27
)";az=RIGHT$(STR$(f(i)),2):zm=2:GOSUB 54
0:f(i)=VAL(az):INK i,f(i):BORDER f(2):NEXT
:GOTO 890
470 WINDOW,10,40,15,17:PRINT TAB(5)am(z,z [7979]
y):PRINT:PRINT" Neuer Icon-Code (ASCII) ";
az=RIGHT$(STR$(ic),3):zm=3:GOSUB 540:ic=V
AL(az):GOTO 890
480 az=" Bild loeschen":GOSUB 530:GOSUB 89 [4900]
0:IF az="J" THEN zf=0:CLS:GOSUB 910:GOTO 84
0 ELSE RETURN
490 az=" Programm beenden":GOSUB 530:IF az [12216]
="J" THEN MODE 2:PRINT:PLOT IT! beendet - N
eustart mit <Ctrl><Enter>.:PRINT:KEY 140,
"RUN"+CHR$(13):DELETE 6000-ELSE 890
500 ai="":WHILE ai="":ai=INKEY$:WEND:RETUR [3357]
N
510 CALL &BB8A:GOSUB 500:CALL &BB8A:RETURN [2394]
520 PRINT:PRINT TAB(5)"Weiter mit <Enter>." [6374]
";ai="":WHILE ai<>CHR$(13):GOSUB 500:WEND
:RETURN
530 WINDOW,20,40,18,20:PRINT az:PRINT:PRI [5455]
NT" Sicher (j/N): ";az="N":zm=1:GOSUB 540
:az=UPPER$(az):RETURN
540 zj=1:IF LEN(az)<zm THEN az=az+SPACE$(z [6220]
m-LEN(az))ELSE IF LEN(az)>zm THEN az=LEFT$(

```

Listing: Plot it

```

(az,zm)
550 zz=1:zx=POS(#0):zy=VPOS(#0):PRINT["az [4461]
"];
560 LOCATE zx+zz,zy:GOSUB 510:zi=ASC(ai):I [7603]
F zi=13 OR zi=224 THEN RETURN ELSE IF zi=2
42 AND zz>1 THEN zz=zz-1:GOTO 560 ELSE IF
zi=243 AND zz<zm THEN zz=zz+1:GOTO 560
570 IF zi=16 THEN az=LEFT$(az,zz-1)+RIGHT$ [10904]
(az,zm-zz)+" ELSE IF zi=127 AND zz>1 THEN
zz=zz-1:az=LEFT$(az,zz-1)+RIGHT$(az,zm-zz
)+" ELSE IF zi=3 THEN zj=zj XOR 1 ELSE 59
0
580 LOCATE zx+1,zy:PRINT az;:GOTO 560 [1839]
590 IF zi<32 OR zi>126 THEN 560 ELSE IF zj [10781]
THEN az=LEFT$(az,zz-1)+ai+MID$(az,zz,zm-z
z):IF zz<zm THEN zz=zz+1:GOTO 580 ELSE 580
ELSE MID$(az,zz,1)=ai:PRINT ai;:IF zz<zm
THEN zz=zz+1:GOTO 560 ELSE 560
600 zy=1 [562]
610 LOCATE 1,zy+zo:|BAR,zz:GOSUB 500:zi=AS [10774]
C(ai):IF zi=224 OR zi=13 THEN RETURN ELSE
IF zi=1 THEN zy=-1:RETURN ELSE LOCATE 1,zy
+zo:|BAR,zz:IF zi=240 THEN zy=zy-1 ELSE IF
zi=241 THEN zy=zy+1
620 IF zy<1 THEN zy=zm ELSE IF zy>zm THEN [2075]
zy=1
630 GOTO 610 [411]
640 GOSUB 880 [1060]
650 IF z=1 THEN|WINDOW,5,20,5,5+zm(z)+2:zz [7607]
=16 ELSE IF z=2 THEN|WINDOW,50,65,14,23:zz
=16 ELSE|WINDOW,20,50,8,8+zm(z)+4:zz=31
660 LOCATE(zz-LEN(am(z,0)))/2+1,1:PRINT am [10256]
(z,0):FOR zi=1 TO zm(z):LOCATE(zz-LEN(am(z
,zi)))/2+1,zi+3:PRINT am(z,zi):NEXT zo=3:z
m=zm(z):GOSUB 600:RETURN
670 LOCATE#1,1,2:PRINT#1,SPACE$(29);:RETUR [2419]
N
680 GOSUB 670:LOCATE#1,1,2:PRINT#1,CHR$(24 [5593]
)CHR$(32)am(7,zz)CHR$(32)CHR$(24);:RETURN
690 zp=0:z1=xil(fs)*xf+x0%-2:z2=xil2(fs)*xf [4885]
+x0%+2:PRINT CHR$(23)CHR$(1);:TAG
700 LOCATE#1,1,1:IF zf=0 THEN PRINT#1,USIN [11684]
G"xpos=### ypos=### ";xp;yp;:GOTO
720 ELSE x=(xp-x0%)/xf:yp=Fny(x)*yf+y0%;PR [720]
INT#1,USING"x=###.### f(x)=#####.### ";x;
Fny(x);
710 IF zp THEN LOCATE#1,12,1:PRINT#1,"nich [10250]
t definiert";:LOCATE#1,12,2:PRINT#1,SPACE$(
14);:zp=0 ELSE LOCATE#1,12,2:PRINT#1,USIN
G"f' =#####.### ";Fny1(x);
720 MOVE xp,yp:CALL &BD19:PRINT CHR$(ic);: [6163]
GOSUB 500:MOVE xp,yp:PRINT CHR$(ic);:MOVE
xp,yp:zi=ASC(ai)
730 IF zi=224 THEN TAGOFF:PRINT CHR$(23)CH [2268]
R$(0);:z1=0:GOTO 880
740 IF zi=240 AND NOT zf THEN yp=yp+xs ELS [13699]
E IF zi=241 AND NOT zf THEN yp=yp-xs ELSE
IF zi=242 THEN xp=xp-xs ELSE IF zi=243 THE
N xp=xp+xs ELSE IF zi>31 AND NOT zf THEN P
RINT ai;:xp=XPOS:yp=YPOS ELSE IF z1=0 THEN
770
750 IF zf THEN IF xp<z1 THEN xp=z1 ELSE IF [7777]
xp>z2 THEN xp=z2 ELSE ELSE IF xp<0 THEN x
p=0 ELSE IF xp>632 THEN xp=632 ELSE IF yp<
0 THEN yp=0 ELSE IF yp>360 THEN yp=360
760 IF z1 THEN IF z1=1 THEN MOVE zx1,zyl:D [14260]
RAW zx2,zy2:MOVE zx1,zyl:zx2=xp:zy2=yp:DRA
W zx2,zy2:IF zi=2 THEN z1=-1:zz=3:GOSUB 68
0:GOTO 700 ELSE ELSE IF zi=2 THEN z1=1:zx1
=xp:zyl=yp:zx2=xp:zy2=yp:zz=4:GOSUB 680:GO
TO 700 ELSE ELSE 770
770 IF zi=1 THEN IF z1 THEN z1=0:GOSUB 670 [14689]
:GOTO 700 ELSE GOSUB 890:GOTO 700 ELSE IF
zi=2 AND af(fs)<>" AND NOT z1 THEN zf=zf X
OR 1:LOCATE#1,12,2:PRINT#1,SPACE$(14);:GOT
O 700 ELSE IF zi=9 THEN GOSUB 830:GOTO 700
780 IF zi<>13 THEN 700 ELSE WINDOW SWAP 1 [2118]
790 IF zf THEN LOCATE 30,2:PRINT"x=";:az=S [5434]
TR$(x):zm=8:GOSUB 540:IF VAL(az)<xil(fs)OR
VAL(az)>xil2(fs)THEN 790 ELSE xp=VAL(az)*x
f+x0%:GOTO 820
800 LOCATE 30,2:PRINT"xpos=";:az=RIGHT$(ST [7127]
R$(INT(xp)),3):zm=3:GOSUB 540:IF VAL(az)<0
OR VAL(az)>640 THEN 800 ELSE xp=VAL(az)
810 LOCATE 30,2:PRINT"ypos=";:az=RIGHT$(ST [5528]
R$(INT(yp)),3):zm=3:GOSUB 540:IF VAL(az)<0

```

Listing:Plot it

```

OR VAL(az)>360 THEN 810 ELSE yp=VAL(az)
820 LOCATE 30,2:PRINT SPACE$(12);:WINDOW S [3767]
WAP 1:GOTO 700
830 xs=xs*8:IF xs>64 THEN xs=0.125 [2486]
840 LOCATE#1,61,2:PRINT#1,"Aufloesung:"as( [3615]
LOG10(xs)+1)CHR$(18);:RETURN
850 |WINDOW,7,25,16,19:PRINT"Intervallgren [9325]
zen":LOCATE 1,3:PRINT"xmin: ";:az=STR$(xil
(fs)):zm=8:GOSUB 540:xil1(fs)=VAL(az):IF xi
1(fs)<xr1 THEN xil1(fs)=xr1
860 LOCATE 1,4:PRINT"xmax: ";:az=STR$(xil2( [10699]
fs)):GOSUB 540:xil2(fs)=VAL(az):IF xil2(fs)<
=xil1(fs)THEN 850 ELSE IF xil2(fs)>xr2 THEN
xil2(fs)=xr2
870 RETURN [555]
880 |SHIFT,&C000,&6000,&4000:RETURN [1624]
890 |SHIFT,&6000,&C000,&4000 [1075]
900 WINDOW 1,80,1,25:LOCATE#1,1,2:PRINT#1, [3767]
CHR$(18);:GOSUB 960:GOTO 840
910 yf=360/(ABS(yr1)+ABS(yr2)):xf=640/(ABS [6538]
(xr1)+ABS(xr2)):y0%=yf*ABS(yr1):x0%=xf*ABS
(xr1):MOVE 0,y0%:DRAW 640,0:MOVE x0%,0:DR
AWR 0,360
920 FOR zi=1 TO fs:IF xil(zi)<xr1 THEN xil [2676]
(zi)=xr1
930 IF xil2(zi)>xr2 THEN xil2(zi)=xr2 [1196]
940 NEXT:RETURN [940]
950 FOR i=1 TO 10:xil(i)=xr1:xil2(i)=xr2:NE [2569]
XT:RETURN
960 LOCATE#1,30,1:IF af(fs)=" THEN PRINT#1 [11862]
,CHR$(24)" Keine Funktion "CHR$(24)CHR$(18
);:RETURN ELSE PRINT#1,USING"##: ";fs;:PRI
NT#1,LEFT$(af(fs),47);:GOTO 970
970 ON fs GOSUB 6000,6001,6002,6003,6004,6 [6500]
005,6006,6007,6008,6009:DEF Fny1(x)=(-Fny(
x)+Fny(x+0.01))*100:RAD:RETURN
980 zz=1:GOSUB 680:GOSUB 960:zz=ASC(RIGHT$( [13647]
(af(fs),1)):zp=-1:z1=(ABS(xr1)-ABS(xil1(fs)
))*xf:z2=(ABS(xr1)+ABS(xil2(fs)))*xf:IF zz>
196 THEN za=SGN(Fny1(xil1(fs))):MOVE Fny(xil
1(fs))*yf+y0%,z1 ELSE MOVE z1,Fny(xil1(fs)
)*yf+y0%:y=0
990 FOR z=z1 TO z2:IF zz>196 THEN IF SGN(F [10774]
ny1((z-x0%)/xf))<>za THEN z=z2+1:GOTO 1020
ELSE zx=Fny((z-x0%)/xf)*xf+x0%:y=(z-x0%)/
xf ELSE IF zz>128 THEN y=y+Fny((z-x0%)/xf)
/xf:zx=z ELSE y=Fny((z-x0%)/xf):zx=z
1000 zy=y*yf+y0%:IF zy>360 OR zy<0 THEN zp [5431]
=-1 ELSE IF zp THEN MOVE zx,zy:zp=0 ELSE D
RAW zx,zy
1010 IF INKEY$=CHR$(1)THEN z=z2+1 [574]
1020 NEXT:GOTO 670 [985]
1030 IF ERR=2 THEN|WINDOW,27,53,10,15:PRIN [12528]
T" SYNTAX ERROR in Funktion.":PRINT:PRINT"
Plotvorgang unterbrochen.":PRINT" Bitte F
ehler korrigieren!":GOSUB 520:RESUME 890
1040 zp=-1:RESUME NEXT [2035]
1050 |WINDOW,27,53,9,14:PRINT TAB(5)"Zugri [10584]
ff verweigert.":PRINT:PRINT" Die Funktion
ist zur Zeit fuer Bearbeitung gesperrt.";
:GOTO 520
1060 |WINDOW,27,53,9,11:PRINT TAB(4)"Druck [5964]
er nicht bereit.":GOSUB 520:GOTO 890
1070 DATA 10,Hauptmenu,Funktionen,Achsenkr [5549]
euz,Integral,Tangente,Gerade,Drucker,Diske
tte,System,Bild loeschen,Beenden
1080 DATA 7,Aktionen,Anwaehlen,Eingeben,Pl [3158]
otten,Ableitung,Aufleitung,Umkehrung,Loesc
hen
1090 DATA 3,Integral,Berechnung,Flaechenbe [5029]
rechnung,Drehkoerpervolumen
1100 DATA 2,Drucker,Hardcopy,Funktionslist [1580]
e
1110 DATA 5,Diskette,Bild laden,Bild speic [4836]
hern,Disk Katalog,Laufwerk wechseln,Benutz
er wechseln
1120 DATA 2,System,Farben aendern,Iconanpa [3098]
ssung
1130 DATA 7,Warten,Plot,Rechnung,Gerade 1. [5207]
Punkt,Gerade 2.Punkt,Laden,Speichern,Druck
en
1140 DATA -1 [199]
1150 DATA Fein,Standard,Mittel,Grob [2708]
1160 DATA Hintergrund,0,Vordergrund,26,Ran [3726]
dfarbe,0
1170 DATA 244 [250]

```

Listing:Plot it



Bild 1: Hier wird das Menüfile erstellt.



Bild 2: Das fertige Menüfile.

Der Menue-Generator

Beim Ansehen des CATALOGS der Diskette sieht man oftmals sehr seltsame Filenamen, bei denen man auf Anhieb nicht genau weiß, welche Funktion diese Programme innerhalb des Computers ausüben, um es einmal vorsichtig zu formulieren. Oder wußten sie z.B. direkt, daß die Datei "SCHPLAVER.BAS" eine Schallplatten - Verwaltung beinhaltet?

Viel komfortabler ist da schon ein Programm, welches Ihnen auf Anhieb sagt, wie die Datei richtig heißt und sie danach auch noch in den Computer lädt und automatisch startet. Der Menue-Generator ist so ein Programm.

Erstellen eines Menues

Nach dem Start des Menue-Generators wird im oberen Bildschirmteil das Directory angezeigt. Der CPC fragt jetzt nach dem Namen des zu erstellenden Menues. Man kann hier auch einfach die ENTER- (RETURN-) Taste drücken, dann wird als Name automatisch "0" angenommen. Dies kann ganz nützlich sein, denn ein Ladefile mit dem Namen "0" läßt sich gut mer-

ken und leicht starten. (Beim CPC 6128 betätigt man bei gedrückter CONTROL-Taste nacheinander die "0" im Zehnerblock, die Cursor-Links-Taste und die kleine ENTER-Taste. Durch diese einfache Tastenkombination wird das File "0" gestartet).

Als Zweites kann man die Überschrift für das Menue eingeben. Danach ist noch die Anzahl der Programme (nicht der Files) einzutragen. Ist dieses auch geschehen, werden nacheinander die jeweiligen Namen der Programme und die dazugehörigen Loaderfiles eingetippt. Beim Programmnamen kann man ruhig die volle Bezeichnung angeben - bis zu 30 Zeichen sind möglich. Wenn alle Eingaben abgeschlossen sind, wird das Menüfile auf die Diskettenseite geschrieben. Das geschieht zuerst in Form einer ASCII-Datei, die dann noch vollautomatisch in ein Basic-Programm umgewandelt wird.

Benutzen des Menues

Nachdem das Menüfile aufgerufen wurde (wie oben beschrieben), kann man entweder mit Joystick 1, Joystick 2 oder den Cursorstasten einen Pfeil auf und ab bewegen. Wenn der Pfeil auf dem Namen des zu ladenden Programms zeigt, drückt man den Feuerknopf/Copy und sofort wird das dazugehörige Loaderfile geladen und gestartet. (K. Multerer/JB)

FÜR NIX GIBT'S NIX

Für den
Programmmhit
des Monats
1.000,-

Und für den
Top-Tip
des Monats
500,-

Das sind doch gute Argumente, Ihr Programm auch einmal zum Hit des Monats werden zu lassen. Bitte richten Sie Ihre Einsendungen an:

DMV Daten & Medien Verlagsgesellschaft mbh - Fuldaerstr 6 - 3440 Eschwege

für 464-664-6128



```

10 ' ***** Menue-Generator ***** [1304]
20 ' [117]
30 ' geschrieben im Juli 1987 [2269]
40 ' von Klaus Multerer (Multi) [2819]
50 ' [117]
60 ' ***** ----- ***** [1091]
70 ' [117]
80 ' *** Initialisierung *** [1272]
90 ' [117]
100 MODE 1:CLS:INK 0,3:INK 1,15:INK 2,24:I [4379]
NK 3,26:BORDER 3
110 SYMBOL 240,102,68,102,102,102,102,60,0 [2376]
120 SYMBOL 241,102,0,102,102,102,62,0 [2755]
130 SYMBOL 242,60,102,102,108,102,102,106, [2801]
96
140 LOCATE 14,2:PEN 2:PRINT "MEN"+CHR$(240 [3565]
)+"-GENERATOR":PRINT:PRINT:PEN 1
150 PLOT 200,388,3:DRAW 434,388:DRAW 434,3 [3602]
64:DRAW 200,364:DRAW 200,388
160 PLOT 194,394:DRAW 440,394:DRAW 440,358 [3013]
:DRAW 194,358:DRAW 194,394
170 DRAW 200,388:PLOT 434,388:DRAW 440,394 [3518]
:PLOT 440,358:DRAW 434,364:PLOT 194,358:DR
AW 200,364
180 LOCATE 4,12:PRINT "Bitte gew'+CHR$(241 [5569]
)+"-nschte Diskette einlegen"
190 LOCATE 10,14:PRINT "und eine Taste dr" [3314]
+CHR$(241)+"cken"
200 WHILE INKEY$<>"" :WEND [1786]
210 WHILE INKEY$="" :WEND [1607]
220 PRINT STRING$(50,CHR$(11)) [1531]
230 ' [117]
240 ' *** Bildschirmaufbau *** [1678]
250 ' [117]
260 MODE 2:WINDOW 5,77,2,16:WINDOW #1,3,78 [6257]
,19,24:DIR:PAPER #1,1:PEN #1,0:CLS #1
270 PLOT 4,4:DRAW 636,4:DRAW 636,122:DRAW [2115]
4,122:DRAW 4,4
280 PLOT 8,8:DRAW 632,8:DRAW 632,118:DRAW [2163]
8,118:DRAW 8,8
290 PLOT 4,130:DRAW 636,130:DRAW 636,394:D [2360]
RAW 4,394:DRAW 4,130
300 PLOT 8,134:DRAW 632,134:DRAW 632,390:D [3986]
RAW 8,390:DRAW 8,134
310 PRINT #1,"Wie soll das Men"+CHR$(241)+ [3951]
"file hei"+CHR$(242)+"en"
320 LOCATE #1,29,1:INPUT #1,name$:IF name$ [3411]
="" THEN name$="0":LOCATE #1,31,1:PRINT #1
,"0"
330 a=INSTR(name$,"."):IF a=0 THEN ex=0 EL [2418]
SE ex=1
340 IF ex=1 AND a>9 OR ex=1 AND a=1 OR ex= [6568]
1 AND LEN(name$)>12 THEN LOCATE #1,30,1:PR
INT #1,SPACES(47):GOTO 320
350 IF ex=1 AND LEN(name$)-st>3 OR ex=1 AN [3155]
D st=1 THEN LOCATE#1,30,1:PRINT#1,SPACES(4
7):GOTO 320
360 IF ex=0 AND LEN(name$)>8 THEN LOCATE # [4178]
1,40,1:PRINT #1,SPACES(47):GOTO 320
370 LOCATE #1,1,2:PRINT #1,SPACES(150):LOC [4620]
ATE #1,1,2:PRINT #1,"Wie soll die "+CHR$(2
40)+"berschrift des Men"+CHR$(241)+"'s lau
ten";
380 LOCATE #1,43,2:INPUT #1,ueber$:IF LEN( [3247]
ueber$)>40 THEN GOTO 370
390 INPUT #1,"Wieviele Programme";progs:IF [4785]
progs<1 OR progs>20 THEN CLS #1:GOTO 390
400 IF progs>10 THEN DIM nam$(progs):DIM 1 [2089]
os$(progs)
410 ' [117]
420 ' *** Abfrage der Namen *** [2278]
430 ' [117]
440 CLS #1 [373]
450 FOR i=1 TO progs [384]

```

```

460 LOCATE #1,3,2:PRINT #1," Name Progra [3690]
mm Nummer ";i:INPUT #1,nam$(1)
470 IF nam$(1)="" THEN 460 ELSE IF LEN(nam [3900]
$(1))>30 THEN CLS #1:GOTO 460
480 IF i=progs THEN PEN 0:KEY 128,CHRS(13) [11231]
+"a$="+CHR$(34)+"0"+CHR$(34)+"":!era,@a$:sa
ve"+CHR$(34)+"0"+CHR$(34)+CHR$(13)+"call &
bb00:pen 1:cls"+CHR$(13):KEY DEF 18,0,128
ELSE PEN 1
490 LOCATE #1,3,4:PRINT #1," Loadname Pro [8962]
gramm Nummer";i:INPUT #1,lo$(1):IF lo$(1)
="" THEN WHILE INKEY$<>"" :WEND:GOTO 490
500 st=INSTR(lo$(1),"."):IF st=0 THEN ex=0 [2987]
ELSE ex=1
510 IF ex=1 AND st>9 OR ex=1 AND st=1 OR e [8488]
x=1 AND LEN(lo$(1))>12 OR ex=1 AND LEN(lo$
(1))-st>3 THEN LOCATE #1,3,4:PRINT #1,SPAC
Es(73);:WHILE INKEY$<>"" :WEND:GOTO 490
520 IF ex=0 THEN IF LEN(lo$(1))>8 THEN LOC [7370]
ATE #1,3,4:PRINT #1,SPACES(73);:WHILE INKE
Ys<>"" :WEND:GOTO 490
530 CLS #1 [373]
540 NEXT [350]
550 ' [117]
560 ' *** Erzeugung der ASCII-Datei *** [2936]
570 ' [117]
580 MODE 1:CLS [1073]
590 OPENOUT name$ [700]
600 PRINT #9,"10 MODE 1:INK 0,0:BORDER 0:I [2809]
NK 1,15:INK 2,6:INK 3,26"
610 PRINT #9,"20 PAPER 0:PEN 3:CLS" [1967]
620 xp=(40-LEN(ueber$))/2+1 [1204]
630 PRINT #9,"30 LOCATE "+STR$(xp)+" ,2:PEN [3807]
1:PRINT "+CHR$(34)+ueber$+CHR$(34)+" :LOC=
5"
640 lin=30 [543]
650 FOR i=1 TO progs [384]
660 IF progs>11 THEN zeil=i+3 ELSE zeil=i* [3123]
2+3
670 lin=lin+10 [867]
680 PRINT #9,STR$(lin)+" LOCATE 5,"+STR$(z [13658]
eil)+" :pen 3:print "+CHR$(34)+CHR$(1+64)+C
HR$(34)+" :pen 2:print "+CHR$(34)+" - "+CH
R$(34)+" :pen 3:print "+CHR$(34)+nam$(1)+C
HR$(34)
690 NEXT [350]
700 IF progs>11 THEN locc=4 ELSE locc=5 [2025]
710 IF progs>11 THEN ste=1:zeil=progs+4:ze [7299]
il=progs+3:ze=4 ELSE ste=2:zeil=progs*2+5
:zeil=progs*2+3:ze=5
720 linn=lin+10:PRINT #9,STR$(lin+10)+"FOR [2945]
I=1 TO 100:NEXT I"
730 PRINT #9,STR$(lin+20)+" IF INKEY(49)=0 [9574]
OR INKEY(73)=0 OR INKEY(2)=0 THEN LOC=LOC
+"+STR$(ste)+" :IF LOC="+STR$(zeil)+" then
LOC="+STR$(zeil)
740 PRINT #9,STR$(lin+30)+" LOCATE 4,LOC-" [6954]
+STR$(ste)+" :PRINT "+CHR$(34)+" "+CHR$(34)
+" :LOCATE 4,LOC:PRINT CHR$(243)"
750 PRINT #9,STR$(lin+40)+" IF INKEY(48)=0 [10619]
OR INKEY(72)=0 OR INKEY(0)=0 THEN LOC=LOC
-"+STR$(ste)+" :IF LOC=3 THEN LOC="+STR$(ze
)
760 PRINT #9,STR$(lin+50)+" LOCATE 4,LOC+" [5749]
+STR$(ste)+" :PRINT "+CHR$(34)+" "+CHR$(34)
+" :LOCATE 4,LOC:PRINT CHR$(243)"
770 PRINT #9,STR$(lin+60)+" IF INKEY(76)=0 [6220]
OR INKEY(52)=0 OR INKEY(9)=0 THEN FLAG=-1
"
780 li=lin+60 [284]
790 FOR i=1 TO progs [384]
800 IF progs>11 THEN po=i+3 ELSE po=i*2+3 [2008]
810 li=li+10 [340]
820 PRINT #9,STR$(li)+" IF FLAG AND LOC="+ [6606]
STR$(po)+" THEN CALL &BB03:RUN"+CHR$(34)+1
os$(1)+CHR$(34)
830 NEXT i [375]
840 PRINT#9,STR$(li+10)+" GOTO "+STR$(linn [3428]
)
850 CLOSEOUT [902]
860 ' [117]
870 ' *** Umwandlung in Basicprogramm [1640]
880 ' [117]
890 POKE &AC00,1:LOAD"0" [953]

```

Listing: Menue-Generator

Listing: Menue-Generator



Angeregt zum Schreiben dieses Programmes wurde ich durch einen Freund, welcher mir eines Tages mal wieder eines seiner Fraktale á la »Apfelmännchen« auf dem Atari ST präsentierte. Auf dem Schirm war gerade eine ganz besonders filigrane Struktur zu erkennen. »Sieh mal, was jetzt passiert« sagte er zu mir, »klickte« mit der Maus ein farbenprächtiges Menue an, veränderte im Handumdrehen einige der Farbtöne und wählte dann einen weiteren Menüepunkt an, worauf das »Apfelmännchen« in Bewegung geriet und die Farben ineinander überliefen. »Fantastisch«, dachte ich mir, »das müßte der CPC aber auch schaffen...«.

Kaum zu Hause angekommen, machte ich mich auch schon an die Arbeit: ein Programm, mit dem man Farben innerhalb solch kurzer Zeit verändern und in einander überfließen lassen kann, wollte ich auch für meinen Computer besitzen.

Als erstes wurde die Routine für das Animieren der Farben geschrieben, nach und nach stellte ich immer höhere Anforderungen an mein Projekt. So sollte es denn auch möglich sein, die Farben – wie beim Atari ST – per Menue-Leiste auszuwählen, den Screen zu invertieren und die zusammengestellten Farben dann auch auf Disk-/Kassette zu sichern etc, etc.

Als das Programm dann endlich fertig war, ließ ich es auch schon auf meine Fraktale los und habe somit manche Stunde die Farbenpracht des CPC's bewundert...

Doch nun zur eigentlichen Handhabung des Programmes, welches – nachdem Sie es sorgfältig in Ihren Rechner eingegeben und abgespeichert haben – mit RUN gestartet werden muß. Alsdann gibt der CPC das Inhaltsverzeichnis der eingelegten Diskette aus, worauf Sie aufgefordert werden, den Namen des Screens, den Sie laden wollen, einzugeben, was Sie dann bitte auch tun.

Der CPC fragt Sie nach dem Screenmodus, in welchem der Screen dargestellt ist. Sie haben die Wahl zwischen Mode 0,

FARBENEINSTELLER

Ein Programm zum komfortablen Einstellen aller Bildschirmfarben Ihres CPC's!

Bisher war es immer recht kompliziert und knifflig, für ein liebevoll gemaltes Bild oder ein in stundenlanger Rechenzeit (vom Computer) errechnetes Fraktal letztendlich auch die passenden Bildschirmfarben aus der Farbpalette des CPC's auszusuchen – zumal die im Handbuch aufgeführten Farbtöne sowieso recht wenig mit den tatsächlichen Farben der CPC's zu tun haben. Das wird jetzt anders, denn der »Farbeneinsteller« erleichtert Ihnen diese Arbeit.

Mode 1 sowie Mode 2. Der CPC lädt den Screen, das Hauptmenue erscheint und Sie können die aufgeführten Menüepunkte mittels der Cursortasten <HOCH> und <RUNTER> auswählen und mit <COPY> quittieren.

Sie haben die Möglichkeit, über die einzelnen Menüepunkte

- a) den Screenmodus zu ändern,
- b) die Farben zu animieren,
- c) die Farben zu verändern,
- d) die Farbeinstellung abzuspeichern,
- e) den Screen normal darzustellen
- f) invers auf den Schirm zu bringen,
- g) das Programm zu verlassen,
- h) einen neuen Screen zu laden und
- i) den gesamten Screen zu kopieren.

Nach dem Selektieren eines jeden Menüepunktes öffnet sich ein Textfenster, welches Ihnen die Funktion des soeben angewählten Menüepunktes verdeutlicht. Zum Verändern der Farben benutzen Sie unter dem Menüepunkt "Farben ändern" ebenfalls die Cursortasten <HOCH> und <RUNTER> sowie <LINKS> und <RECHTS> zum Selektieren der Farbe.

Wird unter dem gleichen Menüepunkt <COPY> und eine der Cursortasten <HOCH> oder <RUNTER> gleichzeitig gedrückt, wird die Randfarbe verändert. Das Zeichen ganz rechts an der Farb-Einstell-Leiste wird angewählt, um den Screen zu invertieren (Sie kriegen dann allerdings Probleme mit der Farbzuoordnung), das Zeichen davor dient zum Verlassen des Menüepunktes. Die restlichen Menüepunkte erklären sich alle von selbst recht gut. Das Programm selber ist - wie ich meine - mit ausreichend vielen REMs ausgestattet, so daß die Einbringung eigener Ideen ins Programm nicht allzu schwierig sein dürfte.

(M. Wessel/JB)

```

für 464-664-6128
100 ' [117]
110 '=====[1467]
120 '= Farbeinsteller = [756]
130 '= (C) 1987 by = [1185]
140 '= Michael Wessel = [566]
150 '= 2000 Hamburg 61 = [1252]
160 '=====[1467]
Listing Farbeinsteller
    
```

```

170 ' [117]
180 'Maschinenprogramm [1465]
190 ' [117]
200 DEFINT a-z [553]
210 MEMORY &4FFF [174]
220 FOR adr=&5000 TO &5065:READ a$:POKE ad [3248]
   r,VAL("&" +a$):NEXT
230 DATA 01,09,50,21,25,50,C3,D1 [1462]
240 DATA BC,14,50,C3,2A,50,C3,3C [1518]
250 DATA 50,C3,4E,50,53,56,53,43 [675]
Listing Farbeinsteller
    
```



```

260 DATA D2,4C,44,53,43,D2,53,43 [1657]
270 DATA 52,49,4E,D6,00,FC,A6,09 [611]
280 DATA 50,00,FE,01,C0,DD,56,01 [1611]
290 DATA DD,5E,00,21,00,C0,01,FF [653]
300 DATA 3F,ED,B0,C9,FE,01,C0,DD [2030]
310 DATA 66,01,DD,6E,00,11,00,C0 [2190]
320 DATA 01,FF,3F,ED,B0,C9,21,00 [1305]
330 DATA C0,7E,EE,FF,77,7D,D6,FF [1564]
340 DATA CA,5F,50,23,C3,51,50,7C [919]
350 DATA D6,FF,C8,C3,5B,50,00,00 [1439]
360 CALL &5000:svscr=adr+1 [1409]
370 ' [117]
380 'Dimensionierungen [2081]
390 ' [117]
400 DIM c(15) [449]
410 ENV 1,15,-1,5 [644]
420 BNT -1,20,5,1,40,-10,1,20,5,2 [1514]
430 FOR i=2 TO 15 [610]
440 c(i)=i+10 [388]
450 INK 1,i+5 [654]
460 NEXT [350]
470 ' [117]
480 'Farben [742]
490 ' [117]
500 BORDER 0 [1008]
510 MODE 2 [513]
520 c(1)=26 [470]
530 INK 0,0:INK 1,26 [418]
540 ' [117]
550 'Screen [744]
560 ' [117]
570 RESTORE 690 [729]
580 READ x,y,xx,yy:GOSUB 2390 [1256]
590 LOCATE #6,9,1 [681]
600 PRINT #6,"Farbeneinsteller "+CHR$(164) [5687]
+" 1987 by Michael Wessel . MW-SOFT 7.7.19
87"
610 READ x,y,xx,yy:GOSUB 2390 [1256]
620 WINDOW SWAP 6,0 [1068]
630 CAT [96]
640 WINDOW SWAP 0,6 [1038]
650 READ x,y,xx,yy:GOSUB 2390 [1256]
660 ' [117]
670 'Window-Daten [780]
680 ' [117]
690 DATA 1,1,78,2 [561]
700 DATA 1,4,78,17 [683]
710 DATA 1,22,78,2 [475]
720 ' [117]
730 'Eingabe [925]
740 ' [117]
750 WINDOW SWAP 6,0 [1068]
760 CLS [91]
770 LOCATE 14,1:INPUT "Geben Sie den Namen [2733]
des Bildes ein : ",name$
780 IF (LEN(name$)>12 OR LEN(name$)<1) THE [4509]
N GOSUB 2590:GOTO 760
790 CLS [91]
800 LOCATE 11,1 [470]
810 INPUT "Setzen Sie den Bildschirmmodus [5202]
.< > <1> o. auch <2> : ",modus
820 IF (modus<0 OR modus>2) THEN GOSUB 259 [2725]
0:GOTO 790
830 ' [117]
840 'Screen laden [974]
850 ' [117]
860 MODE modus [903]
870 LOAD name$,&C000 [788]
880 :SVSCR,svscr [1479]
890 GOSUB 2510 [853]
900 ' [117]
910 'Hauptmenue [836]
920 ' [117]
930 MODE 2 [513]
940 INK 1,26:BORDER 0:INK 0,0:PEN 1:PAPER [2546]
0
950 RESTORE 690 [729]
960 FOR i=1 TO 3 [435]
970 READ x,y,xx,yy [978]
980 GOSUB 2390 [909]
990 IF i<>2 THEN LOCATE #6,33,1:PRINT #6," [2651]
* Haupt-menue *"
1000 NEXT [350]
1010 WINDOW #1,23,79,5,20:WINDOW SWAP 1,0 [1386]
1020 ' [117]
1030 'Wahlmoeglichkeiten [1608]
1040 ' [117]
1050 LOCATE 1,4 [620]
1060 PRINT "1 Screenmode aendern..... [2087]
.....
1070 PRINT "2 Farben des Screens animi [2901]
eren...
1080 PRINT "3 Farben des Screens abspe [3705]
ichern.

```

Listing Farbeinsteller

```

1090 PRINT "4 Farben des Screens aende [2084]
rn.....
1100 PRINT "5 Screen invertieren..... [3915]
.....
1110 PRINT "6 Zurueck zum Screen..... [3163]
.....
1120 PRINT "7 Neuen Screen laden..... [2625]
.....
1130 PRINT "8 Programm verlassen..... [3335]
.....
1140 PRINT "9 Screen kopieren..... [4119]
.....
1150 ' [117]
1160 'Menuepunkt anwaehlen [2566]
1170 ' [117]
1180 y=4 [367]
1190 a$="" [388]
1200 GOTO 1250 [361]
1210 a$=INKEY$:IF a$="" THEN 1210 [1431]
1220 LOCATE 3,y:PRINT " " [1386]
1230 IF a$=CHR$(240) THEN y=y-1:y=(1-ABS(y [1632]
<4)*)+(ABS(y<4)*4)
1240 IF a$=CHR$(241) THEN y=y+1:y=(y<13)+1 [2398]
+(1-(y<13)+1)*y)+ABS((y<13)*3)
1250 LOCATE 3,y:PRINT "=>":SOUND 129,20,1 [2225]
00,15,1
1260 IF a$=CHR$(224) THEN WINDOW SWAP 0,1: [4140]
ON y-3 GOTO 1680,1800,2280,1300,1890,1980,
1760,2070,2130
1270 GOTO 1210 [305]
1280 ' [117]
1290 'Veraendern der Farben [1549]
1300 ' [117]
1310 MODE modus:LDSCR,svscr [1241]
1320 PEN 1:x=1:y=1:xx=18:yy=3:GOSUB 2390 [1516]
1330 ' [117]
1340 'Farbleiste darstellen [1176]
1350 ' [117]
1360 BORDER b:FOR i=0 TO 15 [1337]
1370 INK 1,c(i):PEN 1:LOCATE i+2,2:PRINT C [2623]
HR$(143);
1380 NEXT [350]
1390 PRINT CHR$(227)+CHR$(134) [1095]
1400 IF UPPER(wahl$)="J" THEN 1840 [1594]
1410 PRINT CHR$(22)+CHR$(0) [1676]
1420 x=1:GOTO 1600 [847]
1430 ' [117]
1440 'Farben einstellen [1244]
1450 ' [117]
1460 WHILE INKEY$="" :WEND [1607]
1470 CALL &BD19 [352]
1480 LOCATE x+1,3:PRINT " " [1380]
1490 IF INKEY(8)=0 THEN x=x-1:x=ABS(x=0)+x [2488]
:GOTO 1630
1500 IF INKEY(1)=0 THEN x=x+1:x=ABS(x=19)+ [2488]
(1+(x=19))*x:GOTO 1630
1510 IF INKEY(9)<>0 THEN 1530 [537]
1520 IF x=17 THEN 900 ELSE IF x=18 THEN :S [2796]
CRINV
1530 IF INKEY(0)<>0 OR x>16 THEN 1560 [1288]
1540 IF INKEY(9)=0 THEN b=b+1:b=b*(b<27): [2796]
GOTO 1590
1550 IF c(x-1)<26 THEN c(x-1)=c(x-1)+1:INK [4825]
x-1,c(x-1) ELSE IF x=1 THEN INK 0,c(x-1)
1560 IF INKEY(2)<>0 OR x>16 THEN 1600 [1129]
1570 IF INKEY(9)=0 THEN b=b-1:b=b*(b>0):G [2262]
OTO 1590
1580 IF c(x-1)>0 THEN c(x-1)=c(x-1)-1:INK [4351]
x-1,c(x-1) ELSE IF x=1 THEN INK 0,c(x-1)
1590 BORDER b [870]
1600 nn=0:n=0:FOR i=1 TO 15:IF n<c(i) THEN [1703]
n=c(i):nn=i
1610 NEXT:PEN nn [955]
1620 LOCATE 18,2:PRINT CHR$(227)+CHR$(134) [1436]
1630 LOCATE x+1,3:PRINT CHR$(244) [1781]
1640 GOTO 1460 [391]
1650 ' [117]
1660 'Screen-Mode eingeben [3128]
1670 ' [117]
1680 RESTORE 2670 [848]
1690 GOSUB 3280 [1192]
1700 INPUT #6,"",modus:IF (modus<0 OR modu [5485]
s>2) THEN 1680
1710 GOSUB 2510 [853]
1720 GOTO 900 [320]
1730 ' [117]
1740 'Programm starten [2110]
1750 ' [117]
1760 RUN [243]
1770 ' [117]
1780 'Animieren der Farben [1132]
1790 ' [117]
1800 RESTORE 2820 [866]
1810 GOSUB 3280 [1192]

```

Listing Farbeinsteller

```

T: RETURN
1820 INPUT #6, "", wahl$ [1326]
1830 IF UPPER$(wahl$) <> "J" THEN MODE modus [4312]
: LDSCR, svscr ELSE 1300
1840 ii=0: GOSUB 3370 [525]
1850 FOR i=1 TO 15: INK i, c((1+ii) MOD 14): [2271]
NEXT: ii=ii+1
1860 WHILE INKEY$="" : GOTO 1850: WEND [2323]
1870 wahl$="" [767]
1880 GOSUB 3370: GOSUB 2510: GOTO 900 [1971]
1890 ' [117]
1900 ' Farben invertieren [1412]
1910 ' [117]
1920 RESTORE 3120 [743]
1930 GOSUB 3280 [1192]
1940 GOSUB 2510 [853]
1950 MODE modus [903]
1960 GOSUB 3370 [967]
1970 : LDSCR, svscr: : SCRINV: CALL &BB06: GOSUB [3401]
2510: GOTO 900
1980 ' [117]
1990 ' Zurueck zum Screen [1556]
2000 ' [117]
2010 RESTORE 3040 [743]
2020 GOSUB 3280 [1192]
2030 GOSUB 2510 [853]
2040 MODE modus [903]
2050 GOSUB 3370 [967]
2060 : LDSCR, svscr: CALL &BB06: GOSUB 2510: GO [2798]
TO 900
2070 ' [117]
2080 ' Programmende [1226]
2090 ' [117]
2100 x=30: y=14: xx=19: yy=3: GOSUB 2390 [2236]
2110 PRINT #6, "Ende des Programmes <TASTE> [3110]
druecken ."; : CALL &BB06: MODE 2
2120 END [110]
2130 ' [117]
2140 ' Screen kopieren [1222]
2150 ' [117]
2160 RESTORE 3190 [737]
2170 GOSUB 3280 [1192]
2180 INPUT #6, "", name$ [1301]
2190 IF LEN(name$) > 12 THEN 2160 ELSE IF LE [2114]
N(name$) < 1 THEN 2260
2200 MODE modus [903]
2210 : LDSCR, svscr [1436]
2220 GOSUB 2510 [853]
2230 GOSUB 3370 [967]
2240 : LDSCR, svscr [1436]
2250 SAVE name$, b, &C000, &4000 [2204]
2260 GOSUB 2510 [853]
2270 GOTO 900 [320]
2280 ' [117]
2290 ' Farben abspeichern [1484]
2300 ' [117]
2310 RESTORE 2930 [820]
2320 GOSUB 3280 [1192]
2330 INPUT #6, "", name$ [1301]
2340 IF LEN(name$) > 12 THEN 2310 ELSE IF LE [3446]
N(name$) < 1 THEN 2370
2350 GOSUB 3370 [967]
2360 SAVE name$, b, &B1D9, 38 [2000]
2370 GOSUB 2510 [853]
2380 GOTO 900 [320]
2390 ' [117]
2400 ' Sub : Rahmen [355]
2410 ' [117]
2420 WINDOW #6, x+1, xx+x, y+1, yy+y-1: CLS #6 [1957]
2430 INK 1, 26 [56]
2440 PEN 1: PEN #6, 1 [1321]
2450 LOCATE x, y: PRINT CHR$(150)+STRING$(xx [1873]
, 154)+CHR$(156)
2460 FOR iy=y+1 TO y+yy-1: LOCATE x, iy: PRIN [1891]
T CHR$(149)
2470 LOCATE x+xx+1, iy: PRINT CHR$(149): NEXT [2655]
2480 LOCATE x, y+yy: PRINT CHR$(147)+STRING$ [2781]
(xx, 154)+CHR$(153)
2490 SOUND 129, 100, 100, 15, 1, 1 [2078]
2500 RETURN [555]
2510 ' [117]
2520 ' Sub : Taste druecken [981]
2530 ' [117]
2540 y=10: xx=18: yy=2 [1292]
2550 IF PEEK(&B1C8)=0 THEN x=1 ELSE x=10 [2382]
2560 GOSUB 2390 [909]
2570 PRINT #6, "<TASTE> druecken ."; [2800]
2580 CALL &BB06: GOSUB 3370: RETURN [2150]
2590 ' [117]
2600 ' Sub : Fehler [160]
2610 ' [117]
2620 CLS [91]
2630 LOCATE 31, 1 [680]
2640 PRINT "Eingabe-Fehler !" [1488]
2650 FOR i=1 TO 2000: NEXT [998]

```

Listing Farbeneinsteller

```

2660 RETURN [555]
2670 ' [117]
2680 ' Text [127]
2690 ' [117]
2700 DATA 30, 10, 10 [309]
2710 DATA " Menuepunkt Screenmode aendern [3002]
"
2720 DATA "-----" [1549]
"
2730 DATA " Dieser Menuepunkt dient dazu, [3618]
"
2740 DATA " den eingestellten Bildschirm- [2919]
"
2750 DATA " modus zu aendern. Erlaubt ist [3178]
"
2760 DATA " Modus 0 (16 Farben), Modus 1 [2196]
"
2770 DATA " (4 Farben) und Modus 2 [2395]
"
2780 DATA " (2 Farben). Bitte waehlen Sie [1858]
"
2790 DATA " Modus 0, 1 oder Modus 2 : " [485]
2800 DATA * [109]
2810 ' [117]
2820 DATA 8, 6, 9 [182]
2830 DATA " Menuepunkt Farben animieren " [2519]
2840 DATA "-----" [1399]
2850 DATA " Hier werden interessante " [2869]
2860 DATA " Farbeffekte erzeugt. Sie " [2293]
2870 DATA " koennen zusaetzlich noch " [2137]
2880 DATA " die Farbleiste einblenden . " [2759]
2890 DATA " Bitte waehlen Sie <J>a oder " [2008]
2900 DATA " <N>ein : " [1505]
2910 DATA * [109]
2920 ' [117]
2930 DATA 21, 14, 9 [541]
2940 DATA " Menuepunkt Farben abspeichern [1859]
"
2950 DATA "-----" [1549]
"
2960 DATA " Sie haben die Moeglichkeit, [2541]
"
2970 DATA " die eingestellte Farbkombi- [2051]
"
2980 DATA " nation auf Disk-/Kassette zu [2290]
"
2990 DATA " sichern. Geben Sie also bitte [3398]
"
3000 DATA " den Dateinamen [2374]
"
3010 DATA " an : " [365]
3020 DATA * [109]
3030 ' [117]
3040 DATA 2, 2, 6 [235]
3050 DATA " Menuepunkt Screen einblenden " [2971]
3060 DATA "-----" [1494]
3070 DATA " Auf Tastendruck koennen Sie " [1451]
3080 DATA " den geladenen Bildschirm be- " [3467]
3090 DATA " trachten ." [598]
3100 DATA * [109]
3110 ' [117]
3120 DATA 40, 11, 6 [261]
3130 DATA " Menuepunkt Screen invertieren [2469]
"
3140 DATA "-----" [1549]
"
3150 DATA " Auf Tastendruck koennen Sie [1345]
"
3160 DATA " den geladenen Bildschirm [3155]
"
3170 DATA " farbinvers betrachten . [3805]
"
3180 DATA * [109]
3190 ' [117]
3200 DATA 12, 9, 7 [514]
3210 DATA " Menuepunkt Screen kopieren " [1340]
3220 DATA "-----" [1383]
3230 DATA " Zum Kopieren eines " [1749]
3240 DATA " Bildschirms geben Sie " [1886]
3250 DATA " bitte den Dateinamen " [1545]
3260 DATA " an : " [365]
3270 DATA * [109]
3280 ' [117]
3290 ' Sub : Textboxen [1176]
3300 ' [117]
3310 READ x, y, yy, a$ [1166]
3320 xx=LEN(a$): GOSUB 2390 [1696]
3330 WHILE a$ <> "*" [1402]
3340 PRINT #6, a$; : READ a$ [1043]
3350 WEND [390]
3360 RETURN [555]
3370 ' [117]
3380 ' Sub : Eingestellte Farben [991]
3390 ' [117]
3400 BORDER b: FOR i=0 TO 15: INK i, c(i): NEX [2841]

```

Listing Farbeneinsteller

Ordnung ist die halbe Musik!

CD-Liste für den CPC

Da hat man sich nun den neuen CD-Player angeschafft und zudem schon eine ansehnliche Sammlung dieser kleinen silbernen Scheiben, man tauscht auch schon gerne mal mit anderen CD-Besitzern, und auf der anderen Seite hat man seinen Computer, der mangels neuer Aufgaben statt der nicht vorhandenen Daumen den Cursor dreht. Was liegt also näher, als dem zur Zeit arbeitslosen Gesellen die Verwaltung der CD's zu übertragen?

Dieses Programm, das diese Aufgabe übernimmt, ist voll menuegesteuert und erklärt sich deshalb fast von selbst. Der Autor war bemüht, das Programm so gut wie möglich zu strukturieren (viel GOSUB, wenig GOTO).

Das Programm benutzt außerdem die Windowtechnik: Im oberen Fenster wird das Datum und der Stand der Liste angezeigt. Das Mittlere ist das Hauptfenster zur Ein- und Ausgabe. Im unteren Fenster erscheinen links der jeweilige Programmpunkt, rechts davon Eingabehilfen.

'CD-Liste' kommt mit einem Array aus: CD\$(MA,B). MA steht in Zeile 80 und bestimmt die Anzahl der maximal zu verwaltenden CD's. Die Variable ist von vornherein auf 100 gesetzt, sollte dies nicht ausreichen, so braucht man sie nur in Zeile 80 zu verändern.

Der deutsche Zeichensatz ist integriert. Die Abfragen sind, wo möglich, als 'INKEY\$'-Abfragen geschrieben, um größtmöglichen Komfort bei der Eingabe zu erreichen.

Ohnehin wurde großer Wert auf eine komfortable Ein- und Ausgabe gelegt. Eingaben in falscher Länge sind unmöglich; es erfolgt ein Rücksprung zu der falsch beantworteten Frage, ohne die Bildschirmmaske zu zerstören.

Die Ausgabe ist in allen nur erdenklichen Variationen möglich. Vor dem Ausdrucken können Sie auswählen, ob Sie NLQ oder Entwurfsdruck (bei jedem Punkt separat, um Teile hervorzuheben) haben möchten.

Im übrigen wurde der Artikel aus Heft 6/87, Seite 48, über effektivere BASIC-Programmierung, beherzigt, um das Programm schneller im Ablauf zu machen. Sie können also auch, nach erfolgter Prüfung des korrekten Ablaufs, alle REM-Zeilen weglassen, sie werden nicht angesprungen.

Das Programm wurde auf dem CPC 464 mit Hilfe einer DDI-1, einer Vortex FI-X und eines NLQ 401 Druckers geschrieben. (G. Lange / JB)

für 464-664-6128



```

10 REM ***** [722]
20 REM *** CD *** [713]
30 REM ***** [722]
40 ' [117]
50 REM *** INIT *** [487]
60 ' [117]
70 MODE 2 [513]
80 ma=100: maxanz=0 [520]
90 DIM cd$(ma,8) [374]
100 GOSUB 3140:GOSUB 3640 [2075]
110 PRINT#4:PRINT#4,"Heutiges Datum eingeb [3669]
en!"
120 INPUT#4, datum$ [967]
130 IF LEN(datum$)<>10 THEN CLS#4:GOTO 110 [3223]
140 ' [117]
150 REM *** BILDSCHIRM *** [1181]
160 ' [117]
170 WINDOW #1,1,40,1,3 [1373]
180 WINDOW #2,41,80,1,3 [1310]
190 WINDOW #4,1,80,4,22 [1279]
200 WINDOW #6,1,80,24,25 [962]
210 FOR a=0 TO 6 [726]
220 CLS #a [485]
230 NEXT a [383]
240 PRINT#1,"CD-LISTE" [2116]
250 PRINT#1,CHR$(164)+" G. Lange" [1994]
260 LOCATE#2,10,1:PRINT#2,"Datum: "datum$ [2847]
270 FOR a=1 TO 80 [713]
280 LOCATE a,3:PRINT CHR$(154) [1579]
290 NEXT a [383]

```

Listing: Ordnung ist die halbe Musik

```

300 FOR a=1 TO 2 [770]
310 LOCATE 39,a:PRINT CHR$(149) [744]
320 NEXT a [383]
330 LOCATE 39,3:PRINT CHR$(155) [2019]
340 FOR a=1 TO 80 [713]
350 LOCATE a,24:PRINT CHR$(154) [1078]
360 NEXT a [383]
370 FOR a=24 TO 25 [586]
380 LOCATE 39,a:PRINT CHR$(149) [744]
390 NEXT a [383]
400 LOCATE 39,24:PRINT CHR$(158) [593]
410 ' [117]
420 REM *** AUSWAHLMENUE *** [1527]
430 ' [117]
440 CLS#4 [386]
450 PRINT#4:PRINT#4,"MENUE:" [1464]
460 PRINT#4:PRINT#4:LOCATE#6,42,2:PRINT#6, [5112]
SPC(36):LOCATE#6,1,2:PRINT#6,SPC(36)
470 PRINT#4,"1. -> Eingabe" [2330]
480 PRINT#4,"2. -> Ausgabe" [2140]
490 PRINT#4,"3. -> [ndern" [941]
500 PRINT#4,"4. -> Speichern" [1121]
510 PRINT#4,"5. -> Laden" [1497]
520 PRINT#4,"6. -> Neues Datum/Neuer Stand [3003]
"
530 PRINT#4,"7. -> Lischen" [1676]
540 PRINT#4,"8. -> Programm beenden" [1954]
550 a$=INKEY$:IF a$="" THEN 550 [1370]
560 a=VAL(a$):IF a<1 THEN IF a>8 THEN 550 [1769]
570 ON a GOSUB 620,980,2380,2620,2820,3030 [2434]
,3800,4060
580 GOTO 440 [444]
590 ' [117]
600 REM *** EINGABE *** [757]
610 ' [117]
620 CLS#4:LOCATE #6,1,2:PRINT#6,"1. Eingab [3230]
e
630 IF maxanz>1 THEN GOTO 650 [1080]
640 PRINT#4:PRINT#4,"Stand der Eingaben: " [7692]
;:INPUT#4,stands$:IF LEN(stands$)<>10 THEN G
OTO 620 ELSE LOCATE#2,10,2:PRINT#2,"Stand:
"stands$:CLS#4
650 anf=maxanz+1 [66]
660 FOR a=anf TO ma [1793]
670 CLS#4 [386]
680 IF a+1=ma THEN PRINT#4:PRINT#4,"ACHTUN [13478]
G! Speicherplatz wird zu gering!":PRINT#4:
PRINT#4,"Daten abspeichern und ma (Zeile 8
0) erh:hen!":GOSUB 3300:RETURN
690 PRINT#4:PRINT#4,USING"###";a;:PRINT#4, [2823]
". CD: "
700 LOCATE#6,42,2:PRINT#6,"Beenden mit >En [2902]

```

Listing: Ordnung ist die halbe Musik

Anwendungen

```

de<
710 LOCATE#4,1,4:PRINT#4,cd$(0,1);SPC(10-L [3835]
EN(cd$(0,1));:INPUT#4,cd$(a,1)
720 IF LEN(cd$(a,1))>68 THEN cd$(a,1)=LEFT [2666]
$(cd$(a,1),68)
730 IF UPPER$(cd$(a,1))="ENDE" THEN maxanz [3573]
=a-1:RETURN
740 LOCATE#6,42,2:PRINT#6," [1609]
"
750 LOCATE#4,1,6:PRINT#4,cd$(0,2);SPC(10-L [4089]
EN(cd$(0,2));:INPUT#4,cd$(a,2)
760 IF LEN(cd$(a,2))>68 THEN cd$(a,2)=LEFT [3680]
$(cd$(a,2),68)
770 LOCATE#4,1,8:PRINT#4,cd$(0,3);SPC(10-L [7054]
EN(cd$(0,3));:LOCATE#6,42,2:PRINT#6,"Ein
eben: TT.MM.JJJJ"SPC(14):INPUT#4,cd$(a,3)
780 IF LEN(cd$(a,3))>10 THEN LOCATE#4,11, [3409]
8:PRINT#4,SPC(25):GOTO 770
790 LOCATE#4,1,10:PRINT#4,cd$(0,4);SPC(10- [6660]
LEN(cd$(0,4));:LOCATE#6,42,2:PRINT#6,"Ein
geben: DM.PF"SPC(18):INPUT#4,cd$(a,4)
800 IF LEN(cd$(a,4))<5 THEN IF LEN(cd$(a,4 [2824]
))>6 THEN LOCATE #4,11,10:PRINT SPC(25):GO
TO 790
810 LOCATE#4,1,12:PRINT#4,cd$(0,5);SPC(10- [6685]
LEN(cd$(0,5));:LOCATE#6,42,2:PRINT#6,"Ein
geben: JJJJ"SPC(19):INPUT#4,cd$(a,5)
820 IF LEN(cd$(a,5))>4 THEN LOCATE#4,11,1 [3313]
2:PRINT#4,SPC(25):GOTO 810
830 LOCATE#4,1,14:PRINT#4,cd$(0,6);SPC(10- [6227]
LEN(cd$(0,6));:LOCATE#6,42,2:PRINT#6,"Ein
geben: XX:xx"SPC(18):INPUT#4,cd$(a,6)
840 IF LEN(cd$(a,6))>5 THEN LOCATE#4,11,1 [3836]
4:PRINT#4,SPC(25):GOTO 830
850 LOCATE#6,42,2:PRINT#6,"1 = DDD ; 2 = A [1895]
DD ; 3 = AAD"
860 LOCATE#4,1,16:PRINT#4,cd$(0,7);SPC(10- [2389]
LEN(cd$(0,7));:"?";
870 AS=INKEYS:IF AS="" THEN 870 [1266]
880 c=VAL(AS):IF c<1 OR c>3 THEN 860 [2714]
890 IF c=1 THEN cd$(a,7)="DDD" ELSE IF c=2 [4027]
THEN cd$(a,7)="ADD" ELSE cd$(a,7)="AAD"
900 LOCATE#4,13,16:PRINT#4,cd$(a,7) [1167]
910 LOCATE#4,1,18:PRINT#4,cd$(0,8);SPC(10- [6298]
LEN(cd$(0,8));:LOCATE#6,42,2:PRINT#6,"Ein
geben: X oder XX"SPC(12):INPUT#4,cd$(a,8)
920 IF LEN(cd$(a,8))=2 THEN GOTO 930 ELSE [8332]
IF LEN(cd$(a,8))=1 THEN cd$(a,8)="0"+cd$(a
,8) ELSE LOCATE#4,11,18:PRINT#4,SPC(25):GO
TO 910
930 NEXT a [383]
940 RETURN [555]
950 ' [117]
960 REM *** AUSGABEMENUE *** [1587]
970 ' [117]
980 CLS#4 [386]
990 PRINT#4:PRINT#4,"UNTERMENUE AUSGABE:" [2647]
1000 PRINT#4:PRINT#4:LOCATE#6,42,2:PRINT#6 [5112]
,SPC(36):LOCATE#6,1,2:PRINT#6,SPC(36)
1010 PRINT#4,"1. -> Anzeige aller Felder" [2927]
1020 PRINT#4,"2. -> Anzeige bestimmter Fel [3809]
der"
1030 PRINT#4,"3. -> Anzeige alle Felder in [4632]
einer Zeile"
1040 PRINT#4,"4. -> Anzeige Statistik" [2299]
1050 PRINT#4,"5. -> Druck aller Felder" [2412]
1060 PRINT#4,"6. -> Druck bestimmter Felde [3894]
r"
1070 PRINT#4,"7. -> Druck alle Felder in e [2872]
iner Zeile"
1080 PRINT#4,"8. -> Druck Statistik" [1331]
1090 PRINT#4,"9. -> Hauptmenue" [1794]
1100 ES=INKEYS:IF ES="" THEN 1100 [938]
1110 e=VAL(ES):IF e<1 THEN GOTO 980 [2049]
1120 IF e=9 THEN RETURN [1238]
1130 IF e>0 THEN IF e<5 THEN flagd=0 ELSE [2700]
flagd=1
1140 ON e GOSUB 1190,1380,1770,1920,2160,2 [2484]
210,2260,2310
1150 GOTO 980 [496]
1160 ' [117]
1170 REM *** ANZEIGE ALLER FELDER *** [2125]
1180 ' [117]
1190 CLS#4:G=4:LOCATE#6,1,2:PRINT#6,"Anzei [5023]
ge aller Felder":GOSUB 3230
1200 PRINT#G," LISTE ALLER [5545]

```

Listing: Ordnung ist die halbe Musik

```

CD:"SPC(15);:IF flagd THEN PRINT#G,"Stand
: "stands
1205 PRINT#G:PRINT#G [609]
1210 IF flagf THEN GOTO 1330 [2269]
1220 FOR a=1 TO maxanz [1622]
1230 IF 71-LEN(cd$(a,1))<LEN(cd$(a,2)) THE [1225]
N GOTO 1250
1240 PRINT#G,USING"###";a;:PRINT#G,". "cd$ [3952]
(a,1)" - "cd$(a,2):GOTO 1270
1250 PRINT#G,USING"###";a;:PRINT#G,". "cd$ [3014]
(a,1)" - "
1260 PRINT#G," "cd$(a,2) [1423]
1270 PRINT#G," ";:FOR b=3 TO 8 [1217]
1280 PRINT#G,cd$(a,b);" / "; [2021]
1290 NEXT b [378]
1300 IF flagd=0 THEN IF a/6=INT(a/6) THEN [3852]
GOSUB 3300
1310 PRINT#G:NEXT a [661]
1320 GOSUB 3300:RETURN [1648]
1330 PRINT#G:FOR a=1 TO maxanz:FOR b=1 TO [4455]
8:PRINT#G,cd$(0,b)SPC(10-LEN(cd$(0,b)))":
"cd$(a,b):NEXT b:PRINT#G:GOSUB 3300:NEXT a
1340 GOSUB 3300:RETURN [1648]
1350 ' [117]
1360 REM *** ANZEIGE BESTIMMTER FELDER *** [2446]
1370 ' [117]
1380 CLS#4:G=4:LOCATE#6,1,2:PRINT#6,"Anzei [5964]
ge bestimmter Felder"
1390 PRINT#4:PRINT#4,"2. UNTERMENUE:":PRIN [2446]
T#4
1400 PRINT#4,"1. -> Ein Feld" [2060]
1410 PRINT#4,"2. -> Mehrere Felder" [2024]
1420 PRINT#4,"3. -> Untermenue Ausgabe" [2382]
1430 W$=INKEYS:IF W$="" THEN 1430 [1856]
1440 w=VAL(W$):IF w<1 OR w>3 THEN GOTO 138 [2256]
0
1450 IF w=3 THEN RETURN [797]
1460 ON w GOSUB 1480,1570 [857]
1470 GOTO 1380 [423]
1480 CLS#4:PRINT#4:FOR b=1 TO 8:PRINT#4,b" [4273]
: "cd$(0,b):NEXT b:PRINT#4:PRINT#4
1490 IF flagd THEN PRINT#4,"Welches Feld m [4266]
:chten Sie drucken?":GOTO 1510
1500 PRINT#4,"Welches Feld m:chten Sie anz [5264]
eigen?"
1510 W$=INKEYS:IF W$="" THEN 1510 [1864]
1520 w=VAL(W$):IF w<1 OR w>8 THEN GOTO 148 [2323]
0
1530 CLS#4:IF flagd THEN PRINT#G,"Druck al [5865]
ler CD - Teilbereich "cd$(0,w)"spc(10)"S
tand: "stand$:goto 1390
1540 CLS#4:PRINT#G,"Anzeige aller CD - T [3907]
eilbereich "cd$(0,w)": "
1550 PRINT#G:FOR a=1 TO maxanz:PRINT#G,USI [3426]
NG"###";a;:PRINT#G,". "cd$(a,w):NEXT a
1560 GOSUB 3300:RETURN [1648]
1570 CLS#4:PRINT#4:FOR b=1 TO 8:PRINT#4,b" [4273]
: "cd$(0,b):NEXT b:PRINT#4:PRINT#4
1580 IF flagd THEN PRINT#4,"Wieviele Felde [6454]
r m:chten Sie drucken?":GOTO 1600
1590 PRINT#4,"Wieviele Felder m:chten Sie [3751]
anzeigen "
1600 ES=INKEYS:IF ES="" THEN 1600 [918]
1610 wahl=VAL(ES):IF wahl<1 OR wahl>8 THEN [2799]
GOTO 1570
1620 IF flagd THEN PRINT#4:PRINT#4,"Welche [6434]
Felder m:chten Sie drucken?":GOTO 1640
1630 PRINT#4:PRINT#4,"Welche Felder m:chte [4899]
n Sie anzeigen?"
1640 FOR d=1 TO wahl:INPUT#4," ";lis(d) [2869]
:NEXT d
1650 FOR a=1 TO 300:NEXT:CLS#4 [2208]
1660 IF flagd THEN PRINT#G,"CD-Liste -- [13225]
- Druck mehrerer Felder:"SPC(10)"Stand:
"stands:PRINT#G ELSE PRINT#G,"CD-Liste
--- Anzeige mehrerer Felder:":PRINT#G
1670 FOR d=1 TO wahl:PRINT#G,cd$(0,lis(d)) [4670]
;" / ";:NEXT d:PRINT#G:PRINT#G
1680 FOR a=1 TO maxanz [1622]
1690 PRINT#G,USING"###";a;:PRINT#G,". CD: [2191]
";
1700 FOR d=1 TO wahl [1534]
1710 PRINT#G,cd$(a,lis(d));" / "; [2382]
1720 NEXT d:PRINT#G:NEXT a [1536]
1730 GOSUB 3300:RETURN [1648]
1740 ' [117]

```

Listing: Ordnung ist die halbe Musik

```

1750 REM *** ANZEBIGE IN EINER ZEILE *** [3090]
1760 ' [117]
1770 G=4 [424]
1780 CLS#4:PRINT#G,"Anzeige aller Felder i [5189]
n einer Zeile:":PRINT#G:PRINT#G
1790 PRINT#G,"Interpret /Titel (7890]
/Datum /Preis/Jahr/Zei
t /Au./St.":PRINT#G
1800 FOR a=1 TO maxanz [1622]
1810 PRINT#G,LEFT$(cd$(a,22);SPC(22-LEN [5677]
(cd$(a,1));"/";LEFT$(cd$(a,2),21);SPC(21-
LEN(cd$(a,2)));"/";
1820 FOR b=3 TO 8 [1162]
1830 PRINT#G,cd$(a,b);"/"; [1607]
1840 NEXT b [378]
1850 IF flagd THEN PRINT#8 [1415]
1860 NEXT a [383]
1870 PRINT#G [552]
1880 GOSUB 3300:RETURN [1648]
1890 ' [117]
1900 REM *** ANZEIGE STATISTIK *** [2403]
1910 ' [117]
1920 G=4 [424]
1930 CLS#4:PRINT#G,"Anzeige Statistik: ":P [5374]
RINT#G,"Moment bitte! RECHNER arbeit
et!"
1940 LOCATE#6,1,2:PRINT#6,"Statistik":GOSU [1985]
B 3340
1950 CLS#4:PRINT#G,"Anzeige Statistik: ":P [2717]
RINT#G
1960 PRINT#G,"Die billigste CD ";minp [5097]
rcd;". CD: ";cd$(minprcd,4);" DM"
1970 PRINT#G,"Die teuerste CD ";maxp [3555]
rcd;". CD: ";cd$(maxprcd,4);" DM"
1980 PRINT#G,"Die geringste Zeit ";minz [4730]
ecd;". CD: ";cd$(minzecd,6);" Minuten"
1990 PRINT#G,"Die l"ngste Zeit ";maxz [3954]
ecd;". CD: ";cd$(maxzecd,6);" Minuten"
2000 PRINT#G,"Die meisten St"cke ";maxs [5143]
tcd;". CD: ";cd$(maxstcd,8);" St"cke"
2010 PRINT#G,"Die wenigsten St"cke ";mins [5197]
tcd;". CD: ";cd$(minstcd,8);" St"cke"
2020 PRINT#G [552]
2030 PRINT#G,"Summe aller St"cke ";:PRI [4993]
NT#G,USING"#####";sumst;:PRINT#G," St)
cke";
2040 PRINT#G," Durchschnitt: ";:PRINT [4431]
#G,USING"#####";sumst/maxanz;:PRINT#G,"
St./CD"
2050 PRINT#G,"Summe aller Preise ";:PRI [4534]
NT#G,USING"#####.##";sumpr;:PRINT#G," DM"
;
2060 PRINT#G," Durchschnitt: ";:P [6005]
RINT#G,USING"#####.##";sumpr/maxanz;:PRINT#
G," DM /CD"
2070 PRINT#G,"Summe aller Zeiten ";:PRI [8511]
NT#G,USING"#####";sumh;:PRINT#G,".";USING"##
";summi;:PRINT#G,".";USING"###";sumse;:PRIN
T#G," h.min:sec";
2080 PRINT#G," Durchschnitt: ";:PRINT#G, [9139]
USING"#####";sumschmi;:PRINT#G,".";USING"##
";sumschse;:PRINT#G," MIN/CD"
2090 PRINT#G:PRINT#G,"Aufnahme in DDD-Qual [2280]
it(t:"auf(1)
2100 PRINT#G,"Aufnahme in ADD-Qualit(t:"a [2137]
uf(2)
2110 PRINT#G,"Aufnahme in AAD-Qualit(t:"a [2334]
uf(3)
2120 GOSUB 3300:RETURN [1648]
2130 ' [117]
2140 REM *** DRUCK ALLER FELDER *** [743]
2150 ' [117]
2160 GOSUB 3560:CLS#4:PRINT#4,"Druck aller [3917]
Felder: ":GOSUB 3230
2170 G=8:GOSUB 1200:RETURN [2372]
2180 ' [117]
2190 REM *** DRUCK BESTIMMTER FELDER *** [1942]
2200 ' [117]
2210 GOSUB 3560:CLS#4:G=8:LOCATE#6,1,2:PRI [5133]
NT#6,"Druck bestimmter Felder"
2220 GOSUB 1390:RETURN [2098]
2230 ' [117]
2240 REM *** DRUCK IN EINER ZEILE *** [1609]
2250 ' [117]
2260 G=8:GOSUB 3560:CLS#4:PRINT#G,"Druck a [9333]
ller Felder in einer Zeile:"SPC(10)"Stand:

```

Listing: Ordnung ist die halbe Musik

```

"stand$:PRINT#G:PRINT#G
2270 GOSUB 1790:RETURN [1581]
2280 ' [117]
2290 REM *** DRUCK STATISTIK *** [501]
2300 ' [117]
2310 GOSUB 3560:CLS#4:PRINT#4,"Druck Stati [3677]
stik"
2320 LOCATE#6,1,2:PRINT#6,"Statistik":GOSU [1985]
B 3340
2330 G=8:CLS#4:PRINT#G,"Druck Statistik: " [4866]
SPC(15)"Stand: "stand$:PRINT#G
2340 GOSUB 1960:PRINT#G:RETURN [2523]
2350 ' [117]
2360 REM *** I'NDERN *** [828]
2370 ' [117]
2380 CLS#4:LOCATE#6,1,2:PRINT#6,"3. I'ndern [2929]
":FOR a=1 TO maxanz
2390 PRINT#4,USING"###";a;:PRINT#4,". : "L [3851]
EFT$(cd$(a,1),35)" - "LEFT$(cd$(a,2),35)
2400 NEXT a [383]
2410 PRINT#4:PRINT#4:PRINT#4,"Bei welcher [5215]
CD soll ge'ndert werden";:INPUT#4,wah
2420 IF wah>maxanz THEN GOTO 2410 [2438]
2430 CLS#4:FOR b=1 TO 8 [1451]
2440 PRINT#4,b;CHR$(8);". : "cd$(wah,b) [1135]
2450 NEXT b [378]
2460 PRINT#4:PRINT#4:PRINT#4,"Welches Feld [4873]
soll ge'ndert werden?"
2470 R$=INKEY$:IF R$="" THEN 2470 [2483]
2480 r=VAL(R$):IF r<1 OR r>8 THEN 2460 [644]
2490 CLS#4:PRINT#4,"I'nderung: " [2203]
2500 PRINT#4:PRINT#4,"Altes Feld: "; [2624]
2510 PRINT#4,CHR$(24)cd$(wah,r);CHR$(24) [1864]
2520 PRINT#4:PRINT#4,"Neues Feld: ";:PRIN [5654]
T#4,CHR$(24);:INPUT#4,cd$(wah,r):PRINT#4,C
HR$(24)
2530 FOR e=1 TO 300:NEXT:PRINT#4:PRINT#4:P [2221]
RINT#4
2540 PRINT#4,"I'nderung angenommen!" [2329]
2550 PRINT#4:PRINT#4,"Noch eine I'nderung J [3748]
/N"
2560 E$=UPPER$(INKEY$):IF E$="" THEN 2560 [2151]
2570 IF E$="J" THEN GOTO 2380 ELSE IF E$="" [1875]
N" THEN RETURN ELSE GOTO 2530
2580 RETURN [555]
2590 ' [117]
2600 REM *** SPEICHERN *** [773]
2610 ' [117]
2620 CLS#4:LOCATE#6,1,2:PRINT#6,"4. Speich [2677]
ern"
2630 PRINT#4,"Bestimmter Dateiname oder St [3441]
andard-Name B/S?"
2640 A$=UPPER$(INKEY$):IF A$="" THEN 2640 [1721]
2650 IF A$="S" THEN GOTO 2670 ELSE IF A$="" [4858]
B" THEN 2660 ELSE 2620
2660 PRINT#4:PRINT#4:INPUT#4,"Dateiname mi [5444]
t Zusatz ";dat$:GOTO 2680
2670 dat$="cdhaben.cdf" [1087]
2680 dat$=dat$ [421]
2690 OPENOUT dat$ [749]
2700 PRINT#9,stand$ [1335]
2710 PRINT#9,maxanz [1488]
2720 FOR a=1 TO maxanz [1622]
2730 FOR b=1 TO 8 [1114]
2740 PRINT#9,cd$(a,b) [1500]
2750 NEXT b [378]
2760 NEXT a [383]
2770 CLOSEOUT [902]
2780 RETURN [555]
2790 ' [117]
2800 REM *** LADEN *** [706]
2810 ' [117]
2820 CLS#4:LOCATE#6,1,2:PRINT#6,"5. Laden" [1820]
2830 PRINT#4,"Bestimmter Dateiname oder St [3441]
andard-Name B/S?"
2840 A$=UPPER$(INKEY$):IF A$="" THEN 2840 [1776]
2850 IF A$="S" THEN GOTO 2870 ELSE IF A$="" [2906]
B" THEN 2860 ELSE 2820
2860 PRINT#4:PRINT#4:INPUT#4,"Dateiname mi [5756]
t Zusatz ";dat$:GOTO 2880
2870 dat$="cdhaben.cdf" [1087]
2880 dat$=dat$ [421]
2890 OPENIN dat$ [883]
2900 INPUT#9,stand$ [805]
2910 INPUT#9,maxanz [1118]
2920 FOR a=1 TO maxanz [1622]

```

Listing: Ordnung ist die halbe Musik

```

2930 FOR b=1 TO 8 [1114]
2940 INPUT#9,cd$(a,b) [1508]
2950 NEXT b [378]
2960 NEXT a [383]
2970 CLOSEIN [752]
2980 LOCATE #2,10,2:PRINT#2,"Stand: "stand [2143]
$
2990 RETURN [555]
3000 ' [117]
3010 REM *** DATUMS(NDERUNG *** [1558]
3020 ' [117]
3030 CLS#4 [386]
3040 PRINT#4:PRINT#4:PRINT#4,"Altes Datum: [4347]
";CHR$(24);datum$;CHR$(24)
3050 PRINT#4:PRINT#4,"Neues Datum: ";CHR$( [6543]
24)::INPUT#4, datum$:PRINT#4,CHR$(24)
3060 PRINT#4:PRINT#4:PRINT#4,"Alter Stand: [4846]
";CHR$(24);stand$;CHR$(24)
3070 PRINT#4:PRINT#4,"Neuer Stand: ";CHR$( [5852]
24)::INPUT#4,stand$:PRINT#4,CHR$(24)
3080 LOCATE #2,10,1:PRINT#2,"Datum: "datum [2847]
$
3090 LOCATE #2,10,2:PRINT#2,"Stand: "stand [2143]
$
3100 CLS#4:RETURN [677]
3110 ' [117]
3120 REM *** INIT BEGRIFFE *** [1934]
3130 ' [117]
3140 RESTORE 3190 [737]
3150 FOR b=1 TO 8 [1114]
3160 READ cd$(0,b) [691]
3170 NEXT b [378]
3180 RETURN [555]
3190 DATA Interpret, Titel, Kaufdatum, Preis, [3151]
P-Jahr, Laufzeit, Aufnahme, St)cke
3200 ' [117]
3210 REM *** FELDERABFRAGE *** [854]
3220 ' [117]
3230 CLS#4:PRINT#4:PRINT#4,"Alle Felder hi [6787]
ntereinander oder untereinander H/U?"
3240 R$=UPPER$(INKEY$):IF R$="" THEN 3240 [2407]
3250 IF R$="H" THEN flagf=0 ELSE IF R$="U" [2877]
THEN flagf=1 ELSE GOTO 3230
3260 CLS#4:RETURN [677]
3270 ' [117]
3280 REM *** WEITERABFRAGE *** [1104]
3290 ' [117]
3300 LOCATE#6,71,2:PRINT#6,">SPACE<" [1059]
3310 A$=INKEY$:IF A$="" THEN 3310 [1435]
3320 IF A$=" " THEN RETURN ELSE 3310 [2269]
3330 REM *** STATISTIK *** [1572]
3340 sumpr=0:sumst=0:sumse=0:minpr=40:maxpr [3551]
r=0:minst=20:maxst=0:minze=3600:maxze=0:FO
R s=1 TO 3:auf(s)=0:NEXT
3350 FOR a=1 TO maxanz [1622]
3360 minu=0 [477]
3370 sumpr=sumpr+VAL(cd$(a,4)) [969]
3380 sumst=sumst+VAL(cd$(a,8)) [949]
3390 minu=60*VAL(LEFT$(cd$(a,6),2)) [1476]
3400 sumse=sumse+VAL(RIGHT$(cd$(a,6),2))+m [2130]
inu
3410 IF VAL(cd$(a,4))<minpr THEN minpr=VAL [2817]
(cd$(a,4)):minprd=a
3420 IF VAL(cd$(a,4))>maxpr THEN maxpr=VAL [3697]
(cd$(a,4)):maxprd=a
3430 IF VAL(cd$(a,8))>maxst THEN maxst=VAL [3825]
(cd$(a,8)):maxstcd=a
3440 IF VAL(cd$(a,8))<minst THEN minst=VAL [3262]
(cd$(a,8)):minstcd=a
3450 IF VAL(RIGHT$(cd$(a,6),2))+minu<minze [6545]
THEN minze=VAL(RIGHT$(cd$(a,6),2))+minu:m
inze=cd=a
3460 IF VAL(RIGHT$(cd$(a,6),2))+minu>maxze [5238]
THEN maxze=VAL(RIGHT$(cd$(a,6),2))+minu:m
axze=cd=a
3470 IF cd$(a,7)="DDD" THEN auf(1)=auf(1)+ [4490]
1 ELSE IF cd$(a,7)="ADD" THEN auf(2)=auf(2
)+1 ELSE auf(3)=auf(3)+1
3480 NEXT a [383]
3490 sumsch=INT(sumse/maxanz) [1882]
3500 sumh=FIX(sumse/3600):summi=FIX(sumse/ [5390]
60)-sumh*60:sumse=sumse-sumh*3600-summi*60
3510 sumschmi=FIX(sumsch/60):sumschse=sums [3740]
ch-sumschmi*60
3520 RETURN [555]
3530 ' [117]

```

Listing: Ordnung ist die halbe Musik

