

RUN

RUN

SONDER-HEFT 1

SH 1985

12,— DM Einzelpreis: ÖS 90,—; sfr. 12,—; L. 10 000; ptas 750,—; hfl. 14,50

180 SEITEN

SUPERPROGRAMME

**700 PROZENT
SCHNELLER
LADEN**

**VERBOTENE
GEHEIMBEFEHLE
MIT PROFIMONITOR
ZED SETZT ZEICHEN**

GRAFIK

**50 NEUE BEFEHLE
PLOTTER MIT BILLIGDRUCKER
HARDCOPY IM QUERFORMAT**

FLOPPY

**KAMPF DEM
DISKETTENCHAOS
DIRECTORY IN REIH
UND GLIED**

AUSLESE

**DIE BESTEN TIPS
UND TRICKS**



64 SPEZIAL

RUSH WARE

Online with the trend.

präsentiert

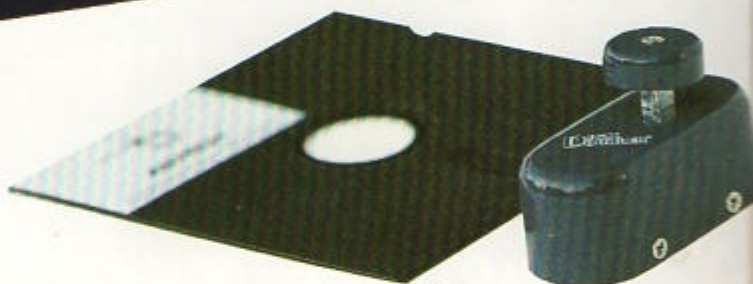


EXTAS
Externe Zehnertastatur
für C64, Atari, VC20

DATAREKORDER
Mit Verstärker
für C64 / VC20
auch als Atari-
Stereoversion
erhältlich



COBRA
Professioneller
Spielhallen-Joystick
Dauerfeuer, 3 Feuerknöpfe,
LED-Anzeige



TACKO
Diskettendoppler
aus Metall



PLEXI COVER
Hartplastik-Abdeckhaube für
C64, VC20, CPC464/664

CHALLENGER
Joystick mit
schaltbarem
Dauerfeuer



IHR VERTRIEBSPARTNER IN DEUTSCHLAND

**RUSH
WARE**
Online with the trend.

RUSHWARE Microhandelsgesellschaft mbH
An der Gumpgesbrücke 24 · D-4044 Kaarst 2

RUSHWARE-Produkte erhalten Sie in allen
führenden Computerfachgeschäften und
den Computershops der Warenhäuser

WWW.HOMECOMPUTERWORLD.COM

Randvoll!

Hier ist es! Das RUN-Sonderheft, randvoll mit neuen Programmen. Aus Hunderten von Listings haben wir die besten und originellsten ausgewählt. Ob Anfän-

ger oder Experte — RUN Spezial bringt für jeden etwas. Besondere Schwerpunkte sind Grafik und Floppy-Ansteuerung. Der Zeicheneditor ZED verändert Bildschirmzeichen fast von selbst. Mit über 50 Befehlen bearbeitet GRAFIK-BASIC gleich zwei Bildschirmseiten. Laden von der Floppy im Turbo-Tempo gibt's durch den FAST LOAD GENERATOR. DISKORG rückt dem Diskettenchaos zu Leibe. Auch Bandbesitzer kommen nicht zu kurz. Eine frei programmierbare Dateiverwaltung macht die Datasette zur Datenbank. Für alle, die einen Blick hinter die Kulissen werfen wollen, gibt es einen besonderen Leckerbissen: Als erster Maschinensprachemonitor knackt LINEASS alle verbotenen Assembler-Befehle. Abgerundet wird das Ganze durch eine Auslese der besten Tips und Tricks aus dem ersten RUN-Jahrgang. Prüfsummen bei allen Programmen sorgen für fehlerfreies Abtippen.

Viel Spaß beim Computern!

Wolfgang Schmabel

Peter Klerings

Für nur 29,90 Mark (plus 3,- Mark Versandkosten) gibt es alle Programme fertig auf zwei RUN-o-thek-Disketten zu kaufen. Was muß man tun? Den Verrechnungsscheck über den Gesamtbeitrag in einen Briefumschlag stecken und ab geht die Post an:
RUN-o-thek
Postfach 40 04 29, D-8000 München 40

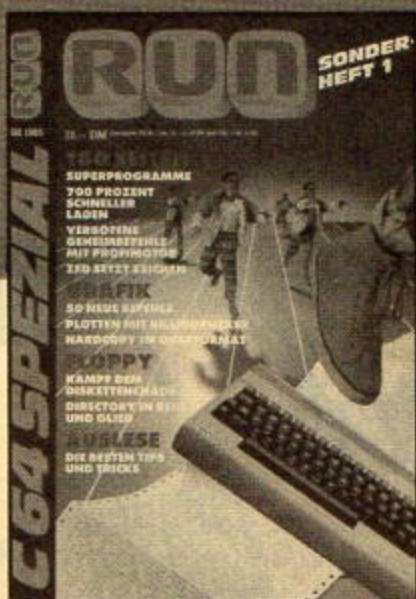


Foto: Roth

„Ich laufe meilenweit für ein gutes Programm“. Hier sind 180 Seiten Top-Programme, Tips und Tricks für den C64.

Aus der Redaktion

3

Korrektoren

| | |
|----------------|-------|
| Superkorrektor | 8—10 |
| Checker | 10—13 |

Super-Tool

| | |
|------------------------------|-------|
| Monitor knackt Geheimbefehle | 12—28 |
|------------------------------|-------|

Grafik

| | |
|-----------------------------|-------|
| Spitzen-Grafik ohne Mühe | 30—51 |
| Computergrafik für Bilanzen | 52—57 |
| ZED setzt Zeichen | 58—74 |
| Hardcopy groß rausgebracht | 76—79 |
| C64 mit besonderer Note | 80—83 |
| Plotter ohne Plotter | 84—94 |

Floppy

| | |
|-------------------------------------|---------|
| Siebenmal schneller in den Speicher | 95—98 |
| DISKORG gegen Diskettenchaos | 100—105 |
| Directory in Reih und Glied | 106—121 |
| Direkt-Datei-relativ-richtig | 122—125 |

Datsette

| | |
|--------------------|---------|
| Karte auf Kassette | 126—128 |
|--------------------|---------|

Anwendung

| | |
|----------------------------------|---------|
| Siebzehn und Vier | 130—135 |
| Einmaleins gleich Vierundsechzig | 136—139 |
| Dreiecksverhältnis | 140—141 |

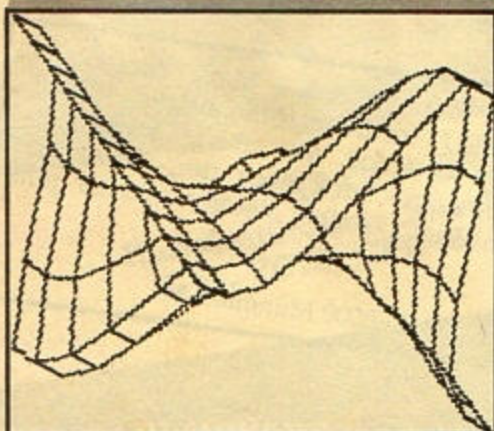
Basic-Erweiterung

| | |
|-------------------|---------|
| Input mit Komfort | 142—149 |
|-------------------|---------|