```

3540 REM *** DRUCKERABFRAGE *** [899]
3550 ' [117]
3560 CLS#4 [386]
3570 PRINT#4:PRINT#4,"NLQ- oder Entwurfsdr [4072]
uck N/E?"
3580 T$=UPPER$(INKEY$):IF T$="" THEN 3580 [1865]
3590 IF T$="E" THEN PRINT#8,CHR$(27);"I";C [6293]
HR$(1); ELSE IF T$="N" THEN PRINT#8,CHR$(2
7);"I";CHR$(3); ELSE GOTO 3570
3600 RETURN [555]
3610 ' [117]
3620 REM *** DEUTSCHE ZEICHEN *** [727]
3630 ' [117]
3640 SYMBOL AFTER 91 [1213]
3650 SYMBOL 91,&66,&18,&3C,&66,&7B,&66,&66 [1766]
,&0
3660 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C [2126]
,0
3670 SYMBOL 93,&66,&44,&66,&66,&66,&66,&3C [2286]
,0
3680 SYMBOL 123,&CC,0,&78,&C,&7C,&CC,&76,0 [2418]
3690 SYMBOL 124,&66,0,&3C,&66,&66,&66,&3C, [2274]
0
3700 SYMBOL 125,&66,0,&66,&66,&66,&66,&3E, [1843]
0
3710 SYMBOL 126,&3C,&66,&66,&6C,&66,&66,&6 [2404]
C,&60
3720 KEY DEF 17,1,125,93 [746]
3730 KEY DEF 19,1,123,91 [1140]
3740 KEY DEF 26,1,124,92,128 [874]
3750 KEY DEF 24,1,126,94,0 [1480]
3760 RETURN [555]
3770 ' [117]
3780 REM *** L\SCHEN *** [956]
3790 ' [117]
3800 CLS#4:LOCATE#6,1,2:PRINT#6,"7. L'sche [4134]
n":FOR a=1 TO maxanz
3810 PRINT#4,USING"###";a;:PRINT#4,". : "L [3851]
EFT$(cd$(a,1),35)" - "LEFT$(cd$(a,2),35)
3820 NEXT a [383]
3830 PRINT#4:PRINT#4:PRINT#4,"Welche CD so [3998]
ll gel'scht werden ";INPUT#4,loe
3840 IF loe>maxanz THEN GOTO 3830 [1783]
3850 PRINT#4:PRINT#4,loe;CHR$(8);". CD l's [5122]
chen. Sind Sie sicher J/N?"
3860 A$=UPPER$(INKEY$):IF A$="" THEN 3860 [1780]
3870 IF A$="J" THEN 3880 ELSE RETURN [2185]
3880 CLS#4:PRINT#4,loe;CHR$(8);". CD l'sch [2900]
en!"
3890 FOR a=loe TO maxanz-1 [1660]
3900 FOR b=1 TO 8 [1114]
3910 cd$(a,b)=cd$(a+1,b) [1277]
3920 NEXT b [378]
3930 NEXT a [383]
3940 FOR b=1 TO 8 [1114]
3950 cd$(maxanz,b)="" [942]
3960 NEXT b [378]
3970 maxanz=maxanz-1 [947]
3980 PRINT#4:PRINT#4,"L'schung durchgef}hr [3541]
t!"
3990 PRINT#4:PRINT#4,"Noch mehr l'schen J/ [3402]
N?"
4000 A$=UPPER$(INKEY$):IF A$="" THEN 4000 [1896]
4010 IF A$="J" THEN 3800 ELSE RETURN [1998]
4020 RETURN [555]
4030 ' [117]
4040 REM *** PROGRAMMENDE *** [1366]
4050 ' [117]
4060 CLS#4:LOCATE#6,1,2:PRINT#6,"8. Progra [3492]
mmende"
4070 PRINT#4:PRINT#4:PRINT#4,"Programm sol [4453]
l beendet werden!"
4080 PRINT#4:PRINT#4:PRINT#4,"Sind Sie sic [5235]
her - Alle Daten abgespeichert J/N?"
4090 A$=UPPER$(INKEY$):IF A$="" THEN 4090 [1886]
4100 IF A$="J" THEN MODE 2:CLEAR:END ELSE [2011]
RETURN
4110 RETURN [555]

```



Listing: Ordnung ist die halbe Musik

CPC DATABOX

— Das ist die Software zum CPC Sonderheft 6/88 —

DATABOX:

- mehr als der übliche Softwareservice
- bringt alle Programme des Sonderheftes auf Kassette oder 3" Diskette
- Die Programme sind, durch den auf dem Datenträger enthaltenen CPC-Emulator, auf allen drei CPC-Rechnern lauffähig.
- trägt das Titelbild des gleichzeitig erscheinenden Sonderheftes.
- über 500 KByte Software für Ihren CPC!



Für alle CPC's als Kassette und 3"-Diskette

Inhalt der Databox zu Sonderheft 6/88

Diskette/Kassette 1:

Coding	Farbeneinsteller
Comex	Disc-Manager
Exodos	Fast Screen
Lineare Gleichungssysteme	Magische Quadrate
Menue-Generator	Plot it
Printfix	Ref-Programme
Pro-DATA	Animierte Textausgabe
Buchführung	CD-Liste
Großbuchstaben	OCF-Fonts
Schulnoten	Scroll
Sort	Schnelle Dreiecke
Overflow-Unterdrücker	

Diskette/Kassette 2

Subsunk	3D-Life
Wölfe	SHIFT
Rob the Roboter	Test Me
Komplexe Grafiken	Ballcracker
Faktu-Pro	Catlist
CP/M Verbess.	Grafik u. Co.
Hardcopies (Turbo Pascal)	Hardcopies (CP/M)
Directory unter Turbo Pascal	Deutsches Datum

Einzelbezugspreise für DATABOX:

je Diskette 3" 24,- DM zuzüglich 3,- DM Porto/Verpackung (im Ausland zuzüglich 5,- DM Porto/Verpackung).
Kombi-Pack (3"Diskette 1 + 2)

nur 38,-DM

je Kassette 14,- DM zuzüglich 3,- DM Porto/Verpackung (im Ausland zuzüglich 5,- DM Porto/Verpackung).
Kombi-Pack (Kassette 1 + 2)

nur 25,- DM

Coupon ausschneiden, auf Postkarte aufkleben oder per Brief senden an:

DMV-Verlag, Postfach 250, 3440 Eschwege



»CPC-Bestellservice«

Ich bestelle hiermit aus Ihrem Angebot:

Stck. _____

Copyshop
Das universelle Hardcopy-Programm für Schneider CPC 464/664/6128

- 201 Kassettenversion **59,- DM**
202 Diskette 3" **69,- DM**
203 Diskette 5 1/4" (Vortex) **69,- DM**

Stck. _____

Power-Spiele-Paket CPC

Dieses Angebot beinhaltet 4 Kassetten bzw. 4 Disketten Das bedeutet saite 18 Spiele von Action über Taktik bis Adventure für CPC 464/664 und 6128

- 101 Kassettenversion (4 Kassetten) **nur 50,- DM**
102 Diskettenversion (4 Disketten 3") **nur 70,- DM**

Stck. _____

»Startest« Das Superspiel für 464, 664 und 6128

- 103 Kassettenversion **24,- DM**
104 Diskette 3" **29,- DM**

Stck. _____

»Know« Das Spiel für die ganze Familie, für CPC 464/664/6128

- 105 Kassettenversion **24,- DM**
106 Diskette 3" **29,- DM**

Stck. _____

CPC-Sonderhefte

- Nr. 3 Nr. 4 Nr. 5 **à 14,- DM**

Stck. _____

Databox zum CPC-Sonderheft Nr. 6/88

- Kassette 1 Kassette 2 **à 14,- DM**
 Diskette 1 Diskette 2 **à 24,- DM**
 Kombi-Pack (Kassette 1 + 2) **à 25,- DM**
 Kombi-Pack (Diskette 1 + 2) **à 38,- DM**

Gesamtbetrag

zzgl. DM 3,- DM Porto/Verpackung. (Ausland 5,- DM)

- Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.
 Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum

Unterschrift (bei Minderjährigen Unterschrift des gesetzl. Vertreters)

Absender: (Bitte genaue Anschrift angeben)

Name _____

Vorname _____

Str./Nr. _____

PLZ/Ort _____

Der Joystick schießt alleine!

Da hat man sich nun, in der Hoffnung, bei Ballerspielen einer gewissen Erleichterung für seine Finger entgegenzusehen, einen Joystick mit AUTO-FEUER gekauft, und muß beim Testen bestürzt feststellen, daß die Sache mit dem AUTO-FEUER überhaupt nicht funktioniert. Statt dessen schreibt der neue Joystick ein störendes Z nach der Einschaltmeldung auf den Bildschirm und läßt es dann noch nicht einmal zu, den Computer mit den drei bekannten Tasten zurückzusetzen. Was also tun??? Den Joystick in die Ecke schmeißen? Nein, es gibt da noch eine andere Lösung.

Man muß sich nur das kleine Gerät einmal von innen etwas näher zu Gemüte führen. Da befindet sich eine kleine Schaltung drin, genauer gesagt ein Taktgenerator. Dieser kann über den AUTO-FEUER Schalter aktiviert werden, gleichzeitig wird der oben eingebaute Feuerknopf ausgeschaltet. Nun, die Schaltung erwartet auf Pin 7 eine Spannung von +5 Volt und auf Pin 8 die dazugehörige Masse (GND), = 0 Volt. Diese 5 Volt kommen aber keinesfalls bei unserem Schneider dort heraus. Pin 7 ist nämlich der Anschluß für Feuer 1 und an Pin 8 liegt bei der Abfrage des Joysticks immer ein Taktsignal. Das kommt daher, daß der Schneider nur einen Stecker hat, aber damit zwei Joysticks abfragt. Dieses geschieht in einer Art Multiplexing-

Verfahren. Der Joystick 1 kann z.B. nur dann abgefragt werden, wenn an Pin 8 Masse (0 V) vom Computer anliegt. Der Zweite wird infolgedessen nur dann abgefragt, wenn an Pin 9 Masse (0 V) anliegt. Somit schaltet unser Computer immer schnell hin und her, um die Binärwerte der beiden Joysticks abwechselnd einzulesen. Das ominöse Z verdankt seine Entstehung dem Innenwiderstand, den diese kleine Schaltung hat, denn es fließt zwischen Pin 8 und Pin 7 ein kleiner Strom. Unser Computer denkt somit, es sei immer Feuer 1 gedrückt. Das naheliegenste wäre doch nun, den Joystick seine verlangten +5 V an Pin 7 und an Pin 8 die Masse zu geben. Der Joystick müßte dann nur noch kleine Relais ansteuern, die wiederum die entsprechenden An-

schlüsse miteinander verbinden. Doch wer will beim Spielen schon klickende Relais hören? Es gibt eine viel elegantere und preiswertere Lösung: sogenannte Analog-bilaterale Schalter. Mit ihnen kann man z.B. auch NF-Signale für Verstärker oder Mischpulte übertragen. Diese werden nun genauso wie die Taster im Joystick an die Eingänge des Joyports angeschlossen. Diese elektronischen Schalter schließen sich, wenn am Steuereingang logisch 1, also +5V, anliegt, und öffnen sich, wenn der Steuereingang an logisch 0 (Masse=0 Volt) angelegt wird. Jetzt gibt es aber noch ein Problem: Die Taster im Joystick selbst sind alle an 0 Volt angeschlossen, d.h., daß die elektronischen Schalter sich öffnen, wenn die entsprechende Taste im Joystick gedrückt wird. Sie sollten sich doch eigentlich schließen. Dieses Problem ist aber denkbar einfach zu lösen. Man baut in die Steuereingänge der Analog-Schalter Inverter ein. Diese erzeugen an ihrem Ausgang eine 1, wenn am Eingang dieser Bauteile eine 0 liegt und umgekehrt. Nun stimmt die Sache wieder. Ein gedrückter Taster im Joystick liefert eine 0, der Inverter macht daraus eine 1, die an den Steuereingang des elektronischen Schal-

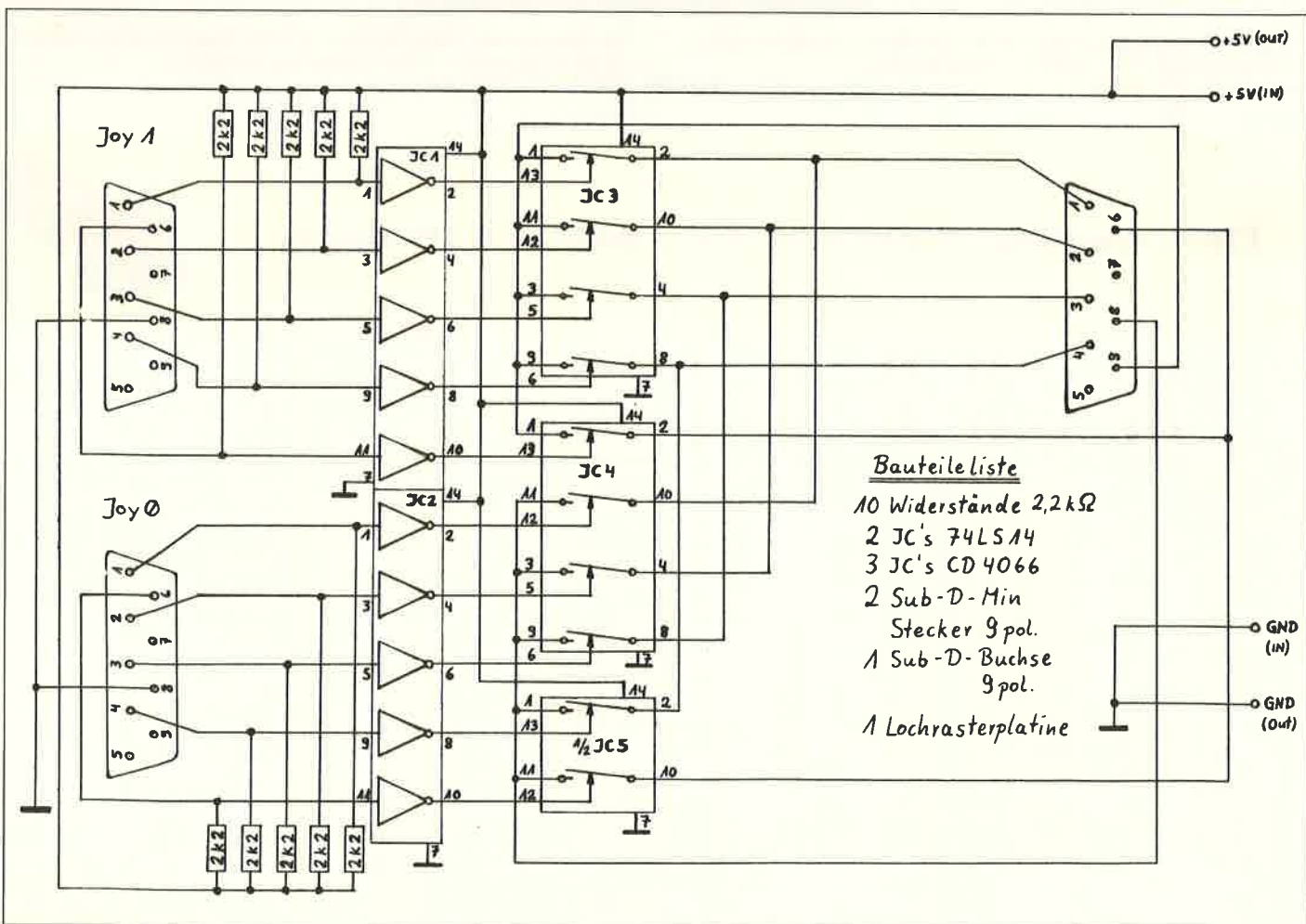


Abb. 1: Der Schaltplan für das Automatik-Feuer des Joysticks.

ters gelangt, und der schließt sich. Man sollte jedoch die Eingänge der Inverter mit je einem 2.2 kOhm Widerstand an +5V legen, weil die Dinger manchmal anschwingen, wenn die Eingänge frei sind.

Das äußert sich darin, daß man beim Berühren offener Kontakte oder Drähte des Aufbaus auf einmal einen Pfeilsalat auf dem Bildschirm wiederfindet, wenn der Computer gerade eingeschaltet wurde.

Die 5 Volt holt man sich am besten vom Monitor, indem man in das kleine Gerätchen, wie beim Rechner, eine Buchse mit einem Stiftdurchmesser von 2.1 mm einbaut und an einer anderen Stelle ein Kabel herausführt, an dem sich ein kleiner Stecker befindet, der dann in den Computer eingesteckt wird und den Strom liefert. Aber bloß nicht verpolen!!!

Messen Sie lieber vorher durch, ob der Rechner die Spannung in der richtigen Polung bekommt, sonst büßt man mindestens CPU und Gatearray ein, weil die sowas überhaupt nicht abkönnen.

Die Stecker für die beiden Joysticks können direkt in ein Gehäuse eingebaut, oder aber an Kabeln herausgeführt werden. Bei dem Anschluß für den Computer läßt

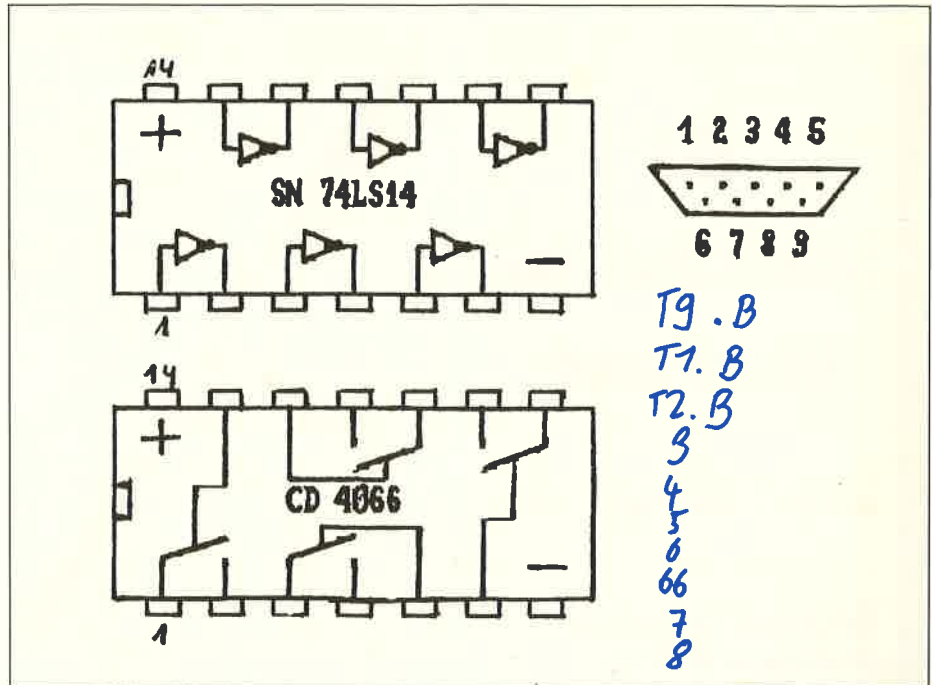


Abb. 2: Die Anschlußbelegung für ICs und Stecker.

sich ohne ein Kabel allerdings nichts machen. Die Schaltung in Abb. 1 kann auf einer Platine mit den Maßen 6 x 8 cm untergebracht werden. Die Anschlußbelegungen der IC's und der Stecker für die

Joysticks sind der Abb. 2 und der Liste 2 zu entnehmen. So, dann viel Spaß beim Basteln.

(H. Kleinert/JB)

Peeks und Pokes zur DDI-1

Auch wenn Sie mit Maschinensprache (noch) nichts anfangen können, gibt es für Sie einige Möglichkeiten, dem AMSDOS ein wenig (wenn auch nur begrenzt) »ins Handwerk zu pfuschen«. Dabei ergibt sich natürlich die eine oder andere interessante Anwendung:

Fast schon ein »Oldie« ist der Tip, wie man eine versehentlich gelöschte Datei wieder »aus der Versenkung holt«: In &A701 steht die aktuelle Usernummer. Poken Sie hier eine 229 (Kennnummer für gelöschte Files) und laden Sie die gewünschte Datei. Nun setzen Sie wieder !USER,0 (Das »!« soll den RSX-Strich darstellen) und speichern Sie die Datei wieder ab. Dieses klappt aber nur, wenn Sie die Datei erst kurz zuvor gelöscht haben!

Interessant ist auch die Speicherzelle &A89F: Hier findet sich das momentane Diskettenformat wieder, wenn Sie vorher CAT o.ä. eingegeben haben: Eine 65 zeigt an, daß Sie es mit dem CP/M-Format zu tun haben, eine 193 läßt auf das AMSDOS-Format (»Data-Only-«) schließen.

Wahrscheinlich haben Sie auch schon einmal ein »BIN-File« gefunden, von dem Sie weder die Startadresse noch die Länge kennen. Hier hilft nur eines: Laden Sie die Datei wie gewohnt und sehen Sie nach: &A767 enthält den Filetyp (bei BIN-Files also 2),

PRINT PEEK (&A76A) + 256 * PEEK (&A76B)

ergibt die Startadresse,

PRINT PEEK (&A76D) + 256 * PEEK (&A76E)

ergibt die Länge und

PRINT PEEK (&A76F) + 256 * PEEK (&A770)

ergibt die Autostart-Adresse.

Wenn Sie eine fremde Diskette lesen und der Rechner nach etli-

chem Hin und Her auf der Diskette ein »Read fail« ausgibt, ist es günstig, die Anzahl der Leseversuche zu erhöhen und so den Rechner anzuweisen, sich etwas mehr Mühe zu geben: Poken Sie in &BE66 einfach einen höheren Wert als den Standardwert 16 (d.h. max. 16 Versuche).

Setzen Sie in &BE78 einen Wert < > 0, so werden alle Fehlermeldungen der Floppy, die ein »Retry, Ignore or Cancel?« beinhalten, unterdrückt. Leider wird das nachfolgende »Bad Command« doch noch ausgegeben, aber wenigstens bleibt die Routine nicht mehr an der o.g. Abfrage hängen.

Die Bytes &BE44 bis &BE47 verändern die Wartezeiten während des Floppy-Betriebs: In &BE44 und &BE45 steht die Zeit, die nach dem Einschalten des Motors bis zum ersten Zugriff auf die Diskette gewartet wird, in &BE46/7 die Zeit nach dem letzten Zugriff, beide Zeiten sind in Einheiten von 1/50 Sekunde dargestellt.

Möchten Sie jedoch nach dem letzten Disketten-Zugriff den Motor eingeschaltet lassen, so müssen Sie nur, während der Motor nachläuft, in das Byte &BE5F eine 0 poken.

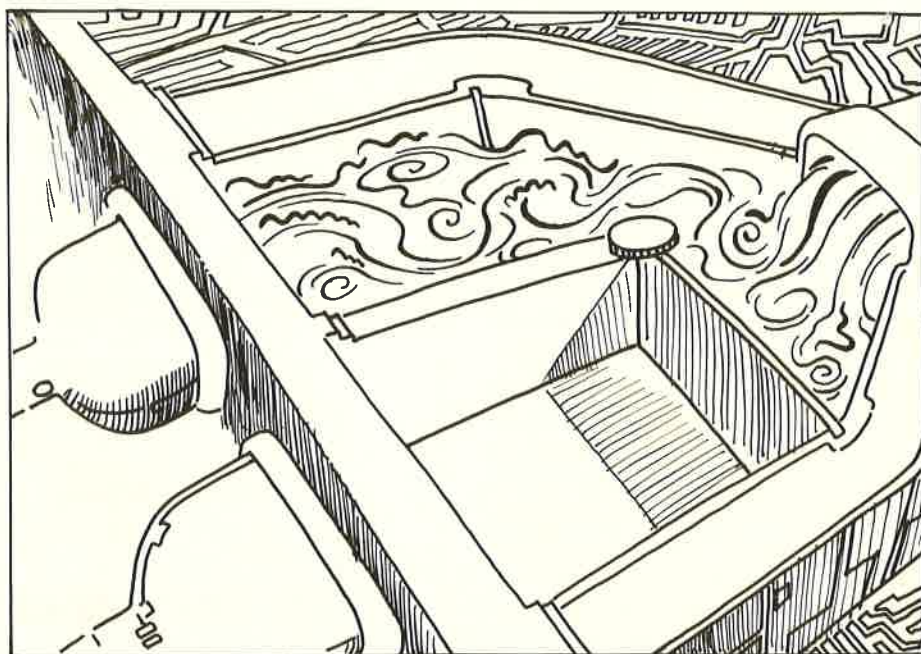
Zum Schluß noch ein kurzer Tip für alle Leser, die stets irgendwelche Dateien geheimhalten müssen: Legen Sie eine Diskette im CP/M-System-Format ein, führen einen CAT durch und tippen dann: POKE &A8A8,1 und POKE &A89D,0. Danach speichern Sie eine kleine Testdatei ab und setzen den gesamten Rechner zurück. Wenn Sie sich nun Ihre Diskette per CAT ansehen, werden Sie Ihre Testdatei vergeblich suchen. Geben Sie jedoch die obige Befehlsfolge ein, so wird Ihre Datei auf magische Art wieder erscheinen! Nachteil dieser Methode ist nur, daß Sie im »Geheimkatalog« nicht mehr als 7 KByte Dateien ablegen können, wenn Sie den normalen Katalog auch noch benutzen wollen, andernfalls wird letzterer einfach überschrieben.

(T. Kochmann)

Der Schnittstellen- Umschalter

CPC · JOYCE · PC

Die Zeiten, da Profi-Programmierer über die sogenannten Hobby-Computerfreaks die Nase rümpften, sind – von einigen Unbelehrbaren einmal abgesehen – nahezu vorbei. Im Gegenteil, die Leistungen, die heute von den »Amateuren« auf dem Gebiet der Datenverarbeitung und Programm-Erstellung erbracht werden, rufen nicht selten bei »Fachleuten« Erstaunen hervor. Als Beispiel sei nur der Wettbewerb »Jugend forscht« genannt. Auch hier ist die Datenverarbeitung und der Einsatz von Home- und Personalcomputern mittlerweile in einem solchen Ausmaß und in einer solchen Qualität vertreten, daß es zur Pflicht für die großen Softwarehäuser gehört, sich dort die Konkurrenz anzuschauen und evtl. junge Mitarbeiter anzuwerben. Auch auf dem Gebiet der Hardware-Ausrüstung ziehen die Amateure nach.



Der Drucker – früher eine Seltenheit – ist mittlerweile selbstverständlich. Modem, Maus, Terminal und Zweitcomputer findet man heute immer häufiger auf den Schreibtischen junger Computerfreaks.

Mit steigender Anzahl von Peripheriegeräten nimmt auch die Anzahl der Kabel, Stecker, Verlängerungen, Adapter und damit auch die Gefahr der Vertauschungsmöglichkeiten zu. Hier sieht es dann manchmal gar nicht profihaft aus und Verbindungen werden oft nach der Methode »Probieren geht über studieren« hergestellt. Abhilfe bei »Kabelsalat« und Schnittstellenwarr soll der hier vorgestellte Umschalter bringen.

Kein Kabeltauschen, Umstecken, Einfügen von Adaptern mehr, ein Dreh am Wahlschalter genügt und die Daten werden nicht mehr zum Matrixdrucker, sondern zum Zweitrechner, Typenradrucker oder Modem gesendet.

Schnittstellenprobleme

Zunächst jedoch einige Erläuterungen zur Schnittstellenstruktur. Das rapide Wachstum der Computerindustrie in den sechziger Jahren führte zu Schwierigkeiten bei der Zusammenschaltung von Rechnern und Peripheriegeräten. Jeder Hersteller hatte seine eigene Schnittstellennorm. Der Drucker der Marke X konnte nicht – oder zumindestens nur mittels teurer Spezialadapter – am Rechner der Marke Y betrieben werden. Ein

Zustand, der uns ja auch auf anderen Gebieten der Computerhardware bekannt ist.

»Gottseidank« hat sich die Lage mittlerweile zum Wohle der Anwender geändert. Von den vielen Hausnormen haben sich zwei durchsetzen können und werden mittlerweile von allen Rechnerherstellern (zumindest als Option) angeboten.

Die Parallele Centronics

Da wäre zunächst die Centronics-Schnittstelle. Sie ist von der Struktur her recht einfach aufgebaut. Acht parallele Datenleitungen (Bit 0 bis Bit 7), eine STROBE- und eine BUSY-Leitung werden zur Datenübertragung benutzt. Vor der Ausgabe eines Zeichens prüft der Rechner die BUSY-Leitung, ob der Drucker zur Aufnahme eines Zeichens bereit ist. Wenn BUSY = 0 Volt, so setzt der Rechner die acht Datenleitungen in der ASCII-Codierung des entsprechenden Zeichens. Etwa 1 Mikrosekunde später wird mit der STROBE-Leitung signalisiert, daß die Daten gültig sind. Zeichen für Zeichen wird so nacheinander übertragen. Man spricht, wenn man es exakt ausdrücken will, von einer Bit-parallelen/Zeichenseriellen Übertragung. Der Signalpegel beträgt 5 Volt; aus diesem Grund ist auch die maximale Kabellänge auf etwa 2 Meter begrenzt. Die Datenrichtung ist fest vorgegeben, d.h. es können nur Daten vom Rechner ausgegeben, nicht jedoch

eingelassen werden. Bild 1 zeigt die Belegung und den Aufbau einer Centronics-Schnittstelle.

Die Serielle RS-232C

Leistungsfähiger und flexibler ist eine serielle Schnittstelle, die 1969 in der amerikanischen Norm RS-232C festgelegt wurde. In Europa wird sie auch noch als V.24-Verbindung bezeichnet; ebenfalls eingebürgert hat sich die Bezeichnung SIO (Seriell Input/Output). Bei ihr wird die Bit-Information eines jeden Zeichens seriell, d.h. nacheinander, übertragen. Dies bedeutet zwar eine Reduzierung der Leitungen, jedoch auch eine komplexere Datenkommunikation. Die Vorteile der RS-232C-Schnittstelle überwiegen jedoch, so daß es sicherlich sinnvoll ist, seinen CPC oder seinen Joyce mit einer solchen zusätzlichen Schnittstelle auszurüsten. Beim Schneider PC waren die Konstrukteure fortschrittlicher, hier wurde bereits in die Grundausstattung eine Centronics und eine RS-232C eingebaut. Der 25-polige Stecker ist mit Pin-Belegung, Name und Aufgabe der Signalleitungen in Bild 2 aufgezeigt. Der Signalpegel beträgt +12/-12 Volt, das entspricht einem Spannungshub von 24 Volt. Deshalb ist auch eine Übertragung bis zu 15 Metern (bei langsamer Übertragungsrates auch weiter) problemlos. Vom Datenaustausch über Telefonnetz bis nach USA ganz zu schweigen. Die Übertragung der Daten erfolgt über die Leitung TxD (Daten senden), bzw. RxD (Daten empfangen). Dies geschieht bitseriell, d.h. über TxD wird für jedes zu übertragende Zeichen ein komplettes Impulspaket gesendet. Es besteht aus:

- Startbit 1 Bit
- Zeichencode 5, 6, 7 oder 8 Bit
- Parität 0 oder 1 Bit
- Stopbit 1, 1.5 oder 2 Bit

Voraussetzung ist hierbei, daß Sender und Empfänger im gleichen Übertragungsmodus arbeiten müssen, anderenfalls gibt es »Verständigungsschwierigkeiten«. Ebenso übereinstimmen muß die Übertragungsgeschwindigkeit. Denn anders als bei der Parallelschnittstelle, ist eine Quittierung jedes Einzelsignals (hier also Bit) nicht möglich. Die Geschwindigkeit wird in Baud gemessen (Anzahl der Signale pro Sekunde) und läßt sich

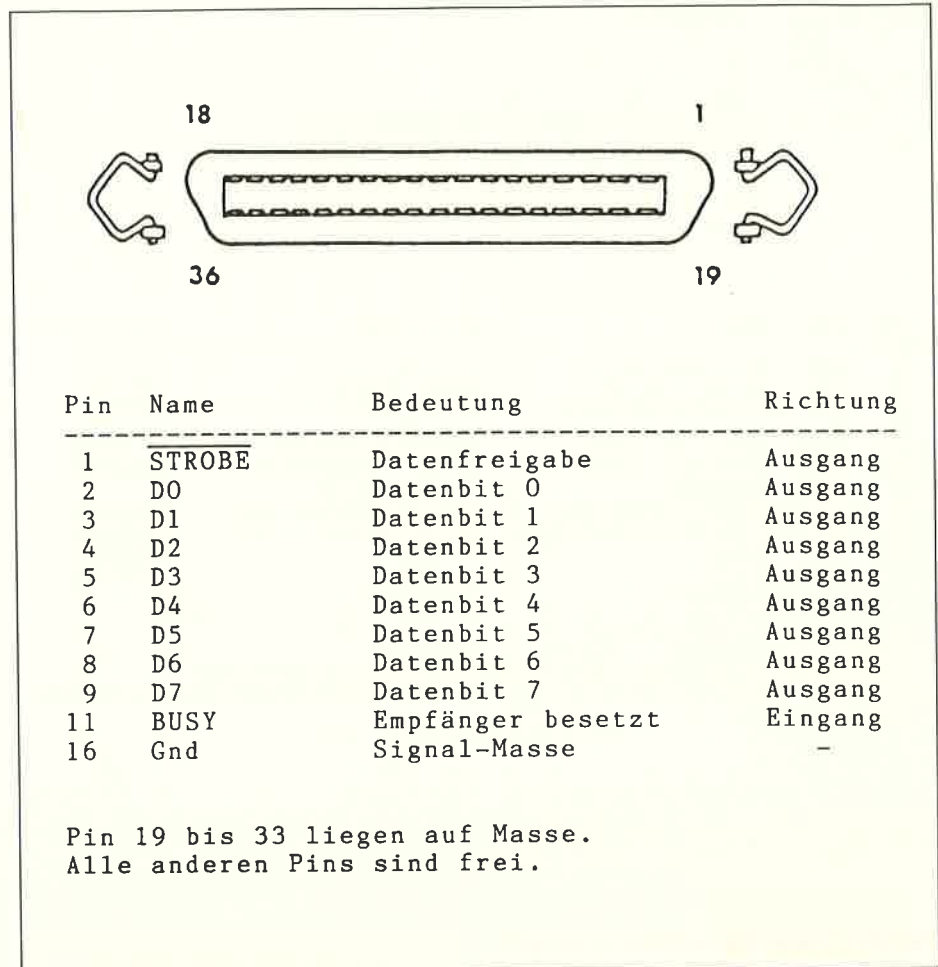


Bild 1: Aufbau und Belegung der Centronics-Buchse

bei den Schneider Computern zwischen 50 und 19200 Baud in vorgegebenen Stufen wählen. Da pro Zeichen etwa 10 Bit übertragen werden müssen, kann man für die Übertragung der Zeichen mit etwa 1/10 der Baudrate rechnen. Bei 9600 Baud würden also ca. 960 Zeichen pro Sekunde übertragen.

Die Einstellung der Parameter wird bei den Peripheriegeräten meist an DIL-Schaltern vorgenommen, während die Rechnerschnittstellen (SIO's) per Programm eingestellt werden. Wenn Baudrate und die anderen Parameter von Sender und Empfänger übereinstimmen, so könnten theoretisch mit nur einer Signalleitung (und Masse) bereits Daten übertragen werden: Der TxD-Ausgang des Rechners wird auf den RxD-Eingang des Druckers gelegt, dann kann's losgehen. Was jedoch, wenn der Drucker gar nicht eingeschaltet war?

Der Rechner – ohne Kenntnis davon – würde trotzdem die Daten ausgeben. Oder, wenn der Empfänger die Datenflut nicht schnell genug verarbeiten kann und vielleicht zwischenspeichern muß? Das

sendende Gerät würde ihn mit Daten überrollen. Man sieht schnell ein, daß ein Quittierungsverfahren – im Fachjargon Handshake genannt – in der Datenkommunikation unbedingt erforderlich ist.

Handshaking

Bei der V.24/RS-232C kann auf zweierlei Art die Datensteuerung vorgenommen werden: mit dem Software-Handshake oder mit dem Hardware-Handshake. Beim Ersteren werden zusätzlich zu den Daten auch noch Steuerzeichen auf TxD und RxD übertragen. Das Sendegerät erwartet also auf seiner Empfangsleitung die Quittierung des Empfangsgerätes. Anderenfalls stoppt es die Übertragung. Dieses Verfahren wird – da nur zwei Leitungen benötigt werden – bei der Kommunikation über Telefonnetz angewendet.

Bei Druckern oder ähnlichen Peripheriegeräten wird die Steuerung meist jedoch hardwaremäßig vorgenommen. Diesem

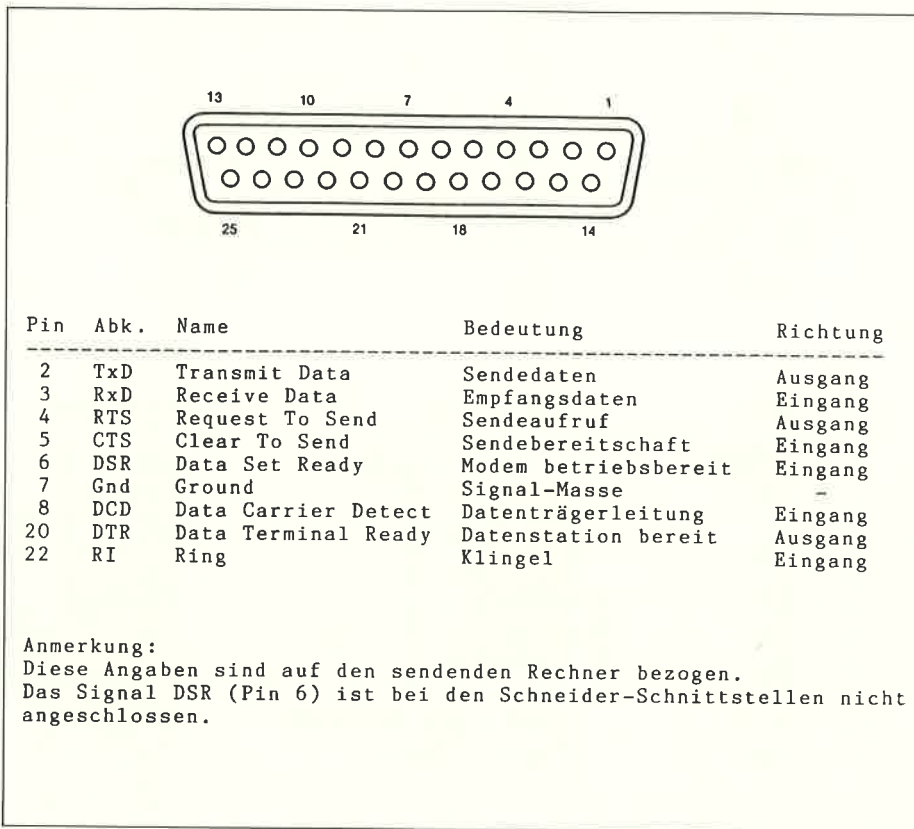


Bild 2: Aufbau und Belegung des RS-232 C-Steckers

Zwecke dienen die anderen Leitungen der Schnittstelle. Sie signalisieren durch ihren Spannungspegel den Zustand des Gerätes. Wird z.B. der Drucker während der Datenübertragung auf Off-Line geschaltet, um neues Papier einzulegen, so setzt er das DTR-Signal (Data Terminal Ready) zurück. Der Rechner erkennt dies und stoppt die Datenausgabe.

Ähnliche Aufgaben haben auch die anderen Steuerleitungen: Mit RTS (Request To Send) signalisiert der Rechner, daß er Daten übermitteln will. Mit CTS (Clear To Send) zeigt daraufhin das Empfangsgerät seine Bereitschaft an, die Daten zu empfangen.

Das Signal DSR (Data Set Ready) wird ähnlich wie das oben beschriebene Signal DTR zur Anzeige der Betriebsbereitschaft des Empfängers benutzt. Auf die Abfrage des DSR-Signals wird jedoch in den meisten Fällen verzichtet, so auch bei allen Schneider Schnittstellen. Mit dem Signal DCD (Data Carrier Detect) wird bei Modems das Erkennen des Trägersignals gemeldet. Häufig ist es mit dem Signal DTR gekoppelt.

Oft vergessen wird die Klingelleitung RI an Pin 22. Ein historisches Überbleibsel, wohl als eine Art »Wecker« für den eingeschlafenen Programmierer gedacht.

Immer noch Probleme

Eigentlich dürfte nun einer Datenübertragung nichts mehr im Wege stehen. Doch weit gefehlt! Die Möglichkeit, Daten in beide Richtungen senden zu können, macht die Sache etwas komplizierter. Denn wer ist Sender, wer Empfänger der Daten? Ist doch klar, wird man zunächst vermuten: Ein Drucker ist doch immer Empfänger, der Rechner immer Sender. Doch wie ist das bei Anschluß eines Telefon-Modems oder der Kopplung zweier Rechner???

Ursprünglich war diese Schnittstelle für den Datentransfer über Telefon-Modems konzipiert. Hier gab es keine Anschlußprobleme: Tastatur an TxD, Bildschirm an RxD. Verzichtet man jedoch auf das Modempaar und verbindet z.B. zwei Rechner direkt miteinander, so muß logischerweise das Ausgangssignal TxD des sendenden Rechners für den empfangenden Rechner das Eingangssignal RxD sein. Pin 2 des Senders ist also mit Pin 3 des Empfängers zu verbinden. Dies gilt ebenso für die Steuerleitungen, auch hier müssen Eingänge und Ausgänge über Kreuz angeschlossen werden. Ein Verbindungskabel, das die Aufgabe der Signalvertauschung übernimmt, wird als

»Null-Modem-Kabel« bezeichnet. Wird dagegen über Modem gefahren, so muß eine Verbindung mit einem ein-zu-eins-Kabel hergestellt werden.

Einen Strich durch unsere Logik machen nun noch die Druckerhersteller. Von der Funktion her eigentlich ausschließlich Empfänger, sind Drucker mal als Modem, mal als Terminal geschaltet. Mit anderen Worten: Einmal liegt an Pin 2 das Signal TxD, ein anderes mal RxD. Auch das »Geschlecht« des 25-poligen Anschlusses gibt nicht immer Aufschluß über dessen Belegung. Terminals (also Rechner) sollten eigentlich immer mit einem »männlichen« Stecker, Modems einer »weiblichen« Buchse ausgerüstet sein. Doch auch hier gilt: Im Zweifelsfalle im Handbuch oder dem Schaltbild des Gerätes nachsehen.

Der Schnittstellenumschalter

Dieses Problem (Null-Modem oder Direktverbindung) ist mit dem vorgeschlagenen Schnittstellen-Umschalter ebenfalls gelöst. Hier genügt ein Tastendruck um die Kreuzung der entsprechenden Signal- und Datenleitungen vorzunehmen. In Bild 3 ist der Stromlaufplan dargestellt. Die Funktion ist leicht überschaubar, der Nachbau für Computerfreaks mit etwas Löterfahrung sicher nicht schwierig. Die Signale TxD, RxD, CTS und DTR werden je nach Stellung des Wahlschalters einem der drei Ausgangskanäle zugeführt. Auf die Umschaltung der Signale RTS, DSR und DCD wurde verzichtet und durch die Brückung von Pin 4, 6 und 8 eine Art »Selbstquittierung« vorgenommen. Die Besonderheit dieses Gerätes: Jeder Ausgang ist mit einem »Null-Modem-Schalter« ausgerüstet. Mit diesem werden sowohl die Datenleitungen, als auch die beiden Steuerleitungen entweder vertauscht oder direkt verbunden. Das häufig benötigte und oben bereits erläuterte Null-Modem-Kabel wird so eingespart und eine einfache Anpassung der angeschlossenen Geräte ermöglicht. Die vier Leuchtdioden sind für die Funktion des Umschalters zwar nicht erforderlich, sie sind aber ein gutes Hilfsmittel, um den Datenaustausch optisch zu verfolgen. Für die beiden Signale TxD und RxD zeigt jeweils eine LED den positiven, die andere LED den negativen Pegel an. Empfohlen werden

klare Leuchtdioden, so lassen sich selbst bei hohen Übertragungsraten noch kurze Datenblöcke gut erkennen.

Dummy-Funktion

Auf eine weitere Besonderheit des Gerätes sei noch hingewiesen: Die »Dummy-Funktion«. In Stellung 4 des Kanalwahlschalters wird dem Rechner ein Ausgabegerät vorgegaukelt. Die Daten werden über RxD zurückgesendet und mit der DTR-Leitung quittiert.

So kann die Schnittstelle ausprobiert werden oder Programme mit Schnittstellen-ausgabe getestet werden, ohne Papier zu produzieren. Ebenso können Textteile während des Ausdrucks quasi ausgeblendet werden. So lassen sich beispielsweise die Seiten 3, 5 und 9 alleine ausdrucken, alle anderen Seiten schluckt der »Dummy«. Der Aufbau bereitet sicher keine

Probleme. Bei Gehäuse und Schaltern kann nach eigenem Geschmack oder auch Geldbeutel gewählt werden.

Für die Kanalumschaltung ist ein Stufen-Drehschalter zweckmäßig, als »Null-Modem-Schalter« können vierpolige Kipp-schalter oder Schiebeschalter eingesetzt werden.

Um ein zusätzliches Adapterkabel einzusparen, wurde auf eine Eingangsbuchse verzichtet. Der Rechneranschluß erfolgt direkt über die 25-polige Buchse des Eingangskabels.

Hier noch ein Hinweis: Wie aus der Schaltung ersichtlich, werden für die Übertragung 5 Datenleitungen und Signal-Masse benötigt. Es würde also ein 6-adriges Kabel ausreichen. Zum Schutz vor Störstrahlungen (sowohl in die Datenleitung, als auch durch diese selbst) sollte ein abgeschirmtes Kabel verwendet werden. Die Abschirmung kann dann beid-seitig mit Ground (Pin 7) verbunden wer-

den. Steht nur ein 5-adriges abgeschirmtes Kabel zur Verfügung, so kann die Abschirmung auch als GND-Verbindung benutzt werden. Die Länge des Kabels richtet sich nach dem vorgesehenen Standort. Die »Dummy«- und die »Null-Modem-Funktion«, aber auch die Anzeige der Signalpegel machen diesen Umschalter auch für solche Programmierer oder Programmiererinnen interessant, die bisher nur ein Peripheriegerät besitzen. Sollten dagegen bei großem Gerätepark die 3 Kanäle nicht ausreichen, so läßt sich der SIO-Umschalter beliebig erweitern.

Der Kanalstufenschalter ist entsprechend der gewünschten Kanalanzahl auszulegen, die weiteren Kanäle sind äquivalent zu den ersten drei zu verdrahten. So etwas wie eine maximale Belastung mit Schnittstellen gibt es nicht, es kann nach Lust und Laune erweitert werden.

(W. Otternberg/JB)

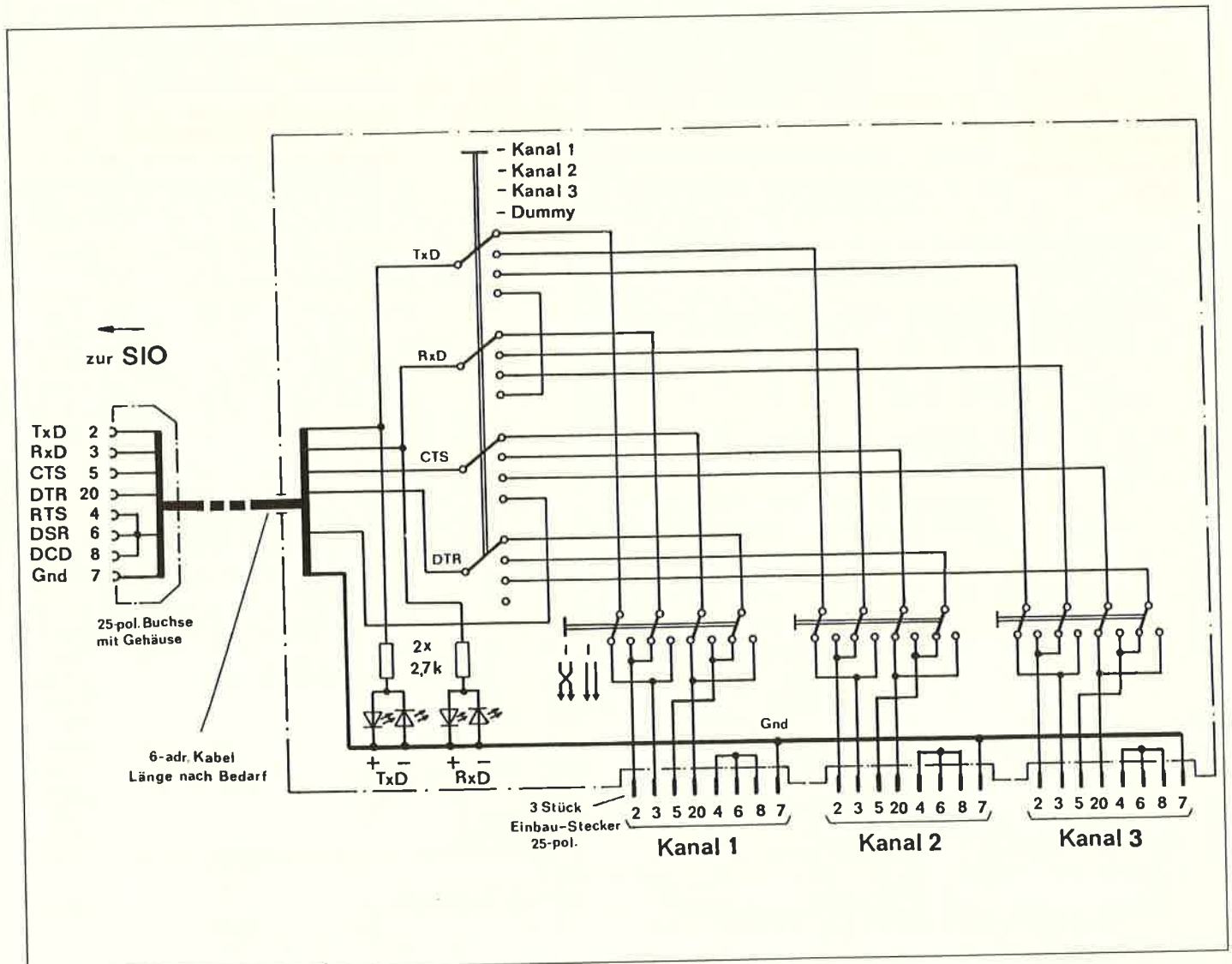


Bild 3: Schaltung des Schnittstellen-Umschalters

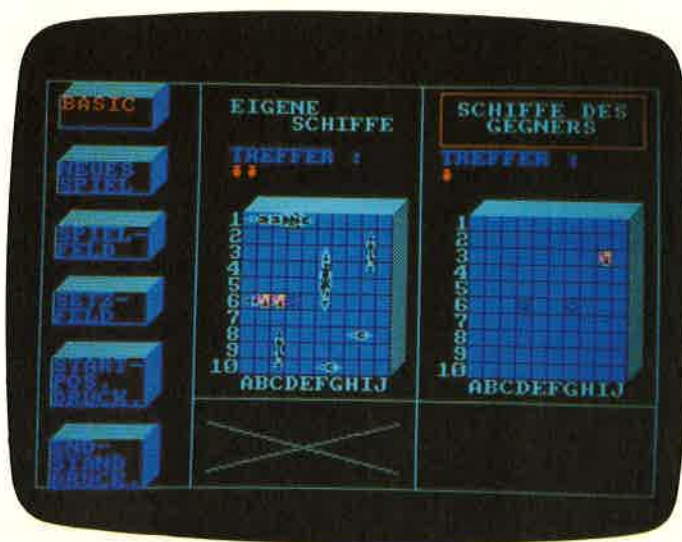


Bild 1: Das Spielfeld von Subunk

SUB SUNK

Dieses Programm ist eine Umsetzung des Spieles »Schiffeversenken«. Da die groben Regeln fast jedem bekannt sind, brauchen diese kaum noch erklärt zu werden. Die Regeln, die von jedem anders gehandhabt werden, sind wie folgt festgelegt worden:

- Es wird gesetzt:
 ein Kreuzer (Länge 5)
 ein Zerstörer (Länge 4)
 drei Torpedoboote (Länge 3)
 zwei U-Boote (Länge 2)
- Die Boote müssen immer ein Quadrat Abstand zwischen einander haben.
- Es kann waagrecht u. senkrecht gesetzt werden (nicht diagonal).
- Es wird immer abwechselnd geschossen.

Zum Programm

Das Programm ist ganz in Basic geschrieben. Der Benutzer spielt gegen den Computer. Beim Setzen der Schiffe muß zuerst der Anfangspunkt angeklickt werden und dann die Richtung, in die das Schiff gesetzt werden soll. Das Programm versteht sich weitgehend von selbst, so daß man so gut wie keine Fehler machen kann.

- Mit der Leertaste kommt man, außer vom Setzfeld, immer zurück zum Menue!
- Die beiden letzten Funktionen können aus »Mogelgründen« erst am Ende betätigt werden.
- Der Drucker muß Epson kompatibel sein!
- Hier noch eine Liste der Variablen aus denen man die Bedeutung nicht eindeutig entnehmen kann:

esc\$=chr\$(27)	norm\$=esc\$+"2"
klein\$=esc\$+"0"	xfeld=Setzfeld oder Spielfeld
sende=Spieldende	l= xfeld 1 oder 2
n= window 2 oder 3	ax und ay= Suchvariablen sx und sy
lx und ly= Merkvariablen von ax und ay	xc und yc= Computer-schiess-Koordinaten
waag= waagrecht	senk= senkrecht
merke und merken= Merkvariablen von senk und waag	s und w = senkrechte u. waagerechte Malvariable
zähli und zählen = Treffer Zähler	px und py = Schiffsanfangskordinaten
xende und yende = Schiffsendekordinaten	voll = Wassersuchvariable
hl = Musikstück 1 o. 2	

(Carsten Hesender/SH)

```

10 SYMBOL AFTER 255 [1409]
20 GOSUB 6040 [899]
30 h1=0:GOSUB 6290 [618]
40 '***** INITIALISIERUNG ***** [2792]
*****
50 DIM xfeld(11,11,1) [715]
60 MODE 1:PRINT CHR$(23);CHR$(0); [2441]
70 EI [86]
80 GOSUB 4340 [1061]
90 esc$=CHR$(27):norm$=esc$+"2":klein$=esc [1762]
$+"0"
100 zaehler=0 [699]
110 true=-1:false=NOT true:taste=false [3629]
120 punkt=false:versunken=false:geschossen [6840]
=false:voll=0:wasser=false:gezielt=false:z
iel=false
130 sende=false:setz=false [1688]
140 INK 0,0:INK 1,13:INK 2,2:INK 3,6:BORDE [1251]
R 0
150 '***zum SCHRIFT AUFBAU *** [1339]
160 GOSUB 3710 [1011]
170 '***zur MASKE u. WINDOWS*** [1277]
180 GOSUB 3510 [1019]
190 GOSUB 3100:GOSUB 3090 [2481]
200 '***SCHRIFTKASTEN*** [1152]
210 FOR kastn=1 TO 16 STEP 4 [1190]
220 GOSUB 3630 [1194]
230 LOCATE#1,8,kastn+2:PRINT#1,CHR$(212) [2509]
240 MOVE 112,-((kastn+2)*16)+400:DRAWR -98 [2752]
,0,2:DRAWR 0,30,1
250 NEXT kastn [723]
260 FOR kastn=17 TO 25 STEP 5 [364]
270 GOSUB 3630 [1194]
280 LOCATE#1,8,kastn+2:PRINT#1,CHR$(233); [2482]
290 LOCATE#1,8,kastn+3:PRINT#1,CHR$(212); [1333]
300 MOVE 112,-((kastn+3)*16)+400:DRAWR -98 [2030]
,0,2:DRAWR 0,46,1
310 NEXT kastn [723]
320 PEN 1 [549]
330 MOVE 16,399:DRAWR 16,0:MOVE 112,0:DRAW [2369]
R 16,0
340 '***** NUMERIERUNG U. UMRAMUNG ***** [1369]
350 LOCATE 13,9 [518]
360 FOR zahl=49 TO 57 [1219]
370 LOCATE 13,zahl-40 [685]
380 GOSUB 3920 [1001]
390 NEXT zahl [419]
400 LOCATE 12,18:PRINT"10" [545]
410 FOR zahl=49 TO 57 [1219]
420 LOCATE 28,zahl-40 [686]
430 GOSUB 3920 [1001]
440 NEXT [350]
450 LOCATE 27,18:PRINT"10" [1100]
460 LOCATE 14,19 [644]
470 FOR bustab=65 TO 74 [1056]
480 GOSUB 3940 [997]
490 NEXT [350]
500 LOCATE 29,19 [456]
510 FOR bustab=65 TO 74:GOSUB 3940 [1745]
520 NEXT [350]
530 x=14:y=8 [729]
540 GOSUB 3980 [989]
550 x=29:y=8 [1030]
560 GOSUB 3980 [989]
570 PEN#2,2 [524]
580 PEN#3,2 [528]
590 '***** SETZEN DER FELDER ***** [1567]
**
600 FOR x=1 TO 10 [803]

```

Listing: Sub Sunk

```

610 FOR y=1 TO 10 [1115]
620 LOCATE#2,x,y:xfeld(x,y,0)=223:PRINT#2, [1908]
CHR$(223);
630 LOCATE#3,x,y:xfeld(x,y,1)=223:PRINT#3, [3223]
CHR$(223);
640 NEXT y [359]
650 NEXT x [356]
660 setzen=false [1072]
670 '***** Steuerung von Menu **** [3232]
*****
680 j=0:PEN#1,1 [1051]
690 sty=2 [564]
700 alty=sty [211]
710 GOSUB 4070 [1011]
720 WHILE j<>16 [847]
730 CALL &BB06:j=JOY(0) [913]
740 IF j=1 THEN sty=sty-4 [825]
750 IF j=2 THEN sty=sty+4 [2190]
760 IF sty<2 THEN sty=2 [1522]
770 IF sty>22 THEN sty=22 [1148]
780 GOSUB 4070 [1011]
790 WEND [390]
800 LOCATE#1,9,sty:PRINT#1," "; [1210]
810 wahl=(sty-2)\4+1 [1528]
820 ON wahl GOTO 1450,1420,2010,840,1490,1 [1952]
500
830 ' [117]
840 '***** SETZEN DER SCHIFFE ***** [2443]
*****
850 ' [117]
860 IF setzen=true THEN GOTO 670 [977]
870 n=2:l=0 [149]
880 PAPER#4,0:CLS#4:transein$=CHR$(22)+CHR [4543]
$(1):transaus$=CHR$(22)+CHR$(0)
890 LOCATE#2,1,1:PEN#2,3:PRINT#2,transein$ [3671]
;CHR$(226);
900 n=2:x=1:y=1 [773]
910 PEN#4,3:LOCATE#4,1,1:PRINT#4,"BITTE KR [1837]
EUZER"
920 LOCATE#4,1,2:PRINT#4,"SETZEN!"; [1330]
930 kreuzer$=CHR$(129)+CHR$(131)+CHR$(133) [4052]
+CHR$(135)+CHR$(137)
940 transkreuzer$=CHR$(130)+CHR$(132)+CHR$ [4327]
(134)+CHR$(136)+CHR$(138)
950 PEN#4,1:LOCATE#4,9,2:PAPER#4,2:PRINT#4 [10571]
,kreuzer$;:PEN#4,0:LOCATE#4,9,2:PRINT#4,tr
ansein$;:PRINT#4,transkreuzer$;transaus$;:
PAPER#4,2
960 kreuzer=1 [1485]
970 laenge=5 [561]
980 GOTO 1280 [347]
990 ' [117]
1000 kreuzer=0:laenge=4:zerst=1 [2655]
1010 PAPER#4,0:CLS#4:PEN#4,3:LOCATE#4,1,1: [4082]
PRINT#4," ZERSTOERER";
1020 LOCATE#4,1,2:PRINT#4,"SETZEN!" [1216]
1030 zerst$=CHR$(149)+CHR$(151)+CHR$(153)+ [3370]
CHR$(155)
1040 transzerst$=CHR$(150)+CHR$(152)+CHR$( [2022]
154)+CHR$(156)
1050 LOCATE#4,9,2:PAPER#4,2:PEN#4,1:PRINT# [3794]
4,zerst$;
1060 LOCATE#4,9,2:PEN#4,0:PRINT#4,transein [4790]
$;transzerst$;transaus$;
1070 GOTO 1280 [347]
1080 ' [117]
1090 zerst=0:laenge=3:torp=1 [1524]
1100 zaehler=zaehler+1 [2242]
1110 PAPER#4,0:CLS#4:PEN#4,3:LOCATE#4,1,1: [4942]
PRINT#4," TORPEDOBOOT";
1120 LOCATE#4,2,2:PRINT#4,"SETZEN"; [2277]
1130 torp$=CHR$(173)+CHR$(175)+CHR$(177) [1747]

```

Listing: Sub Sunk

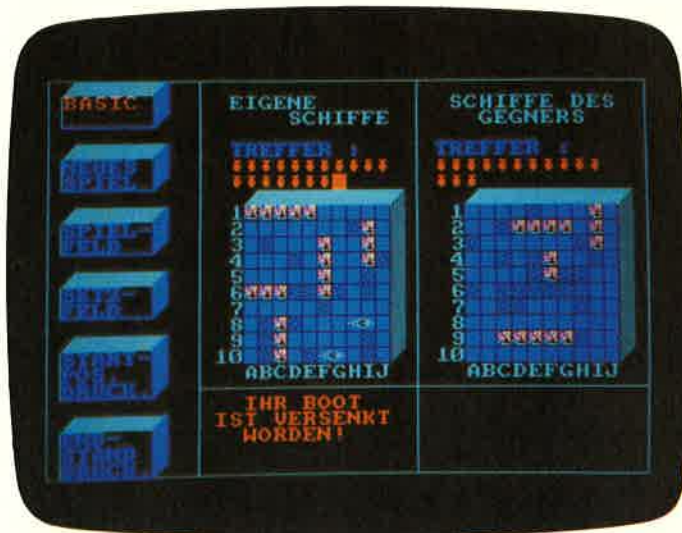


Bild 2: So sieht ein Volltreffer aus

```

1140 transtorp$=CHR$(174)+CHR$(176)+CHR$(1 [2109]
78)
1150 LOCATE#4,10,2:PAPER#4,2:PEN#4,1:PRINT [3538]
#4,torp$;
1160 LOCATE#4,10,2:PEN#4,0:PRINT#4,transei [4103]
n$;transtorp$;transaus$;
1170 GOTO 1280 [347]
1180 ' [117]
1190 torp=0:untbot=1:laenge=2:PAPER#4,0:CL [5103]
S#4:LOCATE#4,1,1:PEN#4,3:PRINT#4,"BITTE U-
BOOT";
1200 zaehler=zaehler+1 [2242]
1210 LOCATE#4,1,2:PRINT#4,"SETZEN"; [1257]
1220 uboot$=CHR$(165)+CHR$(167) [2008]
1230 transuboot$=CHR$(166)+CHR$(168) [1364]
1240 LOCATE#4,10,2:PAPER#4,2:PEN#4,1:PRINT [3714]
#4,uboot$;
1250 LOCATE#4,10,2:PEN#4,0:PRINT#4,transei [3618]
n$;transuboot$;transaus$;
1260 ' [117]
1270 ' ***** JOYSTICK ABFRAGE NACH [3893]
SETZ BEFEHLEN *****
1280 ok=false:GOSUB 4140 [1620]
1290 IF j=16 AND punkt=false AND xfeld(x,y [7579]
,1)=223 THEN GOSUB 5200:ELSE IF j=16 AND
punkt=true AND (x=px AND y=py) THEN GOSUB
5210
1300 ' [117]
1310 IF j=16 AND punkt=true AND (x>px AND [17335]
y=py) AND xfeld(x,y,1)=223 THEN xende=x:ye
nde=y:LOCATE#2,xende,yende:PEN#2,2:PRINT#2
,CHR$(229);:PEN#2,3:LOCATE#2,xende,yende:P
RINT#2,transein$;:LOCATE#2,x,y:PRINT#2,CHR
$(230);transaus$;:waage=1:senk=0:GOSUB 522
0:
1320 ' [117]
1330 IF j=16 AND punkt=true AND (x=px AND [13930]
y>py) THEN xende=x:yende=y:LOCATE#2,x,y:PE
N#2,2:PRINT#2,CHR$(231);:LOCATE#2,x,y:PRIN
T#2,transein$;:PEN#2,3:LOCATE#2,x,y:PRINT#
2,CHR$(232);transaus$;:waage=0:senk=1:GOSU
B 5480:GOSUB 5720
1340 ' [117]
1350 IF ok=true AND fehler=false AND senk= [4444]
0 THEN GOSUB 5380
1360 IF ok=true AND fehler=false AND senk= [4848]
1 THEN GOSUB 5530
1370 IF kreuzer=1 AND setz=true THEN setz= [1952]
false:GOTO 1000
1380 IF zerst=1 AND setz=true THEN setz=fa [1971]
lse:GOTO 1090

```

Listing: Sub Sunk

```

1390 IF torp=1 AND setz=true AND zaehler<3 [9656]
  THEN setz=false:GOTO 1090 ELSE IF torp=1
  AND setz=true AND zaehler>=3 THEN setz=fal
  se:zaehler=0:GOTO 1190
1400 IF untbot=1 AND setz=true AND zaehler [9459]
  <2 THEN setz=false:GOTO 1190 ELSE IF untbo
  t =1 AND setz=true AND zaehler>=2 THEN set
  z=false:zaehler=0:GOSUB 5940:GOTO 1680
1410 ok=false:GOTO 1280 [1648]
1420 '***** NEUES SPIEL ***** [1531]
1430 GOSUB 3300 [965]
1440 GOTO 60 [390]
1450 '***** ENDE ***** [432]
1460 GOSUB 3300:INK 0,1:INK 1,24 [2253]
1470 BORDER 1:PEN 1:PAPER 0:LOCATE 1,1:NEW [1870]
1480 '*****DRUCKEN***** [1541]
1490 alles=false:GOTO 1510 [1403]
1500 alles=true [927]
1510 'IF sende=false THEN GOTO 660 [1001]
1520 taste$="":PAPER#4,0:CLS#4:PEN#4,1:LOC [6538]
  ATE#4,1,2:PRINT#4,"Drucken(j/n)?"
1530 WHILE taste$="":taste$=INKEY$:WEND [2089]
1540 IF taste$<>"j" THEN CLS#4:GOTO 670 [2611]
1550 CLS#4:PRINT#8,norm$;esc$;"W1";" [4766]
  Subunk";esc$;"W0":PRINT#8:PRINT#8
1560 PRINT#8," Eigene [3707]
  Schiffe des"
1570 PRINT#8," Schiffe [3992]
  Gegners":PRINT#8,klein$
1580 FOR y=1 TO 10 [1115]
1590 zaehler=0:l=0 [982]
1600 PRINT#8," "; [789]
1610 FOR x=1 TO 10 [803]
1620 IF alles=false THEN GOSUB 3440 ELSE G [2413]
  OSUB 3460
1630 NEXT x [356]
1640 zaehler=zaehler+1 [2242]
1650 IF zaehler=1 THEN l=1:GOTO 1600 ELSE [3139]
  PRINT#8," ":NEXT y
1660 GOTO 670 [431]
1670 ' [117]
1680 '***** SETZEN DER SCHIFFE DES COM [2608]
  PUTERS *****
1690 ' [117]
1700 EVERY 120 GOSUB 6520:laenge=5 [1032]
1710 fehler=false [1020]
1720 boot=1 [593]
1730 l=1:n=3:zaehler=1 [1197]
1740 CLS#5:LOCATE#5,2,1:PEN#5,2:PRINT#5,"D [2768]
  ER COMPUTER"
1750 LOCATE#5,6,2:PRINT#5,"SETZT" [1096]
1760 LOCATE#5,1,3:PRINT#5,"SEINE SCHIFFE!" [1345]
1770 RANDOMIZE TIME:IF boot>4 THEN GOTO 18 [5502]
  60 ELSE px=INT(RND*9)+1:py=INT(RND*9)+1
1780 zuff=ROUND(RND):IF zuff=0 THEN senk=0 [5766]
  :waag=1 ELSE IF zuff=1 THEN senk=1:waag=0
1790 IF senk=0 AND (px+laenge)-1<=10 THEN [4917]
  xende=px+laenge-1:yende=py ELSE IF senk=0
  AND (px+laenge)-1>10 THEN GOTO 1770
1800 IF senk=1 AND (py+laenge)-1<=10 THEN [7187]
  yende=py+laenge-1:xende=px ELSE IF senk=1
  AND (py+laenge)-1>10 THEN GOTO 1770
1810 ' [117]
1820 IF boot=1 THEN kreuzer=1:untbot=0:lae [3445]
  nge=5:GOSUB 2430:GOTO 1770
1830 IF boot=2 THEN kreuzer=0:zerst=1:laen [2742]
  ge=4 :GOSUB 2430:GOTO 1770
1840 IF boot=3 THEN zerst=0 :torp=1:laenge [3611]
  =3 :GOSUB 2430:GOTO 1770
1850 IF boot=4 THEN torp=0:untbot=1:laenge [2793]
  =2 :GOSUB 2430:GOTO 1770
1860 CLS#5:PEN#5,3 [884]

```

Listing: Sub Sunk

```

1870 zuffi=ROUND(RND) [1140]
1880 IF zuffi=1 THEN LOCATE#5,4,1:PRINT#5, [3660]
  "DER COPUTER BEGINNT!";
1890 IF zuffi=0 THEN LOCATE#5,1,1:PRINT#5, [3883]
  "BITTE BEGINNEN SIE!";
1900 FOR a=1 TO 1800:NEXT:CLS#5 [1551]
1910 ' [117]
1920 '***** SPIEL SYSTEM DES COP [2322]
  UTERS *****
1930 x=1:y=1 [850]
1940 dummy=REMAIN(0) [623]
1950 zaehl=0 [651]
1960 treffer=0 [1094]
1970 LOCATE#3,1,1:PEN#3,3:PRINT#3,transein [3717]
  $;CHR$(226);
1980 xc=1:yc=1 [1008]
1990 zaehli=0:zaehlen=0 [1138]
2000 IF zuffi=0 THEN gezielt=true:ziel=fal [6290]
  se:GOSUB 4140 ELSE gezielt=false:ziel=true
2010 IF setzen=false THEN GOTO 670 [2211]
2020 IF taste=true AND gezielt=true THEN t [7410]
  aste=false ELSE IF taste=true AND ziel=tru
  e THEN taste=false:FOR a=1 TO 200:NEXT a:G
  OSUB 4140
2030 IF taste=true THEN GOTO 670 [1544]
2040 IF sende=true GOTO 670 [1931]
2050 IF zaehli=22 THEN sende=true:GOSUB 33 [2852]
  50:GOTO 670
2060 MOVE 416,392:DRAWR 208,0,0:DRAWR 0,-5 [5156]
  2:DRAWR -208,0:DRAWR 0,52
2070 l=0:n=2:tests=false:ax=lx:ay=ly:waag= [4056]
  merke:IF treffer=0 THEN xc=ROUND(RND*9)+1:
  yc=ROUND(RND*9)+1
2080 ' [117]
2090 IF xfeld(xc,yc,0)=236 AND treffer=0 T [2640]
  HEN GOTO 2030
2100 IF xfeld(xc,yc,0)=234 AND treffer=0 T [3935]
  HEN GOTO 2030
2110 IF xfeld(xc,yc,0)=223 AND treffer=0 T [3246]
  HEN s=yc:w=xc:GOSUB 3240:GOSUB 3130
2120 IF xfeld(xc,yc,0)=223 AND treffer=0 A [9943]
  ND voll=0 AND wasser=false THEN GOSUB 2960
  :GOSUB 2970:gezielt=false:ziel=true:GOSUB
  3110:GOSUB 4140:GOTO 2030
2130 ' [117]
2140 IF xfeld(xc,yc,0)=223 AND treffer=0 A [6555]
  ND (voll<>0 OR wasser=true) THEN voll=0:wa
  sser=false:GOTO 2030
2150 IF (xfeld(xc,yc,0)>=259 AND xfeld(xc, [9694]
  yc,0)<=367) AND treffer=0 THEN xfeld(xc,yc
  ,0)=236:GOSUB 3190:s=yc:w=xc:GOSUB 2940:za
  ehlen=zaehlen+1:GOSUB 3040:GOSUB 3270:GOSU
  B 4140:GOTO 2030
2160 ' [117]
2170 IF treffer=1 THEN zuff=ROUND(RND):xoy [2537]
  =ROUND(RND)
2180 IF zuff=0 AND treffer=1 AND xoy=0 AND [3802]
  xfeld(xc,yc,0)<>0 THEN xsuch=xc-1:ysuch=y
  c
2190 IF zuff=1 AND treffer=1 AND xoy=0 AND [4029]
  xfeld(xc,yc,0)<>0 THEN xsuch=xc+1:ysuch=y
  c
2200 IF zuff=0 AND treffer=1 AND xoy=1 AND [4062]
  xfeld(xc,yc,0)<>0 THEN xsuch=xc:ysuch=yc-
  1
2210 IF zuff=1 AND treffer=1 AND xoy=1 AND [4392]
  xfeld(xc,yc,0)<>0 THEN xsuch=xc:ysuch=yc+
  1
2220 IF treffer=1 AND (xfeld(xsuch,ysuch,0 [5767]
  )=0 OR xfeld(xsuch,ysuch,0)=234) THEN GOTO
  2170
2230 IF xfeld(xsuch,ysuch,0)=223 AND treff [5592]

```

Listing: Sub Sunk


```

er=1 THEN w=xsuch:s=ysuch:GOSUB 3130
2240 IF xfeld(xsuch,ysuch,0)=223 AND treff [8226]
er=1 AND voll=1 THEN GOSUB 3280:GOSUB 2960
:GOSUB 2970:l=1:n=3:lx=ax:ly=ay:GOSUB 4140
:GOTO 2030
2250 IF xfeld(xsuch,ysuch,0)=223 AND treff [4599]
er=1 AND voll>1 THEN voll=0:GOTO 2170
2260 ' [117]
2270 IF (xfeld(xsuch,ysuch,0)>=259 AND xfe [15736]
ld(xsuch,ysuch,0)<=367) AND treffer=1 THEN
GOSUB 3290:GOSUB 3190:GOSUB 2940:zaehlen=
zaehlen+1:GOSUB 3040:lx=ax:ly=ay:GOSUB 269
0:treffer=2:merke=waag:l=1:n=3:GOSUB 4140:
GOTO 2030
2280 IF treffer=2 AND versunken=true THEN [4311]
treffer=0:versunken=false:GOTO 2030
2290 ' [117]
2300 IF merke=true AND xfeld(xc,yc,0)<>234 [4306]
THEN xc=xc+1 ELSE IF merke=false AND xfel
d(xc,yc,0)<>234 THEN yc=yc+1
2310 IF xfeld(xc,yc,0)=234 THEN GOTO 2380 [2533]
2320 IF xfeld(xc,yc,0)=236 THEN GOTO 2300 [2049]
2330 IF xfeld(xc,yc,0)=223 THEN s=yc:w=xc: [2860]
GOSUB 3130
2340 IF xfeld(xc,yc,0)=223 AND voll=1 THEN [9725]
voll=0 :xfeld(xc,yc,0)=234:w=xc:s=yc:GOSU
B 2960:GOSUB 2970:l=1:n=3:lx=ax:ly=ay:gezi
elt=false:ziel=true:GOSUB 4140:GOTO 2030
2350 IF xfeld(xc,yc,0)=223 AND voll>1 THEN [3172]
voll=0:GOTO 2380
2360 IF (xfeld(xc,yc,0)>=259 AND xfeld(xc, [8185]
yc,0)<=367) THEN GOSUB 3190:s=yc:w=xc:GOSU
B 2940:xfeld(xc,yc,0)=236:zaehlen=zaehlen+
1:GOSUB 3040:GOSUB 2690:GOTO 2410
2370 ' [117]
2380 IF waag=true THEN xc=xc-1 ELSE yc=yc- [2665]
1
2390 IF xfeld(xc,yc,0)=236 THEN GOTO 2380 [1969]
2400 IF (xfeld(xc,yc,0)>=259 AND xfeld(xc, [7731]
yc,0)<=367) THEN GOSUB 3190:s=yc:w=xc:GOSU
B 2940:xfeld(xc,yc,0)=236:zaehlen=zaehlen+
1:GOSUB 3040:GOSUB 2690
2410 IF versunken=true THEN versunken=fals [9138]
e:treffer=0:l=1:n=3:GOSUB 4140:GOTO 2030
ELSE l=1:n=3:lx=ax:ly=ay:merke=waag:geziel
t=false:ziel=true:GOSUB 4140:GOTO 2030
2420 END [110]
2430 IF senk=0 THEN GOSUB 5240 ELSE GOSUB [1797]
2620
2440 IF fehler=true THEN RETURN ELSE GOSUB [1993]
5260
2450 IF fehler=false THEN GOSUB 5380 ELSE [3054]
IF fehler=true THEN RETURN
2460 RETURN [555]
2470 ' [117]
2480 '***** JOYSTICK ABFRAGE FUER SPIE [2168]
LFELD *****
2490 ' [117]
2500 tests=false:waag=merken:ax=sx:ay=sy:I [11449]
F xfeld(altx,alty,1)=223 OR (xfeld(altx,al
ty,1)>=259 AND xfeld(altx,alty,1)<=367) TH
EN PEN#3,2:LOCATE#3,altx,alty:PRINT#3,CHR$
(223);
2510 IF xfeld(altx,alty,1)=234 THEN s=alty [3962]
:w=altx:GOSUB 2960
2520 IF j=16 AND xfeld(x,y,1)=223 THEN xf [12956]
eld(x,y,1)=234:w=x:s=y:GOSUB 2960:GOSUB 29
70:PEN#5,2:CLS#5:LOCATE#5,5,3:PRINT#5,"WAS
SER";:FOR a=1 TO 1100:NEXT:CLS#5:n=2:l=0:b
en=true:gezielt=true:ziel=false:GOTO 2600
2530 ' [117]
2540 IF j=16 AND (xfeld(x,y,1)>=259 AND xf [14278]

```

Listing: Sub Sunk

```

eld(x,y,1)<=367) THEN GOSUB 3190:s=y:w=x:G
OSUB 2940:xfeld(x,y,1)=236:ax=x:ay=y:tests
=false:GOSUB 2930:zaehli=zaehli+1:GOSUB 30
60:GOSUB 2690:GOSUB 3260:gezielt=true:ziel
=false:GOTO 2600
2550 ' [117]
2560 IF xfeld(altx,alty,1)=236 THEN s=alty [2897]
:w=altx:GOSUB 2940
2570 LOCATE#3,x,y:PEN#3,3:PRINT#3,transein [6401]
$;:LOCATE#3,x,y:PRINT#3,CHR$(226);:LOCATE#
3,x,y:PRINT#3,transaus$;
2580 setzen=true [846]
2590 sx=ax:sy=ay [356]
2600 RETURN [555]
2610 IF (xfeld(x,y,0)>=259 AND xfeld(x,y,0 [7308]
)<=367) THEN geschossen=true:s=y:w=x:GOSUB
2940:xfeld(x,y,0)=236:GOSUB 2690:RETURN
2620 IF senk=1 THEN GOSUB 5490 [554]
2630 IF fehler=true THEN RETURN ELSE GOSUB [1993]
5260
2640 IF fehler=true THEN RETURN ELSE GOSUB [2043]
5530
2650 RETURN [555]
2660 ' [117]
2670 '***** ERKENNEN VON SCHIFFEN ** [1776]
*****
2680 ' [117]
2690 IF (xfeld(ax+1,ay,1)>=259 AND xfeld(a [10418]
x+1,ay,1)<=367) OR xfeld(ax+1,ay,1)=236 OR
(xfeld(ax-1,ay,1)>=259 AND xfeld(ax-1,ay,
1)<=367) OR xfeld(ax-1,ay,1)=236 THEN waag
=true:senk=false
2700 IF (xfeld(ax,ay+1,1)>=259 AND xfeld(a [13584]
x,ay+1,1)<=367) OR xfeld(ax,ay+1,1)=236 OR
(xfeld(ax,ay-1,1)>=259 AND xfeld(ax,ay-1,
1)<=367) OR xfeld(ax,ay-1,1)=236 THEN senk
=true:waag=false
2710 ' [117]
2720 WHILE tests =false [1125]
2730 IF waag=true THEN ax=ax+1:GOSUB 2860 [4377]
ELSE ay=ay+1:GOSUB 2860
2740 IF tests=true THEN GOTO 2820 [2588]
2750 IF xfeld(ax,ay,1)=236 THEN GOTO 2730 [2348]
2760 IF xfeld(ax,ay,1)=223 OR xfeld(ax,ay, [5979]
1)=234 OR xfeld(ax,ay,1)=0 THEN GOTO 2770
2770 IF waag=true THEN ax=ax-1 ELSE ay=ay- [2420]
1
2780 GOSUB 2860:IF tests=true THEN 2820 [2030]
2790 IF xfeld(ax,ay,1)=236 THEN GOTO 2770 [2340]
2800 IF l=1 THEN GOSUB 2840:GOTO 2820 [1999]
2810 IF l=0 THEN GOSUB 2850 [725]
2820 WEND [390]
2830 RETURN [555]
2840 IF xfeld(ax,ay,1)=223 OR xfeld(ax,ay, [6342]
1)=234 OR xfeld(ax,ay,1)=0 THEN tests=true
:GOSUB 2910:RETURN
2850 IF xfeld(ax,ay,0)=223 OR xfeld(ax,ay, [9763]
0)=234 OR xfeld(ax,ay,0)=0 THEN tests=true
:versunken=true:GOSUB 2990:RETURN
2860 IF (xfeld(ax,ay,1)>=259 AND xfeld(ax, [4519]
ay,1)<=367) THEN tests=true
2870 RETURN [555]
2880 ' [117]
2890 '***** MELDUNGEN FUER DEN BENU [2461]
TZER *****
2900 ' [117]
2910 PAPER#5,0:CLS#5:LOCATE#5,1,3:PEN#5,3: [7523]
PRINT#5,"BOOT VERSUNKEN";:FOR a=1 TO 1800:
NEXT a:CLS#5:RETURN
2920 RETURN [555]
2930 CLS#5:LOCATE#5,5,3:PEN#5,3:PRINT#5,"T [5102]
REFFER";:FOR a=1 TO 1100:NEXT:CLS#5:RETURN

```

Listing: Sub Sunk

```

2940 LOCATE#n,w,s:PEN#n,1:PRINT#n,CHR$(236 [9892]
);:LOCATE#n,w,s:PRINT#n,transein$;:LOCATE#
n,w,s:PEN#n,2:PRINT#n,CHR$(237);:LOCATE#n,
w,s:PEN#n,3
2950 PRINT#n,CHR$(238);:LOCATE#n,w,s:PRINT [3924]
#n,transaus$;:RETURN
2960 PEN#n,0:LOCATE#n,w,s:PRINT#n,CHR$(239 [5637]
);:LOCATE#n,w,s:PEN#n,2:PRINT#n,CHR$(234);
:RETURN
2970 ENV 1,2,5,1,1,0,3,10,-1,6:SOUND 1,0,0 [2677]
,0,1,0,1:RETURN
2980 LOCATE#n,w,s:PRINT#n,CHR$(234);:RETUR [3594]
N
2990 PAPER#4,0:CLS#4:LOCATE#4,3,1:PEN#4,3: [3975]
PRINT#4,"IHR BOOT"
3000 LOCATE#4,1,2:PRINT#4,"IST VERSENKT" [1715]
3010 LOCATE#4,3,3:PRINT#4,"WORDEN!":FOR a= [2987]
1 TO 1600:NEXT:CLS#4:GOSUB 3090
3020 RETURN [555]
3030 ' [117]
3040 IF zaehlen <=11 THEN LOCATE 12+zaehle [4476]
n,6:PEN 3:PRINT CHR$(252);:RETURN
3050 IF zaehlen>11 THEN LOCATE 12+zaehlen- [3775]
11,7:PEN 3:PRINT CHR$(252);:RETURN
3060 IF zaehli<=11 THEN LOCATE 26+zaehli,6 [4213]
:PEN 3:PRINT CHR$(252);:RETURN
3070 IF zaehli>11 THEN LOCATE 26+zaehli-11 [2879]
,7:PEN 3:PRINT CHR$(252);:RETURN
3080 ' [117]
3090 MOVE 176,64:DRAWR 202,-48,1:MOVE 176, [3384]
16:DRAWR 202,48:RETURN
3100 MOVE 400,64:DRAWR 218,-48,1:MOVE 400, [3988]
16:DRAWR 218,48:RETURN
3110 xfeld(xc,yc,0)=234:l=1:n=3:RETURN [1424]
3120 ' [117]
3130 FOR a=s-1 TO s+1 [1162]
3140 FOR b=w-1 TO w+1 [842]
3150 IF xfeld(b,a,0)=236 THEN voll=voll+1 [2799]
3160 NEXT b [378]
3170 NEXT a [383]
3180 RETURN [555]
3190 ' [117]
3200 '**** KNALL *** [505]
3210 ENV 1,1,15,1,2,-1,2,13,-1,15 [1469]
3220 SOUND 1,2400,10,0,1,0,15:SOUND 1,2400 [2136]
,0,0,1,0,10
3230 RETURN [555]
3240 IF (xfeld(w+1,s,0)=234 OR xfeld(w+1,s [10098]
,0)=0) AND (xfeld(w-1,s,0)=234 OR xfeld(w-
1,s,0)=0) AND (xfeld(w,s-1,0)=234 OR xfeld
(w,s-1,0)=0) AND (xfeld(w,s+1,0)=234 OR xf
eld(w,s+1,0)=0) THEN wasser=true:RETURN
3250 ' [117]
3260 sx=ax:sy=ay:merken=waag:l=0:n=2:ben=t [3142]
rue:RETURN
3270 ax=xc:ay=yc:treffer=1:l=1:n=3:lx=ax:l [6226]
y=ay:gezielt=false:ziel=true:RETURN
3280 voll=0:xfeld(xsuch,ysuch,0)=234:s=ysu [8578]
ch=w:xsuch:gezielt=false:ziel=true:RETURN
3290 xfeld(xsuch,ysuch,0)=236:s=ysuch:w=xs [5086]
uch:gezielt=false:ziel=true:RETURN
3300 ENT 1,50,4,1:ENV 1,55,3,1:SOUND 1,10, [3072]
0,1,1,1:FOR a=1 TO 25
3310 LOCATE 40,25:PRINT " "; [976]
3320 NEXT a [383]
3330 ENV 1,10,3,3:SOUND 1,0,0,0,1,0,5 [2185]
3340 RETURN [555]
3350 FOR b=1 TO 5 [1133]
3360 PEN#5,3:LOCATE#5,1,3:PRINT#5," HERZL [3913]
ICHEN GLUECKWUNSCH!";:FOR a=1 TO 400:NEX
T a
3370 FOR a=1 TO 5 [787]

```

Listing: Sub Sunk

```

3380 LOCATE#5,14,4:PRINT#5," ";:NEXT [1412]
3390 NEXT:h1=1:GOSUB 6290:RETURN [2589]
3400 CLS#5:PEN#5,3:PRINT#5," VIELLEICHT [7614]
KLAPPT'S BEIM NAECHSTEN MAL!";:FOR a=1 TO
1500:NEXT:h1=1:GOSUB 6290:RETURN
3410 ' [117]
3420 '***** UNTERPROGRAMM FUER DRUCKEN [2788]
*****
3430 ' [117]
3440 IF xfeld(x,y,1)=223 OR xfeld(x,y,1)=2 [5332]
34 THEN PRINT#8,CHR$(126);:RETURN
3450 IF xfeld(x,y,1)=236 OR (xfeld(x,y,1)> [5782]
=259 AND xfeld(x,y,1)<=367) THEN PRINT#8,"
O";:RETURN
3460 IF xfeld(x,y,1)=223 THEN PRINT#8,CHR$ [3538]
(126);:RETURN
3470 IF xfeld(x,y,1)=234 THEN PRINT#8,"-"; [2593]
:RETURN
3480 IF xfeld(x,y,1)=236 THEN PRINT#8,"#"; [3229]
:RETURN
3490 IF (xfeld(x,y,1)>=259 AND xfeld(x,y,1 [4051]
)<=367) THEN PRINT#8,"O";:RETURN
3500 ' [117]
3510 '***** BILDSCHIRM AUFBAU **** [1459]
*****
3520 ' [117]
3530 WINDOW#1,1,10,1,25 [1307]
3540 WINDOW#2,14,23,9,18 [1750]
3550 WINDOW#3,29,38,9,18 [1414]
3560 WINDOW#4,12,24,21,24 [1665]
3570 WINDOW#5,26,39,21,24 [1468]
3580 ORIGIN 0,0:DRAWR 639,0,1:DRAWR 0,399: [4108]
DRAWR -639,0:MOVE 0,0:DRAWR 0,399,1
3590 MOVE 160,399:DRAWR 0,-399,1 [1044]
3600 MOVE 160,88:DRAWR 480,0 [2276]
3610 MOVE 396,399:DRAWR 0,-399 [1152]
3620 RETURN [555]
3630 '***** [665]
3640 PEN#1,1:LOCATE#1,2,kastn:PRINT#1,CHR$ [2352]
(214);
3650 PRINT#1,STRING$(5,CHR$(233)); [1425]
3660 PRINT#1,CHR$(212); [1024]
3670 LOCATE#1,8,kastn:PEN#1,2:PRINT#1,CHR$ [1949]
(22);CHR$(1);CHR$(214);
3680 PRINT#1,CHR$(22);CHR$(0); [1000]
3690 LOCATE#1,8,kastn+1:PRINT#1,CHR$(233); [1609]
3700 RETURN [555]
3710 '*** schrift *** [776]
3720 PEN#1,3:LOCATE#1,2,2:PRINT#1,"BASIC"; [2796]
3730 PEN#1,2:LOCATE#1,2,6:PRINT#1,"NEUES" [1239]
3740 LOCATE#1,2,7:PRINT#1,"SPIEL" [1516]
3750 LOCATE#1,2,10:PRINT#1,"SPIEL-" [2335]
3760 LOCATE#1,2,11:PRINT#1,"FELD" [1597]
3770 LOCATE#1,2,14:PRINT#1,"SETZ-" [1877]
3780 LOCATE#1,2,15:PRINT#1,"FELD" [1806]
3790 LOCATE#1,2,18:PRINT#1,"START-" [1057]
3800 LOCATE#1,2,19:PRINT#1,"POS." [1690]
3810 LOCATE#1,2,20:PRINT#1,"DRUCK." [2000]
3820 LOCATE#1,2,23:PRINT#1,"END-" [2063]
3830 LOCATE#1,2,24:PRINT#1,"STAND" [2016]
3840 LOCATE#1,2,25:PRINT#1,"DRUCK." [2025]
3850 LOCATE 13,5:PEN 2:PRINT#1,"TREFFER : " [2192]
3860 LOCATE 27,5:PEN 2:PRINT#1,"TREFFER : " [2029]
3870 LOCATE 13,2:PEN 1:PRINT#1,"EIGENE" [1835]
3880 LOCATE 17,3:PEN 1:PRINT#1,"SCHIFFE" [1874]
3890 LOCATE 28,2:PEN 1:PRINT#1,"SCHIFFE DES" [3012]
3900 LOCATE 30,3:PEN 1:PRINT#1,"GEGNERS" [1246]
3910 RETURN [555]
3920 PRINT CHR$(zahl) [1180]
3930 RETURN [555]
3940 PRINT CHR$(bustab); [1413]
3950 RETURN [555]

```

Listing: Sub Sunk

```

3960 '*** UMRAMUNG der SPIELFELDER*** [3076]
3970 ' [117]
3980 LOCATE x,y:PRINT CHR$(214);STRING$(9, [2809]
CHR$(233));
3990 PRINT CHR$(212);CHR$(22);CHR$(1); [2090]
4000 LOCATE x+10,y:PRINT CHR$(222);CHR$(22 [4059]
);CHR$(0)
4010 LOCATE x+10,y+1:FOR i=y+1 TO y+9 [1859]
4020 LOCATE x+10,i [981]
4030 PRINT CHR$(207):NEXT [1131]
4040 LOCATE x+10,y+10:PRINT CHR$(220) [1568]
4050 RETURN [555]
4060 '*** MALEN DES MENU FEILES *** [2802]
4070 LOCATE#1,9,alty:PRINT#1," "; [1426]
4080 PEN#1,1:LOCATE#1,9,sty:PRINT#1,CHR$(2 [2587]
24);CHR$(225);
4090 alty=sty [211]
4100 RETURN [555]
4110 ' [117]
4120 '***** STEUERUNG DES CURSORS ** [1834]
*****
4130 ' [117]
4140 ben=false:altx=x:alty=y [2263]
4150 IF setzen=true THEN MOVE 416,392:DRAW [3696]
R 208,0,3:DRAWR 0,-52:DRAWR -208,0:DRAWR 0
,52
4160 IF zaehlen=22 AND setzen=true THEN se [4069]
nde=true:GOSUB 3400:RETURN
4170 j=0:WHILE j=0:j=JOY(0):WEND [2274]
4180 taste$=INKEY$:IF taste$=" " AND setze [6115]
n=true THEN taste=true:RETURN
4190 IF j AND 4 THEN x=x-1 [1032]
4200 IF j AND 8 THEN x=x+1 [210]
4210 IF j AND 1 THEN y=y-1 [1530]
4220 IF j AND 2 THEN y=y+1 [1012]
4230 IF x<1 THEN x=1 [512]
4240 IF x>10 THEN x=10 [512]
4250 IF y<1 THEN y=1 [770]
4260 IF y>10 THEN y=10 [900]
4270 IF setzen=false THEN GOSUB 5750 ELSE [2484]
GOSUB 2500
4280 altx=x:alty=y [244]
4290 IF ben=true THEN RETURN ELSE IF ben=f [4811]
alse AND setzen=true GOTO 4170
4300 IF setzen=false THEN RETURN [2320]
4310 ' [117]
4320 '***** SCHIFFE ***** [853]
4330 ' [117]
4340 SYMBOL AFTER 129 [1443]
4350 '***** KREUZER ***** [1560]
4360 SYMBOL 129,0,0,31,251,31,0,0,0 [1952]
4370 SYMBOL 130,0,0,0,4,0,0,0,0 [2076]
4380 SYMBOL 131,0,255,6,199,6,255,0,0 [1860]
4390 SYMBOL 132,0,0,249,56,249,0,0,0 [1494]
4400 SYMBOL 133,15,251,24,152,24,251,15,0 [1635]
4410 SYMBOL 134,0,4,231,103,231,4,0,0 [2204]
4420 SYMBOL 135,231,187,150,2,162,31,243,0 [2500]
4430 SYMBOL 136,24,68,105,253,93,224,12,0 [2083]
4440 SYMBOL 137,0,192,24,116,24,192,0,0 [1908]
4450 SYMBOL 138,0,0,224,128,224,0,0,0 [1595]
4460 '***** [768]
4470 SYMBOL 139,8,8,8,28,28,20,28,28 [1968]
4480 SYMBOL 140,0,0,0,0,0,8,0,0 [1365]
4490 SYMBOL 141,42,42,34,34,34,62,62,42 [2168]
4500 SYMBOL 142,20,20,28,28,28,0,0,20 [2134]
4510 SYMBOL 143,42,34,34,62,127,65,99,99 [2302]
4520 SYMBOL 144,20,28,28,0,0,62,28,28 [2244]
4530 SYMBOL 145,87,65,83,102,34,37,127,99 [2357]
4540 SYMBOL 146,40,62,44,25,93,90,0,28 [2121]
4550 SYMBOL 147,34,42,8,28,20,8,0,0 [1443]
4560 SYMBOL 148,28,20,20,0,0,0,0,0 [1755]
4570 '***** ZERSTOERER ***** [1653]

```

Listing: Sub Sunk