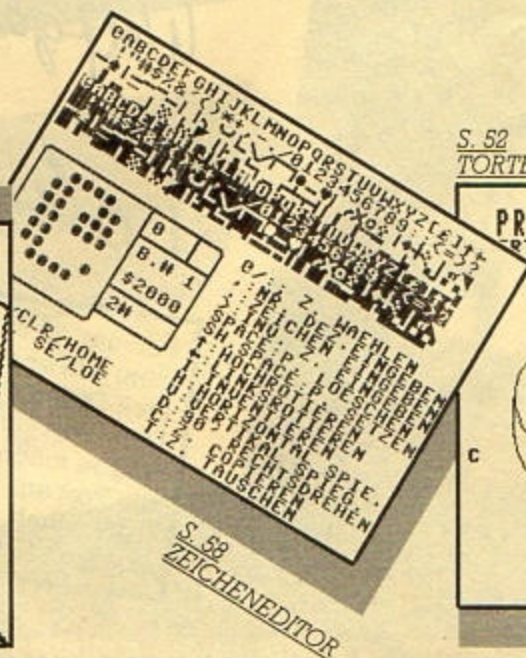
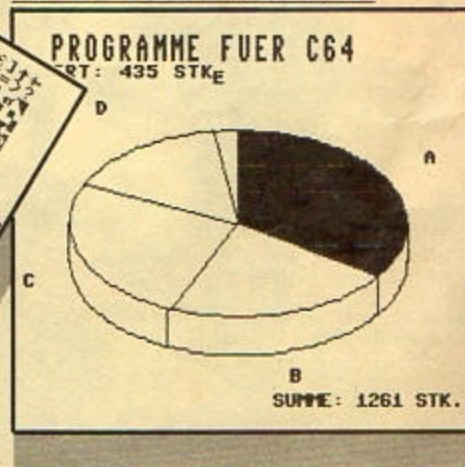
Tips & Tricks

| | |
|--------------------------------------|---------|
| Lokale Variablen für den C64 | 150—153 |
| Computer unter Kontrolle | 154—156 |
| Input ohne type mismatch | 157 |
| Zeilennummern im Zehnerschritt | 158—159 |
| Fehler ohne Programmabsturz | 160 |
| Speed up | 161 |
| Taube Taste | 161—162 |
| Flexibler Schreibschutz | 162—163 |
| Blocksave und Blockload | 163—164 |
| Speicherbelegung auf einen Blick | 164—165 |
| Vier Tasten — ein Dutzend Funktionen | 166—167 |
| Interpreter überlisten | 167—168 |
| Tipphilfe für Datas | 168—170 |
| Komfort-Listing | 171 |
| Antiratter-Routine | 172 |
| Shorties im Sechserpack | 172—178 |

S. 84
PLOTTER OHNE PLOTTER



S. 52
TORTENGRAFIK FÜR BILANZEN



BOSTON-COMPUTER BIETET PREIS- WERTE PERIPHERIE!



Der Profi für Einsteiger – Umsteiger – Aussteiger!

Schnellstes Basic der 8-Bit-Generation. Assembler-Programmierung. Komplettes Zubehörprogramm über B:C erhältlich! Steigen Sie für **DM 1499,-** jetzt ein.

Als **ACORN-Vertragspartner** garantieren wir für **Qualität und Service!**

ZUBEHÖRPROGRAMM COMMODORE C 64 Details, auf die Sie nicht verzichten sollten!

32K-CMOS-RAM-Platine mit Batterie-Puffer, lädt in Sekundenschnelle unabhängig von Floppy oder Datensette, mit 4 Steckplätzen, 20 Jahre Datensicherheit! **DM 198,-**

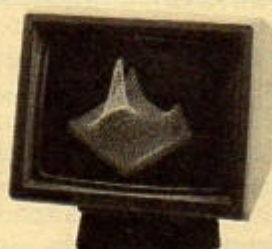
INTERPOD, extrem flexibles Interface mit IEEE-Bus und RS 232 C. Eigener 6502-Processor, der die Speicherkapazität Ihres C 64/VC 20 nicht belegt. **309,-**

OXFORD PASCAL, Disc Compiler f. C 64, Vollständige Implementierung des ANSI-Standards. Mit Residentem Compiler. Deutsches Handbuch erhältlich. Disk-Version **DM 199,-**

PRINTLINK: Epson-kompatible Centronics-Interface, für

C 64, Treiber-Software wahlweise: auf Diskette **DM 146,-** oder Cassette **DM 136,-**

Software auf Cassette für Ihr C 64-Abenteurer:
737 Flugsimulator **DM 48,-**
The HOBBIT, Abenteuer-Spiel **DM 54,-**
MANIC MINER, Arcade-Game **DM 38,-**
HUNCHBACK, Arcade-Game **DM 38,-**
REVENGE OF THE MUTANT CAMELS, Arcade **DM 38,-**
KILLERWATT, Arcade-Game **DM 38,-**
SHEEP IN SPACE, Arcade-Game **DM 38,-**
C 64-PROGRAMM-BIBLIOTHEKAR **DM 99,-**



DM 798,-

CABEL-FARBMONITORE für höchste Ansprüche! Super in Preis, Qualität und Leistung

Exzellente Farbauflösung 14"-Bildschirm – robuste Technik – integrierter Standfuß, dreh- und schwenkbar – Entmagnetisierknopf.

In jeder Klasse preisgünstig!!!
MC 3700-00
PAL/64-Farbmonitor, 430 Pixel (280 mm), Video-Composite-Signal **DM 798,-**
Anschluß-Kabel für C 64 **DM 18,-**

MC 3700-04
RGB-Farbmonitor, IBM-kompatibel, 660 Pixel, mittlere Auflösung, im Testmode monochrom, dark glass, entspiegelt, mit Anschlußkabel **DM 1495,-**

MC 3700 monochrom
monochromer PROFi-Monitor, extra hohe Auflösung über 1200 Line, IBM-kompatibel, 22 kHz **Industrie-Standard!** incl. Kabel **DM 698,-**

MC 3700-05
RGB-/PAL-Farbmonitor umschaltbar, 660 Pixel, Lochmaske 0,42 mm, voll kompatibel zu C 128 **DM 1398,-**

20"-Farbmonitore für Präsentationen auf Anfrage!
Information

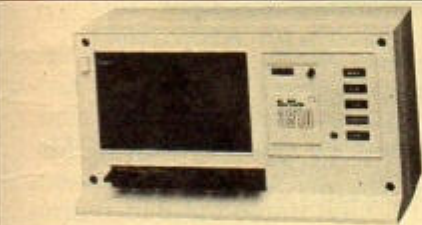
Kommunikations-Software!

- Acorn B • Apple • IBM • CBM • Kaypro • Sirius • Aprikot
- Menüsteuerung
- ECHO-Steuerung
- Datenpuffer
- 2 Betriebsmodi
- aktuelles Zubehör

natürlich: von Pace!
Information

NEU: Diskettenlaufwerk im Industrie-Standard, 3", einfache Dichte, 200 KB, voll kompatibel zu 5 1/4" für Acorn B **DM 919,-**

Neu: Apple II/+. Jetzt gibt's die Supercolor-Karte **SCC2000**, 16 Farben ohne „Apple“-Streifen!
RGB-Version **DM 419,-**
Video-Vers. **DM 439,-**



DM 69,-

Profi Daten Recorder – System MC 3810

Präzisionslaufwerk 5 Funktionsanzeigen: „LOAD“, „SAVE“, „READY“, „MOTOR CONTROL“, „DATA“ Übertragungsrate 1200 Baud Bandzahlwerk, ölgedämpfter Auswurfschacht, Cassettendeckel abnehmbar zur Justierung und Reinigung des Tonkopfes

Universell einsetzbar für die bekanntesten Computer-Typen!

Preis incl. Interface, (*) = incl. Netzteil
• C64, VC20, C128, etc.

- Atari 600 (*) **DM 69,-**
- Atari 800 (*) **DM 79,-**
- Sinclair Spectrum (*) **DM 79,-**
- Acorn B/electron (*) **DM 79,-**
- Sanyo Laser 210 (*) **DM 79,-**

Jetzt auch für NDR-Klein-Computer (*) **DM 129,-**
Bezugsquelle für Spectra-video SVI 728 und ORIC

Schneider

Mini-Büro
Software-Paket mit:
- Textverarbeitung
- Datenbank
- Kalkulation, incl. Graphische Darstellung



Menüsteuerung, leicht verständliche Anleitung in Deutsch!
auf Cassette **DM 98,-**
auf Diskette 3" **DM 129,-**

* = eingetragenes Zeichen der Schneider Computer Division

SHINWA Matrixdrucker CP 80 – Serie:

80 Zeichen, bidirektional, 228 ASCII-Zeichensatz, Traktor-Führung und Friktionswalze für Einzelblatteinzug, 640 Punkte pro Linie, Bit Image Graphik, 8 Betriebsmodi, komplett mit Farbbandcassette und Anwenderhandbuch



DM 698,-

Centronics-Schnittstelle **DM 698,-**
CBM-Standard, für C 64/C 128 **DM 698,-**
IBM-Standard, A 4 Format **DM 898,-**
IBM-Standard, A 3 Format **DM 1389,-**

Anschlußkabel:
Acorn B **DM 63,-**
C 64 **DM 63,-**
CPC 464 **DM 63,-**
IBM-PC's **DM 98,-**

Die Peripherie, die keinem fehlen sollte!

Bestellung:

Bestellen Sie durch den gewünschten Artikel bei:
Boston Computer Handelsges. m. b. H.
Rosenheimer Straße 145 a,
D-8000 München 80
Telefon (0 89) 49 10 73
Mo.-Do. 9-12, 13-17 Uhr, Fr. bis 14 Uhr

per Nachnahme (nur BRD) mit beigefügtem Verr.-Scheck

Name: _____
Straße: _____
PLZ/Ort: _____
Datum/Unterschrift: _____

Auslandslieferungen erfolgen nur gegen Vorauskasse (Scheck) zuzügl. Porto. Unter DM 50,- Warenwert berechnen wir RUN DM 3.50 Bearbeitungsgebühr. Alle Angebote erfolgen freibleibend. Preise sind incl. MwSt. Wir gewähren 6 Monate Garantie auf Hardware. Außerdem Sonderkonditionen für Bildungseinrichtungen. Fachhandelsanfragen willkommen!

Besuchen Sie uns in unserem Ausstellungsraum!



PROFI BOSTON COMPUTER KONZERT

Fotokopieren oder ausschneiden und senden an: Boston Computer Handelsges. m. b. H., Rosenheimer Str. 145 a, 8000 München 80.

Superkorrektor

Ein Korrektor hilft beim Abtippen von Listings. Damit er das kann, muß man ihn vor der Eingabe eines Programms eintippen.

Wer sich die Zeit sparen will und doch einen richtigen Korrektor in seiner Programmbibliothek haben möchte, schreibt an RUN, Kennwort „Korrektor“. Für nur 10,00 Mark bekommt er die RUN-Aktionsdiskette. Wer die RUN-o-thek bestellt, kann sich das sparen: Auf der Programmdiskette sind alle Programme eines Heftes enthalten, also auch die Korrektoren.

Funktionsweise

Hinter jeder Listing-Zeile im Heft findet man eine (meist) vierstellige Zahl. Das ist die Prüfsumme. Sie wird nicht eingegeben, sondern mit der Statuszeile verglichen, nachdem man RETURN gedrückt hat. Stimmt die Zahl in der Statuszeile mit der Prüfsumme überein, weiß man, daß die Zeile richtig eingetippt ist. Zwei Dinge sind wichtig:

1) Das eingetippte Programm save und nicht unter dem Korrektor starten.

2) Die Anweisungen in eckigen Klammern nicht mit eingeben!!! Es sind die „übersetzten“ Steuerzeichen, die die Lesbarkeit erhöhen. „DOWN“ ist Cursor abwärts, „UP“ aufwärts; „SPACE“ die Leertaste. Die Zahlen bedeuten die Häufigkeit der Betätigung der jeweiligen Taste. Ansonsten hält sich die Übersetzung an die Aufschrift der jeweiligen Tasten (zum Beispiel RED oder CLR). CBM ist die Commodoretaste.

Besonderheiten

1) Unsinnige Zeilen besitzen keine Prüfsummen; so Zeilen, die nur aus Doppelpunkt(en) und/oder REMarks bestehen.

2) Spaces außerhalb von Anführungszeichen werden nicht berücksichtigt. Es ist also egal, ob man PRINT SPC(5) oder PRINTSPC(5) eingibt.

Korrektoren helfen Eingabefehler vermeiden.

Doch nur, wenn man genau versteht, wie sie funktionieren.

RUN-Programmierer Valentin Hilbig hat deshalb alle Einzelheiten beschrieben.

3) Spaces innerhalb von Anführungszeichen gehen in die Prüfsumme ein!

4) Spaces in DATA-Zeilen werden nur dann berücksichtigt, wenn sie (auch hier) innerhalb von Anführungszeichen liegen. Also Achtung: DATA GRÜN, ROT, BLAU, GELB, ... ist verschieden von DATA, GRÜN, ROT, BLAU, GELB, ... obwohl gleiche Prüfsummen ausgegeben werden. (Das liegt daran, das Spaces nach einem DATA-Inhalt nicht überlesen werden und dies der Korrektor aus Platzgründen nicht berücksichtigen kann.)

5) Zwischen Listingteil und Prüfsumme befindet sich niemals ein einzugebender Space. Ein solcher Space wird immer an den Anfang der folgenden Zeile gestellt.

6) Die Zeilen sind verwechslungssicher, das soll heißen, daß erst über acht Zeichen hinweg gewechselt werden kann: PRINT "OSTERFEST" und PRINT "TSTERFESO" haben gleiche Prüfsummen.

7) Zeilennummern werden nicht in die Prüfsumme einbezogen! Also bitte überprüfen, ob die Zeilennummer stimmt. Beim C64 wer-

den die Zeilennummern extra deswegen mit ausgegeben.

8) REMarks können vollständig weggelassen werden (eine der gewollt eingebauten Spezialoptionen). Sollte in der Zeile ein :REM stehen, so muß der „:“ jedoch eingegeben werden, um die korrekte Prüfsumme zu erhalten.

9) Es ist nicht möglich, von der falschen Prüfsumme aus auf den Fehler zurückzurechnen.

10) Außerdem beherrscht der C64-Korrektor Simon's Basic, indem man das erste Komma in der letzten Zeile wegläßt.

Korrektor auch

mit Simon's Basic

11) Auf alten C64-Computern kann es passieren, daß die Korrektorzeilen nicht sichtbar sind. Das liegt an den falschen ROMs, die den Bildschirm bei einem SHIFT/CLR-Home mit Bildschirmhintergrund- statt mit der -vordergrundfarbe belegen. Die oberste Zeile ist vor dem Anfahren mit dem Cursor sicher.

RUN/STOP-Restore hat keine Auswirkung auf den Korrektor.

Man kann ausprobieren: RUN/STOP-Restore berührt die oberste Zeile nicht.

Prellt die RETURN-Taste, passiert beim C64 nichts. Die letzte Prüfsumme steht immer auf dem Bildschirm. Die Bildschirmhintergrundfarbe ist schwarz.

SYS49152

Schaltet ein/aus. Das kann man sehr deutlich an der Bildschirmfarbe und an der obersten Zeile sehen.

Weil eine Erweiterung vorgesehen ist, fehlen am Ende der obersten Korrekturzeile acht Zeichen. Der Korrektor belegt etwas mehr als zwei Pages (circa 550 Bytes) im

Print "Q":

```

10 RESTORE :FOR A=49152 TO 49629:POKE A,234:NEXT :PRINT "MOMENT!
20 F=0:READ A:Y=PEEK (63)+PEEK (64)*256:IF A=0 THEN 200
30 READ B,C:FOR D=A TO B:READ A$:IF LEN (A$)-2 THEN 90
40 E=ASC (A$)-48-(7 AND A$>"0"):IF A$<"0" OR A$>"G" OR A$>":" AND A$<
  "A" THEN 90
50 A$=MID$ (A$,2):E=E*16+ASC (A$)-48-(7 AND A$>"0")
60 IF A$<"0" OR A$>"F" OR A$>"9" AND A$<"A" THEN 90
70 F=F+E:POKE D,E:NEXT :IF F=C THEN 20
80 PRINT "PRUEFSUMMENFEHLER: SOLL"C", IST"F:GOTO 100
90 PRINT "EINGABEFehler: "A$:Y=0
100 Z=PEEK (63)+PEEK (64)*256:A$=STR$ (Y)+STR$ (-Z):IF Y=0 THEN A$=STR$
  (Z)
110 PRINT "[DOWN]LIST"A$"[2UP]":POKE 198,1:POKE 631,13:END
200 PRINT "[DOWN]FERTIG.":PRINT "[DOWN]SOLL DAS MASCHINENPROGRAMM
  ABGESPEICHERT
210 INPUT "[UP]WERDEN[2SPACES]J[3LEFT]":A$:IF A$<>"J" THEN 300 <
220 SYS 57812@:49152 KORREKTOR",8:POKE 193,0:POKE 194,192:POKE 174,222
230 POKE 175,193:SYS 62957:VERIFY "49152 KORREKTOR",8,1
300 A$="":INPUT "[DOWN]KORREKTOR AKTIVIEREN? J[3LEFT]":A$:IF A$="J" THEN
  SYS 49152
999 REM HAUPTPROGRAMM
1000 DATA 49187,49208,2342,A9,AE,A2,C0,8D,02,03,8E,03,03,00,A0,1F,B9,E2
1001 DATA C1,99,00,04,88,10,F7
1002 DATA 49323,49387,9136,4C,31,EA,78,A9,2E,A2,C0,8D,14,03,8E,15,03,58
1003 DATA A9,00,8D,E1,C1,20,60,A5,8C,E1,C1,86,7A,84,7B,AC,E0,C1,F0,13,A0
1004 DATA 10,89,E1,C1,99,F1,C1,A9,A0,99,E1,C1,88,D0,F2,8C,E0,C1,20,73,00
1005 DATA AA,F0,D3,88,84,3A,90,55
1006 DATA 49467,49629,19950,4C,96,A4,4C,08,AF,20,6B,A9,20,79,A5,84,0B,A0
1007 DATA 00,84,08,84,45,84,46,84,09,C8,BE,FF,01,F0,38,A5,08,30,0C,E0,20
1008 DATA F0,F2,E0,3A,F0,0C,E0,8F,F0,28,E0,22,D0,02,49,80,09,01,85,08,8A
1009 DATA A6,09,F0,06,0A,69,00,CA,D0,FA,18,65,45,85,45,90,02,E6,46,E6,09
1010 DATA A5,09,29,07,85,09,10,C2,46,08,90,21,A0,FF,8C,E0,C1,A2,14,20,B9
1011 DATA C1,C8,A9,BD,99,E2,C1,A9,BE,C8,99,E2,C1,C8,A2,45,20,B9,C1,A9,BB
1012 DATA 99,E2,C1,4C,A4,A4,84,08,B5,00,85,63,B5,01,85,62,A2,90,38,20,49
1013 DATA BC,20,DF,BD,A4,08,AA,C8,BD,00,01,F0,08,09,80,99,E2,C1,E8,D0,F2,
  60
1999 REM EIN/AUSSCHALTER
2000 DATA 49152,49186,5131,AD,03,03,C9,C0,D0,09,AD,DE,C1,AE,DF,C1,4C,27
2001 DATA C0,8D,DF,C1,AD,02,03,8D,DE,C1,A9,A0,A2,1F,9D,E2,C1,CA,10,FA
2999 REM EINGABESICHERUNG
3000 DATA 49285,49322,4626,AD,E1,C1,D0,21,8D,20,D0,8D,21,D0,A5,DA,30,0E
3001 DATA 09,80,85,DA,A0,27,A9,20,99,28,04,88,10,FA,A6,D6,D0,05,A9,11,20,
  35,EB
4998 REM ENDE BZW. SIMON'S ANPASSUNG
4999 REM DUCH WEGLASSEN DES 1. KOMMAS:
5000 DATA ,49293,49295,702,EA,EA,EA,