```

4580 SYMBOL 149,0,15,61,241,61,15,0,0 [1540]
4590 SYMBOL 150,0,0,2,14,2,0,0,0 [1538]
4600 SYMBOL 151,15,175,160,0,224,239,15,0 [2577]
4610 SYMBOL 152,0,80,95,254,31,16,0,0 [2078]
4620 SYMBOL 153,254,38,140,7,128,39,248,0 [1998]
4630 SYMBOL 154,1,217,115,248,127,216,0,0 [1989]
4640 SYMBOL 155,0,248,110,227,238,184,0,0 [1751]
4650 SYMBOL 156,0,0,144,28,16,64,0,0 [1496]
4660 '***** [768]
4670 SYMBOL 157,8,8,28,28,54,54,34,62 [1521]
4680 SYMBOL 158,0,0,0,0,8,8,28,0 [1695]
4690 SYMBOL 159,54,48,54,0,99,99,99,99 [1812]
4700 SYMBOL 160,8,14,8,62,28,28,28,28 [1854]
4710 SYMBOL 161,85,65,99,65,69,46,42,40 [2326]
4720 SYMBOL 162,42,62,28,62,58,16,20,23 [2330]
4730 SYMBOL 163,58,62,62,34,54,20,28,8 [1607]
4740 SYMBOL 164,4,32,0,28,8,0,0,0 [2145]
4750 '***** U-BOOT ***** [1121]
4760 SYMBOL 165,0,3,30,244,30,3,0,0 [1643]
4770 SYMBOL 166,0,0,1,11,1,0,0,0 [1898]
4780 SYMBOL 167,0,224,56,254,56,224,0,0 [2055]
4790 SYMBOL 168,0,0,192,0,192,0,0,0 [1689]
4800 '***** [1416]
4810 SYMBOL 169,16,16,16,56,40,56,108,68 [2203]
4820 SYMBOL 170,0,0,0,0,16,0,16,56 [1841]
4830 SYMBOL 171,84,84,124,56,56,16,16,0 [2229]
4840 SYMBOL 172,40,40,0,0,0,0,0,0 [1563]
4850 '***** TORPEDOBOOT ***** [2228]
4860 SYMBOL 173,0,7,54,196,54,7,0,0 [2097]
4870 SYMBOL 174,0,0,9,59,9,0,0,0 [1608]
4880 SYMBOL 175,0,140,121,1,127,140,0,0 [1993]
4890 SYMBOL 176,132,115,134,254,128,115,13 [2517]
2,0
4900 SYMBOL 177,0,112,208,192,208,112,0,0 [2081]
4910 SYMBOL 178,0,128,32,56,32,128,0,0 [2730]
4920 '***** [1416]
4930 SYMBOL 179,8,8,20,20,0,62,54,34 [1461]
4940 SYMBOL 180,0,0,8,8,28,0,8,28 [1815]
4950 SYMBOL 181,34,20,20,20,54,50,16,28 [1806]
4960 SYMBOL 182,93,42,42,42,8,77,46,34 [1589]
4970 SYMBOL 183,28,62,34,54,0,0,0,0 [1631]
4980 SYMBOL 184,34,0,28,8,8,0,0,0 [1952]
4990 '*** SONSTIGE SYMBOLS *** [1239]
5000 SYMBOL 223,254,254,254,254,254,25 [1982]
4,0
5010 SYMBOL 224,2,12,56,247,56,12,2,0 [1841]
5020 SYMBOL 225,0,0,0,240,0,0,0,0 [1595]
5030 SYMBOL 226,56,16,130,214,130,16,56,0 [1874]
5040 SYMBOL 227,4,126,126,100,96,240,96,0 [1787]
5050 SYMBOL 228,250,0,0,26,30,14,30,0 [2226]
5060 SYMBOL 229,254,222,158,6,6,158,222,0 [2126]
5070 SYMBOL 230,0,0,64,248,248,64,0,0 [2044]
5080 SYMBOL 231,253,230,194,128,230,230,23 [2904]
0,0
5090 SYMBOL 232,0,24,60,126,24,24,24,0 [1622]
5100 SYMBOL 233,255,255,255,255,255,25 [2673]
5,255
5110 SYMBOL 234,108,170,214,56,214,170,108 [2176]
,0
5120 SYMBOL 236,20,4,12,16,192,160,62,0 [1598]
5130 SYMBOL 237,96,160,32,130,0,64,128,0 [1800]
5140 SYMBOL 238,138,90,210,100,54,2,0,0 [1959]
5150 SYMBOL 239,146,84,40,198,40,84,146,0 [2342]
5160 RETURN [555]
5170 ' [117]
5180 '***** UEBERPRUEFEN DER UMGEBUNG U. [3925]
DER LAGE DER SCHIFFE *****
5190 ' [117]
5200 px=x:py=y:punkt=true:PEN#2,3:LOCATE#2 [8001]
,px,py:PEN#2,2:PRINT#2,CHR$(228);:LOCATE#2
,px,py:PEN#2,3:PRINT#2,transaus$;:LOCATE#2
,px,py:PRINT#2,CHR$(227);transaus$;:RETURN

```

Listing: Sub Sunk

```

5210 punkt=false:LOCATE#2,px,py:PEN#2,2:PR [5782]
INT#2,CHR$(223);:px=0:py=0:RETURN
5220 IF (px+laenge)-1>10 THEN fehler=true: [4781]
GOSUB 5850:RETURN ELSE fehler=false:GOSUB
5850:RETURN
5230 ' [117]
5240 IF setzen=true AND (px+laenge)-1>10 [8811]
THEN fehler=true:RETURN ELSE IF setzen=tru
e AND (px+laenge)-1<=10 THEN fehler=false:
RETURN
5250 ' [117]
5260 FOR xanfng=px-1 TO xende+1 [2802]
5270 FOR yanfang=py-1 TO yende+1 [2235]
5280 IF xfeld(xanfng,yanfang,1)<>0 AND xf [4478]
eld(xanfng,yanfang,1)<>223 THEN fehler=tr
ue
5290 NEXT yanfang [444]
5300 NEXT xanfng [359]
5310 ' [117]
5320 IF fehler=true AND setzen=false THEN [10696]
LOCATE#4,1,4:PEN#4,3:PRINT#4,"SCHON BELEGT
!";:FOR a=1 TO 1500:NEXT:LOCATE#4,1,4:PEN#
4,0:PRINT#4,STRING$(13,CHR$(233));:RETURN
5330 IF setzen=true AND fehler=true THEN R. [2975]
ETURN ELSE IF setzen=true AND fehler=false
THEN RETURN
5340 ok=true:RETURN [1103]
5350 ' [117]
5360 '***** MALEN DER WAAGRECHTEN SC [2408]
HIFFE *****
5370 ' [117]
5380 IF kreuzer=1 THEN xfeld(px,py,1)=259: [12790]
xfeld(px+1,py,1)=263:xfeld(px+2,py,1)=267:
xfeld(px+3,py,1)=271:xfeld(px+4,py,1)=275:
GOSUB 5700:x$=kreuzer$:y$=transkreuzer$:GO
SUB 5630:GOSUB 5460:setz=true:RETURN
5390 ' [117]
5400 IF zuerst=1 THEN xfeld(px,py,1)=299:xf [11112]
eld(px+1,py,1)=303:xfeld(px+2,py,1)=307:xf
eld(px+3,py,1)=311:GOSUB 5700:x$=zuerst$:y$
=transzuerst$:GOSUB 5630:GOSUB 5460:setz=tr
ue:RETURN
5410 ' [117]
5420 IF torp=1 THEN xfeld(px,py,1)=347:xfe [12127]
ld(px+1,py,1)=351:xfeld(px+2,py,1)=355: GO
SUB 5700:x$=torp$:y$=transtorp$:GOSUB 5630
:GOSUB 5890:setz=true:RETURN
5430 ' [117]
5440 IF untbot=1 THEN xfeld(px,py,1)=331:x [9008]
feld(px+1,py,1)=335:GOSUB 5700:x$=uboot$:y
$=transuboot$:GOSUB 5630:GOSUB 5910:setz=t
rue:RETURN
5450 ' [117]
5460 IF setzen=false THEN ok=false:RETURN [5410]
ELSE boot=boot+1:RETURN
5470 ' [117]
5480 IF (py+laenge)-1<=10 THEN fehler=fals [6207]
e:RETURN ELSE IF (py+laenge)-1>10 THEN feh
ler=true:RETURN
5490 IF (py+laenge)-1<=10 THEN fehler=fals [6207]
e:RETURN ELSE IF (py+laenge)-1>10 THEN feh
ler=true:RETURN
5500 ' [117]
5510 '***** MALEN DER SENKRECHTEN SCHIFF [2354]
E *****
5520 ' [117]
5530 IF kreuzer=1 THEN xfeld(px,py,1)=279: [12548]
xfeld(px,py+1,1)=283:xfeld(px,py+2,1)=287:
xfeld(px,py+3,1)=291:xfeld(px,py+4,1)=295:
GOSUB 5700:GOSUB 5650:GOSUB 5460:setz=true
:RETURN
5540 ' [117]

```

Listing: Sub Sunk

```

5550 IF zuerst=1 THEN xfeld(px,py,1)=315:xf [11844]
eld(px,py+1,1)=319:xfeld(px,py+2,1)=323:xf
eld(px,py+3,1)=327:GOSUB 5700:GOSUB 5650:G
OSUB 5460:setz=true:RETURN
5560 ' [117]
5570 IF torp=1 THEN xfeld(px,py,1)=359:xfe [9850]
ld(px,py+1,1)=363:xfeld(px,py+2,1)=367:GOS
UB 5700:GOSUB 5650:GOSUB 5890:setz=true:RE
TURN
5580 ' [117]
5590 IF untbot=1 THEN xfeld(px,py,1)=339:x [8133]
feld(px,py+1,1)=343:GOSUB 5700:GOSUB 5650:
GOSUB 5910:setz=true:RETURN
5600 ' [117]
5610 '***** UNTERPROGRAMM ZUM MERKEN U. [2495]
MALEN *****
5620 ' [117]
5630 IF setzen=false THEN LOCATE#n,px,py:P [14593]
EN#n,1:PRINT#n,transein$;:LOCATE#n,px,py:P
RINT#n,x$;:PEN#n,0:LOCATE#n,px,py:PRINT#n,
y$;transaus$;:xende=0:yende=0:px=0:py=0:pu
nkt=false:RETURN ELSE RETURN
5640 ' [117]
5650 IF setzen=false THEN LOCATE#n,px,py:P [8282]
RINT#n,transein$:FOR z=py TO (py-1)+laenge
:ELSE RETURN
5660 LOCATE#n,px,z:PEN#n,2:PRINT#n,CHR$(22 [5043]
3);:LOCATE#n,px,z:PEN#n,1:PRINT#n,CHR$(xfe
ld(px,z,1)\2);
5670 LOCATE#n,px,z:PEN#n,0:PRINT#n,CHR$(xf [4551]
eld(px,z,1)-xfeld(px,z,1)\2);:NEXT
5680 IF setzen=false THEN px=0:py=0:xende= [5562]
0:yende=0:punkt=false:RETURN ELSE RETURN
5690 LOCATE#4,1,4:PRINT#4,"ZU LANG !";:FOR [8034]
a=1 TO 1500:NEXT:LOCATE#4,1,4:PRINT#4,STR
ING$(9,CHR$(233));:xende=0:yende=0:RETURN
5700 IF setzen=false THEN LOCATE#2,px,py:P [8292]
EN#2,2:PRINT#2,CHR$(223);:LOCATE#2,x,y:PRI
NT#2,CHR$(223);:RETURN ELSE RETURN
5710 ' [117]
5720 IF fehler=false THEN yende=py+laenge- [7434]
1:GOSUB 5260:RETURN ELSE IF fehler=true T
HEN GOSUB 5690:LOCATE#2,x,y:PEN#2,2:PRINT#
2,CHR$(223);:RETURN
5730 LOCATE#2,altx,alty:LOCATE#2,altx,alty [8175]
:PEN#2,0:PRINT#2,CHR$(xfeld(altx,alty,1)-x
feld(altx,alty,1)\2);:RETURN
5740 ' [117]
5750 '***** UEBERMAL U. LOESCHFUNKTION [911]
FUER CURSOR *****
5760 ' [117]
5770 IF xfeld(altx,alty,1)=223 AND (altx=p [14025]
x AND alty=py) THEN PEN#n,2:LOCATE#n,altx,
alty:PRINT#n,CHR$(228);:LOCATE#n,altx,alty
:PEN#2,3:PRINT#n,transein$;:LOCATE#n,altx,
alty:PRINT#n,CHR$(227);transaus$;
5780 ' [117]
5790 IF xfeld(altx,alty,1)=223 AND (altx<> [4778]
px OR alty <>py) THEN PEN#2,2:LOCATE#2,alt
x,alty:PRINT#2,CHR$(223);
5800 ' [117]
5810 IF xfeld(altx,alty,1)<>223 THEN LOCAT [12588]
E#2,altx,alty:PEN#2,2:PRINT#2,CHR$(223);:L
OCATE#2,altx,alty:PEN#2,1:PRINT#2,transein
$;:LOCATE#2,altx,alty:PRINT#2,CHR$(xfeld(a
ltx,alty,1)\2);:GOSUB 5730
5820 PEN#n,3:PRINT#2,transein$;:LOCATE#n,x [3929]
,y:PRINT#n,CHR$(226);transaus$;
5830 RETURN [555]
5840 ' [117]
5850 IF fehler=true THEN LOCATE#2,xende,ye [7097]

```

Listing: Sub Sunk

Alle verzweifelten HUNCHBACKs können aufatmen: wahlweise können Sie bei Tod ins nächste Bild gelangen und/oder unendliche Leben haben: Da der Lader von HUNCHBACK als ungeschütztes BASIC-Programm abgeSAVED ist, können Sie so vorgehen:

– Kassette zurückgespult einlegen Im Direktmodus eingeben:
 LOAD"" >ENTER< 432 POKE &5111,0 >ENTER<
 (wenn Sie unendliche Leben haben wollen) 434 POKE &5129,0
 >ENTER< (wenn Sie bei Tod ins nächste Bild wollen) RUN
 >ENTER<

Auch OH MUMMY können Sie nun mit unendlichen Leben genießen:

```
10 ' OH MUMMY (Kassette)
20 ' 1987 by J-M.M.
30 '
100 MODE 1:PRINT"LOADING ...
110 MEMORY 15000
120 LOAD"!MUMMY1",&6000 130 POKE 30222,&b7
140 CALL &6000
```

Dieses Listing verschafft Ihnen unendlich viele Raumschiffe bei Konami's NEMESIS. Sowohl PLAYER 1 als auch PLAYER 2 sind nun unsterblich.

```
10 ' NEMESIS (Kassette)
20 ' 1987 by J-M.M.
30 '
100 ENV 8,2,2,2,5,-1,2,1,-7,5:ENV 7,10,-1,5,30,0,1,10,1,10:ENV
5,2,-1,1
110 ENV 3,2,3,1,3,2,1:ENV 1,2,-1,2:ENT -1,1,9,1:ENT
-3,2,-6,1,2,6,1
120 ENT -5,1,-2,1:ENT -7,2,-1,1,2,1,1,5,0,2:ENT
-8,5,0,1,2,-1,1,2,1,1: MODE 1
130 BORDER 0:INK 0,0:INK 1,9:INK 2,11:INK 3,23:LOAD"
!nemesis",&C000: OPENOUT"j
140 MEMORY 999:CLOSEOUT:LOAD"nem code",&3E8
150 POKE 40308,183:PEN 15:MODE 0
160 FOR r=0 TO 15:READ x:INK r,x:NEXT:FOR i=&226 TO &3D4:POKE i,0
170 NEXT:CALL 38341
180 DATA 0,1,2,3,6,7,8,9,11,12,13,15,16,18,24,26
```

Für ein weiteres Weltraumspiel gibt es Hilfe: unendliche Anti-Missile-Missiles (AMMs), unendliche künstliche Sonnen (Flares) und unendlich viele Raketen (Missiles) erhalten Sie bei TAU CETI, wenn Sie dieses Listing eintippen (lassen Sie einfach die Zeilen mit den POKEs weg, die Sie nicht wollen). Getestet habe ich diesen POKE nur für Kassette, es kann aber sein, daß er auch mit der normalen Disk-Version funktioniert.

```
10 ' TAU CETI
20 ' 1986 by J-M.M.
30 '
100 INK 0,0:INK 1,11:INK 2,26:INK 3,24:BORDER 0:MODE 1:PAPER 0:
PEN 1 110 OPENOUT"J":MEMORY &5F9:CLOSEOUT :WINDOW
4,36,21,24:LOAD"!title": CALL &7D0
120 LOAD"!part1":LOAD"!part2
130 POKE &5025,0 'unendl. AMMs
140 POKE &4FC0,0 'unendl. Missiles
150 POKE &4FF2,0 'unendl. Flares
160 INK 1,20:CALL 35003
```

Beenden möchte ich diese POKE-Sammlung mit einem POKE für ein aktuelles Spiel: PAPERBOY. So werden Sie unsterblich und können durch alles durchfahren, ohne daß das Spiel anhält – es gibt keine Crashes mehr – keine Zeit geht mehr verloren! Der POKE für unendliche Zeitungen klappt zwar, scheint aber das Programm etwas zu verwirren, denn die Score- und Bonus-Anzeige werden z.B. vertauscht. Wie immer gilt – Sie brauchen nur die POKEs einzugeben, die Sie auch haben wollen (wenn Sie also nicht unendlich viele Zeitungen möchten, lassen Sie Zeile 130 einfach weg!).

```
10 ' PAPERBOY (Kassette)
20 ' 1987 by J-M.M.
30 '
100 OPENOUT"d":MEMORY 767: LOAD"!elite",&300:
POKE &37E,&C9:CALL 768
110 POKE &9AC,0 'unendl. Leben
120 POKE &9A0,195 'durch alles durch
130 POKE 6407,0 'unendl. Zeitungen
140 CALL 48896
```

So, nun bleibt mir nur noch übrig, Ihnen »Gut Spiel und viel Spaß!« zu wünschen!
 (Jan Mirco Maczewski)

Von Schafen und Wölfen

Brettspiele auf Computer umzusetzen scheint immer noch beliebt bei den Programmierern zu sein, vor allen Dingen deshalb, weil es eine unüberschaubare Menge dieser »Brettl-Games« gibt.

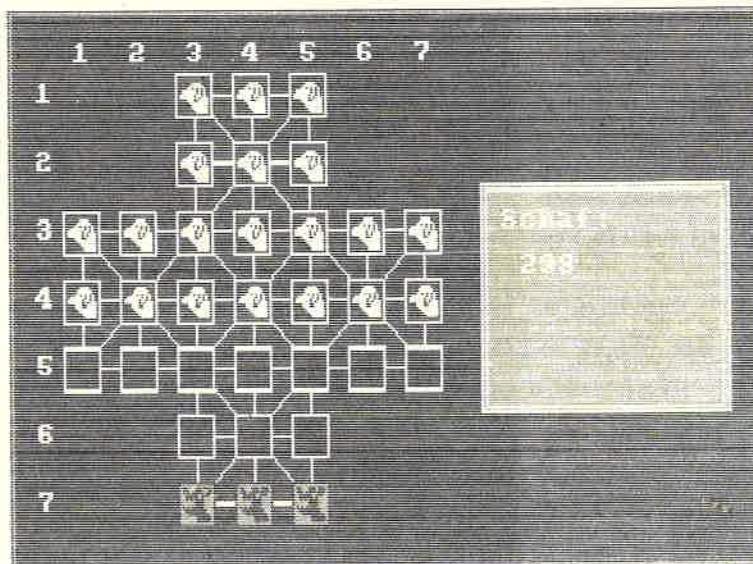


Abb.1: Schafe und Wölfe: Eine Brettspielsimulation

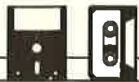
Wir haben Ihnen hier ebenfalls Eins vorzustellen mit dem Namen »Der Wolf jagt die Schafe«. Dieses Spiel ist für 2 Personen gedacht, wovon eine den Part der Wölfe übernimmt, während die andere Person sich um das Wohlergehen der Schafe Sorgen machen muß.

Das Spiel erklärt sich nach dem Start selber, so daß man nicht sofort im Regen steht, bzw. erst lange nach der Anleitung suchen muß.

Und nun viel Spaß beim Wolfs- und Schafsspiel.

(M. Peter/JB)

für 464-664-6128



```

10 DIM feld(75):schafe=20:szeit=300:wzeit= [2877]
300
20 GOSUB 2720:GOSUB 3120:GOSUB 260 [2035]
30 CLS:GOSUB 640 [890]
40 WINDOW#1,26,38,9,18:PAPER#1,3:CLS#1:PLO [9506]
T 400,272,1:DRAW 608,272:DRAW 608,112:DRAW
400,112:DRAW 400,272:PLOT 405,268:DRAW 60
3,268:DRAW 603,117:DRAW 405,117:DRAW 405,2
68
50 ' [117]
60 ' Steuerung [787]
70 ' [117]
80 LOCATE#1,8,2:PRINT#1," ":LOCATE#1,8, [5261]
7:PRINT#1," ":zug=zug+1:IF (zug MOD 2)=
0 THEN 160 ELSE 90
90 LOCATE#1,2,2:PRINT#1,"Schaf:" [1552]
100 set1$=INKEY$:IF set1$="" THEN 140 ELSE [5106]
LOCATE#1,8,2:PRINT#1,set1$:GOTO 110
110 set2$=INKEY$:IF set2$="" THEN 140 ELSE [3981]
LOCATE#1,9,2:PRINT#1,set2$:GOTO 120
120 set3$=INKEY$:IF set3$="" THEN 140 ELSE [5183]
LOCATE#1,10,2:PRINT#1,set3$:GOTO 130
130 set4$=INKEY$:IF set4$="" THEN 140 ELSE [4186]
LOCATE#1,11,2:PRINT#1,set4$:GOTO 1650
140 szeit=szeit-1:IF szeit<1 THEN zug=zug+ [12700]
1:LOCATE#1,2,4:PRINT#1,"0";" ":GOTO 2590
ELSE LOCATE#1,2,4:PRINT#1,szeit;" ":SOUND
1,10+szeit,3,10:FOR a=1 TO 500:NEXT:IF se
t1$="" THEN 100 ELSE IF set2$="" THEN 110
ELSE IF set3$="" THEN 120 ELSE 150
150 IF set4$="" THEN 130 [705]
160 LOCATE#1,2,7:PRINT#1,"Wolf : " [1787]
170 set1$=INKEY$:IF set1$="" THEN 210 ELSE [4167]
LOCATE#1,8,7:PRINT#1,set1$:GOTO 180
180 set2$=INKEY$:IF set2$="" THEN 210 ELSE [4167]
LOCATE#1,9,7:PRINT#1,set2$:GOTO 190
190 set3$=INKEY$:IF set3$="" THEN 210 ELSE [4824]
LOCATE#1,10,7:PRINT#1,set3$:GOTO 200
200 set4$=INKEY$:IF set4$="" THEN 210 ELSE [5721]
LOCATE#1,11,7:PRINT#1,set4$:GOTO 1650
210 wzeit=wzeit-1:IF wzeit<1 THEN zug=zug+ [14500]
1:LOCATE#1,2,9:PRINT#1,"0";" ":GOTO 2590
ELSE LOCATE#1,2,9:PRINT#1,wzeit;" ":SOUND
1,10+wzeit,3,10:FOR a=1 TO 500:NEXT:IF se
t1$="" THEN 170 ELSE IF set2$="" THEN 180
ELSE IF set3$="" THEN 190 ELSE 220
220 IF set4$="" THEN 200 [914]
230 ' [117]
240 ' Instruktionen [406]
250 ' [117]
260 INK 1,26:PEN 1:CLS:MODE 1:LOCATE 11,1: [4135]
PRINT "DER WOLF JAGT SCHAFE"
270 PRINT:PRINT "Dieses Spiel wird zu zwei [2971]
t gespielt."
280 PRINT:PRINT "Ein Spieler uebernimmt di [3972]
e Schafe, waeh-";
290 PRINT "rend der andere die Woelfe uebe [4079]
rnimmt."
300 PRINT:PRINT "Deren gibt es 3, gegenue [4837]
ber einer Herde";
310 PRINT "von 20 Schafen." [1608]
320 PRINT:PRINT "Die Aufgabe der Woelfe [5546]
besteht darin,";
330 PRINT "die Schafe durch Ueberspringen [3833]
derart zu";
340 PRINT "dezimieren,dass am Ende weniger [3267]
als drei";
350 PRINT "Schafe ueberleben." [1957]
360 PRINT:PRINT "Die Aufgabe der Schafe i [3560]
st es, eine der";
370 PRINT "Start-Positionen der Woelfe zu [5042]
besetzen,";
380 PRINT:PRINT "Gesprungen werden kann n [4412]
ur in Richtung";
390 PRINT "der Verbindungslinien und auch [5089]
nur, wenn";
400 PRINT "das anvisierte Feld frei ist." [3568]
410 PRINT:LOCATE 17,25:PRINT "<TASTE>" [2934]
420 A$=INKEY$:IF a$="" THEN 420 [1380]
430 IF a$<>"" THEN CLS:GOTO 440 [1652]
440 PRINT "Um zu springen, oder zu ziehen, [6789]
gibt man";

```

Listing Schafe und Wölfe

```

450 PRINT "zuerst den aktuellen Standpunkt [4232]
, und da-";
460 PRINT "nach den Zielpunkt ein. Will [2444]
man z.B.";
470 PRINT "von 43 nach 44 springen,so gibt [3566]
man 4344";
480 PRINT "ein. Ist der Sprung moeglich, s [5871]
o wird er";
490 PRINT "ausgefuehrt, anderenfalls ert [4296]
oent ein";
500 PRINT "Warnsignal." [1153]
510 PRINT:PRINT "Schafe koennen keine Woe [5406]
lfe uebersprin-";
520 PRINT "gen und nur vor- und seitwaerts [2988]
ziehen."
530 PRINT:PRINT "Wolfe koennen in alle Ric [4153]
htungen sprin-"
540 PRINT "gen und ziehen." [549]
550 PRINT:PRINT:LOCATE 14,20:PRINT " [4078]
VIEL SPASS !"
560 LOCATE 17,25:PRINT "<TASTE> [2025]
570 a$=INKEY$:IF a$="" THEN 570 [1382]
580 IF a$<>"" THEN RETURN [767]
590 ' [117]
600 ' [117]
610 ' Symbole [369]
620 ' [117]
630 ' [117]
640 CLS:INK 1,26:INK 2,15:INK 3,11:BORDER [3025]
15
650 MODE 1 [506]
660 SYMBOL AFTER 200 [1432]
670 SYMBOL 200,255,128,128,128,128,128,128 [2458]
,128
680 SYMBOL 201,255,1,1,1,1,1,1 [1497]
690 SYMBOL 202,128,128,128,128,128,128,128 [2532]
,255
700 SYMBOL 203,1,1,1,1,1,1,1,255 [1782]
710 SYMBOL 204,255,128,177,187,159,159,149 [2414]
,129
720 SYMBOL 205,255,1,221,253,57,161,225,22 [2010]
5
730 SYMBOL 206,171,191,159,143,128,128,128 [2379]
,255
740 SYMBOL 207,225,241,241,249,253,253,1,2 [2096]
55
750 SYMBOL 208,255,128,128,129,131,134,159 [2683]
,191
760 SYMBOL 209,255,1,1,225,241,169,169,105 [2140]
770 SYMBOL 210,191,135,131,129,128,128,128 [2197]
,255
780 SYMBOL 211,93,61,253,253,253,125,1,255 [2045]
790 ' [117]
800 ' [117]
810 ' Berandung [615]
820 ' [117]
830 ' [117]
840 PLOT 1,1,1:DRAW 639,1:DRAW 639,399:DRA [2957]
W 1,399:DRAW 1,1
850 PLOT 5,5,2:DRAW 635,5:DRAW 635,395:DRA [4113]
W 5,395:DRAW 5,5:PEN 1
860 ' [117]
870 ' [117]
880 ' Spielsteine [765]
890 ' [117]
900 ' [117]
910 leer1$=CHR$(200)+CHR$(201):leer2$=CHR$ [2997]
(202)+CHR$(203)
920 wolf1$=CHR$(204)+CHR$(205):wolf2$=CHR$ [4206]
(206)+CHR$(207)
930 shaf1$=CHR$(208)+CHR$(209):shaf2$=CHR$ [2787]
(210)+CHR$(211)
940 ' [117]
950 ' [117]
960 ' DATA's [368]
970 ' [117]
980 ' [117]
990 DATA 160,48,256,48,160,96,256,96,64,14 [4231]
4,352,144,64,192,352,192,64,240,352,240,16
0,288,256,288,160,336,256,336
1000 DATA 64,240,64,144,112,240,112,144,16 [7019]
0,336,160,48,208,336,208,48,256,336,256,48
,304,240,304,144,352,240,352,144
1010 DATA 160,336,352,144,160,240,256,144, [1790]
64,240,256,48
1020 DATA 256,336,64,144,256,240,160,144,3 [2649]

```

Listing Schafe und Wölfe

```

52,240,160,48
1030 DATA 344,296,248,200,152,104,56 [697]
1040 DATA 40,88,136,184,232,280,328 [1200]
1050 DATA 10,22,10,23,13,22,13,23,16,22,16 [2121]
,23
1060 DATA 10,4,10,5,13,4,13,5,16,4,16,5,10 [10721]
,7,10,8,13,7,13,8,16,7,16,8,4,10,4,11,7,10
,7,11,10,10,10,11,13,10,13,11,16,10,16,11,
19,10,19,11,22,10,22,11,4,13,4,14,7,13,7,1
4,10,13,10,14,13,13,14,16,13,16,14,19,1
3,19,14,22,13,22,14
1070 DATA 4,16,4,17,7,16,7,17,10,16,10,17, [5219]
13,16,13,17,16,16,17,19,16,19,17,22,16,
22,17,10,19,10,20,13,19,13,20,16,19,16,20
1080 ' [117]
1090 ' [117]
1100 ' Spielfeld [77]
1110 ' [117]
1120 ' [117]
1130 RESTORE 990:FOR i=1 TO 20 [1270]
1140 READ x1,y1,x2,y2 [937]
1150 PLOT x1,y1,1 [498]
1160 DRAW x2,y2 [55]
1170 NEXT [350]
1180 ' [117]
1190 ' [117]
1200 ' Koordinaten [829]
1210 ' [117]
1220 ' [117]
1230 PEN 2:TAG [686]
1240 FOR a=1 TO 7 [785]
1250 READ y [436]
1260 MOVE 8,y:PRINT a; [1002]
1270 NEXT [350]
1280 TAGOFF [1066]
1290 TAG [318]
1300 FOR a=1 TO 7 [785]
1310 READ x [433]
1320 MOVE x,372:PRINT a; [1086]
1330 NEXT [350]
1340 TAGOFF [1066]
1350 ' [117]
1360 ' [117]
1370 ' Feldaufbau [1291]
1380 ' [117]
1390 ' [117]
1400 PEN 3:FOR a=1 TO 3 [1232]
1410 READ x1,y1,x2,y2 [937]
1420 PRINT CHR$(24):LOCATE x1,y1:PRINT wol [2858]
f1$;
1430 LOCATE x2,y2:PRINT wolf2$;:PRINT CHR$ [2800]
(24)
1440 NEXT [350]
1450 PEN 1:FOR a=1 TO 20 [1026]
1460 READ x1,y1,x2,y2 [937]
1470 LOCATE x1,y1:PRINT shaf1$; [2117]
1480 LOCATE x2,y2:PRINT shaf2$; [2007]
1490 NEXT [350]
1500 FOR a=1 TO 10 [926]
1510 READ x1,y1,x2,y2 [937]
1520 LOCATE x1,y1:PRINT leer1$; [1859]
1530 LOCATE x2,y2:PRINT leer2$; [1919]
1540 NEXT [350]
1550 ' [117]
1560 ' [117]
1570 ' Feldinhalte [1453]
1580 ' [117]
1590 ' [117]
1600 feld(13)=1:feld(14)=1:feld(15)=1:feld [9468]
(23)=1:feld(24)=1:feld(25)=1:feld(31)=1:fe
ld(32)=1:feld(33)=1:feld(34)=1:feld(35)=1:
feld(36)=1:feld(37)=1
1610 feld(41)=1:feld(42)=1:feld(43)=1:feld [4431]
(44)=1:feld(45)=1:feld(46)=1:feld(47)=1
1620 feld(51)=0:feld(52)=0:feld(53)=0:feld [4818]
(54)=0:feld(55)=0:feld(56)=0:feld(57)=0
1630 feld(63)=0:feld(64)=0:feld(65)=0:feld [3935]
(73)=2:feld(74)=2:feld(75)=2
1640 RETURN [555]
1650 ' [117]
1660 ' [117]
1670 ' Zugzerlegung [1651]
1680 ' [117]
1690 ' [117]
1700 a=(VAL(set1$)*10+VAL(set2$)):a1=VAL(s [3881]
et1$):a2=VAL(set2$)

```

Listing Schafe und Wölfe

```

1710 n=(VAL(set3$)*10+VAL(set4$)):n1=VAL(s [3570]
et3$):n2=VAL(set4$)
1720 IF ((zug MOD 2)=0 AND feld(a1*10+a2)< [5760]
>2) THEN GOTO 2580 ELSE IF ((zug MOD 2)<>0
AND feld(a1*10+a2)<>1) THEN GOTO 2580 ELS
E 1730
1730 IF (zug MOD 2)<>0 THEN 1740 ELSE 1760 [2148]
1740 IF a=14 OR a=15 OR a=24 OR a=25 OR a= [11288]
32 OR a=33 OR a=34 OR a=35 OR a=36 OR a=37
OR a=42 OR a=43 OR a=44 OR a=45 OR a=46 O
R a=47 OR a=52 OR a=53 OR a=54 OR a=55 OR
a=56 OR a=57 OR a=64 OR a=65 OR a=74 OR a=
75 THEN 1750 ELSE 1760
1750 IF a2>n2 AND feld(n)=0 THEN 2220 ELSE [1462]
1760
1760 IF a1>n1 AND a1-n1>2 THEN 2580 ELSE 1 [1305]
770
1770 IF a1<n1 AND n1-a1>2 THEN 2580 ELSE 1 [1277]
780
1780 IF a2>n2 AND a2-n2>2 THEN 2580 ELSE 1 [873]
790
1790 IF a2<n2 AND n2-a2>2 THEN 2580 ELSE 1 [2989]
870
1800 IF a<n THEN 1850 ELSE 1810 [1657]
1810 IF a1=n1+2 AND a2=n2+1 THEN 2580 ELSE [2714]
1820
1820 IF a1=n1+2 AND a2=n2-1 THEN 2580 ELSE [2297]
1830
1830 ' [117]
1840 ' [117]
1850 IF a1=n1-2 AND a2=n2-1 THEN 2580 ELSE [1802]
1860
1860 IF a1=n1-2 AND a2=n2+1 THEN 2580 [1306]
1870 ' [117]
1880 ' Feld besetzt [699]
1890 ' [117]
1900 IF feld(n1*10+n2)>0 THEN GOTO 2580 EL [2268]
SE 1940
1910 ' [117]
1920 ' Verbotener Schafzug [1274]
1930 ' [117]
1940 IF (zug MOD 2)<>0 AND a1*10+a2>n1*10+ [4307]
n2 THEN GOTO 2580 ELSE 2070
1950 IF a=13 AND n2<3 OR a=23 AND n2<3 THE [3337]
N GOTO 2580 ELSE 1960
1960 IF a=31 AND n1<3 OR a=32 AND n1<3 THE [3996]
N GOTO 2580 ELSE 1970
1970 IF a=15 AND n2>5 OR a=25 AND n2>5 THE [4295]
N GOTO 2580 ELSE 1980
1980 IF a=36 AND n1<3 OR a=37 AND n1<3 THE [3774]
N GOTO 2580 ELSE 1990
1990 IF a=51 AND n2>5 OR a=52 AND n2>5 THE [2408]
N GOTO 2580 ELSE 2000
2000 IF a=63 AND n1<3 OR a=73 AND n1<3 THE [3419]
N GOTO 2580 ELSE 2010
2010 IF a=65 AND n2>5 OR a=75 AND n2>5 THE [3796]
N GOTO 2580 ELSE 2020
2020 IF a=56 AND n1>5 OR a=57 AND n1>5 THE [2222]
N GOTO 2580 ELSE 2190
2030 ' [117]
2040 ' [117]
2050 ' Entlang der Linien [1305]
2060 ' [117]
2070 IF a=34 OR a=43 OR a=54 OR a=45 THEN [1075]
GOTO 2080 ELSE 2100
2080 IF n1=a1-1 AND n2=a2-1 THEN GOTO 2580 [5713]
ELSE IF n1=a1-1 AND n2=a2+1 THEN GOTO 258
0 ELSE IF n1=a1+1 AND n2=a2-1 THEN GOTO 25
80 ELSE 2090
2090 IF n1=a1+1 AND n2=a2+1 THEN GOTO 2580 [2450]
2100 IF a=14 OR a=32 OR a=36 THEN GOTO 211 [2923]
0 ELSE 2120
2110 IF n1=a1+1 AND n2=a2-1 THEN GOTO 2580 [3739]
ELSE IF n1=a1+1 AND n2=a2+1 THEN GOTO 258
0
2120 IF a=52 OR a=56 OR a=74 THEN GOTO 213 [2464]
0 ELSE 2140
2130 IF n1=a1-1 AND n2=a2-1 THEN GOTO 2580 [4175]
ELSE IF n1=a1-1 AND n2=a2+1 THEN GOTO 258
0
2140 IF a=41 OR a=23 OR a=63 THEN GOTO 215 [2112]
0 ELSE 2160
2150 IF n1=a1-1 AND n2=a2+1 THEN GOTO 2580 [4127]
ELSE IF n1=a1+1 AND n2=a2+1 THEN GOTO 258
0
2160 IF a=65 OR a=47 OR a=25 THEN GOTO 217 [3111]

```

Listing Schafe und Wölfe

```

0 ELSE 2180
2170 IF n1=a1-1 AND n2=a2-1 THEN GOTO 2580 [4364]
ELSE IF n1=a1+1 AND n2=a2-1 THEN GOTO 258
0
2180 ' [117]
2190 ' [117]
2200 ' Zug ausfuehren [1150]
2210 ' [117]
2220 feld(a1*10+a2)=0:IF (zug MOD 2)=0 THE [3984]
N feld(n1*10+n2)=2 ELSE feld(n1*10+n2)=1
2230 LOCATE 3+2*(a2*0.5+(a2-1)),3+2*(a1*0. [3345]
5+(a1-1)):PRINT leer1$;
2240 LOCATE 3+2*(a2*0.5+(a2-1)),4+2*(a1*0. [2040]
5+(a1-1)):PRINT leer2$;
2250 IF (zug MOD 2)=0 THEN 2310 ELSE 2260 [2048]

2260 LOCATE 3+2*(n2*0.5+(n2-1)),3+2*(n1*0. [3451]
5+(n1-1)):PRINT shaf1$;: SOUN
D 1,239,8,8
2270 LOCATE 3+2*(n2*0.5+(n2-1)),4+2*(n1*0. [7717]
5+(n1-1)):PRINT shaf2$;:SOUND 1,119,8,8:IF
feld(73)=1 OR feld(74)=1 OR feld(75)=1 TH
EN GOTO 2590 ELSE GOTO 80
2280 GOTO 80 [306]
2290 ' [117]
2300 ' Wolfszug [710]
2310 ' [117]
2320 IF (zug MOD 2)=0 AND (a1=n1+2 OR a1=n [8172]
1-2 OR a2=n2-2 OR a2=n2+2) AND feld(((a1+n
1)/2)*10+(a2+n2)/2)=1 THEN GOTO 2500 ELSE
2330
2330 PEN 3:LOCATE 3+2*(n2*0.5+(n2-1)),3+2* [5111]
(n1*0.5+(n1-1)):PRINT CHR$(24);wolf1$;CHR$(
24);:SOUND 1,1911,10,10,,,31
2340 LOCATE 3+2*(n2*0.5+(n2-1)),4+2*(n1*0. [5577]
5+(n1-1)):PRINT CHR$(24);wolf2$;CHR$(24);:
PEN 1:GOTO 80
2350 ' [117]
2360 ' Wolf springt [978]
2370 ' [117]
2380 IF feld(((a1+n1)/2)*10+((a1+n1)/2)) [2965]
<>1 THEN 2580 ELSE 2390
2390 IF a2=n2-2 THEN 2500 ELSE 2400 [1433]
2400 IF a2=n2+2 THEN 2500 ELSE 2410 [1180]
2410 IF a1=n1+2 THEN 2500 ELSE 2420 [1412]
2420 IF a1=n1-2 THEN 2500 ELSE 2430 [1167]
2430 IF (a1=n1+2 AND a2=n2-2) THEN 2500 EL [925]
SE 2440
2440 IF (a1=n1-2 AND a2=n2+2) THEN 2500 EL [2762]
SE 2450
2450 IF (a1=n1+2 AND a2=n2+2) THEN 2500 EL [1271]
SE 2460
2460 IF (a1=n1-2 AND a2=n2-2) THEN 2500 EL [3256]
SE 2570
2470 ' [117]
2480 ' Schaf uebersprungen [1105]
2490 ' [117]
2500 ' [117]
2510 ' [117]
2520 LOCATE 3+2*((n2+a2)/2)*0.5+(((n2+a2) [3248]
/2)-1)),3+2*((n1+a1)/2)*0.5+(((n1+a1)/2)-
1)):PRINT leer1$;
2530 LOCATE 3+2*((n2+a2)/2)*0.5+(((n2+a2) [5967]
/2)-1)),4+2*((n1+a1)/2)*0.5+(((n1+a1)/2)-
1)):PRINT leer2$;:feld(((n1+a1)/2)*10+((n2
+a2)/2))=0
2540 PEN 3:LOCATE 3+2*(n2*0.5+(n2-1)),3+2* [4535]
(n1*0.5+(n1-1)):PRINT CHR$(24);wolf1$;CHR$(
24);
2550 LOCATE 3+2*(n2*0.5+(n2-1)),4+2*(n1*0. [5554]
5+(n1-1)):PRINT CHR$(24);wolf2$;CHR$(24);:
PEN 1
2560 FOR c=1900 TO 2500 STEP 10:SOUND 1,c, [7659]
1,INT((2500-c)/40),,,31:NEXT:schafe=schafe
-1:IF Schafe<3 THEN 2590
2570 GOTO 80 [306]
2580 PRINT CHR$(7):zug=zug-1:GOTO 80 [2074]
2590 ' [117]
2600 CLS#1:PLOT 400,272,1:DRAW 608,272:DRA [8042]
W 608,112:DRAW 400,112:DRAW 400,272:PLOT 4
05,268:DRAW 603,268:DRAW 603,117:DRAW 405,
117:DRAW 405,268
2610 LOCATE#1,3,2:PRINT#1,"Damit hat" [1329]
2620 IF (zug MOD 2)=0 THEN win$=name2$ ELS [2618]
E win$=name1$
2630 LOCATE#1,3,5:PRINT#1,win$ [1411]

```

Listing Schafe und Wölfe

```

2640 LOCATE#1,3,6:PRINT#1,"=====":"LOCA [3613]
TE#1,3,9:PRINT#1,"gewonnen."
2650 FOR t=1 TO 2000:NEXT [1608]
2660 CLS#1:PLOT 400,272,1:DRAW 608,272:DRA [8042]
W 608,112:DRAW 400,112:DRAW 400,272:PLOT 4
05,268:DRAW 603,268:DRAW 603,117:DRAW 405,
117:DRAW 405,268
2670 LOCATE#1,3,3:PRINT#1,"Noch ein" [3099]
2680 LOCATE#1,3,5:PRINT#1," Spiel ?" [1335]
2690 LOCATE#1,3,8:PRINT#1," (j/n)" [1201]
2700 a$=INKEY$:IF a$="" THEN 2700 ELSE 271 [1970]
0
2710 IF a$="j" OR a$="J" THEN RUN ELSE IF [2425]
a$="n" OR a$="N" THEN CALL 0
2720 MODE 1:BORDER 0:PAPER 0:CLS:INK 1,5:I [4890]
NK 2,10:INK 3,15:INK 4,20:INK 5,25:INK 6,2
5
2730 FOR a=1 TO 20:b=b+1:IF b>5 THEN GOSUB [3842]
3110 ELSE 2740
2740 PLOT 240-a*2,144-a*2,b [913]
2750 DRAW 400+a*2,144-a*2 [540]
2760 DRAW 400+a*2,256+a*2 [479]
2770 DRAW 240-a*2,256+a*2 [1542]
2780 DRAW 240-a*2,144-a*2 [1462]
2790 NEXT [350]
2800 INK 7,26:PEN 7:LOCATE 17,11:PRINT "De [1967]
r Wolf"
2810 LOCATE 19,13:PRINT "jagt" [1578]
2820 LOCATE 18,15:PRINT "Schafe" [1461]
2830 a=5 [339]
2840 FOR b=1 TO 50 [1258]
2850 INK 1,a [292]
2860 INK 2,2*a [284]
2870 INK 3,3*a [120]
2880 INK 4,4*a [515]
2890 a=a+1 [725]
2900 IF a>6 THEN a=0:FOR c=1 TO 50:NEXT [2287]
2910 NEXT [350]
2920 INK 1,5:INK 2,10:INK 3,15:INK 4,20:IN [2362]
K 5,25
2930 INK 1,25:PEN 1:LOCATE 17,1:PRINT "(c) [5534]
1987":LOCATE 20,3:PRINT "by":LOCATE 14,5:
PRINT "Michael Peter"
2940 LOCATE 8,22:PRINT "Music by Lennon/Mc [5589]
Cartney":LOCATE 16,24:PRINT "< Taste >"
2950 ENV 1,5,-1,24:ENV 2,5,-1,12:ENV 3,5,- [3571]
1,6:ENV 4,5,-1,3:ENV 5,5,-1,9
2960 RESTORE 3000:WHILE INKEY$="" [2280]
2970 READ kanal,ton,en:IF kanal=-1 THEN [7549]
RESTORE 3000:GOTO 2970 ELSE SOUND kanal,to
n,0,15,en:GOTO 2980
2980 WEND [390]
2990 SOUND 7,179,0,15,2 [1389]
3000 DATA 7,142,4,7,119,2,7,179,5,7,478,3 [2346]

3010 DATA 7,358,4,7,119,4,7,106,4,7,89,3,7 [2798]
,106,4,7,119,3
3020 DATA 17,150,4,10,451,4,7,179,3,7,150, [2209]
5,7,451,3
3030 DATA 49,478,4,42,338,4,28,239,4,7,150 [2993]
,4,7,142,4,7,106,3,7,142,4,7,159,3
3040 DATA 49,179,4,42,253,4,28,319,4,7,213 [2544]
,3,7,179,5,17,253,3,10,319,3
3050 DATA 17,268,4,10,478,4,7,239,4,7,142, [4730]
4,7,134,4,17,119,4,10,190,4,7,150,4,17,159
,3,10,190,3
3060 DATA 7,179,4,7,358,4,7,284,4,7,239,3, [2239]
7,284,4,7,239,3
3070 DATA 49,150,3,42,253,3,28,451,3,7,159 [4630]
,3,49,179,3,42,268,3,28,478,3,49,119,3,42,
190,3,28,478,3,-1,-1,-1
3080 ' [117]
3090 ' [117]
3100 RETURN [555]
3110 b=0:RETURN [1221]
3120 INK 1,26:PEN 1:MODE 1:CLS [952]
3130 LOCATE 11,3:PRINT "Spieler 1 (Schafe) [6143]
":LOCATE 11,5:PRINT "Name":LOCATE 16,5:INP
UT name1$
3140 LOCATE 11,8:PRINT "Spieler 2 (Wolf)": [7828]
LOCATE 11,10:PRINT "Name":LOCATE 16,10:INP
UT name2$
3150 LOCATE 18,15:PRINT "Na, dann los Ihr [5440]
zwei !";:FOR a=1 TO 1000:NEXT
3160 RETURN [555]

```

Listing Schafe und Wölfe



LIFE in 3D

Bei 3-D-Life handelt es sich um ein Game of Life, das die Feldelemente nicht in zwei, sondern in drei Dimensionen, also in Würfelform, darstellt.

Dabei kann aus Geschwindigkeitsgründen x (Breite) nur die Werte 8 und 16, y (Höhe) 8 und z (Tiefe) 4 und 8 annehmen. Bei der Eingabe der Feldelemente kann immer nur eine Ebene editiert werden (Anzahl = z). Je weiter vorne sich ein Würfel befindet, desto größer erscheint er auch auf dem Bildschirm. Das ganze Feld ist so zu denken, daß die Enden jeweils verbunden sind. Das Programm wird fast nur mit dem Joystick gesteuert. Sobald Sie das Programm gestartet haben, erscheint eine Aufzählung der Menüpunkte, und über den Joystick kann einer davon angewählt werden.

Die Menüpunkte

1) Eingabe

Der Computer gibt die hinterste Ebene aus; diese kann wieder mit dem Joystick verändert werden. Zwei Pfeile markieren das Feldelement links oben und rechts unten. Unterhalb des Feldes wird die Nummer der Ebene angezeigt. Der Editor kennt folgende Funktionen:



- auf,ab,links,rechts = Cursorbewegung
- Feuer = Feldelement invertieren, Cursor eins weiter
- DEL = Ebene invertieren
- CLR = Ebene löschen, Cursor in die Mitte setzen
- COPY = Nummer der Ebene merken bei zweiter Betätigung wird die gemerkte Ebene in die Aktuelle kopiert
- CURSOR UP,DOWN = eine Ebene vor oder zurück
- ENTER = zurück ins Menue

2) Zufall

Das ganze Feld wird mit 0 oder 1 (unbesetzt oder besetzt) belegt (Verhältnis 1:1)

3) Start

Der Computer berechnet das nächste Bild und zeigt es an. Dabei gilt:

- p Pause (Warten auf Tastendruck)
- auf,ab, links,rechts Sicht verändern
- Feuer Zurück ins Menue, ansonsten weiterberechnen

4) Auflösung

Hier kann die maximale x- bzw. y- Ausdehnung gewählt werden. Die Nummern, die sich der Computer beim SPEICHERN gemerkt hat, werden gelöscht, die Daten nicht.

5) Sicht

Das aktuelle Bild erscheint; dabei kann die Sicht mit dem Joystick verändert werden, (während die einzelnen Feldelemente ihren Zustand nicht verändern).

6) Bedingungen

Hier können die Bedingungen gestellt werden, bei wievielen Nachbarn ein Feldelement stirbt, so bleibt, wie es ist, oder neu besetzt wird.

7) Laden/Speichern

Hier können Sie ein Bild unter einer Nummer aus dem bzw. in den Arbeitsspeicher laden, bzw. speichern (max. 248; bei x=16,z=8 nur 124). Beim Speichern merkt sich der Computer die Nummer und zeigt sie dann bei jeder Bilderauswahl an, bis die Auflösung geändert wird.

8) Film sehen

Mehrere aufeinanderfolgende Bilder werden schnell geladen und ausgegeben; nach dem letzten Bild wiederholt sich das Ganze. Hierbei gelten die gleichen Funktionen wie bei START.

9) Filmen

Eine ganze Reihe von Bildern wird im voraus berechnet und abgespeichert. Sollen viele Bilder berechnet werden, kann es schon ein paar Minuten dauern, dafür geht es hinterher bei FILM SEHEN umso schneller. Wenn alle gewünschten Bilder berechnet worden sind, ertönt ein Piepston mit CHR\$(7).

10) Speicherorganisation

In folgenden Speicherbereichen befinden sich genannte Routinen (alle Speicheradressen in hexadezimaler Form):

- 0200 - 3FFF = gespeicherte Bilder 'boolean'
- 4000 - 7FFF = Zweiter Bildschirm 'OUT &BC00,12:OUT &BD00,&10'
- 8000 - 83FF = aktuelles Feld (bei geringerer Auflösung nur halb, bzw. viertel so groß).
- 8400 - 87FF = Hilfsfeld ('')
- 9000 - 91F7 = Programm 3D-Life (100 % Maschinen-Code).
- A300 - A3F7 = Fragt, ob Bild vorhanden
- A400 - A403 = vier Grenzen für Bedingungen
- A500 - A50C = Variablen

Viel Vergnügen nun mit diesem neuartigen LIFE-Game.

(H. Kellermeyer/JB)

für 464-664-6128



```

10 MODE 1 [506]
15 MEMORY &8FFF:DEFINT a-z:i=&9000:z=990 [1233]
20 WHILE i<>&9F18 [1085]
25 y=1:x=0:z=z+10 [1439]
30 WHILE i<>&9F18 AND y<>76 [2453]
35 READ a$:a=VAL("&"+a$):x=x XOR a:POK [2759]
E i,a
40 y=y+1:i=i+1 [1111]
45 WEND [390]
50 READ a$:IF VAL("&"+a$)<>x THEN PRINT [3795]
Error in Zeile "z":END
55 LOCATE 14,18:PRINT"Zeile"z"OK!" [1236]
60 WEND [390]
65 END [110]
9046]
1000 DATA C3,90,96,3A,01,A5,21,A8,D1,B7,28
,07,11,A0,00,47,19,10,FD,3A,02,A5,07,07,85
,30,01,24,6F,3A,00,A5,FE,02,DA,6E,93,FE,04
,DA,EC,92,FE,06,DA,61,92,C3,B9,91,2A,01,A5
,3A,00,A5,57,7D,17,17,17,00,B4,CB,0A,CB,0A
,62,CB,82,CB,8A,B2,6F,7C,88
1010 DATA E6,03,C9,CD,32,90,F6,80,67,C9,CD [13318]
,32,90,F6,84,67,C9,3E,0F,CD,5A,BB,3E,01,C3
,5A,BB,21,8F,B2,11,2F,B7,3A,06,00,FE,80,28
,01,EB,36,78,C9,2A,01,A5,CB,04,CB,05,3E,0E
,84,67,3E,07,85,6F,C9,06,00,21,40,C1,3E,08
,32,01,A5,E5,54,5D,13,70,75
1020 DATA 01,F0,05,ED,B0,E1,CD,26,BC,3A,01 [14590]
,A5,3D,C2,8E,90,C9,21,00,80,11,01,80,75,01
,FF,07,ED,B0,C9,67,CD,39,90,F6,80,67,7E,81
,4F,C9,E5,F5,CD,75,BB,F1,E1,C9,F5,C5,D5,57
,4F,FE,C8,38,0A,3E,32,CD,5A,BB,79,D6,C8,18
,14,FE,64,38,0A,3E,31,CD,F9
1030 DATA 5A,BB,79,D6,64,18,06,3E,20,CD,5A [11376]
,BB,79,06,00,FE,0A,38,05,D6,0A,04,18,F7,4F
,78,C6,30,FE,30,20,0B,7A,FE,64,30,04,3E,20
,18,02,3E,30,CD,5A,BB,79,C6,30,CD,5A,BB,D1
,C1,F1,C9,F5,79,CD,5A,BB,F1,C9,2A,01,A5,E5
,CD,87,90,CD,66,90,21,38,D1
1040 DATA 91,06,11,7E,CD,5A,BB,23,10,F9,18 [12174]
,11,1F,0C,06,F3,1F,1D,15,F2,1F,10,16,3C,45
,42,45,4E,45,3A,00,A5,CD,C7,90,3E,3E,CD,5A
,BB,AF,32,02,A5,32,01,A5,CD,4E,90,AF,32,01
,A5,AF,32,02,A5,E5,7E,B7,C4,03,90,E1,23,3A
,02,A5,3C,FE,08,20,ED,3A,F4
1050 DATA 01,A5,3C,FE,08,20,E1,E1,22,01,A5 [13307]
,C9,CD,78,BB,E5,21,10,14,CD,BF,90,DD,21,00
,A4,DD,7E,00,3D,CD,C7,90,7C,D6,07,67,2C,CD
,BF,90,DD,7E,00,CD,C7,90,7C,C6,07,67,CD,BF
,90,DD,23,DD,7D,FE,04,20,DC,E1,C3,75,BB,01
,C6,11,11,EE,0F,7E,A3,F6,C7
1060 DATA 01,77,23,72,23,7E,A0,F6,0E,77,2B [11673]
,2B,CD,26,BC,7E,A1,F6,03,77,23,72,23,7E,A0
,F6,2E,77,2B,2B,CD,26,BC,7E,E6,88,F6,07,77
,23,72,23,7E,A0,F6,6E,77,2B,2B,CD,26,BC,16
,F0,72,23,72,23,7E,B3,77,2B,2B,CD,26,BC,72
,23,72,23,7E,B3,77,2B,2B,38
    
```

Listing: Life In 3D

```

1070 DATA CD,26,BC,72,23,72,23,7E,B3,77,2B [11356]
,2B,CD,26,BC,72,23,72,23,7E,B3,77,2B,2B,CD
,26,BC,72,23,72,23,7E,B3,77,2B,2B,CD,26,BC
,72,23,72,23,7E,B3,77,2B,2B,CD,26,BC,72,23
,72,23,7E,B3,77,2B,2B,CD,26,BC,72,23,72,23
,7E,B1,77,2B,2B,CD,26,BC,55
1080 DATA 72,23,72,23,7E,F6,88,77,C9,01,CC [11880]
,33,11,EE,0F,7E,A3,F6,01,77,23,72,23,7E,A0
,F6,0C,77,2B,2B,CD,26,BC,7E,A1,F6,03,77,23
,72,23,7E,A0,F6,4C,77,2B,2B,CD,26,BC,7E,E6
,88,F6,07,77,23,72,23,7E,B1,77,2B,2B,CD,26
,BC,11,F0,F1,73,23,72,23,EA
1090 DATA 7E,B1,77,2B,2B,CD,26,BC,73,23,72 [10583]
,23,7E,B1,77,2B,2B,CD,26,BC,73,23,72,23,7E
,B1,77,2B,2B,CD,26,BC,73,23,72,23,7E,B1,77
,2B,2B,CD,26,BC,73,23,72,23,7E,B1,77,2B,2B
,CD,26,BC,73,23,72,23,7E,F6,88,77,2B,2B,CD
,26,BC,73,23,72,C9,CD,26,CE
1100 DATA BC,01,88,0F,11,11,70,7E,E6,EE,F6 [11813]
,01,77,23,70,23,7E,E6,77,F6,08,77,2B,2B,CD
,26,BC,7E,E6,CC,F6,03,77,23,70,23,7E,B1,77
,2B,2B,CD,26,BC,06,F1,7E,A1,B2,77,23,70,23
,7E,B1,77,2B,2B,CD,26,BC,7E,A1,B2,77,23,70
,23,7E,B1,77,2B,2B,CD,26,FF
1110 DATA BC,7E,A1,B2,77,23,70,23,7E,B1,77 [11032]
,2B,2B,CD,26,BC,7E,A1,B2,77,23,70,23,7E,B1
,77,2B,2B,CD,26,BC,7E,A1,B2,77,23,70,23,7E
,B1,77,2B,2B,CD,26,BC,7E,A1,B2,77,23,70,C9
,CD,26,BC,0E,88,11,70,F3,7E,E6,EE,F6,01,77
,23,36,0F,2B,CD,26,BC,7E,14
1120 DATA E6,CC,F6,03,77,23,36,1F,2B,CD,26 [13198]
,BC,7E,A1,B3,77,23,72,2B,CD,26,BC,7E,A1,B3
,77,23,72,2B,CD,26,BC,7E,A1,B2,77,23,72,2B
,CD,26,BC,7E,A1,B3,77,23,72,2B,CD,26,BC,7E
,A1,B3,77,23,7E,E6,11,F6,E2,77,C9,21,00,CO
,11,00,40,42,4B,ED,B0,01,20
1130 DATA 10,BD,CD,19,BD,ED,49,CD,87,90,21 [9867]
,6E,93,22,F8,93,21,5F,E9,11,00,80,AF,32,00
,A5,AF,E5,32,01,A5,AF,E5,32,02,A5,E5,D5,1A
,B7,C4,00,00,D1,E1,23,23,23,13,3A,02,A5
,3C,FE,08,C2,F0,93,E1,3E,50,85,6F,D2,13,94
,24,CD,26,BC,CD,26,BC,CD,47
1140 DATA 26,BC,CD,26,BC,3A,01,A5,3C,FE,08 [12657]
,C2,EB,93,E1,06,01,2B,10,FD,06,04,CD,26,BC
,10,FB,3A,00,A5,3C,FE,08,CA,6A,94,E5,FE,02
,D2,4A,94,21,6E,93,C3,63,94,FE,04,D2,55,94
,21,EC,92,C3,63,94,FE,06,D2,60,94,21,61,92
,C3,63,94,21,B9,91,22,F8,48
1150 DATA 93,E1,C3,E6,93,01,30,BD,CD,19,BD [12677]
,ED,49,C9,CD,87,90,CD,66,90,AF,32,0A,A5,06
,07,21,8B,94,7E,CD,5A,BB,23,10,F9,18,07,1F
,08,14,42,69,6C,64,21,14,0E,CD,BF,90,16,A3
,3A,0A,A5,5F,1A,B7,28,05,CD,5C,90,18,0A,3E
,0F,CD,5A,BB,3E,02,CD,5A,84
1160 DATA BB,3A,0A,A5,CD,C7,90,CD,18,BB,FE [11791]
,58,28,23,D8,08,28,05,3D,28,0E,18,F0,3A,0A
,A5,3D,FE,FF,20,0D,3E,F7,18,09,3A,0A,A5,3C
,FE,FB,20,01,AF,32,0A,A5,18,B4,CD,5C,90,11
,40,00,21,00,02,3A,0A,A5,47,B7,28,03,19,10
,FD,11,00,80,3E,40,EB,C9,3F
1170 DATA AF,47,DD,21,00,84,1E,01,32,00,A5 [12333]
,3D,E6,07,FD,67,C6,02,E6,07,FD,6F,AF,32,01
,A5,3D,E6,07,32,03,A5,C6,02,E6,07,32,05,A5
,AF,32,02,A5,3D,E6,07,32,0A,A5,C6,02,E6,07
,32,06,A5,48,FD,54,2A,03,A5,CD,B5,90,FD,54
,2A,03,A5,3A,02,A5,CD,B4,AA
1180 DATA 90,FD,54,2A,03,A5,3A,06,A5,CD,B4 [11831]
,90,FD,54,2A,01,A5,3A,04,A5,CD,B4,90,FD,54
,2A,01,A5,CD,B5,90,FD,54,2A,01,A5,3A,06,A5
,CD,B4,90,FD,54,2A,05,A5,3A,04,A5,CD,B4,90
    