```

Korrektor für den C64: oben Diskettenversion, unten Änderungen für Datasette

```

200 PRINT "[DOWN]FERTIG.":PRINT "[DOWN]SOLL DAS MASCHINENPROGRAMM AUF
  KASSETTE
210 INPUT "ABGESPEICHERT WERDEN[2SPACES]J[3LEFT]":A$:IF A$<>"J" THEN 300
220 SYS 57812"49152 KORREKTOR",1,1:POKE 193,0:POKE 194,192:POKE 174,222
230 POKE 175,193:SYS 62957:PRINT "BITTE DIE KASSETTE ZURUECKSPULEN UND
240 INPUT "RETURN DRUECKEN":A$:VERIFY "49152 KORREKTOR",1

```

C000-Bereich des Computers. Simon's Basic verwendet diesen beim Grafikbetrieb! Also: vor Grafik Korrektor ausschalten! Laden und Saven.

Die Abspeicherfunktion des Korrektors speichert nicht das BASIC-Programm ab, sondern das generierte Maschinenprogramm.

LOAD "49152 KORREKTOR", 8,1

beziehungsweise LOAD "49152 KORREKTOR",1 (genügt hier)

dann:

NEW

dann:

SYS49152 (Einschalten)

Das abgespeicherte Maschinenprogramm wird automatisch ge-VERIFYt. Deshalb muß man die Kassette nach dem Abspeichern

zurückspulen und dann RETURN drücken.

Da SIMON's-BASIC eine schwarze Schrift besitzt, muß die Bildschirmhintergrundfarbe auf einen anderen Wert gesetzt werden. Dies übernimmt die SIMON's-Anpassung, indem man das erste Komma auf Zeile 5000 wegläßt.

(Valentin Hilbig)

Checker für Maschinenprogramme

Der Checker reduziert die Tipparbeit und weist während des Eintippens auf Fehler hin. Außerdem entfällt das langwierige Einlesen der Datas durch den Computer. Der Checker ist eine Art Texteditor für Maschinenprogramme, der jede einzelne „Programmzeile“ (zu je 8 Byte) anhand einer miteingebenden Prüfsumme auf Richtigkeit testet und sich bei Tippfehlern akustisch bemerkbar macht.

Anwendung

Die Daten werden in hexadezimaler Form eingegeben. Dazu stehen die Tasten 0-9 und A-F zur Verfügung. Zur Steuerung des Programms kommen noch folgende Tasten dazu:

—Y—

Bevor man Daten eingibt, muß man mit Y die Startadresse eingeben, ab der das Programm einge-

lesen werden soll. Beim Saven wird dieser Wert dann als Anfangsadresse genommen.

—INST/DEL—

löscht ein einzelnes Zeichen oder fügt eines ein.

—CLR—

löscht eine ganze Zeile und läßt die Zeile zur Veränderung frei.

—K—

löscht ebenfalls eine Zeile, jedoch verbleibt keine Leerzeile, son-

| | | |
|---|--|------|
| 60 GOTO 10000 | | 565 |
| 70 : | | |
| 1000 DATA 11,8,193,7,158,50,48,54,52,,,,,,169,,133,253,133,254 | | 5818 |
| 1010 DATA 32,248,15,32,138,9,169,63,141,138,2,169,24,141,5,220 | | 5341 |
| 1020 DATA 32,47,8,32,163,10,76,38,8,32,39,10,169,,133,198,165,198 | | 6185 |
| 1030 DATA 240,252,32,39,10,173,119,2,96,169,,141,32,208,169,14 | | 5394 |
| 1040 DATA 141,33,208,160,,162,216,132,27,132,31,134,28,134,32,169 | | 6353 |
| 1050 DATA 4,133,30,132,29,169,,145,27,169,160,145,29,200,208,245 | | 6000 |
| 1060 DATA 230,28,230,30,232,224,220,208,236,162,23,160,85,169,6 | | 5032 |
| 1070 DATA 145,31,200,145,31,200,200,192,108,144,243,200,200,169 | | 4864 |
| 1080 DATA 6,145,31,200,145,31,165,31,24,105,40,133,31,144,2,230 | | 5562 |
| 1090 DATA 32,202,208,218,160,5,162,48,169,176,153,,4,138,9,128 | | 5354 |
| 1100 DATA 153,1,4,200,200,200,232,224,56,208,237,162,6,189,238 | | 5864 |
| 1110 DATA 8,153,,4,200,202,208,246,96,160,80,169,4,132,27,133,28 | | 5875 |
| 1120 DATA 169,23,133,2,165,97,133,25,165,98,133,26,32,245,8,165 | | 6098 |
| 1130 DATA 25,24,105,8,133,25,144,2,230,26,165,27,24,105,7,133,27 | | 6207 |
| 1140 DATA 144,2,230,28,198,2,208,227,96,141,149,147,139,136,131 | | 5376 |
| 1150 DATA 165,253,24,101,25,72,165,254,101,26,32,102,9,104,32,102 | | 5830 |
| 1160 DATA 9,32,131,9,165,25,133,29,165,26,24,105,64,133,30,165 | | 5261 |
| 1170 DATA 26,133,31,165,25,10,38,31,10,38,31,169,80,133,32,165 | | 5412 |
| 1180 DATA 25,74,74,74,41,7,168,185,152,9,160,,49,31,240,34,160 | | 5657 |
| 1190 DATA ,132,13,177,29,32,102,9,32,131,9,164,13,200,192,8,208 | | 5476 |
| 1200 DATA 239,32,42,11,32,131,9,32,131,9,32,102,9,76,101,9,162 | | 5406 |
| 1210 DATA 28,160,,169,160,145,27,32,131,9,202,208,244,96,72,74 | | 5768 |
| 1220 DATA 74,74,74,32,116,9,104,41,15,76,116,9,24,105,246,144,2 | | 6007 |
| 1230 DATA 105,6,105,186,41,191,160,,145,27,230,27,208,2,230,28 | | 5840 |
| 1240 DATA 96,169,15,141,24,212,32,48,10,32,65,8,76,169,8,128,64 | | 5456 |
| 1250 DATA 32,16,8,4,2,1,169,80,160,4,133,29,132,30,162,22,160,39 | | 5679 |