```

Listing: Life In 3D



,FD,54,2A,05,A5,3A,02,A5,CD,B4,90,FD,54,2A,05,A5,CD,B5,90,3A,00,A5,5A
 1190 DATA 57,2A,03,A5,CD,B5,90,3A,00,A5,57 [13930]
 ,2A,03,A5,3A,02,A5,CD,B4,90,3A,00,A5,57,2A,03,A5,3A,06,A5,CD,B4,90,3A,00,A5,57,2A,01,A5,3A,04,A5,CD,B4,90,3A,00,A5,57,2A,01,A5,3A,06,A5,CD,B4,90,3A,00,A5,57,2A,05,A5,3A,04,A5,CD,B4,90,3A,00,A5,9A
 1200 DATA 57,2A,05,A5,3A,02,A5,CD,B4,90,3A,00,A5,57,2A,05,A5,CD,B5,90,FD,55,2A,03,A5,CD,B5,90,FD,55,2A,03,A5,3A,02,A5,CD,B4,90,FD,55,2A,03,A5,3A,06,A5,CD,B4,90,FD,55,2A,01,A5,3A,04,A5,CD,B4,90,FD,55,2A,01,A5,CD,B5,90,FD,55,2A,01,A5,3A,40
 1210 DATA 06,A5,CD,B4,90,FD,55,2A,05,A5,3A,04,A5,CD,B4,90,FD,55,2A,05,A5,3A,02,A5,CD,B4,90,FD,55,2A,05,A5,CD,B5,90,FE,00,DA,5E,96,FE,00,DA,67,96,FE,00,DA,64,96,FE,00,DA,67,96,DD,70,00,C3,67,96,DD,73,00,FD,23,3A,02,A5,3C,FE,08,C2,23,95,50
 1220 DATA 3A,01,A5,3C,FE,08,C2,12,95,3A,00,A5,3C,FE,08,C2,03,95,21,00,84,11,00,80,01,00,02,ED,80,C9,06,1A,21,9E,96,7E,CD,5A,BB,23,10,F9,18,1A,04,01,0E,00,1D,0F,0F,1C,00,00,00,1C,01,1A,1A,1C,02,0F,0F,1C,03,05,05,1F,0F,02,CD,66,90,06,55,70
 1230 DATA 21,C9,96,7E,CD,5A,BB,23,10,F9,18,55,47,41,4D,45,20,4F,46,20,4C,49,46,45,1F,0A,04,0F,02,76,6F,6E,20,48,61,72,72,79,20,4B,65,6C,6C,65,72,6D,65,79,65,72,1F,02,18,01,0B,01,0A,01,08,01,09,20,3D,20,53,69,63,68,74,20,20,20,58,20,CB
 1240 DATA 3D,20,4D,65,6E,75,65,20,20,20,70,20,3D,20,50,61,75,73,65,0F,01,CD,A7,90,21,00,02,11,01,02,01,FF,3D,75,ED,80,21,00,A3,11,01,A3,01,02,01,75,ED,80,69,36,07,2C,36,08,2C,36,0D,2C,36,0E,01,03,BC,ED,49,01,14,BD,ED,49,01,0C,BC,ED,62
 1250 DATA 49,21,95,B2,11,53,B7,3A,06,00,FE,80,28,01,EB,36,00,CD,87,90,CD,66,90,06,6A,21,78,97,7E,CD,5A,BB,23,10,F9,18,6A,1F,09,0C,45,69,6E,67,61,62,65,1F,09,0D,5A,75,66,61,6C,6C,1F,09,0E,53,74,61,72,74,1F,09,0F,41,75,66,6C,6F,65,73,75,BC
 1260 DATA 6E,67,1F,09,10,53,69,63,68,74,1F,09,11,42,65,64,69,6E,67,75,6E,67,65,6E,1F,09,12,4C,61,64,65,6E,1F,09,13,53,70,65,69,63,68,65,72,6E,1F,09,14,46,69,6C,6D,20,73,65,68,65,6E,1F,09,15,46,69,6C,6D,65,6E,0F,01,3E,05,32,01,A5,C6,08,36
 1270 DATA 6F,26,14,CD,75,BB,3E,F2,CD,5A,BB,3E,08,CD,5A,BB,CD,18,BB,D6,0A,28,3F,3D,28,2E,FE,4D,20,F2,3A,01,A5,3D,CA,4F,98,3D,CA,98,99,3D,CA,40,9B,3D,CA,17,9D,3D,CA,6D,9B,3D,CA,F3,99,3D,CA,DB,9B,3D,CA,08,9C,3D,CA,70,9C,C3,DA,9C,3E,10,CD,77
 1280 DATA 5A,BB,3A,01,A5,3D,20,AB,3C,18,A5,3E,10,CD,5A,BB,3A,01,A5,3C,FE,08,20,98,3D,18,95,CD,87,90,AF,32,00,A5,3D,32,0C,A5,21,03,03,22,01,A5,CD,20,91,CD,5C,90,CD,4F,99,CD,18,BB,F5,CD,69,99,F1,FE,7F,CA,7D,99,06,08,28,37,3D,28,3A,3D,28,90
 1290 DATA 1F,3D,28,22,FE,4D,28,3C,FE,E5,28,64,FE,E6,28,66,FE,05,28,6E,FE,02,CA,64,97,FE,D5,CA,19,99,18,C7,3A,01,A5,3C,18,04,3A,01,A5,3D,E6,07,32,01,A5,C3,66,98,3A,02,A5,3D,18,04,3A,02,A5,3C,E6,07,32,02,A5,C3,66,98,CD,4E,90,7E,EE,01,77,BE
 1300 DATA 28,05,CD,03,90,18,E6,CD,77,90,25,2D,CD,75,BB,06,06,21,E7,98,7E,CD,5A,BB,23,10,F9,18,06,20,10,08,0A,20,10,18,C8,3A,00,A5,3D,18,04,3A,00,A5,3C,E6,07,32,00,A5,C3,60,98,CD,87,90,21,00,00,22,01,A5,CD,4E,90,54,5D,13,01,3F,00,70,ED,21
 1310 DATA B0,C3,5A,98,3A,0C,A5,3C,20,09,3A,00,A5,32,0C,A5,C3,66,98,3D,CD,42,99,E5,3A,00,A5,CD,42,99,EB,E1,01,40,00,ED,80,3E,FF,32,0C,A5,C3,60,98,21,00,80,B7,C8,11,40,00,47,19,10,FD,C9,CD,77,90,CD,75,BB,06,05,21,63,99,7E,CD,5A,BB,23,10,5B
 1320 DATA F9,18,05,16,01,F7,16,00,C9,CD,77,90,CD,75,BB,3E,20,CD,5A,BB,CD,4E,90,7E,3D,0C,C3,03,90,21,00,00,3A,00,A5,57,CD,39,90,F6,80,67,3A,48,99,47,7E,EE,01,77,23,10,F9,C3,60,98,11,00,80,ED,5F,6F,26,90,ED,5F,AE,AC,AD,E6,1F,B4,67,7E,E6,72
 1330 DATA 01,12,23,13,7A,FE,84,20,F5,C3,64,97,CD,C4,93,CD,FB,94,CD,1B,BB,30,F5,FE,70,CA,67,9B,FE,58,CA,64,97,D6,08,20,05,CD,AF,9B,18,E2,3D,20,05,CD,BA,9B,18,DA,3D,20,05,CD,A2,9B,18,D2,3D,20,CF,CD,99,98,18,CA,CD,83,91,C3,93,9A,CD,87,90,99
 1340 DATA CD,66,90,06,43,21,07,9A,7E,CD,5A,BB,23,10,F9,18,43,1F,0E,0E,4E,61,63,68,62,61,72,6E,3A,1F,06,10,54,6F,64,20,20,3A,1F,06,11,62,6C,65,69,62,74,3A,1F,06,12,4C,65,62,65,6E,20,3A,1F,06,13,62,6C,65,69,62,74,3A,1F,06,14,54,6F,64,52
 1350 DATA 20,20,20,3A,1F,11,10,0F,02,3E,05

Listing: Life in 3D

,F5,06,07,21,5B,9A,7E,CD,5A,BB,23,10,F9,18,07,62,69,73,08,08,0A,F1,3D,20,E6,3E,01,CD,90,BB,21,10,0D,CD,75,BB,AF,CD,C7,90,CD,83,91,06,08,21,86,9A,7E,CD,5A,BB,23,10,F9,18,08,1F,15,14,32,36,1F,63
 1360 DATA 18,12,3E,02,32,01,A5,3E,F2,CD,5A,8B,3E,08,CD,5A,BB,CD,18,BB,0E,10,CD,19,91,FE,58,CA,64,97,D6,08,28,23,3D,28,42,3D,28,0E,3D,20,DC,3A,01,A5,B7,28,D6,3D,0E,08,18,0A,3A,01,A5,FE,03,28,CA,3C,0E,0A,CD,19,91,18,BF,2A,01,A5,26,A4,7E,7C
 1370 DATA 3D,28,B9,F5,3A,01,A5,B7,F1,20,06,32,00,A4,C3,ED,99,4F,2B,7E,B9,28,A5,23,35,C3,ED,99,2A,01,A5,26,A4,7E,FE,1A,28,96,F5,3A,01,A5,FE,03,F1,20,07,3C,32,03,A4,C3,ED,99,4F,23,7E,3D,B9,CA,93,9A,2B,34,C3,ED,99,21,00,A4,11,48,96,ED,A0,86
 1380 DATA 13,13,13,13,ED,A0,13,13,13,ED,A0,13,13,13,ED,A0,13,13,13,ED,A0,21,00,84,01,00,04,ED,80,C9,CD,1A,9B,C3,87,99,3E,1D,CD,5A,BB,3E,05,CD,5A,BB,3E,05,CD,5A,BB,CD,18,BB,3E,1D,CD,5A,BB,3E,0F,CD,5A,BB,3E,0F,C3,5A,BB,CD,46,9B,C3,87,99,D4
 1390 DATA CD,C4,93,CD,18,BB,FE,58,CA,64,97,D6,08,28,13,3D,28,15,3D,28,08,3D,20,EB,CD,99,9B,18,E3,CD,A2,9B,18,DE,CD,AF,9B,18,D9,CD,BA,9B,18,04,3A,2F,94,3D,20,0C,3C,18,09,3A,2F,94,3C,FE,07,20,01,3D,32,2F,94,C9,3A,2A,94,3C,FE,03,20,0A,3D,8C
 1400 DATA 18,07,3A,2A,94,3D,20,01,3C,32,2A,94,C9,32,01,A5,06,08,1A,17,30,02,36,01,23,10,F8,3A,01,A5,13,3D,20,EB,C9,CD,A7,90,CD,73,94,CD,C5,9B,C3,64,97,32,01,A5,06,08,AF,4E,CB,19,17,23,10,F9,12,13,3A,01,A5,3D,20,EB,C9,CD,E7,9B,2A,0A,A5,86
 1410 DATA 26,A3,36,01,C9,CD,73,94,CD,FD,9B,C3,64,97,CD,87,90,CD,66,90,06,09,21,25,9C,7E,CD,5A,BB,23,10,F9,18,09,1F,08,12,53,74,61,72,74,3A,CD,79,94,3A,0A,A5,FE,F7,CA,11,9C,D5,F5,CD,66,90,06,09,21,4C,9C,7E,CD,5A,BB,23,10,F9,18,09,1F,08,CA
 1420 DATA 12,45,6E,64,65,3A,20,3A,0A,A5,CD,7A,94,F1,D1,21,0A,A5,BE,30,D6,4F,32,0B,A5,3A,0A,A5,91,3C,32,07,A5,C9,CD,11,9C,D5,32,08,A5,D5,21,00,80,11,01,80,01,FF,03,75,ED,80,D1,21,00,80,3E,40,CD,C5,9B,D5,CD,C4,93,D1,CD,1B,BB,38,11,3A,08,9E
 1430 DATA A5,3D,C2,74,9C,3A,07,A5,32,08,A5,D1,C3,73,9C,D6,08,20,05,CD,AF,9B,18,E6,3D,20,05,CD,BA,9B,18,DE,3D,20,05,CD,A2,9B,18,D6,3D,20,05,CD,99,9B,18,CE,FE,65,20,05,CD,46,9B,18,C5,FE,4D,20,C1,01,C3,64,97,CD,11,9C,05,2A,0B,A5,26,A3,54,A5
 1440 DATA 5D,13,36,01,06,00,3A,07,A5,3D,4F,ED,B0,CD,1A,9B,D1,3A,07,A5,32,09,A5,D5,CD,FB,94,D1,3E,40,21,00,84,CD,7E,9B,3A,09,A5,3D,C2,F8,9C,3E,07,CD,5A,BB,C3,64,97,CD,87,90,CD,66,90,3E,01,32,01,A5,06,26,21,30,9D,7E,CD,5A,BB,23,10,F9,18,D3
 1450 DATA 26,1F,08,0E,41,75,66,6C,6F,65,73,75,6E,67,3A,1F,08,12,0F,02,66,75,65,72,20,78,3A,1F,08,14,66,75,65,72,20,7A,3A,0F,01,CD,A5,9D,CD,18,BB,FE,58,CA,E2,9D,06,08,28,20,3D,28,1D,3D,28,0C,3D,20,EB,3A,01,A5,3D,20,0C,3C,18,09,3A,01,A5,BC
 1460 DATA 3C,FE,03,20,01,3D,32,01,A5,18,D1,3A,01,A5,3D,28,0D,3A,3A,94,ED,44,C6,0C,32,3A,94,C3,56,9D,3A,6E,96,ED,44,C6,18,32,6E,96,C3,56,9D,21,12,0F,CD,75,BB,3A,6E,96,CD,C7,90,3E,20,CD,5A,BB,3A,01,A5,3D,3E,20,20,02,3E,F2,CD,5A,BB,21,14,0F
 1470 DATA 0F,CD,75,BB,3A,3A,94,CD,C7,90,3E,20,CD,5A,BB,3A,01,A5,FE,02,3E,20,20,02,3E,F2,C3,5A,BB,3A,6E,96,DD,21,74,9E,FD,21,94,9E,FE,08,28,04,FD,21,A4,9E,06,10,DD,6E,00,DD,66,01,FD,7E,00,DD,23,DD,23,FD,23,77,10,EE,3A,3A,94,DD,21,B4,9E,37
 1480 DATA FD,21,C8,9E,FE,04,28,04,FD,21,D2,9E,06,0A,DD,6E,00,DD,66,01,FD,7E,00,DD,23,DD,23,FD,23,77,10,EE,3A,6E,96,DD,21,DC,9E,4F,3A,3A,94,B1,21,F0,9E,FE,0C,28,11,21,FA,9E,FE,08,28,0A,21,04,9F,FE,14,28,03,21,0E,9F,01,FF,0A,DD,5E,00,DD,7F
 1490 DATA 56,01,ED,A0,DD,23,DD,23,10,F2,21,00,A3,11,01,A3,01,FF,00,75,ED,80,C3,64,97,07,90,3D,90,41,90,42,90,7F,90,39,91,30,91,73,91,06,94,28,95,2F,95,5C,98,BC,98,11,99,36,99,48,99,A8,00,CB,0A,0E,0C,1D,08,08,07,07,03,07,3F,40,40,98,17,5E
 1500 DATA 00,00,06,04,25,10,10,0F,0F,0F,0F,7F,80,80,21,90,26,90,2B,90,40,94,4B,94,56,94,08,95,0E,95,80,96,FA,98,01,02,03,01,02,03,03,03,04,03,02,04,06,02,04,06,07,07,08,07,EO,93,DO,94,D8,94,E5,94,F8,94,8C,96,35,9C,80,9C,89,9C,01,9D,5B,44
 1510 DATA F7,F8,20,20,01,F7,00,20,20,5F,F7,FB,40,40,02,F7,01,40,40,40,4C,F7,F8,40,40,02,F7,01,40,40,50,7B,7C,80,80,04,7B,03,80,80,C1

Listing: Life in 3D

WARLOCK

Wollten Sie schon immer mal so richtig böse und unartig sein? Dann ran an »Warlock«. Hier nun die Karte und einige Tips für den ambitionierten Bösewicht.

Bei aller Moral, ich finde es ganz amüsant mal in die Rolle des »Bösen« zu schlüpfen, dem das »Gute« auf die Pelle gerückt ist, denn allmählich ist das Image des großen Weltenretters doch etwas überstrapaziert.

Da auch die anderen Komponenten wie Grafik, Sound und Spielgeschehen sich sehen lassen können, machte ich mich schon bald daran, eine Karte zu zeichnen. Das sah am Anfang ja auch recht einfach aus, aber spätestens in der 12. Ebene war ich dem Nervenzusammenbruch recht nahe.

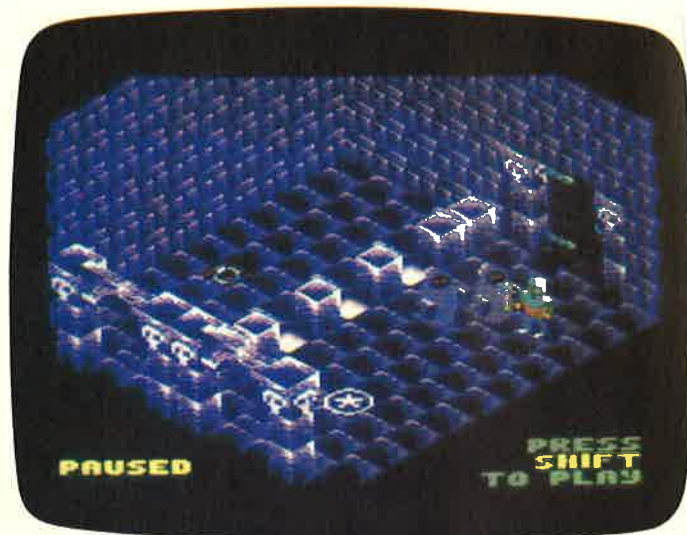
Zum einen, da ich ab Level 9 dachte, daß der nächste ja wohl nun wirklich der letzte sein müßte und zum anderen, da es doch anfang ziemlich zu nerven, daß man dauernd von vorne anfangen darf, denn mit 3 Leben in die höheren Level zu kommen ist eine anfangs nicht ganz einfache Angelegenheit.

Wie Sie an der Karte erkennen können, I've done it und Sie werden dem Guten eher den Garaus machen können als Ihnen lieb ist. Dazu muß man nur wissen, daß der »Weiße Magier« recht zart besaitet ist und sich anscheinend leicht aufregt. Wenn man nämlich leicht, locker, flockig vom Anfangsraum (3. Ebene) in die Behausung des holden Herren (1. Ebene) vordringt und sich diesem von seiner besten Seite (beim Warlock hinten) zeigt, regt dieser sich so auf, daß er einem Herzkasper erliegt und Sie gewonnen haben. Aber ein richtiger Spielefreak läßt sich ja so schnell die Freude an einem solchem Spiel nicht nehmen.

Doch auch auf dem Weg in den 16. Level, in dem sich die Kristallkugel befindet (linke obere Ecke), sind einem Bugs der Programmierer behilflich.

Es wird zwar schon in der Anleitung erwähnt, daß es eine Möglichkeit gibt, zwischen verschiedenen Orten hin und her zu springen, doch so ganz beabsichtigt sieht das nicht aus. Geht man nämlich in einem Raum, der nicht mit Durchgängen von den anderen Räumen getrennt ist, sondern sich anschließt, an der Wand lang zum Rand, so kommt man gelegentlich nicht in den anschließenden Raum, sondern wird durch eine nicht vorhandene Barriere aufgehalten.

Die Spielfigur erscheint dann zu einer Hälfte in der einen Ecke, zur anderen in der schräg gegenüberliegenden. Lenkt man nun auf die Wand zu, erscheint man entweder im Raum hinter der



Mauer, oder, wenn sich dort kein Raum mehr befindet, in einem anderen Level. So kann man z.B. vom 14. in den 1. Level oder auch vom 9. in den 13. springen. Diese Stellen sind in der Karte nicht eingezeichnet, da ich Ihnen ja nicht alle Motivation rauben will.

Als Tip:

Im Level 11 am linken äußeren Rand klappt es bestimmt.

Was macht man aber, wenn man nur noch ein Leben hat und kurz vor dem Exitus steht? Tja, da hilft nur noch Geduld.

Stellen Sie sich in einem Raum mit zwei oder mehr Generatoren an eine günstige Stelle, von der aus Sie nun ersteinmal geraume Zeit lang Gegner bekämpfen.

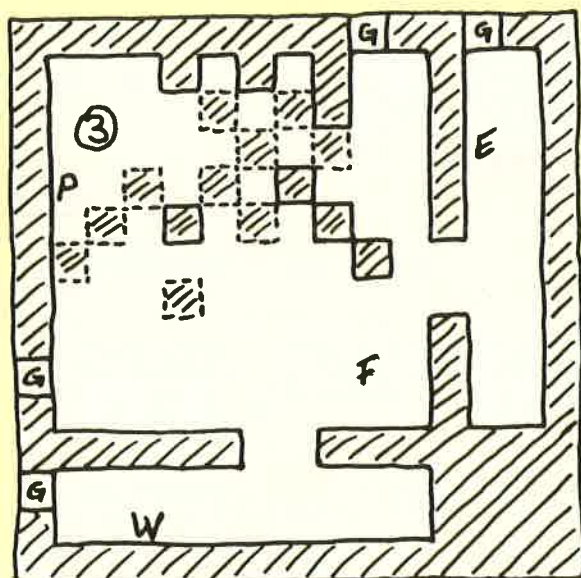
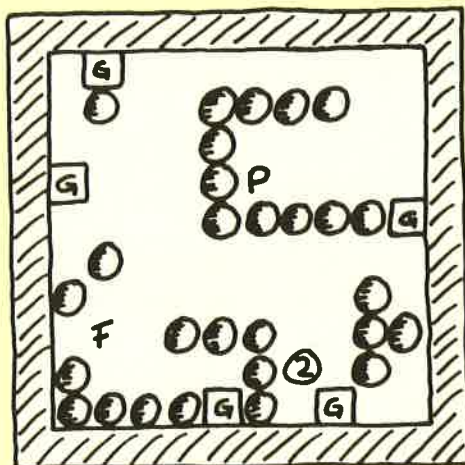
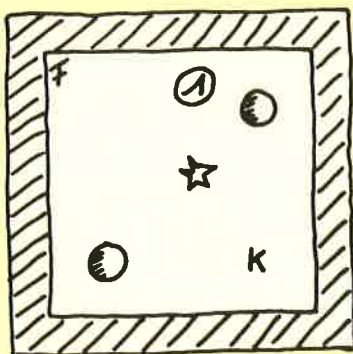
Für je 50000 Points gibt es ein Leben: Man sollte allerdings aufpassen, daß der Generator nicht gerade Schneemänner auswirft, denn für die gibt es nur 100 Punkte. Für die Prinzessinen und Flammen gibt es nämlich 900 und für Krieger immerhin 400 Punkte, was die Prozedur doch erheblich verkürzt.

Zu den drei Erscheinungsformen unseres Finsterlings wäre eigentlich nur zu sagen, daß man als Magier meistens am Besten durchkommt, bloß in Abschnitten in denen von allen Seiten die Leute um Autogramme betteln (was könnten die sonst wollen?) hat der Troll Vorteile.

In den Goblin sollte man sich nur verwandeln, wenn man springen will.

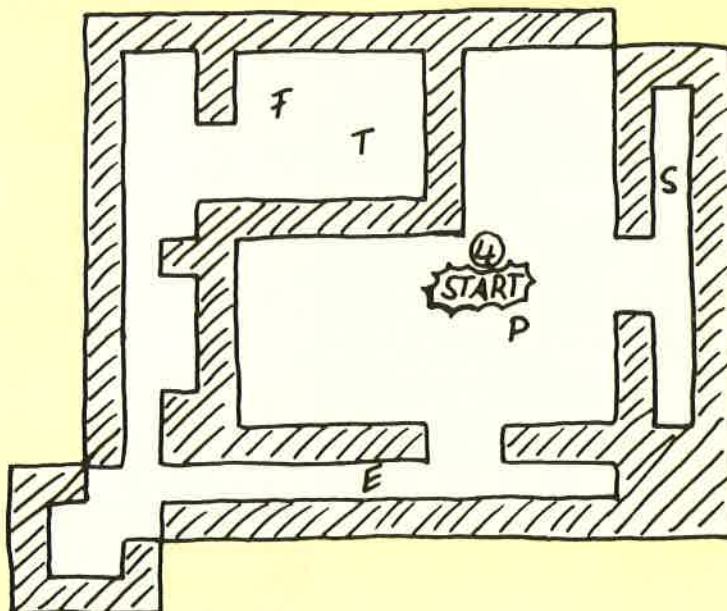
Ich habe so das dumpfe Gefühl, daß man an die Flaschen, die in Ebene 8 in der Wand eingeschlossen sind als Goblin durch zwei Sprünge (auf die Trennmauer und von dort hoch) herankommen kann. Leider ist es mir noch nicht gelungen, mit 2 Zap-Effekten dort hin zu kommen. Dieses Feature ist nämlich sehr hilfreich, wenn nicht gar lebensrettend, wenn man von »fliegenden Gegnern« angegriffen wird, da man diese nur schlecht mit den Kräfteblitzen trifft.

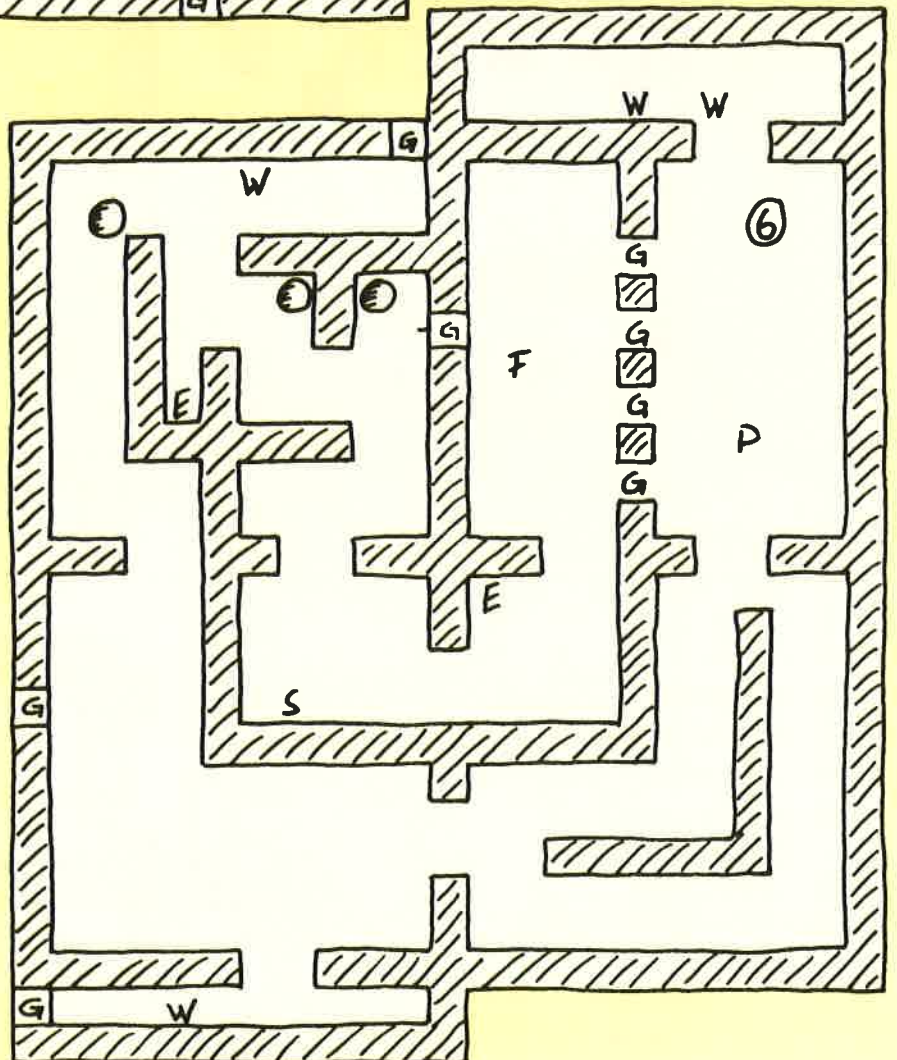
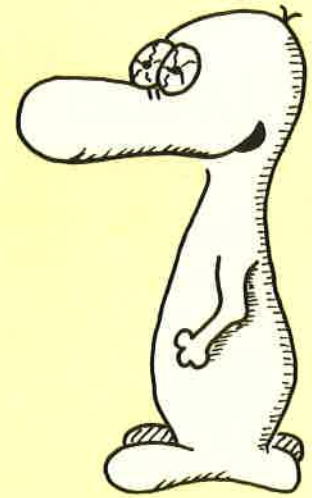
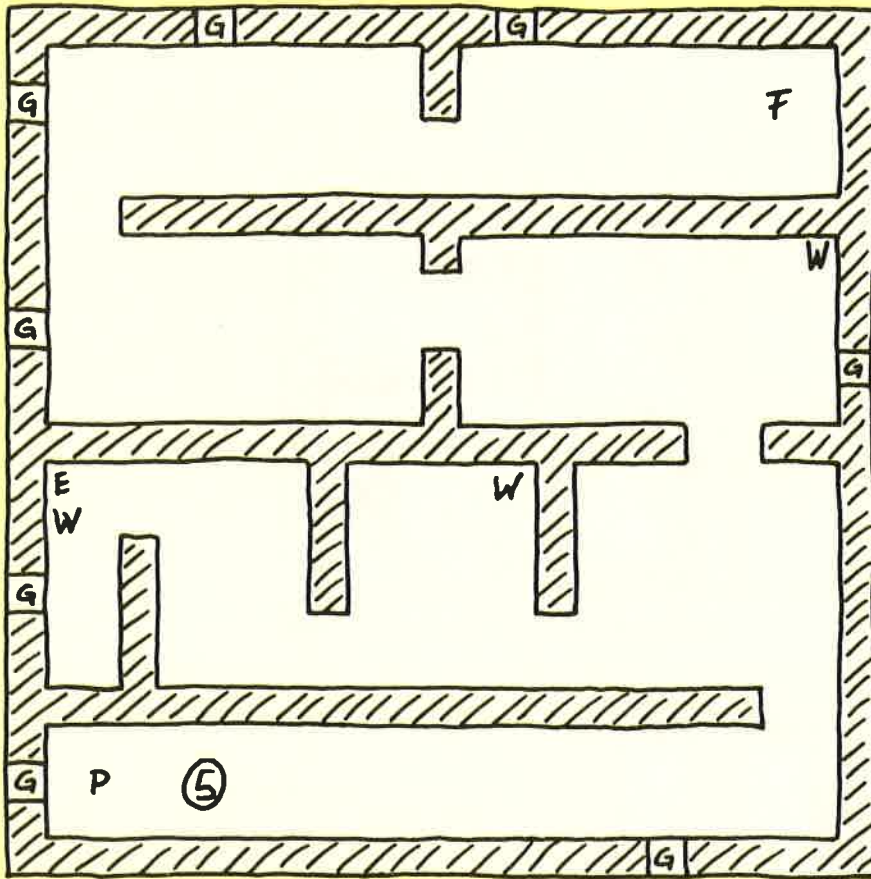
(BB)

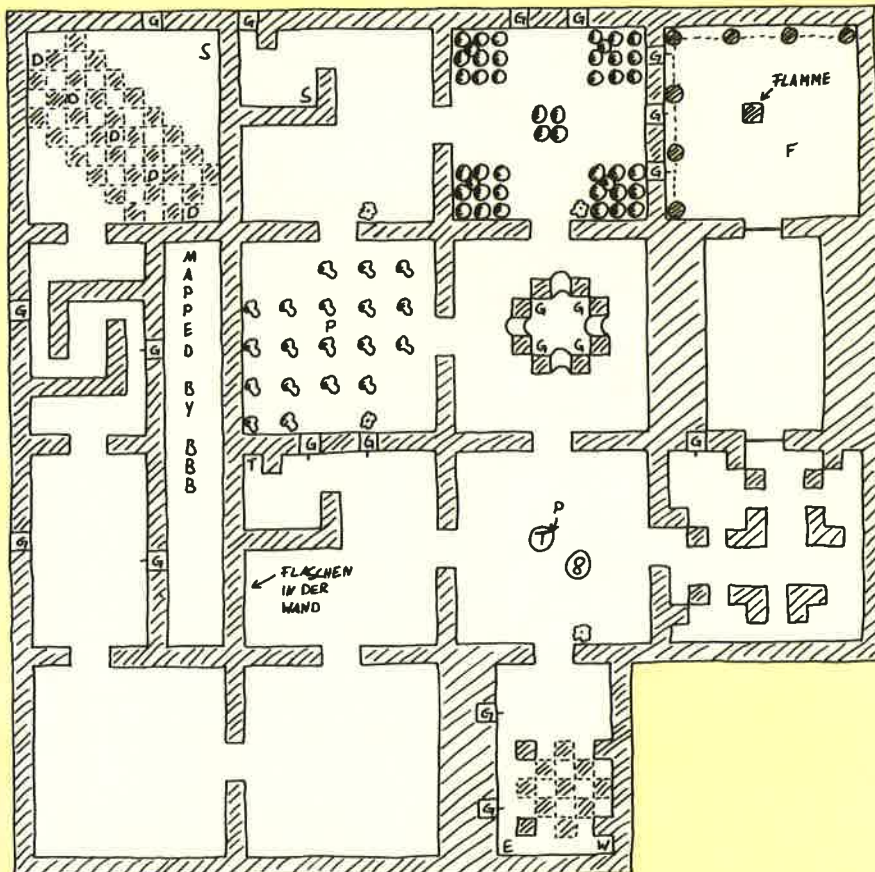
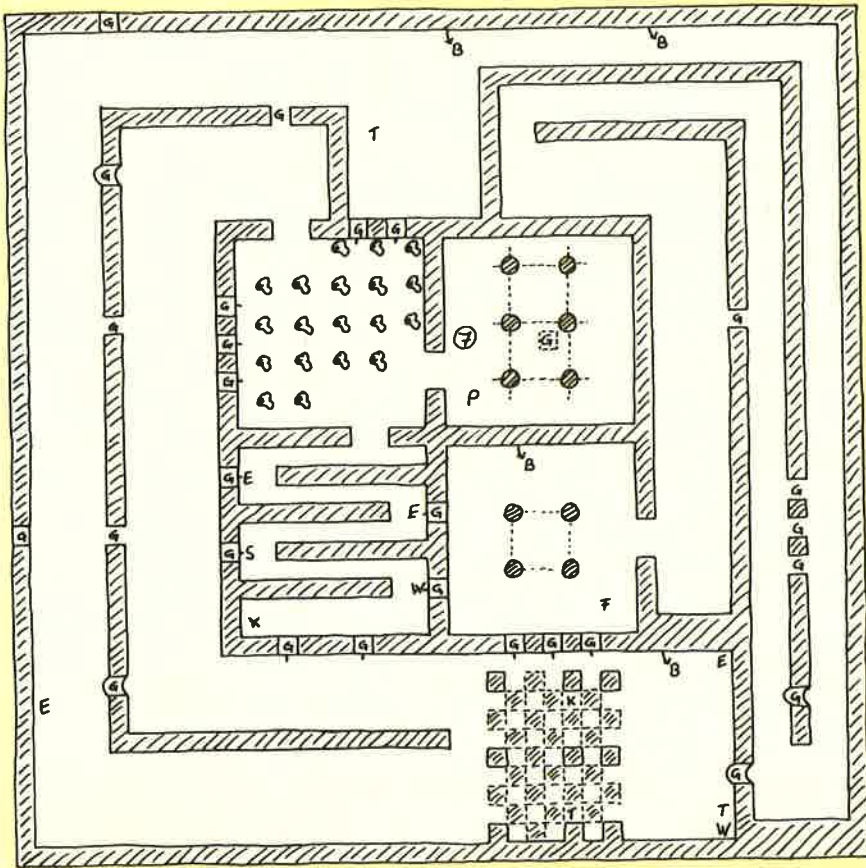


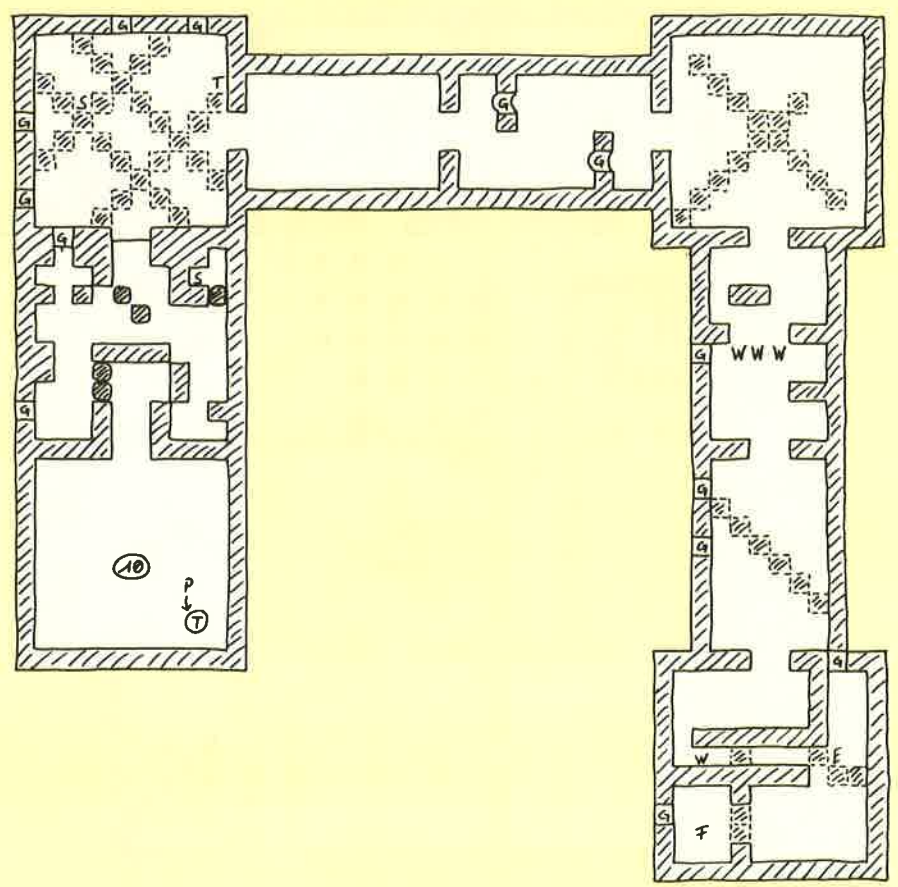
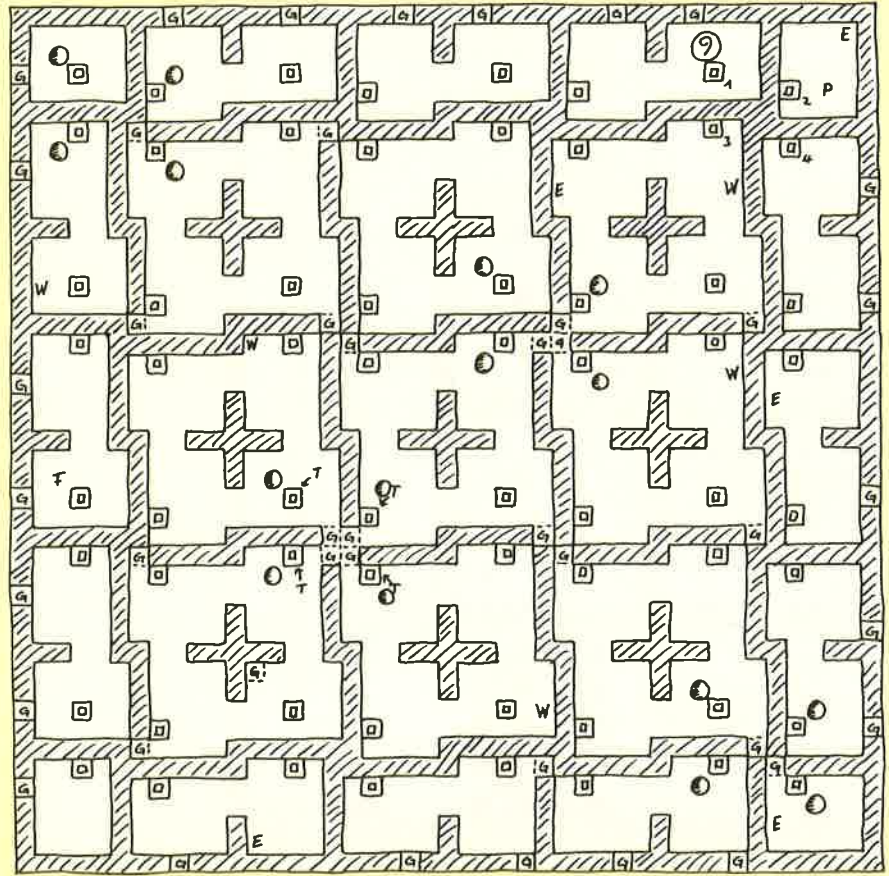
- F - FALLTÜR
- P - PENTAGRAM
- S - SCHLÜSSEL
- E - ESSEN
- T - ZAUBERTRANK
- G - GENERATOR
- W - WERTGEGENSTAND
- K - KERZE / APFEL
- D - DRACHE
- B - BATS
- K - KRISTALL

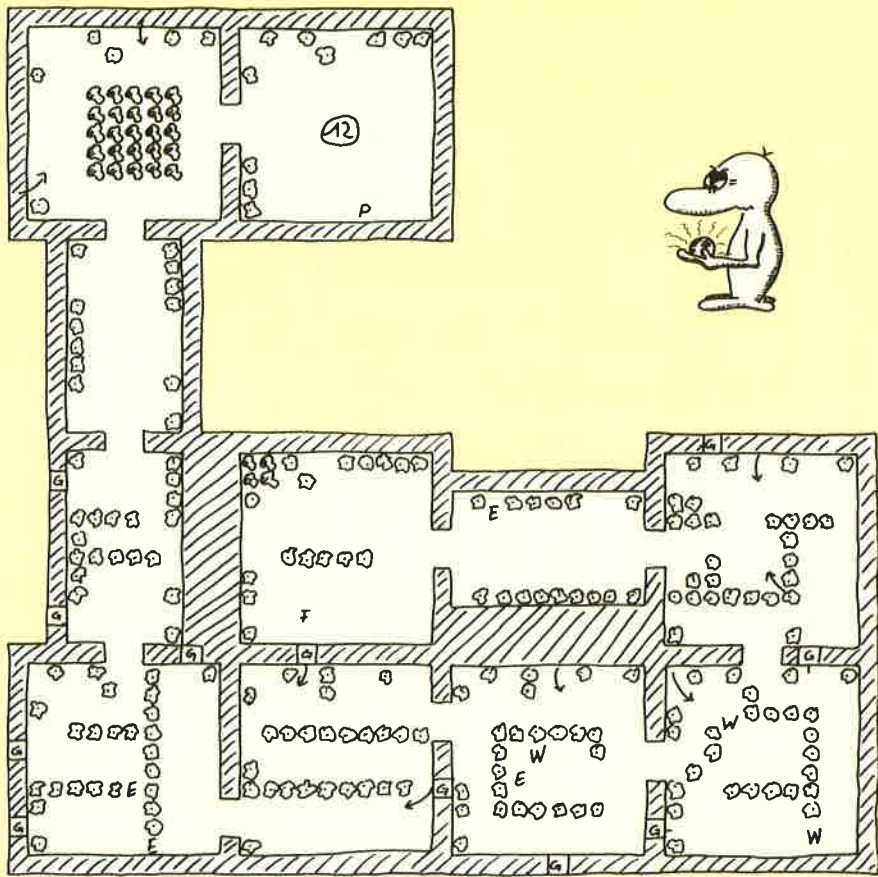
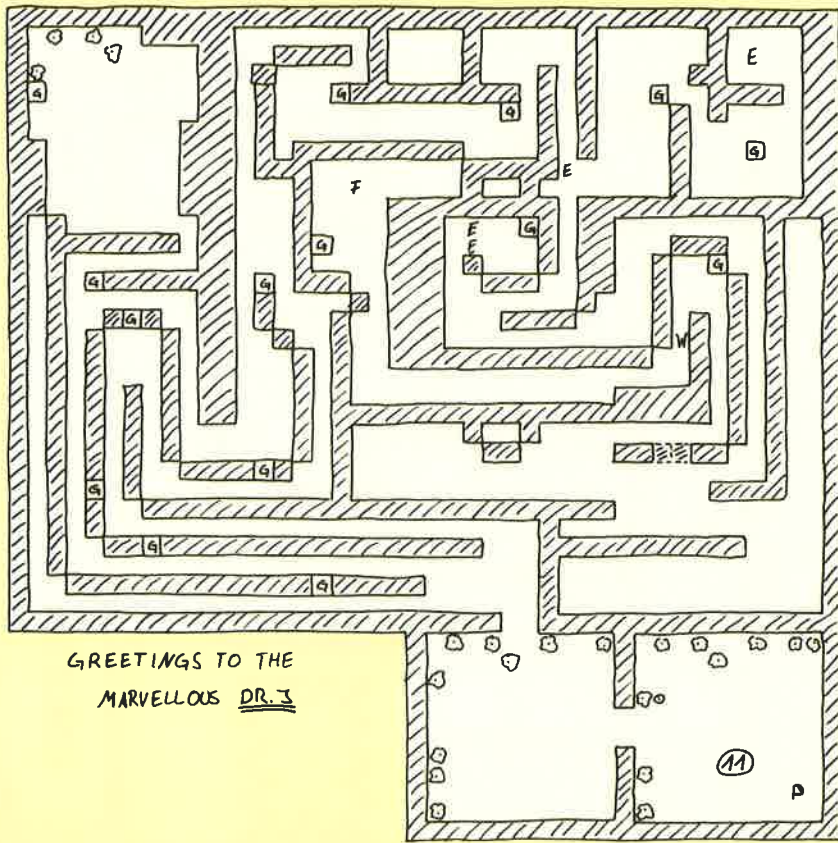
- ☆ - WHITE WIZARD
- ▨ - MAUER ODER STEIN
- - KUGEL
- - SÄULE
- ⊙ - TOTENKOPF
- ⚡ - TOR ALS GENERATOR
- ⊕ - STEINSPITZE
- ▣ - SÄULE
- - BAU ÜBER SÄULE
- ⊠ - SPRUNGFELD
- ⊞ - VERSCHIEBBARER STEIN
- ↘ - FLIEGENDE GEGNER
- - TÜR EINSEITIG VERSCHLOSSEN
- ① - EBENE + ANFANGS ORT SELBIGER

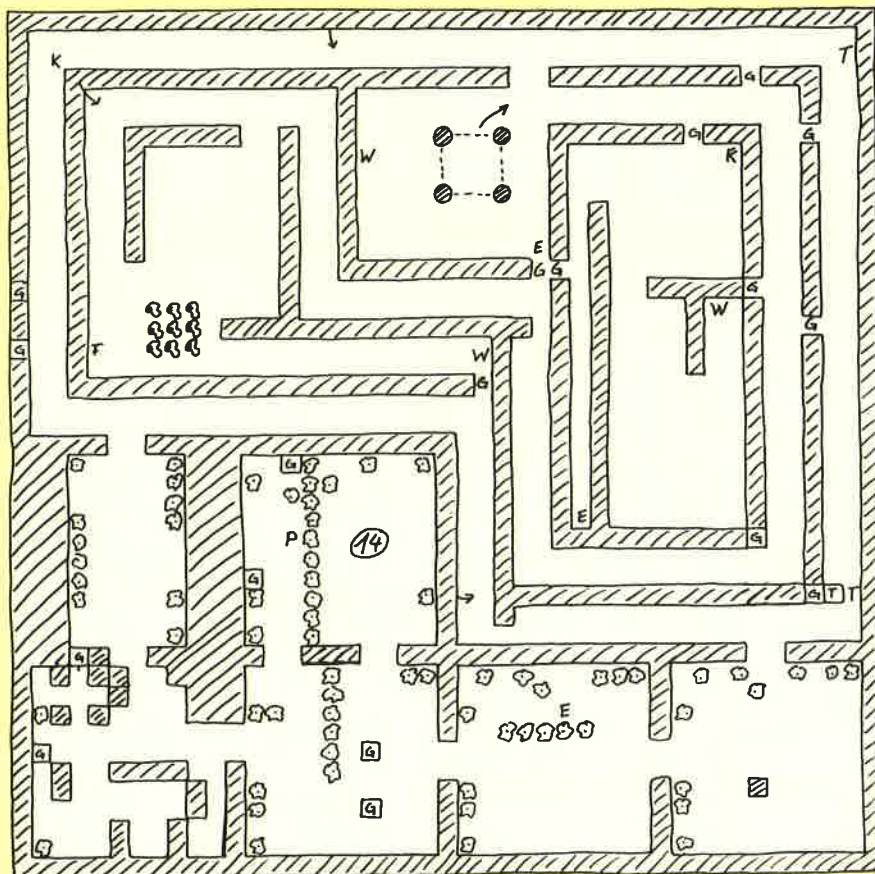
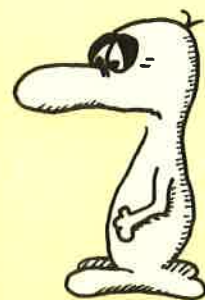
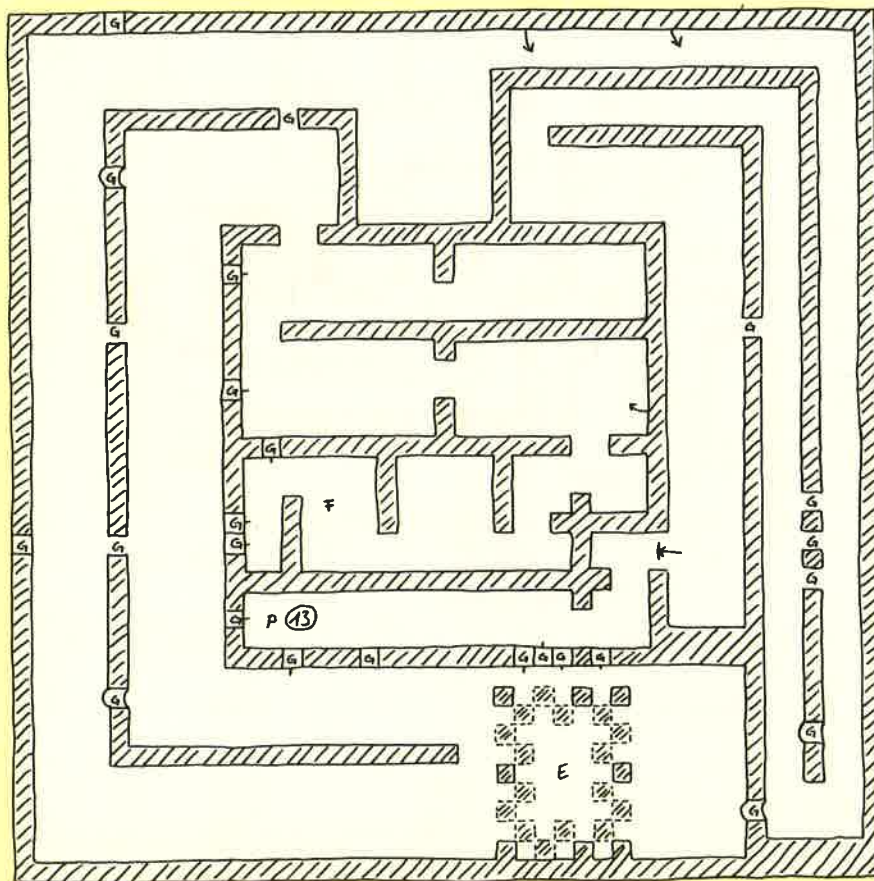


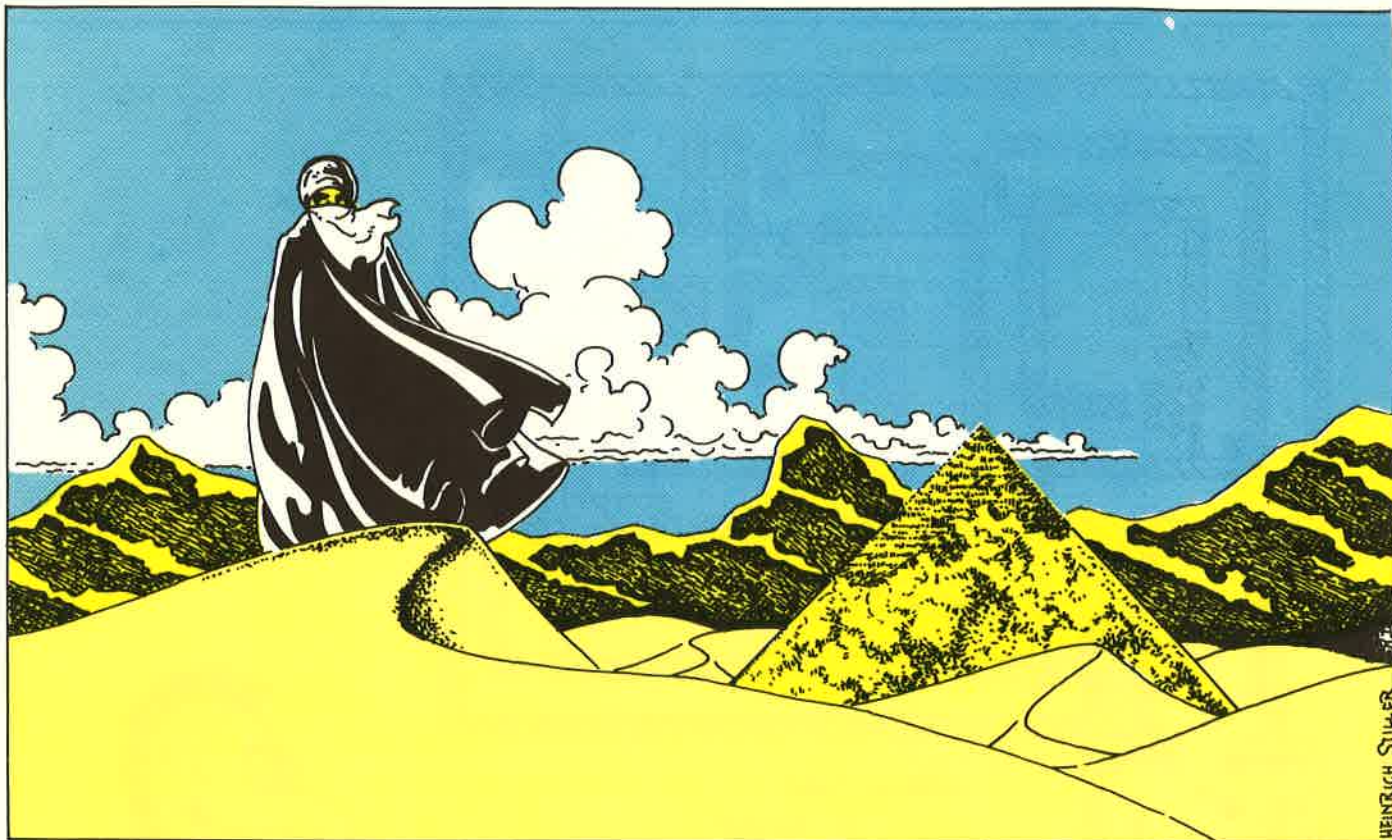












//////////////////////////////////// **INFIDEL** //////////////////////////////////////

Fi aman, Sucher der verlorenen Pyramide! Ich will all den Besitzern dieses ausgesprochen guten und schwierigen Adventures helfen, an das Ziel ihrer Träume zu kommen. Im letzten Sonderheft wurden ja schon ein paar Tips veröffentlicht, aber es soll ja Leute geben, die dieses Heft nicht besitzen (wer denn?).

Also, noch einmal ganz von vorne. Zuerst einmal aus der Hängematte aufstehen. Jetzt verläßt man das Zelt und betritt das Camp. Nun müßte eigentlich ein Flugzeug über Ihnen kreisen und etwas abwerfen. Das »Etwas« ist die schon seit Tagen erwartete Orientierungsausrüstung, die man unbedingt zum Lokalisieren der Pyramide braucht. Nun sollte man die anderen Zelte besuchen und alles mitnehmen. Dann geht man mit der Spitzhacke ins eigene Zelt zurück und zerschlägt das Schloß. Aus der Kiste nimmt man die Karte und das Essen. Die Karte ist unwichtig, den Inhalt sollte man allerdings beachten.

Jetzt geht man noch zum Nil, trinkt Wasser bis es einem aus den Ohren rauskommt und füllt die Feldflasche. Jetzt aber ab in die Wüste. Zum Lokalisieren der Pyramide muß man folgendermaßen vorgehen:

Man nimmt ein Lineal und die Karte. Jetzt legt man das Lineal möglichst gerade an das Kreuz an, und muß nun zwischen den auf der Karte vermerkten Koordinaten suchen. Man sollte sich auch vergewissern, in wie weit sich die Koordinaten bei einem Schritt in Ost- und Südrichtung verändern.

So, an dem Ort, an dem die Pyramide sein sollte, muß man mit der Schaufel graben. Nach ein paar Schaufeln sieht die Welt schon ganz anders aus: entweder liegen Sie unter ein paar Tonnen Sand (dann ist die Pyramide doch wo anders), oder Sie sehen die Pyramide.

Wenn Ihr Held Durst oder Hunger hat, sollten Sie ihn füttern, da sich Hunger auch nachteilig auf die Gesundheit auswirkt. Sollte man nun doch die Spitze sehen, nimmt man den Steinwürfel und steckt ihn in das Loch. Jetzt öffnet sich ein Eingang, in den Sie steigen sollten. Unten angelangt, muß man die Fackel und den Topf nehmen. Den Topf öffnen und die Fackel hineintunken. Nun nimmt man ein Streichholz und zündet so die Fackel an. Jetzt geht man durch den südlichen Torbogen, und holt sich nach ein wenig Laufarbeit den Mast und die beiden Krüge. Den Ring sollte man nicht anziehen, da er einen kleinen Stachel auf der Innenseite hat. Am besten legt man alle Schätze auf den Altar.

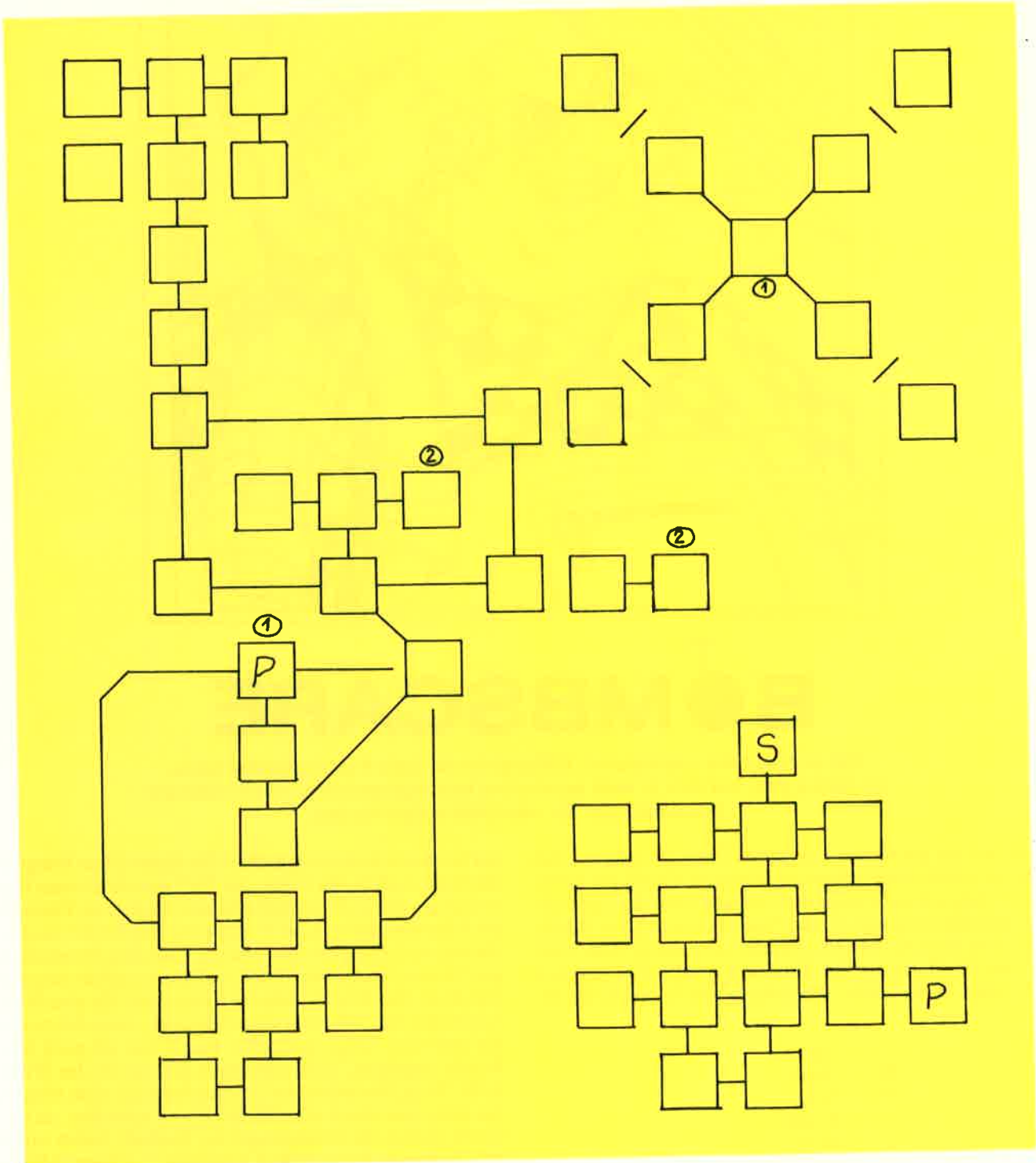
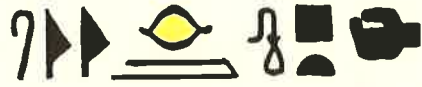
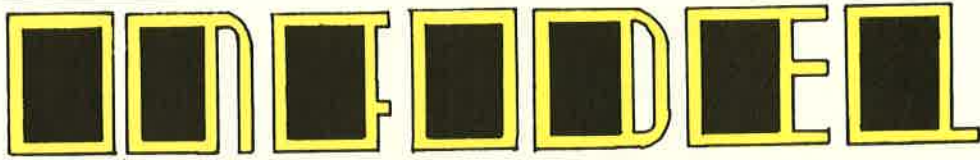
Wieder zurück in der Eingangshalle, nimmt man das Seil, knotet es an den Altar und schmeißt es durch den nördlichen Torbogen. Daran muß man herunterklettern. Unten angelangt, wollen wir erstmal Denkmäler schänden: Wir schmeißen die Statue um, wobei die den Kopf verliert.

Der untere Komplex hat ein ganz raffiniertes Fallensystem. Wenn man beispielsweise durch die nordöstliche Tür geht, verschließt sich prompt die Tür vor einem und im Südosten wird sie geöffnet. Das läßt sich umgehen, indem man die Statue samt Kopf in einen Gang rollt. Nun kann man durch den gegenüberliegenden Eingang durch. Haben Sie nun alle Zimmer geplündert, sollten Sie auch die goldene Traube mitgehen lassen.

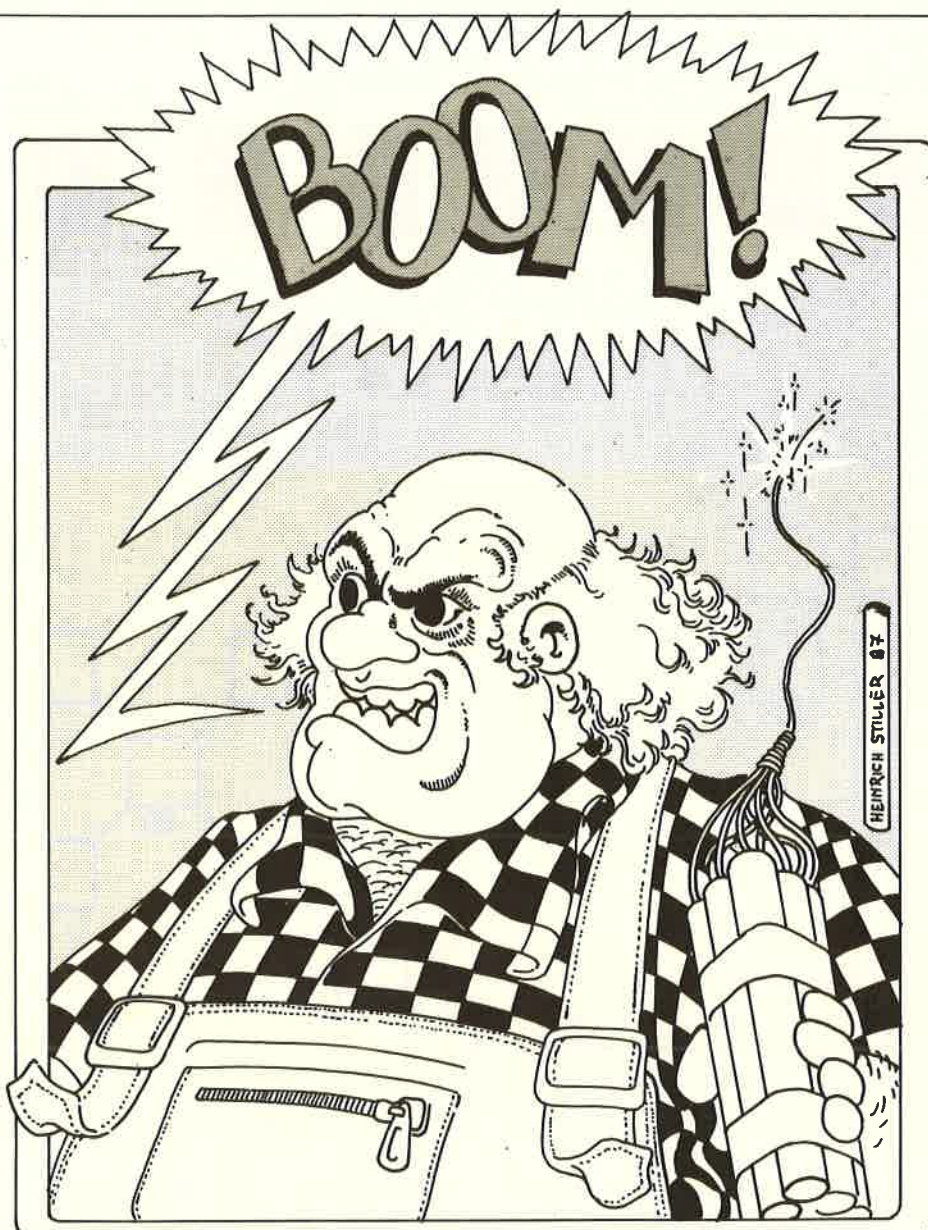
So, damit trennen Sie noch knapp 100 Punkte vom Sieg, doch das Ende des Spiels sollten Sie selber herausfinden, allzu schwer ist es nicht.

Noch ein Tip: tragen Sie sich auf der Karte am besten alle gefundenen Gegenstände und Raumnamen ein, es kann nur hilfreich sein.

(JH/HS)



DESIGNED AND CREATED BY JAN HENNIG THANKS TO TUT-ENCHI-AMUN



BOMBSCARE

Das es unter den sogenannten Billigspielen so manch interessantes Game zu finden gibt, hat sich ja wohl inzwischen herumgesprochen. Bei Bombscare bekommt man für sein Geld einiges zu tun.

Obwohl mir ein bayrischer Combjuda Friek versicherte, daß Bombscare »ein rechter Schmarrn sei und die Grofik gar greißlich wäre« habe ich mich hingesezt und eine Karte dazu gezeichnet. Meiner Meinung nach ist das Programm recht guter Durchschnitt, aber lösen konnte ich es bisher noch nicht. Aber vielleicht kann ich Ihnen mit der Karte und einigen Tips ja schon mal ein Stück weiter helfen. Zunächst mal die Codes für die Teleporter:

- T1 = Zepha
- T2 = Quart
- T3 = Delta
- T4 = Xylem
- T5 = Nitro
- T6 = Crypt
- T7 = Astra
- T8 = Ytron

So, jetzt sollten Sie problemlos überall hin kommen. Da die Gegenstände bei jedem Spiel neu verteilt werden, kann ich Ihnen

nur bei deren Bedeutung helfen: Die kleine Figur bringt ein zusätzliches Leben, der Stern, das TNT und die kleinen Bomben vernichten alle im Raum befindlichen Gegner, die Pistole frischt die Schußenergie auf, die Flasche und das Schild tun das gleiche mit der Lebensenergie. Mit der einfachen Pyramide gelangen Sie zum Starraum zurück, mit der doppelten kommen Sie zum Exit. Der Wecker hält den Zeitcounter für eine Weile an. Ansonsten habe ich noch einen Hammer, einen Schraubenzieher und eine Zange gefunden. Damit bin ich auch brav zur Bombe gepilgert, doch anwenden konnte ich das Werkzeug nicht. Na ja, besonders viel Energie habe ich nicht reingesteckt das Zeug nun durch alle 160 Räume zu schleifen, da ich zum einem endlich die Kristallkugel bei Warlock finden wollte und zum andern doch nun endlich mal Heinie's Arkanoid Highscore schlagen mußte. Wie ist der bloß in den 18. Level gekommen...? Tschä, da müssen Sie wohl selber dran, ich hoffe es lohnt sich. Für 15,- DM kann man ja schon mehr als die Meldung »Sie haben es geschafft« verlangen oder?

(BB)

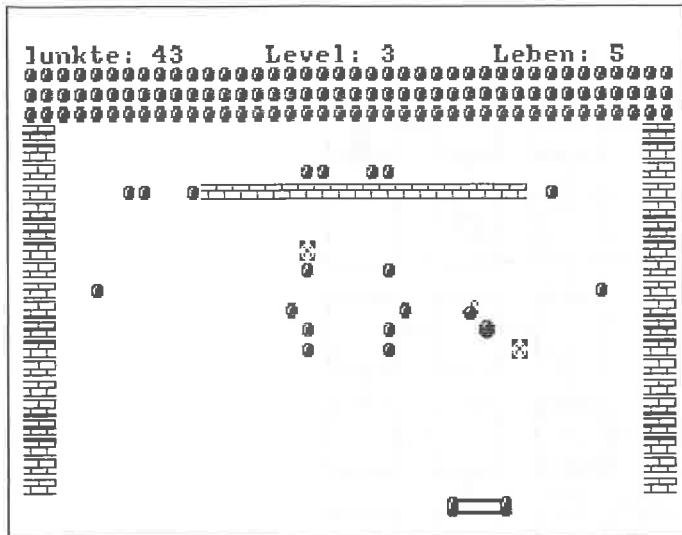


Bild 1: Nicht einfach zu spielen: Ball-Cracker

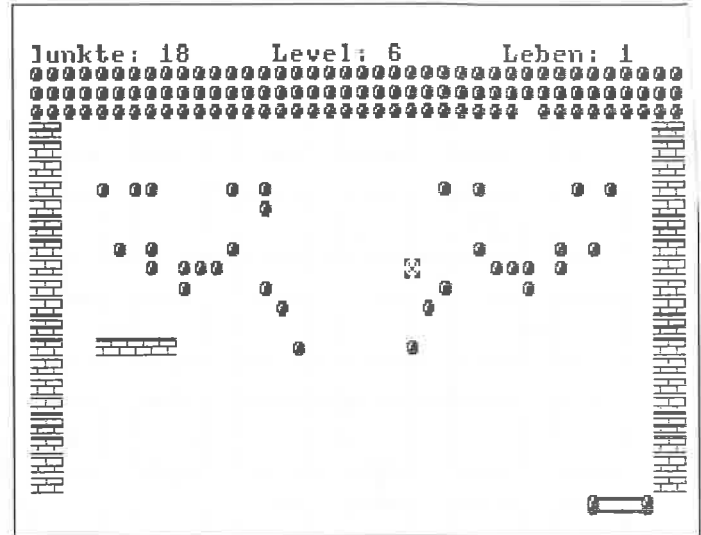


Bild 2: Mauern und Bomben verstellen Ihnen den Weg

Freiheit für den Ball!

Break-Out-Version für alle CPC's!

Dieses Spiel ist eines aus der Sparte Break-Out und Kollegen. Maßgebend beteiligt an der Idee zur Programmierung war wahrscheinlich das Spiel Arkanoid. Ziel ist es, eine Mauer zu durchbrechen, um in einen höheren Level zu gelangen.

Verschiedene Hindernisse erschweren Ihnen den Weg zum Ziel, das Ihnen gesetzt wird. So gibt es z.B. plötzliche Richtungsänderungen, tödliche Bomben und einige andere unangenehme Überraschungen.

Programmanleitung

Nachdem das Programm abgetippt und abgespeichert wurde, kann Ball-Cracker gestartet werden. Für die Besitzer eines CPC 464 allerdings ist es notwendig, vor dem Start den EMULATOR aus Sonderheft 4/87 zu laden und zu starten. Nach dem Start erscheint ein kleines Menü, indem Sie das Startlevel, (insgesamt 10) sowie auch die Anzahl der Leben (insgesamt 20) einstellen können. Zur Anwahl benutzen Sie bitte den Joystick mit den Funktionen UP und DOWN. Die Steuerung des Spiels erfolgt ebenfalls über den Joystick oder über ein Joystick-Simulationsprogramm, das die Cursortasten abfragt, z.B. aus dem Sonderheft 4/87 oder aus dem Heft 12/87.

Erklärung des Spiels

Versuchen Sie so lange wie möglich den Ball in Bewegung zu halten, also den Ball mit dem Schläger zu treffen und gegen die Mauer prallen zu lassen. Hier nun eine Erklärung der Hindernisse, die auf dem Bildschirm zu sehen sind:

- Eine Bombe läßt den Ball explodieren und ein Leben wird abgezogen.
- Ein M läßt irgendwo eine Mauer erscheinen (unzerstörbar). Ball prallt ab.
- Ein Wegweiser läßt den Ball in eine zufällige Richtung abprallen.
- Ein roter Ball gibt ein Freileben.
- Ein Doppelpfeil ändert die Ballrichtung auf 0 Grad.

- Ein Mosaikstein gibt Punkte (je höher der Level, desto mehr Punkte)

Die Variablenliste

leben	= Anzahl der Leben
level	= Spielstufe
pun	= Punkte
hi	= Hi-Score
s	= Schlägerkoordinaten
brs	= Ballrichtung Spalte
brz	= Ballrichtung Zeile
bs,bz	= Ballkoordinaten
ms,mz	= Mosaikkordinaten
mf	= Mosaiksteinfarbe
mauer	= Entscheidet ob ein 'M' auf den Bildschirm ist.
dpfeil	= Entscheidet ob ein Doppelpfeil auf dem Bildschirm erscheint
zleben	= Entscheidet ob ein Freiball auf dem Bildschirm erscheint
wweiser	= Entscheidet ob ein Wegweiser auf dem Bildschirm erscheint
bombe	= Entscheidet ob sich eine Bombe auf dem Bildschirm befindet
richt	= Richtung Wegweiser
anz,zeit,	= Schleifenvariablen
vers,pe	
beru\$	= enthält das ASCII-Zeichen das gerade berührt wird

(R.v.Terwingen/JB)

FÜR 464-664-6128

```

10 DEFINT a-z          [553]
20 ' *****          [1285]
30 ' *                * [175]
40 ' *                * [460]
   '          BALL CRACKER
50 ' *                * [175]
60 ' *                * [1236]
   '          Copyright by R.v.T
70 ' *                * [175]
80 ' *****          [1285]
90 'LAUFT AUF 464 MIT EMULATOR [115]
100 '                 [117]
110 ' ***** GERAUSCHE ***** [1432]
120 '                 [117]
130 ENV 1,11,-1,25    [350]
140 ENV 2,15,-1,1     [481]
150 ENT 1,9,49,5,9,-10,15 [1126]
160 ENT 2,50,-1,1    [590]
170 '                 [117]
180 ' ***** ZEICHEN ***** [1045]
Listing Ball-Cracker
    
```



```

190 ' [117]
200 SYMBOL AFTER &7A [1137]
210 SYMBOL &7B,&78,&DF,&BC,&BC,&BC,&FC,&FF [2068]
, &78
220 SYMBOL &7C,&0,&FF,&0,&0,&0,&0,&FF,&0 [838]

230 SYMBOL &7D,&1E,&F7,&2F,&2F,&3F,&3F,&FF [2217]
, &1E
240 SYMBOL &7E,&3C,&6E,&DF,&FF,&FF,&FF,&7E [2901]
, &3C
250 SYMBOL &7F,&0,&3C,&6E,&5E,&5E,&7E,&3C, [1519]
&0
260 SYMBOL &81,&FF,&1,&1,&FF,&10,&10,&FF,& [2077]
0
270 SYMBOL &82,&0,&41,&63,&55,&49,&41,&41, [1848]
&0
280 SYMBOL &84,&78,&DC,&BC,&BC,&FC,&FC,&FC [2903]
, &78
290 SYMBOL &86,&6,&9,&3C,&6E,&DF,&FF,&7E,& [2546]
3C
300 SYMBOL &87,&E7,&C3,&A5,&18,&18,&A5,&C3 [2340]
, &E7
310 SYMBOL &88,&2,&40,&22,&0,&2,&24,&12,&8 [2177]
1
320 SYMBOL &89,&1,&60,&0,&C,&45,&0,&24,&81 [1512]

330 hi=1000 : pun=100 [543]
340 ' [117]
350 ' ***** MENUE ***** [475]
360 ' [117]
370 leben=1 : level=1 [530]
380 MODE 1 [506]
390 BORDER 0 [1008]
400 FOR pe=0 TO 3:INK pe,0:NEXT [1209]
410 PLOT 320,399:DRAW 470,299:DRAW 170,299 [5581]
:DRAW 320,399:DRAW 170,299:DRAW 170,249:DR
AW 470,249:DRAW 470,299
420 DRAW 638,399:DRAW 638,349:DRAW 470,249 [1914]
430 PLOT 170,300 [596]
440 DRAW 2,399:DRAW 2,350:DRAW 170,250 [979]
450 PLOT 2,399:DRAW 639,399 [744]
460 PLOT 2,350:DRAW 170,50 [487]
470 PLOT 639,350:DRAW 470,50 [911]
480 IF pun>hi THEN hi=pun [865]
490 LOCATE 15,8:PEN 3:PRINT"BALL CRACKER" [1791]

500 LOCATE 14,9:PRINT"||||||||||||":PEN [3154]
1
510 PLOT 170,250:DRAW 170,50:DRAW 470,50:D [2153]
RAW 470,250
520 LOCATE 14,13:PRINT"STARTLEVEL:":level [2525]

530 LOCATE 14,15:PRINT"STARTLEBEN:":leben [851]

540 LOCATE 14,20:PEN 2:PRINT"HISCORE:":hi [2467]

550 LOCATE 13,24:PEN 1:PRINT"LAST SCORE:": [2473]
pun
560 FOR anz=1 TO 15:ms=INT(RND*7)+1:mz=INT [7672]
(RND*5)+20:mf=INT(RND*3)+1:PEN mf:LOCATE m
s,mz:PRINT CHR$(127):LOCATE 41-ms,mz:PRINT
CHR$(127):NEXT
570 INK 1,26:INK 2,15:INK 3,3 [1484]
580 WHILE JOY(0)<16:IF JOY(0)=1 THEN level [5189]
=level+1 ELSE IF JOY(0)=2 THEN level=level
-1
590 IF level=0 THEN level=level+1 ELSE IF [3671]
level=11 THEN level=level-1
600 PEN 1:LOCATE 14,13:PRINT CHR$(24)"STAR [7132]
TLEVEL:":level;CHR$(24):WEND:LOCATE 14,13:
PRINT "STARTLEVEL:":level;" "
610 FOR zeit=1 TO 500:NEXT [2536]
620 WHILE JOY(0)<16:IF JOY(0)=1 THEN leben [4386]
=leben+1 ELSE IF JOY(0)=2 THEN leben=leben
-1
630 IF leben=0 THEN leben=leben+1 ELSE IF [3190]
leben=21 THEN leben=leben-1
640 PEN 1:LOCATE 14,15:PRINT CHR$(24)"STAR [7304]
TLEBEN:":leben;CHR$(24):WEND:LOCATE 14,15:
PRINT "STARTLEBEN:":LEBEN;" "
650 FOR zeit=1 TO 500:NEXT [2536]
660 WHILE JOY(0)<16:LOCATE 14,17:PRINT "PR [4706]
ESS BUTTON":WEND
670 LOCATE 1,25 [731]
680 FOR vers=1 TO 25:SOUND 1,100-(3*vers), [2156]
1,15:PRINT:NEXT
690 ' [117]
700 ' ***** BILDSCHIRMAUFBAU ***** [2779]
710 ' [117]
720 bo=0:pun=0:BORDER bo [1172]
730 MODE 1 [506]
740 FOR pe=0 TO 3:INK pe,0:NEXT [1209]
750 LOCATE 16,1:PRINT "Level:":level:LOCAT [4307]
E 1,1:PRINT "Punkte:"
760 FOR anz=1 TO level*5:ms=INT(RND*15)+5: [12310]
mz=INT(RND*12)+6:mf=INT(RND*3)+1:LOCATE ms
Listing Ball-Cracker

```

```

,mz:PEN mf:PRINT CHR$(127):LOCATE 41-ms,mz
:PEN mf:PRINT CHR$(127):NEXT
770 PEN 3:FOR smaue=4 TO 23:LOCATE 1,smau [8953]
er:PRINT CHR$(129);CHR$(129):LOCATE 39,sma
uer:PRINT CHR$(129);CHR$(129):NEXT:PEN 1

780 mauer=INT(RND*2)+1:IF mauer=1 THEN LOC [5887]
ATE INT(RND*35)+3,INT(RND*15)+6:PRINT CHR$(
130)
790 dpfeil=INT(RND*5)+1:IF dpfeil=1 THEN L [5021]
OCATE INT(RND*35)+3,INT(RND*15)+4:PRINT CH
R$(254)
800 zleben=INT(RND*4)+1:IF zleben=1 THEN P [5689]
EN 3:LOCATE INT(RND*33)+3,INT(RND*15)+4:PR
INT CHR$(126)
810 FOR anz=0 TO level:wweiser=INT(RND*2)+ [6445]
1:IF wweiser=1 THEN PEN 1:LOCATE INT(RND*3
0)+5,INT(RND*15)+4:PRINT CHR$(135)
820 NEXT [350]
830 FOR anz=0 TO level:bombe=INT(RND*5)+1: [7552]
IF bombe=1 THEN PEN 3:LOCATE INT(RND*30)+5
,INT(RND*15)+4:PRINT CHR$(134)
840 NEXT [350]
850 LOCATE 1,2:PRINT STRING$(120,127) [1997]
860 LOCATE 30,1:PEN 1:PRINT "Leben:":leben [3191]
:bz=22:bs=3:brz=-1:brs=-1
870 LOCATE 1+s,24:PRINT " {}{}{} " [1630]
880 IF level>4 THEN PEN 3:LOCATE 5,16:PRIN [3042]
T STRING$(5,129):PEN 1
890 INK 1,26:INK 2,15:INK 3,3 [1484]
900 ' [117]
910 ' ***** SPIELSTART ***** [1125]
920 ' [117]
930 IF leben=0 THEN 1160 [834]
940 PEN 1 [549]
950 j=JOY(0) [892]
960 IF j=0 THEN 1020 [1183]
970 IF j=4 THEN s=s-2 [719]
980 IF j=8 THEN s=s+2 [543]
990 IF s=34 THEN s=s-2 [1213]
1000 IF s=-2 THEN s=s+2 [331]
1010 LOCATE 1+s,24 : PRINT " {}{}{} " [1630]
1020 IF bz=23 AND bs>s AND bs<7+s THEN brz [8321]
=-brz/-1 ELSE IF bz=24 THEN LOCATE bs+brz,2
4:PRINT" ":SOUND 1,800,20,10:leben=leben-1
:GOTO 860
1030 IF bs=37 THEN LOCATE bs+1,bz:PRINT CH [3618]
R$(132) :SOUND 1,800,1,15:brs=brs+1/-1
1040 IF bs=2 THEN LOCATE bs+1,bz:PRINT CHR [5271]
$(132):SOUND 1,800,1,15:brs=brs-1/-1
1050 IF bz=1 THEN level=level+1:bo=bo+1:BO [5971]
RDER bo:LOCATE 1,1:FOR vers=1 TO 25:PRINT
CHR$(11);CHR$(11):NEXT:GOTO 730
1060 LOCATE bs+brs+1,bz+brz:beru$=COPYCHR$ [11762]
(#0):IF beru$=CHR$(127) THEN SOUND 1,100,1
,15:LOCATE bs+brs+1,bz+brz:PRINT" ":brz=br
z/-1:pun=pun+level:LOCATE 8,1:PRINT pun:GO
TO 1060
1070 IF beru$=CHR$(130) THEN LOCATE INT(RN [5051]
D*10)+5,INT(RND*10)+5:PEN 3:PRINT STRING$(
20,129):PEN 1:GOTO 1140
1080 IF beru$=CHR$(129) THEN SOUND 1,800,1 [3063]
,15:brz=brz/-1:GOTO 950
1090 IF beru$=CHR$(254) THEN LOCATE bs+brs [6458]
+1,bz+brz:PRINT" ":brs=0:brz=brz/1:GOTO 95
0
1100 IF beru$=CHR$(126) THEN SOUND 1,60,50 [6538]
,15,0,2:leben=leben+1:LOCATE 30,1:PRINT "L
eben:":leben:GOTO 1140
1110 IF beru$=CHR$(134) THEN leben=leben-1 [13914]
:LOCATE bs,bz:PRINT" ":SOUND 2,145,255,0
,1,1,12:LOCATE bs+brs,bz+brz:PRINT CHR$(13
6);CHR$(137):LOCATE bs+brs,bz+brz+1:PRINT
CHR$(137);CHR$(136):GOTO 1120 ELSE 1130

1120 FOR zeit=1 TO 200:NEXT:LOCATE bs+brs, [6230]
bz+brz:PRINT" ":LOCATE bs+brs,bz+brz+1:P
RINT" ":GOTO 860
1130 IF beru$=CHR$(135) THEN SOUND 1,300,1 [13079]
5,15,2:richt=INT(RND*4)+1:IF richt=1 THEN
brs=1:brz=1:GOTO 1060 ELSE IF richt=2 THEN
brs=-1:brz=-1:GOTO 1060 ELSE IF richt=3 T
HEN brs=1:brz=-1:GOTO 1060 ELSE brs=-1:brz
=1:GOTO 1060
1140 LOCATE bs+1,bz:PRINT" ":bz=bz+brz:bs= [6220]
bs+brs:LOCATE bs+1,bz:PRINT" ":GOTO 950

1150 GOTO 950 [318]
1160 ' ***** GAME OVER ***** [1273]
1170 LOCATE 16,12:PRINT"GAME OVER" [579]
1180 LOCATE 14,14:PRINT"PRESS BUTTON" [2337]
1190 WHILE JOY(0)<16:WEND [1816]
1200 LOCATE 1,1 [611]
1210 FOR vers=1 TO 25:SOUND 1,100+(3*vers) [3915]
,1,15:PRINT CHR$(11);CHR$(11):NEXT:GOTO 35
0
Listing Ball-Cracker

```

O L A L A

SOFTWARE IN FRANKREICH



Foto: Benefici

Meinen diesjährigen Frankreichurlaub nutzte ich diesmal, um einmal den französischen Softwaremarkt zu sondieren. So vertauschte ich also von Zeit zu Zeit das Handtuch mit dem Joystick und den Strand mit der Spielhalle. Meine Hoffnungen wurden weit übertroffen: Die CPC's sind die definitive Nummer Eins in Frankreich! Das merkte ich schon bei meinem ersten Besuch in einem Zeitschriftenladen. Dort fand ich allein fünf Hefte für den Schneider (der in Frankreich AMSTRAD heißt), ein Heft für MS-DOS Computer und ein Heft für den MSX, der dort den ungefähren Stellenwert unseres Spectrums einnimmt.

Da ich mit Anwendungen nicht allzuviel am Hut habe, beschränke ich mich bei den späteren Softwarebesprechungen auf mein Spezialgebiet, die Spiele. Zuerst einmal zu den Spielhallen. Da Frankreich selber keine Automatenproduzenten hat, werden hier allgemein Automaten aus England oder Amerika eingekauft. Die Renner dieses Jahres unterschieden sich nicht allzusehr von den deutschen: Schnelle Prügel- und Schießspiele, die über eine fantastische Grafik verfügen, und bei denen man möglichst wenig nachdenken muß. Besonders dicht umhangen war der IRON HORSE-Automat, der meines Wissens schon einige Zeit auf dem Markt ist. Hierbei geht es darum, daß ein Cowboy einen Zug überqueren muß, und währenddessen von ganzen Armeen von Feinden angegriffen wird. Zur Abwehr stehen drei verschie-

dene Waffen zur Verfügung. Da dieses Spiel von Konami stammt, wird es bestimmt bald von Imagine für den Schneider konvertiert umgesetzt werden.

Am JACKAL-Automaten war auch nur unter größerer Kraftanstrengung ein Sitzplatz zu bekommen. Sitzplatz? Jawohl, dieser Automat verfügt über eine hervorragende Zwei-Spieler-Option. Bei der wenig einfallsreichen Hintergrundstory geht es darum, möglichst viele Soldaten aus den Camps des Feindes zu bekommen. Den beiden furchtlosen Recken steht jeweils ein schwerbewaffneter Jeep zur Verfügung. Der Automat ist an sich nicht schlecht gemacht, nach meinem Geschmack aber etwas zu brutal.

Genauso beliebt wie in Deutschland waren die beiden Coin-Op's RUN OUT und ARKANOID, die ich vor kurzem mit einem Freund ausführlich testete. Hierbei ging eine ganze Menge Geld drauf, was für den Unterhaltungswert der Automaten spricht.

Doch nun zu dem eigentlichen Thema unseres Berichtes: den Programmen. Unter dem Gros der französischen Softwareproduzenten tun sich besonders vier hervor: INFOGRAMMES, UBI-SOFT, ERE INFORMATIQUE & LORICIELS. Jede dieser Firmen landete schon mindestens einen Hit in deutschen Landen, und so warten schon viele Freaks auf Neuerscheinungen.

Von INFOGRAMMES, dem Softwarehaus, das ich am besten finde, kommt eine Flut von neuen Programmen. Eines der interessantesten Produkte dieser Firma ist BIVOUAC, bei dem sich alles rund/um/über/unter einem Berg dreht. Als Held dieses Spiels muß man den namensgebenden Biwak erklimmen, Gebirgswandern und einiges mehr. Das Spiel verfügt über eine hervorragende Grafik, die nicht von ungefähr kommt: Die sogenannte »10e Art«, das Digitalisieren, wurde hier verwendet. Eine nette Alternative zu anderen Winterolympiaden.

Weiterhin fiel mir das Spiel OMEGA PLANETE INVISIBLE auf. Die Grafik kam mir irgendwie bekannt vor, denn OMEGA ist der Nachfolger zu MANDRAGORE, dem ersten Rollenspiel für die CPC Reihe. Im großen und ganzen hat sich nicht besonders viel geändert. Es gibt neue Charaktere und die Grafiken (die immer noch den eigentlichen Schwachpunkt des Programmes bilden), wurden noch einmal überarbeitet. Alles in allem recht empfehlenswert.

Den dritten im Bunde bildet MARCHE A L'OMBRE. Der Held dieses Spiels ist das französische Equivalent zu Werner. Renaud fährt gerne Motorrad (die Dinger sehen nach allem aus, aber nicht nach Motorrad!), kippt gern ein paar Kisten Bier und vereimert mit Vorliebe Polizisten. Das ganze läuft als animierter Cartoon ab, bei dem auch mit absurden Einfällen nicht gespart wurde. Glücklicherweise ist das Spiel wesentlich schwieriger

als die grafisch ausgereifere, aber leider zu leichte Comic-Adaption.

REISENDE IM WIND

Zu diesem Spiel wurde bereits ein zweiter Teil veröffentlicht, der sinnigerweise REISENDE IM WIND 2 heißt. Das Spiel bezieht sich auf Band 4 der Comicausgabe und schlägt den Vorgänger grafisch noch um Längen.

Auch UBI-SOFT, die in Deutschland erst durch ZOMBI und MASQUE bekannt wurden, waren nicht untätig. Sie blieben ihrem alten Erfolgsrezept »hochauflösende Grafik und Icon-Steuerung« treu. Doch bei dem Mega-Spiel FER ET FLAMME kommen zusätzlich Rollenspielelemente und einige Farben hinzu. Die Komplexität wird schon dadurch gesichert, daß das Spiel auf zwei doppelseitig bespielten Disketten geliefert wird.

Mit INERTIE bringt auch diese Firma einen Beitrag zur momentan wieder aktuell gewordenen Raumschiff-Thematik. Doch wie für UBI-SOFT üblich, verlagert es sich vom Ballern wieder einmal auf das Nachdenken. Man schwebt mit einem kugelförmigen Raumschiff durch eine Höhle und sammelt Astronauten ein. Doch es treiben sich nicht nur Astronauten in der Mine herum, auch Saboteure machen die Gegend unsicher, die, sobald man sie an Bord nimmt, das Schiff fein säuberlich zerlegen. Man muß also zwischen Gut und Böse unterscheiden, was nicht immer ganz einfach ist. Auch dieses Spiel kann dem, der lieber zweimal nachdenkt als einmal schießt, nur wärmstens empfohlen werden.

Der nächste Hersteller, den wir uns genauer ansehen, ist zweifellos der kreativste. ERE INFORMATIQUE ist das erste Softwarehaus, das bei der Entwicklung eines Theaterstücks mitwirkt, und dazu prompt ein Spiel programmiert.

Es dreht sich dabei um einen kleinen Helden namens OXPBAR, der die siebte Stufe der Vollkommenheit zu erreichen versucht.

Das wird ihm allerdings durch diverse Bösewichter und Monster erschwert. Bei allen Kreaturen erkennt man die Handschrift von Michel Rho, dem Grafiker von ERE. Diese Firma hat inzwischen den Adventuremarkt für sich entdeckt und produziert eifrig Spiele.

Programme wie SRAM I+II, HARRY ET HARRY und LES PASSAGERS DU TEMPS weisen alle einen interessanten Spielverlauf und gute Grafiken auf.

Besonders gefiel mir LES PASSAGERS DU TEMPS (Die Zeitreisenden), da man bei diesem Spiel eine Piratenkatze spielt, die stilecht mit Augenklappe und Kopftuch ausgestattet ist. Auch zu ihrem Debutprogramm CRAFTON & XUNK ist der Nachfolger, der CRAFTON 2 heißen soll, fast fertig.

Last but not least: LORICIELS.

Sie denken, nach all diesen interessanten Spielen kann doch nichts tolles mehr kommen. Doch das stimmt bei weitem nicht, denn das beste Spiel, das ich dieses Jahr in Frankreich sah, stammte von LORICIELS. Das gute Stück nennt sich RELIEF ACTION und ist ein WALKTHRU GAME. Hierbei sieht man alles aus der Sicht der Spielfigur. Nichts besonderes werden Sie sagen. ELITE, TAU CETI, MERCENARY usw. unterscheiden sich im großen und ganzen auch nicht davon. Doch was würden Sie sagen, wenn man den Sternenhintergrund mit einer vollmöblierten Station vertauscht? Genau darum geht es bei diesem Spiel nämlich, das neben einer sehr schnellen Grafik noch ein besonderes Extra bietet: Wählt man nämlich den Menüpunkt RELIEF nicht an, werden aus den hellen Strichen rote und grüne, die, wenn man eine entsprechende Brille aufsetzt, zu einer hervorragenden, plastischen Grafik verschmelzen.

Weiterhin kommt von LORICIELS noch ein schnelles Actionspiel namens FLASH, bei dem man wieder einmal einen furchtlosen Soldaten spielt. Das Spiel besteht aus drei Teilen. Im ersten läuft man als Soldat durch die Gegend, beim zweiten darf man einen Panzer benutzen und beim dritten schließlich mit einem Helikopter durch die Lüfte schweben.

COSA NOSTRA ist schließlich ein Spiel, bei dem man ein Mitglied des gleichnamigen Kinderchors spielt, der Chef der Unterwelt werden will. Hierzu muß man möglichst viele gleichgesinnte finden oder nicht gleichgesinnte mit einer Feuerspritze oder ähnlichen Gegenständen, die einfach so auf der Straße liegen, umzustimmen.

Fazit:

Neben England ist Frankreich zweifellos der aktivste Softwaremarkt für den Schneider Computer, wobei ich sogar sagen möchte, daß französische Software die englische meist um Längen schlägt.

(JH/HS)



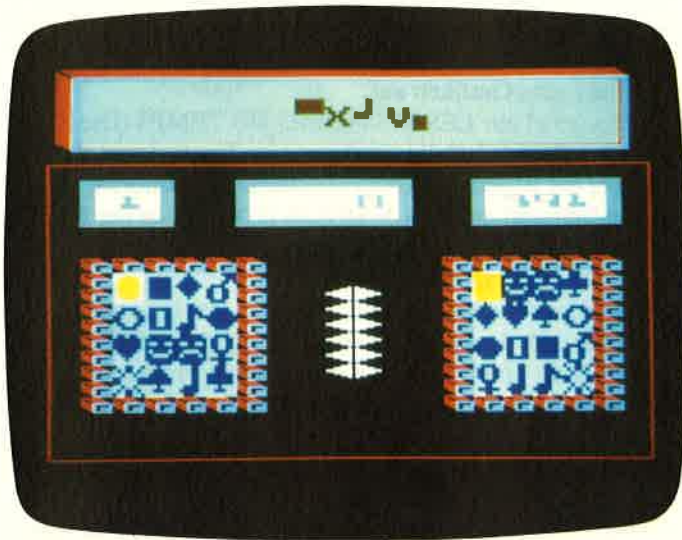


Bild 1: Das Spielfeld von SHIFT

SHIFT

Das englische Wort "shift" bedeutet auf deutsch übersetzt soviel wie "verschieben", "wechsell", aber auch "Kniff" oder "List". Und die brauchen Sie schon, wenn Sie sämtliche 6 Level meistern wollen.

"SHIFT" ist eine Computeradaption der bekannten Logikknochelei "Schiebefix", jedoch mit einigen "aufregenden" Features. Um alles verständlich zu erklären, spielen wir am besten einmal eine Partie durch! Nachdem das Programm gestartet wurde, befinden wir uns im Titelbild mit seinem Menue. Während die Titelmusik läuft, kann man zwischen den drei Punkten:

- Difficulty 1
- Difficulty 2
- Ende

mit Hilfe des Pfeils wählen. Steuern Sie den Pfeil mit dem Joystick zum gewünschten Punkt und drücken Sie FEUER. Haben Sie sich für Difficulty 1 entschieden, wird von Ihnen höchste Spielkunst gefordert, denn die Stoppuhr wird auf 250 Sekunden gesetzt (was das bedeutet, werden Sie spätestens dann erfahren, wenn Sie zum erstenmal mit dieser Zeit spielen). Bei Difficulty 2 bekommen Sie 400 Sekunden Zeit (weit über sechs Minuten). Wenn Sie das Spiel jedoch beenden wollen (ein übrigens völlig unverständliches Verhalten), steuern Sie den dritten Punkt an. Die aktuelle High-Score-Tabelle wird automatisch abgespeichert. Nun denn, nun wollen wir aber spielen! Sogleich erscheinen auf unserem "CTM 640", "CTM 644", oder was auch sonst vor unseren Augen flimmert, zwei Fenster und darüber drei Anzeigen. In der linken Anzeige steht die Nummer des aktuellen Levels, rechts daneben unser Score, und noch ein Stückchen rechter die schon oben erwähnte Zeit in Sekunden. Im rechten Fenster erscheint dann sogleich ein geordnetes Motiv aus vier mal vier Feldern.

Im linken Fenster erscheint dieses Motiv dann zwar schließlich auch, jedoch ein wenig "chaotisiert". Nun haben Sie, als Spieler, die Aufgabe, das linke Motiv dem rechten durch Verschieben der einzelnen Felder gleich zu machen - in der vorgegebenen Zeit! Wie beim guten alten "Schiebefix" können Sie ein Feld natürlich nur auf das Leerfeld schieben. Und das machen Sie mit Hilfe des schwarz/weiß blinkenden Cursors, den Sie mit dem Joystick steuern. Setzen Sie den Cursor nun auf das Feld, das Sie verschieben wollen, und drücken Sie FEUER. Das Feld geht dann automatisch auf das Leerfeld. Handeln Sie aber schnell, denn die Zeit wartet nicht auf Sie! Jeder, der schon einmal ein "Schiebefix" in der Hand hielt, weiß jedoch, daß ein normaler Mensch kaum in der Lage ist, dieses Spiel innerhalb von wenigen Minuten zu lösen. Da Sie es aber MÜSSEN, werden ihnen 6 "Changer" zur Verfügung gestellt. Die Anzahl Ihrer noch verbliebenen Changer wird durch die weißen Zeichen zwischen den beiden Windows dargestellt. Ein Changer ermöglicht es, zwei Felder miteinander zu vertauschen, zu wechseln. Und

das geht so: Setzen Sie den Cursor auf das Feld, das Sie vertauschen wollen und drücken Sie gleichzeitig FEUER und in die Richtung, in der sich das Feld befindet, das ebenfalls vertauscht werden soll. Schwuppdwupp... die beiden betroffenen Felder haben sich vertauscht und ein weißes Changerzeichen wird grau. Wenn Sie übrigens merken, daß Sie keine Chance mehr haben, das Motiv in der vorgegebenen Zeit zu ordnen, können Sie "A" (für aufgeben, 'A'ufhören, 'A'bbrechen oder 'A'bandon) drücken und das Spiel endet. Ja, es braucht schon einige Übung, bis Sie den ersten Level meistern können. Sollten Sie es aber geschafft haben, winken Ihnen erst einmal einige Punkte! Und zwar:

Level * 200 + Bonus (Bonus = Übriggebliebene Sekunden + übriggebliebene Changer * 100)

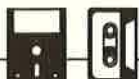
Anschließend kommen Sie in den nächsten Level und die Zeit verringert sich um 10 Sekunden.

Allerdings bekommen Sie nun auch ein neues Motiv vorgesetzt. War es im ersten Level noch das griechische Alphabet (easy), erwarten Sie nun Motive, die einen Kürbis, eine berühmte Comicfigur aus Schleswig-Holstein, Miss Monroe, den Berliner Bären (anlässlich der 750-Jahr-Feier), und das "6.Level"-Muster darstellen. Sie werden schnell feststellen, daß es nun etwas schwieriger wird. Lassen Sie sich jedoch nicht von dem chaotischen Wirrwarr im linken Fenster abschrecken, sondern vergleichen Sie schön mit dem Motiv im rechten Fenster. Sie werden sehen, Ihr Auge trainiert sich mit der Zeit, und irgendwann werden Sie sämtliche Level einmal meistern. Haben Sie nun also ein beachtliches Ergebnis erzielt, können Sie sich in die High-Score-Tabelle eintragen. Das wird ganz komfortabel mit dem Joystick gemacht.

Programmtechnisches

Kurz nachdem das Programm gestartet wurde, lädt es auch schon die in der Datei "SHIFT.DAT" abgespeicherte High-Score-Tabelle. Lassen Sie die Diskette also im Laufwerk stecken, solange das Programm SHIFT läuft. Schließlich wird, nachdem man im Menue "End" wählte, später die aktualisierte High-Score-Tabelle abgespeichert, nachdem die alte von der Diskette gelöscht wurde. Falls auf der Diskette jedoch noch keine Datei "SHIFT.DAT" vorhanden sein sollte, geben Sie einfach das Programm "SHIFTDAT.BAS" ein, und starten Sie es. Es erzeugt automatisch eine korrekte Datei "SHIFT.DAT" und löscht sich anschließend von selbst. (M. Prange / JB)

für 464-664-6128



```

10 ' ***** [13911]
20 ' * [175]
30 ' * ----- SHIFT ----- * [2011]
40 ' * [175]
50 ' * (C)opyright * [457]
60 ' * 1987 by BePpSoft * [1414]
70 ' * [175]
80 ' * written by * [812]
90 ' * Matthias Prange * [179]
100 ' * [175]
110 ' * Thanx to K.P. & H.P. for some * [1010]
120 ' * of their great ideas * [391]
130 ' * [175]
140 ' ***** [13911]
150 ' ***** [1164]
160 ' **** ***** [768]
170 ' *** ***** [665]
180 ' ** ***** [528]
190 ' * ***** * [502]
200 ' ***** [315]
210 ' ***** [51]
220 ' ** [175]
230 ' [117]
240 ' Sounds [821]
250 ENV 1,3,-2,5,1,0,15,6,1,5 [934]
260 ENV 2,6,-1,10:ENT 1,20,-5,1:ENV 3,5,-1 [1329]
,2
270 RANDOMIZE TIME^2 [1617]
280 'Dimensionierungen [2081]
290 CLS: DIM p$(5,5), char$(4,4), hsc(8,2), hn [2214]
ame$(8,2)
300 scc=175:FOR sy=1 TO 4:FOR sx=1 TO 4 [2021]
310 char$(sx,sy)=CHR$(scc) [1020]
320 scc=scc+1 [649]
330 NEXT sx,sy [1045]
340 char$(1,1)=CHR$(32) [906]
350 'High-Scores laden [944]
360 OPENIN "SHIFT.DAT" [1326]
370 FOR hw=1 TO 2:FOR hl=1 TO 8 [1838]
380 INPUT #9,hname$(hl,hw): INPUT #9,hsc(hl [2352]
,hw)
390 NEXT hl,hw:CLOSEIN [857]
400 'Titelbild [797]
410 MODE 0:PAPER 0:BORDER 0 [2015]
420 FOR sco=0 TO 15:INK sco,0:NEXT sco:INK [3173]
7,26,0:SPEED INK 2,2
430 WINDOW #1,2,19,2,5:PAPER #1,3:CLS#1 [1434]
440 WINDOW 1,20,7,25:PEN 7:LOCATE 2,2:PRIN [4845]
T"BePpSoft presents"
450 GOSUB 4170:titelschr$="SHIFT" [2414]
460 FOR sbt=1 TO 5 [704]
470 at=ASC(MID$(titelschr$,sbt,1)) [2632]
480 DEF FNlettert(bt)=PEEK(HIMEM+bt+8*(at- [1846]
PEEK(&B294)))
490 SYMBOL 255, FNlettert(1), FNlettert(1), F [8467]
Nlettert(2), FNlettert(2), FNlettert(3), FNle
ttert(3), FNlettert(4), FNlettert(4)
500 PEN #1,6:LOCATE #1,sbt+7,2:PRINT #1,CH [994]
R$(255)
510 SYMBOL 255, FNlettert(5), FNlettert(5), F [8715]
Nlettert(6), FNlettert(6), FNlettert(7), FNle
ttert(7), FNlettert(8), FNlettert(8)
520 LOCATE #1,sbt+7,3:PRINT #1,CHR$(255) [674]
530 NEXT sbt [573]
540 MOVE 30,384: DRAW 610,384,2: DRAW 610,31 [2563]
8: DRAW 30,318: DRAW 30,384
550 MOVE 26,386: DRAW 614,386: DRAW 614,316: [3857]
DRAW 26,316: DRAW 26,386
560 sdx=22:FOR sdy=318 TO 324 STEP 2 [2279]
570 MOVE sdx,sdy: DRAW 0,70,1: DRAW 590,0 [1236]
580 sdx=sdx-4 [512]
590 NEXT sdy [504]
600 MOVE 22,388: DRAW -16,8,0 [1791]
610 MOVE 606,382: DRAW 606,320,1: DRAW 34,32 [2562]
0

```

Listing SHIFT

```

620 wahl=1 [677]
630 FOR srx=3 TO 18:LOCATE srx,5:GOSUB 250 [4852]
0:LOCATE srx,13:GOSUB 2500:NEXT srx
640 FOR sry=6 TO 12:LOCATE 3,sry:GOSUB 250 [5181]
0:LOCATE 18,sry:GOSUB 2500:NEXT sry
650 PEN 4:LOCATE 7,7:PRINT"Diffic. I" [2131]
660 LOCATE 7,9:PRINT"Diffic. II" [1694]
670 LOCATE 7,11:PRINT"End....." [1228]
680 LOCATE 1,17:PEN 8:PAPER 9:PRINT CHR$(1 [7170]
35);STRING$(18,131);CHR$(139):LOCATE 1,19:
PRINT CHR$(141);STRING$(18,140);CHR$(142);
690 LOCATE 1,18:PAPER 9:PRINT CHR$(133);CH [5784]
R$(164);"1987 by BePpSoft";CHR$(138):PAPE
R 0
700 INK 0,0:INK 1,6:INK 2,14:INK 3,19:INK [3981]
4,24:INK 5,7:INK 6,3:INK 7,0:INK 8,2,14:IN
K 9,14,2:SPEED INK 10,10
710 RESTORE 2590:EVERY 30,0 GOSUB 2530 [2308]
720 PEN 5:LOCATE 5,5+2*wahl:PRINT CHR$(174 [3508]
):FOR sj=1 TO 100:NEXT sj
730 IF JOY(0)=1 AND wahl>1 THEN LOCATE 5,5 [3073]
+2*wahl:PRINT " ":wahl=wahl-1:GOTO 720
740 IF JOY(0)=2 AND wahl<3 THEN LOCATE 5,5 [5398]
+2*wahl:PRINT " ":wahl=wahl+1:GOTO 720
750 IF JOY(0)<>16 THEN 720 [1450]
760 stev=REMAIN(0) [1369]
770 ON wahl GOTO 780,790,800 [1044]
780 zeit=250:GOTO 900 [1646]
790 zeit=400:GOTO 900 [1144]
800 FOR send=1 TO 30 [1402]
810 LOCATE 1,1:PRINT CHR$(11):NEXT send [1400]
820 'High-Scores abspeichern [1144]
830 eras$="SHIFT.DAT": ERA,@eras$:OPENOUT [4004]
"SHIFT.DAT"
840 FOR hw=1 TO 2:FOR hs=1 TO 8 [1532]
850 PRINT #0,hname$(hs,hw):hsc(hs,hw):PRIN [3246]
T #9,hname$(hs,hw):PRINT #9,hsc(hs,hw)
860 NEXT hs,hw:CLOSEOUT [655]
870 MODE 1:CALL &BC02:PEN 1:PAPER 0:SYMBOL [2944]
AFTER 32
880 END [110]
890 'Maske [1033]
900 CLS:INK 4,15:INK 5,20:INK 7,26:INK 8,1 [3935]
3:INK 9,26,0:INK 10,0,26:SPEED INK 6,6
910 PEN 8:PAPER 7:LOCATE 2,2 [1470]
920 PRINT CHR$(135);CHR$(131);CHR$(139);CH [6465]
R$(9);CHR$(9);CHR$(135);STRING$(4,131);CHR
$(139);CHR$(9);CHR$(9);CHR$(135);STRING$(3
,131);CHR$(139)
930 LOCATE 2,3:PRINT CHR$(133);CHR$(32);CH [7724]
R$(138);CHR$(9);CHR$(9);CHR$(133);SPACE$(4
);CHR$(138);CHR$(9);CHR$(9);CHR$(133);SPAC
E$(3);CHR$(138)
940 LOCATE 2,4:PRINT CHR$(141);CHR$(140);C [10495]
HR$(142);CHR$(9);CHR$(9);CHR$(141);STRING$(
4,140);CHR$(142);CHR$(9);CHR$(9);CHR$(141
);STRING$(3,140);CHR$(142)
950 WINDOW #2,3,6,14,21:PAPER #2,4:CLS#2 [1931]
960 WINDOW #3,15,18,14,21:PAPER #3,4:CLS#3 [2597]
970 FOR srx=2 TO 7:LOCATE srx,7:GOSUB 2500 [4525]
:LOCATE srx,16:GOSUB 2500:NEXT srx
980 FOR sry=8 TO 15:LOCATE 2,sry:GOSUB 250 [4447]
0:LOCATE 7,sry:GOSUB 2500:NEXT sry
990 FOR srx=14 TO 19:LOCATE srx,7:GOSUB 25 [3651]
00:LOCATE srx,16:GOSUB 2500:NEXT srx
1000 FOR sry=8 TO 15:LOCATE 14,sry:GOSUB 2 [7004]
500:LOCATE 19,sry:GOSUB 2500:NEXT sry
1010 MOVE 0,302: DRAW 639,302,1: DRAW 639,0: [3444]
DRAW 0,0: DRAW 0,302
1020 ' [117]
1030 score=0:level=1:changer=6:FOR spy=1 T [8557]
O 5:FOR spx=1 TO 5:p$(spx,spy)=CHR$(0):NEX
T spx,spy
1040 PEN 8:PAPER 7:LOCATE 3,3:PRINT USING" [8087]
#";level:LOCATE 8,3:PRINT USING"####";scor
e:LOCATE 16,3:PRINT USING"###";zeit
1050 PAPER 0:FOR sch=9 TO 14:LOCATE 10,sch [5171]
:PEN 7:PRINT CHR$(247);CHR$(246):NEXT sch
1060 'Symbols fuer die verschiedenen Motiv [1635]
e
1070 ON level GOTO 1890,1080,1240,1400,156 [2408]
0,1720

```

Listing SHIFT

```

1080 SYMBOL 176, &X0, &X0, &X0, &X0, &X0, &X1, &X [3173]
11, &X11
1090 SYMBOL 177, &X0, &X0, &X0, &X1111100, &X11 [3487]
000011, &X10111111, &X1000000, &X1000000
1100 SYMBOL 178, &X0, &X0, &X0, &X0, &X0, &X1001 [3186]
0000, &X1100000, &X0
1110 SYMBOL 179, &X0, &X0, &X1, &X110, &X1001, & [3278]
X10001, &X100001, &X100001
1120 SYMBOL 180, &X11, &X1111111, &X11000011, & [5613]
X1100, &X10010000, &X11100000, &X11111000, &X1
1111110
1130 SYMBOL 181, &X1000000, &X11111110, &X1100 [4781]
0001, &X10110000, &X10001011, &X10001111, &X10
111111, &X11111111
1140 SYMBOL 182, &X0, &X0, &X10000000, &X11000 [3029]
00, &X10000, &X1000, &X100, &X100
1150 SYMBOL 183, &X1000001, &X1000001, &X1000 [3396]
001, &X10000010, &X10000010, &X10000010, &X100
00010, &X1000010
1160 SYMBOL 184, &X11111000, &X11100001, &X10 [3584]
000001, &X11, &X11, &X111, &X111, &X1111
1170 SYMBOL 185, &X1111111, &X1001111, &X1000 [4746]
011, &X10100000, &X10100000, &X11100000, &X111
00000, &X11100000
1180 SYMBOL 186, &X10, &X10, &X10000001, &X100 [4563]
00001, &X10000001, &X10000001, &X10000010, &X1
0000010
1190 SYMBOL 187, &X1000001, &X1000001, &X1000 [3181]
00, &X100000, &X10000, &X1110, &X1, &X0
1200 SYMBOL 188, &X1111, &X0, &X10000000, &X10 [4763]
000000, &X1000000, &X110000, &X11001100, &X111
111
1210 SYMBOL 189, &X11100001, &X1000001, &X100 [4440]
0010, &X10000010, &X10000100, &X10011000, &X11
100111, &X11111000
1220 SYMBOL 190, &X100, &X100, &X1000, &X10000 [4140]
, &X100000, &X11000000, &X0, &X0
1230 GOTO 1890 [361]
1240 SYMBOL 176, &X0, &X100100, &X10010, &X100 [3896]
1, &X100, &X100, &X0, &X1110
1250 SYMBOL 177, &X10000000, &X10000000, &X100 [5919]
0000, &X100000, &X10100000, &X10100000, &X1000
0000, &X1110000
1260 SYMBOL 178, &X111000, &X111000, &X101000 [4095]
, &X101000, &X1000100, &X1110100, &X1110100, &X
1110100
1270 SYMBOL 179, &X0, &X0, &X0, &X0, &X111100, & [3978]
X1000010, &X10011001, &X10111101
1280 SYMBOL 180, &X10001, &X101100, &X101100, [4044]
&X100000, &X10000, &X1111000, &X10000111, &X10
0
1290 SYMBOL 181, &X10001000, &X11100100, &X11 [4525]
100100, &X10000100, &X10000100, &X10001110, &X
1110001, &X100000
1300 SYMBOL 182, &X1000100, &X1111100, &X0, &X [4491]
0, &X111100, &X1000010, &X10011001, &X10111101
1310 SYMBOL 183, &X10111110, &X1011100, &X111 [4213]
000, &X1000, &X10000, &X10000, &X1000, &X1100
1320 SYMBOL 184, &X100, &X1000, &X1000, &X1000, &X1000 [3471]
0, &X100000, &X100000, &X1000000, &X1001000
1330 SYMBOL 185, &X10000, &X10000, &X10000, & [3736]
X1000, &X100, &X100, &X10, &X10
1340 SYMBOL 186, &X1111101, &X111010, &X11100 [3840]
, &X10000, &X1000, &X1000, &X1000, &X110000
1350 SYMBOL 187, &X11, &X10, &X10, &X1, &X0, &X0 [3286]
, &X1, &X1110
1360 SYMBOL 188, &X11001000, &X100100, &X1000 [4163]
11, &X11111000, &X111, &X111000, &X11000111, &X
111111
1370 SYMBOL 189, &X11, &X100, &X100, &X11111, & [4897]
X11110000, &X1110, &X11100001, &X11111100
1380 SYMBOL 190, &X11000000, &X1000000, &X100 [3929]
0000, &X10000000, &X0, &X0, &X11000000, &X11100
0
1390 GOTO 1890 [361]
1400 SYMBOL 176, &X10111, &X101100, &X1011000 [5470]
, &X10000011, &X11111, &X11111110, &X1000000, &
X1000000
1410 SYMBOL 177, &X11000100, &X111011, &X1000 [4255]
10, &X1001100, &X10010000, &X11111101, &X10, &X
1
1420 SYMBOL 178, &X0, &X0, &X11000000, &X11100 [3847]

```

Listing SHIFT

```

000, &X1100000, &X100000, &X10110000, &X100110 [3173]
00
1430 SYMBOL 179, &X11101, &X10111, &X110111, & [4724]
X1011111, &X11001111, &X10111111, &X1101111, &
X1101111
1440 SYMBOL 180, &X100000, &X10100000, &X1110 [4679]
0000, &X10000000, &X10111100, &X11000110, &X11
, &X1111101
1450 SYMBOL 181, &X0, &X0, &X0, &X0, &X11111, &X [2977]
100000, &X0, &X11110
1460 SYMBOL 182, &X11001100, &X1001110, &X101 [3841]
011, &X100111, &X10110, &X10010101, &X10001, &X
10010
1470 SYMBOL 183, &X10011111, &X1101111, &X101 [4818]
11111, &X10011111, &X1100111, &X1010111, &X11
111, &X1001111
1480 SYMBOL 184, &X11101100, &X0, &X0, &X10000 [4068]
000, &X10000010, &X10000001, &X11000000, &X110
00011
1490 SYMBOL 185, &X101111, &X0, &X0, &X0, &X0, & [3533]
X10000000, &X0, &X1100000
1500 SYMBOL 186, &X10010011, &X10101, &X10101 [4088]
, &X10010, &X100010, &X100010, &X100010, &X1001
00
1510 SYMBOL 187, &X1001011, &X110111, &X10110 [3535]
1, &X11011, &X10100, &X1001, &X1010, &X1100
1520 SYMBOL 188, &X11100111, &X11101100, &X11 [5304]
100111, &X11100011, &X11110000, &X11111000, &X
1111000, &X11111111
1530 SYMBOL 189, &X11110000, &X11000, &X11110 [4108]
000, &X11100000, &X1, &X11, &X1101, &X11100001
1540 SYMBOL 190, &X1001000, &X1101000, &X1010 [5588]
1000, &X10101000, &X10011000, &X11010000, &X11
00000, &X100000
1550 GOTO 1890 [361]
1560 SYMBOL 176, &X0, &X0, &X0, &X111, &X10011, [2991]
&X10001011, &X11000111, &X11100001
1570 SYMBOL 177, &X1, &X11111, &X11111111, &X11 [4338]
100111, &X11111111, &X11111111, &X11111111, &X
11111111
1580 SYMBOL 178, &X10000000, &X11100000, &X10 [5230]
100000, &X11000000, &X11100000, &X11110000, &X
11110000, &X11110000
1590 SYMBOL 179, &X0, &X1111100, &X11111110, & [3480]
X10100010, &X10010010, &X10001010, &X11111110
, &X1111100
1600 SYMBOL 180, &X11110000, &X1111111, &X111 [3592]
111, &X11111, &X1111, &X0, &X1111, &X11111111
1610 SYMBOL 181, &X1111111, &X11111111, &X111 [5928]
11111, &X11111111, &X11111111, &X11111111, &X1
1111111, &X11111111
1620 SYMBOL 182, &X11110000, &X11110000, &X11 [6052]
110000, &X11110000, &X11100000, &X11000000, &X
11000000, &X10000000
1630 SYMBOL 183, &X1, &X11001101, &X10011110, [5383]
&X10010010, &X10010010, &X11110110, &X1110010
0, &X0
1640 SYMBOL 184, &X11111111, &X11111111, &X11 [3074]
110000, &X0, &X0, &X1, &X11, &X111
1650 SYMBOL 185, &X11111111, &X11111111, &X11 [5747]
1111, &X1111111, &X11111111, &X11111111, &X11
11111, &X11110011
1660 SYMBOL 186, &X10000000, &X10000000, &X11 [4391]
000000, &X11000000, &X11100000, &X11100000, &X
11110000, &X11110000
1670 SYMBOL 187, &X0, &X11100000, &X11110000, [4394]
&X10011110, &X10001110, &X10000000, &X1100000
0, &X0
1680 SYMBOL 188, &X111, &X111, &X111, &X11, &X1 [1996]
111, &X11111, &X1111, &X111
1690 SYMBOL 189, &X11100001, &X11100001, &X11 [5075]
000000, &X11000000, &X11100000, &X11100001, &X
11000000, &X0
1700 SYMBOL 190, &X11111000, &X11111000, &X11 [4983]
111100, &X111110, &X1111110, &X11111100, &X111
10000, &X1100000
1710 GOTO 1890 [361]
1720 SYMBOL 176, &X0, &X1, &X1, &X1, &X1, &X11, & [3363]
X11000011, &X110011
1730 SYMBOL 177, &X0, &X10000000, &X10000000, [5250]
&X10000000, &X10000000, &X11000000, &X1100001
1, &X11001111

```

Listing SHIFT

```

1740 SYMBOL 178,&X0,&X0,&X0,&X0,&X0,&X1110 [3133]
0000,&X11100000,&X100000
1750 SYMBOL 179,&X11,&X11,&X1,&X1,&X0,&X0, [2363]
&X111,&X1111111
1760 SYMBOL 180,&X1111,&X11,&X10001100,&X1 [6029]
0010000,&X11100011,&X11100111,&X11001011,&
X11001001
1770 SYMBOL 181,&X11111100,&X11110000,&X11 [4426]
0000,&X1000,&X11000101,&X11100101,&X110100
11,&X10010011
1780 SYMBOL 182,&X10000000,&X10000000,&X1000 [4574]
0000,&X10000000,&X0,&X0,&X11100000,&X11111
110
1790 SYMBOL 183,&X11111111,&X1111,&X0,&X0,&X [3354]
1,&X1,&X10,&X10
1800 SYMBOL 184,&X11001001,&X11001011,&X10 [3851]
100111,&X10100011,&X10000,&X1100,&X1111,&X
111111
1810 SYMBOL 185,&X10010011,&X11010011,&X11 [5059]
100111,&X11000111,&X1001,&X110001,&X110000
00,&X111110000
1820 SYMBOL 186,&X11111110,&X11100000,&X0, [5003]
&X0,&X10000000,&X10000000,&X11000000,&X110
00000
1830 SYMBOL 187,&X100,&X111,&X111 [1523]
1840 SYMBOL 188,&X11110011,&X11000011,&X11 [3363]
,&X1,&X1,&X1,&X1,&X1,&X0
1850 SYMBOL 189,&X11001100,&X11000011,&X11 [4820]
000000,&X10000000,&X10000000,&X10000000,&X
10000000,&X0
1860 SYMBOL 190,&X1100000,&X1100000,&X1110 [2481]
0000
1870 '***** [1383]
1880 'Ausgabe des Motivs im r. Fenster [3852]
1890 IF level/2=level\2 THEN INK 11,9 ELSE [1555]
INK 11,1
1900 FOR smy=1 TO 4:FOR smx=1 TO 4 [1163]
1910 DEF FNvz(b)=PEEK(HIMEM+b*8*(ASC(char$ [3202]
(smx,smy))-PEEK(&B294)))
1920 SYMBOL 254, FNvz(1), FNvz(1), FNvz(2), FN [4350]
vz(2), FNvz(3), FNvz(3), FNvz(4), FNvz(4)
1930 SYMBOL 255, FNvz(5), FNvz(5), FNvz(6), FN [4394]
vz(6), FNvz(7), FNvz(7), FNvz(8), FNvz(8)
1940 PEN #3,11:IF smx=1 AND smy=1 THEN PAP [7640]
ER #3,4:LOCATE #3,smx,smy*2-1:PRINT #3,CHR
$(32):LOCATE #3,smx,smy*2:PRINT #3,CHR$(32
);:GOTO 1960 ELSE PAPER #3,5
1950 LOCATE #3,smx,smy*2-1:PRINT #3,CHR$(2 [5459]
54):LOCATE #3,smx,smy*2:PRINT #3,CHR$(255)
;
1960 NEXT smx,smy [974]
1970 "Chaos-Motiv" im l. Fenster ausgeben [2562]
1980 p$(1,1)=char$(1,1) [594]
1990 FOR smy=1 TO 4:FOR smx=1 TO 4 [1163]
2000 IF smx=1 AND smy=1 THEN PAPER #2,4:XR [9833]
nd=smx:yrnd=smy:LOCATE #2,xrnd,yrnd*2-1:PR
INT #2,CHR$(32):LOCATE #2,xrnd,yrnd*2:PRIN
T #2,CHR$(32);:GOTO 2080 ELSE PAPER #2,5
2010 xrnd=INT(RND(1)*4)+1:yrnd=INT(RND(1)* [1851]
4)+1
2020 IF p$(xrnd,yrnd)=CHR$(32) THEN GOTO [1540]
2010
2030 p$(xrnd,yrnd)=char$(smx,smy) [2368]
2040 DEF FNvz1(b)=PEEK(HIMEM+b*8*(ASC(char [3123]
$(smx,smy))-PEEK(&B294)))
2050 SYMBOL 254, FNvz1(1), FNvz1(1), FNvz1(2) [6748]
, FNvz1(2), FNvz1(3), FNvz1(3), FNvz1(4), FNvz1
(4)
2060 SYMBOL 255, FNvz1(5), FNvz1(5), FNvz1(6) [5042]
, FNvz1(6), FNvz1(7), FNvz1(7), FNvz1(8), FNvz1
(8)
2070 PEN #2,11:LOCATE #2,xrnd,yrnd*2-1:PRI [5358]
NT #2,CHR$(254):LOCATE #2,xrnd,yrnd*2:PRIN
T #2,CHR$(255);
2080 NEXT smx,smy [974]
2090 SOUND 1,239,30,7:SOUND 2,284,30,7:SOU [3037]
ND 4,358,30,7
2100 SOUND 1,253,30,7:SOUND 2,319,30,7:SOU [3698]
ND 4,379,30,7
2110 SOUND 1,319,0,0,2:SOUND 2,379,0,0,2:S [2951]
OUND 4,478,0,0,2
2120 FOR sw=1 TO 1000:NEXT sw [921]
2130 'Hauptschleife [866]

```

Listing SHIFT

SHIFT
SHIFT
SHIFT

```

2140 ' [117]
2150 'Variable fuer Cursor [1829]
2160 xc=1:yc=1:EVERY 50,1 GOSUB 3280 [1508]
2170 IF p$(xc,yc)=CHR$(32) THEN PAPER #2,4 [4364]
ELSE PAPER #2,5
2180 LOCATE #2,xc,yc*2-1:PEN #2,9:PRINT #2 [3665]
,CHR$(22);CHR$(1);CHR$(170)
2190 LOCATE #2,xc,yc*2:PEN #2,10:PRINT #2, [3492]
CHR$(171);CHR$(22);CHR$(0);
2200 hxc=xc:hyc=yc [268]
2210 IF INKEY(69)>-1 THEN 3330 [1122]
2220 IF JOY(0)=0 THEN 2170 [843]
2230 IF JOY(0)=1 AND yc>1 THEN yc=yc-1:GOT [1482]
O 2290
2240 IF JOY(0)=2 AND yc<4 THEN yc=yc+1:GOT [1645]
O 2290
2250 IF JOY(0)=4 AND xc>1 THEN xc=xc-1:GOT [3249]
O 2290
2260 IF JOY(0)=8 AND xc<4 THEN xc=xc+1:GOT [3166]
O 2290
2270 GOSUB 2650 [817]
2280 IF JOY(0)=16 THEN 2360 [1643]
2290 IF p$(hxc,hyc)=char$(1,1) THEN LOCATE [5914]
#2,hxc,hyc*2-1:PRINT #2,CHR$(32):LOCATE #
2,hxc,hyc*2:PRINT #2,CHR$(32);:GOTO 2170
2300 DEF FNvz2(b)=PEEK(HIMEM+b*8*(ASC(p$(h [2417]
xc,hyc))-PEEK(&B294)))
2310 SYMBOL 254, FNvz2(1), FNvz2(1), FNvz2(2) [6176]
, FNvz2(2), FNvz2(3), FNvz2(3), FNvz2(4), FNvz2
(4)
2320 SYMBOL 255, FNvz2(5), FNvz2(5), FNvz2(6) [6162]
, FNvz2(6), FNvz2(7), FNvz2(7), FNvz2(8), FNvz2
(8)
2330 PEN #2,11:LOCATE #2,hxc,hyc*2-1:PRINT [5089]
#2,CHR$(254):LOCATE #2,hxc,hyc*2:PRINT #2
,CHR$(255);
2340 GOTO 2170 [361]
2350 'Zeichen verschieben [1585]
2360 IF p$(xc,yc-1)=CHR$(32) THEN ry=-1:rx [2769]
=0:GOTO 2410
2370 IF p$(xc,yc+1)=CHR$(32) THEN ry=1:rx= [3461]
0:GOTO 2410
2380 IF p$(xc-1,yc)=CHR$(32) THEN ry=0:rx= [1721]
-1:GOTO 2410
2390 IF p$(xc+1,yc)=CHR$(32) THEN ry=0:rx= [1709]
1:GOTO 2410
2400 FOR sfs=1 TO 10:SOUND 1,100,2,7:SOUND [4175]

```

Listing SHIFT

```

2,200,2,7:SOUND 3,0,1,0:NEXT sfs:GOTO 217
0
2410 SOUND 1,120,20,7,0,1:SOUND 2,240,20,7 [6399]
,0,1:LOCATE #2,xc,yc*2-1:PAPER #2,4:PRINT
#2,CHR$(32):LOCATE #2,xc,yc*2:PRINT #2,CHR
$(32);
2420 DEF FNvz3(b)=PEEK(HIMEM+b+8*(ASC(p$(x [2940]
c,yc))-PEEK(&B294)))
2430 SYMBOL 254, FNvz3(1), FNvz3(1), FNvz3(2) [5234]
, FNvz3(2), FNvz3(3), FNvz3(3), FNvz3(4), FNvz3
(4)
2440 SYMBOL 255, FNvz3(5), FNvz3(5), FNvz3(6) [5268]
, FNvz3(6), FNvz3(7), FNvz3(7), FNvz3(8), FNvz3
(8)
2450 PEN #2,11:PAPER #2,5:LOCATE #2,xc+rx, [6651]
(yc+ry)*2-1:PRINT #2,CHR$(254):LOCATE #2,x
c+rx,(yc+ry)*2:PRINT #2,CHR$(255);
2460 p$(xc+rx,yc+ry)=p$(xc,yc):p$(xc,yc)=c [5208]
har$(1,1):xc=xc+rx:yc=yc+ry
2470 GOSUB 2830 [909]
2480 GOTO 2170 [361]
2490 'Sub : Zeichne Randsymbole [1426]
2500 PEN 2:PRINT CHR$(22);CHR$(1);CHR$(172 [6965]
);CHR$(8);:PEN 1:PRINT CHR$(173);CHR$(22);
CHR$(0)
2510 RETURN [555]
2520 'Musik [1087]
2530 READ note:IF note=-1 THEN RESTORE 259 [2075]
0:GOTO 2530
2540 SOUND 1,note*2,0,0,1 [1908]
2550 SOUND 2,note/2,0,0,2 [1737]
2560 SOUND 4,note/6,0,0,2 [1452]
2570 RETURN [555]
2580 'Datas fuer Titelmusik [1846]
2590 DATA 239,284,358,239,213,253,319,213 [1930]
2600 DATA 253,284,253,239,213 [1212]
2610 DATA 239,239,253,253 [759]
2620 DATA 159,179,213,179,213,159,127 [1542]
2630 DATA 127,127,-1 [524]
2640 'Zeichen wechseln [1691]
2650 IF JOY(0)=17 THEN wx=0:wy=-1:GOTO 270 [2500]
0
2660 IF JOY(0)=24 THEN wx=1:wy=0:GOTO 2700 [2823]
2670 IF JOY(0)=18 THEN wx=0:wy=1:GOTO 2700 [1455]
2680 IF JOY(0)=20 THEN wx=-1:wy=0:GOTO 270 [1433]
0
2690 RETURN [555]
2700 IF p$(xc+wx,yc+wy)<=CHR$(32) OR p$(xc [10868]
,yc)=CHR$(32) OR changer=0 THEN FOR sfs=1
TO 10:SOUND 1,50,2,7:SOUND 2,500,2,7:SOUND
3,0,1,0:NEXT sfs:RETURN
2710 SOUND 3,0,0,0,3,0,5:SOUND 3,0,0,0,3,0 [8161]
,20:DEF FNvz4(b)=PEEK(HIMEM+b+8*(ASC(p$(xc
+wx,yc+wy))-PEEK(&B294)))
2720 DEF FNvz5(b)=PEEK(HIMEM+b+8*(ASC(p$(x [3238]
c,yc))-PEEK(&B294)))
2730 SYMBOL 254, FNvz4(1), FNvz4(1), FNvz4(2) [10703]
, FNvz4(2), FNvz4(3), FNvz4(3), FNvz4(4), FNvz4
(4):SYMBOL 255, FNvz4(5), FNvz4(5), FNvz4(6),
FNvz4(6), FNvz4(7), FNvz4(7), FNvz4(8), FNvz4(
8)
2740 PEN #2,11:PAPER #2,5:LOCATE #2,xc,yc* [5761]
2-1:PRINT #2,CHR$(254):LOCATE #2,xc,yc*2:P
RINT #2,CHR$(255);
2750 SYMBOL 254, FNvz5(1), FNvz5(1), FNvz5(2) [11705]
, FNvz5(2), FNvz5(3), FNvz5(3), FNvz5(4), FNvz5
(4):SYMBOL 255, FNvz5(5), FNvz5(5), FNvz5(6),
FNvz5(6), FNvz5(7), FNvz5(7), FNvz5(8), FNvz5(
8)
2760 LOCATE #2,xc+wx,(yc+wy)*2-1:PRINT #2, [4807]
CHR$(254):LOCATE #2,xc+wx,(yc+wy)*2:PRINT
#2,CHR$(255);
2770 hilf$(xc,yc):p$(xc,yc)=p$(xc+wx,yc [2987]
+wy):p$(xc+wx,yc+wy)=hilf$
2780 xc=xc+wx:yc=yc+wy [952]
2790 changer=changer-1:DI:LOCATE 10,14-cha [2640]
nger:PEN 8:PAPER 0:PRINT CHR$(247);CHR$(24
6):EI
2800 GOSUB 2830 [909]
2810 RETURN [555]
2820 'Test, ob Motiv fertig [114]
2830 FOR sty=1 TO 4:FOR stx=1 TO 4 [1219]

```

Listing SHIFT

```

2840 IF p$(stx,sty)<>char$(stx,sty) THEN R [3044]
ETURN
2850 NEXT stx,sty [402]
2860 'Motiv fertig [1250]
2870 stevl=REMAIN(1) [837]
2880 FOR sfe=1 TO 100:FOR sfc=&BD23 TO &BD [4667]
25:CALL sfc:NEXT sfc,sfe:' Super-Farb-Effe
kt !!!
2890 RESTORE 2590 [872]
2900 READ note:IF note=-1 THEN 2930 [1567]
2910 SOUND 1,note,0,0,3:SOUND 2,note/2,0,0 [4879]
,3:SOUND 4,note/4,0,0,3
2920 GOTO 2900 [391]
2930 LOCATE 2,18:PEN 1:PAPER 7:PRINT"SCORE [3009]
:";level*200
2940 score=score+level*200 [1799]
2950 FOR sws=1 TO 2000:NEXT sws [1883]
2960 LOCATE 2,18:PAPER 0:PRINT SPACE$(18) [1776]
2970 FOR sws=1 TO 500:NEXT sws [2229]
2980 bonus=zeit+(changer*100) [1210]
2990 LOCATE 2,18:PEN 1:PAPER 7:PRINT"BONUS [2805]
:";bonus
3000 FOR sws=1 TO 2000:NEXT sws [1883]
3010 score=score+bonus [1572]
3020 PEN 8:PAPER 7:LOCATE 8,3:PRINT USING" [2605]
####";score
3030 FOR sws=1 TO 50:SOUND 1,200,2,7:SOUND [6462]
2,400,2,7:SOUND 4,600,2,7:SOUND 7,0,2,0:N
EXT sws
3040 FOR sws=1 TO 1000:NEXT sws [2216]
3050 LOCATE 2,18:PAPER 0:PRINT SPACE$(18): [2613]
level=level+1
3060 IF level>6 THEN 3120 [1193]
3070 changer=6 [225]
3080 IF wahl=1 THEN zeit=260 ELSE zeit=410 [2363]
3090 zeit=zeit-level*10 [2688]
3100 FOR spy=1 TO 5:FOR spx=1 TO 5:p$(spx, [4638]
spy)=CHR$(0):NEXT spx,spy:PAPER #2,4:PAPER
#3,4:CLS#2:CLS#3:GOTO 1040
3110 'Spiel gewonnen (alle 6 Level gemeist [3370]
ert)
3120 CLS:PEN 1:LOCATE 2,5:PRINT"The" [1106]
3130 LOCATE 2,6:PRINT"ABSOLUTE INSANITY" [2773]
3140 LOCATE 2,7:PRINT"has come !!!" [2000]
3150 LOCATE 2,9:PRINT"You've mastered" [2265]
3160 LOCATE 2,10:PRINT"ALL levels" [2405]
3170 LOCATE 2,11:PRINT"and made" [3044]
3180 LOCATE 1,12:PEN 7:PRINT score;:PEN 1: [3372]
PRINT"points !!!"
3190 FOR ssg1=1 TO 3:FOR ssg=7 TO 1 STEP-1 [1612]
3200 SOUND 1,300,35,ssg:SOUND 2,400,35,ssg [2820]
:SOUND 4,500,35,ssg
3210 NEXT ssg,ssg1 [1055]
3220 OUT &BC00,8:OUT &BD00,1:FOR ssg=7 TO [3265]
1 STEP-1
3230 SOUND 1,300,60,ssg:SOUND 2,400,60,ssg [2882]
:SOUND 4,500,60,ssg
3240 NEXT ssg [344]
3250 FOR ssw=1 TO 4000:NEXT ssw:OUT &BC00, [1541]
8:OUT &BD00,0
3260 GOTO 3420 [509]
3270 'Interrupt fuer Uhr [721]
3280 SOUND 4,50,2,7:zeit=zeit-1 [2742]
3290 PEN 8:PAPER 7:LOCATE 16,3:PRINT USING [4269]
"###";zeit
3300 IF zeit=0 THEN 3330 [1488]
3310 RETURN [555]
3320 'Zeit vorbei , Spiel verloren od. Spi [5404]
el abgebrochen/aufgegeben
3330 stevl=REMAIN(1) [837]
3340 ssvc=640:FOR ssv=0 TO 320 STEP 4:SOUn [6871]
D 1,ssv+20,1,3:SOUND 2,(ssv+10)*2,1,3:SOUn
D 4,(ssv+50)*2,1,3:MOVE ssv,0:DRAW ssv,220
,1
3350 MOVE ssvc,0:DRAW ssvc,220:ssvc=ssvc-4 [2173]
:NEXT ssv
3360 FOR ssv=0 TO 218 STEP 2:MOVE 0,ssv:DR [3982]
AW 640,ssv,0:NEXT ssv
3370 PAPER 0:PEN 4:LOCATE 7,12:PRINT"GAME [3399]
OVER"
3380 FOR ssv=7 TO 1 STEP-1 [874]
3390 SOUND 1,300,35,ssv:SOUND 2,400,35,ssv [3109]

```

Listing SHIFT


```

:SOUND 4,500,35,ssv
3400 NEXT ssv:FOR ssw=1 TO 3000:NEXT ssw [2310]
3410 'High-Score ? [901]
3420 IF score<hsc(8,wahl) THEN 3760 [1909]
3430 MODE 1:BORDER 17:INK 0,4 [1780]
3440 INK 1,26:INK 2,25:INK 3,6 [1452]
3450 LOCATE 5,5:PEN 2:PAPER 0:PRINT"Congra [4125]
tulations !!!"
3460 LOCATE 5,7:PEN 1:PRINT"You've got a" [3743]
3470 LOCATE 5,9:PEN 2:PRINT"HIGH-SCORE !!! [2271]
"
3480 LOCATE 5,11:PEN 1:PRINT"Please enter [3450]
your name !"
3490 LOCATE 5,14:PAPER 2:PEN 3 [1597]
3500 FOR salph=65 TO 90:PRINT CHR$(salph); [4187]
:NEXT salph:PRINT CHR$(32)
3510 PAPER 0:LOCATE 5,17:PRINT STRING$(5," [3033]
")
3520 name$="":inx=1 [746]
3530 PEN 1:LOCATE inx+4,15:PRINT "~" [2108]
3540 PEN 2:LOCATE LEN(name$)+5,17:IF inx=2 [4540]
7 THEN PRINT CHR$(32) ELSE PRINT CHR$(inx+
64)
3550 IF JOY(0)=4 AND inx>1 THEN LOCATE inx [6943]
+4,15:PRINT CHR$(32):inx=inx-1:GOTO 3530
3560 IF JOY(0)=8 AND inx<27 THEN LOCATE in [6097]
x+4,15:PRINT CHR$(32):inx=inx+1:GOTO 3530
3570 IF JOY(0)<>16 THEN GOTO 3530 [1402]
3580 FOR slf=1 TO 8:SOUND 7,200,2,7,0,0,10 [2913]
:SOUND 7,0,2,0:NEXT slf
3590 IF inx=27 THEN name$=name$+CHR$(32) E [3617]
LSE name$=name$+CHR$(inx+64)
3600 IF LEN(name$)<5 THEN FOR swg=1 TO 100 [3939]
:NEXT swg:GOTO 3530
3610 PEN 1:LOCATE 5,20:PRINT"Input correct [2967]
? (Enter J/N)"
3620 ic$=INKEY$:IF ic$="" THEN 3620 [1882]
3630 IF UPPER$(ic$)="N" THEN 3430 [1367]
3640 IF UPPER$(ic$)<>"J" THEN 3620 [1395]
3650 FOR shs=8 TO 1 STEP-1 [1048]
3660 IF score<hsc(shs,wahl) THEN 3680 [1613]
3670 NEXT shs [470]
3680 FOR shss=shs+1 TO 8 [1518]
3690 hhsc(shss,wahl)=hsc(shss,wahl):hname [3442]
$(shss,wahl)=hname$(shss,wahl)
3700 NEXT shss [350]
3710 FOR shss=shs+2 TO 8 [1558]
3720 hsc(shss,wahl)=hhsc(shss-1,wahl):hnam [4674]
e$(shss,wahl)=hname$(shss-1,wahl)
3730 NEXT shss [350]
3740 hsc(shs+1,wahl)=score:hname$(shs+1,wa [3185]
hl)=name$
3750 'High-Score ausgeben [1127]
3760 MODE 1:BORDER 19:FOR sco=0 TO 3:INK s [4162]
co,19:NEXT sco
3770 PAPER 3:PRINT SPACES(160) [1669]
3780 MOVE 0,399:DRAW 639,399,1:DRAW 639,33 [3625]
6:DRAW 0,336:DRAW 0,399
3790 th$="HALL OF FAME":FOR sth=1 TO LEN(t [4811]
h$):lth=ASC(MID$(th$,sth,1)):IF lth=32 THE
N 3850
3800 DEF FNth(b)=PEEK(HIMEM+b+8*(lth-PEEK( [2180]
&B294)))
3810 SYMBOL 254, FNth(1), FNth(1), FNth(2), FN [3979]
th(2), FNth(3), FNth(3), FNth(4), FNth(4)
3820 SYMBOL 255, FNth(5), FNth(5), FNth(6), FN [4767]
th(6), FNth(7), FNth(7), FNth(8), FNth(8)
3830 PEN 1:LOCATE sth+14,2:PRINT CHR$(254) [2384]
3840 PEN 2:LOCATE sth+14,3:PRINT CHR$(255) [2155]
3850 NEXT sth [357]
3860 MOVE 0,336:DRAW 0,0:DRAW 639,0:DRAW 6 [1926]
39,336
3870 INK 1,2:INK 2,1:INK 3,6 [750]
3880 PAPER 0:PEN 1:LOCATE 13,6:PRINT "- DI [2967]
FFICULTY 1 -"
3890 FOR shsc=8 TO 1 STEP-1 [1702]
3900 PEN 2:PRINT STRING$(10,9);shsc;CHR$(1 [5635]
49);" ";hname$(shsc,1);TAB(25);CHR$(149);h
sc(shsc,1)
3910 NEXT shsc [366]
3920 PEN 1:LOCATE 13,16:PRINT "- DIFFICULT [1812]
Y 2 -"
3930 FOR shsc=8 TO 1 STEP-1 [1702]

```

Listing SHIFT

```

3940 PEN 2:PRINT STRING$(10,9);shsc;CHR$(1 [5076]
49);" ";hname$(shsc,2);TAB(25);CHR$(149);h
sc(shsc,2) [366]
3950 NEXT shsc [1533]
3960 PEN 3:LOCATE 3,13:PRINT"Your" [1743]
3970 LOCATE 3,14:PRINT"Score" [1626]
3980 LOCATE 3,15:PRINT"was :" [2221]
3990 PEN 2:LOCATE 2,16:PRINT score [4436]
4000 LOCATE 3,17:PRINT"(D.";USING"#";wahl;
:PRINT")"
4010 WINDOW 39,39,6,24:PAPER 3:CLS [1520]
4020 IF INKEY(47)>-1 THEN 4040 [391]
4030 PEN 2:PRINT"Press SPACE ";:FOR swl=1 [4283]
TO 500:NEXT swl:GOTO 4020
4040 bwl=20:bwr=21:bwo=13:bwu=14 [2417]
4050 WINDOW bwl,bwr,bwo,bwu:PAPER 1:CLS [2889]
4060 bwl=bwl-1:bwr=bwr+1:bwo=bwo-1:bwu=bwu [1972]
+1
4070 IF bwo<1 THEN bwo=1:bwu=26 [1147]
4080 IF bwl<1 THEN 4100 [259]
4090 GOTO 4050 [503]
4100 BORDER 2 [1014]
4110 FOR swb=1 TO 1000:NEXT swb [2350]
4120 FOR sco=0 TO 15:INK sco,0:NEXT sco [1899]
4130 BORDER 0:MODE 0:PAPER 0:CALL &BB4E [2769]
4140 WINDOW #1,2,19,2,5:PAPER #1,3:CLS#1 [1434]
4150 WINDOW 1,20,7,25:GOTO 450 [1397]
4160 'Symbols [339]
4170 SYMBOL AFTER 48 [1272]
4180 SYMBOL 48,&X11111111,&X11000111,&X11000 [4891]
11,&X11000111,&X11000111,&X11000111,&X11111111
4190 SYMBOL 49,&X111000,&X11000,&X11000,&X [3059]
11000,&X111110,&X111110,&X111110
4200 SYMBOL 50,&X11111111,&X111,&X111,&X11111 [4206]
11,&X11000000,&X11000000,&X11111111
4210 SYMBOL 51,&X11111110,&X110,&X110,&X111 [3755]
111,&X111,&X111,&X11111111
4220 SYMBOL 52,&X11100000,&X11100000,&X111100 [4751]
00,&X1110111,&X1110111,&X11111111,&X111
4230 SYMBOL 53,&X11111111,&X11100000,&X111000 [3809]
00,&X11111111,&X1111,&X1111,&X1111111
4240 SYMBOL 54,&X11111100,&X11011000,&X11000 [4995]
00,&X11111111,&X11000111,&X11000111,&X1111111
4250 SYMBOL 55,&X11111111,&X111,&X111,&X11111 [4256]
,&X11000,&X11000,&X11000
4260 SYMBOL 56,&X1111110,&X110110,&X110110, [3837]
&X11111111,&X1110111,&X1110111,&X11111111
4270 SYMBOL 57,&X11111111,&X11000111,&X11000 [3518]
11,&X11111111,&X111,&X111,&X111
4280 SYMBOL 170,&X11111111,&X10100101,&X11 [5265]
000011,&X10000001,&X10000001,&X10000001,&X
10000001,&X10000001
4290 SYMBOL 171,&X10000001,&X10000001,&X10 [4174]
000001,&X10000001,&X10000001,&X11000011,&X
10100101,&X11111111
4300 SYMBOL 172,0,0,0,&X111111,&X10001,&X10 [3584]
111,&X101111,&X11111
4310 SYMBOL 173,&X111111000,&X10111100,&X11 [5272]
011110,&X11100000,&X11100000,&X1100000,&X1
00000
4320 SYMBOL 174,&18,&14,&F2,&81,&81,&F2,&1 [1857]
4,&18
4330 RETURN [555]

```



```

10 ' ***** [768]
20 ' * SHIFTDAT.BAS * [607]
30 ' ***** [768]
40 ' [117]
50 ' Programm zur Erzeugung [1785]
60 ' eines ASCII-Files [1517]
70 ' "SHIFT.DAT" [185]
80 ' [117]
90 OPENOUT "SHIFT.DAT" [1871]
100 FOR hw=1 TO 2:FOR hs=1 TO 8 [1532]
110 PRINT #9,"CPC":PRINT #9,100 [1507]
120 NEXT hs, hw:CLOSEOUT [655]
130 NEW [318]

```

Listing SHIFT



Bild 1: Level 1 und...

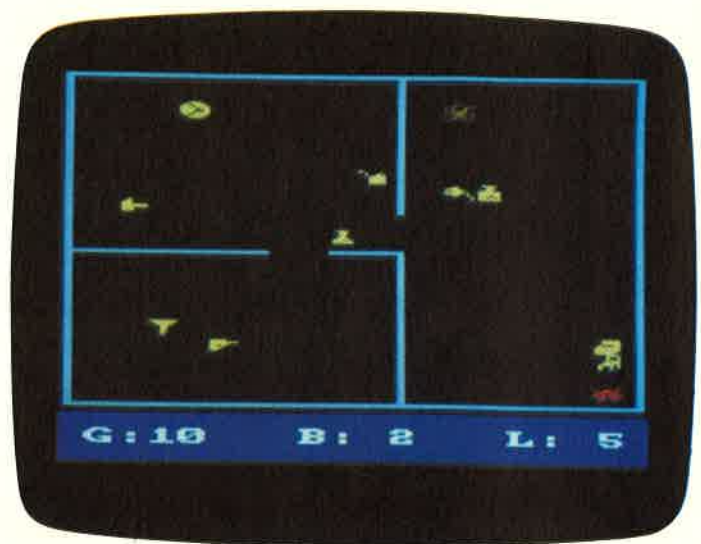


Bild 2: Level 2 des Spiels

Rob the Roboter Ein Spiel für Joystick- Akrobaten.

Wenn Sie einem kleinen Roboter beim Entkommen aus einem Labor behilflich sein wollen, brauchen Sie nur zwei Sachen: Die Eine ist ein Joystick, und die Andere Geschicklichkeit.

Die Beschreibung des Spiels ist recht einfach: Rob, der Roboter will einem Labor enttrinnen. Dazu muß er in insgesamt 10 Räumen alle Gegenstände, die dort herumliegen, einsammeln. Allerdings muß er höllisch darauf aufpassen, daß er die Alarmanlage, die in den Wänden verborgen ist, nicht berührt, da ihm dieses eins seiner Roboter-Existenzen kosten würde. Zwischenzeitlich ist ihm außerdem einer der (Robot-) Wächter auf die

Schliche gekommen und sucht nach ihm. Auch ein Zusammenstoß mit jenem Wächter vermindert die 'Leben' unseres kleinen Freundes. Hat Rob alle Gegenstände aufgesammelt, so ist er dem Labor entronnen und gleichzeitig das Spiel beendet. Gesteuert wird Rob mit dem Joystick oder den Cursortasten. Hier nun die verwendeten Variablen des Spiels: MUSIK Zählervariable für Sound. BILD springt das Unterprogramm für den zu malenden Raum an. LEBEN gibt die Anzahl der Leben an; kann bis auf 99 erhöht werden. ROBX, ROBY x- und y-Position des Roboters. WAECHTER Zählervariable für den Einsprung des Wächters. BEWEG gibt die Bewegung des Roboters wieder. XA, YA, ZA XA und YA ergeben die Position für die Rahmen der Bilder; ZA für das Symbol. TGES Testvariable.

Seien Sie also so nett und helfen Rob the Roboter aus seiner Gefangenschaft. Noch ein Hinweis für CPC 464/664 Benutzer: Sie müssen vor dem Starten des Programms den Emulator aus Sonderheft 4/87 laden.

(T. Kiene / JB)

für 464-664-6128



```

10 ' [117]
20 '*** Rob the Roboter *** [2063]
30 ' [117]
40 '*** (c) Sept. 1987 *** [456]
50 ' [117]
60 '*** Tina Kiene und *** [665]
70 ' [117]
80 '*** Gerd Wartala *** [1934]
90 ' [117]
100 ' [117]
110 ' [117]
120 ' [117]
130 ' [117]
140 ' [117]
150 '*** Initialisierung *** [1272]
160 ' [117]
170 MODE 0 [507]
180 BORDER 0: INK 0,0 [1028]
190 SYMBOL AFTER 32 [1296]
200 SYMBOL 147,24,24,24,31,31,0,0,0 [1524]
210 SYMBOL 165,129,90,60,255,165,255,36,10 [2413]
2 '
220 SYMBOL 166,24,231,36,188,255,25,60,231 [1934]

```

Listing Rob

```

230 SYMBOL 167,16,48,112,127,127,112,112 [1851]
240 SYMBOL 168,0,224,160,160,191,165,165,2 [1964]
24 '
250 SYMBOL 150,0,0,0,31,31,24,24,24 [1873]
260 SYMBOL 153,24,24,24,248,248,0,0,0 [1982]
270 SYMBOL 156,0,0,0,248,248,24,24,24 [2192]
280 SYMBOL 226,0,48,248,60,250,50,1,1 [1965]
290 SYMBOL 227,60,24,24,126,126,118,106,12 [2186]
29 '
300 SYMBOL 228,255,126,60,24,24,24,24,16 [2160]
310 SYMBOL 229,30,18,18,30,62,122,74,72 [1985]
320 SYMBOL 230,60,24,24,24,60,126,126,0 [2442]
330 SYMBOL 231,0,192,34,37,31,31,31,31 [2259]
340 SYMBOL 232,62,62,62,70,254,254,252,248 [2370]
350 SYMBOL 233,60,94,239,247,243,239,94,60 [2262]
360 SYMBOL 234,0,0,248,255,248,208,224,192 [3043]
370 ENT 1,10,-20,1,10,100,1: ENT 4,4,-30,1: [7747]
ENV 4,2,-125,1,2,-50,1: ENV 2,2,-128,1: ENV
5,10,40,5,10,80,5: ENT 5,10,30,5,10,100,5: E
NV 6,10,40,5,10,80,5: ENT 6,10,-13,5,10,-15
,5: ENT 7,5,1,2,5,-1,2
380 ' [117]
390 '*** Titel + Auswahl *** [1310]
400 ' [117]
410 CLS: INK 1,6: INK 2,6,15: INK 3,8 [2812]
420 PEN 1 [549]

```

Listing Rob

```

430 LOCATE 5,3:PRINT CHR$(165); [1275]
440 PEN 2 [548]
450 PRINT" R O B "; [1010]
460 PEN 1 [549]
470 PRINT CHR$(165) [1004]
480 PEN 2 [548]
490 LOCATE 9,5:PRINT"THE" [1406]
500 LOCATE 4,7:PRINT"R O B O T E R" [1731]
510 PEN 3 [547]
520 LOCATE 1,11:PRINT STRING$(20,207); [2651]
530 LOCATE 4,15:PRINT"1--> ANLEITUNG" [2110]
540 LOCATE 4,17:PRINT"2--> SPIEL" [1342]
550 LOCATE 4,19:PRINT"3--> ENDE" [1913]
560 LOCATE 4,23:PRINT"IHRE WAHL ?" [2198]
570 IF SQ(1)<64 THEN GOSUB 620 ELSE IF INK [3311]
BY(64)=0 OR INKEY(13)=0 THEN 740
580 IF INKEY(65)=0 OR INKEY(14)=0 THEN 970 [1100]
590 IF INKEY(57)=0 OR INKEY(5)=0 THEN CALL [1386]
0
600 GOTO 570 [379]
610 ' [117]
620 '*** Titelmelodie *** [1312]
630 ' [117]
640 ON musik GOSUB 670,680,690,700,710,720 [2510]
650 IF musik=7 THEN musik=0 [2546]
660 musik=musik+1:RETURN [1765]
670 FOR i=1 TO 8:SOUND 1,INT(RND*215+40),2 [3732]
0,8,,7:SOUND 2,239,20,4:NEXT:RETURN
680 SOUND 1,239,20,8:SOUND 2,169,20,8:RETU [2253]
RN
690 SOUND 1,239,20,8:SOUND 2,169,20,8:RETU [2253]
RN
700 SOUND 1,142,80,8:SOUND 2,338,80,8:RETU [3170]
RN
710 SOUND 1,478,40,8:SOUND 2,119,40,8:RETU [2906]
RN
720 SOUND 1,213,60,8:SOUND 2,478,60,8:RETU [2774]
RN
730 ' [117]
740 '*** Anleitung *** [939]
750 ' [117]
760 MODE 1 [506]
770 PRINT"ANLEITUNG" [1047]
780 PRINT:PRINT:PEN 1:PRINT CHR$(165);:PEN [5165]
3:PRINT" will dem Labor enttrinnen."
790 PRINT"Dazu muss er in 10 Raeumen alle [4240]
Gegen-"
800 PRINT"staende, die herumliegen, einsam [4205]
mein."
810 PRINT"Allerdings darf er die Alarmanla [5040]
ge, die"
820 PRINT"sich in den Waenden befindet, ni [5057]
cht be-"
830 PRINT"ruehren, da ihm das 1 Leben kost [3506]
et."
840 PRINT"Zwischenzeitlich ist ihm der Wae [3119]
chter"
850 PRINT"auf die Schliche gekommen und su [3363]
cht"
860 PRINT"ihn. Auch ein Zusammenstoss mit [3449]
diesem"
870 PRINT"kostet 1 Leben. Hat ";:PEN 1:PRI [6391]
NT CHR$(165);:PEN 3:PRINT" alle Gegen-"
880 PRINT"staende + einen Schluessel aufge [3453]
sammelt"
890 PRINT"ist er dem Labor entronnen." [2559]
900 PRINT:PRINT"Gesteuert wird ";:PEN 1:PR [6588]
INT CHR$(165);:PEN 3:PRINT" ueber Joystick
oder"
910 PRINT"Cursortasten." [2332]
920 LOCATE 15,24:PRINT"< SPACE >" [980]
930 IF INKEY(47)<>0 THEN 930 [651]
940 MODE 0 [507]
950 GOTO 390 [494]
960 ' [117]
970 '*** Bildschirmaufteilung *** [2286]
980 ' [117]
990 bild=1:leben=5 [753]
1000 WINDOW#1,1,20,1,22:PEN#1,0:CLS#1 [2161]
1010 WINDOW#2,1,20,23,25:INK 5,0:PAPER#2,5 [2706]
:CLS#2

```

Listing Rob

```

1020 INK 7,0:PEN#2,7:FOR i=1 TO 3:LOCATE#2 [6845]
,1,1:PRINT#2,CHR$(211);:LOCATE#2,20,1:PRIN
T#2,CHR$(209);:NEXT
1030 ' [117]
1040 '** Spielfeld *** [1033]
1050 ' [117]
1060 PEN#1,0:CLS#1 [945]
1070 INK 0,0:INK 1,0:INK 2,0:INK 3,0:INK 4 [2507]
,0:robx=19:roby=21
1080 gst=10:GOSUB 2380 [817]
1090 IF bild>1 THEN ON bild GOSUB 2380,255 [3132]
0,2680,2900,3020,3200,3500,3690,4060,4210,
4710
1100 PEN#1,1 [465]
1110 LOCATE#1,robx,roby [647]
1120 PRINT#1,CHR$(165); [1004]
1130 PEN#1,2 [472]
1140 RESTORE 1260 [699]
1150 FOR i=1 TO 10 [315]
1160 x=INT(RND*20) [1413]
1170 IF x<2 OR x>19 THEN 1160 [1706]
1180 y=INT(RND*20) [1273]
1190 IF y<2 OR y>21 THEN 1180 [1284]
1200 tges=TEST(x*32-18,400-(y*16-6)) [1845]
1210 IF tges<>0 THEN 1160 [892]
1220 READ za [558]
1230 LOCATE#1,x,y [392]
1240 PRINT#1,CHR$(za); [717]
1250 NEXT [350]
1260 DATA 167,226,227,228,229,230,231,232, [1875]
233,234
1270 PEN#2,6 [520]
1280 LOCATE#2,2,2 [595]
1290 PRINT#2,"G:";:PRINT#2,USING"###";gst; [1968]
1300 LOCATE#2,9,2 [622]
1310 PRINT#2,"B:";:PRINT#2,USING"###";bild; [1007]
1320 LOCATE#2,16,2 [775]
1330 PRINT#2,"L:";:PRINT#2,USING"###";leben [1952]
;
1340 INK 1,6:INK 2,18:INK 3,11:INK 4,15,8; [4304]
INK 5,1:INK 6,20:INK 7,0
1350 GOTO 1540 [391]
1360 waechter=waechter+1:IF waechter<>20 T [4532]
HEN RETURN:'Einsprung Waechter
1370 SOUND 1,127,20,,1 [1553]
1380 x=INT(RND*20) [1413]
1390 IF x<2 OR x>19 THEN 1380 [1487]
1400 y=INT(RND*20) [1273]
1410 IF y<2 OR y>21 THEN 1400 [1288]
1420 tges=TEST(x*32-18,400-(y*16-6)) [1845]
1430 IF tges<>0 THEN 1380 [816]
1440 PEN#1,4 [478]
1450 LOCATE#1,x,y [392]
1460 PRINT#1,CHR$(166); [856]
1470 RETURN [555]
1480 PEN#1,0:SOUND 2,478,20,,1 [2292]
1490 LOCATE#1,x,y [392]
1500 PRINT#1," "; [582]
1510 waechter=0 [1151]
1520 RETURN [555]
1530 ' [117]
1540 '*** Hauptprogramm *** [1683]
1550 ' [117]
1560 waechter=0 [1151]
1570 beweg=0 [231]
1580 IF INKEY(0)=0 OR JOY(0)=1 THEN beweg= [2028]
1
1590 IF INKEY(2)=0 OR JOY(0)=2 THEN beweg= [1070]
2
1600 IF INKEY(1)=0 OR JOY(0)=8 THEN beweg= [2199]
3
1610 IF INKEY(8)=0 OR JOY(0)=4 THEN beweg= [1611]
4
1620 GOSUB 1360:IF waechter>50 THEN GOSUB [2748]
1480
1630 SOUND 4,213,2,2,4,4 [1407]
1640 ' [117]
1650 '*** nach oben *** [1033]
1660 ' [117]
1670 IF beweg<>1 THEN GOTO 1860 [1042]
1680 roby=roby-1 [219]

```

Listing Rob

```

1690 IF roby<1 THEN roby=1 [11551]
1700 tges=TEST(robx*32-18,400-(roby*16-6)) [16501]
1710 PEN#1,0 [4661]
1720 LOCATE#1,robx,roby+1 [6681]
1730 PRINT#1," "; [5821]
1740 PEN#1,1 [4651]
1750 LOCATE #1,robx,roby [6471]
1760 PRINT#1,CHR$(165); [10041]
1770 IF tges=3 OR tges=4 THEN SOUND 1,506, [39481]
100,,5,5:leben=leben-1 ELSE GOTO 1800
1780 IF leben<1 THEN 4610 ELSE PEN#2,6:LOC [51051]
ATE#2,18,2:PRINT#2,USING"###";leben;
1790 GOTO 1040 [3631]
1800 IF tges=2 THEN SOUND 2,75,2,,2:gst=gs [44231]
t-1 ELSE GOTO 1840
1810 IF gst=0 THEN SOUND 1,358,110,,6,6:bi [57491]
ld=bild+1: PEN#2,6:LOCATE#2,4,2:PRINT#2,USI
NG"###";gst;:LOCATE#2,11,2:PRINT#2,USING"##
";bild;:GOTO 1040
1820 PEN#2,6 [5201]
1830 LOCATE#2,4,2:PRINT#2,USING"###";gst; [20751]
1840 GOTO 1580 [3191]
1850 ' [1171]
1860 '*** nach unten *** [9121]
1870 ' [1171]
1880 IF beweg<>2 THEN 2040 [15621]
1890 roby=roby+1 [2231]
1900 IF roby>22 THEN roby=22 [15311]
1910 tges=TEST(robx*32-18,400-(roby*16-6)) [16501]
1920 PEN#1,0 [4661]
1930 LOCATE#1,robx,roby-1 [6641]
1940 PRINT#1," "; [5821]
1950 PEN#1,1 [4651]
1960 LOCATE#1,robx,roby [6471]
1970 PRINT#1,CHR$(165); [10041]
1980 IF tges=3 OR tges=4 THEN SOUND 1,506, [75091]
100,,5,5:leben=leben-1:IF leben<1 THEN 461
0 ELSE PEN#2,6:LOCATE#2,18,2:PRINT#2,USING
"###";leben;:GOTO 1040
1990 IF tges=2 THEN SOUND 2,75,2,,2:gst=gs [45251]
t-1 ELSE 2020
2000 IF gst=0 THEN SOUND 1,358,110,,6,6:bi [77941]
ld=bild+1: PEN#2,6:LOCATE#2,11,2:PRINT#2,US
ING"###";bild;:LOCATE#2,4,2:PRINT#2,USING"#
";gst;:GOTO 1040
2010 PEN#2,6:LOCATE#2,4,2:PRINT#2,USING"## [19631]
";gst;
2020 GOTO 1580 [3191]
2030 ' [1171]
2040 '*** nach rechts *** [12441]
2050 ' [1171]
2060 IF beweg<>3 THEN 2200 [7481]
2070 robx=robx+1 [10481]
2080 IF robx>20 THEN robx=20 [11081]
2090 tges=TEST(robx*32-18,400-(roby*16-6)) [16501]
2100 PEN#1,0:LOCATE#1,robx-1,roby [17691]
2110 PRINT#1," "; [5821]
2120 PEN#1,1:LOCATE#1,robx,roby [15181]
2130 PRINT#1,CHR$(165); [10041]
2140 IF tges=3 OR tges=4 THEN SOUND 1,506, [75091]
100,,5,5:leben=leben-1:IF leben<1 THEN 461
0 ELSE PEN#2,6:LOCATE#2,18,2:PRINT#2,USING
"###";leben;:GOTO 1040
2150 IF tges=2 THEN SOUND 2,75,2,,2:gst=gs [45471]
t-1 ELSE GOTO 2180
2160 IF gst=0 THEN SOUND 1,358,110,,6,6:bi [77941]
ld=bild+1: PEN#2,6:LOCATE#2,11,2:PRINT#2,US
ING"###";bild;:LOCATE#2,4,2:PRINT#2,USING"#
";gst;:GOTO 1040
2170 PEN#2,6:LOCATE#2,4,2:PRINT#2,USING"## [19631]
";gst;
2180 GOTO 1580 [3191]
2190 ' [1171]
2200 '*** nach links *** [14741]
2210 ' [1171]
2220 IF beweg<>4 THEN 1580 [6241]
2230 robx=robx-1 [10601]
2240 IF robx<1 THEN robx=1 [17911]
2250 tges=TEST(robx*32-18,400-(roby*16-6)) [16501]
2260 PEN#1,0 [4661]
2270 LOCATE#1,robx+1,roby [12281]

```

Listing Rob

```

2280 PRINT#1," "; [5821]
2290 PEN#1,1 [4651]
2300 LOCATE#1,robx,roby [6471]
2310 PRINT#1,CHR$(165); [10041]
2320 IF tges=3 OR tges=4 THEN SOUND 1,506, [75091]
100,,5,5:leben=leben-1:IF leben<1 THEN 461
0 ELSE PEN#2,6:LOCATE#2,18,2:PRINT#2,USING
"###";leben;:GOTO 1040
2330 IF tges=2 THEN SOUND 2,75,2,,2:gst=gs [45431]
t-1 ELSE GOTO 2360
2340 IF gst=0 THEN SOUND 1,358,110,,6,6:bi [77941]
ld=bild+1: PEN#2,6:LOCATE#2,11,2:PRINT#2,US
ING"###";bild;:LOCATE#2,4,2:PRINT#2,USING"#
";gst;:GOTO 1040
2350 PEN#2,6:LOCATE#2,4,2:PRINT#2,USING"## [19631]
";gst;
2360 GOTO 1580 [3191]
2370 ' [1171]
2380 '*** Bildrahmen / Bild 1 *** [26101]
2390 ' [1171]
2400 PEN#1,3 [4711]
2410 LOCATE#1,1,1:PRINT#1,CHR$(150); [16291]
2420 LOCATE#1,20,1:PRINT#1,CHR$(156); [27261]
2430 LOCATE#1,1,22:PRINT#1,CHR$(147); [23411]
2440 LOCATE#1,20,22:PRINT#1,CHR$(153); [22741]
2450 FOR xa=2 TO 19 [9951]
2460 LOCATE#1,xa,1:PRINT#1,CHR$(154); [5901]
2470 LOCATE#1,xa,22:PRINT#1,CHR$(154); [23601]
2480 NEXT [3501]
2490 FOR ya=2 TO 21 [11331]
2500 LOCATE#1,1,1,1:PRINT#1,CHR$(149); [21071]
2510 LOCATE#1,20,1,1:PRINT#1,CHR$(149); [23411]
2520 NEXT [3501]
2530 RETURN [5551]
2540 ' [1171]
2550 '*** Bild 2 *** [13561]
2560 ' [1171]
2570 RESTORE 2640 [8541]
2580 PEN#1,3 [4711]
2590 READ xa,ya,za [6511]
2600 IF xa=0 THEN RETURN [14041]
2610 LOCATE#1,xa,1,1:PRINT#1,CHR$(149); [4251]
2620 PRINT#1,CHR$(za); [7171]
2630 GOTO 2590 [3891]
2640 DATA 12,1,158,12,2,149,12,3,149,12,4, [67591]
149,12,5,149,12,6,149,12,7,149,12,8,149,12
,9,149,1,12,151,2,12,154,3,12,154,4,12,154
,5,12,154,6,12,154
2650 DATA 7,12,154,10,12,154,11,12,154,12, [30321]
12,156,12,13,149,12,14,149,12,15,149
2660 DATA 12,16,149,12,17,149,12,18,149,12 [36421]
,19,149,12,20,149,12,21,149,12,22,155,0,0,
0
2670 ' [1171]
2680 '*** Bild 3 *** [14291]
2690 ' [1171]
2700 FOR xa=2 TO 19 [9951]
2710 IF xa>8 AND xa<11 THEN 2740 [24121]
2720 LOCATE#1,xa,6 [5041]
2730 PRINT#1,CHR$(154); [11911]
2740 IF xa>11 AND xa<14 THEN 2770 [11351]
2750 LOCATE#1,xa,12 [4651]
2760 PRINT#1,CHR$(154); [11911]
2770 NEXT [3501]
2780 FOR ya=15 TO 21 [13721]
2790 LOCATE#1,11,1,1:PRINT#1,CHR$(149); [7081]
2800 PRINT#1,CHR$(149); [10041]
2810 NEXT [3501]
2820 RESTORE 2880 [9021]
2830 READ xa,ya,za [6511]
2840 IF xa=0 THEN RETURN [14041]
2850 LOCATE#1,xa,1,1:PRINT#1,CHR$(149); [4251]
2860 PRINT#1,CHR$(za); [7171]
2870 GOTO 2830 [3971]
2880 DATA 1,6,151,20,6,157,1,12,151,20,12, [25191]
157,11,22,155,0,0,0
2890 ' [1171]
2900 '*** Bild 4 *** [13411]
2910 ' [1171]
2920 RESTORE 2980 [8501]
2930 READ xa,ya,za [6511]

```

Listing Rob

```

2940 IF xa=0 THEN RETURN [1404]
2950 LOCATE#1, xa, ya [425]
2960 PRINT#1, CHR$(za); [717]
2970 GOTO 2930 [337]
2980 DATA 11,4,150,12,4,154,13,4,154,14,4, [8046]
154,15,4,154,16,4,154,17,4,156,11,5,149,17
,5,149,11,6,149,17,6,149,11,7,151,2,7,154,3
,7,154,4,7,154,5,7,154,6,7,154,11,7,149,17
,7,149,11,8,149 [10182]
2990 DATA 17,8,149,11,9,149,17,9,149,11,10
,149,17,10,149,11,11,147,12,11,154,13,11,1
54,14,11,154,17,11,149,17,12,149,17,13,149
,11,14,150,12,14,154,13,14,154,14,14,154,1
7,14,149,1,15,151,2,15,154 [9028]
3000 DATA 3,15,154,4,15,154,5,15,154,6,15,
154,11,15,149,17,15,149,11,16,149,17,16,14
9,11,17,149,17,17,149,11,18,149,17,18,149,
11,19,147,12,19,154,13,19,154,14,19,154,15
,19,154,16,19,154,17,19,153,0,0,0 [117]
3010 ' [1349]
3020 '*** Bild 5 *** [117]
3030 ' [1339]
3040 FOR xa=4 TO 17 [506]
3050 LOCATE#1, xa, 4 [1191]
3060 PRINT#1, CHR$(154); [503]
3070 LOCATE#1, xa, 7 [1191]
3080 PRINT#1, CHR$(154); [459]
3090 LOCATE#1, xa, 10 [1191]
3100 PRINT#1, CHR$(154); [464]
3110 LOCATE#1, xa, 13 [1191]
3120 PRINT#1, CHR$(154); [461]
3130 LOCATE#1, xa, 16 [1191]
3140 PRINT#1, CHR$(154); [450]
3150 LOCATE#1, xa, 19 [1191]
3160 PRINT#1, CHR$(154); [350]
3170 NEXT [555]
3180 RETURN [117]
3190 ' [1514]
3200 '*** Bild 6 *** [117]
3210 ' [1123]
3220 FOR ya=2 TO 19 [622]
3230 LOCATE#1, 4, ya [1004]
3240 PRINT#1, CHR$(149); [1448]
3250 IF ya<5 THEN 3290 [117]
3260 ' [822]
3270 LOCATE#1, 7, ya [1004]
3280 PRINT#1, CHR$(149); [1011]
3290 IF ya<8 THEN 3320 [659]
3300 LOCATE#1, 17, ya [1004]
3310 PRINT#1, CHR$(149); [1208]
3320 IF ya<11 THEN 3350 [843]
3330 LOCATE#1, 14, ya [1004]
3340 PRINT#1, CHR$(149); [350]
3350 NEXT [1276]
3360 FOR xa=8 TO 17 [506]
3370 LOCATE#1, xa, 4 [1191]
3380 PRINT#1, CHR$(154); [3182]
3390 IF xa<17 THEN LOCATE#1, xa, 7:PRINT#1,C
HR$(154); [2956]
3400 IF xa<14 THEN LOCATE#1, xa, 10:PRINT#1,
CHR$(154); [350]
3410 NEXT [783]
3420 RESTORE 3480 [651]
3430 READ xa, ya, za [1404]
3440 IF xa=0 THEN RETURN [425]
3450 LOCATE#1, xa, ya [717]
3460 PRINT#1, CHR$(za); [515]
3470 GOTO 3430 [3488]
3480 DATA 4,1,158,7,4,150,7,7,151,17,7,156
,7,10,151,14,10,156,0,0,0 [117]
3490 ' [1459]
3500 '*** Bild 7 *** [117]
3510 ' [1293]
3520 FOR xa=5 TO 16 [506]
3530 LOCATE#1, xa, 4 [1191]
3540 PRINT#1, CHR$(154); [450]
3550 LOCATE#1, xa, 19 [1191]
3560 PRINT#1, CHR$(154); [350]
3570 NEXT [751]
3580 RESTORE 3640 [651]
3590 READ xa, ya, za

```

Listing Rob

```

3600 IF xa=0 THEN RETURN [1404]
3610 LOCATE#1, xa, ya [425]
3620 PRINT#1, CHR$(za); [717]
3630 GOTO 3590 [491]
3640 DATA 4,4,150,11,4,158,17,4,156,4,5,14
9,11,5,149,17,5,149,4,6,149,17,6,149,4,7,1
49,17,7,149,4,8,149,11,8,149,17,8,149,4,9,
147,5,9,154,6,9,154,7,9,154,8,9,154 [8210]
3650 DATA 9,9,154,10,9,154,11,9,153,14,9,1
54,15,9,154,16,9,154,17,9,157,17,10,149,17
,11,149,4,12,150,5,12,154,6,12,154,7,12,15
4,8,12,154,9,12,154,10,12,154,11,12,156,14
,12,154,15,12,154 [4603]
3660 DATA 16,12,154,17,12,157,4,13,149,11,
13,149,17,13,149,4,14,149,11,14,149,17,14,
149 [6739]
3670 DATA 4,15,149,11,15,149,17,15,149,4,1
6,149,11,16,149,17,16,149,4,17,149,17,17,1
49,4,18,149,17,18,149,4,19,147,17,19,153,0
,0,0 [117]
3680 ' [1370]
3690 '*** Bild 8 *** [117]
3700 ' [1123]
3710 FOR ya=2 TO 19 [622]
3720 LOCATE#1, 4, ya [1004]
3730 PRINT#1, CHR$(149); [350]
3740 NEXT [1228]
3750 FOR xa=9 TO 17 [459]
3760 LOCATE#1, xa, 10 [1191]
3770 PRINT#1, CHR$(154); [464]
3780 LOCATE#1, xa, 13 [1191]
3790 PRINT#1, CHR$(154); [350]
3800 NEXT [1233]
3810 FOR xa=9 TO 12 [506]
3820 LOCATE#1, xa, 4 [1191]
3830 PRINT#1, CHR$(154); [461]
3840 LOCATE#1, xa, 16 [1191]
3850 PRINT#1, CHR$(154); [450]
3860 LOCATE#1, xa, 19 [1191]
3870 PRINT#1, CHR$(154); [350]
3880 NEXT [1381]
3890 FOR xa=15 TO 19 [506]
3900 LOCATE#1, xa, 4 [1191]
3910 PRINT#1, CHR$(154); [995]
3920 IF xa<18 THEN 3950 [534]
3930 LOCATE#1, xa, 8 [1191]
3940 PRINT#1, CHR$(154); [350]
3950 NEXT [882]
3960 RESTORE 4020 [651]
3970 READ xa, ya, za [1404]
3980 IF xa=0 THEN RETURN [425]
3990 LOCATE#1, xa, ya [717]
4000 PRINT#1, CHR$(za); [479]
4010 GOTO 3970 [2852]
4020 DATA 4,1,158,8,4,150,20,4,157,8,5,149
,8,6,149,8,7,149,8,8,149,17,8,150,20,8,157
4030 DATA 8,9,149,17,9,149,8,10,147,17,10,
153,8,13,150,17,13,156,8,14,149,17,14,149,
8,15,149 [4032]
4040 DATA 17,15,149,8,16,147,17,16,149,17,
17,149,17,18,149,8,19,150,17,19,149,8,20,1
49,17,20,149,8,21,149,17,21,149,8,22,155,1
7,22,155,0,0,0 [5719]
4050 ' [117]
4060 '*** Bild 9 *** [1378]
4070 ' [117]
4080 RESTORE 4140 [779]
4090 READ xa, ya, za [651]
4100 IF xa=0 THEN RETURN [1404]
4110 LOCATE#1, xa, ya [425]
4120 PRINT#1, CHR$(za); [717]
4130 GOTO 4090 [335]
4140 DATA 4,4,150,5,4,154,6,4,158,7,4,154,
8,4,154,9,4,154,12,4,154,13,4,154,14,4,154
,15,4,154,16,4,154,17,4,156,6,5,149,17,5,1
49,6,6,149,8,6,149,9,6,150,10,6,154,11,6,1
54,12,6,154,13,6,154,14,6,154,17,6,149 [7618]
4150 DATA 4,7,149,6,7,149,8,7,149,9,7,149,
17,7,149,4,8,149,6,8,149,8,8,149,9,8,149,1
1,8,150,12,8,154,13,8,154,14,8,156,17,8,14
9,4,9,149,6,9,149,8,9,149,9,9,149,11,9,149

```

Listing Rob

```

, 14, 9, 149, 17, 9, 149, 4, 10, 149
4160 DATA 6, 10, 149, 8, 10, 149, 9, 10, 149, 11, 10 [11290]
, 149, 14, 10, 149, 17, 10, 149, 4, 11, 149, 6, 11, 149
, 8, 11, 149, 9, 11, 149, 11, 11, 149, 14, 11, 149, 17,
11, 149, 4, 12, 149, 6, 12, 149, 8, 12, 149, 9, 12, 149
, 11, 12, 149, 14, 12, 149, 4, 13, 149, 6, 13, 149
4170 DATA 8, 13, 149, 9, 13, 149, 11, 13, 149, 14, 1 [7531]
3, 149, 4, 14, 149, 6, 14, 149, 8, 14, 147, 9, 14, 153,
11, 14, 149, 14, 14, 149, 17, 14, 149, 4, 15, 149, 6, 1
5, 149, 11, 15, 149, 14, 15, 149, 17, 15, 149
4180 DATA 4, 16, 149, 6, 16, 147, 7, 16, 154, 8, 16, [7749]
154, 9, 16, 154, 10, 16, 154, 11, 16, 153, 14, 16, 149
, 17, 16, 149, 4, 17, 149, 14, 17, 149, 17, 17, 149, 4,
18, 149, 17, 18, 149, 4, 19, 147, 5, 19, 154, 6, 19, 15
4
4190 DATA 9, 19, 154, 10, 19, 154, 11, 19, 154, 12, [2763]
19, 154, 15, 19, 154, 16, 19, 154, 17, 19, 153, 0, 0, 0
4200 '
4210 '*** Bild 10 ***
4220 '
4230 gst=11
4240 FOR xa=5 TO 16
4250 LOCATE#1, xa, 4
4260 PRINT#1, CHR$(154);
4270 IF xa>13 THEN 4300
4280 LOCATE#1, xa, 19
4290 PRINT#1, CHR$(154);
4300 NEXT
4310 FOR xa=8 TO 13
4320 LOCATE#1, xa, 7
4330 PRINT#1, CHR$(154);
4340 IF xa>10 THEN 4370
4350 LOCATE#1, xa, 16
4360 PRINT#1, CHR$(154);
4370 NEXT
4380 FOR ya=5 TO 21
4390 LOCATE#1, 17, ya
4400 PRINT#1, CHR$(149);
4410 IF ya>18 THEN 4440
4420 LOCATE#1, 4, ya
4430 PRINT#1, CHR$(149);
4440 NEXT
4450 FOR ya=8 TO 18
4460 LOCATE#1, 14, ya
4470 PRINT#1, CHR$(149);
4480 IF ya>15 THEN 4510
4490 LOCATE#1, 7, ya
4500 PRINT#1, CHR$(149);
4510 NEXT
4520 RESTORE 4580
4530 READ xa, ya, za
4540 IF xa=0 THEN PEN#1, 2: LOCATE#1, 9, 12: PR [2710]
INT#1, CHR$(168);: RETURN
4550 LOCATE#1, xa, ya [425]
4560 PRINT#1, CHR$(za); [717]
4570 GOTO 4530 [335]
4580 DATA 4, 4, 150, 17, 4, 156, 7, 7, 150, 14, 7, 15 [4527]
6, 10, 10, 154, 11, 10, 156, 7, 16, 147, 11, 16, 153, 1
1, 11, 149, 11, 12, 149, 11, 13, 149
4590 DATA 11, 14, 149, 11, 15, 149, 4, 19, 147, 14, [2347]
19, 153, 17, 22, 155, 0, 0, 0
4600 ' [117]
4610 '*** Bild nicht geschafft *** [1413]
4620 ' [117]
4630 CLS [91]
4640 INK 3, 8: PEN 3 [172]
4650 LOCATE 5, 8: PRINT"Leider nicht" [2738]
4660 LOCATE 5, 11: PRINT"geschafft !!" [2472]
4670 LOCATE 8, 23: PRINT"<SPACE>" [1782]
4680 IF INKEY(47)<>0 THEN 4680 [802]
4690 GOTO 390 [494]
4700 ' [117]
4710 '*** 10. Bild geschafft *** [617]
4720 ' [117]
4730 CLS: INK 0, 0: INK 1, 0: INK 2, 0: INK 3, 2: I [4169]
NK 4, 26: INK 5, 0: INK 6, 18: INK 8, 8: INK 9, 3: I
NK 12, 0
4740 DEG: GRAPHICS PEN 2: MOVE 0, 192: DRAW 64 [5017]
0, 192: MOVE 0, 190: FILL 2: RESTORE 4800
4750 GRAPHICS PEN 1: MOVE 224, 80: DRAW 352, 8 [5087]
0: DRAW 416, 112: DRAW 416, 192: DRAW 288, 192: D

```

Listing Rob

```

RAW 224, 160: DRAW 224, 80
4760 MOVE 256, 82: FILL 1: MOVE 410, 110: FILL [2250]
1: MOVE 256, 162: FILL 1
4770 FOR i=1 TO 8: READ xa(i): IF xa(i)=1000 [2472]
THEN 4830
4780 NEXT [350]
4790 GRAPHICS PEN xa(1): ORIGIN xa(2), xa(3) [8482]
: MOVE xa(4), xa(5): FOR i=1 TO 360 STEP 10: D
RAW xa(5)*SIN(i), xa(5)*COS(i): NEXT: MOVE xa
(6), xa(7): FILL xa(8): GOTO 4770
4800 DATA 5, 480, 320, 0, 32, 0, 0, 5, 4, 96, 352, 0, [6970]
16, -6, 0, 4, 4, 112, 352, 0, 16, 6, 0, 4, 4, 112, 336, 0
, 16, 0, -6, 4, 4, 96, 336, 0, 16, 0, -6, 4, 4, 125, 344,
0, 16, 6, 0, 4, 4, 80, 344, 0, 16, -6, 0, 4
4810 DATA 3, 256, 128, 0, 8, 0, 0, 1, 3, 256, 128, 0, [4478]
2, 0, 0, 3, 3, 320, 128, 0, 8, 0, 0, 1, 3, 320, 128, 0, 2,
0, 0, 3, 9, 288, 112, 0, 2, 0, 0, 9
4820 DATA 9, 416, 96, 0, 6, 0, 0, 1, 9, 192, 96, 0, 6, [4369]
0, 0, 1, 9, 224, 224, 0, 6, 0, 0, 1, 9, 416, 224, 0, 6, 0,
0, 1, 9, 384, 128, 0, 6, 0, 0, 1, 1000
4830 ORIGIN 0, 0: GRAPHICS PEN 5: RESTORE 488 [2017]
0
4840 FOR i=1 TO 4: READ xa(i): IF xa(i)=0 TH [2014]
EN 4890
4850 NEXT [350]
4860 MOVE xa(1), xa(2): DRAW xa(3), xa(4) [2039]
4870 GOTO 4840 [403]
4880 DATA 480, 304, 480, 256, 496, 312, 544, 288, [7121]
512, 320, 560, 320, 496, 328, 544, 352, 480, 336, 48
0, 384, 464, 328, 416, 352, 448, 320, 400, 320, 464,
312, 416, 288, 0
4890 GRAPHICS PEN 12: RESTORE 4940 [1724]
4900 FOR i=1 TO 4: READ xa(i): IF xa(i)=0 TH [2010]
EN 4950
4910 NEXT [350]
4920 MOVE xa(1), xa(2): DRAW xa(3), xa(4) [2039]
4930 GOTO 4900 [438]
4940 DATA 160, 304, 150, 310, 160, 304, 170, 310, [5223]
288, 336, 278, 342, 288, 336, 298, 342, 224, 368, 21
4, 374, 224, 368, 234, 374, 0
4950 GRAPHICS PEN 9: MOVE 224, 80: DRAW 352, 8 [10688]
0: DRAW 416, 112: DRAW 416, 192: DRAW 288, 192: D
RAW 224, 160: DRAW 224, 80: MOVE 224, 160: DRAW
352, 160: DRAW 416, 192: MOVE 352, 160: DRAW 352
, 80
4960 MOVE 272, 96: DRAW 304, 96: DRAW 310, 100: [8938]
MOVE 272, 96: DRAW 270, 100: GRAPHICS PEN 1: MO
VE 272, 80: DRAW 272, 32: DRAW 230, 32: DRAW 234
, 34: DRAW 260, 44: DRAW 260, 80: MOVE 270, 77: FI
LL 1
4970 GRAPHICS PEN 9: MOVE 272, 80: DRAW 272, 3 [9505]
2: DRAW 230, 32: DRAW 234, 34: DRAW 260, 44: DRAW
260, 80: GRAPHICS PEN 1: MOVE 320, 80: DRAW 31
2, 64: DRAW 326, 48: DRAW 318, 32: DRAW 342, 48: D
RAW 324, 64: DRAW 332, 80: MOVE 328, 78: FILL 1
4980 GRAPHICS PEN 9: MOVE 320, 80: DRAW 312, 6 [9787]
4: DRAW 326, 48: DRAW 318, 32: DRAW 342, 48: DRAW
324, 64: DRAW 332, 80: MOVE 224, 80: DRAW 352, 8
0: MOVE 420, 96: DRAW 388, 128: MOVE 412, 96: DRA
W 380, 128: MOVE 408, 104: FILL 1
4990 MOVE 196, 96: DRAW 224, 112: MOVE 188, 96: [7191]
DRAW 224, 120: MOVE 222, 112: FILL 1: MOVE 206,
104: FILL 1: MOVE 288, 176: DRAW 230, 224: MOVE
352, 176: DRAW 412, 224
5000 GRAPHICS PEN 6: RESTORE 5040 [1770]
5010 FOR i=1 TO 4: READ xa(i): IF xa(i)=1000 [2043]
THEN 5060
5020 NEXT [350]
5030 MOVE xa(1), xa(2): DRAW xa(3), xa(4): GOT [1971]
O 5010
5040 DATA 128, 144, 144, 160, 128, 144, 132, 162, [10714]
128, 144, 120, 156, 64, 64, 94, 80, 64, 64, 72, 84, 64
, 64, 36, 79, 128, 16, 160, 30, 128, 16, 125, 32, 125,
16, 96, 28, 448, 32, 475, 44, 448, 32, 444, 45, 448, 3
2, 418, 50
5050 DATA 576, 48, 609, 60, 576, 48, 584, 64, 576, [7871]
48, 568, 64, 576, 48, 544, 61, 512, 80, 528, 96, 512,
80, 512, 88, 512, 80, 496, 97, 480, 128, 508, 140, 48
0, 128, 484, 140, 480, 128, 472, 144, 480, 128, 462,
134, 1000
5060 GRAPHICS PEN 8: PLOT 64, 128: PLOT 70, 17 [10341]

```

Listing Rob

```

6:PLOT 32,154:PLOT 144,88:PLOT 48,32:PLOT
348,16:PLOT 448,70:PLOT 580,120:PLOT 512,1
76:PLOT 608,160:PLOT 544,32:PLOT 128,120
5070 INK 0,2:INK 1,6,15:INK 2,9:INK 3,2:IN [3656]
K 4,26:INK 5,24:INK 6,18:INK 8,8:INK 9,3:I
NK 12,0
5080 PEN 11:INK 11,1:LOCATE 1,8:PRINT"H U [5005]
R R A":LOCATE 7,10:PRINT"GESCHAFFT"
5090 ' [117]
5100 '*** Geschäftsound *** [2026]
5110 ' [117]
5120 RESTORE 5180 [715]
5130 READ xa,ya [693]
5140 IF INKEY(47)=0 THEN FOR i=1 TO 600:NE [2292]
XT:INK 0,0:CLS:GOTO 390
5150 IF xa=10 THEN GOTO 5120 [938]
5160 SOUND 1,xa,ya,10:SOUND 2,xa,ya,8:SOU [4111]
D 2,190,ya,4:SOUND 4,284,ya
5170 GOTO 5130 [377]
5180 DATA 95,15,95,15,95,30 [745]
5190 DATA 119,15,142,15,142,30,71,15 [1135]
5200 DATA 71,15,71,30,89,15 [977]
5210 DATA 106,15,95,30,95,15,95,15 [1227]
5220 DATA 95,30,89,15,71,15 [906]
5230 DATA 119,30,127,15,142,15 [1365]
5240 DATA 127,30,95,15,95,15 [1161]
5250 DATA 95,30,119,15,142,15 [645]
5260 DATA 142,15,71,15,71,15,71,30 [1359]
5270 DATA 89,15,106,15,95,30 [692]
5280 DATA 89,15,71,15,119,30 [751]
5290 DATA 117,15,142,15,127,30 [844]
5300 DATA 119,15,127,15,142,30 [1185]
5310 DATA 71,15,71,15,63,30 [723]
5320 DATA 95,15,95,15,95,30 [745]
5330 DATA 80,15,63,15,71,30 [860]
5340 DATA 89,15,106,15,95,30 [692]
5350 DATA 71,15,71,15,63,15,71,15 [950]
5360 DATA 80,15,89,15,95,30 [890]
5370 DATA 80,15,63,15,71,30 [860]
5380 DATA 89,15,106,15,95,30 [692]
5390 DATA 95,15,95,15,95,30 [745]
5400 DATA 119,15,142,15,142,30 [765]
5410 DATA 71,15,71,15,71,30 [690]
5420 DATA 89,15,106,15,95,30 [692]
5430 DATA 89,15,71,15,119,30 [751]
5440 DATA 127,15,142,15,127,30 [1413]
5450 DATA 119,15,127,15,142,30,10,10 [1555]

```

Listing Rob



Deutsches Datum unter CP/M Plus

CP/M Plus ist ein sehr benutzerfreundliches Betriebssystem. Das zeigt sich besonders darin, daß Dienstprogramme wie SETKEYS und LANGUAGE eine einfache Möglichkeit darstellen, Bildschirm und Tastatur der jeweiligen Landessprache anzupassen.

Die Eingabe von LANGUAGE 2 zum Beispiel schaltet den deutschen Zeichensatz ein. Der Aufbau von Tastaturdefinitionen für SETKEYS wird in der HELP-Datei recht gut beschrieben.

Schade nur, daß all diese Freundlichkeit beim Datum endet. Denn die Datums- und Uhrzeit-Funktionen, die mit INIT-DIR und SET zu erreichen sind, haben es ganz schön in sich. Aber wer findet es

schon toll wenn sich der 2. Aug. 88 dabei als 08/02/88 präsentiert und auch so eingegeben werden muß. Zu leicht können dabei Fehler entstehen.

Daher die Idee zu drei kleinen Programmänderungen. Betroffen sind die Dienstprogramme DATE.COM, DIR.COM und SHOW.COM.

Nach Eingabe der Assembler-Quellcodes (Mit WS, ED oder einem anderen,

geeigneten Editor) werden diese mit MAC.COM assembliert. Dazu werden auf eine Leerdiskette (Programmfehler können grauenhaftes bewirken) die Programme MAC.COM und SID.COM kopiert. Um nun DATE zu ändern, kopiert man noch DATE.COM und den vorher erstellten Quellcode P-DATE.ASM hinzu.

Die weiteren Eingaben lauten dann