| | | | |
|------|------|---|------|
| 1260 | DATA | 132,32,169,79,133,31,164,31,177,29,164,32,145,29,198 | 5772 |
| 1270 | DATA | 31,198,32,16,242,165,29,24,105,40,133,29,144,2,230,30 | 5090 |
| 1280 | DATA | 202,208,220,169,192,133,27,169,7,133,28,165,97,24,105 | 6223 |
| 1290 | DATA | 176,133,25,165,98,105,,133,26,76,245,8,169,152,160,7 | 5664 |
| 1300 | DATA | 133,29,132,30,162,23,169,39,133,31,169,79,133,32,164 | 5652 |
| 1310 | DATA | 31,177,29,164,32,145,29,198,32,198,31,16,242,165,29,56 | 6457 |
| 1320 | DATA | 233,40,133,29,176,2,198,30,202,208,220,169,80,133,27 | 5076 |
| 1330 | DATA | 169,4,133,28,165,97,133,25,165,98,133,26,76,245,8,164 | 6233 |
| 1340 | DATA | 99,177,101,73,5,145,101,96,169,,133,97,133,98,169,,133 | 6063 |
| 1350 | DATA | 99,133,100,169,85,133,101,169,216,133,102,96,198,100 | 5339 |
| 1360 | DATA | 48,12,165,101,56,233,40,133,101,176,2,198,102,96,230 | 5284 |
| 1370 | DATA | 100,165,97,56,233,8,133,97,176,11,198,98,16,7,169,,133 | 6518 |
| 1380 | DATA | 97,133,98,96,76,230,9,230,100,165,100,201,23,240,12,165 | 5636 |
| 1390 | DATA | 101,24,105,40,133,101,144,2,230,102,96,198,100,165,97 | 5447 |
| 1400 | DATA | 24,105,8,133,97,144,2,230,98,165,98,201,63,208,11,165 | 5517 |
| 1410 | DATA | 97,201,8,208,5,169,,133,97,96,76,160,9,201,17,208,3,76 | 6123 |
| 1420 | DATA | 110,10,201,145,208,3,76,69,10,201,29,208,3,76,104,11 | 5248 |
| 1430 | DATA | 201,157,208,3,76,58,11,201,13,208,3,76,23,12,201,48,144 | 6170 |
| 1440 | DATA | 15,201,58,144,8,201,85,144,7,201,71,176,3,76,152,11,201 | 6055 |
| 1450 | DATA | 19,208,3,76,155,12,201,147,208,3,76,149,12,201,20,208 | 4747 |
| 1460 | DATA | 3,76,230,12,201,148,208,3,76,191,12,201,72,208,6,32,48 | 6109 |
| 1470 | DATA | 10,76,189,8,201,73,208,3,76,17,13,201,75,208,3,76,133 | 5756 |
| 1480 | DATA | 13,201,83,208,3,76,13,16,201,76,208,3,76,121,15,201,89 | 5536 |
| 1490 | DATA | 208,3,76,104,16,201,80,208,3,76,178,16,96,162,8,160, | 5756 |
| 1500 | DATA | 24,169,,113,29,73,255,200,202,208,248,96,198,99,165,99 | 5642 |
| 1510 | DATA | 16,3,230,99,96,201,2,240,243,201,5,240,239,201,8,240 | 5371 |
| 1520 | DATA | 235,201,11,240,231,201,14,240,227,201,17,240,223,201 | 4814 |
| 1530 | DATA | 20,240,219,201,25,208,223,169,22,133,99,96,230,99,165 | 4793 |
| 1540 | DATA | 99,201,28,208,3,198,99,96,201,2,240,241,201,5,240,237 | 5354 |
| 1550 | DATA | 201,8,240,233,201,11,240,229,201,14,240,225,201,17,240 | 5533 |
| 1560 | DATA | 221,201,20,240,217,201,23,208,175,169,26,133,99,96,72 | 5095 |
| 1570 | DATA | 32,160,12,104,41,63,9,128,164,99,145,29,76,104,11,160 | 5300 |
| 1580 | DATA | ,177,27,41,127,201,48,176,2,105,58,233,48,76,131,9,32 | 5861 |
| 1590 | DATA | 168,11,10,10,10,10,133,13,32,168,11,5,13,96,165,101,133 | 6322 |
| 1600 | DATA | 27,165,102,41,3,9,4,133,28,32,4,12,32,232,11,185,152 | 5139 |
| 1610 | DATA | 9,133,103,73,255,160,,49,31,145,31,96,165,30,56,233,64 | 6017 |
| 1620 | DATA | 133,31,165,29,10,38,31,10,38,31,169,60,133,32,165,29 | 6053 |
| 1630 | DATA | 74,74,74,41,7,168,96,165,100,10,10,10,24,101,97,133,29 | 5796 |
| 1640 | DATA | 185,98,105,,105,64,133,30,96,32,200,11,160,,132,2,32 | 5734 |
| 1650 | DATA | 185,11,32,131,9,164,2,145,29,200,192,8,208,239,32,131 | 5535 |
| 1660 | DATA | 9,32,131,9,32,185,11,133,13,32,42,11,197,13,240,6,32 | 5677 |
| 1670 | DATA | 97,12,133,99,96,165,103,160,,17,31,145,31,169,,133,99 | 5660 |
| 1680 | DATA | 32,110,10,169,160,32,110,12,160,20,32,134,12,76,143,12 | 5589 |
| 1690 | DATA | 169,60,32,110,12,160,80,32,134,12,76,143,12,141,1,212 | 5281 |
| 1700 | DATA | 169,,141,,212,169,17,141,5,212,169,241,141,6,212,169 | 5674 |
| 1710 | DATA | 17,141,4,212,96,162,,202,208,253,136,208,248,96,169, | 6077 |
| 1720 | DATA | 141,4,212,96,32,200,11,32,189,8,169,,133,99,96,165,101 | 6039 |
| 1730 | DATA | 133,29,165,102,41,3,9,4,133,30,96,,1,3,4,6,7,9,10,12 | 5291 |
| 1740 | DATA | 13,15,16,18,19,21,22,26,27,32,160,12,160,27,177,29,201 | 5723 |
| 1750 | DATA | 160,208,27,162,17,188,173,12,196,99,240,14,132,13,202 | 5508 |
| 1760 | DATA | 188,173,12,177,29,164,13,145,29,208,235,169,160,145,29 | 7000 |
| 1770 | DATA | 96,32,160,12,165,99,240,248,162,17,221,173,12,240,3,202 | 5415 |
| 1780 | DATA | 16,248,188,173,12,177,29,202,188,173,12,145,29,232,232 | 5712 |
| 1790 | DATA | 224,18,208,239,169,160,160,27,145,29,76,58,11,32,4,12 | 5386 |
| 1800 | DATA | 169,240,133,31,169,127,133,32,162,7,169,15,133,13,138 | 5738 |
| 1810 | DATA | 168,177,31,164,13,145,31,198,13,202,16,249,165,32,197 | 6297 |
| 1820 | DATA | 30,208,6,165,31,197,29,240,13,165,31,56,233,8,133,31 | 5443 |
| 1830 | DATA | 176,216,198,32,208,212,32,232,11,166,31,185,117,13,133 | 6336 |
| 1840 | DATA | 13,185,125,13,160,,49,31,74,8,133,2,177,31,37,13,5,2 | 5404 |
| 1850 | DATA | 145,31,232,240,8,40,126,,60,8,76,101,13,32,189,8,40,96 | 5648 |