```
A < MAC P-DATE $PZ SZ
A < SID DATE.COM #RP-DATE.HEX
# WDATE.COM,100,D00 # G0
```

Für SHOW.COM und DIR.COM wird sinngemäß verfahren, wobei jedoch die #W-Zeile folgende Angaben enthalten muß:

```
# WSHOW.COM,100,2200 bzw.
# WDIR.COM,100,3A00
```

(J. Wieck/JB)

```

1 P-DATE.ASM
2
3 03.10.87
4
5 [ndert DATE.COM auf deutsche Ein-/Ausgaben.
6
7 Assemblieren mit MAC, DATE.COM mit SID laden, P-DATE.HEX mit
8 R-Befehl nachladen und DATUM.COM mit W-Befehl speichern.
9
10
11 ;Wochentage
12
13 ORG 001A4H
14
15 DB
16 ;Son$Mon$Die$Mit$Don$Fre$Sam$
17
18 ;Meldungen
19
20 ORG 001CCH
21
22 DB DB 'U: SS) :$',0 ;Heutiges Datum (TT.MM.JJ) :$',0
23
24 ORG 0020EH
25
26 TEXTST
27 DB 'U: $'
28
29 ORG 00252H
30
31 DB 'F: benötigt CP/M 3$'
32
33 ORG 0026AH
34
35 DB 'F: Falsche Zeit-/Datumsangaben. $'
36
37 ;Programmänderungen
38
39 ORG 00329H
40
41 ORG 00329H
42
43 JNZ CHGE2
44
45 ORG 00348H
46
47 CALL CHGE1
48
49 ORG 00440H
50
51 LXI B,TEXTST
52
53 ORG 005A2H
54
55 MVI C,' '
56
57 ORG 0073BH
58
59 MVI C,' '
60
61 ORG 0074AH
62
63 MVI C,' '
64
65 ORG 00950H
66
67 LHL 00B97H
68
69 ORG 00957H
70
71 LHL 00B96H
72
73 ;Programmerweiterung
74
75
76
77 ORG 00C80H
78
79 CHGE1:
80 CALL 004C5H
81 PUSH H
82 LHL 00B66H
83 PUSH H
84 LHL 00B69H
85 SHLD 00B66H
86 POP H
87 SHLD 00B69H
88 POP H
89 RET

```

Listing: Deutsches Datum

```

90 CHGE2:
91 PUSH H
92 LHL 00082H
93 PUSH H
94 LHL 00085H
95 SHLD 00082H
96 POP H
97 SHLD 00085H
98 POP H
99 JMP 00422H
100
101 END

```

```

1 P-DIR.ASM
2
3 03.10.87
4
5 [ndert DIR.COM auf deutsche Datumsausgaben.
6
7 Assemblieren mit MAC, DIR.COM mit SID laden, P-DIR.HEX mit
8 R-Befehl nachladen und neues DIR.COM mit W-Befehl speichern.
9
10
11 ;Wochentage
12
13
14
15 ORG 032AEH
16
17 DB
18 ;Son$Mon$Die$Mit$Don$Fre$Sam$
19
20 ;Programmänderungen
21
22
23 ORG 0337AH
24
25 MVI C,' '
26
27 ORG 03528H
28
29 LHL 039CBH
30
31 ORG 0352FH
32
33 LHL 039CAH
34
35 END

```

```

1 P-SHOW.ASM
2
3 03.10.87
4
5 [ndert SHOW.COM auf deutsche Datumsausgaben.
6
7 Assemblieren mit MAC, SHOW.COM mit SID laden, P-SHOW.HEX mit
8 R-Befehl nachladen und neues SHOW.COM mit W-Befehl speichern.
9
10
11 ;Wochentage
12
13
14
15 ORG 00319H
16
17 DB
18 ;Son$Mon$Die$Mit$Don$Fre$Sam$
19
20 ;Programmänderungen
21
22
23 ORG 00956H
24
25 MVI C,' '
26
27 ORG 00CEEH
28
29 LHL 020FEH
30
31 ORG 00CF5H
32
33 LHL 020FDH
34
35 END

```

Listing: Deutsches Datum

Grafiken und Texte drucken unter CP/M 2.2

Text- und Grafikharcopies auf Tastendruck, Features die bisher den Computern der Amigaklasse vorbehalten waren, sind nun auch auf allen CPC's unter CP/M 2.2 realisierbar. Zwei leistungsstarke Tools im BIOS-Stack sorgen für die notwendige Power. Text- oder Grafikhotsots aus laufenden CP/M-Programmen? Kein Problem, denn SCRCOPY+ und DINCOPY+ machen's möglich.

Benötigt wird lediglich ein CP/M Editor, ein 8080-Assembler und ein 8080-Linker. All dies finden Sie unter dem Namen ED.COM, ASM.COM und LOAD.COM auf der mitgelieferten CP/M-Systemdiskette. Nun brauchen Sie nur noch die abgedruckten Assembler-Quellprogramme abzutippen, jeweils mit ASM zu assemblieren und mit LOAD zu binden. Doch bevor Sie sich auf Ihren Rechner stürzen, eine kurze Beschreibung, was Sie danach erwartet.

SCRCOPY+ ermöglicht Textharcopies unter CP/M 2.2 aus beliebigen Anwenderprogrammen. Aktiviert wird die Routine durch die Tastenkombination CTRL und F1. Ist Ihr Drucker nicht empfangsbereit, so wird der Aufruf ignoriert. Selbstverständlich erkennt SCRCOPY+ den aktuellen Bildschirmmodus automatisch und richtet die Druckausgabe darauf ein. DINCOPY+ schließlich kann, was SCRCOPY+ nicht kann, nämlich monochrome Grafik-Harcopies auf Ta-

stendruck. Die Ausgabe erfolgt, nach Drücken der Tasten CTRL und F2, im DIN A4-Format. Das ganze funktioniert reibungslos, solange Ihr Drucker den EPSON-Standard-Code versteht. Falls nicht, müssen Sie die Drucksteuerzeichen im Assembler-Quellprogramm entsprechend anpassen.

Beide Tools, also sowohl SCRCOPY+ als auch DINCOPY+, können einzeln oder gemeinsam verwendet werden. Sie benötigen übrigens keinerlei Speicherplatz im TPA-Bereich und kollidieren somit mit keinem anderen CP/M-Programm!! Nach einmaligem Aufruf nisten Sie sich im BIOS-Stack ein und sind auch gegen eventuelle Warm-Boots (das passiert, wenn Sie CTRL-C drücken) völlig immun, d.h. sie bleiben Ihnen bis zu einem kompletten Reset permanent erhalten.

So, nun sollten Sie aber Ihrem CPC und dem antiken CP/M 2.2 auf die Sprünge helfen. Was Sie zu tun haben, wissen Sie bereits. Na denn.....

(B. Ott)

für 464-664-6128



```

1 ;*****
2 ;* DINCOPY VERSION 1.1 (DINCOPY+)*
3 ;* (C) 1987 BY BERND OTT *
4 ;*
5 ;* Speicherbereich BF10H - BFADH *
6 ;*****
7 ;
8     ORG     0100H
9 ;
10    ADDR   EQU   0BF28H      ; Lage der Routine im Bios Stack
11    VERS   EQU   0BD0DH      ; Versatz
12 ;
13    BDOS   EQU   00005H      ; Bdos Einsprung
14    FIRM   EQU   0C168H      ; Firmware Einsprung
15    ADDFAST EQU 0BCE3H      ; New Frame Fly Back
16    GETKEY EQU 0BB1EH      ; Tastaturabfrage
17    GRAINIT EQU 0BBBAH      ; Grafikmodus einschalten
18    GETPAP EQU 0BBE7H      ; Papierfarbe feststellen
19    TESTPIX EQU 0BBF0H      ; Pixeltest an Bildschirm
20    BUSY   EQU   0BD2EH      ; Drucker empfangsbereit?
21    SEND   EQU   0BD28H      ; Zeichen drucken
22 ;
23    INIT   DI           ;
24         LXI   H,TICKB     ; Routine verschieben
25         LXI   D,ADDR      ;
26         LXI   B,ENDE-TICKB ;
27         DW   0B0EDH       ; Z80 LDJR
28         LXI   H,ADDR      ;
29         CALL  FIRM        ;
30         DW   ADDFAST      ; Routine installieren
31         LXI   D,LOGON$    ; Einschaltmeldung ausgeben
32         MVI   C,009H      ;
33         CALL  BDOS        ;
34         EI           ;
35         RET           ; Zurueck ins CP/M
36 ;
37    LOGON$ DW   00A0DH
38         DB   'DINCOPY V1.2 INSTALLED- Use CTRL '

```

Listing Harcopies

```

39     DB   '> F2 < to copy Grafic'
40     DW   00A0DH
41     DB   '$'
42 ;
43    TICKB DW   00000H      ; Kettungspointer
44    EVBLCK DW 00000H      ; Kettung Pending Queue
45     DB   000H           ; Counter
46     DB   081H           ; Klasse
47     DW   CENTRAL+VERS   ; Adresse der Routine
48    STEUER DB   00DH,00AH,01BH ; CR & LF
49         DB   041H,007H,01BH ; 4/72 Zoll Vorschub
50         DB   04BH,000H,002H ; Doppelte Dichte / 800 Punkte
51 ;
52    CENTRAL MVI  A,00EH      ; Pruefe F2-Taste
53         CALL  GETKEY      ;
54         RZ           ; ...nein, dann Rueckkehr
55         DW   079CBH      ; Pruefe CTRL-Taste (Z80 BIT 7,C)
56         CNZ   BUSY       ; Drucker bereit ?
57         RZ           ;
58         RC           ; falls nicht, dann Rueckkehr
59 ;
60    DINCOP CALL  GRAINIT    ; Grafikmodus initialisieren
61         CALL  GETPAP      ; Papierfarbe speichern
62         DW   047EDH      ; (Z80 LD I,A)
63 ;
64         LXI   D,644       ; x:=644
65         DMARK1 MVI  B,009H ; Steuerzeichenfolge an Drucker
66         LXI   H,STEUER+VERS ; senden
67         DLOOP1 MOV   A,H   ;
68         CALL  PRINT+VERS  ;
69         INX   H           ;
70         DW   0F910H      ; (Z80 DJNZ DLOOP1)
71 ;
72         LXI   H,400       ; Y:=400
73         DMARK2 PUSH  D     ; X-Hauptposition sichern
74         DCX   D           ;
75         LXI   B,00700H   ; Schleifenzaehler = 7, Transbyte = 0
76 ;
77         DMARK3 PUSH  H     ; Y-Position sichern
78         PUSH  D           ; X-Hilfsposition sichern
79         PUSH  B           ; Schleifenzaehler und Transbyte sichern
80         CALL  TESTPIX    ; Pixeltest
81         POP   B

```

Listing Harcopies

```

82      MOV    D,A          ;
83      DW    057EDH       ; (Z80 LD A,I)
84      CMP    D           ; Paper = Pixel ?
85      STC                    ;
86      DW    00120H       ; (Z80 JR NZ,DMARKE4)
87      ANA    A           ;
88  DMARKE4 DW    011CBH    ; (Z80 RL C)
89      POP    D           ;
90      POP    H           ;
91      DCX   D           ; x-hilf:=x-hilf-1
92      DW    0E910H       ; (Z80 DJNZ DMARKE3)
93      ;
94      MOV    A,C         ;
95      CALL  PRINT+VERS   ;
96      CALL  PRINT+VERS   ;
97      POP    D           ; X-Position holen
98      DCX   H           ;
99      DCX   H           ;
100     SUB    A           ; Vergleichsoperation, da 16-Bit
101     CMP    L           ; Dekrementierung Flags nicht beeinflusst
102     DW    0D720H       ; (Z80 JR NZ,DMARKE2)
103     CMP    H           ;
104     DW    0D420H       ; (Z80 JR NZ,DMARKE2)
105     ;
106     MVI    B,112       ; Fuellbytes ausgeben
107  DLOOP2 CALL  PRINT+VERS ;
108     DW    0FB10H       ; (Z80 DJNZ DLOOP2)
109     ;
110     MVI    B,007H      ;
111  DLOOP3 DCX   D         ; X-Position-7
112     DW    0FD10H       ; (Z80 DJNZ DLOOP3)
113     CMP    E           ; Vergleichsoperation
114     DW    0B620H       ; (Z80 JR NZ,DMARKE1)
115     CMP    D           ;
116     DW    0B320H       ; (Z80 JR NZ,DMARKE1)
117     ;
118     MVI    A,01BH      ; Drucker neu initialisieren
119     CALL  PRINT+VERS   ;
120  EDINCOF MVI    A,040H ;
121     ;
122  PRINT  CALL  BUSY      ; UP Drucker bereit und Ausgabe
123     DW    0FB3BH       ; (Z80 JR C,PRINT)
124     CALL  SEND         ;
125     RET                    ;
126  ENDE  END              ;
127

```

```

1 ;
2 ;*****
3 ;* SCRCOPY VERSION 1.1 (SCRCOPY+) *
4 ;* (C) 1987 BY BERND OTT *
5 ;* *
6 ;* Speicherbereich BEC8H - BF1CH *
7 ;*****
8 ;
9      ORG    0100H
10 ;
11  ADDR  EQU    0BEC8H    ; Lage der Routine im Bios Stack
12  VERS  EQU    0BD6DH    ; Versatz

```

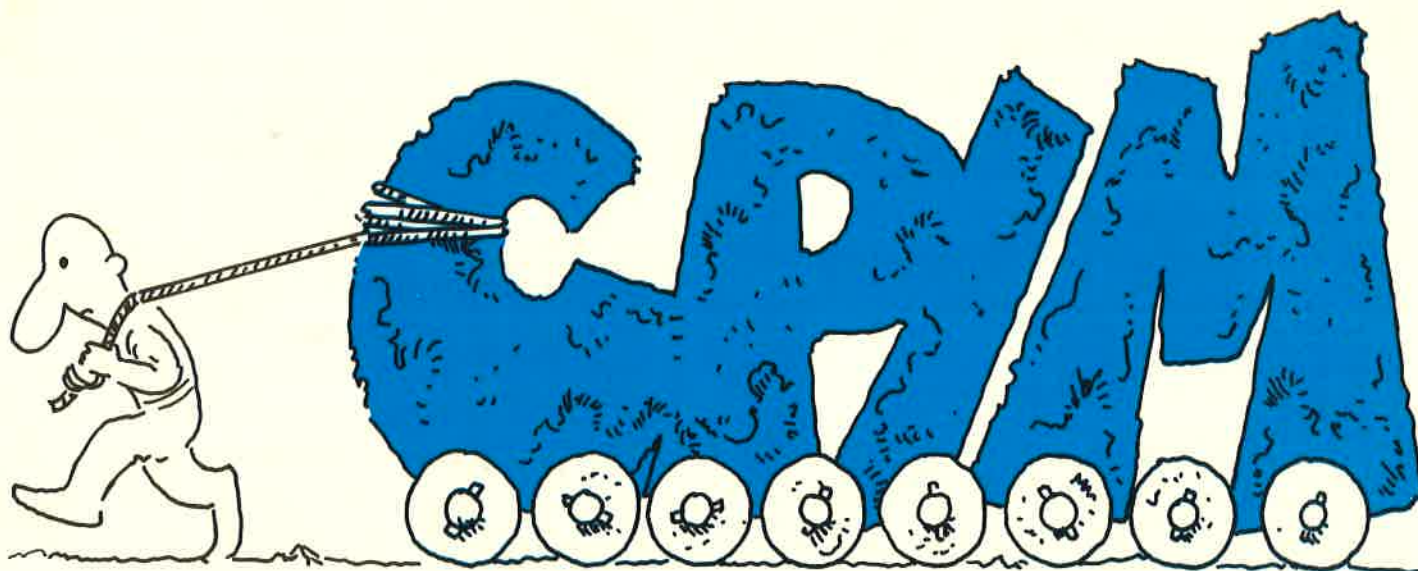
Listing Hardcopies

```

13 ;
14  BDOS  EQU    00005H    ; Bdos Einsprung
15  FIRM  EQU    0C168H    ; Firmware Einsprung
16  ADDFAST EQU    0BCE3H  ; New Frame Fly Back
17  GETKEY EQU    0BB1EH   ; Tastaturabfrage
18  GETMODE EQU    0BC17H  ; Modetest
19  GETCHAR EQU    0BB60H  ; Zeichen vom Bildschirm lesen
20  GETCUR EQU    0BB78H  ; Cursorposition feststellen
21  SETCUR EQU    0BB75H  ; Cursor positionieren
22  BUSY  EQU    0BD2EH    ; Drucker empfangsbereit?
23  SEND  EQU    0BD2BH    ; Zeichen drucken
24 ;
25  INIT  DI                    ;
26      LXI   H,TICKB        ; Routine verschieben
27      LXI   D,ADDR        ;
28      LXI   B,ENDE-TICKB  ;
29      DW    0B0EDH        ; Z80 LDIR
30      LXI   H,ADDR        ;
31      CALL  FIRM          ;
32      DW    ADDFAST       ; Routine installieren
33      LXI   D,LOGON$      ; Einschaltmeldung ausgeben
34      MVI   C,009H        ;
35      CALL  BDOS         ;
36      EI                    ;
37      RET                    ; Zurueck ins CP/M
38 ;
39  LOGON$ DW    00A0DH
40      DB    'SCRCOPY V1.1 INSTALLED- Use CTRL '
41      DB    '> F1 < for Texthardcopy'
42      DW    00A0DH
43      DB    '$'
44 ;
45  TICKB DW    00000H      ; Kettungspointer
46  EVBLOCK DW 00000H     ; Kettung Pending Queue
47      DB    000H          ; Counter
48      DB    081H         ; Klasse
49      DW    CENTRAL+VERS  ; Adresse der Routine
50 ;
51  CENTRAL MVI    A,00DH   ; Pruefe F1-Taste
52      CALL  GETKEY       ;
53      RZ                    ; ...nein, dann Rueckkehr
54      DW    079CBH       ; Pruefe CTRL-Taste (Z80 BIT 7,C)
55      CNZ   BUSY         ; Drucker bereit ?
56      RZ                    ;
57      RC                    ; falls nicht, dann Rueckkehr
58 ;
59  SCRCOP CALL  GETMODE    ; Zeilen und Spaltenanzahl feststellen
60      CALL  GETCUR       ; Cursorposition feststellen
61      XCHG                    ; und in DE speichern
62      MVI   L,001H       ; Zeile = 1
63  SLOOP2 MVI   H,001H    ; Spalte = 1
64  SLOOP1 PUSH  H         ;
65      CALL  SETCUR       ; Cursor setzen
66      CALL  GETCHAR      ; Zeichen vom Bildschirm lesen
67      DW    00430H       ; (Z80 JR NC,SMARKE1)
68      CPI    07FH        ; Zeichen druckbar ?
69      DW    0023BH       ; (Z80 JR C,SMARKE2)
70  SMARKE1 MVI   A,020H   ; Falls ungueltig Space ausgeben
71  SMARKE2 CALL  PRINT+VERS ; Zeichen drucken
72      POP    H           ; Zaehregister vom Stack
73      MOV    A,B         ;
74      CMP    H           ; mit maximaler Spaltenanzahl vergleichen
75      INR   H           ; Soalte = Spalte + 1
76      DW    0EB30H       ; (Z80 JR NC,SLOOP1)
77      MVI   A,00DH       ; CR & LF ausgeben
78      CALL  PRINT+VERS   ;
79      MVI   A,00AH       ;
80      CALL  PRINT+VERS   ;
81      MOV    A,C         ;
82      CMP    L           ; mit maximaler Zeilenanzahl vergleichen
83      INR   L           ; Zeile = Zeile + 1
84      DW    0D730H       ; (Z80 JR NC,LOOP2)
85      XCHG                    ; alte Cursorposition holen
86      CALL  SETCUR       ; und restaurieren
87  ESCRCOP RET            ; Ende Texthardcopy
88 ;
89  PRINT  CALL  BUSY      ; UP Drucker bereit und Ausgabe
90      DW    0FB3BH       ; (Z80 JR C,PRINT)
91      CALL  SEND         ;
92      RET                    ;
93  ENDE  END              ;

```

Listing Hardcopies



Verbesserungen von CP/M 2.2

(nur für CPC 464)

Wenn Sie öfters mit CP/M arbeiten, werden Sie sicher schon festgestellt haben, daß es hier einige Unschönheiten gibt, die einem manchmal das Leben ganz schön schwer machen. Das hier vorgestellte (CP/M-)Programm **INSTALL** verbessert einige dieser Unzulänglichkeiten und macht so das Arbeiten mit CP/M wesentlich angenehmer.

Folgende Verbesserungen werden durchgeführt:

1. Copycursor: Die CP/M-Eingaberoutine wird durch den Basic-Zeileneditor des CPC ersetzt. Damit stehen Ihnen die gewohnten Editiermöglichkeiten zur Verfügung (Copycursor, Bewegen des Hauptcursors im Eingabetext, DEL- und CLR-Taste, sowie CTRL-TAB zum Aus- und Einschalten des Einfügemodus). Mit ESC oder CTRL-C wird die Eingabe unterbrochen und ein CP/M-Warmstart durchgeführt, mit CTRL-COPY kann der Drucker zu- und abgeschaltet werden (wie zuvor mit CTRL-P). Diese Editiermöglichkeiten stehen Ihnen im CP/M-Direktmodus sowie bei den meisten CP/M-Programmen zur Verfügung.

2. Arbeiten mit verschiedenen Usern: Haben Sie schon einmal versucht, mit FILECOPY ein Programm zu kopieren, das sich nicht auf User 0 befindet? Dann werden Sie sicher gemerkt haben, daß dies nicht ganz einfach ist. Zuerst müssen Sie FILECOPY auf den gewünschten User bringen (was nur in Basic mit dem REN-Befehl möglich ist), dann das Programm kopieren und zuletzt FILECOPY wieder zurück auf User 0 schrei-

ben. Auch hier wird Abhilfe geschaffen: Nichtresidente Befehle (also COM-Files) werden in jedem Fall von User 0 geladen – egal, welchen User Sie zuvor eingestellt haben. Auf diese Weise können beliebige User mit CP/M bearbeitet werden:

```
A > user 1
A > stat usr:
A > filecopy programm.bas
A > pip b:=a:code.bin[g0] (code.bin
    wird von Drive A, User 0 nach
    Drive B, User 1 kopiert)
A > pip a:=a:test.bas[g2]
```

Die COM-Files müssen sich jetzt aber unbedingt alle auf User 0 befinden.

3. Verbesserung von PIP: Wie Sie vielleicht wissen, besitzt PIP eine Art Direktmodus (beim Aufruf ohne Parameter). Dieser soll es ermöglichen, nach dem Start von PIP die Diskette in Laufwerk A zu wechseln, um so auch Disketten bearbeiten zu können, auf denen PIP nicht extra abgespeichert wurde. Leider kommt es beim Beschreiben einer solchen Diskette stets zu einem BDOS-Error, da CP/M bei Diskettenwechsel aus Sicherheitsgründen das entsprechende Laufwerk zunächst einmal als schreibge-

schützt deklariert. PIP versäumt es nun, ein solches Laufwerk wieder zum Beschreiben freizugeben, so daß es zum Error kommt. Dieser Fehler von PIP wird ebenfalls behoben, so daß Sie nun auch im PIP-Direktmodus einwandfrei arbeiten können.

Zur Eingabe des Programms:

Geben Sie zunächst den Basic-Loader ein und speichern Sie ihn auf Diskette ab. Legen Sie nun Ihre CP/M-Arbeitsdiskette ein und starten Sie das Programm. Das File **INSTALL.COM** wird generiert und auf der Diskette abgelegt.

Starten Sie nun das CP/M und rufen Sie **INSTALL** auf. Alle Verbesserungen werden direkt auf der Diskette durchgeführt und stehen von nun an immer zur Verfügung.

Für jeden der drei genannten Punkte wird eine Erfolgsmeldung auf dem Bildschirm ausgegeben. Unterbleibt eine der Meldungen, so haben Sie irgend etwas falsch gemacht (z.B. ist die Diskette schreibgeschützt oder PIP.COM befindet sich nicht darauf).

War **INSTALL** erfolgreich, sind alle Verbesserungen sofort aktiv. Um auch andere Systemdisketten zu ändern, können Sie entweder **INSTALL** mehrmals anwenden (es macht dabei nichts, wenn sich PIP nicht auf der Diskette befindet, die Systemspuren werden trotzdem geändert), oder aber die neue CP/M-Version mit **SYSGEN** auf die anderen Disketten kopieren.

Hinweis: Es schadet nicht, wenn Sie **INSTALL** bei einer Diskette versehentlich mehrmals anwenden.

(M. Knaup/JB)

```

1 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1664]
2 REM % [378]
3 REM % VERBESSERUNG VON CP/M 2.2 % [3067]
4 REM % [378]
5 REM % September 1987 % [1417]
6 REM % [378]
7 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1664]
8 : [174]
9 REM Version: Basicloader [1543]
10 DIM n$(6):nu$=STRING$(100,0):k=0 [2151]
20 PRINT:PRINT"Bitte Systemdiskette einleg [3526]
en,"
30 PRINT"Taste druecken.":PRINT [1921]
40 WHILE INKEY$="" :WEND [1607]
50 OPENOUT"Install.Com" [2681]
60 RESTORE:FOR j=1 TO 6 [1105]
70 READ n,n$:n$(j)=LEFT$(n$+nu$,n):k=k+n [2580]
80 NEXT [350]
100 READ a$:a=VAL(a$) [884]
110 IF a>255 THEN 200 [615]
120 IF a<0 THEN PRINT#9,n$(-a):GOTO 100 [1962]
130 a=VAL("&" +a$):k=k+a [1495]
140 PRINT#9,CHR$(a); [969]
150 GOTO 100 [417]
200 CLOSEOUT [902]
210 IF k=a THEN PRINT"Install.Com generier [3042]
t."
220 IF k<>a THEN PRINT"Fehler in Datas !" [2498]
230 PRINT:END [738]
500 : [174]
501 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1239]
502 REM % Maschinenprogramm % [1202]
503 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1239]
504 : [174]
510 DATA 47,"Verbesserung von CP/M 2.2 - ( [3410]
C) 1987 by M.Knaup"
520 DATA 62,"Bitte Systemdiskette in Laufw [5731]
erk A einlegen, Taste druecken: $"
530 DATA 23,"Copycursor installiert." [3569]
540 DATA 56,"Nichtresidente Befehle werden [5658]
stets von User 0 geladen.$"
550 DATA 15,"PIP verbessert." [2406]
560 DATA 76,"" [450]
570 : [174]
580 DATA 31,00,10,11,DA,01,0E,09,CD,05,00, [3266]
0E,01,CD,05,00
590 DATA 0E,0D,CD,05,00,CD,84,01,21,78,A1, [2496]
22,D9,04,21,00
600 DATA 03,11,E1,07,01,CE,00,ED,B0,21,8C, [3329]
BE,22,A3,01,CD
610 DATA 84,01,11,4C,02,0E,09,CD,05,00,21, [2604]

```

Listing: Verbesserungen von CP/M 2.2

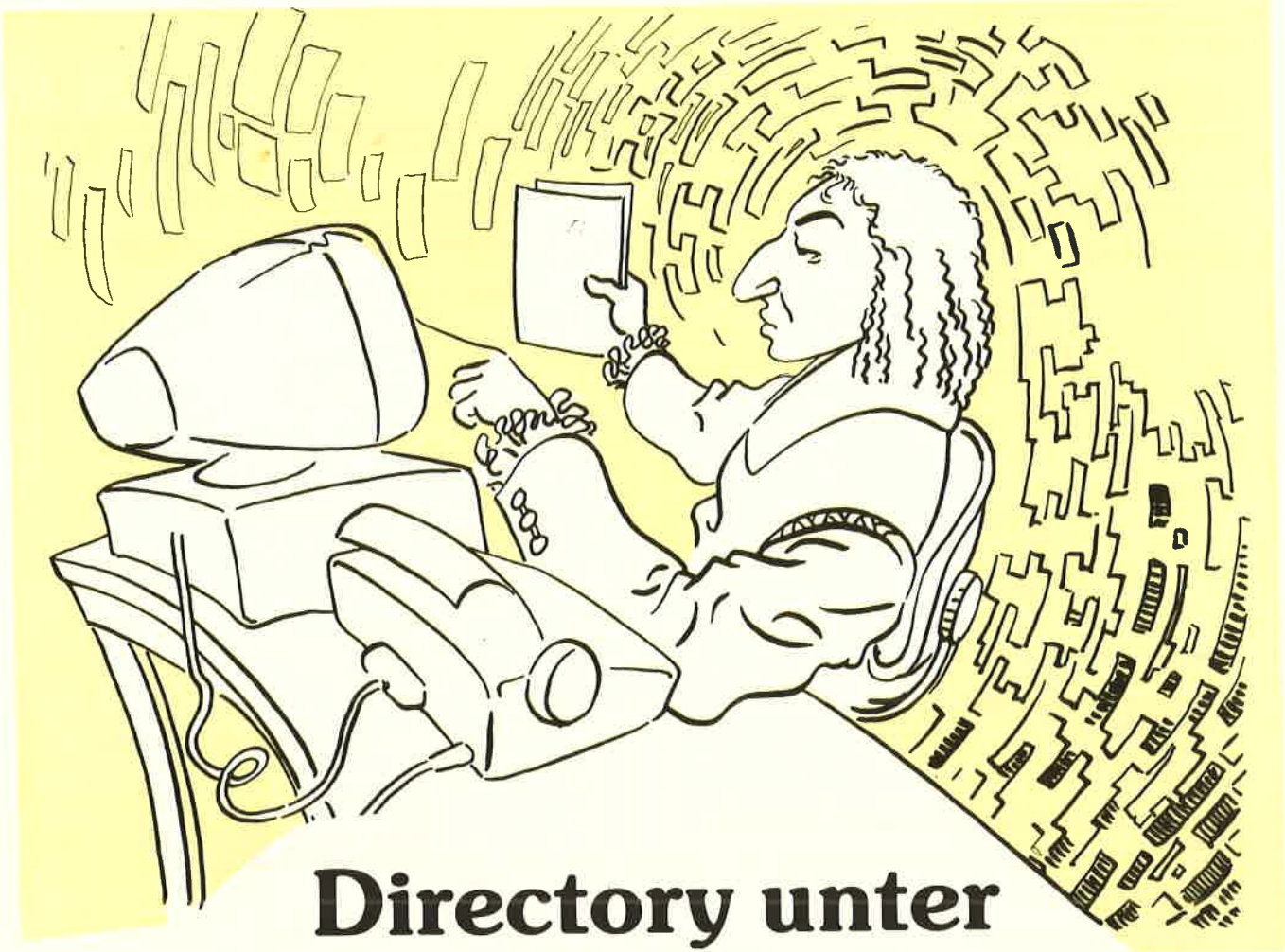
```

5C,00,11,5D,00
620 DATA 01,23,00,36,00,ED,B0,21,CE,01,11, [2947]
5C,00,01,0C,00
630 DATA ED,B0,0E,0F,CD,C7,01,FE,04,30,26, [2375]
0E,21,CD,A5,01
640 DATA 21,CF,03,11,7E,04,01,08,00,ED,B0, [2085]
0E,22,CD,A5,01
650 DATA 0E,10,CD,C7,01,FE,04,30,08,11,A0, [2604]
02,0E,09,CD,05
660 DATA 00,C3,00,00,21,00,04,11,00,01,01, [1432]
42,03,C5,D5,E5
670 DATA CD,A0,01,30,EC,E1,11,00,02,19,D1, [1809]
C1,0C,10,BE,C9
680 DATA CD,33,AD,89,BE,3E,23,06,02,11,00, [1833]
04,32,7D,00,F5
690 DATA C5,0E,1A,CD,05,00,C1,C5,CD,C7,01, [2904]
B7,20,C3,C1,F1
700 DATA 3C,11,80,04,10,E6,C9,11,5C,00,CD, [2538]
05,00,C9,01,50
710 DATA 49,50,20,20,20,20,20,43,4F,4D,0D, [2183]
0A,-1,0D,0A,0A
720 DATA -2,0D,0A,0A,-3,0D,0A,-4,0D,0A,-5, [2492]
0D,0A,24,-6,D5
730 DATA CD,4E,A1,21,42,AD,7E,36,00,21,00, [2231]
B7,E5,B7,C4,39
740 DATA A1,36,00,E1,CD,70,A1,F5,CD,4E,A1, [2834]
F1,E1,CA,00,00
750 DATA 46,23,E5,23,E5,21,00,B7,0E,00,1E, [2067]
FF,7E,23,B7,28
760 DATA 12,FE,A0,28,17,A3,1E,1F,FE,5E,28, [3337]
F0,E3,77,23,E3
770 DATA 0C,10,E7,E1,E1,71,0E,0D,CD,0C,AD, [3618]
C9,3A,0D,A2,2F
780 DATA E6,01,32,0D,A2,18,D5,EB,2A,C3,AD, [2632]
4F,06,00,ED,B0
790 DATA EB,2B,7E,23,FE,0D,C0,32,E0,B4,2B, [2006]
C9,06,0A,11,87
800 DATA A1,21,4C,B3,C5,06,03,1A,13,85,6F, [3292]
1A,4E,77,79,12
810 DATA 13,3E,50,85,6F,8C,95,67,10,F1,C1, [3087]
10,E4,C9,CD,33
820 DATA AD,75,A1,CF,98,AA,21,41,A2,7E,F5, [2826]
AF,77,E5,CD,D0
830 DATA 97,E1,D1,72,C9,00,F0,F4,F8,01,F3, [2302]
F7,FB,02,F1,F5
840 DATA F9,08,F2,F6,FA,09,E0,E0,A0,10,10, [2038]
10,10,4F,7F,7F
850 DATA 7F,3E,63,43,FC,42,FC,FC,9F,44,09, [1959]
09,E1,00,FE,03
860 DATA D0,FE,01,CC,55,08,43910 [1540]
870 [174]

```

Listing: Verbesserungen von CP/M 2.2





Directory unter TURBO PASCAL

TURBO-PASCAL Benutzer unter CP/M 80 müssen manche Nachteile gegenüber den MS-DOS Benutzern des Systems in Kauf nehmen. So sucht man beispielsweise vergeblich nach der Funktion DIR, die es unter MS-DOS ermöglicht, das Directory einer Diskette ins Programm einzublenden. Dies hat mit den Modulen DIRMOD1.INC und DIRMOD2.INC ein Ende.

Unter Ausnutzung der BDOS-Routinen 17 und 18 ist es möglich, diesem Mangel abzuweichen. Von allen bisher bekannten Varianten zur Lösung dieses Problems unterscheiden sich die vorliegenden Mo-

dule vor allem durch eine möglichst effektive Ausnutzung des ohnehin unter CP/M 80 knappen Speicherplatzes. Dies zeigt sich beispielsweise daran, daß der original FCB-Speicherplatz (Adresse 5C) benutzt und bei der Option eines sortierten Directories auf Zeigervariablen zurückgegriffen wird. Beide Module werden mittels des \$I-Compilerbefehls in das Programm eingebunden.

Falls Sie auf ein sortiertes Directory verzichten wollen, so genügt die Einbindung des Moduls DIRMOD1.INC. Der Aufruf der Prozeduren ist denkbar einfach. Als Parameter sind lediglich das Laufwerk und ein dem CP/M-Standard entspre-

chender Dateiname zu übergeben. Auch Wildcards sind erlaubt. So bewirkt z.B. der Aufruf Dir (A,??????.PAS) die Auflistung aller Pascal-Sourcefiles auf dem Laufwerk A, und SDIR(B,??????.COM) listet alle Systemfiles auf dem Laufwerk B sortiert auf. Dateien mit SYS-Attribut werden nicht mit aufgelistet, R/O-Files erhalten wie unter BASIC gewohnt einen Stern am Ende des Namens. Das Ganze funktioniert übrigens nicht nur auf sämtlichen CPC-Rechnern und dem Joyce, sondern grundsätzlich auf allen CP/M-Maschinen.

(B.Ott)

```

1 (*****
2 * DIRMOD1.INC -2421 Bytes-                (c) 1986 by B.O. *
3 *
4 * Ermoglicht die Ausgabe eines Directories innerhalb eines *
5 * Turbo-Pascal Programmes unter CP/M-80
6 * *****)
7
8 type filename = array[1..8] of char;
9 filetype = array[1..3] of char;
10
11 fcbtype = record
12     drive:byte;
13     filename:filename;
14     filetype:filetype;
15     ex:byte;
16     s1:array[1..2] of byte;
17     rc:byte;
18     s2:array[1..16] of byte;
19 end;
20
21 dirtytype = record
22     filename:filename;
23     filetype:filetype;
24     roflag:boolean;
25     sysflag:boolean;
26 end;
27
28 dmaentry = array[0..3] of fcbtype;
29
30 var fcbentry:fcbtype absolute$5C;
31 direntry:dirtytype;
32 dmaentry:dmaentry;
33 dirflag:byte;
34
35 procedure initdir(drive:char;filename:filename;filetype:filetype);
36 var hvar:byte;
37 begin
38     bdos(26,addr(dmaentry));
39     fcbentry.ex:=0;
40     fcbentry.drive:=ord(drive)-64;
41     fcbentry.filename:=filename;
42     fcbentry.filetype:=filetype;
43     dirflag:=bdos(17,addr(fcbentry));
44 end;
45
46 procedure scandir;
47 var hvar:byte;
48 begin
49     direntry.filename:=dmaentry[dirflag].filename;
50     if dmaentry[dirflag].filetype[1] > chr(127)
51     then direntry.roflag:=true
52     else direntry.roflag:=false;
53     if dmaentry[dirflag].filetype[2] > chr(127)
54     then direntry.sysflag:=true
55     else direntry.sysflag:=false;
56     for hvar:=1 to 3 do
57     begin
58         direntry.filetype[hvar]:=
59         chr(ord(dmaentry[dirflag].filetype[hvar])and 127);
60     end;
61     dirflag:=bdos(18);
62 end;
63
64 procedure dir(drive:char;filename:filename;filetype:filetype);
65 begin
66     initdir(drive,filename,filetype);
67     while dirflag<>255 do
68     begin
69         scandir;
70         if not direntry.sysflag then
71         begin
72             write(direntry.filename,' ');
73             write(direntry.filetype);
74             if direntry.roflag then write('* ')
75             else write(' ');
76         end;
77     end;
78 end;

```

```

1 (*****
2 * DIRMOD2.INC -1905 Bytes-                (c) 1986 by B.O. *
3 *
4 * Ausgabe eines sortierten Directories. DIRMOD1.INC ist zu- *
5 * vor in das Programm einzubinden
6 * *****)
7
8 ttype pointer = ^dirtytype;
9
10 dirtytype = record
11     data:dirtytype;
12     next:pointer;
13 end;
14
15 var dirbase:pointer;
16
17 {A-}procedure chaindir(var hpointer:pointer);
18 var hpointer1:pointer;
19 begin
20     if hpointer=nil then
21     begin
22         new(hpointer);
23         hpointer^.data:=direntry;
24         hpointer^.next:=nil;
25     end
26     else
27     if direntry.filename < hpointer^.data.filename then
28     begin
29         new(hpointer1);
30         hpointer1^:=hpointer^;
31         hpointer^.data:=direntry;
32         hpointer^.next:=hpointer1;
33     end
34     else
35         chaindir(hpointer^.next);
36 end;
37
38 {A-}procedure outdir(hpointer:pointer);
39 begin
40     if hpointer<>nil then
41     begin
42         direntry:=hpointer^.data;
43         if not direntry.sysflag then
44         begin
45             write(direntry.filename,' ');
46             write(direntry.filetype);
47             if direntry.roflag then write('* ')
48             else write(' ');
49         end;
50         dispose(hpointer);
51         outdir(hpointer^.next);
52     end;
53 end;
54
55 procedure sdir(drive:char;filename:filename;filetype:filetype);
56 begin
57     dirbase:=nil;
58     initdir(drive,filename,filetype);
59
60     while dirflag<>255 do
61     begin
62         scandir;
63         chaindir(dirbase);
64     end;
65     outdir(dirbase);
66 end;

```

Mit SCREEN zur Hardcopy!

Tolle Grafiken auf den Bildschirm zu zaubern ist nun wahrlich beim CPC mit Turbo Pascal und Grafikerweiterung kein Problem.

Aber Bildschirmhardcopies – Ohje! Zum einen gibt es da die ominöse Druckerschnittstelle mit 7 Bit und zum anderen jede Menge nützlicher Firmwareroutinen, die unter CP/M 2.2 einfach brach liegen und bei falscher Aktivierung den Rechner auf eine Reise ohne Wiederkehr schicken.

Dennoch, es mußte einfach möglich sein, eine Hardcopy-Routine zu installieren, die Grafiken im DIN A 4-Format zu Papier bringt und dies in sämtlichen Modes

mit allen Farben. Das Modul SCREEN2.INC tut genau dies.

Eingebunden in Ihr Pascal-Programm können Sie, wann immer Sie es sich wünschen, eine Hardcopy des Bildschirms auf den Drucker zaubern.

Sie müssen dazu lediglich an der gewünschten Stelle in Ihrem Programm die Prozedur »hardcopy« aufrufen. Die Routine stellt dann automatisch fest in welchem Bildschirmmodus Sie sich befinden und bringt in Abhängigkeit davon den Bildschirm mehr oder weniger 'farbig' zu Papier.

Falls Sie dann und wann das Bedürfnis verspüren sollten, Ihren Bildschirm auf

Diskette abzuspeichern, so hilft Ihnen dabei das Modul SCREEN1.INC.

Rufen Sie einfach die Routine »savescr« oder »loadscr« auf und übergeben Sie einen Dateinamen.

Der Bildschirminhalt landet dann unter dem vergebenen Namen und der Extension »SCR« auf der Disk.

Ach ja, vor dem Laden eines Bildschirms sollten Sie eine Mode-Anweisung setzen, sonst laufen Sie Gefahr, das Ganze etwas verschoben auf den Monitor zu bekommen.

(B. Ott)

für 464-664-6128



```

1 (*****
2 * SCREEN1.INC -919 Bytes-          (c) 1986 by B.O. *
3 *
4 * Ermöglicht speichern und laden von Bildschirmen in Turbo *
5 * Pascal
6 * *****
7
8 type scrtyp   = array[0..16383] of byte;
9   scrnametyp = string[12];
10
11 var screen:scrtyp absolute $C000;
12   scrfile:file;
13
14 procedure loadscr(scrname:scrnametyp);
15 begin
16   insert('.SCR',scrname,9);
17   assign(scrfile,scrname);
18   reset(scrfile);
19   blockread(scrfile,screen,128);
20   close(scrfile);
21 end;
22
23 procedure savescr(scrname:scrnametyp);
24 begin
25   insert('.SCR',scrname,9);
26   assign(scrfile,scrname);
27   rewrite(scrfile);
28   blockwrite(scrfile,screen,128);
29   close(scrfile);
30 end;

```

```

1 (*****
2 * SCREEN2.INC -5374 Bytes-        (c) 1986 by B.O. *
3 *
4 * Ermöglicht die Ausgabe einer Screenhardcopy in allen *
5 * modes unter CP/M 2.2
6 * *****
7
8 const initlst = #27#64;           (-Drucksteuerzeichen )
9   formlst = #12;                  ( fuer Initialisierung, )
10  line1st = #27#65#4;             ( Seitenvorschub, Zei- )
11  graf1st = #27#42#5#0#2;         ( lenvorschub und Grafik)
12
13 function testmode:byte;           (-Teste Bildschirmmodus)
14 var help1:byte;
15 begin
16   inline ($CD/$C168/             ( CALL C168
17   $BC11/                          ( DW BC11
18   $32/help1);                     ( LD HELP1,A

```

Listing SCREEN

```

19   testmode:=help1;
20 end;
21
22 function testpaper:byte;          (-Paperfarbe ermitteln )
23 var help1:byte;
24 begin
25   inline ($CD/$C168/             ( CALL C168
26   $BBE7/                          ( DW BBE7
27   $32/help1);                     ( LD HELP1,A
28   testpaper:=help1;
29 end;
30
31 function testpixel(x,y:integer):byte; (-Pixelpen ermitteln )
32 var help1:byte;
33 begin
34   inline($ED/$5B/x/              ( LD DE,X
35   $2A/y/                          ( LD HL,Y
36   $CD/$C168/                     ( CALL C168
37   $BF0/                          ( DW BF0
38   $32/help1);                   ( LD HELP1,A
39   testpixel:=help1;
40 end;
41
42 function testink(pen:byte):byte; (-Ink ermitteln )
43 var help1:byte;
44 begin
45   inline($3A/pen/                ( LD A,PEN
46   $CD/$C168/                     ( CALL C168
47   $BC35/                          ( DW BC35
48   $7B/                          ( LD A,B
49   $32/help1);                   ( LD HELP1,A
50   testink:=help1;
51 end;
52
53 procedure scrhc0(var x,y:integer);
54 const color:array[0..26,0..11] of byte = (-Druckmatrix fuer
55   (15,15), (14,14), (07,07), (07,14), (14,07), ( verschiedene Farben
56   (15,06), (06,15), (03,15), (03,15), (15,12), ( in Mode 0
57   (12,15), (06,06), (10,10), (05,05), (15,00),
58   (00,15), (03,12), (12,03), (10,05), (05,10),
59   (06,02), (02,06), (06,04), (04,06), (02,04),
60   (04,02), (00,00));
61 var pixel,transbyte1,transbyte2:byte;
62 begin
63   transbyte1:=0;transbyte2:=0;
64   pixel:=testink(testpixel(x,y));
65   transbyte1:=color[pixel,0];
66   transbyte2:=color[pixel,1];
67   write(1st,chr(transbyte1),chr(transbyte2));
68 end;
69
70 procedure scrhc1(x:integer;var y:integer);
71 const color:array[0..26,0..11] of byte =
72   ((3,3), (3,3), (3,1), (3,1), (1,3), (-Druckmatrix fuer
73   (1,3), (3,2), (3,2), (2,3), (2,3), ( verschiedene Farben
74   (3,0), (3,0), (0,3), (0,3), (1,1), ( in Mode 1
75   (1,1), (2,2), (2,1), (2,1), (1,2),
76   (1,2), (2,0), (2,0), (0,2), (0,1),
77   (1,0), (0,0));

```

Listing SCREEN

```

78 var c1,pixel,transbyte1,transbyte2:byte;
79 begin
80   transbyte1:=0;transbyte2:=0;
81   for c1:=0 to 1 do
82     begin
83       pixel:=testink(testpixel(x,y));
84       transbyte1:=transbyte1+color[pixel,0] shl (2-c1*2);
85       transbyte2:=transbyte2+color[pixel,1] shl (2-c1*2);
86       x:=x+2;
87     end;
88   write(lst,chr(transbyte1),chr(transbyte2));
89 end;
90
91 procedure scrhc2(x:integer;var y:integer;var paper:byte);
92 var c1,transbyte1:byte;
93 begin
94   transbyte1:=0;
95   for c1:= 0 to 3 do
96     begin
97       if testpixel(x,y)<>paper then
98         transbyte1:=transbyte1+(1 shl (3-c1));
99       x:=succ(x);
100    end;
101    write(lst,chr(transbyte1),chr(transbyte1));
102  end;
103
Listing SCREEN

```

```

104 procedure hardcopy;
105 var x,y:integer;
106   paper,mode:byte;
107 begin
108   inline(%CD/%C168/$BBBA);           (-Grafikmodus initiali-)
109   inline(%CD/%C168/$BB8D);           (-sieren und Cursor dar-)
110   mode:=testmode;                     (-stellung verbieten -)
111   paper:=testpaper;
112   write(lst,initlst,linelst);
113   for x:=14 downto 0 do writeln(lst);
114   repeat write(lst,graflst);
115     for y:= 87 downto 0 do write(lst,chr(0));
116     repeat case mode of 0:scrhc0(x,y);
117               1:scrhc1(x,y);
118               2:scrhc2(x,y,paper);
119           end;
120     y:=y+2;
121     until y=400;
122     for y:= 1 to 24 do write(lst,chr(0));
123     writeln(lst);
124     x:=x+4;
125   until x=640;
126   inline(%CD/%C168/$BBBA);           (-Cursordarstellung er-)
127   writeln(lst,formlst,initlst);      (- lauben -)
128 end;
Listing SCREEN

```

Grafik & Co. unter Turbo Pascal 3.0

»Natürlich bin ich besser als du«, sagte Turbo Pascal zum Basic, »ich bin schließlich strukturiert und außerdem viel schneller!« — »Kann schon sein, aber hast du denn auch Grafik und Sound, so wie ich?« Hier mußte Turbo Pascal allerdings passen — bisher!

Wenn Sie nämlich die untenstehenden Prozeduren und Funktionen abtippen, kann es Turbo Pascal auch in dieser Beziehung mit dem Basic aufnehmen. Alle Routinen sind kurz erläutert und sinngemäß zusammengefaßt. Sie können einzelne Routinen herausnehmen oder blockweise in Include-Files ablegen.

Befehlsweiterung — wie geht das?

Dreh- und Angelpunkt solcher neuen Befehle ist die **INLINE**-Anweisung des Tur-

bo Pascal, mit der man bekanntlich kurze Maschinenprogramme in Turbo-Pascal integrieren kann. Variablenamen werden als Adressen interpretiert, so daß z.B.

```
INLINE($2A/var/...
```

den Inhalt der Integervariable var im Register HL ergibt. Sprünge ins Betriebssystem des CPC sind leider nicht ganz so einfach.

So kann man nicht z.B. mit

```
INLINE(%CD/%06/%BB);
```

auf einen Tastendruck warten. Grund dafür ist die besondere Speicheraufteilung unter CP/M+: Sämtliche User-Programme (also auch Ihr Pascal-Programm) stehen nämlich in der zweiten 64K-Speicherbank, die Betriebssystem-Vektoren aber in der ersten! Um die Betriebssystem-Vektoren zu erreichen, müssen Sie einen CALL ähnlich

den RST-Befehlen benutzen. Die Syntax lautet:

```
CALL FD10h / DW sprungadresse
```

Dieser CALL hat zur Folge, daß der Bereich 0 bis BFFFh des RAMs (also alles außer den oberen 16K) für die Dauer der Routine auf die andere 64K-Bank umgeschaltet wird. Auch der Bildschirmspeicher ist hier abgelegt (4000h bis 7FFFh), so daß ein direkter Bildschirmzugriff in Turbo Pascal nicht ganz einfach zu programmieren ist.

Der CALL hat übrigens mit dem Turbo Pascal nichts zu tun; er ist Teil des CP/M+ und steht auch für CP/M-Programme zur Verfügung. Der oben als Beispiel verwendete Firmware-CALL sieht also richtig so aus:

```
INLINE(%CD/%10/%FD/%06/%BB);
```

Damit wäre das Wichtigste erklärt; weitere Tips finden Sie vielleicht in den abgedruckten Routinen.

(T. Kochmann/JB)

Für 6128

```

-----
Befehlsweiterung zum Screen-Manager
-----
PROCEDURE scr_init;
(Screen-Manager initialisieren)
BEGIN
  INLINE(%CD/%10/%FD/%02/%BC);
END;
Listing Grafik & Co.

```

```

PROCEDURE scr_mgde(mode:BYTE);
(Bildschirmmodus einstellen; Basic: MODE)
BEGIN
  GOTOXY(1,1);
  INLINE(%CD/%10/%FD/%81/%BB/%3A/ mode /
    %CD/%10/%FD/%0E/%BC/%CD/%10/%FD/%81/%BB);
END;

PROCEDURE scr_ink(stift,farbe1,farbe2:BYTE);
(Bildschirmfarben setzen; Basic: INK)
BEGIN
  INLINE(%3A/farbe1/%47/%3A/farbe2/%4F/%3A/stift/%CD/%10/%FD/%32/%BC);
END;

PROCEDURE scr_border(farbe1,farbe2:BYTE);
(Randfarben setzen; Basic: BORDER)
BEGIN

```

Listing Grafik & Co.


```

  INLINE($3A/Farbe1/$47/$3A/Farbe2/$4F/$CD/$10/$FD/$3B/$BC);
END;

PROCEDURE scr_speed(wieder1,wieder2:BYTE);
  (Blinkperioden setzen; Basic: SPEED INK)
BEGIN
  INLINE($3A/wieder1/$67/$3A/wieder2/$6F/$CD/$10/$FD/$3E/$BC);
END;

PROCEDURE scr_flyback;
  (Monitor-Strahlruecklauf abwarten; Basic: FRAME)
BEGIN
  INLINE($CD/$10/$FD/$19/$BD);
END;

PROCEDURE scr_scroll(richtg:BYTE);
  (ganzen Bildschirm scrollen, richtg=0/1: ab/auf)
BEGIN
  INLINE($3A/richtg/$47/$97/$CD/$10/$FD/$4D/$BC);
END;

PROCEDURE scr_invert(anzahl:BYTE);
  (ab Cursorposition anzahl Zeichen invertieren)
BEGIN
  INLINE($CD/$10/$FD/$78/$BB/$E5/$3A/anzahl/$47/$
    $CD/$10/$FD/$8A/$BB/$3E/$09/$CD/$10/$FD/$5A/$BB/$
    $10/$F2/$E1/$CD/$10/$FD/$75/$BB);
END;

```

```

-----
Befehls-Erweiterung zum Screen-Manager II
-----

```

```

TYPE filename=STRING(12);

PROCEDURE scr_load(str:filename);
  (Bildschirm-Datei laden)
VAR pic:FILE;buffer:ARRAY[0..127] OF BYTE; i,adr:INTEGER;
BEGIN
  ASSIGN(pic,str);
  RESET(pic);
  BLOCKREAD(pic,buffer,1);
  INLINE($CD/$10/$FD/$B1/$BB);
  adr:=$4000;
  FOR i:=1 TO 128 DO
  BEGIN
    BLOCKREAD(pic,buffer,1);
    INLINE($21/buffer/$ED/$5B/adr/$01/$80/$00/$
      $CD/$10/$FD/$1B/$B9/$ED/$53/adr);
  END;
  INLINE($CD/$10/$FD/$8A/$BB);
  CLOSE(pic);
END;

PROCEDURE scr_save(str:filename);
  (Bildschirm als Bin-File ablegen)
VAR pic:FILE;buffer:ARRAY[0..127] OF BYTE; i,adr:INTEGER;
BEGIN
  ASSIGN(pic,str);
  REWRITE(pic);
  FOR i:=$00 TO $42 DO buffer[i]:=0;
  buffer[$12]:=2;
  buffer[$16]:=$C0;
  buffer[$19]:=$40;
  buffer[$11]:=$40;
  buffer[$13]:=$42;
  buffer[$14]:=$01;
  BLOCKWRITE(pic,buffer,1);
  adr:=$4000;
  FOR i:=1 TO 128 DO
  BEGIN
    INLINE($2A/adr/$11/buffer/$01/$80/$00/$
      $CD/$10/$FD/$1B/$B9/$22/adr);
    BLOCKWRITE(pic,buffer,1);
  END;
  CLOSE(pic);
END;

PROCEDURE scr_hardcopy;
  (nur Fuer Mode 2)
VAR buffer:ARRAY[1..640] OF BYTE; i,j:INTEGER;
BEGIN
  WRITELN(LST,$1B,'M',$16); (linken Rand auf 16)
  WRITELN(LST,$1B,'1'); (Zellenabstand 7/72 inch)
  INLINE($CD/$10/$FD/$0B/$BC/$84/$67/$22/$00/$FF);
  MEM($FF02):=$7F;
  FOR i:=1 TO 28 DO BEGIN
    IF i=28 THEN MEM($FF02):=$78;
    WRITE(LST,$1B,'L',$7F,$802); (Grafik mit 120 Pkt./inch)
    INLINE($2A/$00/$FF/$97/$32/$01/$FF/$11/buffer/$06/$50/$
      $C5/$E5/$D5/$11/$03/$FF/$D5/$06/$07/$
      $CD/$10/$FD/$20/$00/$12/$13/$CD/$10/$FD/$26/$BC/$
      $10/$F2/$3A/$01/$FF/$B7/$20/$03/$22/$00/$FF/$
      $E1/$D1/$0E/$0B/$E5/$06/$07/$0B/$26/$23/$17/$10/$FA/$
      $E1/$2B/$A6/$23/$12/$13/$0D/$20/$EE/$
      $E1/$CD/$10/$FD/$20/$BC/$C1/$10/$C1);
    FOR j:=1 TO 638 DO WRITE(LST,CHR(buffer[j]));
  WRITELN(LST);
  END;
END;

```

```

-----
Befehls-Erweiterung zum Grafik-Bildschirm
-----

```

```

PROCEDURE gra_init;
  (Grafik-Bildschirm initialisieren)
BEGIN
  INLINE($CD/$10/$FD/$8A/$BB);
END;

PROCEDURE gra_clear;
  (Grafik-Window loeschen; Basic: CLG)
BEGIN
  INLINE($CD/$10/$FD/$B4/$BB/$CD/$10/$FD/$D8/$BB);

```

```

END;

PROCEDURE gra_pen(Farbe:BYTE);
  (Grafik-Pen setzen)
BEGIN
  INLINE($3A/Farbe/$CD/$10/$FD/$DE/$BB);
END;

PROCEDURE gra_paper(Farbe:BYTE);
  (Grafik-Paper setzen)
BEGIN
  INLINE($3A/Farbe/$CD/$10/$FD/$E4/$BB);
END;

PROCEDURE gra_mode(mode:BYTE);
  (Grafik-Plotmodus setzen; 0: Force, 1: xor, 2: and, 3: or)
BEGIN
  INLINE($3A/Mode/$CD/$10/$FD/$59/$BC);
END;

PROCEDURE gra_window(links,rechts,oben,unten:INTEGER);
  (Grenzen des Grafik-Window setzen; Basic: ORIGIN)
BEGIN
  INLINE($ED/$5B/links/$2A/rechts/$CD/$10/$FD/$CF/$BB/$
    $ED/$5B/oben/$2A/unten/$CD/$10/$FD/$D2/$BB);
END;

PROCEDURE gra_origin(xorg,yorg:INTEGER);
  (Grafik-Origin setzen; Basic: ORIGIN)
BEGIN
  INLINE($ED/$5B/xorg/$2A/yorg/$CD/$10/$FD/$C9/$BB);
END;

PROCEDURE gra_move(x,y:INTEGER);
  (Grafik-Cursor setzen (absolut); Basic: MOVE)
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$CO/$BB);
END;

PROCEDURE gra_mover(x,y:INTEGER);
  (Grafik-Cursor setzen (relativ); Basic: MOVER)
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$C3/$BB);
END;

PROCEDURE gra_plot(x,y:INTEGER);
  (Grafik-Punkt zeichnen (absolut); Basic: PLOT)
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$EA/$BB);
END;

PROCEDURE gra_plotr(x,y:INTEGER);
  (Grafik-Punkt zeichnen (relativ); Basic: PLOTR)
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$ED/$BB);
END;

PROCEDURE gra_draw(x,y:INTEGER);
  (Linie zeichnen (absolut); Basic: DRAW)
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$F6/$BB);
END;

PROCEDURE gra_drawr(x,y:INTEGER);
  (Linie zeichnen (relativ); Basic: DRAWR)
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$F9/$BB);
END;

FUNCTION gra_test(x,y:INTEGER):BYTE;
  (Farbe des Grafik-Punkts (x,y) (absolut); Basic: TEST)
VAR hilf:BYTE;
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$F0/$BB/$32/hilf);
  gra_test:=hilf;
END;

FUNCTION gra_testr(x,y:INTEGER):BYTE;
  (Farbe des Grafik-Punkts (x,y) (relativ); Basic: TESTR)
VAR hilf:BYTE;
BEGIN
  INLINE($ED/$5B/x/$2A/y/$CD/$10/$FD/$F3/$BB/$32/hilf);
  gra_testr:=hilf;
END;

FUNCTION gra_xpos:INTEGER;
  (aktuelle x-Position des Grafik-Cursors; Basic: XPOS)
VAR hilf:INTEGER;
BEGIN
  INLINE($CD/$10/$FD/$C6/$BB/$ED/$53/hilf);
  gra_xpos:=hilf;
END;

FUNCTION gra_ypos:INTEGER;
  (aktuelle y-Position des Grafik-Cursors; Basic: YPOS)
VAR hilf:INTEGER;
BEGIN
  INLINE($CD/$10/$FD/$C6/$BB/$22/hilf);
  gra_ypos:=hilf;
END;

PROCEDURE gra_write(zeichen:CHAR);
  (Zeichen auf Grafik-Bildschirm schreiben)
BEGIN
  INLINE($3A/zeichen/$CD/$10/$FD/$FC/$BB);
END;

VAR patch:INTEGER;
  (Adresse der normalen ConOut-Prozedur)

PROCEDURE gra_tagon;
  (Text auf Grafik-Bildschirm; Basic: TAG)
BEGIN
  patch:=conoutptr;
  conoutptr:=ADDR(gra_write);
END;

PROCEDURE gra_tegoff;
  (Text wieder auf Text-Bildschirm; Basic: TAGOFF)
BEGIN
  conoutptr:=patch;
END;

```

```
(-----)
  Befehlsweiterung zum Text-Bildschirm
(-----)
```

```
PROCEDURE txt_init;
(Text-Bildschirm initialisieren)
BEGIN
  INLINE($CD/$10/$FD/$4E/$8B);
  GOTOXY(1,1);
END;

PROCEDURE txt_stream(stream:BYTE);
(Text-Window 0..7 weehlen)
VAR spalte,zeile:BYTE;
BEGIN
  INLINE($CD/$10/$FD/$84/$8B/$3A/stream/$CD/$10/$FD/$84/$8B/
$CD/$10/$FD/$7B/$8B/$7C/$32/spalte/$7D/$32/zeile);
  GOTOXY(spalte,zeile);
END;

(Die folgenden Prozeduren und Funktionen
beziehen sich auf das aktuelle Text-Window)

PROCEDURE txt_window(links,rechts,oben,unten:BYTE);
(Grenzen des Text-Window setzen; Basic: WINDOW)
BEGIN
  INLINE($3A/links/$3D/$67/$3A/rechts/$3D/$57/
$3A/oben/$3D/$6F/$3A/unten/$3D/$5F/$CD/$10/$FD/$66/$8B);
  GOTOXY(1,1);
END;

PROCEDURE txt_locate(spalte,zeile:BYTE);
(Text-Cursor setzen; Basic: LOCATE)
BEGIN
  INLINE($3A/spalte/$67/$3A/zeile/$6F/$CD/$10/$FD/$75/$8B);
  GOTOXY(spalte,zeile);
END;

FUNCTION txt_spalte:BYTE;
(aktuelle Text-Cursorspalte; Basic: POS)
VAR hilf:BYTE;
BEGIN
  INLINE($CD/$10/$FD/$7B/$8B/$7C/$32/hilf);
  txt_spalte:=hilf;
END;

FUNCTION txt_zeile:BYTE;
(aktuelle Text-Cursorzeile; Basic: UPOS)
VAR hilf:BYTE;
BEGIN
  INLINE($CD/$10/$FD/$7B/$8B/$7D/$32/hilf);
  txt_zeile:=hilf;
END;

PROCEDURE txt_cursoroff;
(Text-Cursor ein-/ausschalten)
BEGIN
  INLINE($CD/$10/$FD/$8A/$8B);
END;

PROCEDURE txt_clear;
(Text-Window loeschen; Basic: CLS)
BEGIN
  INLINE($CD/$10/$FD/$6C/$8B);
  GOTOXY(1,1);
END;

PROCEDURE txt_pen(farbe:BYTE);
(Text-Pen setzen; Basic: PEN)
BEGIN
  INLINE($3A/farbe/$CD/$10/$FD/$80/$8B);
END;

PROCEDURE txt_paper(farbe:BYTE);
(Text-Paper setzen; Basic: PAPER)
BEGIN
  INLINE($3A/farbe/$CD/$10/$FD/$86/$8B);
END;

PROCEDURE txt_inverse;
(Text-Pen und -Paper vertauschen)
BEGIN
  INLINE($CD/$10/$FD/$9C/$8B);
END;

PROCEDURE txt_transon;
(Transparent-Modus einschalten)
BEGIN
  INLINE($3E/$FF/$CD/$10/$FD/$9F/$8B);
END;

PROCEDURE txt_transoff;
(Transparent-Modus ausschalten)
BEGIN
  INLINE($87/$CD/$10/$FD/$9F/$8B);
END;

PROCEDURE txt_control(zeichen:BYTE);
(CPC-Steuerzeichen 0..31 aufrufen)
BEGIN
  INLINE($3A/zeichen/$CD/$10/$FD/$5A/$8B);
END;

TYPE zeichenmatrix=ARRAY[1..8] OF BYTE;
(Zeichenmatrix fuer Bildschirmzeichen)

PROCEDURE txt_symbol(zeichen:BYTE;bytes:zeichenmatrix);
(Zeichen neu definieren; Basic: SYMBOL)
BEGIN
  INLINE($3A/zeichen/$CD/$10/$FD/$A5/$8B/$EB/
$21/bytes/$01/$0B/$00/$CD/$10/$FD/$1B/$8B);
END;
```

```
(-----)
  Befehlsweiterung zum Keyboard-Manager
(-----)
```

```
PROCEDURE km_init;
(Keyboard-Manager initialisieren)
BEGIN
  INLINE($CD/$10/$FD/$00/$8B);
END;
```

Listing Grafik & Co.

```
FUNCTION km_testkey(teste:BYTE):BOOLEAN;
(Test, ob Taste gedrueckt; Basic: INKEY)
VAR test:BOOLEAN;
BEGIN
  INLINE($3A/teste/$CD/$10/$FD/$1E/$8B/
$2B/$02/$3E/$FF/$32/test);
  km_testkey:=test;
END;

FUNCTION km_getchar:CHAR;
(Zeichen holen, falls vorhanden; 0: keines vorhanden)
VAR zeichen:CHAR;
BEGIN
  INLINE($CD/$10/$FD/$1B/$8B/$38/$01/$97/$32/zeichen);
  km_getchar:=zeichen;
END;

FUNCTION km_joy(stick:BYTE):BYTE;
(Joystick abfragen; Basic: JOY)
VAR hilf:INTEGER;
BEGIN
  INLINE($CD/$10/$FD/$24/$8B/$22/hilf);
  CASE stick OF
    0: km_joy:=hi(hilf);
    1: km_joy:=lo(hilf);
  END;
END;

PROCEDURE km_setkey(teste,wieder,ch1,ch2,ch3:BYTE);
(Taste neu belegen, wieder=0/1: Wiederholung aus/ein;
Basic: KEY DEF)
BEGIN
  INLINE($3A/wieder/$2F/$47/$3A/teste/$F5/$CD/$10/$FD/$39/$8B/
$3A/ch1/$47/$F1/$F5/$CD/$10/$FD/$27/$8B/
$3A/ch2/$47/$F1/$F5/$CD/$10/$FD/$2D/$8B/
$3A/ch3/$47/$F1/$CD/$10/$FD/$33/$8B);
END;

TYPE expatring=STRING[16];
(Laenge kann willkuerlich gesendert werden)

PROCEDURE km_expand(teste:BYTE;str:expatring);
(Expansion-String setzen; Basic: KEY)
BEGIN
  INLINE($21/str/$7E/$47/$87/$4F/$11/$00/$FF/$05/
$23/$7E/$12/$13/$12/$13/$10/$F8/
$E1/$3A/teste/$47/$CD/$10/$FD/$0F/$8B);
END;

PROCEDURE km_speed(start,wieder:BYTE);
(Tastaturverzoegerungen setzen; Basic: SPEED KEY)
BEGIN
  INLINE($3A/start/$67/$3A/wieder/$6F/$CD/$10/$FD/$3F/$8B);
END;
```

```
(-----)
  Befehlsweiterung zum Sound-Manager
(-----)
```

```
PROCEDURE snd_init;
(Sound-Manager initialisieren)
BEGIN
  INLINE($CD/$10/$FD/$A7/$8C);
END;

PROCEDURE snd_sound(status:BYTE;ton,dauer:INTEGER;
lautst,env,ent,tausch:BYTE);
(Tonprogramm in Warteschlange setzen; Basic: SOUND)
VAR param:ARRAY[1..9] OF BYTE;
BEGIN
  param[1]:=status;
  param[2]:=env;
  param[3]:=ent;
  MOVE(ton,param[4],2);
  param[6]:=tausch;
  param[7]:=lautst;
  MOVE(dauer,param[8],2);
  INLINE($21/param/$CD/$10/$FD/$AA/$8C);
END;

TYPE progr=ARRAY[0..15] OF BYTE;
(Programm fuer die Huellkurven; progr[0] muss
die Anzahl der Abschnitte (je 3 Bytes) enthalten!)

PROCEDURE snd_env(nc:BYTE;bytes:progr);
(Lautstaerkehuellkurve setzen; Basic: ENV)
BEGIN
  INLINE($3A/nc/$21/bytes/$CD/$10/$FD/$8C/$8C);
END;

PROCEDURE snd_ent(nc:BYTE;bytes:progr);
(Tonhuellkurve setzen; Basic: ENT)
BEGIN
  INLINE($3A/nc/$21/bytes/$CD/$10/$FD/$8F/$8C);
END;

PROCEDURE snd_release(maske:BYTE);
(Haltepunkte aufheben; Bit 0/1/2: Kanal A/B/C; Basic: RELEASE)
BEGIN
  INLINE($3A/maske/$CD/$10/$FD/$83/$8C);
END;

PROCEDURE snd_hold;
(Sound sofort anhalten)
BEGIN
  INLINE($CD/$10/$FD/$8E/$8C);
END;

PROCEDURE snd_continue;
(Sound wieder freigeben)
BEGIN
  INLINE($CD/$10/$FD/$89/$8C);
END;

FUNCTION snd_check(maske:BYTE):BYTE;
(Kanalstatus abfragen; maske=1/2/4: Kanal A/B/C; Basic: SQ)
VAR hilf:BYTE;
BEGIN
  INLINE($3A/maske/$CD/$10/$FD/$AD/$8C/$32/hilf);
  snd_check:=hilf;
END;
```

Listing Grafik & Co.

Tips & Tricks

226,102,102,102,102,102,60,0,1232
 1470 DATA 66,102,102,102,102,60,8,0,[3164]
 132,198,198,214,254,238,68,0,1844
 1480 DATA 198,36,56,56,108,198,131,0,[2564]
 74,102,102,60,24,24,40,0,1209
 1490 DATA 126,134,12,24,48,98,252,0,[2419]
 60,48,48,48,48,48,60,0,1054
 1500 DATA 192,96,48,24,12,6,2,0,[2483]
 60,12,12,12,12,12,60,0,560
 1510 DATA 24,60,126,24,24,24,24,0,[2675]
 0,0,0,0,0,0,0,255,561
 1520 DATA 48,24,12,0,0,0,0,0,[2874]
 0,128,120,20,116,212,116,3,799
 1530 DATA 160,96,124,106,106,106,188,0,[2160]
 0,0,60,86,80,86,60,0,1258
 1540 DATA 20,12,124,172,172,172,116,2,[2832]
 0,0,60,102,94,80,60,0,1186
 1550 DATA 26,52,48,112,56,48,48,32,[2781]
 0,0,62,86,86,62,134,124,976
 1560 DATA 160,96,108,116,102,102,196,8,[1848]
 24,0,56,24,24,28,8,0,1052
 1570 DATA 6,0,10,6,6,6,102,60,[2961]
 160,96,102,108,120,108,166,1,1057
 1580 DATA 40,24,24,24,24,26,12,0,[2408]
 0,0,212,106,106,106,34,4,742
 1590 DATA 0,0,220,118,102,102,36,8,[2614]
 0,0,60,86,86,86,60,0,964
 1600 DATA 0,0,220,106,106,124,96,96,[3053]
 0,0,118,172,172,124,12,12,1358
 1610 DATA 0,0,220,114,96,96,160,0,[3067]
 0,2,60,104,60,38,124,128,1202
 1620 DATA 40,48,124,48,48,50,28,0,[2600]
 0,0,226,102,102,102,62,0,980
 1630 DATA 0,0,34,102,102,60,24,0,[2817]
 0,0,138,214,214,254,108,0,1250
 1640 DATA 0,2,198,44,56,108,197,2,[3569]
 0,0,106,106,106,62,130,124,1241
 1650 DATA 0,0,62,76,24,50,124,0,[3084]
 14,24,24,112,24,24,14,0,572
 1660 DATA 24,24,24,24,24,24,24,0,[2474]
 112,24,24,14,24,24,112,0,502
 1670 DATA 118,220,0,0,0,0,0,0,[2725]
 60,102,96,248,96,102,254,0,1296

1190 SAVE"GOTHIC.FNT",B,40000,768 [1603]
 1200 DATA 0,0,0,0,0,0,0,0,[2014]
 16,16,16,16,16,0,16,0,96
 1210 DATA 36,36,36,0,0,0,0,0,[1754]
 0,36,126,36,36,126,36,0,504
 1220 DATA 16,60,80,56,20,120,16,0,[3343]
 0,98,100,8,16,38,70,0,698
 1230 DATA 56,68,68,60,74,68,58,0,[1640]
 8,8,16,0,0,0,0,0,484
 1240 DATA 4,8,16,16,16,8,4,0,[2545]
 32,16,8,8,8,16,32,0,192
 1250 DATA 0,0,36,24,126,24,36,0,[2399]
 0,0,16,16,124,16,16,0,434
 1260 DATA 0,0,0,0,0,0,16,16,32,[2283]
 0,0,0,124,0,0,0,0,188
 1270 DATA 0,0,0,0,0,0,16,16,0,[2514]
 0,2,4,8,16,32,64,0,158
 1280 DATA 24,36,66,66,66,36,24,0,[1960]
 16,48,16,16,16,16,0,462
 1290 DATA 56,68,4,8,16,32,124,0,[1734]
 56,68,4,8,4,68,56,0,572
 1300 DATA 16,16,32,32,72,124,8,0,[2279]
 120,64,64,120,4,68,56,0,796
 1310 DATA 56,68,64,120,68,68,56,0,[2150]
 124,4,8,16,32,32,0,748
 1320 DATA 56,68,68,56,68,68,56,0,[2745]
 56,68,68,60,4,68,56,0,820
 1330 DATA 0,0,16,16,0,16,16,0,[2659]
 0,0,16,16,0,16,16,32,160
 1340 DATA 4,8,16,32,16,8,4,0,[1635]
 0,0,124,0,0,124,0,0,336
 1350 DATA 32,16,8,4,8,16,32,0,[2247]
 60,66,2,12,16,0,16,0,288
 1360 DATA 56,68,164,152,128,68,56,0,[2799]
 16,16,40,68,124,130,130,0,1216
 1370 DATA 252,130,130,252,130,130,252,0,[3469]
 60,66,128,128,128,66,60,0,1912
 1380 DATA 248,132,130,130,130,132,248,0,[3627]
 254,128,128,248,128,254,0,2418
 1390 DATA 254,128,128,248,128,128,128,0,[2404]
 60,66,128,142,130,66,60,0,1794
 1400 DATA 130,130,130,254,130,130,130,0,[2673]
 16,16,16,16,16,16,16,0,1146
 1410 DATA 2,2,2,2,130,68,56,0,[2177]
 132,136,144,224,144,136,132,0,1310
 1420 DATA 128,128,128,128,128,254,0,[3059]
 130,198,170,146,130,130,130,0,2056

1430 DATA 130,194,162,146,138,134,130,0,[2879]
 56,68,130,130,130,68,56,0,1672
 1440 DATA 252,130,130,130,252,128,128,0,[2735]
 56,68,130,130,138,68,58,0,1798
 1450 DATA 252,130,130,130,252,132,130,0,[3238]
 60,66,64,60,2,130,124,0,1662
 1460 DATA 254,16,16,16,16,16,0,[2201]
 130,130,130,130,130,68,56,0,1124
 1470 DATA 130,130,130,68,68,40,16,0,[2726]
 130,130,130,146,146,84,40,0,1388
 1480 DATA 130,68,40,16,40,68,130,0,[2362]
 130,130,68,40,16,16,16,0,908
 1490 DATA 254,4,8,16,32,64,254,0,[2093]
 28,16,16,16,16,16,28,0,768
 1500 DATA 0,64,32,16,8,4,2,0,[2208]
 56,8,8,8,8,8,56,0,278
 1510 DATA 16,56,84,16,16,16,16,0,[2913]
 0,0,0,0,0,0,0,255,475
 1520 DATA 32,16,8,0,0,0,0,0,[2529]
 0,0,58,70,66,70,58,0,378
 1530 DATA 64,64,92,98,66,98,92,0,[2079]
 0,0,60,66,64,66,60,0,890
 1540 DATA 2,2,58,70,66,70,58,0,[1909]
 0,0,56,68,124,64,60,0,698
 1550 DATA 28,34,32,120,32,32,32,0,[2722]
 0,0,58,70,70,58,2,60,628
 1560 DATA 64,64,92,98,66,66,66,0,[1910]
 16,0,16,16,16,16,16,0,612
 1570 DATA 2,0,2,2,2,2,66,36,24,[2246]
 64,64,68,88,96,88,68,0,670
 1580 DATA 16,16,16,16,16,16,8,0,[953]
 0,0,36,90,66,66,66,0,428
 1590 DATA 0,0,92,98,66,66,66,0,[2179]
 0,0,60,66,66,66,60,0,706
 1600 DATA 0,0,92,98,66,66,98,64,[2049]
 0,0,58,70,66,70,58,2,834
 1610 DATA 0,0,92,98,64,64,64,0,[2072]
 0,0,60,64,60,2,60,0,628
 1620 DATA 32,32,120,32,32,34,28,0,[2754]
 0,0,66,66,66,66,60,0,634
 1630 DATA 0,0,68,68,40,16,16,0,[1816]
 0,0,68,68,84,84,56,0,568
 1640 DATA 0,0,68,40,16,40,68,0,[1911]
 0,0,66,66,34,30,66,60,554
 1650 DATA 0,0,124,8,16,32,124,0,[2174]
 12,16,16,32,16,16,12,0,424
 1660 DATA 24,24,24,24,24,24,24,24,[2027]
 16,8,8,4,8,8,16,0,260
 1670 DATA 4,42,16,0,0,0,0,0,[2041]
 24,36,32,112,32,36,120,0,454

1190 SAVE"HI-INDEX.FNT",B,40000,768 [1417]
 1200 DATA 0,0,0,0,0,0,0,0,[2051]
 16,16,16,0,16,0,0,0,64
 1210 DATA 40,0,0,0,0,0,0,0,[2114]
 24,60,24,60,24,0,0,0,232
 1220 DATA 16,56,48,24,56,16,0,0,[2460]
 0,36,8,16,36,0,0,0,312
 1230 DATA 16,40,20,40,52,0,0,0,[1895]
 8,16,0,0,0,0,0,0,192
 1240 DATA 8,16,16,16,8,0,0,0,[2768]
 16,8,8,8,16,0,0,0,120
 1250 DATA 40,16,56,16,40,0,0,0,[1597]
 16,16,56,16,16,0,0,0,288
 1260 DATA 0,0,0,0,8,16,0,0,[2345]
 0,0,56,0,0,0,0,0,80
 1270 DATA 0,0,0,0,16,0,0,0,[2232]
 0,8,16,32,0,0,0,0,72
 1280 DATA 56,40,40,40,56,0,0,0,[2008]
 16,48,16,16,56,0,0,0,384
 1290 DATA 56,8,56,32,56,0,0,0,[2231]
 56,8,24,8,56,0,0,0,360
 1300 DATA 32,32,40,56,8,0,0,0,[2131]
 56,32,56,8,56,0,0,0,376
 1310 DATA 56,32,56,40,56,0,0,0,[1838]
 56,8,8,16,16,0,0,0,344
 1320 DATA 56,40,56,40,56,0,0,0,[2376]
 56,40,56,8,56,0,0,0,464
 1330 DATA 0,0,16,0,16,0,0,0,[1912]
 0,0,8,0,8,16,0,0,64
 1340 DATA 8,16,32,16,8,0,0,0,[1814]
 0,56,0,56,0,0,0,0,192
 1350 DATA 32,16,8,16,32,0,0,0,[2553]
 56,8,16,0,16,0,0,0,200
 1360 DATA 24,44,44,32,24,0,0,0,[2455]
 16,40,40,56,40,0,0,0,360
 1370 DATA 48,40,48,40,48,0,0,0,[2048]
 56,40,32,40,56,0,0,0,448
 1380 DATA 48,40,40,40,48,0,0,0,[1543]
 56,32,48,32,56,0,0,0,440
 1390 DATA 56,32,48,32,32,0,0,0,[2158]
 56,32,40,40,56,0,0,0,424

1400 DATA 40, 40, 56, 40, 40, 0, 0, 0, [1976]
 56, 16, 16, 16, 56, 0, 0, 0, 376
 1410 DATA 8, 8, 8, 40, 56, 0, 0, 0, [1705]
 40, 40, 48, 40, 40, 0, 0, 0, 328
 1420 DATA 32, 32, 32, 32, 56, 0, 0, 0, [2758]
 40, 56, 40, 40, 40, 0, 0, 0, 400
 1430 DATA 56, 40, 40, 40, 40, 0, 0, 0, [2327]
 56, 40, 40, 40, 56, 0, 0, 0, 448
 1440 DATA 56, 40, 40, 56, 32, 0, 0, 0, [1900]
 56, 40, 40, 40, 48, 8, 0, 0, 456
 1450 DATA 56, 40, 40, 48, 40, 0, 0, 0, [2282]
 56, 32, 56, 8, 56, 0, 0, 0, 432
 1460 DATA 56, 16, 16, 16, 16, 0, 0, 0, [2219]
 40, 40, 40, 40, 56, 0, 0, 0, 336
 1470 DATA 40, 40, 40, 56, 16, 0, 0, 0, [2876]
 40, 40, 40, 56, 40, 0, 0, 0, 408
 1480 DATA 40, 40, 16, 56, 40, 0, 0, 0, [2703]
 40, 40, 16, 16, 0, 0, 0, 320
 1490 DATA 56, 8, 16, 32, 56, 0, 0, 0, [2400]
 56, 32, 32, 32, 56, 0, 0, 0, 376
 1500 DATA 0, 32, 16, 8, 0, 0, 0, 0, [2078]
 56, 8, 8, 8, 56, 0, 0, 0, 192
 1510 DATA 16, 56, 56, 16, 16, 0, 0, 0, [2373]
 0, 0, 0, 0, 0, 60, 0, 0, 220
 1520 DATA 16, 8, 0, 0, 0, 0, 0, 0, [1915]
 0, 56, 8, 56, 56, 0, 0, 0, 200
 1530 DATA 32, 32, 56, 40, 56, 0, 0, 0, [2696]
 0, 56, 32, 32, 56, 0, 0, 0, 392
 1540 DATA 8, 8, 56, 40, 56, 0, 0, 0, [2187]
 0, 56, 56, 32, 56, 0, 0, 0, 368
 1550 DATA 24, 32, 48, 32, 32, 0, 0, 0, [2912]
 0, 56, 40, 56, 8, 56, 0, 0, 384
 1560 DATA 32, 32, 56, 40, 40, 0, 0, 0, [2656]
 16, 0, 16, 16, 16, 0, 0, 0, 264
 1570 DATA 8, 0, 8, 8, 40, 56, 0, 0, [1885]
 32, 32, 40, 48, 40, 0, 0, 0, 312
 1580 DATA 16, 16, 16, 16, 8, 0, 0, 0, [1263]
 0, 40, 56, 40, 40, 0, 0, 0, 248
 1590 DATA 0, 48, 40, 40, 40, 0, 0, 0, [1579]
 0, 56, 40, 40, 56, 0, 0, 0, 360
 1600 DATA 0, 56, 40, 56, 32, 32, 0, 0, [1977]
 0, 56, 40, 56, 8, 8, 0, 0, 384
 1610 DATA 0, 24, 32, 32, 32, 0, 0, 0, [2526]
 0, 56, 32, 56, 56, 0, 0, 0, 320
 1620 DATA 16, 56, 16, 16, 24, 0, 0, 0, [2341]
 0, 40, 40, 40, 56, 0, 0, 0, 304
 1630 DATA 0, 40, 40, 40, 40, 16, 0, 0, [1511]
 0, 40, 40, 56, 40, 0, 0, 0, 312
 1640 DATA 0, 40, 16, 40, 40, 0, 0, 0, [2449]
 0, 40, 40, 56, 8, 56, 0, 0, 336
 1650 DATA 0, 56, 8, 16, 56, 0, 0, 0, [2602]
 24, 16, 32, 16, 24, 0, 0, 0, 248
 1660 DATA 16, 16, 16, 16, 16, 0, 0, 0, [1297]
 48, 16, 8, 16, 48, 0, 0, 0, 216
 1670 DATA 20, 40, 0, 0, 0, 0, 0, 0, [2255]
 28, 16, 56, 16, 60, 0, 0, 0, 236

1190 SAVE"LIGHT.FNT",B,40000,768 [1382]
 1200 DATA 0, 0, 0, 0, 0, 0, 0, 0, [1644]
 8, 8, 8, 8, 8, 0, 8, 0, 48
 1210 DATA 36, 36, 36, 36, 0, 0, 0, 0, [1862]
 36, 36, 126, 36, 126, 36, 36, 0, 540
 1220 DATA 8, 30, 40, 28, 10, 60, 8, 0, [2298]
 0, 98, 100, 8, 16, 38, 70, 0, 514
 1230 DATA 48, 72, 72, 48, 74, 68, 58, 0, [1614]
 8, 16, 32, 0, 0, 0, 0, 0, 496
 1240 DATA 4, 8, 16, 16, 16, 8, 4, 0, [2545]
 32, 16, 8, 8, 8, 16, 32, 0, 192
 1250 DATA 8, 42, 28, 8, 28, 42, 8, 0, [2438]
 0, 8, 8, 62, 8, 8, 0, 0, 258
 1260 DATA 0, 0, 0, 0, 0, 8, 8, 16, [2786]
 0, 0, 0, 126, 0, 0, 0, 0, 158
 1270 DATA 0, 0, 0, 0, 0, 0, 0, 8, [1803]
 0, 2, 4, 8, 16, 32, 64, 0, 134
 1280 DATA 56, 68, 76, 84, 100, 68, 56, 0, [1667]
 16, 48, 80, 16, 16, 16, 124, 0, 824
 1290 DATA 60, 66, 2, 12, 48, 64, 126, 0, [1566]
 60, 66, 2, 28, 2, 66, 60, 0, 662
 1300 DATA 4, 12, 20, 36, 126, 4, 4, 0, [1788]
 126, 64, 120, 4, 2, 68, 56, 0, 646
 1310 DATA 28, 32, 64, 124, 66, 66, 60, 0, [2548]
 126, 66, 4, 8, 8, 8, 8, 0, 668
 1320 DATA 60, 66, 66, 60, 66, 66, 60, 0, [2431]
 60, 66, 66, 62, 2, 4, 56, 0, 760
 1330 DATA 0, 0, 8, 0, 0, 8, 0, 0, [1619]
 0, 0, 8, 0, 0, 8, 8, 16, 56
 1340 DATA 4, 8, 16, 32, 16, 8, 4, 0, [1077]
 0, 0, 126, 0, 0, 126, 0, 0, 340
 1350 DATA 32, 16, 8, 4, 8, 16, 32, 0, [2247]
 60, 66, 2, 12, 16, 0, 16, 0, 288
 1360 DATA 28, 34, 74, 86, 76, 32, 28, 0, [2314]
 24, 36, 66, 66, 126, 66, 66, 0, 808

1370 DATA 124, 34, 34, 60, 34, 34, 124, 0, [3114]
 28, 34, 64, 64, 64, 34, 28, 0, 760
 1380 DATA 120, 36, 34, 34, 34, 36, 120, 0, [3438]
 126, 64, 64, 124, 64, 64, 126, 0, 1046
 1390 DATA 126, 64, 64, 124, 64, 64, 64, 0, [2917]
 28, 34, 64, 78, 66, 34, 28, 0, 902
 1400 DATA 66, 66, 66, 126, 66, 66, 66, 0, [2237]
 28, 8, 8, 8, 8, 8, 28, 0, 618
 1410 DATA 14, 4, 4, 4, 4, 68, 56, 0, [3081]
 66, 68, 72, 112, 72, 68, 66, 0, 678
 1420 DATA 64, 64, 64, 64, 64, 64, 126, 0, [2290]
 66, 102, 90, 90, 66, 66, 66, 0, 1056
 1430 DATA 66, 98, 82, 74, 70, 66, 66, 0, [2463]
 60, 66, 66, 66, 66, 66, 60, 0, 972
 1440 DATA 124, 66, 66, 124, 64, 64, 64, 0, [2241]
 60, 66, 66, 66, 74, 68, 58, 0, 1030
 1450 DATA 124, 66, 66, 124, 72, 68, 66, 0, [2901]
 60, 66, 64, 60, 2, 66, 60, 0, 964
 1460 DATA 62, 8, 8, 8, 8, 8, 8, 0, [1799]
 66, 66, 66, 66, 66, 66, 60, 0, 566
 1470 DATA 66, 66, 66, 36, 36, 24, 24, 0, [2154]
 66, 66, 66, 90, 90, 102, 66, 0, 864
 1480 DATA 66, 66, 36, 24, 36, 66, 66, 0, [2265]
 34, 34, 34, 28, 8, 8, 8, 0, 514
 1490 DATA 126, 2, 4, 24, 32, 64, 126, 0, [2601]
 28, 16, 16, 16, 16, 16, 28, 0, 514
 1500 DATA 0, 64, 32, 16, 8, 4, 2, 0, [2208]
 56, 8, 8, 8, 8, 8, 56, 0, 278
 1510 DATA 8, 28, 42, 8, 8, 8, 8, 0, [1482]
 0, 0, 0, 0, 0, 0, 255, 365
 1520 DATA 16, 8, 4, 0, 0, 0, 0, 0, [1963]
 0, 0, 56, 4, 60, 68, 58, 0, 274
 1530 DATA 64, 64, 92, 98, 66, 98, 92, 0, [2079]
 0, 0, 60, 66, 64, 66, 60, 0, 890
 1540 DATA 2, 2, 58, 70, 66, 70, 58, 0, [1977]
 0, 0, 60, 66, 126, 64, 60, 0, 702
 1550 DATA 12, 18, 16, 124, 16, 16, 16, 0, [1892]
 0, 0, 58, 70, 70, 58, 2, 60, 536
 1560 DATA 64, 64, 92, 98, 66, 66, 66, 0, [1976]
 8, 0, 24, 8, 8, 8, 28, 0, 600
 1570 DATA 4, 0, 12, 4, 4, 4, 68, 56, [1899]
 64, 64, 68, 72, 80, 104, 68, 0, 672
 1580 DATA 24, 8, 8, 8, 8, 8, 28, 0, [2159]
 0, 0, 118, 73, 73, 73, 73, 0, 502
 1590 DATA 0, 0, 92, 98, 66, 66, 66, 0, [2179]
 0, 0, 60, 66, 66, 66, 60, 0, 706
 1600 DATA 0, 0, 92, 98, 98, 92, 64, 64, [2235]
 0, 0, 58, 70, 70, 58, 2, 2, 768
 1610 DATA 0, 0, 92, 98, 64, 64, 64, 0, [1858]
 0, 0, 60, 64, 60, 2, 124, 0, 692
 1620 DATA 16, 16, 124, 16, 16, 18, 12, 0, [2804]
 0, 0, 66, 66, 66, 70, 58, 0, 544
 1630 DATA 0, 0, 66, 66, 66, 36, 24, 0, [1970]
 0, 0, 65, 73, 73, 73, 54, 0, 596
 1640 DATA 0, 0, 66, 36, 24, 36, 66, 0, [2633]
 0, 0, 66, 66, 70, 58, 2, 60, 550
 1650 DATA 0, 0, 126, 4, 24, 32, 126, 0, [2450]
 12, 16, 8, 48, 8, 16, 12, 0, 432
 1660 DATA 8, 8, 8, 8, 8, 8, 8, 0, [2140]
 48, 8, 16, 12, 16, 8, 48, 0, 212
 1670 DATA 118, 220, 0, 0, 0, 0, 0, 0, [2111]
 28, 34, 32, 120, 32, 34, 126, 0, 744

1190 SAVE"LO-INDEX.FNT",B,40000,768 [1623]
 1200 DATA 0, 0, 0, 0, 0, 0, 0, 0, [1820]
 0, 0, 0, 16, 16, 16, 0, 16, 64
 1210 DATA 0, 0, 0, 40, 0, 0, 0, 0, [2450]
 0, 0, 0, 24, 60, 24, 60, 24, 232
 1220 DATA 16, 0, 0, 16, 56, 48, 24, 56, [2235]
 0, 0, 0, 0, 36, 8, 16, 36, 312
 1230 DATA 0, 0, 0, 16, 40, 20, 40, 52, [2656]
 0, 0, 0, 8, 16, 0, 0, 0, 192
 1240 DATA 0, 0, 0, 8, 16, 16, 16, 8, [2384]
 0, 0, 0, 16, 8, 8, 8, 16, 120
 1250 DATA 0, 0, 0, 40, 16, 56, 16, 40, [2971]
 0, 0, 0, 16, 16, 56, 16, 16, 288
 1260 DATA 0, 0, 0, 0, 0, 0, 0, 8, 16, [2148]
 0, 0, 0, 0, 56, 0, 0, 0, 80
 1270 DATA 0, 0, 0, 0, 0, 0, 0, 0, 16, [2506]
 0, 0, 0, 0, 8, 16, 32, 0, 72
 1280 DATA 0, 0, 0, 56, 40, 40, 40, 56, [2455]
 0, 0, 0, 16, 48, 16, 16, 56, 384
 1290 DATA 0, 0, 0, 56, 8, 56, 32, 56, [2979]
 0, 0, 0, 56, 8, 24, 8, 56, 360
 1300 DATA 0, 0, 0, 32, 32, 40, 56, 8, [2133]
 0, 0, 0, 56, 32, 56, 8, 56, 376
 1310 DATA 0, 0, 0, 56, 32, 56, 40, 56, [2266]
 0, 0, 0, 56, 8, 8, 16, 16, 344
 1320 DATA 0, 0, 0, 56, 40, 56, 40, 56, [2937]
 0, 0, 0, 56, 40, 56, 8, 56, 464
 1330 DATA 0, 0, 0, 0, 0, 16, 0, 16, [2446]
 0, 0, 0, 0, 8, 0, 8, 16, 64

Tips & Tricks