| | | |
|-----------|---|------|
| 1860 DATA | ,128,192,224,240,248,252,254,255,127,63,31,15,7,3,1,32 | 6670 |
| 1870 DATA | 4,12,160,7,162,15,132,13,138,168,177,29,164,13,145,29 | 5840 |
| 1880 DATA | 202,198,13,16,243,165,29,24,105,8,133,29,144,2,230,30 | 5294 |
| 1890 DATA | 165,29,201,248,208,220,165,30,201,127,208,214,32,4,12 | 4877 |
| 1900 DATA | 32,232,11,162,255,24,8,40,62,,60,8,202,228,31,208,246 | 4919 |
| 1910 DATA | 185,117,13,61,,60,133,2,189,,60,40,42,57,125,13,5,2,157 | 6012 |
| 1920 DATA | ,60,76,189,8,162,,189,,60,201,255,208,3,232,208,246,134 | 5648 |
| 1930 DATA | 2,162,255,228,2,240,63,189,,60,208,5,202,224,255,208 | 5743 |
| 1940 DATA | 242,166,2,189,,60,133,2,160,,165,2,57,152,9,240,3,200 | 5434 |
| 1950 DATA | 16,246,132,13,169,,133,98,138,10,38,98,10,38,98,10,38 | 5232 |
| 1960 DATA | 98,5,13,10,38,98,10,38,98,10,38,98,133,97,32,54,10,56 | 5623 |
| 1970 DATA | 96,160,7,189,,60,217,117,13,240,5,136,16,248,48,188,132 | 6104 |
| 1980 DATA | 2,169,,133,28,138,10,38,28,10,38,28,10,38,28,5,2,10,38 | 5293 |
| 1990 DATA | 28,10,38,28,10,38,28,133,27,165,28,24,105,64,133,28,24 | 6114 |
| 2000 DATA | 96,134,13,133,29,132,30,165,99,72,165,100,72,165,101 | 5274 |
| 2010 DATA | 72,165,102,72,160,38,169,160,153,,4,136,16,248,160,,177 | 6027 |
| 2020 DATA | 29,240,9,32,42,15,153,,4,200,208,243,162,,132,99,169 | 5394 |
| 2030 DATA | 6,153,,216,153,1,216,200,232,228,13,208,247,169,216,133 | 6303 |
| 2040 DATA | 102,169,,133,100,133,101,133,32,32,47,8,72,41,96,208 | 5632 |
| 2050 DATA | 26,104,201,13,240,45,201,20,208,239,165,32,240,235,198 | 5890 |
| 2060 DATA | 99,198,32,169,160,164,99,153,,4,16,222,164,32,104,153 | 5141 |
| 2070 DATA | ,2,196,13,240,212,164,99,32,42,15,153,,4,230,32,230,99 | 5496 |
| 2080 DATA | 208,198,164,32,169,,153,,2,104,133,102,104,133,101,104 | 5278 |
| 2090 DATA | 133,100,104,133,99,96,72,169,32,160,15,162,15,32,102 | 6071 |
| 2100 DATA | 14,152,162,,160,2,32,189,255,104,168,162,8,169,8,32,186 | 5729 |
| 2110 DATA | 255,32,192,255,169,1,76,64,15,70,73,76,69,78,65,77,69 | 6297 |
| 2120 DATA | 32,,72,74,74,74,74,170,104,41,31,29,56,15,96,,160 | 5721 |
| 2130 DATA | 128,192,64,224,192,224,169,8,32,180,255,169,111,32,150 | 5778 |
| 2140 DATA | 255,160,39,169,160,153,,4,169,,153,,216,136,16,243,160 | 5711 |
| 2150 DATA | ,32,165,255,201,13,240,9,32,42,15,153,,4,200,208,240 | 5641 |
| 2160 DATA | 173,,4,13,1,4,41,15,72,32,171,255,104,96,169,,32,253 | 5827 |
| 2170 DATA | 14,208,110,162,8,32,198,255,169,,133,144,133,27,32,207 | 5735 |
| 2180 DATA | 255,133,253,32,207,255,141,254,,32,248,15,169,60,133 | 5441 |
| 2190 DATA | 28,169,,133,2,133,29,169,64,133,30,165,144,208,53,160 | 5622 |
| 2200 DATA | ,32,207,255,145,29,200,192,8,208,246,160,,166,2,177,27 | 5993 |
| 2210 DATA | 29,152,9,145,27,230,2,232,224,8,208,6,169,,133,2,230 | 5739 |
| 2220 DATA | 27,165,29,24,105,8,133,29,144,207,230,30,165,30,201,128 | 5698 |
| 2230 DATA | 208,199,32,204,255,169,8,32,195,255,32,48,10,76,146,9 | 5928 |
| 2240 DATA | 169,,133,198,165,198,240,252,208,235,160,,152,153,,60 | 6024 |
| 2250 DATA | 200,208,250,96,32,204,255,169,8,32,195,255,76,146,9,32 | 5545 |
| 2260 DATA | 222,13,176,83,165,27,72,165,28,72,169,1,32,253,14,133 | 5946 |
| 2270 DATA | 31,104,133,28,104,133,27,165,31,208,197,169,,133,29,169 | 6163 |
| 2280 DATA | 64,133,30,162,8,32,201,255,165,253,32,210,255,165,254 | 5112 |
| 2290 DATA | 32,210,255,165,29,197,27,208,6,165,30,197,28,240,148 | 5435 |
| 2300 DATA | 160,,177,29,32,210,255,208,192,8,208,246,165,29,24,105 | 6007 |
| 2310 DATA | 8,133,29,144,223,230,30,208,219,76,189,8,162,4,160,16 | 5469 |
| 2320 DATA | 169,136,32,102,14,169,15,133,27,169,4,133,28,32,185,11 | 6335 |
| 2330 DATA | 141,254,,32,185,11,141,253,,76,146,9,65,78,70,65,78,71 | 6281 |
| 2340 DATA | 83,65,68,82,69,83,83,69,32,,24,105,246,144,2,105,6,105 | 5670 |
| 2350 DATA | 58,76,210,255,72,74,74,74,74,32,152,16,104,41,15,76,152 | 6199 |
| 2360 DATA | 16,32,222,13,176,174,169,,32,189,255,169,4,170,160,,32 | 5597 |
| 2370 DATA | 186,255,32,192,255,162,4,32,201,255,180,6,169,32,32,210 | 6089 |
| 2380 DATA | 255,136,208,248,162,7,160,,152,32,164,16,169,32,32,210 | 6181 |
| 2390 DATA | 255,200,202,16,243,160,5,185,108,17,32,210,255,136,16 | 5741 |
| 2400 DATA | 247,169,13,32,210,255,32,210,255,165,253,133,31,165,254 | 6483 |
| 2410 DATA | 133,32,169,,133,29,169,64,133,30,165,29,197,27,208,6 | 6322 |
| 2420 DATA | 165,30,197,28,240,78,165,32,32,164,16,165,31,32,164,16 | 5930 |
| 2430 DATA | 169,32,32,210,255,32,210,255,160,,177,29,32,164,16,169 | 5882 |
| 2440 DATA | 32,32,210,255,200,192,8,208,241,169,32,32,210,255,32 | 5182 |
| 2450 DATA | 210,255,32,42,11,32,164,16,169,13,32,210,255,165,31,24 | 5974 |

```

2460 DATA 105,8,133,31,144,2,230,32,165,29,24,105,8,133,29,144      5194
2470 DATA 170,230,30,208,166,32,204,255,169,4,76,195,255,77,85    5575
2480 DATA 83,75,72,67                                             1530
10000 RESTORE :FOR I=2049 TO 4465:READ S:POKE I,S:NEXT I          2535
10010 POKE 45,114:POKE 46,17                                       1480
10020 PRINT "[CLR,2DOWN]SAVE" CHR$(34)"CHECKER.PRG" CHR$(34)",8" 3128
10030 PRINT "[HOME]";:END                                           596
10040 :
    
```

Checker für Diskettenbetrieb

```

2100 DATA 14,152,162,,160,2,32,189,255,104,134,147,102,157,76,91,255 6203
2150 DATA ,32,165,255,201,13,240,9,32,42,15,153,,4,200,208,160     5653
2160 DATA 1,177,178,133,253,200,177,178,133,254,56,76,125,245,169,,32,253 7076
2170 DATA 14,234,234,32,248,15,32,208,247,32,23,248,176,88,32,175  5732
2180 DATA 245,165,183,240,7,32,234,247,144,9,176,74,32,44,247    5859
2190 DATA 240,63,176,67,165,144,41,16,208,61,133,195,169,64,133,196 6082
2200 DATA 32,106,15,176,50,138,105,7,170,144,2,230,175,134,29,138,41 6484
2210 DATA 56,74,74,74,168,165,175,56,233,64,6,29,42,6,29,42,133    6193
2220 DATA 29,162,,228,29,240,8,169,255,157,,60,232,208,244,185    6040
2230 DATA 117,13,157,,60,76,25,8,76,146,9,48,10,76,146,9         5079
2260 DATA 222,13,176,83,32,253,14,32,208,247,32,56,248,176,59,32  5962
2270 DATA 143,246,165,27,133,174,185,28,56,233,64,133,175,165,254,133 6650
2280 DATA 194,165,253,133,193,24,101,174,133,174,165,175,101,194,133 6223
2290 DATA 175,169,1,32,106,247,176,19,169,,133,193,169,64,133    6291
2300 DATA 194,165,28,133,175,165,27,133,174,32,103,248,76,25,8,32  6228
10020 PRINT "[CLR,2DOWN]SAVE" CHR$(34)"CHECKER" CHR$(34)         2751
    
```

Checker: Änderungen für Datasette

dern der Rest des Programms rutscht nach.

—I—

kann eine Zeile einfügen; der Rest des Programms rutscht eine Zeile nach unten.

—H—

setzt den Cursor an den Textanfang.

—HOME—

setzt ihn an den Zeilenanfang.

—CSRS—

Der Cursor wird mit den Cursor-tasten gesteuert. Erreicht er den oberen oder unteren Bildschirmrand, wird der Bildschirm entsprechend gescrollt.

—P—

Ein im Checker befindliches Programm wird ausgedruckt.

—S—

Das Programm wird auf Diskette abgespeichert. Sollte sich im „Text“ eine Leerstelle befinden, springt der Cursor an diesen

Punkt. Ein Abspeichern ist erst möglich, wenn alle Leerzeilen entfernt wurden.

—L—

lädt ein abgesavtes Programm wieder, um es zum Beispiel zu ergänzen oder zu verändern.

Eine Zeile wird mit RETURN in den Speicher übernommen. Hier findet auch der Prüfsummentest statt. Stimmt die vom Computer errechnete Prüfsumme mit der eingegebenen überein, ertönt ein kurzer hoher Ton. Ist die Zeile fehlerhaft, ist ein langer, tiefer Ton zu hören, und der Cursor springt wieder an den Zeilenanfang (Zeile wird nicht übernommen). Die akustische Kontrolle ermöglicht, Programme einzugeben, ohne ständig den Bildschirm im Auge haben zu müssen.

Der Basicclader ist so gemacht, daß man nach dem Einlesen der Datas nur RETURN drücken muß,

um das Programm als Maschinensprachfile auf Diskette zu schreiben. Das File kann dann mit Load „CHECKER.PRG“.8 geladen und mit RUN gestartet werden. Auf keinen Fall den Lader starten und den vorgesehen Save-Befehl nicht bestätigen. Das Programm überschreibt sich selbst und ist deshalb nicht mehr auf normalem Weg zu saven. Ist man mit RUN/STOP-RESTORE im Basic und hat den abgetippten File noch nicht gesaved, kann man mit SYS 2073 wieder in den Checker springen, ohne das eingegebene Programm zu verlieren.

Der Checker belegt den Speicherbereich \$0801-\$1175. Im Bereich von \$4000-\$8000 stehen die „Textdaten“. Die Flags zum Kennzeichnen belegter Zeilen liegen von \$3000-\$3FFF.

(M. Köfferlein/W. Wintersberger/wosch)

Monitor knackt Geheimbefehle

Der Mikroprozessor des C64 hat über hundert geheime Befehle. Von offizieller Seite werden sie verschwiegen oder in ihrer Bedeutung heruntergespielt. Nur gelegentlich geistern sie als „illegale Opcodes“ durch Insider-Kreise. Profis nutzen die verborgenen Kommandos für raffinierte Kopierschutzverfahren. Auch der umstrittene Killerschutz von Dirk Hasinger (RUN 4/85) setzt illegale Befehle ein, um dem C64 den Garaus zu machen.

Daß der 6510/6502 mehr kann, als behauptet wird, läßt schon ein Blick in seine Befehlstabelle erahnen. Dort finden sich insgesamt 151 Opcodes. Jeder Opcode ist ein Byte lang. Ein Byte kann aber $2^8 = 256$ verschiedene Werte annehmen. Wie reagiert der Prozessor, wenn er auf eine der restlichen 105 Zahlen stößt? Die Fachliteratur hält sich in dieser Frage vornehm zurück. Die weißen Flecken auf der Prozessor-Landkarte werden meistens gar nicht erwähnt oder mit dem Begriff 'Future Expansion' (zukünftige Erweiterung) abgehandelt.

105 Befehle

RUN-Programmierer Klaus Burkert gab sich damit nicht zufrieden. Durch eigenes Experimentieren knobelte er die Wirkung der illegalen Opcodes heraus. Als Krönung seiner Arbeit hat er 'LINEASS' geschrieben, einen Maschinensprache-Monitor, der dort ansetzt, wo andere Monitore passen müssen. Zusätzlich zu den üblichen Funktionen kann er nämlich alle 105 Geheimbefehle disassemblieren und auch assemblieren.

'LINEASS' ist ein voll bildschirmorientierter Maschinenmonitor mit folgenden Befehlen:

- @: Dos-Befehle:
- @ **Befehl** wie im DOS 5.1
- @\$ Directory

*Geheimbefehle
schützen und kürzen
Programme. Wer sie
nutzen will, braucht
einen Spezialmonitor.
Nur 'LINEASS' verar-
beitet alle 105
illegalen Opcodes.*

@#n (n = 8 oder 9) stellt die Floppyadresse ein

@ gibt den Disk-Status aus

B aaaa,eeee: Der Speicherbereich von aaaa bis eeee wird mit einem Byte pro Zeile binär ausgegeben. Die Inhalte können durch Überschreiben und anschließendes Drücken der Return-Taste geändert werden.

C aaaa,eeee: Der Bereich von aaaa bis eeee wird mit 16 Bytes pro Zeile als ASCII-Zeichen ausgegeben. Steuerzeichen werden als Space dargestellt. Die Inhalte können durch Überschreiben und anschließendes Drücken von Return geändert werden. Steuerzeichen werden dabei durch eingegebene Spaces nicht verändert.

D aaaa,eeee: Der Bereich von aaaa bis eeee wird als Disassemblerlisting in der Form Adresse, Hex-Werte, Mnemonics ausgegeben. Das Kommando dient auch zum Assemblieren: Durch Überschreiben der Mnemonics und Drücken der Return-Taste wird ein Befehl geändert. Er wird dann vom Lineassembler ausgewertet und vom Disassembler neu ausgegeben. Anschließend wird die nächste Adresse vorgegeben.

E aaaa: Eine mit RTS endende Routine ab aaaa wird aufgerufen. Nach Ausführung wird in den Monitor gesprungen, und die Register werden angezeigt.

F aaaa,eeee,xx: Der Speicherbereich von aaaa bis eeee wird mit dem Wert xx gefüllt.

G aaaa: Es wird ein Maschinenprogramm ab Adresse aaaa ausgeführt. Falls in den Monitor zurückgesprungen werden soll, muß das Programm mit BRK enden. Danach werden die CPU-Register angezeigt.

H aaaa,eeee,xx: Der Speicherbereich von aaaa bis eeee wird nach dem Wert xx durchsucht. Die Adressen, an denen der Wert auftaucht, werden ausgegeben.

L "name",xx: Das Programm "name" wird vom Gerät xx (xx von 0 bis 15) absolut geladen.

M aaaa,eeee: Der Speicherbereich von aaaa bis eeee wird mit acht Bytes pro Zeile hexadezimal ausgegeben. Durch Überschreiben und Drücken der Return-Taste können Änderungen vorgenommen werden.

Q: Der Monitor wird durch einen Software-Reset verlassen (sys64738).

R: Die CPU-Register und der IRQ-Vektor werden angezeigt. Auch sie lassen sich durch Überschreiben und Drücken von Return ändern.

S "name",xx,aaaa,eeee: Der Bereich von aaaa bis eeee wird als Programm "name" auf dem Gerät xx (xx von 0 bis 15) gespeichert.

T aaaa,eeee,nnnn: Der Speicherbereich von aaaa bis eeee wird in den Bereich ab der Adresse nnnn verschoben.

V "name",xx: Das Programm "name" vom Gerät xx (xx von 0 bis 15) wird mit dem seiner Lage entsprechenden Speicherbereich verglichen. Bei Differenzen wird ein IO-Error ausgegeben.

X: Der Monitor wird mit einem Basic-Warmstart verlassen.

Die Speicherbereichsangaben aaaa und eeee gelten immer ab einschließlich aaaa bis ausschließlich (!) eeee.

Sämtliche Eingaben können in binärer Form mit vorangestelltem Prozentzeichen (%) oder in hexa-

dezimaler Form mit oder ohne vorangestelltes Dollarzeichen (\$) erfolgen.

Wird beim B-, C- oder M-Befehl nur eine einzige Adresse angegeben, so wird nur eine Zeile ausgegeben. Wenn beim D-Befehl die Endadresse weggelassen wird, so wird eine Bildschirmseite ausgegeben. Wenn gar keine Adresse angegeben wird, wird an der aktuellen Adresse mit der Ausgabe fortgefahren.

Beispiele:

D C000,C040

Bereich von C000 bis C03F einschließlich disassemblieren

D C000

Eine Bildschirmseite ab C000 disassemblieren

D

Eine Bildschirmseite ab der zuletzt ausgegebenen Adresse disassemblieren

Die Ausgabe bei den Befehlen B, C, D und M läßt sich mit der STOP-Taste abbrechen.

Der Monitor gibt bei Eingabefehlern und Bereichsüberschreitungen folgende Fehlermeldungen aus:

CM-Error

Der eingegebene Befehl kann nicht identifiziert werden.

CR-Error

Die Syntax des eingegebenen Befehls ist fehlerhaft.

OV-Error

Der erlaubte Bereich für einen Parameter wurde überschritten.

IO-Error

Bei der Ein-/Ausgabe wurde ein Fehler festgestellt.