```

1340 DATA 0, 0, 0, 8, 16, 32, 16, 8, [2365]
0, 0, 0, 0, 56, 0, 56, 0, 192
1350 DATA 0, 0, 0, 32, 16, 8, 16, 32, [2988]
0, 0, 0, 56, 8, 16, 0, 16, 200
1360 DATA 0, 0, 0, 24, 44, 44, 32, 24, [2725]
0, 0, 0, 16, 40, 40, 56, 40, 360
1370 DATA 0, 0, 0, 48, 40, 48, 40, 48, [2107]
0, 0, 0, 56, 40, 32, 40, 56, 448
1380 DATA 0, 0, 0, 48, 40, 40, 40, 48, [2082]
0, 0, 0, 56, 32, 48, 32, 56, 440
1390 DATA 0, 0, 0, 56, 32, 48, 32, 32, [2514]
0, 0, 0, 56, 32, 40, 40, 56, 424
1400 DATA 0, 0, 0, 40, 40, 56, 40, 40, [2964]
0, 0, 0, 56, 16, 16, 16, 56, 376
1410 DATA 0, 0, 0, 8, 8, 8, 40, 56, [2149]
0, 0, 0, 40, 40, 48, 40, 40, 328
1420 DATA 0, 0, 0, 32, 32, 32, 32, 56, [2493]
0, 0, 0, 40, 56, 40, 40, 40, 400
1430 DATA 0, 0, 0, 56, 40, 40, 40, 40, [2500]
0, 0, 0, 56, 40, 40, 40, 56, 448
1440 DATA 0, 0, 0, 56, 40, 40, 56, 32, [2058]
0, 0, 0, 56, 40, 40, 56, 468
1450 DATA 0, 0, 0, 56, 40, 40, 48, 40, [2335]
0, 0, 0, 56, 32, 56, 8, 56, 432
1460 DATA 0, 0, 0, 56, 16, 16, 16, 16, [2693]
0, 0, 0, 40, 40, 40, 40, 56, 336
1470 DATA 0, 0, 0, 40, 40, 40, 56, 16, [2591]
0, 0, 0, 40, 40, 56, 40, 408
1480 DATA 0, 0, 0, 40, 40, 16, 56, 40, [2263]
0, 0, 0, 40, 40, 16, 16, 16, 320
1490 DATA 0, 0, 0, 56, 8, 16, 32, 56, [2453]
0, 0, 0, 56, 32, 32, 32, 56, 376
1500 DATA 0, 0, 0, 32, 16, 8, 0, [2829]
0, 0, 0, 56, 8, 8, 8, 56, 192
1510 DATA 0, 0, 0, 16, 56, 56, 16, 16, [2976]
0, 0, 0, 0, 0, 0, 60, 220
1520 DATA 0, 0, 0, 16, 8, 0, 0, 0, [2103]
0, 0, 0, 0, 56, 8, 56, 56, 200
1530 DATA 0, 0, 0, 32, 32, 56, 40, 56, [2946]
0, 0, 0, 0, 56, 32, 32, 56, 392
1540 DATA 0, 0, 0, 8, 8, 56, 40, 56, [2462]
0, 0, 0, 0, 56, 56, 32, 56, 368
1550 DATA 0, 0, 0, 24, 32, 48, 32, 32, [2433]
0, 0, 0, 56, 40, 56, 8, 56, 384
1560 DATA 0, 0, 0, 32, 32, 56, 40, 40, [2319]
0, 0, 0, 16, 0, 16, 16, 16, 264
1570 DATA 0, 0, 8, 0, 8, 8, 40, 56, [2822]
0, 0, 0, 32, 32, 40, 48, 40, 312
1580 DATA 0, 0, 0, 16, 16, 16, 16, 8, [2769]
0, 0, 0, 40, 56, 40, 40, 248
1590 DATA 0, 0, 0, 48, 40, 40, 40, [2198]
0, 0, 0, 0, 56, 40, 40, 56, 360
1600 DATA 0, 0, 0, 56, 40, 56, 32, 32, [3505]
0, 0, 0, 56, 40, 56, 8, 8, 384
1610 DATA 0, 0, 0, 24, 32, 32, 32, [2686]
0, 0, 0, 0, 56, 32, 56, 56, 320
1620 DATA 0, 0, 0, 16, 56, 16, 16, 24, [2098]
0, 0, 0, 40, 40, 40, 56, 304
1630 DATA 0, 0, 0, 40, 40, 40, 40, 16, [3105]
0, 0, 0, 40, 40, 56, 40, 312
1640 DATA 0, 0, 0, 40, 16, 40, 40, [3248]
0, 0, 0, 40, 40, 56, 8, 56, 336
1650 DATA 0, 0, 0, 56, 8, 16, 56, [2300]
0, 0, 0, 24, 16, 32, 16, 24, 248
1660 DATA 0, 0, 0, 16, 16, 16, 16, [2911]
0, 0, 0, 48, 16, 8, 16, 48, 216
1670 DATA 0, 0, 0, 20, 40, 0, 0, [2108]
0, 0, 0, 28, 16, 56, 16, 60, 236

```

```

1190 SAVE"NARROW.FNT",B,40000,768 [1722]
1200 DATA 0, 0, 0, 0, 0, 0, 0, [2136]
32, 32, 32, 32, 0, 32, 0, 160
1210 DATA 80, 80, 80, 0, 0, 0, 0, [1909]
80, 80, 248, 80, 248, 80, 80, 0, 1136
1220 DATA 32, 120, 160, 112, 40, 240, 32, 0, [3322]
192, 200, 16, 32, 64, 152, 24, 0, 1416
1230 DATA 64, 160, 64, 168, 144, 152, 96, 0, [2389]
16, 32, 64, 0, 0, 0, 0, 960
1240 DATA 16, 32, 64, 64, 64, 32, 16, 0, [2711]
64, 32, 16, 16, 32, 64, 0, 528
1250 DATA 32, 168, 112, 32, 112, 168, 32, 0, [2739]
0, 32, 32, 248, 32, 32, 0, 0, 1032
1260 DATA 0, 0, 0, 0, 0, 32, 32, 64, [2055]
0, 0, 0, 120, 0, 0, 0, 0, 248
1270 DATA 0, 0, 0, 0, 0, 96, 96, 0, [2859]
0, 8, 16, 32, 64, 128, 0, 0, 440
1280 DATA 112, 136, 152, 168, 200, 136, 112, 0, [2625]
32, 96, 160, 32, 32, 32, 248, 0, 1648
1290 DATA 112, 136, 8, 16, 96, 128, 248, 0, [2391]
112, 136, 8, 48, 8, 136, 112, 0, 1304
1300 DATA 16, 48, 80, 144, 248, 16, 16, 0, [2927]
248, 128, 224, 16, 8, 16, 224, 0, 1432
1310 DATA 48, 64, 128, 240, 136, 136, 112, 0, [2817]

```

```

248, 136, 16, 32, 32, 32, 32, 0, 1392
1320 DATA 112, 136, 136, 112, 136, 136, 112, 0, [3083]
112, 136, 136, 120, 8, 16, 96, 0, 1504
1330 DATA 0, 0, 32, 0, 0, 32, 0, [2571]
0, 0, 32, 0, 0, 32, 32, 64, 224
1340 DATA 24, 48, 96, 192, 96, 48, 24, 0, [1834]
0, 0, 248, 0, 0, 248, 0, 0, 1024
1350 DATA 192, 96, 48, 24, 48, 96, 192, 0, [2143]
112, 136, 8, 16, 32, 0, 32, 0, 1032
1360 DATA 112, 136, 8, 104, 168, 168, 112, 0, [2565]
32, 80, 136, 136, 248, 136, 136, 0, 1712
1370 DATA 240, 72, 72, 112, 72, 72, 240, 0, [2636]
48, 72, 128, 128, 128, 72, 48, 0, 1504
1380 DATA 224, 80, 72, 72, 72, 80, 224, 0, [3033]
248, 128, 128, 240, 128, 128, 248, 0, 2072
1390 DATA 248, 128, 128, 240, 128, 128, 128, 0, [3773]
112, 136, 128, 184, 136, 136, 112, 0, 2072
1400 DATA 136, 136, 136, 248, 136, 136, 136, 0, [2416]
224, 64, 64, 64, 64, 64, 224, 0, 1832
1410 DATA 56, 16, 16, 16, 16, 144, 96, 0, [2353]
136, 144, 160, 192, 160, 144, 136, 0, 1432
1420 DATA 128, 128, 128, 128, 128, 128, 248, 0, [3330]
136, 216, 168, 168, 136, 136, 136, 0, 2112
1430 DATA 136, 200, 200, 168, 152, 152, 136, 0, [3214]
112, 136, 136, 136, 136, 136, 112, 0, 2048
1440 DATA 240, 136, 136, 240, 128, 128, 128, 0, [2920]
112, 136, 136, 168, 144, 104, 0, 2072
1450 DATA 240, 136, 136, 240, 160, 144, 136, 0, [2893]
112, 136, 128, 112, 8, 136, 112, 0, 1936
1460 DATA 248, 32, 32, 32, 32, 32, 32, 0, [3032]
136, 136, 136, 136, 136, 136, 112, 0, 1368
1470 DATA 136, 136, 136, 136, 80, 80, 32, 0, [2912]
136, 136, 136, 168, 168, 216, 136, 0, 1832
1480 DATA 136, 136, 80, 32, 80, 136, 136, 0, [2679]
136, 136, 136, 112, 32, 32, 32, 0, 1352
1490 DATA 248, 8, 16, 32, 64, 128, 248, 0, [3794]
112, 64, 64, 64, 64, 64, 112, 0, 1288
1500 DATA 0, 0, 128, 64, 32, 16, 8, 0, [2577]
56, 8, 8, 8, 8, 8, 56, 0, 400
1510 DATA 32, 80, 136, 0, 0, 0, 0, 0, [2170]
0, 0, 0, 0, 0, 0, 248, 496
1520 DATA 64, 32, 16, 0, 0, 0, 0, 0, [1039]
0, 0, 112, 8, 120, 136, 120, 0, 608
1530 DATA 128, 128, 176, 200, 136, 200, 176, 0, [2475]
0, 0, 112, 136, 128, 136, 112, 0, 1768
1540 DATA 8, 8, 104, 152, 136, 152, 104, 0, [3274]
0, 0, 112, 136, 248, 128, 112, 0, 1400
1550 DATA 16, 40, 32, 248, 32, 32, 32, 0, [2719]
0, 0, 104, 152, 152, 104, 8, 112, 1064
1560 DATA 128, 128, 240, 136, 136, 136, 136, 0, [3129]
32, 0, 96, 32, 32, 112, 0, 1376
1570 DATA 16, 0, 48, 16, 16, 16, 144, 96, [2929]
64, 64, 72, 80, 96, 80, 72, 0, 880
1580 DATA 96, 32, 32, 32, 32, 32, 112, 0, [1617]
0, 0, 208, 168, 168, 168, 168, 0, 1248
1590 DATA 0, 0, 176, 200, 136, 136, 136, 0, [2080]
0, 0, 112, 136, 136, 136, 112, 0, 1416
1600 DATA 0, 0, 176, 200, 200, 176, 128, 128, [3218]
0, 0, 104, 152, 152, 104, 8, 8, 1536
1610 DATA 0, 0, 176, 200, 128, 128, 128, 0, [2814]
0, 0, 120, 128, 112, 8, 240, 0, 1368
1620 DATA 64, 64, 240, 64, 64, 72, 48, 0, [1340]
0, 0, 144, 144, 144, 144, 104, 0, 1296
1630 DATA 0, 0, 136, 136, 136, 80, 32, 0, [3536]
0, 0, 136, 168, 168, 168, 80, 0, 1240
1640 DATA 0, 0, 136, 80, 32, 80, 136, 0, [2902]
0, 0, 136, 136, 152, 104, 8, 112, 1112
1650 DATA 0, 0, 248, 16, 32, 64, 248, 0, [2123]
16, 32, 32, 64, 32, 32, 16, 0, 832
1660 DATA 32, 32, 32, 0, 32, 32, 32, 0, [2143]
96, 16, 16, 8, 16, 16, 96, 0, 456
1670 DATA 64, 168, 16, 0, 0, 0, 0, 0, [1863]
24, 36, 32, 248, 32, 226, 92, 0, 938

```

```

1190 SAVE"SILICON.FNT",B,40000,768 [1641]
1200 DATA 0, 0, 0, 0, 0, 0, 0, 0, [2258]
56, 56, 56, 56, 24, 0, 24, 0, 272
1210 DATA 108, 108, 108, 0, 0, 0, 0, 0, [3090]
102, 255, 102, 102, 255, 102, 0, 0, 1242
1220 DATA 24, 62, 88, 60, 26, 124, 24, 0, [2429]
0, 198, 204, 24, 48, 102, 198, 0, 1182
1230 DATA 56, 108, 56, 118, 220, 204, 118, 0, [2366]
24, 24, 24, 0, 0, 0, 0, 0, 952
1240 DATA 30, 24, 24, 56, 56, 56, 62, 0, [3448]
120, 24, 24, 28, 28, 124, 0, 684
1250 DATA 0, 102, 60, 255, 60, 102, 0, 0, [2515]
0, 24, 24, 126, 24, 24, 0, 0, 801
1260 DATA 0, 0, 0, 0, 0, 0, 24, 24, 48, [2786]
0, 0, 0, 126, 0, 0, 0, 0, 222
1270 DATA 0, 0, 0, 0, 0, 24, 24, 0, [2192]
6, 12, 24, 48, 96, 192, 128, 0, 554
1280 DATA 127, 99, 99, 99, 99, 99, 127, 0, [2470]
56, 24, 24, 24, 62, 62, 62, 0, 1063

```

```

1290 DATA 127, 3, 3,127, 96, 96,127, 0, [2538]
254, 6, 6,127, 7, 7,255, 0,1241
1300 DATA 112,112,112,119,119,127, 7, 0, [2961]
127, 96, 96,127, 7, 7,127, 0,1295
1310 DATA 124,108, 96,127,103,103,127, 0, [2439]
127, 3, 3, 31, 24, 24, 24, 0,1024
1320 DATA 62, 54, 54,255,231,231,255, 0, [2119]
127, 99, 99,127, 7, 7, 7, 0,1615
1330 DATA 0, 24, 24, 0, 24, 24, 0, 0, [2830]
0, 24, 24, 0, 24, 24, 48, 0, 240
1340 DATA 12, 24, 48, 96, 48, 24, 12, 0, [2511]
0,126, 0, 0,126, 0, 0, 0, 516
1350 DATA 96, 48, 24, 12, 24, 48, 96, 0, [2522]
127, 99, 3, 31, 28, 0, 28, 0, 664
1360 DATA 127, 99,111,111,111,111, 96,127, 0, [4142]
63, 63, 35, 35,127,115,115, 0,1335
1370 DATA 126,102,102,127,103,103,127, 0, [2709]
127,103,103, 96, 99, 99,127, 0,1544
1380 DATA 126,102,102,102,119,119,119,127, 0, [3451]
127, 96, 96,127,112,112,127, 0,1611
1390 DATA 127, 96, 96,127,112,112,112, 0, [3624]
127, 99, 96,111,103,103,127, 0,1548
1400 DATA 115,115,115,127,115,115,115, 0, [2541]
12, 12, 12, 12, 60, 60, 60, 0,1045
1410 DATA 12, 12, 12, 14, 14,110,126, 0, [2621]
102,102,108,127,103,103,103, 0,1048
1420 DATA 48, 48, 48,112,112,112,126, 0, [3168]
103,127,127,119,103,103,103, 0,1391
1430 DATA 103,103,119,127,111,103,103, 0, [3135]
127, 99, 99,103,103,103,127, 0,1530
1440 DATA 127, 99, 99,127,112,112,112, 0, [3747]
127, 99, 99,103,103,103,127, 7,1556
1450 DATA 126,102,102,127,119,119,119, 0, [3039]
127, 96,127, 7,119,119,127, 0,1536
1460 DATA 127, 28, 28, 28, 28, 28, 28, 0, [2994]
103,103,103,103,103,103,127, 0,1040
1470 DATA 103,103,103,103,111, 62, 28, 0, [2789]
103,103,111,127,127,127,103, 0,1414
1480 DATA 115,115,115, 62,103,103,103, 0, [2748]
103,103,103,127, 28, 28, 28, 0,1236
1490 DATA 127,102,108, 24, 55,103,127, 0, [2527]
60, 48, 48, 48, 48, 48, 60, 0,1006
1500 DATA 192, 96, 48, 24, 12, 6, 2, 0, [2483]
60, 12, 12, 12, 12, 60, 0, 560
1510 DATA 24, 60,126, 24, 24, 24, 24, 0, [2675]
0, 0, 0, 0, 0, 0, 0,255, 561
1520 DATA 48, 24, 12, 0, 0, 0, 0, 0, [2200]
0, 0, 62, 6,126,102,126, 0, 506
1530 DATA 0,112,112,126,118,118,126, 0, [3104]
0, 0,124,112,112,124, 0,1296
1540 DATA 0, 6, 6,126,110,110,126, 0, [2598]
0, 0,126,102,126,112,126, 0,1076
1550 DATA 0, 30, 24, 62, 24, 24, 24, 0, [1975]
0, 0,126,110,110,126, 14,126, 800
1560 DATA 0,112,112,124,118,118,118, 0, [2547]
0, 24, 0, 24, 24, 60, 60, 0, 894
1570 DATA 0, 14, 0, 14, 14, 14, 14, 62, [2663]
0, 96, 96,108,120,108,102, 0, 762
1580 DATA 0, 60, 28, 28, 28, 28, 28, 0, [3208]
0, 0,103,127,127,107, 99, 0, 763
1590 DATA 0, 0,126,126,102,102,102, 0, [2409]
0, 0,126,110,110,110,126, 0,1140
1600 DATA 0, 0,126,118,118,126,112,112, [3504]
0, 0,126,110,110,126, 14, 14,1212
1610 DATA 0, 0,126,118,112,112,112, 0, [3196]
0, 0,126, 96,126, 14,126, 0,1068
1620 DATA 0, 48,124, 56, 56, 56, 62, 0, [3046]
0, 0,110,110,110,110,126, 0, 968
1630 DATA 0, 0,110,110,110, 60, 24, 0, [3027]
0, 0, 99,107,127,127, 54, 0, 928
1640 DATA 0, 0,102, 60, 24, 60,102, 0, [2574]
0, 0,110,110,110,126, 14,126, 944
1650 DATA 0, 0,126, 12, 24, 48,126, 0, [2860]
14, 24, 24,112, 24, 24, 14, 0, 572
1660 DATA 24, 24, 24, 24, 24, 24, 24, 0, [2474]
112, 24, 24, 14, 24, 24,112, 0, 502
1670 DATA 118,220, 0, 0, 0, 0, 0, 0, [2650]
60,102, 96,248,112,118,254, 0,1328

```

```

0, 0, 0,126, 0, 0, 0, 0, 198
1270 DATA 0, 0, 0, 0, 0, 16, 56, 16, [2603]
6, 12, 24, 48, 96,192,128, 0, 594
1280 DATA 56,108,206,214,230,108, 56, 0, [2492]
24, 56, 24, 24, 24, 24,126, 0,1280
1290 DATA 60, 70, 6, 12, 24, 48,126, 0, [2333]
60,102, 6, 28, 6,102, 60, 0, 710
1300 DATA 28, 60,108,204,254, 12, 12, 0, [2341]
62, 96, 96,124, 6, 70, 60, 0,1192
1310 DATA 28, 48, 96,124,102,102, 60, 0, [2914]
126, 70, 6, 12, 24, 48, 48, 0, 894
1320 DATA 60,102,102, 60,102,102, 60, 0, [2788]
60,102,102, 62, 6, 12, 56, 0, 988
1330 DATA 0, 0, 8, 16, 0, 8, 16, 0, [2168]
0, 0, 8, 16, 0, 24, 48, 0, 144
1340 DATA 12, 24, 48, 96, 48, 24, 12, 0, [2742]
0, 0,126, 0, 0,126, 0, 0, 516
1350 DATA 96, 48, 24, 12, 24, 48, 96, 0, [3146]
60,102,102, 12, 24, 0, 24, 0, 672
1360 DATA 124,198,222,222,222,192,124, 0, [2632]
54,110,198,198,198,206,118, 0,2386
1370 DATA 92,102,102,124,102,102,124, 0, [3162]
60,102,192,192,192,102, 60, 0,1648
1380 DATA 6, 54,110,198,198,206,124, 0, [2680]
60,102,198,252,192,102, 60, 0,1862
1390 DATA 56,100, 96,248, 96, 96, 64, 0, [2943]
58,110,198,198,110, 62,198,124,1814
1400 DATA 192,216,252,230,198,204,222, 0, [2299]
28, 56, 24, 24, 24, 28, 56, 0,1754
1410 DATA 126, 12, 12, 12,140,204,120, 0, [2983]
194,102,108,120,108,102, 66, 0,1426
1420 DATA 240, 48, 96, 96, 96,102,252, 0, [3097]
172,214,214,214,214,214,148, 0,2320
1430 DATA 220,246,230,198,206,220,222, 0, [3488]
56,108,198,198,198,108, 56, 0,2464
1440 DATA 220,102,102,124, 96, 96, 64, 0, [2790]
60,102,198,198,218,204,118, 0,1902
1450 DATA 108,118,102,124,120,108,102, 0, [2906]
60, 98, 96, 60, 6, 70, 60, 0,1232
1460 DATA 126, 24, 48, 48, 48, 48, 24, 0, [2303]
118, 54,102,102,102,110, 54, 0,1008
1470 DATA 118, 54,102,102,102, 54, 28, 0, [2244]
204, 70,214,214,214,254,108, 0,1838
1480 DATA 198,108, 56, 56,108,198,198, 0, [2210]
102, 54, 54, 30, 6,108, 56, 0,1332
1490 DATA 126,198, 12, 24, 48,102,252, 0, [2532]
60, 48, 48, 48, 48, 48, 60, 0,1122
1500 DATA 192, 96, 48, 24, 12, 6, 2, 0, [2483]
60, 12, 12, 12, 12, 60, 0, 560
1510 DATA 24, 60,126, 24, 24, 24, 24, 0, [2675]
0, 0, 0, 0, 0, 0, 0,255, 561
1520 DATA 48, 24, 12, 0, 0, 0, 0, 0, [2277]
0, 0,120, 12,124,204,118, 0, 662
1530 DATA 224, 96,124,102,102,102,220, 0, [2607]
0, 0, 60,102, 96,102, 60, 0,1390
1540 DATA 28, 12,124,204,204,204,118, 0, [2268]
0, 0, 60,102,126, 96, 60, 0,1338
1550 DATA 28, 54, 48,120, 48, 48,120, 0, [3191]
0, 0, 62,102,102, 62, 6,124, 924
1560 DATA 224, 96,108,118,102,102,230, 0, [3423]
24, 0, 56, 24, 24, 24, 60, 0,1192
1570 DATA 6, 0, 14, 6, 6,102,102, 60, [2839]
224, 96,102,108,120,108,230, 0,1284
1580 DATA 56, 24, 24, 24, 24, 24, 60, 0, [2998]
0, 0,108,254,214,214,198, 0,1224
1590 DATA 0, 0,220,102,102,102,102, 0, [2686]
0, 0, 60,102,102,102, 60, 0,1054
1600 DATA 0, 0,220,102,102,124, 96,240, [2384]
0, 0,118,204,204,124, 12, 30,1576
1610 DATA 0, 0,220,118, 96, 96,240, 0, [3032]
0, 0, 60, 96, 60, 6,124, 0,1116
1620 DATA 48, 48,124, 48, 48, 54, 28, 0, [2642]
0, 0,102,102,102,102, 62, 0, 868
1630 DATA 0, 0,102,102,102, 60, 24, 0, [3039]
0, 0,198,214,214,254,108, 0,1378
1640 DATA 0, 0,198,108, 56,108,198, 0, [2963]
0, 0,102,102,102, 62, 6,124,1166
1650 DATA 0, 0,126, 76, 24, 50,126, 0, [2970]
14, 24, 24,112, 24, 24, 14, 0, 638
1660 DATA 24, 24, 24, 24, 24, 24, 24, 0, [2474]
112, 24, 24, 14, 24, 24,112, 0, 502
1670 DATA 118,220, 0, 0, 0, 0, 0, 0, [2725]
60,102, 96,248, 96,102,254, 0,1296

```

```

1190 SAVE"SORCERY.FMT",B,40000,768 [2074]
1200 DATA 0, 0, 0, 0, 0, 0, 0, 0, [2460]
24, 24, 24, 24, 24, 0, 24, 0, 144
1210 DATA 108,108,108, 0, 0, 0, 0, 0, [2663]
108,108,254,108,254,108,108, 0,1372
1220 DATA 24, 62, 88, 60, 26,124, 24, 0, [2841]
0, 70,140, 24, 48, 98,196, 0, 984
1230 DATA 56,108, 56,118,220,204,118, 0, [2668]
24, 24, 48, 0, 0, 0, 0, 0, 976
1240 DATA 12, 24, 48, 48, 48, 24, 12, 0, [3592]
48, 24, 12, 12, 12, 24, 48, 0, 396
1250 DATA 0,102, 60,255, 60,102, 0, 0, [2515]
0, 24, 24,126, 24, 24, 0, 0, 801
1260 DATA 0, 0, 0, 0, 0, 24, 48, 0, [1879]

```

```

1190 SAVE"SPLIT.FMT",B,40000,768 [1428]
1200 DATA 0, 0, 0, 0, 0, 0, 0, 0, [2460]
24, 24, 24, 24, 24, 0, 24, 0, 144
1210 DATA 108,108,108, 0, 0, 0, 0, 0, [2663]
108,108,254,108,254,108,108, 0,1372
1220 DATA 24, 62, 88, 60, 26,124, 24, 0, [2429]
0,198,204, 24, 48,102,198, 0,1182
1230 DATA 40,108, 40,102,204,204,102, 0, [2154]
24, 24, 48, 0, 0, 0, 0, 0, 896

```

Tips & Tricks

```

1240 DATA 12, 24, 48, 48, 48, 24, 12, 0, [3592]
48, 24, 12, 12, 12, 24, 48, 0, 396
1250 DATA 0, 102, 60, 255, 60, 102, 0, 0, [2425]
24, 24, 0, 102, 0, 24, 24, 0, 777
1260 DATA 0, 0, 0, 0, 0, 0, 24, 24, 48, [2786]
0, 0, 0, 126, 0, 0, 0, 0, 222
1270 DATA 0, 0, 0, 0, 0, 0, 24, 24, 0, [2192]
6, 12, 24, 48, 96, 192, 128, 0, 554
1280 DATA 108, 198, 198, 198, 198, 198, 108, 0, [3199]
24, 88, 24, 24, 24, 24, 92, 0, 1506
1290 DATA 44, 102, 6, 44, 96, 102, 110, 0, [2279]
44, 102, 6, 12, 6, 102, 44, 0, 820
1300 DATA 12, 44, 108, 204, 238, 12, 14, 0, [2989]
110, 98, 96, 108, 6, 102, 44, 0, 1196
1310 DATA 44, 102, 96, 108, 102, 102, 44, 0, [3151]
118, 102, 6, 12, 24, 24, 24, 0, 908
1320 DATA 44, 102, 102, 44, 102, 102, 44, 0, [2615]
52, 102, 102, 54, 6, 102, 52, 0, 1010
1330 DATA 0, 0, 24, 24, 0, 24, 24, 0, 24, 24, 0, [2008]
0, 0, 24, 24, 0, 24, 24, 48, 240
1340 DATA 12, 24, 48, 96, 48, 24, 12, 0, [2742]
0, 0, 126, 0, 0, 126, 0, 0, 516
1350 DATA 96, 48, 24, 12, 24, 48, 96, 0, [3146]
60, 102, 102, 12, 24, 0, 24, 0, 672
1360 DATA 124, 198, 222, 222, 222, 192, 124, 0, [3026]
8, 44, 102, 102, 110, 102, 102, 0, 1874
1370 DATA 236, 102, 100, 108, 102, 102, 236, 0, [2923]
54, 98, 192, 192, 192, 98, 54, 0, 1866
1380 DATA 232, 108, 102, 102, 102, 108, 232, 0, [1977]
238, 98, 100, 108, 100, 98, 238, 0, 1966
1390 DATA 238, 98, 100, 108, 100, 96, 224, 0, [2962]
44, 102, 192, 192, 206, 102, 46, 0, 1848
1400 DATA 102, 102, 102, 110, 102, 102, 102, 0, [2707]
116, 48, 48, 48, 48, 48, 116, 0, 1194
1410 DATA 14, 12, 12, 12, 204, 204, 104, 0, [2923]
230, 102, 108, 104, 108, 102, 230, 0, 1546
1420 DATA 224, 96, 96, 96, 98, 102, 238, 0, [3099]
198, 206, 222, 222, 214, 198, 198, 0, 2408
1430 DATA 198, 230, 230, 206, 206, 198, 198, 0, [3457]
40, 68, 198, 198, 198, 68, 40, 0, 2276
1440 DATA 236, 102, 102, 108, 96, 96, 224, 0, [3097]
108, 198, 198, 198, 202, 204, 102, 0, 2174
1450 DATA 236, 102, 102, 96, 108, 102, 230, 0, [2886]
54, 98, 112, 60, 14, 70, 108, 0, 1492
1460 DATA 246, 178, 48, 48, 48, 48, 116, 0, [2957]
102, 102, 102, 102, 102, 102, 44, 0, 1388
1470 DATA 102, 102, 102, 102, 102, 44, 8, 0, [3519]
198, 198, 198, 198, 238, 238, 198, 0, 2028
1480 DATA 198, 108, 40, 40, 108, 198, 198, 0, [2595]
102, 102, 102, 48, 24, 24, 60, 0, 1352
1490 DATA 242, 198, 140, 24, 50, 102, 206, 0, [3198]
60, 48, 48, 48, 48, 48, 60, 0, 1322
1500 DATA 192, 96, 48, 24, 12, 6, 2, 0, [2483]
60, 12, 12, 12, 12, 12, 60, 0, 560
1510 DATA 24, 60, 126, 24, 24, 24, 24, 0, [2675]
0, 0, 0, 0, 0, 0, 0, 0, 255, 561
1520 DATA 48, 24, 12, 0, 0, 0, 0, 0, [2426]
0, 0, 104, 12, 108, 204, 110, 0, 622
1530 DATA 224, 96, 108, 102, 102, 102, 236, 0, [3298]
0, 0, 54, 98, 96, 98, 54, 0, 1370
1540 DATA 12, 12, 108, 204, 204, 204, 110, 0, [2750]
0, 0, 44, 102, 108, 96, 44, 0, 1248
1550 DATA 22, 50, 48, 116, 48, 48, 116, 0, [3044]
0, 0, 54, 102, 102, 54, 6, 116, 882
1560 DATA 224, 96, 108, 102, 102, 102, 230, 0, [3106]
24, 0, 88, 24, 24, 24, 92, 0, 1240
1570 DATA 6, 0, 6, 6, 6, 102, 102, 52, [2172]
224, 96, 102, 108, 104, 108, 230, 0, 1252
1580 DATA 112, 48, 48, 48, 48, 48, 116, 0, [2956]
0, 0, 76, 222, 214, 198, 198, 0, 1376
1590 DATA 0, 0, 236, 102, 102, 102, 102, 0, [2917]
0, 0, 44, 102, 102, 102, 44, 0, 1038
1600 DATA 0, 0, 204, 102, 102, 108, 96, 224, [2616]
0, 0, 102, 204, 204, 108, 12, 14, 1480
1610 DATA 0, 0, 108, 102, 96, 96, 96, 0, [2742]
0, 0, 44, 96, 44, 6, 108, 0, 796
1620 DATA 96, 96, 108, 96, 96, 102, 12, 0, [2317]
0, 0, 102, 102, 102, 102, 54, 0, 1068
1630 DATA 0, 0, 102, 102, 102, 44, 8, 0, [3029]
0, 0, 198, 214, 214, 222, 76, 0, 1282
1640 DATA 0, 0, 198, 108, 40, 108, 198, 0, [2716]
0, 0, 102, 102, 102, 54, 6, 124, 1142
1650 DATA 0, 0, 102, 76, 24, 50, 102, 0, [2955]
12, 24, 24, 48, 24, 24, 12, 0, 522
1660 DATA 24, 24, 24, 0, 24, 24, 24, 0, [2918]
48, 24, 24, 12, 24, 24, 48, 0, 348
1670 DATA 118, 220, 0, 0, 0, 0, 0, 0, [1919]
44, 102, 96, 232, 96, 102, 238, 0, 1248

```

```

1190 SAVE"UNGER. FNT", B, 40000, 768 [1278]
1200 DATA 0, 0, 0, 0, 0, 0, 0, 0, [1604]
8, 24, 24, 24, 16, 0, 24, 0, 120
1210 DATA 36, 108, 72, 0, 0, 0, 0, 0, [3480]

```

```

108, 108, 254, 108, 254, 108, 108, 0, 1264
1220 DATA 24, 62, 88, 60, 26, 124, 24, 0, [2429]
0, 198, 204, 24, 48, 102, 198, 0, 1182
1230 DATA 56, 108, 56, 118, 220, 204, 118, 0, [2668]
24, 24, 48, 0, 0, 0, 0, 0, 976
1240 DATA 8, 16, 32, 32, 32, 16, 8, 0, [1918]
32, 16, 8, 8, 8, 16, 32, 0, 264
1250 DATA 0, 70, 40, 126, 52, 70, 0, 0, [1726]
0, 16, 16, 124, 16, 16, 0, 0, 546
1260 DATA 0, 0, 0, 0, 0, 0, 8, 24, 16, [2249]
0, 0, 4, 24, 32, 0, 0, 0, 108
1270 DATA 0, 0, 0, 0, 0, 0, 24, 24, 0, [2122]
2, 4, 8, 16, 32, 64, 128, 0, 302
1280 DATA 28, 38, 35, 65, 65, 34, 28, 0, [2306]
16, 48, 16, 16, 16, 16, 56, 0, 477
1290 DATA 56, 68, 4, 56, 64, 68, 124, 0, [1965]
56, 68, 4, 24, 4, 68, 56, 0, 720
1300 DATA 8, 24, 40, 72, 252, 8, 28, 0, [2253]
124, 68, 64, 120, 4, 68, 56, 0, 936
1310 DATA 56, 68, 64, 120, 68, 68, 56, 0, [1940]
124, 68, 4, 8, 16, 16, 16, 0, 752
1320 DATA 56, 68, 68, 56, 68, 68, 56, 0, [2745]
56, 68, 68, 60, 4, 68, 56, 0, 820
1330 DATA 0, 0, 16, 16, 0, 16, 16, 0, [2850]
0, 0, 24, 24, 0, 8, 24, 16, 160
1340 DATA 4, 8, 16, 32, 16, 8, 4, 0, [1784]
0, 4, 24, 32, 4, 24, 32, 0, 208
1350 DATA 32, 16, 8, 4, 8, 16, 32, 0, [2569]
56, 68, 8, 16, 16, 0, 16, 0, 296
1360 DATA 124, 198, 222, 222, 222, 192, 124, 0, [2261]
49, 74, 6, 26, 34, 119, 10, 0, 1622
1370 DATA 62, 73, 10, 30, 33, 114, 76, 0, [1767]
30, 35, 36, 68, 72, 33, 30, 0, 702
1380 DATA 60, 74, 73, 17, 33, 49, 78, 0, [1535]
27, 36, 75, 78, 72, 49, 30, 0, 751
1390 DATA 63, 72, 75, 14, 8, 36, 24, 0, [2667]
26, 37, 72, 78, 89, 33, 30, 0, 657
1400 DATA 14, 16, 20, 26, 17, 82, 36, 0, [1965]
50, 76, 4, 8, 8, 36, 24, 0, 417
1410 DATA 50, 76, 4, 8, 8, 68, 56, 0, [3255]
14, 17, 44, 18, 12, 74, 49, 0, 498
1420 DATA 24, 36, 64, 32, 32, 50, 76, 0, [2914]
42, 85, 85, 21, 37, 74, 41, 0, 699
1430 DATA 46, 83, 81, 17, 18, 82, 33, 0, [3424]
28, 38, 35, 65, 65, 34, 92, 0, 717
1440 DATA 44, 82, 81, 18, 60, 16, 32, 0, [2969]
28, 38, 35, 73, 69, 34, 93, 0, 703
1450 DATA 44, 82, 17, 30, 20, 18, 35, 0, [2464]
29, 38, 64, 82, 77, 34, 28, 0, 598
1460 DATA 49, 78, 40, 8, 8, 57, 70, 0, [2414]
34, 102, 34, 34, 34, 55, 10, 0, 613
1470 DATA 36, 110, 33, 33, 34, 108, 48, 0, [3061]
42, 121, 41, 41, 41, 106, 20, 0, 814
1480 DATA 67, 36, 24, 24, 40, 69, 66, 0, [1830]
34, 102, 34, 34, 28, 72, 48, 0, 678
1490 DATA 56, 68, 4, 56, 68, 2, 50, 76, [1553]
56, 32, 32, 32, 32, 32, 56, 0, 652
1500 DATA 0, 64, 32, 16, 8, 4, 2, 0, [2231]
28, 4, 4, 4, 4, 4, 28, 0, 202
1510 DATA 24, 60, 126, 24, 24, 24, 24, 0, [2675]
0, 0, 0, 0, 0, 0, 0, 0, 255, 561
1520 DATA 48, 24, 12, 0, 0, 0, 0, 0, [2173]
0, 0, 18, 44, 68, 78, 52, 0, 344
1530 DATA 80, 32, 32, 40, 50, 36, 88, 0, [1672]
0, 0, 28, 38, 64, 34, 28, 0, 550
1540 DATA 16, 16, 8, 52, 66, 68, 56, 0, [2077]
0, 0, 24, 44, 80, 102, 56, 0, 588
1550 DATA 24, 44, 32, 120, 48, 32, 32, 32, [2916]
0, 0, 16, 40, 68, 60, 8, 112, 668
1560 DATA 80, 32, 44, 50, 33, 34, 36, 8, [2264]
0, 16, 0, 48, 16, 24, 16, 0, 437
1570 DATA 0, 8, 0, 24, 8, 8, 20, 32, [2435]
40, 20, 24, 20, 16, 16, 8, 0, 244
1580 DATA 40, 16, 16, 16, 16, 16, 8, 0, [2331]
0, 0, 84, 42, 42, 42, 21, 0, 359
1590 DATA 0, 0, 88, 36, 36, 54, 36, 0, [2104]
0, 0, 24, 36, 70, 68, 56, 0, 504
1600 DATA 0, 0, 80, 40, 36, 120, 32, 32, [1950]
0, 0, 20, 44, 68, 76, 52, 4, 604
1610 DATA 0, 0, 88, 36, 32, 32, 16, 0, [2633]
8, 4, 56, 64, 56, 68, 56, 0, 516
1620 DATA 8, 16, 56, 16, 16, 24, 16, 0, [2248]
0, 0, 36, 108, 36, 46, 20, 0, 398
1630 DATA 0, 0, 36, 106, 34, 100, 24, 0, [2735]
0, 0, 42, 105, 41, 106, 20, 0, 614
1640 DATA 0, 0, 104, 20, 16, 16, 44, 64, [2226]
0, 0, 88, 36, 36, 40, 40, 4, 508
1650 DATA 0, 0, 56, 68, 24, 68, 36, 24, [1951]
12, 16, 16, 32, 16, 16, 12, 0, 396
1660 DATA 16, 16, 16, 0, 16, 16, 16, 0, [2415]
48, 8, 4, 8, 8, 48, 0, 228
1670 DATA 118, 220, 0, 0, 0, 0, 0, 0, [1753]
0, 0, 0, 0, 0, 0, 0, 0, 338

```

Listing OCP

Komplexe Grafiken auf dem CPC

Dieses Programm liefert Ihnen die mathematischen Grundlagen für die wichtigsten Abbildungen **VERSCHIEBEN**, **STRECKEN**, **SPIEGELN** und **DREHEN** im zwei- und dreidimensionalen Raum. Es besteht aus verschiedenen Unterprogrammen, die Sie je nach Bedarf in eigene Basicprogramme einbinden, oder auch leicht in andere Programmiersprachen (z.B. Pascal) umsetzen können.

Die grundsätzliche Funktionsweise ist allen Unterprogrammen gemeinsam: Sie übergeben die Koordinaten eines beliebigen Punktes X in den Variablen x_1 , x_2 und x_3 , sowie die nötigen Parameter für die jeweilige Abbildung, und erhalten die Koordinaten des Bildpunktes X' – ebenfalls in den Variablen x_1 , x_2 und x_3 – zurück. Um eine komplexere Grafik zu erzeugen, müssen Sie jeden der Eckpunkte einzeln für sich abbilden lassen. Die meisten der Programme benötigen eigene Variablen. Diese werden im REM-Kopf jeder Routine genannt («lokale Variablen»). Das Hauptprogramm sollten Sie nicht verwenden, um unerwünschte Kollisionen zu vermeiden. Das Koordinatensystem ist wie folgt aufgebaut: Die Koordinatenachsen stehen senkrecht aufeinander und werden – wie in der analytischen Geometrie üblich – mit x_1 , x_2 und x_3 bezeichnet, wobei die x_1 -Achse horizontal (von links nach rechts), die x_2 -Achse vertikal (von unten nach oben) und die x_3 -Achse in die Tiefe (in den Bildschirm hinein) verläuft. Der Ursprung fällt mit dem gewählten Grafikursprung (Origin) zusammen, seine x_3 -Koordinate ist immer 0. Ein Punkt P besitzt die Koordinaten (p_1, p_2, p_3) , ein Punkt X entsprechend (x_1, x_2, x_3) . Beim Arbeiten mit 2D-Grafiken fällt die x_3 -Achse weg. Die Funktionsweise der Routinen im einzelnen:

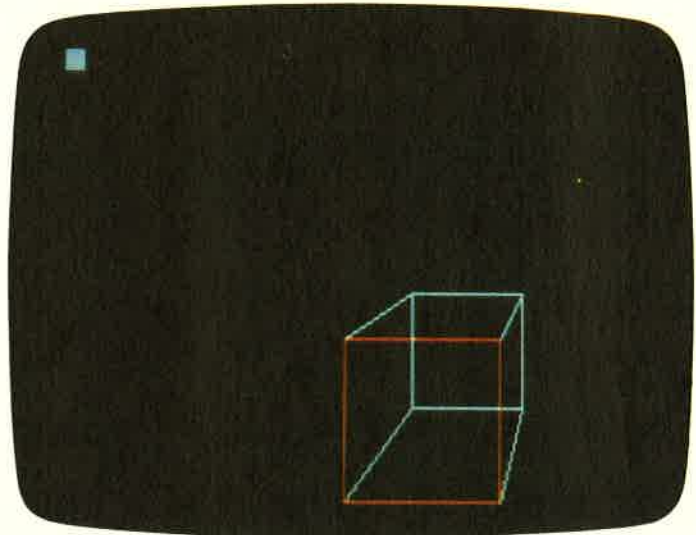
1. 2D VERSCHIEBEN: Geben Sie einen Punkt X in den Variablen x_1 und x_2 , sowie den Vektor v , um den X verschoben werden soll, in den Variablen v_1 und v_2 vor. Sie erhalten dann die Koordinaten des Bildpunktes in den Variablen x_1 und x_2 zurück.

2. 2D STRECKEN: Vorgegeben werden muß ein Punkt $X(x_1, x_2)$, das Streckzentrum $Z(z_1, z_2)$, sowie der Streckfaktor m . Der Streckfaktor gibt an, wie stark die Grafik gestreckt werden soll: Alle Streckenlängen innerhalb einer Figur werden m -fach. Für $m > 1$ wird eine Figur damit vergrößert, für $m < 1$ verkleinert. Bei $m < 0$ wird auf die andere Seite des Zentrums gestreckt. Die Lage des Zentrums gibt an, wie stark eine Figur beim Streckvorgang verschoben wird. Ist Z sehr weit entfernt, wird stark verschoben, liegt es sehr nahe, wird kaum verschoben.

3. 2D SPIEGELN: Der Punkt X wird an der Achse PQ gespiegelt. Als Parameter müssen hier die Koordinaten der Punkte $P(p_1, p_2)$ und $Q(q_1, q_2)$ übergeben werden.

4. 2D DREHEN: X wird um den Drehpunkt M und um den Winkel a gedreht. Mit den Befehlen **DEG** und **RAD** können Sie einstellen, ob Ihr übergebener Winkel a im Winkel- oder Bogenmaß sein soll. Gedreht wird gegen den Uhrzeigersinn, um in die andere Richtung zu drehen, müssen Sie einen negativen Winkel angeben.

5. 3D DARSTELLEN: Grundlage für alle 3D-Grafiken ist eine Darstellungsmethode auf dem zweidimensionalen Bildschirm. Dazu dient diese Routine, die die Bildschirmkoordinaten eines 3D-Punktes berechnet. Hierbei wird die Zentralperspektive gewählt, welche wohl am realistischsten wirkt, da sie der menschlichen Wahrnehmung am nächsten kommt. Übergeben Sie der Routine die 3D-Koordinaten eines Punktes X in x_1 , x_2 und x_3 , sowie die Bildschirmkoordinaten (2D) des Blickzentrums



$S(s_1, s_2)$. Als Rückgabe erhalten Sie die Bildschirmkoordinaten (y_1, y_2) des Punktes X . Das Blickzentrum kann dabei ohne weiteres auch außerhalb des Bildschirmbereichs liegen.

Hinweis: Die Konstante "300" in Zeile 2010 muß ggf. je nach Lage des Blickzentrums geändert werden, um keine allzu große oder kleine Verzerrung zu erhalten.

6. 3D VERSCHIEBEN: Der Punkt $X(x_1, x_2, x_3)$ wird um den Vektor $v(v_1, v_2, v_3)$ verschoben.

7. 3D STRECKEN: Wie bei 2D Strecken müssen Sie auch hier als Parameter das Zentrum Z (diesmal 3D: (z_1, z_2, z_3)) und den Streckfaktor m angeben.

8. 3D SPIEGELN: X wird an der Spiegelebene PQR gespiegelt. Beachten Sie, daß Sie hier 9 Koordinaten allein als Parameter angeben müssen.

9. 3D DREHEN: X wird um die Drehachse LM und um den Winkel a gedreht. Drehrichtung gegen den Uhrzeigersinn, von L nach M gesehen. Umkehrung der Drehrichtung durch Vertauschen von L und M oder Angabe eines negativen Winkels.

Allgemeine Hinweise

Muß als Parameter eine Gerade angegeben werden (2D Spiegeln, 3D Drehen), so sind jeweils die Koordinaten zweier verschiedener Punkte zu übergeben, die auf dieser Geraden liegen. Bei Angabe einer Ebene (3D Spiegeln) müssen 3 Punkte angegeben werden, die auf dieser Ebene, jedoch nicht auf einer Geraden liegen (also ein Dreieck bilden). Bei Nichtbeachtung dieser Punkte kommt es meist zu »Division by zero«-Errors. Für schnelle Animationen ist die Geschwindigkeit des Programms viel zu gering. Deshalb empfiehlt sich folgende Vorgehensweise: Berechnen Sie zunächst mit Hilfe des Programms alle nötigen Bildschirmkoordinaten für die gesamte Grafik und speichern diese in einer Datei ab. Das eigentliche Hauptprogramm lädt dann diese Datei in ein großes Variablenfeld. Alle Grafikkoordinaten stehen dann sofort zur Verfügung, und das Grafikprogramm wird gar nicht mehr benötigt. Auf diese Weise ist es auch möglich, die Animation in Maschinensprache zu schreiben, da keinerlei Berechnungen mehr durchgeführt werden müssen. Somit können auch sehr komplexe Grafiken programmiert werden. Dem Grafikprogramm sind zwei Demos (eines für 2D- und eines für 3D-Grafiken) beigelegt. Diese erfüllen einen doppelten Zweck: Zum einen können Sie daraus entnehmen, wie die Anwendung der Routinen in der Praxis aussieht, zum anderen bieten sie eine Kontrollmöglichkeit: Erscheinen »unmögliche« Grafiken, haben Sie sich wahrscheinlich irgendwo vertippt. Dies erspart dann großen Ärger mit Ihren eigenen Grafiken.

Und nun wünschen wir Ihnen viel Spaß und Erfolg beim Programmieren Ihrer ersten 3D-Animation! (M. Knaup/JB)

FÜR 464-664-6128



```

1000 : [174]
1001 REM 2D verschieben (um Vektor v) [1877]
1002 REM lokale Variablen: - [2020]
1010 x1=x1+v1:x2=x2+v2 [470]
1020 RETURN [555]
1100 : [174]
1101 REM 2D strecken (Zentrum Z, Streckfa [2636]
ktor m)
1102 REM lokale Variablen: - [2020]
1110 x1=m*(x1-z1)+z1 [1057]
1120 x2=m*(x2-z2)+z2 [1531]
1130 RETURN [555]
1200 : [174]
1201 REM 2D spiegeln (Spiegelachse PQ) [1727]
1202 REM lokale Variablen: n0,n1,n2 [2468]
1210 n1=q2-p2:n2=p1-q1:n0=SQR(n1^2+n2^2):n [3507]
l=n1/n0:n2=n2/n0
1220 n0=2*(n1*(x1-p1)+n2*(x2-p2)) [1651]
1230 x1=x1-n1*n0:x2=x2-n2*n0 [1345]
1240 RETURN [555]
1300 : [174]
1301 REM 2D drehen (Drehzentrum M, Winkel [3039]
a)
1302 REM lokale Variablen: a1,a2,sia,coa [2374]
1310 sia=SIN(a):coa=COS(a) [1094]
1320 a1=x1-m1:a2=x2-m2 [1848]
1330 x1=a1*coa-a2*sia+m1 [1174]
1340 x2=a1*sia+a2*coa+m2 [870]
1350 RETURN [555]
2000 : [174]
2001 REM 3D darstellen ( X[3D] -> Y[2D], [3715]
Blickzentrum S[2D] )
2002 REM lokale Variablen: ms [1922]
2010 ms=EXP(-x3/300) [978]
2020 y1=ms*(x1-s1)+s1 [798]
2030 y2=ms*(x2-s2)+s2 [1297]
2040 RETURN [555]
2100 : [174]
2101 REM 3D verschieben (um Vektor v) [2792]
2102 REM lokale Variablen: - [2020]
2110 x1=x1+v1:x2=x2+v2:x3=x3+v3 [779]
2120 RETURN [555]
2200 : [174]
2201 REM 3D strecken (Zentrum Z, Streckfa [3120]
ktor m)
2202 REM lokale Variablen: - [2020]
2210 x1=m*(x1-z1)+z1 [1057]
2220 x2=m*(x2-z2)+z2 [1531]
2230 x3=m*(x3-z3)+z3 [451]
2240 RETURN [555]
2300 : [174]
2301 REM 3D spiegeln (Spiegelebene PQR) [3427]
2302 REM lokale Variablen: a1,a2,a3,b1,b2 [3886]
,b3,n0,n1,n2,n3
2310 a1=q1-p1:a2=q2-p2:a3=q3-p3 [1871]
2320 b1=r1-p1:b2=r2-p2:b3=r3-p3 [1221]
2330 n1=a2*b3-a3*b2:n2=a3*b1-a1*b3:n3=a1*b [1917]
2-a2*b1
2340 n0=SQR(n1^2+n2^2+n3^2):n1=n1/n0:n2=n2 [3571]
/n0:n3=n3/n0
2350 n0=2*(n1*(x1-p1)+n2*(x2-p2)+n3*(x3-p3 [2593]
))
2360 x1=x1-n1*n0:x2=x2-n2*n0:x3=x3-n3*n0 [2062]
2370 RETURN [555]
2400 : [174]
2401 REM 3D drehen (Drehachse LM, Winkel [3081]
a)
2402 REM lokale Variablen: a0,a1,a2,a3,b0 [5189]
,b1,b2,b3,n0,n1,n2,n3,sia,coa
2410 sia=SIN(a):coa=COS(a) [1094]
2420 a1=l1-m1:a2=l2-m2:a3=l3-m3:a0=SQR(a1^ [2895]
2+a2^2+a3^2)
2430 a1=a1/a0:a2=a2/a0:a3=a3/a0 [1553]
2440 b0=a1*(x1-m1)+a2*(x2-m2)+a3*(x3-m3) [1795]
2450 b1=a1*b0+m1:b2=a2*b0+m2:b3=a3*b0+m3 [1624]
2460 a1=a1*sia:a2=a2*sia:a3=a3*sia [1767]
2470 n1=x1-b1:n2=x2-b2:n3=x3-b3:n0=SQR(n1^ [2668]
2+n2^2+n3^2)
2480 IF n0=0 THEN RETURN [840]
2490 n1=ROUND(n1/n0,6):n2=ROUND(n2/n0,6):n [3858]
3=ROUND(n3/n0,6)
2500 IF n1<>0 THEN GOSUB 2600 [1091]
2510 IF n1=0 AND n2<>0 THEN GOSUB 2650 [2150]
2520 IF n1=0 AND n2=0 THEN GOSUB 2700 [1233]
2530 x1=x1*n0+b1:x2=x2*n0+b2:x3=x3*n0+b3 [2526]
2540 RETURN [555]

```

Listing Komplexe Grafiken

```

2600 x1=a2*n3-a3*n2+n1*coa [964]
2610 x2=(a3*n3^2+a2*n2*n3)/n1+n2*coa+a3*n1 [2332]
2620 x3=- (a2*n2^2+a3*n2*n3)/n1+n3*coa-a2*n [2306]
1
2630 RETURN [555]
2650 x1=- (a3*n3^2)/n2-a3*n2 [1225]
2660 x2=n2*coa-a1*n3 [1154]
2670 x3=a1*n2+n3*coa [889]
2680 RETURN [555]
2700 x1=a2:x2=-a1:x3=coa [1444]
2710 RETURN [555]

```

```

1 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1683]
2 REM % % [378]
3 REM % *** 2D-GRAFIKDEMO *** % [1227]
4 REM % % [378]
5 REM % September 1987 % [1417]
6 REM % % [378]
7 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1683]
8 : [174]
10 DEG:MODE 1:INK 0,0:INK 1,13:INK 2,6:INK [4224]
3,24:BORDER 0
20 WINDOW 1,40,1,1:PAPER 0:OPEN 1:CLS [2979]
30 ORIGIN 0,0,0,640,0,382:CLG 0 [1217]
40 RESTORE:FOR j=0 TO 5:READ x1(j),x2(j):N [3272]
EXT:GOSUB 850
50 v1=10:v2=5:PRINT"Verschieben" [2266]
60 anz=20:ab=1:GOSUB 500 [763]
70 v1=-150:v2=150:anz=1:GOSUB 500 [1569]
80 v1=300:v2=-80:GOSUB 500 [1737]
90 z1=380:z2=200:m=1.1:PLOT 380,200,3:PRIN [3708]
T"Strecken"
100 anz=15:ab=2:GOSUB 500 [1159]
110 m1=380:m2=200:a=-10:PRINT"Drehen" [2914]
120 anz=12:ab=4:GOSUB 500 [1567]
130 z1=30:z2=30:m=0.9 [1445]
140 PLOT 380,200,0:PLOT 50,50,3:PRINT"Stre [2836]
cken"
150 anz=12:ab=2:GOSUB 500 [763]
160 p1=0:p2=300:q1=500:q2=0 [691]
170 PLOT 50,50,0:MOVE 0,300:DRAW 500,0,3:P [2199]
RINT"Spiegeln"
180 anz=5:ab=3:GOSUB 500 [1684]
190 m1=x1(2):m2=x2(2):a=20 [1944]
200 MOVE 0,300:DRAW 500,0,0:PLOT m1,m2,3:P [2696]
RINT"Drehen"
210 anz=18:ab=4:GOSUB 500 [1359]
220 z1=250:z2=250:PLOT 250,250,3:PRINT"Str [6117]
ecken mit negativem Faktor"
230 anz=1:ab=2:m=-3:GOSUB 500 [1464]
240 m=-0.5:GOSUB 500:m=-1.5:GOSUB 500 [2507]
250 GOSUB 560:GOTO 10 [999]
260 : [174]
500 'Figur abbilden [1103]
510 FOR i=1 TO anz [856]
520 FOR j=0 TO 5:x1=x1(j):x2=x2(j) [1042]
530 ON ab GOSUB 1000,1100,1200,1300 [1416]
540 x1(j)=x1:x2(j)=x2:NEXT [1845]
550 GOSUB 800:NEXT [1258]
560 t=TIME+600:WHILE t>TIME:WEND [1872]
570 RETURN [555]
580 : [174]
800 'Figur loeschen [1213]
810 MOVE y1(2),y2(2):FOR j=0 TO 2:DRAW y1( [3400]
j),y2(j),0:NEXT
820 MOVE y1(3),y2(3):DRAW y1(4),y2(4),0:DR [3220]
AW y1(5),y2(5),0
850 'Figur darstellen [606]
860 MOVE x1(2),x2(2):FOR j=0 TO 2:DRAW x1( [4640]
j),x2(j),1:NEXT
870 MOVE x1(3),x2(3):DRAW x1(4),x2(4),2:DR [3096]
AW x1(5),x2(5),2
880 FOR j=0 TO 5:y1(j)=x1(j):y2(j)=x2(j):N [2207]
EXT
890 RETURN [555]
899 : [174]
900 'Daten [442]
910 DATA 0,0,60,20,40,50,30,20,40,20,40,35 [1792]
1000 : [174]
1010 REM GRAFIK.BAS mit MERGE nachladen ! [2682]

```

Listing Komplexe Grafiken

```

1 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1683]
2 REM % [378]
3 REM % *** 3D-GRAFIKDEMO *** % [1241]
4 REM % [378]
5 REM % September 1987 % [1417]
6 REM % [378]
7 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% [1683]
8 ; [174]
10 DEG:MODE 1:INK 0,0:INK 1,13:INK 2,6:INK [4224]
 3,24:BORDER 0
20 WINDOW 1,40,1,1:PAPER 0:PEN 1:CLS [2979]
30 ORIGIN 0,0,0,640,0,382:CLG 0 [1217]
35 s1=450:s2=300:PLOT 450,300,3 [1310]
40 RESTORE:FOR j=0 TO 7:READ x1(j),x2(j),x [3955]
3(j):NEXT:GOSUB 750
50 v1=0:v2=0:v3=100:PRINT"Verschieben in x [3317]
3-Richtung"
60 anz=5:ab=1:GOSUB 500:anz=6:v3=-100:GOSU [3504]
B 500
70 l1=250:l2=0:l3=0:m1=350:m2=300:m3=100:a [2106]
=45
80 x1=11:x2=12:x3=13:GOSUB 2000:MOVE y1,y2 [2881]
90 x1=m1:x2=m2:x3=m3:GOSUB 2000:DRAW y1,y2 [2959]
,3:PRINT"Drehen"
100 anz=8:ab=4:GOSUB 500 [1568]
110 p1=0:p2=0:p3=200:q1=640:q2=0:q3=200:r1 [3806]
=320:r2=400:r3=200
120 x1=11:x2=12:x3=13:GOSUB 2000:MOVE y1,y [2881]
2
130 x1=m1:x2=m2:x3=m3:GOSUB 2000:DRAW y1,y [3976]
2,0:PRINT"Spiegeln"
140 anz=3:ab=3:GOSUB 500 [1306]
150 p1=300:p2=0:p3=0:q1=300:q2=100:q3=0:r1 [3800]
=300:r2=0:r3=100
160 anz=1:GOSUB 500 [1046]
170 p1=0:p2=0:p3=200:q1=640:q2=0:q3=200:r1 [3806]
=320:r2=400:r3=200
180 anz=1:GOSUB 500 [1046]
190 z1=250:z2=150:z3=-50:m=0.8 [2444]
200 x1=z1:x2=z2:x3=z3:GOSUB 2000:PLOT y1,y [4988]
2,3:PRINT"Strecken"
210 anz=10:ab=2:GOSUB 500 [886]
220 x1=z1:x2=z2:x3=z3:GOSUB 2000:PLOT y1,y [2961]
2,0

```

Listing Komplexe Grafiken

```

230 v1=40:v2=30:v3=-120:PRINT"Verschieben" [2199]
240 anz=5:ab=1:GOSUB 500 [1398]
250 v1=-8:v2=-5:v3=0:GOSUB 500 [1289]
260 GOSUB 560:GOTO 10 [999]
270 : [174]
500 REM Abbildung [1293]
510 FOR i=1 TO anz [856]
520 FOR j=0 TO 7:x1=x1(j):x2=x2(j):x3=x3(j) [3076]
)
530 ON ab GOSUB 2100,2200,2300,2400 [2004]
540 x1(j)=x1:x2(j)=x2:x3(j)=x3:NEXT [2380]
550 GOSUB 700:NEXT [1133]
560 t=TIME+600:WHILE t>TIME:WEND [1872]
570 RETURN [555]
580 : [174]
700 'Figur loeschen [1213]
710 FOR j=0 TO 7:x1=y1(j):x2=y2(j):x3=y3(j) [5992]
):GOSUB 2000:a1(j)=y1:a2(j)=y2:NEXT
720 f=0:GOSUB 800 [1200]
750 'Figur darstellen [606]
760 FOR j=0 TO 7:x1=x1(j):x2=x2(j):x3=x3(j) [6329]
):GOSUB 2000:a1(j)=y1:a2(j)=y2:NEXT
770 f=1:GOSUB 800 [702]
780 FOR j=0 TO 7:y1(j)=x1(j):y2(j)=x2(j):y [3319]
3(j)=x3(j):NEXT
790 RETURN [555]
800 'Figur zeichnen [1091]
810 FOR j=0 TO 3 [705]
820 j1=(j+1) MOD 4 [1195]
830 MOVE a1(j1),a2(j1):DRAW a1(j),a2(j),f* [1928]
2
840 DRAW a1(j+4),a2(j+4),f:DRAW a1(j1+4),a [2079]
2(j1+4)
850 NEXT:RETURN [940]
860 : [174]
900 'Daten [442]
910 DATA 300,100,0,400,100,0,400,200,0,300 [2187]
,200,0
920 DATA 300,100,100,400,100,100,400,200,1 [2226]
00,300,200,100
1000 : [174]
1010 REM GRAFIK.BAS mit MERGE nachladen ! [2682]

```

Listing Komplexe Grafiken

Laufschrift – ganz einfach!

Wer kennt sie nicht, die Laufschriften, die bei so vielen Programmen das Tüpfelchen auf dem I bedeuten? Wäre das schön, seine eigenen Programme mit solch feinen Kleinheiten verschönern zu können. »Zu schwer zu programmieren«, hört man viele Computerbenutzer stöhnen. Stimmt das? Nicht, wenn sie diese kleine Routine benutzen.

Dieses Programm ermöglicht das Scrollen einer Bildschirmzeile, in der beliebige ASCII-Zeichen enthalten sein können, z.B. die Originalzeichen oder auch selbstdefinierte. SCROLL arbeitet in allen 3 Modi und in jeder der frei wählbaren 25 Bildschirmzeilen. Der Quellcode dieser Maschinenroutine wurde mit dem Hisoft-Assembler GENA 3.1 erstellt. Die Routine selbst stellt zwei neue RSX-Befehle zur Verfügung:

- 1) !SCROLL,(Klammeraffe)(String)
- 2) !SCROLLOFF

Man geht folgendermaßen vor:

Zuerst definiert man eine Stringvariable mit einem Textinhalt, z.B. a\$="Dies ist ein Test". Dahinter müssen Sie nun ein CHR\$(0) setzen, damit die Routine das Textende erkennt. Der fertige String sieht dann folgendermaßen aus:

a\$="Dies ist ein Test"+CHR\$(0)

Nach dem Sie den String definiert haben, können Sie die Laufschrift mit dem !SCROLL-Befehl aufrufen, achten Sie dabei aber darauf, daß vor dem String das Klammeraffen-Zeichen steht. Nach korrekter Eingabe erscheinen die Stringzeichen nacheinander rechts in der vom Maschinenprogramm definierten Zeile und scrollen nach links bis an den Anfang dieser Zeile durch, wo sie wieder »verschwinden«. Während des Scrollings ist der BASIC-Interpreter weiterhin aktiv, d.h. das Scrolling kann z.B. im Titelbild eines Spiels (BASIC/Maschinensprache) laufen, während zugleich die Titelmusik ertönt. Allerdings ist das Programm dann um ca. 38% langsamer, da der Interrupt natürlich immer die Befehlsabarbeitung unterbricht. Zwei Dinge sollten Sie beim Gebrauch der Routine nicht tun, nämlich:

- a) Zeichen in der laufenden Zeile ausdrucken,
- b) Den MODE wechseln.

Beides führt dazu, daß die Zeichen in ihrer ursprünglichen Form zerstört werden. Allerdings ist es möglich, ein oder mehrere Windows zu erstellen, die sich nicht mit der scrollenden Zeile überlappen. Diese Windows können dann wie gewohnt bearbeitet werden, auch mit dem Befehl CLS. Disketten- oder Cassetteoperationen sind ebenfalls während des Scrollens zu vermeiden, da die Interrupts währenddessen abgeschaltet werden.

!SCROLLOFF schaltet das Scrolling wieder ab, jedoch bleibt die Zeile bestehen und wird nicht gelöscht.

Die Assembler-Routine

Das Assembler-Programm übergibt dem Betriebssystem zunächst die Namen und die Adressen der RSX-Befehle und kehrt daraufhin zum BASIC zurück. Die Routine für !SCROLL (ab Zeile 240) besorgt sich erstmal den momentan aktiven MODE und berechnet daraus die Zahl der Bytes-1, die eine Zeile eines ASCII-Zeichens im Bildschirmspeicher einnimmt. Diese Zahl ist ja bekanntlich vom MODE abhängig und ist für

MODE 0 : 4-1 =3

MODE 1 : 2-1 =1

MODE 2 : 1-1 =0

Diese Zahl wird noch gebraucht und wird nach BYTMOD (Z.1860) gespeichert. Dann wird SCROLF auf 0 gesetzt (Z.350). Dieses Flag zeigt an, welcher Teil eines Zeichens als nächstes ausgegeben wird. Daraufhin wird eine MODE-abhängige Bitmaske errechnet, mit der aus einer Zeichenmatrix, wie sie im LOWER ROM ab &3800 steht, die Pixel für den Bildschirmspeicher berechnet werden können.

Ein Beispiel: Die erste Zeile der Matrix von »K« sieht so aus: &X11100110

Im MODE 2, wo eine Zeichenzeile ein Byte im Bildschirmspeicher belegt (s.o.), sind keinerlei Veränderungen nötig (Z.370-390): POKE &C000,&X11100110 bringt die erste Zeile von »K« in die linke obere Speicherstelle vom Bildschirmspeicher.

Im MODE 1 wird es etwas komplizierter. Die Maske ist &F0 (Z.400). Eine Zeichenzeile benötigt 2 Bytes (s.o.): 1 POKE &C000,&X11100110 AND &F0 ergibt den ersten Teil von »K« 2 POKE &C000,&X01101110 AND &F0 ergibt den zweiten Teil. Die Matrix von »K« wurde 1mal um 4 Bit (4=2*2), Z.1450 und 1580/1590) verschoben:

Ausgangsposition:

&X11100110

Verschiebung 1. &X11001101

2. &X10011011

3. &X00110111

4. &X01101110

Und MODE 0 macht's noch difiziler: Die Maske ist &c0 (Z.440/450), eine Zeichenzeile benötigt 4 Bytes (s.o.): Ausgangsposition:

1. POKE &C000,&X11100110 AND &C0

2. POKE &C001,&X10011011 AND &C0

3. POKE &C002,&X01101110 AND &C0

4. POKE &C003,&X10111001 AND &C0

Die Matrix wurde dreimal um 2 Bit (2=2*1, Z.1430 und 1580/1590) verschoben. Ab Z.490 wird dann der übergebene Stringdeskriptor geholt und daraus die Adresse der Stringzeichen berechnet. Dann wird ein Framefly-Interrupt einge-

hängt, der das eigentliche Scrolling übernimmt. Immer wenn der Elektronenstrahl des Monitors aus der rechten unteren Ecke in die linke obere »zurückspringt« (jede 50stel Sekunde) ohne etwas zu »zeichnen«, wird dieser Interrupt aufgerufen. Dadurch wird erreicht, daß das Scrolling flackerfrei abläuft. Daraufhin kehrt das Programm nach BASIC zurück. Die Interruptroutine beginnt ab Z.750. Zunächst wird das obere ROM gesperrt (Z.750-780), um den »ungestörten« Zugriff auf den Bildschirmspeicher zu ermöglichen. Dann wird der Zeiger auf die Stringzeichen geholt (Z.790). Als nächstes wird die Zeile gescrollt. Dazu werden einfach alle Bytes der Zeile um ein Byte nach links verschoben (Z.820-950).

Danach wird SCROLF um 1 verringert und getestet, ob es dann kleiner als 0 ist (Z.970-1030). Ist das der Fall, bedeutet das, daß ein neues Zeichen des Strings rechts hereingescrollt werden muß; andernfalls wird nur das nächste Stück des momentan selektierten Zeichens hereingescrollt. Soll ein neues Zeichen hereingescrollt werden, so wird es erstmal gelesen (Z.1120). Daraufhin wird mit Hilfe einer Firmwareroutine berechnet, wo die Matrix dieses Zeichens zu finden ist (Z.1250). Diese Routine findet selbstständig heraus, ob ein Zeichen vom User definiert worden ist oder ob noch die »normale« Version vorhanden ist. Jetzt wird berechnet, wie oft die oben gefundene Matrix gescrollt werden muß, damit das gewünschte Stück »vor die Maske kommt« (Z.1340-1460).

Anschließend wird eines der 8 (Z.1470) Matrixstücke geholt (Z.1500), gescrollt (Z.1580 – 1600), mit der MODE-abhängigen Bitmaske verknüpft (Z.1610 – 1650) und anschließend das so »bearbeitete« Stück auf den Bildschirm gebracht (Z.1660). Nun wird die Adresse des nächsten Matrixstücks berechnet (Z.1730) und die Bildschirmspeicherstelle dafür (Z.1670 – 1720), woraufhin das nächste der 8 Matrixstücke »behandelt« wird. Ist das geschehen, wird noch der Zeiger auf den String wieder abgespeichert, denn er zeigt ja vielleicht jetzt auf das Zeichen (Z.1770 – 1820). Und schon geht's wieder zurück in die unterbrochene Routine (Z.1830), bis eine weitere 50stel Sekunde verstrichen ist und der Interrupt erneut aufgerufen wird. Die Routine für !SCROLLOFF (Z. 720 – 730) ist etwas einfacher: Hier wird einfach der Interrupt wieder ausgehängt. Das Programm ist »von Haus aus« auf die vorletzte, die 24. Zeile als Scrollzeile eingestellt. Assemblerbesitzer können dies im Listing natürlich ändern (Z.70). Es muß das zweite Byte der entsprechenden Textzeile angegeben werden, z.B. #C001 für die erste Zeile. Sollten die 254 Textbytes, die der String der Scrollroutine maximal übergeben kann, nicht mehr ausreichen, kann das Assemblerprogramm natürlich so modifiziert werden, daß es Text von einer beliebigen Stelle im Speicher lesen kann, an der dann auch mehr als die 254 Bytes stehen können. In diesem Fall könnte der Text auch unterhalb von &4000 stehen, da das Programm die Zeichen grundsätzlich aus dem RAM liest.

Hinweise zum BASIC-Listing

Das BASIC-Listing speichert auf Diskette (Kassette) ein Binärfile »SCROLL.BIN« ab und initialisiert dann die neuen RSX-Befehle. Das Binärprogramm kann dann immer nach &A400 geladen und mit »CALL &A400« aufgerufen werden. Es sollte allerdings nur einmal initialisiert werden, da an der Einsprungsadresse nach dem ersten Aufruf die 4 Bytes für das Betriebssystem stehen.

(S. Gründel/JB)

für 464-664-6128



```

10 FOR i=&A400 TO &A512:READ a$:POKE i,VAL [2534]
(" "+a$):NEXT i
20 SAVE "SCROLL.BIN",b,&A400,&113:END [2087]
30 DATA 21,00,A4,01,09,A4,C3,D1,BC,11 [2313]
40 DATA A4,C3,21,A4,C3,6F,A4,53,43,52 [1876]
50 DATA 4F,4C,CC,53,43,52,4F,4C,4C,4F [1657]
60 DATA 46,C6,00,CD,11,BC,47,3E,02,4F [2152]
70 DATA 90,B9,20,01,3C,21,10,A5,77,23 [1096]
80 DATA 36,00,06,FF,B7,28,0B,06,FO,CB [1098]
90 DATA 3F,B7,28,04,CB,20,CB,20,21,0F [2028]
100 DATA A5,70,DD,0E,00,DD,66,01,23,5E [2066]
110 DATA 23,56,21,15,A5,73,23,72,21,13 [1446]
120 DATA A5,73,23,72,21,18,A5,06,9F,11 [1376]
130 DATA 75,A4,CD,EF,BC,21,19,A5,C3,DA [2111]
140 DATA BC,21,19,A5,C3,DD,BC,D9,CB,D9 [1434]
150 DATA ED,49,D9,2A,13,A5,E5,06,08,21 [1049]
160 DATA 31,C7,C5,E5,E5,D1,1B,01,4F,00 [1915]
170 DATA ED,B0,E1,11,00,08,19,C1,10,EE [2021]
180 DATA 3A,11,A5,3D,32,11,A5,D1,D5,FE [1425]
190 DATA FF,20,21,21,10,A5,7E,23,77,13 [1139]
200 DATA EB,E3,E7,B7,20,0A,21,15,A5,5E [1966]
210 DATA 23,56,EB,E7,23,E3,CD,A5,BB,EB [1716]
220 DATA 21,17,A5,73,23,72,2A,17,A5,11 [1563]
230 DATA 7F,C7,3A,11,A5,47,3A,10,A5,90 [2510]
240 DATA 28,0D,47,3A,10,A5,FE,01,28,03 [1965]
250 DATA 78,18,02,3E,02,4F,06,08,C5,46 [1689]
260 DATA 79,B7,78,41,28,07,B7,28,0C,07 [829]
270 DATA 07,10,FC,E5,21,0F,A5,46,A0,E1 [1669]
280 DATA 12,EB,D5,11,00,08,19,D1,EB,23 [1692]
290 DATA C1,10,DD,D1,21,13,A5,73,23,72 [2066]
300 DATA C9,00,00,00,00 [1162]

```

```

10 GETMAT: EQU #BBA5 ;Adresse einer Zeichenmatrix holen
20 GETMOD: EQU #BC11 ;aktuellen Bildschirmmodus holen
30 LOGEXT: EQU #BCD1 ;RSX-Erweiterung einbinden
40 ADDRFR: EQU #BCDA ;Framefly-Interrupt übergeben
50 DELFRF: EQU #BCDD ;Framefly-Interrupt löschen
60 INITEV: EQU #BCEF ;Ereignisblock initialisieren
70 SCRADR: EQU #C731 ;zweites Byte der 24. Zeile
80
90 ORG #A400
100 EMT $
110 LD HL,$ ;4 Byte Platz fürs Kernel
120 LD BC,RSXTAB ;RSX-Kommandotabelle
130 JP LOGEXT ;Erweiterung einbinden und Rückkehr
140
150 RSXTAB: DEFW RSXNAM ;Adresse der Befehlsnamen-Tabelle
160 JP SCROLL ;Sprung für ersten RSX: SCROLL
170 JP SCROLOF ;Sprung für zweiten RSX: SCROLOFF
180 RSXNAM: DEFM "SCROL" ;Namen der
190 DEFB "L"##80 ;RSX-Befehle mit
200 DEFM "SCROLOFF" ;gesetzten 7. Bits
210 DEFB "F"##80 ;der letzten Bytes
220 DEFB #00 ;Ende der Namenstabelle
230
240 SCROLL: CALL GETMOD ;aktuellen Bildschirmmodus holen
250 LD B,A ;B-Register enthält MODE
260 LD A,#02 ;jetzt wird ermittelt,
270 LD C,A ;wieviele Bytes-1
280 SUB B ;ein Zeichen
290 CP C ;in diesem MODE im
300 JR NZ,SETFLG ;Bildschirmspeicher
310 INC A ;einnimmt
320 SETFLG: LD HL,BYTMOD ;Breite eines Zeichens in Bytes-1
330 LD (HL),A ;Zwischenspeichern
340 INC HL ;Scrollflag
350 LD (HL),#00 ;setzen
360
370 LD B,#FF ;Bitmaske für MODE 2
380 OR A ;MODE 2 aktiv?
390 JR Z,SETMSK ;ja -> Maske fertig
400 LD B,#F0 ;Bitmaske für MODE 1
410 SRL A ;Akku = Akku / 2
420 OR A ;MODE 1 aktiv?
430 JR Z,SETMSK ;ja -> Maske fertig
440 SLA B ;aus #F0 (Maske für MODE 1)
450 SLA B ;wird #C0 (Maske für MODE 0)
460 SETMSK: LD HL,BITMSK ;MODEabhngige
470 LD (HL),B ;Bitmaske setzen
480
490 LD L,(IX+0) ;Lo-Byte des Stringdescriptors
500 LD H,(IX+1) ;Hi-Byte des Stringdescriptors
510 INC HL ;HL zeigt auf Adresse des Strings
520 LD E,(HL) ;DE-Register
530 INC HL ;zeigt
540 LD D,(HL) ;auf String
550 LD HL,SSTART ;String
560 LD (HL),E ;anfang
570 INC HL ;ab-
580 LD (HL),D ;speichern

```

Listing Laufschrift

```

590 LD HL,STRPNT ;Stringzeiger
600 LD (HL),E ;als Zeiger auf
610 INC HL ;auszugebendes String-
620 LD (HL),D ;Zeichen abspeichern
630
640 LD HL,EBLOCK ;Adresse des Ereignis-Blocks
650 LD B,%10011111 ;Ereignisklasse
660 LD DE,SCROLR ;Adresse der SCROLL-Routine
670 CALL INITEV ;Ereignis-Block initialisieren
680
690 LD HL,FBLOCK ;Adresse des Framefly-Blocks
700 JP ADDRFR ;Framefly-Interrupt übergeben und Ende
710
720 SCROLOF: LD HL,FBLOCK ;Framefly-Block
730 JP DELFRF ;Löschen und Rückkehr
740
750 SCROLR: EXX ;zweiten Registersatz auswhlen
760 SET J,C ;oberes ROM sperren
770 OUT (C),C ;neuen ROM STATE übergeben
780 EXX ;wieder normalen Registersatz auswhlen
790 LD HL,(STRPNT) ;Stringzeiger holen
800 PUSH HL ;Stringzeiger retten
810
820 LD B,#08 ;8 zu verschiebende Bildschirmzeilen
830 LD HL,SCRADR ;Zeiger auf Bildschirmposition
840 LDIRLP: PUSH BC ;Bildschirmzeilenzeiger retten
850 PUSH HL ;Zeiger auf Bildschirmposition retten
860 PUSH HL ;Bildschirmposition
870 POP DE ;im DE-Register
880 DEC DE ;Ziel-Bildschirmposition
890 LD BC,#004F ;zu verschiebende Bytes
900
910 LDIR ;scrollen-
920 POP HL ;Bildschirmposition holen
930 LD DE,#0800 ;Offset
940 ADD HL,DE ;neue Bildschirmposition (nächste Zeile)
950 POP BC ;Bildschirmzeilenzeiger holen
960 DJNZ LDIRLP ;nächste Zeile scrollen
970
980 LD A,(SCROLF) ;Scrollflag holen
990 DEC A ;Scrollflag verringern
1000 LD (SCROLF),A ;und wieder abspeichern
1010 POP DE ;Stringzeiger holen
1020 PUSH DE ;und wieder abspeichern
1030 CP #FF ;wird n[achstes Zeichen hereingescrollt?
1040 JR NZ,WRCHRR ;nein, dann Zeichen ausgeben
1050
1060 LD HL,BYTMOD ;Breite eines Zeichens in Bytes-1
1070 LD A,(HL) ;holen
1080 INC HL ;Scrollflag
1090 LD (HL),A ;wieder setzen
1100 INC DE ;Stringzeiger auf n[chstes Zeichen
1110 EX DE,HL ;Stringzeiger nach HL
1120 EX (SP),HL ;und auf Stack retten
1130 RST #0020 ;Stringzeichen holen
1140 OR A ;Stringende erreicht?
1150 JR NZ,GETMTR ;nein, dann Zeichenmatrix holen
1160
1170 LD HL,SSTART ;Zeiger auf Stringanfang
1180 LD E,(HL) ;Stringanfängerzeiger
1190 INC HL ;wieder
1200 LD D,(HL) ;holen
1210 EX DE,HL ;Stringzeiger nach HL
1220 RST #0020 ;Stringzeichen holen
1230 INC HL ;Stringzeiger auf n[chstes Zeichen
1240 EX (SP),HL ;Stringzeiger retten
1250
1260 GETMTR: CALL GETMAT ;Matrix des Zeichens errechnen
1270 EX DE,HL ;gefundenen Matrixadresse nach DE
1280 LD HL,MATPNT ;Matrixzeigerspeicher
1290 LD (HL),E ;Matrix-
1300 INC HL ;zeiger
1310 LD (HL),D ;abspeichern
1320
1330 WRCHRR: LD HL,(MATPNT) ;Zeichenmatrixzeiger holen
1340 LD DE,SCRADR+004E ;Byte links vom rechten Zeilenende
1350 LD A,(SCROLF) ;Scrollflag holen
1360 LD B,A ;und ins B-Register bringen
1370 LD A,(BYTMOD) ;Breite eines Zeichens in Bytes-1
1380 SUB B ;BYTMOD-SCROLF im Akku
1390 JR Z,MATEND ;keine Matrixverschiebung
1400 LD A,(BYTMOD) ;BYTMOD-SCROLF ins B-Register bringen
1410 CP #01 ;Breite eines Zeichens in Bytes-1
1420 JR Z,MODE_1 ;MODE 1?
1430 LD A,B ;ja -> Sprung
1440 LD A,B ;BYTMOD-SCROLF im Akku
1450 LD A,#02 ;Matrixverschiebung beenden
1460 MATEND: LD C,A ;Matrixverschiebung im B-Register
1470 LD B,#08 ;8 Zeichenbreite sind darzustellen
1480
1490 PIECL: PUSH BC ;Stringzeiger/Matrixversch. retten
1500 LD B,(HL) ;Matrix[ck] holen
1510 LD A,C ;Matrixverschiebung holen
1520 OR A ;Matrixverschiebung = 0?
1530 LD A,B ;Matrix[ck] im Akku
1540 LD B,C ;Matrixverschiebung im B-Register
1550 JR Z,VEREND ;ja -> Verschiebung beenden
1560 OR A ;Matrix[ck] unsichtbar?
1570 JR Z,NOPIEC ;ja -> Darstellung überspringen
1580 VERSCH: RLCA ;Matrix[ck]
1590 RLCA ;verschieben
1600 DJNZ VERSCH ;nocheinmal?
1610 VEREND: PUSH HL ;Zeichenmatrixzeiger retten
1620 LD HL,BITMSK ;Bitmaskenzeiger
1630 LD B,(HL) ;Bitmaske holen
1640 AND B ;Matrix[ck] für Bildschirm aufbereiten
1650 POP HL ;Zeichenmatrixzeiger wieder holen
1660 LD (DE),A ;und auf den Bildschirm bringen
1670 NOPIEC: EX DE,HL ;HL: Bildschirmadr., / DE: Zeichenmatrz.
1680 PUSH DE ;Zeichenmatrixzeiger retten
1690 LD DE,#0800 ;Offset für nächste Bildschirmzeile
1700 ADD HL,DE ;neue Bildschirmzieladresse errechnen
1710 POP DE ;Zeichenmatrixzeiger wieder holen
1720 EX DE,HL ;HL: Zeichenmatrz. / DE: Bildschirmadr.
1730 INC HL ;Zeichenmatrz. zeigt auf n[chstes Stück
1740 POP BC ;Schleifenzeiger holen
1750 DJNZ PIECL ;n[chstes Zeichen[ck] ausgeben
1760
1770 POP DE ;Stringzeiger wieder nach DE holen
1780
1790 LD HL,STRPNT ;Stringzeiger
1800 LD (HL),E ;wieder
1810 INC HL ;ab-
1820 LD (HL),D ;speichern
1830 RLCA ;Rückkehr
1840
1850 BITMSK: DEFS #01 ;ein Byte für MODEabhngige Bitmaske
1860 BYTMOD: DEFS #01 ;Breite eines Zeichens in Bytes-1
1870 SCROLOF: DEFS #01 ;Verschiebe-Merkflag Scrollflag
1880 DEFB #00 ;Hi-Byte von SCROLOF als 16-Bit-Wert
1890
1900 STRPNT: DEFS #02 ;zwei Bytes für den Stringzeiger
1910 SSTART: DEFS #02 ;zwei Bytes für den Stringanfängerzeiger
1920 MATPNT: DEFS #02 ;zwei Bytes für den Stringzeiger
1930
1940 FBLOCK: DEFS #02 ;zwei Byte Framefly-Block und
1950 EBLOCK: DEFS #07 ;sieben Byte Ereignis-Block

```

Listing Laufschrift

Fehlermeldungen im Griff!

Wie oft hat man sich nicht schon über die Meldungen **OVERFLOW** und **DIVISION BY ZERO** geärgert. In Extremfällen und bei Rechenungenauigkeiten mit sehr großen oder kleinen Zahlen kommen sie plötzlich und unerwartet. Dort ist die Meldung aber völlig überflüssig; sie stört nur die Grafik und den Bildschirmaufbau.

Zur Vermeidung dieser Meldungen gibt es zwei Varianten, und jede hat ihre Vor- und Nachteile: Wenn der Gefahrenherd im Programm genau bekannt ist, und der Fehler nur an einer bestimmten Stelle auftritt, kann man ihn ohne großen Aufwand unsichtbar machen.

FÜR 464-664-6128



```

10 REM ***** [1992]
*****
20 REM * Overflow - Division by zero - Unt [2767]
erdruecker *
30 REM * by Dieter Braun Feb. 198 [2901]
7 *
40 REM ***** [1992]
*****
50 RESTORE:p=PEEK(6):IF p<>128 THEN adr=&B [4397]
0*256+&A5 ELSE adr=&B0*256+&C7
60 ad1=&BD61:DATA &ef,&15,&34,&ef,&9e,&34 [2703]
70 DATA &cd,x1,&37,&c9,&cd,x1,&37,&c9 [1387]
80 a=adr:e=adr+16:s=0 [2631]
90 READ a$ [309]
100 IF a$="x1" THEN POKE a,adr-INT(adr/256 [4785]
)*256:POKE a+1,INT(adr/256):GOTO 140
120 b=VAL(a$):POKE a,b:a=a+1:s=s+b [2241]

```

Listing: Fehlermeldungen

Vor dem gefährdeten Term wird der Textbildschirm mit einem **PRINT CHR\$(21)** abgeschaltet und die Fehlermeldung geht ins Leere. Durch **PRINT CHR\$(6)** wird der Textbildschirm wieder aktiviert und die Zeichen werden wieder auf dem Bildschirm dargestellt.

Die schnelle Dreiecksroutine in diesem Heft (s.u.) bedient sich auch dieses Tricks, um einem **DIVISION BY ZERO** zu entgehen. Für größere Programme, bei denen dieser Fehler öfters auftritt, ist das nachstehende Programm gedacht: Es bindet eine kleine MC-Routine in den CPC ein. Sie sitzt in einem kleinen Interpreterspeicher und erzielt Ihre Wirkung durch zurücksetzen der **CARRY-Flagge** nach verlassen der Multiplikations- und Divisionsroutine.

(D. Braun/JB)

```

130 IF a>=e THEN 170 ELSE 90 [1570]
140 a=a+2:adr=adr+3:GOTO 130 [1396]
160 REM installieren [1466]
170 h1=INT(adr/256):l1=adr-h1*256:adr=adr+ [5365]
5:h2=INT(adr/256):l2=adr-h2*256
180 POKE ad1,&C3:POKE ad1+1,l1:POKE ad1+2, [2663]
h1
190 POKE ad1+3,&C3:POKE ad1+4,l2:POKE ad1+ [2261]
5,h2
200 PRINT "unterdruecker ist installiert" [3890]
210 PRINT:PRINT"der quotient -10/0 ist nun [3919]
";-10/0;" !!!"
220 PRINT:PRINT:END [1018]
230 ' [117]
240 ' fuer cpc 664 aendern : zeile 60 [2021]
250 ' 60 ad1=&BD82:DATA &ef,&77,&35,&ef,&0 [2061]
4,&36
260 ' [117]
270 ' fuer cpc 6128 aendern : zeile 60 [2344]

280 ' 60 ad1=&BD85:DATA &ef,&77,&35,&ef,&0 [2722]
4,&36

```

Listing: Fehlermeldungen

Schnelle Dreiecke

Zur Darstellung von ausgefüllten Figuren auf dem Bildschirm haben die CPC's keinen Befehl parat. In Malprogrammen oder Animatoren sollte aber eine Fläche so schnell wie möglich auf dem Bildschirm sein.

Da man alle n-Ecke aus Dreiecken aufbauen kann, ist man auf der Suche nach einer besonders schnellen Dreiecksroutine. Eine solch schnelle Routine ist aber ohne Probleme auch in BASIC zu realisieren, die nachfolgende Routine dürfte durch einen Compiler oder durch Maschinensprache kaum schneller werden. Hier die Erklärung zu den Funktionen der einzelnen Zeilen:

Die Zeilen 10 und 20 zeichnen beliebige Dreiecke nacheinander auf den Bildschirm, und durch Tastendruck wird das nächste Dreieck gezeigt. Die Zeilen 60 bis 160 beinhalten die eigentliche Routine als Unterprogramm. Die Koordinaten der drei Eckpunkte müssen in $x(1-3)/y(1-3)$ übergeben werden. In der Zeile 60 werden die Y-Koordinaten zu geraden Zahlen gemacht. Dadurch werden Rundungsfehler vermieden. Im weiteren wird das Unterprogramm ab Zeile 150 wichtig. Dieses gibt die unterste Ecke des Dreiecks in $y/x(z)$ zurück. Die herausgesuchte Ecke wird so hoch

gemacht ($y(z)=2E+21$), daß sie von dem Minimalzähler m nicht mehr erfaßt wird, sie wird dann als schon erfaßt gewertet und nicht mehr bemerkt. In den Zeilen 70 und Anfang Zeile 80 werden die Eck-Koordinaten nach $x1/y1,x2/y2,x3/y3$ gebracht. Durch die Unterroutine ab 150 wurden sie jedoch von unten nach oben sortiert ($x1/y1$: Ecke unten). Durch das **PRINT CHR\$(21)** wird der Bildschirm ausgeschaltet. Der Grund dafür ist, daß die folgenden Steigungsberechnungen zu einem 'DIVISION BY ZERO' führen könnten. Diese Fehlermeldung wird durch das **CHR\$(21)** aber nicht angezeigt. (Durch **CHR\$(6)** wird dieses wieder rückgängig gemacht).

Die einzelnen Steigungen der Grenzlinien des Dreiecks werden in f 1 bis 3 abgelegt. Ist der y-Wert des einen Punktes gleich dem y-Wertes des anderen, so wird durch 0 geteilt, und das Ergebnis ist dann eine sehr große Zahl ('unendlich'). Zwei Schleifen stehen in den Zeilen 110 bis 130. Sie malen letztendlich das Dreieck zeilenweise auf den Bildschirm.

$x1$ und $x4$ sind die X- Werte der jeweiligen Zeilenanfänge und Zeilenenden. Mit dem **RETURN** in Zeile 140 hat die Routine ihren Dienst getan. Ich hoffe, daß das Programm hilft, Ihre Geschwindigkeitsprobleme zu lösen.

(D. Braun/JB)

FÜR 464-664-6128



```

10 MODE 2:FOR n=1 TO 3:x(n)=INT(RND*640)+1:y(n)=IN [4785]
T(RND*400)+1:NEXT
20 GOSUB 60:CALL &BB18:GOTO 10
30 REM
40 REM Schnelle Dreiecksroutine
50 REM
60 FOR n=1 TO 3:y(n)=INT(y(n)/2)*2:NEXT
70 GOSUB 150:y1=y:x1=x(z):GOSUB 150:y2=y:x2=x(z):L

```

Listing: Schnelle Dreiecke

```

OCATE 1,1
80 GOSUB 150:y3=y:x3=x(z):PRINT CHR$(21)::f1=((x2- [4785]
x1)/(y2-y1))*2
90 f2=((x3-x1)/(y3-y1))*2:f3=((x3-x2)/(y3-y2))*2:P [4785]
RINT CHR$(6);
100
110 x4=x1:FOR n=y1 TO y2-2 STEP 2:PLOT x1,n:DRAW x [4785]
4,n
120 x1=x1+f1:x4=x4+f2:NEXT:x1=x2:FOR n=y2 TO y3 ST [4785]
EP 2
130 PLOT x1,n:DRAW x4,n:x1=x1+f3:x4=x4+f2:NEXT [4785]
140 RETURN
150 m=2E+20:FOR n=1 TO 3:IF y(n)<m THEN z=n:m=y(n) [4785]
160 NEXT:y=y(z):y(z)=2E+21:RETURN
Listing: Schnelle Dreiecke

```

Rasante Textausgabe mit FASTSCREEN

Möchten Sie, daß Ihre Texte in schnellerer Geschwindigkeit als üblich auf dem Bildschirm dargestellt werden? Dann ist FAST SCREEN genau das Richtige für Sie.

FASTSCREEN (im Folgenden mit FSC abgekürzt) beschleunigt die Textausgabe in den MODES 0 – 2, wobei in MODE 2 die Geschwindigkeit um über 100 % zunimmt. FSC besteht aus einem Startprogramm namens FSC.BAS, welches das Hauptprogramm FSC.BIN startet und den RSX-Befehl »FSC« erklärt. Mit dem Data-Lader FSC-LAD.BAS wird das Binär-File FSC.BIN erzeugt und auf Diskette abgespeichert.

Arbeiten mit FASTSCREEN

FSC.BIN kann an eine beliebige Speicherstelle geladen werden, da eine Transferoutine am Anfang des MC-Programmes den FSC-Patch nach &BE00 verschiebt. Die Routine liegt im Speicherbereich von &BE00 – &BE3F und &BE80 – &BEC5. Diese Speicherbereiche werden normalerweise nicht benutzt. Deshalb kann nach dem Rücksetzen des Rechners die FSC-Rou-

tine mit CALL &BE00 wieder aktiviert werden. Somit ist FSC bis zum Ausschalten des Computers vorhanden. Nach dem Transfer werden die Original RAM- und ROM-Adressen in die FSC-Routine eingetragen. Somit ist es möglich, diese Routine auf allen CPC's zu benutzen. Der RSX-Befehl »FSC« wird erzeugt, mit !FSC wird FASTSCREEN eingeschaltet, und mit !FSC,0 kann wieder auf die normale Textausgabe zurückgeschaltet werden.

Für den MODE 2 steht eine eigene Routine zur Verfügung. Hier gibt es nur eine PAPER- und eine PEN-Farbe. Auch gibt es im MODE 2 nur ein Zeichenmatrixbyte, während in den anderen MODES mehrere Bytes zum Bildschirm übertragen werden müssen. Durch diese Maßnahme erreicht die Routine eine besonders hohe Geschwindigkeit bei der Textausgabe.

Mit dem Programm FSC-TEST.BAS ist ein Zeittest für diese neue Routine möglich. Der Bildschirm wird in allen drei MODES komplett beschrieben, zuerst normal und dann mit FSC. Am Ende erscheint eine Tabelle mit den benötigten Zeiten der einzelnen Bildausgaben.

(E. Vosseler/JB)

Für 464-664-6128



```

10 '***** [2632]
*****
20 '* [175]
*
30 '* FSC-LAD.BAS Data-Lader FAST SCREE [4519]
N Vers 1.0 (c) E. Vosseler 7.87 *
40 '* [175]
*
50 '***** [2632]
*****
60 ' [117]
70 CLS:PRINT"Diskette einlegen und Taste d [4933]
ruecken":PRINT:CALL &BB06
80 GOSUB 120:IF f=1 THEN PRINT"Datei nicht [3272]
gespeichert!!!!":GOTO 110
90 PRINT"Datei FSC.BIN wird erzeugt." [4450]
100 SAVE"fsc.bin",b,&A000,&DD [1444]
110 PRINT:END [738]
120 MEMORY &9FFF:PRINT:PRINT"MC-Programm w [3533]
ird erzeugt.":PRINT
130 d=0:s=0:z=190:f=0:FOR a=&A000 TO &A0DD [3310]
140 READ b$:b=VAL("&"+b$):d=d+1:s=s+b:POKE [3288]
a,b
150 IF d<20 AND a<&A0DD THEN 180 [1231]
160 READ c$:IF VAL("&"+c$)<>s THEN f=1:PRI [2677]
NT"Fehler in Zeile :";z
170 z=z+10:s=0:d=0 [1596]
180 NEXT a:PRINT:RETURN [662]
190 DATA C0,21,5B,00,19,01,40,00,11,00,BE, [2253]
ED,B0,01,46,00,11,80,BE,ED,0685
200 DATA B0,2A,D4,BD,22,12,BE,22,20,BE,3E, [3650]
4A,BD,28,08,21,2F,B7,11,38,0722
210 DATA B7,18,06,21,8F,B2,11,98,B2,22,92, [2983]
BE,23,22,32,BE,23,22,1A,BE,0756
220 DATA ED,53,8B,BE,2A,54,BC,CB,BC,22,8F, [5140]
BE,2A,A6,BB,CB,BC,22,24,BE,OACF
230 DATA 2A,1B,BC,CB,BC,22,28,BE,C3,00,BE, [3130]
B7,01,BD,BE,21,C2,BE,CD,D1,0A83
240 DATA BC,AF,BB,21,18,BE,20,03,21,00,00, [3316]
22,D4,BD,C9,4F,3A,00,00,FE,0764
250 DATA 9C,79,D2,00,00,EB,CD,00,00,EB,CD, [4375]
Listing Fastscreen

```

```

00,00,48,06,08,0D,C2,80,BE,07BA
260 DATA 3A,00,00,4F,1A,A9,77,7C,C6,08,67, [3726]
13,10,F6,C9,F3,0C,D9,C5,D5,08C8
270 DATA E5,D9,D5,D9,E1,11,00,00,D5,CD,00, [3396]
00,2A,00,00,D9,D1,79,08,E5,093A
280 DATA 1A,D9,4F,2F,A4,47,79,A5,B0,D9,77, [3178]
13,23,0D,20,F0,08,4F,E1,7C,0881
290 DATA C6,08,67,10,E4,D9,E1,D1,C1,D9,FB, [3754]
C9,46,53,C3,00,B9,BE,C3,0A,0BB2
300 DATA BE,00,00BE [389]
*
10 '***** [2632]
*****
20 '* [175]
*
30 '* FSC.BAS Startprogramm FAST SCREE [4385]
N Vers 1.0 (C) E.Vosseler 7.87 *
40 '* [175]
*
50 '***** [2632]
*****
60 ' [117]
70 MODE 1:m=HIMEM:MEMORY &9FFF:LOCATE 3,1: [5292]
PRINT"**** F A S T - S C R E E N ****"
80 LOCATE 4,4:PRINT"Schnelle Textausgabe i [4321]
n MODE 0-2"
90 LOCATE 5,8:PRINT"|FSC Fast Mode ein [3129]
schalten."
100 LOCATE 5,10:PRINT"|FSC,0 Fast Mode a [3364]
usschalten."
110 LOCATE 3,16:PRINT"Nach Ruecksetzen mit [5942]
CTRL+SHIFT+ESC"
120 LOCATE 6,20:PRINT"CALL &BE00 aktiviert [2744]
FSC neu."
130 LOCATE 1,24 [728]
140 LOAD"fsc.bin",&A000:CALL &A000:CLOSEIN [3358]
:MEMORY m:END
*
10 '***** [2632]
*****
15 '* [175]
*
20 '* FSC-TEST.BAS Zeittest fuer FA [4506]
ST SCREEN (C) E.Vosseler 7.87 *
30 '* [175]
Listing Fastscreen

```

```

35 '***** [2632]
*****
40 ' [117]
50 c$="n":ZONE 10:|FSC,0:GOSUB 110:mn0=m0: [2907]
mn1=m1:mn2=m2:c$="f":|FSC:GOSUB 110
60 PRINT:PRINT:PRINT"Bildschirmausga [11381]
be Normal";TAB(40);"Bildschirmausgabe Fast
Screen":PRINT:PRINT"MODE 0",mn0;"sec";TAB
(40);"MODE 0",m0;"sec"
Listing Fastscreen
    
```

```

70 PRINT:PRINT"MODE 1",mn1;"sec";TAB(40);" [4340]
MODE 1",m1;"sec"
80 PRINT:PRINT"MODE 2",mn2;"sec";TAB(40);" [4244]
MODE 2",m2;"sec":PRINT:PRINT:PRINT:END
90 MODE m:a=TIME:FOR x=1 TO 25:PRINT STRIN [4343]
G$(20*2^m,c$);:NEXT:b=TIME
100 c=(b-a)/300:RETURN [569]
110 m=0:GOSUB 90:m0=c:m=1:GOSUB 90:m1=c:m= [3427]
2:GOSUB 90:m2=c:RETURN
Listing Fastscreen
    
```

SORT

- Eine kleine Sortierroutine

Oft hat man in Basic-Programmen kleinere Datensätze von ca. 20 – 200 Elementen zu sortieren. Man kann dazu zwar ein Programm in Basic schreiben, jedoch hat dieses den Nachteil, daß es meistens sehr langsam ist und zusätzliche Variablen fürs Sortieren braucht.

Deshalb wurde dieses Sortierprogramm als RSX-Erweiterung in Maschinensprache geschrieben. Das Programm »SORT« benötigt zum alphabetischen Sortieren von 50 willkürlich gewählten Wörtern ca. 0.2 sek., für 200 Wörter ca. 4 sek. und für 400 Wörter ca. 12 sek.

Die Wörter werden auf alle Buchstaben genau sortiert, Zahlen kommen vor den Buchstaben und Leerstrings werden hinter die Wörter gesetzt. Falls die Wörter schon sortiert sind und nur ein

neues dazukommt, braucht das Programm selbst bei 1000 Elementen nur maximal 0.5 sek., um dieses einzusortieren. Um die Variablen a\$(0) bis a\$(200) zu sortieren, gibt man !SORT,a\$(0), a\$(200) ein. Um die Variablen name\$(20) bis name\$(89) zu sortieren, gibt man!SORT,name\$(20),name\$(89) ein.

Wenn man z.B. eine Telefonnummern-Verwaltung oder etwas ähnliches machen möchte, kann man die Telefonnummer oder eine Adresse nach dem Namen in die gleiche Stringvariable schreiben. Sie wird dann automatisch mitsortiert.

Durch eine besondere Art der Abspeicherung kann man das Programm an jede beliebige Adresse zwischen &4000 und &A500 laden, indem man der Variablen adr am Anfang des Programmes den jeweiligen Wert zuweist.

Das Programm wurde auf einem CPC 6128 geschrieben, es läuft aber ohne Änderung auch auf dessen Vorgängern CPC 464 und CPC 664, jedoch muß dann beim Aufruf der Routine vor die Variablen jeweils ein Klammeraffe gestellt werden.

(G. Kilian/JB)

für 464-664-6128

```

1 ' [117]
2 ***** [1416]
3 *** Sort c. by Gerd Kilian *** [1444]
4 ***** [1416]
5 ' [117]
10 adr=&A000 an diese Adresse kommt [3882]
das Maschinenspracheprogramm
20 laenge= &AF [593]
30 MEMORY adr-1:IF adr<0 THEN adr=adr+256^ [2863]
2
40 FOR a=adr TO adr+laenge-1:READ a$ [1908]
50 IF LEN(a$)=2 THEN POKE a, VAL("&"a$):GO [3944]
TO 80
60 b=VAL(a$)+adr:IF b<0 THEN b=b+256^2 [1225]
70 POKE a, b-INT(b/256)*256:a=a+1:POKE a, IN [2115]
T(b/256)
80 NEXT:CALL adr:POKE adr,&C9:END [1714]
90 ' [117]
Listing Sort
    
```

```

100 DATA 01,&000A,21,&0014,CD,D1,BC,C9,&00 [2973]
0F,C3,&0018,53,4F,52,D4,00
110 DATA 00,00,00,00,FE,02,C0,2A,FA,BF,2B, [4127]
2B,2B,E5,DD,E1,3E,00,32,&00AE
120 DATA AF,DD,BE,03,28,5B,DD,BE,00,28,2D, [2833]
DD,5E,01,DD,56,02,DD,6E,04
130 DATA DD,66,05,06,00,1A,CB,EF,4F,7E,CB, [3943]
EF,B9,28,04,38,13,18,3A,13
140 DATA 23,04,78,DD,BE,00,28,31,DD,BE,03, [4021]
28,03,C3,&0042,3E,01,32,&00AE
150 DATA DD,46,00,DD,5E,01,DD,56,02,DD,4E, [2949]
03,DD,6E,04,DD,66,05,DD,71
160 DATA 00,DD,75,01,DD,74,02,DD,70,03,DD, [2955]
73,04,DD,72,05,DD,2B,DD,2B
170 DATA DD,2B,DD,E5,E1,ED,5B,FC,BF,B7,ED, [3504]
52,D2,&0029,2A,FC,BF,23,23
180 DATA 23,22,FC,BF,3A,&00AE,B7,C2,&001B, [2202]
C9,00
190 ' [117]
200 ' [117]
210 ' !Sort,@var$(anf),@var$(ende) [1124]
220 ' [117]
Listing Sort
    
```





Bild 1: Ein Beispiel: Titel mit Char erstellt.

Großbuchstaben mit CHAR

Diese RSX-Erweiterung ermöglicht die vergrößerte Darstellung von Zeichen.

Das Programm legt die RSX-Erweiterung ab der von Ihnen angegebenen Anfangsadresse im Speicher ab (üblicherweise &A000). Damit kann das Programm flexibel in den Speicher gelegt werden. Mit GOTO 1000 wird ein kleines Demobild gezeigt. Wenn Sie die Erweiterung abspeichern wollen, brauchen Sie sich nur an die Anweisungen zu halten. Geladen wird sie wie jede MC-Routine:

MEMORY Anfangsadresse - 1 LOAD "<Dateiname>"

Die RSX-Erweiterung besteht aus einem Befehl mit folgendem Format:

TEXT, <ASCII-Code>, <Kursiv-Faktor>, <X>, <Y>, <Breite>, <Höhe>

Es wird nur ein Zeichen ausgegeben, nämlich jenes, welches als <ASCII-Code> in der Programmzeile steht. Dieses geschieht von der Ausgangsgrafikposition <X> und <Y> aus. Es handelt sich hierbei um reelle Grafikkoordinaten. Der Bildschirm hat bei diesen Koordinaten den Ursprung wie üblich un-

ten rechts. In Y-Richtung (nach oben) sind es 200 Punkte, in X-Richtung je nach Mode 160, 320 oder 640 Punkte (160: Mode 0 usw.). Diese anderen Koordinaten sind bei der Programmierung zu beachten. Die <Breite> und <Höhe> bestimmen die Größe, um deren Faktor der Buchstabe vergrößert wird. Eine 1 bedeutet also, daß ein Buchstabe in der normalen Breite/Höhe dargestellt wird. Durch unterschiedliche Werte sind interessante Verzerrungen möglich. Will man den Buchstaben etwas kursiv darstellen, braucht man den <Kursiv-Faktor>. Er gibt an, wie steil der Buchstabe sein soll; je kleiner er ist (> = 1!), umso steiler wird der Buchstabe. Soll der Buchstabe gerade sein, muß der Kursiv-Faktor = 0 sein.

Schauen Sie sich einmal das Demo etwas genauer an und experimentieren Sie vor allem! Mit CHAR lassen sich nämlich tolle Einleitungen für eigene Programme programmieren.

(D. Braun)

für 464-664-6128



```

10 DATA 21,00,00,22,0F,A0,01,11,260 [1521]
20 DATA A0,21,19,A0,C3,D1,BC,00,970 [1286]
30 DATA 00,15,A0,18,0D,54,45,58,459 [1254]
40 DATA D4,00,00,00,00,00,00,00,212 [1821]
50 DATA 00,00,FE,06,C0,DD,7E,0A,809 [1176]
60 DATA DD,6E,00,DD,66,01,DD,5E,970 [1273]
70 DATA 02,DD,56,03,FD,21,1D,A0,787 [2201]
80 DATA F5,3E,08,FD,77,00,FD,77,1059 [1489]
90 DATA 01,DD,7E,04,FD,77,02,DD,947 [1923]
100 DATA 7E,06,FD,77,03,AF,FD,77,1054 [2289]
110 DATA 04,F1,DF,56,A0,C9,59,A0,1164 [1793]
120 DATA FE,F5,CD,1D,BC,F1,E5,CD,1596 [1421]
130 DATA A5,BB,EB,E1,E5,1A,47,CB,1341 [2029]
140 DATA 00,30,1C,C5,3A,38,B3,47,637 [2014]
150 DATA 7E,A8,B1,A9,A8,77,C1,CB,1323 [1250]
160 DATA 09,DC,20,BC,FD,7E,03,3D,892 [2054]
170 DATA FD,77,03,20,E6,18,0E,CB,878 [669]
180 DATA 09,DC,20,BC,FD,7E,03,3D,892 [2054]
190 DATA FD,77,03,20,F2,DD,7E,06,1002 [1139]
200 DATA FD,77,03,FD,7E,02,3D,FD,1068 [1265]
210 DATA 77,00,20,C3,3E,08,FD,77,788 [1369]
220 DATA 00,E1,CD,26,BC,FD,7E,04,1039 [1518]
230 DATA DD,46,08,3C,FD,77,04,B8,919 [1703]
240 DATA 20,09,AF,FD,77,04,CB,01,796 [1082]
250 DATA DC,23,BC,FD,7E,02,3D,FD,1138 [1209]
260 DATA 77,02,20,98,13,DD,7E,04,675 [1544]
270 DATA FD,77,02,FD,7E,01,3D,FD,1068 [651]
280 DATA 77,01,20,88,C9,00 [1031]
290 MODE 2:LOCATE 1,12:INPUT"Anfangsadress [5150]
e (Laenge=&ADD) ";adr
300 IF adr<0 THEN adr=adr+65536 [1131]
310 off=adr-40960:sp=adr:sp=&adr+&DD [2568]
320 MEMORY adr-1:RESTORE 10:bz=8:zz=10 [820]
330 z=0:ps=0 [743]
340 READ a$:a=VAL("&a$"):POKE sp,a:ps=ps+ [2599]
a:z=z+1:sp=sp+1:IF z=bz THEN 360
350 IF sp=sp+1 THEN 380 ELSE 340 [530]

```

Listing Großbuchstaben

```

360 READ p:zz=zz+10:IF p<>ps THEN PRINT" D [3988]
ata-Error in";zz-10:END
370 GOTO 330 [506]
380 a=PEEK(6):IF a<128 THEN POKE adr+&6D, [2568]
&A3:POKE adr+&6E,&B6
390 RESTORE 390:DATA &56,&11,&36,&53,4,7,& [1931]
a
400 FOR n=1 TO 7:READ a:z=adr+a:w=PEEK(z)+ [3281]
PEEK(z+1)*256+off
410 h=INT(w/256):POKE z,w-h*256:POKE z+1,h [3379]
:NEXT
420 CALL adr:PRINT:PRINT"Abspeichern mit S [3478]
AVE"CHR$(34);
430 PRINT"<Dateiname>"CHR$(34)",B,"adr",&D [2757]
D."":PRINT
440 PRINT"Demo mit 'GOTO 1000':"PRINT [2714]
450 END [110]
970 REM [272]
980 REM Demo [446]
990 REM [272]
1000 MODE 1:INK 0,0:BORDER 0:INK 1,2:INK 2 [3307]
,15:INK 3,24
1010 PLOT -2,0,1:a$="PC":bx=20:by=200/7:x= [3062]
10:y=199:kf=0:GOSUB 1400
1020 a=PEEK(6):IF a=128 THEN z=&B338 ELSE [3017]
z=&B6A3
1030 POKE z,&X1010:a$=CHR$(143):bx=40:by=4 [3354]
:x=0:y=48:GOSUB 1400
1040 PLOT -2,0,3:a$="CPC"+CHR$(144)+"Joyce [2214]
"+CHR$(144)+"PC 1512"
1050 bx=2:by=3:x=26:y=43:kf=3:GOSUB 1400:P [2905]
OKE z,&X11011101
1060 a$="Schneider":bx=4:by=9:kf=5:x=25:y= [3530]
180:GOSUB 1400
1070 PLOT -2,0,2:a$="INTERNATIONAL":bx=2:b [4062]
y=4:kf=0:x=58:y=94
1080 GOSUB 1400 [899]
1390 GOTO 1390 [421]
1400 FOR n=1 TO LEN(a$):a=ASC(MID$(a$,n,1) [2927]
)
1410 !TEXT,a,kf,bx,by,x+bx*(n-1)*8,y:NEXT: [1728]
RETURN

```

Listing Großbuchstaben



Kopfnüsse für BASIC-Freaks

Lösen Sie gerne Rätsel und macht es Ihnen Spaß, Probleme zu durchschauen? Hier sind 3 Rätsel für den BASIC-Programmierer auf dem CPC. Die ausführliche Lösung finden Sie unten.

I. Schluckende Tastatur

Geben Sie folgendes BASIC Programm ein:

```
10 KEY DEF 47,1,150,150,150
20 PRINT"Druecken Sie Space !"
30 a$=inkey$
40 if a$ <> CHR$(150) then 30
50 PRINT "OK"
```

Das Programm reagiert nicht auf die Betätigung der Leertaste, obwohl doch das Zeichen 150 daraufgelegt wurde und auch korrekt abgefragt wird. Warum funktioniert das Programm nicht? (Kein Interpreter Fehler!).

II. Ignorierte Schleife

Geben Sie folgende Schleife ein:

```
FOR n=&7E56 to &8023:PRINT PEEK(n):NEXT
```

Warum wird der Speicherinhalt von &7E56 bis &8023 nicht aufgelistet? Sie sollen den Grund des leichten Interpreterfehlers finden und die Schleife korrigieren.

III. Verzwickte Funktion

Die Variable n soll einen natürlichen Wert zwischen 0 und 9 haben. Die Funktion: $a = \text{ABS}(10^{*(n >= 6)} - n)$ ordnet also jedem n (0–9) eine Zahl (a) zu. Wie sieht die Funktion aus, warum gibt die Funktion eine solche Zahlenfolge aus und wie läßt sich der Term für n > 9 erweitern, so daß die Zahlenfolge immer wieder von vorne beginnt?

Viel Spaß bei der Lösung!

(D. Braun)

Lösung der Kopfnüsse!

Gar nicht so einfach, nicht wahr?

Zur Aufgabe I

Die Lösung des Problems liegt in den sogenannten »Expansion Strings« verborgen. Wie Sie wahrscheinlich wissen, lassen sich mit »KEY« kleine Textteile auf die Funktionstasten legen. Diese werden beim Drücken der Funktionstaste ausgegeben. Die einzelnen Textteile werden durch Nummern von 128–159 gekennzeichnet. Dies sind aber nichts anderes als die ASCII-Nummern der Tasten. Wird also mittels KEY DEF auf eine Taste ein ASCII-Code von 128–159 gelegt, so werden bei Tastendruck **nicht** die dort liegenden Grafikzeichen ausgegeben, sondern die Textteile, die mit KEY definiert sind. Nun ist aber das Textteil mit der Nummer 150 noch gar nicht definiert, der Computer empfängt nur ein Leerstring ("") bei dem Drücken der Leertaste, nicht aber ein erwartetes CHR\$(150). Die Funktionstasten funktionieren genauso, nur werden sie beim Einschalten gleich richtig belegt. So liegt z.B. auf der Taste 7 (f7) des Zahlenblockes der ASCII-Code 135, so als wäre er mittels KEY DEF anfangs definiert worden. Der Textteil (auch Exp. String genannt) mit der Nummer 135 enthält die 7, wie nach einem KEY 135, "7".

Sie können also Textteile nicht nur auf die Funktionstasten, sondern auch auf eine beliebige Taste legen, indem Sie die unbenutzten Textteile oberhalb von 140 nehmen:

```
KEY DEF < Tastenmr. > ,0-1,141-159 KEY 141-159,
" < Textteil > "
```

Das Programm läuft dann wieder richtig, wenn das Textteil 150 mit einem CHR\$(150), einem Grafikzeichen, belegt wird: 15 KEY 150, " ".

Zur Aufgabe II:

Der CPC besitzt eine sogenannte Integer-Arithmetik, er kann also, aus Gründen der Schnelligkeit, auch mit ganzen Zahlen von -32767 bis +32767 rechnen. Er stellt also mit 16 Bit nicht nur positive Zahlen bis 65535 dar, sondern teilt den Bereich zur Hälfte für die negativen Zahlen auf. Dies hat aber fatale Folgen für die Hexadezimalen Zahlen. Solange sie noch unter 32768 (= &8000) sind, macht das nichts aus, wohl aber bei Zahlen größer und gleich 32768 (= &8000). Dann wandelt er sie in negative Zahlen um. Testen Sie es doch ein mal mit mehreren Hex-Zahlen und durch einen PRINT-Befehl wie z.B.:

```
"PRINT &8012".
```

Kommen wir nun zur Schleife. Der CPC testet bei einer Schleife die Grenzwerte ab und unterscheidet zwischen REAL und INTEGER-Schleifen, je nach der Größe der Zahlen. Eine HEX-Zahl zählt er aber zu den INT-Zahlen. Eine Schleife mit HEX-Zahlen geht also so lange gut, bis sie über den Grenzwert &8000 geht. So ist in unserem Fall der Anfangswert für den CPC positiv, der Endwert aber negativ, das heißt für ihn, er darf die Schleife nicht durchlaufen, er ignoriert sie. Am einfachsten läßt sich das Programm folgendermaßen korrigieren:

```
FOR n=&7E*256+&56 TO &80*256+&23...
```

Ich spalte also die HEX-Zahl auf: Der vordere Teil, das High-Byte, muß mit 256 malgenommen, dann der hintere Teil addiert werden. Zur HEX-Zahleumformung sind nur noch kleine Zahlen da, die ja korrekt umgeformt werden, und der CPC

erkennt solche errechneten Zahlen in Form eines Termes als REAL-Zahlen an; die Schleife arbeitet einwandfrei.

Zur Aufgabe III:

Die sich ergebenden Zahlen steigen bis 5 und sinken wieder bis 1 herunter:

```
n 0 1 2 3 4 5 6 7 8 9
```

```
a 0 1 2 3 4 5 4 3 2 1
```

Wie das ein einziger Term macht ist gar nicht so schwer zu erklären. Der CPC verarbeitet auch in Ausdrücken Gleichungen, die er bewertet. Ist die Gleichung (und deren Aussage) wahr, so gibt er -1 aus, ist sie falsch, so gibt er eine 0 zurück. Ein PRINT 10 > 1 ergibt also -1, ein PRINT 10 < 1 ergibt 0 als Ergebnis. Bemerkung nebenbei: Binär gesehen ist -1 nur aus Einsen aufgebaut (2-er-Komplement), 0 nur aus Nullen. So verstehen Sie vielleicht auch die AND/OR/XOR-Funktionen besser. Sie können also in BASIC Boolesche Algebra mit 0 und -1 machen. Eine wahrer Ausdruck ist also -1, ein falscher 0. In der Funktionsgleichung ist ein solcher Ausdruck enthalten: $n > = 6$. Nochmals die Gleichung:

```
a=ABS(10*-(n>=6)-n)
```

Machen wir eine Fallunterscheidung:

- Für $n < 6$ gilt: $n > = 6$ ist falsch (0) Eingesetzt also:

```
a=ABS(10*-0-n)
```

oder

```
a=ABS(-n)
```

Für die Zahlen bis 5 wird n in a übernommen (Anstieg bis 5).

- Für $n > 6$ gilt: $n > = 6$ ist wahr (-1) Eingesetzt:

```
a=ABS(10*-(-1)-n)
```

oder

```
a=ABS(10-n)
```

Für die Zahlen größer 5 wird der Abstand zu 10 genommen, bei steigendem n sinkt a, die Zahlen fallen bis zur 1 ab. Um den Term für Zahlen größer 9 tauglich zu machen, müssen wir eine schlecht erklärte Funktion aus dem Keller holen: MOD. Bei PRINT 3 MOD 10 wird der Rest bei der Division von 3 durch 10 gebildet. Dies ist in diesem Falle 3. Interessant wird es bei PRINT 13 MOD 10 oder bei PRINT 23 MOD 10. Jedesmal ist das Ergebnis gleich: 3.

Ein PRINT 0 MOD 10 oder PRINT 10 MOD 10 bringt klarerweise 0. Aber das ist genau das, was wir suchen. Ersetzen wir die n's durch n MOD 10 so werden alle n's größer als 9 zu Zahlen zwischen 0 und 9:

```
a=ABS(10-((n MOD 10) > =6)-n MOD 10)
```

So erhalten wir eine Funktion, die dauernd hoch und runtergeht, zwischen 0 und 5. Mit diesen Gleichungsausdrücken kann man also theoretisch jede beliebige Zahlenfolge darstellen. So kann vielleicht mancheine DATA Zeile verschwinden, die durch solche Ausdrücke (oder Funktionen mit DEF FN) ersetzt werden und helfen können, Speicher zu sparen. Es macht viel Spaß, solche Rätsel zu erfinden, vielleicht überraschen Sie einmal Freunde/Bekanntes mit eigenen kleinen BASIC-Rätsel?

(D. Berger)

Textausgabe mit Pfiff!

Effekte sind bei jedem Spielprogramm das Salz in der Suppe. Dieses Programm macht Schluß mit der langweilig-schnellen Textausgabe der Schneider Computer. Acht verschiedene Effekte mit unzähligen Möglichkeiten bringen die Buchstaben animiert auf den Bildschirm.

Die Textausgabe wird als RSX eingebunden. Untenstehendes Programm liest die Datas in den Speicher. Das eigentliche Maschinenprogramm kann dann mit

```
SAVE" <Dateiname> ",B,&9500,&7d1
```

abgespeichert werden. Geladen wird es mit

```
MEMORY &94FF:LOAD" <Dateiname> ": CALL &9500
```

Braucht man auch einen selbstdefinierten Zeichensatz, so sollte vor dem MEMORY ein SYMBOL AFTER 32 stehen. Nun zu den Befehlen:

Befehlsübersicht

Bei allen Befehlen wird der RSX-Querstrich am Anfang eingetippt. Es sind 3 Befehle: EFFEKT, ON und OFF. Mit EFFEKT werden die einzelnen Effekte angesprochen. Will man den Text wieder normal ausgeben, so ist hierfür OFF zuständig; der Text wird danach wieder normal auf dem Bildschirm dargestellt. Nach einem OFF-Befehl kann der alte Zustand mit ON wiederhergestellt werden. Die Befehle ON und OFF haben keine Parameter, dagegen EFFEKT umso mehr. Es sind bei diesem Befehl immer 5 Parameter. Der erste gibt die Nummer des Effekts von 1 bis 8 an. Die anderen 4 Parameter sind für die unterschiedliche Textausgabe maßgebend. Es gibt Effekte, die zeilenorientiert sind und solche, die zeichenweise arbeiten. Bei Zeilenorientierten wird eine Zeile intern nur dann animiert ausgegeben, wenn wirklich eine Zeile geschrieben wurde. Dies merkt das Programm aber erst, wenn in einer anderen Zeile ein Zeichen gedruckt wurde. Somit ist das Programm also immer eine Zeile in der Ausgabe hinterher; das stört im Programm nicht, im Direktmodus ist es aber etwas gewöhnungsbedürftig.

Textausgabe Nr. 1

Vor der Ausgabe des eigentlichen Schriftzeichens wird zuvor eine Reihe anderer Zeichen an dieser Stelle ausgegeben. Diese Zeichenreihe wird durch den ASCII-Code des 1. Zeichens und die Zeichenanzahl markiert. Eine Wartezeit (>0) gibt die Geschwindigkeit der Ausgabe an: je größer, desto langsamer. Die Flagge zeigt, ob Zeichen- oder Zeilenorientiert ausgegeben werden soll, 0 für Zeichenweise, 1 für die Zeilenausgabe. Mit selbstdefinierten Zeichen ist dieser Effekt sehr flexibel. Das Format des Befehls EFFEKT:

```
EFFEKT,1,<Zeich/ZeilFlagge> , <ASCII des 1.Zeichens> , <Anz. Zeichen> , <Warteschleifenlänge>
```

Eine Eingabe zum Probieren:

```
EFFEKT,1,0,212,4,1400
```

Textausgabe Nr.2

Der Buchstabe wird aus einzelnen Punkten auf den Bildschirm gebrösel. Die 'Bröselung' steuert die 'Bröselzahl'. Sie sollte ungerade sein (ein interessanter Wert ist 11), denn bei geraden

Zahlen wird der Buchstabe strichweise aufgebaut (...). Wie bei allen anderen Effekten läßt sich auch hier die Geschwindigkeit durch die Warteschleifenlänge einstellen, und die zeichen- und zeilenweise Ausgabe ist wie gehabt.

```
EFFEKT,2,<Zeich/Zeil-Flagge> , <Bröselzahl> , <Warteschleifenlänge> , <leerer Parameter>
```

Der 'leere Parameter' füllt nur die Parameteranzahl auf 5 auf; sein Wert ist frei wählbar. Interessante Werte zum Eingeben:

```
EFFEKT,2,0,11,170,0
```

```
EFFEKT,2,1,11,1,0
```

Textausgabe Nr. 3

Das Zeichen wird pixelweise aus dem Nichts an seinen Ort gescrollt. Die Scrollrichtung ist variabel, auch eine diagonale Scrollrichtung ist möglich. Sie wird durch einen X und einen Y-Faktor beschrieben. Er kann jeweils den Wert -7,0 oder 7 beinhalten. Die Zahl -7 steht für Scrollrichtung von links (x) bzw. von oben (y), 7 für Scrollrichtung von rechts (x) und von unten (y). Bei 0 wird in dieser Richtung nicht gescrollt. Werte zwischen 0 und 7 oder 0 und -7 sind auch möglich, hier testen Sie am Besten deren Wirkung!

```
EFFEKT,3,<Zeich/Zeil-Flagge> , <X-Faktor> , <Y-Faktor> , <Wartezahl>
```

Ein Beispiel:

```
EFFEKT,3,0,7,7,400
```

Textausgabe Nr.4

Das Zeichen wird zentral aus der Mitte (A-Flagge = 1) oder von außen zur Mitte (A-Flagge = 0) aufgebaut. Ansonsten stimmt die Eingabe mit den anderen Effekten überein.

```
EFFEKT,4,<Zeich/Zeil-Flagge> , <A-Flagge> , <Wartezahl> , <Leerparam>
```

Ein Beispiel:

```
Effekt,4,1,1,450,0
```

Textausgabe Nr. 5

Bevor der Buchstabe endgültig gezeigt wird, dreht er sich um die eigene Achse. Die D-Flagge gibt die Drehrichtung (0=Rechts, 1=Links) an, APOSITION die Lage des Buchstabens, von der er aus gedreht wird, bis er in der richtigen Lage ist. Diese Position wird mit einer Zahl von 1 bis 3 markiert:

- 1: Buchstabe liegend nach unten
- 2: Buchstabe kopfstehend
- 3: Buchstabe liegend nach oben

Der erste der 4 Parameter des Effekts muß immer Null sein!

```
EFFEKT,5,0,<D-Flagge> , <Aposition> , <Wartezahl=Warteschleifenlänge>
```

Beispiele für ganze Drehung nach rechts:

```
EFFEKT,5,0,0,3,2400
```

links:

```
EFFEKT,5,0,1,1,2400
```

kurze Drehung:

```
EFFEKT,5,0,1,1,3500
```

Textausgabe Nr. 6

Alle Zeichen werden gedreht dargestellt. Dieser Effekt stellt also keine Animation dar, es ist aber trotzdem interessant, sich ein Listing auf dem Kopf anzusehen. Als erster Parameter wird wieder nur eine Null eingegeben, die Position wird wie bei Effekt 5 berechnet.

EFFEKT,6,0,<Position>,<Leerparam>,<Leerparam>

Beispiel für die Ausgabe auf dem Kopf stehender Zeichen:

EFFEKT,6,0,2,0,0

Textausgabe Nr. 7

Eine Zeile fährt von rechts in den Bildschirm bis zu ihrer Position hinein. Diese Funktion ist nur zeilenweise zu haben. Als einziger Parameter ist die Wartezahl anzugeben.

EFFEKT,7,<Wartezahl>,<Leerparam>,<Leerparam>,<Leerparam>

Auch hierzu ein Beispiel:

EFFEKT,7,40,0,0,0

Textausgabe Nr. 8

Haben Sie schon einmal die Anzeige in einem Bahnhof oder Flughafen gesehen? Die mechanischen Anzeigen klappern jeweils ihren gesamten Zeichenvorrat herunter, bis sie bei dem richtigen Zeichen angelangt sind. Das kann ihr CPC natürlich auch...

Vom Ausgangswert (ASCII-Wert) aus wird der gesamte Zeichenvorrat ausgegeben, bis das richtige Zeichen dasteht. Dieser Effekt funktioniert nur zeilenweise.

EFFEKT,8,<Ausgangswert>,<Wartezahl>,<Leerparam>,<Leerparam>

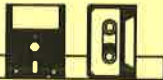
Das letzte Beispiel:

EFFEKT,8,32,900,0,0

Laden Sie einfach einmal die RSX-Erweiterung und probieren Sie die Effekte aus. Viel Spaß bei der effektvollen Programmierung!

(B. Braun)

für 464-664-6128



```

5 SYMBOL AFTER 32:MEMORY &94FF:MODE 2
[2028]
7 LOCATE 1,5:PRINT"Die Datas werden eingel [5319]
aden ...":PRINT:PRINT
10 DATA 21,00,00,22,16,95,01,18,95,21,4 [1782]
45
20 DATA 2E,95,CD,D1,BC,2A,D4,BD,22,33,1 [1476]
325
30 DATA 95,C9,00,00,23,95,C3,C4,95,C3,1 [1280]
269
40 DATA AB,95,C3,0D,96,45,46,46,45,4B,1 [1076]
031
50 DATA D4,4F,46,C6,4F,CE,04,00,00,00,8 [1631]
48
60 DATA C3,00,00,00,00,00,00,00,00,00,1 [2103]
95
70 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
80 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
90 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
100 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
110 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
120 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
130 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
140 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
150 DATA 00,00,46,96,A5,96,F8,97,0A,99,1 [1584]
097
160 DATA 51,9A,9C,9A,77,9B,DA,9B,00,00,1 [929]
192
170 DATA 00,00,00,00,00,00,00,00,00,00,0 [976]
180 DATA 00,2A,33,95,22,D4,BD,C9,C5,D5,1 [2075]
288
190 DATA 11,3F,95,21,3E,95,36,00,01,4F,6 [2762]
07
200 DATA 00,ED,B0,D1,C1,C9,FE,05,C0,DD,1 [2237]
688
210 DATA 7E,08,5F,1D,FE,07,30,05,DD,7E,9 [1033]
19
220 DATA 06,18,02,3E,FF,32,35,95,DD,6E,9 [1679]
32
230 DATA 00,DD,66,01,22,3C,95,DD,6E,02,9 [1082]
00
240 DATA DD,66,03,22,3A,95,DD,6E,04,DD,1 [1703]
123
250 DATA 66,05,22,38,95,DD,6E,06,DD,66,1 [2440]
006
260 DATA 07,22,36,95,CB,23,16,00,21,8E,6 [2134]
79
270 DATA 95,19,5E,23,56,ED,53,43,96,21,9 [2533]
59
280 DATA 16,96,22,D4,BD,C3,B2,95,F5,3A,1 [2604]
432
290 DATA 35,95,B7,28,23,3A,9E,95,AD,20,1 [2052]
030
300 DATA 0E,7C,B7,28,0A,6C,26,00,11,3E,5 [1933]
Listing Pliff
    
```

```

96
310 DATA 95,19,F1,77,C9,E5,37,CD,42,96,1 [1601]
440
320 DATA CD,B2,95,E1,7D,32,9E,95,18,E7,1 [1878]
494
330 DATA F1,B7,C3,00,00,C9,38,19,4F,3A,1 [2544]
038
340 DATA 3A,95,47,3A,38,95,CD,11,9C,3C,9 [2859]
79
350 DATA ED,5B,3C,95,CD,1D,9C,10,F3,79,1 [1400]
307
360 DATA C3,11,9C,3A,9E,95,6F,3A,3A,95,1 [1915]
109
370 DATA 47,3A,38,95,4F,11,3E,95,26,00,6 [2344]
79
380 DATA 1A,B7,28,04,79,CD,11,9C,13,24,8 [2379]
07
390 DATA 7C,FE,4F,20,F1,D5,ED,5B,3C,95,1 [2689]
480
400 DATA CD,1D,9C,D1,0C,10,E0,3A,9E,95,1 [1991]
216
410 DATA 6F,26,00,11,3E,95,1A,B7,28,03,6 [1875]
29
420 DATA CD,11,9C,24,13,7C,FE,4F,20,F2,1 [1367]
164
430 DATA C9,DA,16,97,CD,64,9C,E5,CD,3A,1 [1269]
545
440 DATA 9C,22,9F,95,EB,E1,CD,1A,BC,22,1 [1443]
411
450 DATA A2,95,22,A4,95,CD,B8,9C,79,32,1 [1533]
374
460 DATA A1,95,06,00,3E,40,F5,CD,DA,96,1 [2302]
260
470 DATA D5,ED,5B,3A,95,CD,1D,9C,D1,F1,1 [1845]
588
480 DATA 3D,20,EF,C9,1A,C5,47,CD,E8,BD,1 [2566]
453
490 DATA C1,3A,38,95,F5,CB,09,DC,F6,96,1 [1919]
529
500 DATA 78,C6,20,47,DC,FB,96,F1,3D,20,1 [2171]
376
510 DATA EF,C9,CD,20,BC,13,C9,2A,A2,95,1 [2954]
438
520 DATA 04,78,E6,07,28,07,CD,26,BC,22,8 [2104]
73
530 DATA A2,95,C9,06,00,2A,A4,95,ED,5B,1 [1519]
201
540 DATA 9F,95,18,F1,CD,89,9C,21,D3,9C,1 [2239]
471
550 DATA 22,9F,95,E5,3A,9E,95,6F,26,00,1 [2458]
085
560 DATA CD,1A,BC,22,A2,95,22,A4,95,CD,1 [1379]
316
570 DATA B8,9C,79,32,A1,95,78,32,A6,95,1 [1919]
306
580 DATA 06,00,D1,3E,40,F5,E5,D5,C5,FD,1 [2148]
478
590 DATA 21,3E,95,06,4F,FD,7E,00,FD,23,9 [1604]
96
600 DATA B7,C5,28,05,1A,47,CD,E8,BD,CD,1 [1393]
353
Listing Pliff
    
```

Tips & Tricks

610 DATA 2D,9C,3A,A6,95,47,CD,20,BC,10,1 [2538]
 086
 620 DATA FB,C1,10,E3,C1,D1,E1,CD,E1,96,1 [2508]
 894
 630 DATA D5,ED,5B,3A,95,CD,1D,9C,D1,F1,1 [1845]
 588
 640 DATA 3D,20,C6,C9,CD,89,9C,11,D3,9C,1 [1169]
 374
 650 DATA 3A,9E,95,6F,26,00,CD,1A,BC,CD,1 [1725]
 138
 660 DATA DE,97,E5,D5,C5,CD,A5,98,E5,2A,1 [1759]
 805
 670 DATA A2,95,22,9F,95,E1,CD,AD,97,C1,1 [2048]
 600
 680 DATA ED,5B,3C,95,CD,1D,9C,D1,E1,CD,1 [1321]
 566
 690 DATA 23,98,C8,18,DF,3E,4F,FD,21,3E,1 [2541]
 123
 700 DATA 95,F5,C5,E5,D5,FD,7E,00,FD,23,1 [1516]
 700
 710 DATA B7,28,0B,E5,2A,9F,95,22,A2,95,1 [2225]
 158
 720 DATA E1,CD,4E,98,D1,CD,2D,9C,E1,3A,1 [1347]
 558
 730 DATA A6,95,47,CD,20,BC,10,FB,C1,F1,1 [1904]
 512
 740 DATA 3D,20,D6,C9,CD,B8,9C,78,32,A6,1 [2538]
 389
 750 DATA 95,CD,C7,9C,32,A7,95,3A,38,95,1 [1830]
 338
 760 DATA 32,A4,95,3A,3A,95,32,A5,95,C9,1 [2485]
 193
 770 DATA DA,7A,97,E5,CD,64,9C,CD,3A,9C,1 [1994]
 600
 780 DATA EB,E1,CD,1A,BC,CD,DE,97,E5,D5,1 [2347]
 899
 790 DATA C5,CD,A5,98,CD,4E,98,C1,ED,5B,1 [1145]
 675
 800 DATA 3C,95,CD,1D,9C,D1,E1,CD,23,98,1 [1396]
 425
 810 DATA C8,18,E7,3A,A4,95,47,3A,A5,95,1 [1976]
 269
 820 DATA B0,C8,78,B7,28,06,CD,46,98,32,1 [1555]
 202
 830 DATA A4,95,3A,A5,95,B7,28,06,CD,46,1 [2019]
 189
 840 DATA 98,32,A5,95,3E,01,B7,C9,CB,7F,1 [1955]
 293
 850 DATA 20,02,3D,C9,3C,C9,3A,A2,95,E5,1 [1967]
 155
 860 DATA D5,C5,F5,CD,80,98,CB,09,DC,20,1 [1243]
 604
 870 DATA BC,CB,08,DC,7E,98,F1,3D,20,EE,1 [1497]
 469
 880 DATA C1,D1,3A,A6,95,6F,26,00,19,EB,1 [2194]
 184
 890 DATA E1,CD,26,BC,3A,A3,95,3D,32,A3,1 [1592]
 300
 900 DATA 95,20,D1,C9,13,C9,C5,E5,3A,A7,1 [917]
 462
 910 DATA 95,4F,1A,60,A4,6F,CB,05,CB,04,1 [1435]
 040
 920 DATA 30,FA,CB,0D,AF,B5,CB,0D,0D,20,1 [2187]
 131
 930 DATA FA,E1,C1,C5,47,7E,A8,B1,A9,A8,1 [1672]
 744
 940 DATA 77,C1,C9,3A,A5,95,F5,CD,FD,98,1 [2349]
 740
 950 DATA 32,A3,95,F1,B7,28,06,CB,7F,28,1 [1925]
 202
 960 DATA 2C,18,32,3A,A4,95,F5,CD,FD,98,1 [2620]
 344
 970 DATA 32,A2,95,F1,41,B7,C8,CB,7F,28,1 [2805]
 420
 980 DATA 02,18,0B,F5,CB,09,DC,20,BC,F1,1 [1475]
 175
 990 DATA 3D,20,F6,C9,2F,3C,CB,08,30,01,9 [1686]
 07
 1000 DATA 13,3D,20,F8,C9,47,CD,26,BC,10,1 [1723]
 079
 1010 DATA FB,18,CE,2F,3C,E5,EB,47,3A,A6,1 [1664]
 347
 1020 DATA 95,5F,16,00,19,10,FD,EB,E1,18,1 [1234]
 044
 1030 DATA BC,CB,7F,C4,07,99,47,3E,08,90,1 [966]
 159
 1040 DATA C9,2F,3C,C9,DA,4F,99,CD,64,9C,1 [1443]
 420
 1050 DATA E5,CD,3A,9C,EB,E1,CD,1A,BC,CD,1 [1738]
 732
 1060 DATA B8,9C,78,32,A6,95,3A,38,95,B7,1 [2136]
 Listing Pflff

271
 1070 DATA 06,04,28,02,06,01,CD,A7,99,CD,7 [1747]
 89
 1080 DATA 33,99,C8,18,F7,D5,ED,5B,3A,95,1 [2319]
 423
 1090 DATA CD,1D,9C,D1,3A,38,95,B7,28,03,1 [1692]
 088
 1100 DATA 04,18,02,05,C8,78,FE,05,C8,3E,8 [1501]
 76
 1110 DATA 01,B7,C9,CD,89,9C,3A,9E,95,6F,1 [2245]
 359
 1120 DATA 26,00,CD,1A,BC,11,D3,9C,CD,B8,1 [2218]
 230
 1130 DATA 9C,78,32,A6,95,3A,38,95,B7,06,1 [1703]
 093
 1140 DATA 04,28,02,06,01,E5,D5,C5,CD,7E,1 [2271]
 023
 1150 DATA 99,C1,D1,E1,CD,33,99,C8,18,F1,1 [2097]
 654
 1160 DATA 3E,4F,FD,21,3E,95,F5,FD,7E,00,1 [2737]
 262
 1170 DATA FD,23,B7,C5,E5,D5,28,03,CD,A7,1 [1216]
 525
 1180 DATA 99,D1,CD,2D,9C,E1,3A,A6,95,47,1 [1078]
 437
 1190 DATA CD,20,BC,10,FB,C1,F1,3D,20,DE,1 [1829]
 441
 1200 DATA C9,E5,D5,C5,C5,3E,04,90,28,16,1 [1945]
 309
 1210 DATA 47,F5,CD,26,BC,CD,E7,99,10,F8,1 [1616]
 600
 1220 DATA F1,47,CB,09,30,04,13,CD,20,BC,1 [1842]
 020
 1230 DATA 10,F6,79,C1,4F,CB,00,05,CD,1C,1 [1776]
 096
 1240 DATA 9A,CD,00,9A,CD,2C,9A,CD,0E,9A,1 [1767]
 289
 1250 DATA C1,D1,E1,C9,C5,1A,47,7E,A8,B1,1 [2019]
 593
 1260 DATA A9,A8,77,C1,C9,E5,3A,A6,95,6F,1 [2298]
 563
 1270 DATA 26,00,19,EB,E1,C9,E5,3A,A6,95,1 [1829]
 326
 1280 DATA 6F,26,00,EB,B7,ED,52,EB,E1,C9,1 [2647]
 547
 1290 DATA C5,CD,26,BC,CD,E7,99,CD,DC,99,1 [1720]
 795
 1300 DATA 10,F5,C1,C9,C5,CD,29,BC,CD,F2,1 [2675]
 733
 1310 DATA 99,CD,DC,99,10,F5,C1,C9,C5,CB,1 [2495]
 786
 1320 DATA 09,30,04,13,CD,20,BC,CD,DC,99,1 [2292]
 083
 1330 DATA 10,F3,18,0E,C5,CB,01,30,04,1B,7 [1965]
 77
 1340 DATA CD,23,BC,CD,DC,99,10,F3,79,C1,1 [1801]
 579
 1350 DATA 4F,C9,CD,64,9C,E5,CD,3A,9C,EB,1 [1576]
 624
 1360 DATA CD,C7,9C,32,A7,95,CD,B8,9C,E1,1 [1793]
 696
 1370 DATA C9,E5,F5,CD,3E,9A,3A,3A,95,3D,1 [2110]
 422
 1380 DATA 32,AA,95,E5,D5,C5,41,C5,CD,B3,1 [1615]
 654
 1390 DATA 9A,79,C1,4F,CD,E6,9A,C1,ED,5B,1 [1413]
 657
 1400 DATA 3C,95,CD,1D,9C,D1,E1,22,A2,95,1 [1081]
 378
 1410 DATA 3A,38,95,B7,28,0D,3A,AA,95,3C,9 [1678]
 36
 1420 DATA 32,AA,95,FE,03,20,D4,18,0B,3A,9 [2537]
 63
 1430 DATA AA,95,3D,32,AA,95,FE,FF,20,C7,1 [1464]
 489
 1440 DATA F1,E1,CD,11,9C,C9,CD,3E,9A,3A,1 [3067]
 524
 1450 DATA 38,95,E6,03,3D,32,AA,95,41,C5,1 [1794]
 130
 1460 DATA CD,B3,9A,79,C1,4F,C3,E6,9A,C5,1 [1810]
 707
 1470 DATA CD,1A,BC,C1,3A,AA,95,FE,01,06,1 [817]
 250
 1480 DATA 07,38,16,28,0D,CD,C8,9A,18,16,7 [2017]
 43
 1490 DATA CB,09,DC,20,BC,10,F4,C9,C5,CD,1 [2328]
 515
 1500 DATA C8,9A,79,C1,4F,CD,26,BC,10,FB,1 [1322]
 445
 1510 DATA 18,00,22,A2,95,79,32,A1,95,C9,1 [1942]
 051
 Listing Pflff

```

1520 DATA E5,3A,AA,95,FE,01,21,25,9B,38,1 [905]
142
1530 DATA 08,21,3F,9B,28,03,21,5D,9B,E3,8 [1312]
10
1540 DATA 3E,08,32,A8,95,32,A9,95,CD,80,1 [2659]
138
1550 DATA 98,CB,08,30,01,13,E3,E9,3A,A8,1 [1570]
117
1560 DATA 95,3D,32,A8,95,20,ED,3E,08,32,9 [1429]
66
1570 DATA A8,95,3A,A9,95,3D,32,A9,95,20,1 [1510]
154
1580 DATA DF,E1,C9,E3,3A,A8,95,FE,01,28,1 [1918]
546
1590 DATA 05,CD,29,BC,18,DA,2A,A2,5B [2009]
237
1600 DATA 09,DC,20,BC,22,A2,95,18,CD,E3,1 [1588]
250
1610 DATA 3A,A8,95,FE,01,28,07,CB,01,DC,1 [2143]
101
1620 DATA 23,BC,18,BE,2A,A2,95,3A,A1,95,1 [1032]
158
1630 DATA 4F,CD,29,BC,22,A2,95,18,AF,E3,1 [2439]
284
1640 DATA 3A,A8,95,FE,01,28,05,CD,26,BC,1 [1157]
106
1650 DATA 18,A2,2A,A2,95,CB,01,DC,23,BC,1 [679]
186
1660 DATA 22,A2,95,18,95,CD,89,9C,60,C5,1 [1821]
309
1670 DATA CD,3A,9C,C1,EB,79,90,C8,3C,32,1 [1922]
422
1680 DATA A2,95,3A,9E,95,61,6F,CD,1A,BC,1 [1105]
303
1690 DATA 78,32,A6,95,06,01,C5,E5,D5,C5,1 [1493]
328
1700 DATA CD,C0,9B,CD,20,BC,10,FB,C1,10,1 [2239]
453
1710 DATA F4,ED,5B,36,95,CD,1D,9C,D1,E1,1 [886]
599
1720 DATA 3A,A6,95,47,CD,23,BC,10,FB,C1,1 [1732]
332
1730 DATA 3A,A2,95,A8,C8,04,18,D6,E5,CD,1 [2602]
413
1740 DATA B8,9C,0E,08,E5,C5,1A,77,13,CD,1 [2771]
157
1750 DATA 20,BC,10,F8,C1,E1,CD,26,BC,0D,1 [1872]
346
1760 DATA 20,EE,E1,C9,11,3E,95,3A,9E,95,1 [2683]
289
1770 DATA 6F,26,00,3A,36,95,4F,D5,E5,CD,1 [2142]
136
1780 DATA FB,9B,E1,ED,5B,38,95,CD,1D,9C,1 [1891]
554
1790 DATA D1,78,B7,C8,0C,18,EC,06,00,1A,1 [2166]
016
1800 DATA B9,79,30,08,13,24,7C,FE,50,C8,1 [1681]
075
Listing Pfiff

```

```

1810 DATA 18,F3,CD,11,9C,06,FF,18,F1,E5,1 [771]
400
1820 DATA C5,D5,F5,CD,32,95,F1,D1,C1,E1,1 [1224]
927
1830 DATA C9,F5,D5,F5,F1,F5,F1,F5,F1,1B,2 [2857]
144
1840 DATA 7B,B2,20,F5,D1,F1,C9,C5,E5,CD,1 [1417]
860
1850 DATA 58,9C,68,26,00,19,EB,E1,C1,C9,1 [1184]
265
1860 DATA D5,F5,11,D3,9C,6C,26,00,CD,11,1 [1784]
210
1870 DATA BC,06,05,38,05,06,04,28,01,05,3 [1702]
16
1880 DATA CB,25,CB,14,10,FA,19,F1,D1,C9,1 [1555]
405
1890 DATA CD,11,BC,06,20,D8,06,10,C8,06,8 [2142]
92
1900 DATA 08,C9,E5,D5,C5,CD,3A,9C,E5,CD,1 [2063]
701
1910 DATA A5,BB,D1,D5,CD,53,BC,D1,CD,58,1 [1697]
752
1920 DATA 9C,2A,8F,B2,1A,2F,A4,4F,1A,A5,1 [991]
026
1930 DATA B1,12,13,10,F5,C1,D1,E1,C9,11,1 [2006]
320
1940 DATA 3E,95,01,00,00,26,00,1A,B7,28,4 [2541]
99
1950 DATA 19,CB,78,20,03,44,CB,F8,CD,64,1 [2250]
207
1960 DATA 9C,13,24,7C,FE,4F,20,EB,CB,78,1 [1266]
258
1970 DATA C8,CB,B8,0E,4F,C9,CB,78,28,ED,1 [2165]
481
1980 DATA 4C,0D,CB,B8,18,E7,CD,11,BC,01,1 [1891]
142
1990 DATA AA,04,D8,01,88,02,C8,01,80,01,8 [1308]
59
2000 DATA C9,CD,11,BC,3E,02,D8,3E,04,C8,1 [2331]
157
2010 DATA 3E,08,C9 [481]
2020 RESTORE 10:sp=38144: bz=10:zz=10:spe=4 [2773]
0146
2030 z=0:ps=0 [743]
2040 READ a$:a=VAL("&"+a$):POKE sp,a:ps=ps
+a:z=z+1:sp=sp+1:IF z=bz THEN 2060
2050 IF sp=spe+1 THEN PRINT"Alles OK...":G [2621]
OTO 2100 ELSE 2040
2060 READ p:zz=zz+10:IF p<>ps THEN PRINT" [3988]
Data-Error in":zz-10:END
2070 GOTO 2030 [365]
2100 IF PEEK(6)<>128 THEN POKE &9C78,&2F:P [3377]
OKE &9C79,&B7
2105 CALL &9500:|EFFEKT,8,32,1300,0,0 [1583]
2107 PRINT:PRINT"Abspeichern mit 'SAVE"CHR [4538]
$(34)"<Dateiname>";
2110 PRINT CHR$(34)"",B,&9500,&7d1":PRINT [2192]
2120 PRINT " "":;:OFF:END [1295]
Listing Pfiff

```

Magische Quadrate

Programmbeispiele für den CPC 464, 664 und 6128

Die Anordnung von fortlaufenden natürlichen Zahlen zu magischen Quadraten beschäftigt die Menschen schon seit langer Zeit. So finden wir z.B. auf einem Kupferstich Albrecht Dürers aus dem Jahre 1514 ein magisches Quadrat, gebildet aus den Zahlen von 1 bis 16:

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Bei diesem Quadrat findet man nicht nur in den Summen der Spalten, Zeilen und Diagonalen die gleiche Zahl, nämlich 34, sondern diese »magische Zahl« trifft noch viel, viel häufiger auf. Bei genauerem Betrachten findet man immer wieder aufs Neue diese magische Summe.

Jetzt sollen mit dem Computer magische Quadrate ungerader Ordnung erzeugt werden. Ungerade Ordnung bedeutet, daß die Anzahl der Zeilen und damit auch die Anzahl der Spalten ungerade

ist. Für das Erzeugen solcher Quadrate gibt es eigentlich nur drei Grundregeln, die zu beachten sind:

1. Es wird in der ersten Zeile in der Mitte begonnen.
2. Die folgende Zahl steht rechts über ihrem Vorgänger.
3. Ist ein Feld bereits besetzt, wird die Zahl unter ihren Vorgänger geschrieben.

Am besten läßt sich die Methode an einem Beispiel (n=5) veranschaulichen. Wir erzeugen unser Hauptquadrat DIM quadrat (n, n) und denken uns dazu drei Hilfsquadrate. Das Startfeld wird in der ersten Zeile in der Mitte festgelegt.

$$\text{zeile} = 1 : \text{spalte} = (n + 1) / 2$$

Die Zahl 1 kommt auf das Startfeld-Quadrat (zeile, spalte) = 1 (Bild 1). Es folgt ein Zug nach oben rechts.

$$\text{zeile} = \text{zeile} - 1 : \text{spalte} = \text{spalte} + 1$$

Die Zahl 2 liegt jetzt in einem der Hilfsquadrate. Sie soll aber an die entsprechende Stelle im Hauptprogramm kommen.

$$\text{IF zeile} < 1 \text{ THEN zeile} = \text{zeile} + n$$

Die Zahl 3 ist jetzt problemlos anzufügen (Bild 2). Die Zahl liegt im rechten Hilfsquadrat. Sie kommt an die entsprechende Stelle im Hauptquadrat.

$$\text{IF spalte} > n \text{ THEN spalte} = \text{spalte} - n$$

Die Zahl 5 ist wieder problemlos anzufügen (Bild 3). Das Feld für die Zahl 6 ist bereits durch die 1 belegt, also muß die 6 unter ihren Vorgänger, die 5. Es erfolgt zunächst ein Rückzug.

zeile=zeile+1:spalte=spalte-1 Dann erfolgt ein Zug nach unten.
 zeile=zeile+1 Die Zahlen 7 und 8 sind wieder problemlos anzufügen (Bild 4). Beim Eintrag der Zahlen 9 und 10 müssen wieder das obere bzw. das rechte Hilfsquadrat benutzt werden. Das Feld für die 11 ist bereits durch die 6 belegt, also kommt die 11 unter die 10 (Bild 5). Die Zahlen 12 bis 15 sind wieder leicht auf der Hauptdiagonalen einzutragen. Die Zahl 16 steht im 3. Hilfsquadrat. Da das entsprechende Feld im Hauptquadrat schon durch die 11 belegt ist, kommt

sie unter die 15 (Bild 6). Auf dieselbe Art wird nun fortgefahren, bis alle 25 Zahlen eingetragen sind (Bild 7). Das Quadrat ist nun zu einem »magischen Quadrat« geworden. Alle Zeilen, Spalten und Diagonalen ergeben die Summe 65. Diese Summe läßt sich nach der Formel $(1 + n^2) : 2n$ berechnen. Auf diese Art lassen sich nun alle »Magischen Quadrate« ungerader Ordnung erstellen. Unten steht nun das vollständige und mit Kommentaren versehene Listing.

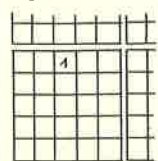
(R. Morlock/JB)

Für das Erzeugen solcher Quadrate gibt es eigentlich nur 3 Grundregeln, die zu beachten sind:

1. Es wird in der ersten Zeile in der Mitte begonnen.
2. Die folgende Zahl steht rechts über ihrem Vorgänger.
3. Ist ein Feld bereits besetzt, wird die Zahl unter ihrem Vorgänger geschrieben.

Am besten läßt sich die Methode an einem Beispiel (n = 5) veranschaulichen:

Magisches Quadrat

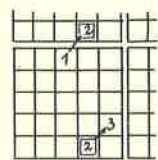


Verbale Erläuterung

Wir erzeugen unser Hauptquadrat und denken uns dazu 3 Hilfsquadrate. Das Startfeld wird in der 1. Zeile in der Mitte festgelegt. Die Zahl 1 kommt auf das Startfeld

Ausführung in BASIC

```
DIM quadrat(n,n)
zeile=1
spalte=(n+1)/2
quadrat(zeile,spalte)=1
Bild 1
```

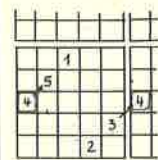


Es folgt ein Zug nach oben rechts

Die Zahl 2 liegt jetzt in einem der Hilfsquadrate. Sie soll aber an die entsprechende Stelle im Hauptquadrat kommen. Die Zahl 3 ist jetzt problemlos anzufügen.

```
zeile=zeile-1
spalte=spalte+1
IF zeile < 1
THEN zeile=zeile+n
```

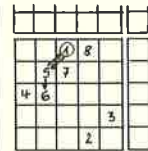
Bild 2



Die Zahl 4 liegt im rechten Hilfsquadrat. Sie kommt an die entsprechende Stelle im Hauptquadrat. Die Zahl 5 ist wieder problemlos anzufügen.

```
IF spalte > n
THEN spalte=spalte-n
```

Bild 3



Das Feld für die Zahl 6 ist bereits durch die 1 belegt, also muß die 6 unter ihren Vorgänger, die 5

Es erfolgt zuerst ein Rückzug.

Dann erfolgt ein Zug nach unten. Die Zahlen 7 und 8 sind wieder problemlos anzufügen.

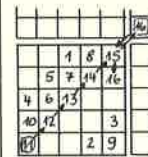
```
zeile=zeile+1
spalte=spalte-1
zeile=zeile+1
```

Bild 4



Beim Eintrag der Zahlen 9 und 10 müssen wieder das obere bzw. das rechte Hilfsquadrat benutzt werden. Das Feld für die 11 ist bereits durch die 6 belegt, also kommt die 11 unter die 10.

Bild 5



Die Zahlen 12 bis 15 sind wieder leicht auf der Hauptdiagonalen einzutragen. Die Zahl 16 steht im 3. Hilfsquadrat. Da das entsprechende Feld im Hauptquadrat schon durch die 11 belegt ist, kommt sie unter die 15.

Bild 6



Auf dieselbe Art wird nun fortgefahren, bis alle 25 Zahlen eingetragen sind.

Bild 7

Das Quadrat ist nun zu einem »Magischen Quadrat« geworden. Alle Zeilen, Spalten und Diagonalen ergeben die Summe 65. Diese Summe läßt sich nach der Formel

$$su = (1 + n^2) : 2n$$

berechnen. Auf diese Art lassen sich nun alle »Magischen Quadrate« ungerader Ordnung erstellen.

```
10 CLS [91]
20 MODE 2 [513]
30 PRINT " ***** Zauberkq [3520]
adrate ***** "
40 PRINT " Mit diesem Programm lasse [6638]
n sich Zauberkquadrate ungerader"
50 PRINT " Ordnung er [3327]
zeugen."
60 PRINT [361]
70 PRINT " Geben Sie bitte eine unger [4957]
ade Zahl groesser als 3 ein."
80 PRINT " Bis zur Ordnung 19 lassen [5723]
sich die Zauberkquadrate auf "
90 PRINT " dem Bildschi [4178]
rm darstellen."
100 : [174]
110 REM Eingabe der Groesse des Quadrates [2234]
120 : [174]
130 INPUT n:REM Seitenlaenge des Quadrates [3507]

140 CLS: IF n/2 = INT(n/2) THEN PRINT " [8115]
Dies ist keine ungerade Zahl.":GOTO 70:
REM Ueberpruefung ob ungerade
150 CLS: IF n<3 OR n>19 THEN PRINT " [6008]
Zahl liegt nicht im angegebenen Bereich.":
GOTO 70
160 CLS [91]
170 DIM Quadrat (n,n):REM * Anlegen einer [3364]
Matrix mit n Zeilen und n Spalten *
190 REM *** Belegen der Felder des Zauberk [2633]
quadrates ***
210 zeile=1: spalte=(n+1)/2 : REM * Festl [3781]
egen des Startfeldes *
220 Quadrat(zeile,spalte)=1 : REM * Start [3804]
feld wird mit 1 belegt *
```

```
230 FOR i=2 TO n*n : REM * Anzah [4189]
l der Zahlen wir festgelegt *
240 zeile=zeile-1:spalte=spalte+1 : REM [2464]
* Zug nach oben und nach recht *
250 IF zeile<1 THEN zeile=zeile+n : REM [3309]
* Randpruefung oben *
260 IF spalte>n THEN spalte=spalte-n : REM [4793]
* Randpruefung rechts *
270 IF Quadrat(zeile,spalte)=0 THEN GOTO 3 [5676]
20: REM * Wenn Feld leer ist, kan
n es belegt werden *
280 zeile=zeile+1:spalte=spalte-1 : REM [3639]
* Rueckzug, wenn Feld belegt *
290 zeile=zeile+1 : REM [2583]
* Ausweichzug nach unten *
300 IF zeile>n THEN zeile=zeile-n : REM [1917]
* Randpruefung unten *
310 IF spalte<1 THEN spalte=spalte+n : REM [3657]
* Randpruefung links *
320 Quadrat(zeile,spalte)=i : REM [4390]
* freies Feld wird belegt *
330 NEXT i [375]
350 REM Ausgabe des Zauberkquadrates [2804]
370 FOR zeile=1 TO n [805]
380 FOR spalte=1 TO n [1332]
390 PRINT USING "####"; Quadrat(zeile,spal [3415]
te);
400 NEXT spalte: PRINT:PRINT [993]
410 NEXT zeile [558]
430 REM *** Berechnen der Summe *** [1129]
450 su=(1+n^2)/2*n [1290]
460 PRINT TAB(5)"Die Summe betraegt "su ". [1766]
"
470 PRINT: INPUT "Noch ein Quadrat (j/n)"; [3034]
e$
480 IF e$="j" OR e$="J" THEN RUN ELSE END [2104]
```




PC Schneider International

Das führende Schneider Magazin berichtet Monat für Monat über Neuerscheinungen rund um die Schneider Computer. Jede Menge Tips & Tricks sowie ausgezeichnete Programme machen PC International für den Anwender unentbehrlich.

Folgende Sonderhefte sind noch erhältlich

Hier einige Kurzinformationen aus dem Inhalt:

Anwendungen:

U.a.
GBasic – ist das Tool zur effektiven Grafikprogrammierung. Die leistungsfähigen Eigenschaften von GBasic schaffen Programmierumgebungen, die sonst nur den absoluten Profis vorbehalten sind.
Unilab – das universelle Programm zum Erstellen und Drucken von Etiketten. Sonderfunktionen wie z.B. Unterstreichen am Bildschirm etc. machen Unilab zu einem komfortablen Anwenderprogramm.

Spiele:

U.a.
Guazuola – komplexes Spiel um Strategie und Macht. Als Herrscher von Guazuola haben Sie alle Fäden in der Hand, um Ihr Land zu Reichum zu verhelfen oder gegen Ihre direkten Neider vorzugehen. Viele originelle Optionen machen dieses Spiel zu einem echten CPC-Hit.
Crazy Stamps – besticht durch eine völlig neue Spielidee. Als Postbeamter haben Sie den Auftrag, Briefe mit dem entsprechenden Porto zu versehen und abzustempeln. Doch die Briefmarken spielen verrückt!
Tennis Manager – Boris Becker läßt grüßen... Sie übernehmen die Rolle des Managers eines Tennis-Stars und sind für den Geschäftsbetrieb verantwortlich.

Grundlagen:
Soundprogrammierung für jedermann. Nach Lektüre dieses Artikels sind Sie in der Lage, Ihrem CPC die herrlichsten Töne zu entlocken. Ergänzend zu diesem Thema finden Sie unseren PSG-Sound-Programmer sowie den Envelope-Generator, die ihre Kompositionen wesentlich erleichtern.
Diskettenwerkzeuge im Überblick: Sie erfahren alles wichtige über Disketten-Tools wie z.B. Kopierprogramme. Eine große Marktübersicht zeigt Ihnen alle im Handel befindlichen Programme.

Tips&Tricks:
 U.a.
Tonkopfjustage – nie wieder "Read error". Mit diesem Programm können Sie die Tonkopfjustierung ihres Kassettenlaufwerks vornehmen.
Disktool – komfortable Befehle für Vortex-Anwender.
Timeout – abdunkeln des Bildschirms nach längerer Arbeitspause. Diese Funktion gibt es i.d.R. nur auf Großrechenanlagen.

Turbo Pascal:
 Deluxe Data System – komfortable und universelle Dateiverwaltung. Auf der Databox ist dieses Programm auch als COM-File enthalten, d.h. Sie können das Programm unter CP/M benutzen!
 Windows: Hilfe zum Erstellen von Textfenstern unter Turbo-Pascal.

Spieltips:
 Ob Action-, Baller- oder Abenteuerspiel, aus jedem Genre haben wir interessante und nützliche Karten, Tips und Tricks zusammenggetragen. Z.B. finden Sie eine komplette Karte zu Starquake, Tips zum Überleben bei Starglider oder einen Leitfaden für die ersten Missionen von Academy.



Sonderheft 3/86:
 Reviews – Spiele – Anwendungen – ein wahres Hit-Sammelsorium birgt das CPC-Sonderheft 3/86. Die besten Spielprogramme im Überblick und viele Tips, Lösungen und Karten zu Computerspielen- und Abenteuern. Begeistern wird Sie auch der Flugsimulator – ein echter Leckerbissen zum Eintippen! Fantasy- und Adventure-Freunde werden sich über das erste Rollenspieladventure Monstergarten sicherlich genauso freuen, wie die Hardware-Freunde über die Echtzeituhr zum Selbstbau. Des weiteren gibt es viele tolle Programme aus den Bereichen Spiel, Anwendung und Utilities sowie fundierte Berichte über die effektive Interruptprogrammierung. Da ist für jeden etwas dabei. Und natürlich: Alle Programme sind auch auf Diskette oder Kassette erhältlich!

Sonderheft 4/87
 Programmiersprachen – Anwendungen in Turbo-Pascal und mannigfaltige Informationen stehen im Mittelpunkt des 4. CPC Sonderheftes. Mit über 200 Seiten praller CPC-Informationen, Tips und wertvollen Programme das ideale Sammelsorium für jeden CPC-Anwender. Interessiert? – dann sollten Sie sich schnell entscheiden, denn es sind nur noch wenige Restposten verfügbar.



Alle in den Heften enthaltenen Programme gibt es auch als Datenträger. Nur DMV hat den »Databox-Service«.

Bitte Bestell-Coupon auf Seite 95 benutzen

INFOCOM™

Kommt!

Die Legenden – die Klassiker – die Neuheiten