Der Monitor benutzt außer den Speicherstellen \$02, \$5c-\$5f und dem Break-Vektor keine Systemadressen. Außerdem wurde nur die Routine zum Zeichen-einlesen direkt angesprungen. Alle anderen ROM-Aufrufe laufen über die Kernal-Sprungtabelle (\$ff81-\$fff3). Dadurch sollte das Programm mit jeder Spracherweiterung zusammenarbeiten.

Nichtstuer und Chaoten

Die illegalen Opcodes lassen sich in vier Gruppen einteilen:

Aufhänger: Sie bewirken einen Systemabsturz, der nur durch einen Hardware-Reset oder Aus- und Einschalten des Computers wieder aufgehoben werden kann. Zu dieser Gruppe gehören folgende Befehle: \$02, \$12, \$22, \$32, \$42, \$52, \$62, \$72, \$92, \$b2, \$d2, \$f2.

Sie werden durch den Disassembler als '!!!' dargestellt und bei der Eingabe durch den Lineassembler ignoriert. Will man sie trotzdem eingeben, so muß man sie als einzelne Bytes in den Programmtext einschmuggeln.

Nichtstuer: Diese Befehle bewirken absolut nichts. Es gibt sie als Ein-, Zwei- und Drei-Byte-Befehle. Zur Ein-Byte-Gruppe gehören — außer dem bekannten NOP (\$ea)

— \$1a, \$3a, \$5a, \$7a, \$da, \$fa.

Die Zwei-Byte-Gruppe wird durch \$04, \$14, \$34, \$44, \$54, \$64, \$74, \$80, \$82, \$89, \$c2, \$d4, \$e2, \$f4 gebildet.

Folgende Befehle sind Drei-Byte-NOPs: \$0c, \$1c, \$3c, \$5c, \$7c, \$dc, \$fc.

Diese Befehle werden als '***' disassembliert und bei der Eingabe ebenfalls ignoriert.

Doppeldecker: Die Befehle dieser Gruppe fassen zwei normale Befehle der gleichen Adressierungsart zu einem einzigen zusammen (Bild 1). So spart man Speicherplatz. Auch laufen Programme mit den zusammengesetzten Befehlen schneller. Ein Doppelbefehl benötigt nämlich weniger Taktzyklen als seine beiden Teilbefehle zusammen.

Nicht genug, daß diese Befehle für zwei schuften: Sie stellen auch noch Adressierungsarten zur Verfügung, die es für ihre Teilbefehle sonst nicht gibt! So können die Schiebe- und Rotierbefehle endlich mit der indirekten Adressierung verwendet werden, was zum Beispiel bei der Manipulation von hochauflösender Grafik sehr nützlich ist.

Beispiel:

SLO (\$fc),Y

Das angesprochene Byte wird um ein Bit nach links verschoben (ASL). Der zweite Teilbefehl (ORA) hat nur Auswirkungen auf den Akku-Inhalt. Für das betroffene Byte ist der neue Befehl daher mit dem (nicht existierenden!) Befehl ASL (\$fc),Y gleichwertig.

Mit dem DCC-Befehl lassen sich Schleifen effektiver gestalten. Statt der konventionellen Befehlsfolge

DEC Zähler

LDA Zähler

CMP #Untergrenze

BNE Loop

geht's jetzt einfacher mit

LDA #Untergrenze

DCC Zähler

BNE Loop

Der SAX-Befehl legt das Ergebnis einer UND-Funktion von A und X im Speicher ab. Beim SAH-Befehl geschieht das gleiche. Zusätzlich wird aber noch mit dem High Byte +1 der Adresse undiert. Beispiel: ▶

| | (,X) | Z | - | (,Y) | Z,X | -,Y | -,X | Z,Y |
|---------------|------|------|------|------|------|------|------|------|
| SLO (ASL,ORA) | \$03 | \$07 | \$0f | \$13 | \$17 | \$1b | \$1f | --- |
| RLA (ROL,AND) | \$23 | \$27 | \$2f | \$33 | \$37 | \$3b | \$3f | --- |
| SRE (LSR,EOR) | \$43 | \$47 | \$4f | \$53 | \$57 | \$5b | \$5f | --- |
| RRA (ROR,ADC) | \$63 | \$67 | \$6f | \$73 | \$77 | \$7b | \$7f | --- |
| SAX | \$83 | \$87 | \$8f | --- | --- | --- | --- | \$97 |
| SAH | --- | --- | --- | \$93 | --- | \$9f | --- | --- |
| LAX (LDA,TAX) | \$a3 | \$a7 | \$af | \$b3 | --- | \$bf | --- | \$b7 |
| DCC (DEC,CMP) | \$c3 | \$c7 | \$cf | \$d3 | \$d7 | \$db | \$df | --- |
| ICS (INC,SBC) | \$e3 | \$e7 | \$ef | \$f3 | \$f7 | \$fb | \$ff | --- |

Bild 1: „Doppeldecker“-Opcodes

Geheimer Programmschutz

Bild 3 zeigt, wie man die Geheimbefehle zum Programmschutz einsetzt. Das abgebildete Programm führt die Operation

LOAD "A",8,1

aus. Dabei soll das geladene Programm "A" wichtige Daten oder den eigentlichen Kopierschutz enthalten. Ziel ist es, den Programmteil, der die Ladeaktion durchführt, vor neugierigen Blicken zu schützen. Dazu wird massiv Gebrauch von illegalen Opcodes gemacht. Das richtig entschlüsselte Programm (Bild 3a) zu verstehen ist schon schwer genug. Wer dann noch ohne Geheimmonitor ans Werk geht, ist vollends verloren. In einem „legalen“ Disassembly (Bild 3b) wimmelt es nämlich von Fragezeichen. Schlimmer noch: Wo der Normalmonitor einen Befehl richtig verdaut zu haben scheint, ist besondere Vorsicht geboten. Oft ist er nur in eine der Fallen gegangen, die im Programmtext gestellt worden sind.

So finden sich an den Adressen \$901b, \$9022, \$9025 und \$9028 „Nichtstuer“-Befehle von zwei Byte Länge. Sie sind für den Programmablauf bedeutungslos und dienen nur dazu, Verwirrung zu stiften. Ein konventioneller Monitor kann nichts mit ihnen anfangen und wird sein Glück mit ihren Operanden versuchen. Und die entsprechen „zufällig“ offiziell zugelassenen Opcodes. Der Monitor disassembliert sie bereitwillig als Programmbefehle und gaukelt Sinn vor, wo keiner ist.

Wie die Beispiele zeigen, lassen sich die Geheimbefehle recht nutzbringend anwenden. Warum finden sie dann so wenig Beachtung? Zwei Einwände werden gegen die illegalen Opcodes erhoben:

Zum einen gibt es keine Garantie, daß sie auf jedem 6510/6502 laufen. Eine neue Prozessorversion könnte intern anders aufgebaut sein, so daß die illegalen Codes plötzlich einen anderen Effekt hätten. Die Praxis hat aber gezeigt,

| \$0 | \$2 | \$4 | \$6 | \$8 | \$a | \$c | \$e |
|---------------|----------|----------|----------|----------|----------|----------|----------|
| \$00 BRK | JSR - | RTI | RTS | XXX Z | LDY # | CPY # | CPX # |
| \$01 ORA (,X) | AND (,X) | EOR (,X) | ADC (,X) | STA (,X) | LDA (,X) | CMP (,X) | SBC (,X) |
| \$02 +++ | +++ | +++ | +++ | XXX Z | LDX # | XXX Z | XXX Z |
| \$03 SLO (,X) | RLA (,X) | SRE (,X) | RRA (,X) | SAX (,X) | LAX (,X) | DCC (,X) | ICS (,X) |
| \$04 XXX Z | BIT Z | XXX Z | XXX Z | STY Z | LDY Z | CPY Z | CPX Z |
| \$05 ORA Z | AND Z | EOR Z | ADC Z | STA Z | LDA Z | CMP Z | SBC Z |
| \$06 ASL Z | ROL Z | LSR Z | ROR Z | STX Z | LDX Z | DEC Z | INC Z |
| \$07 SLO Z | RLA Z | SRE Z | RRA Z | SAX Z | LAX Z | DCC Z | ICS Z |
| \$08 PHP | PLP | PHA | PLA | DEY | TAY | INY | INX |
| \$09 ORA # | AND # | EOR # | ADC # | XXX Z | LDA # | CMP # | SBC # |
| \$0a ASL | ROL | LSR | ROR | TXA | TAX | DEX | NOP |
| \$0b ANC # | ANC # | ASR # | ARR # | AXA # | ??? # | ASX # | SBC # |
| \$0c XXX - | BIT - | JMP - | JMP (| STY - | LDY - | CPY - | CPX - |
| \$0d ORA - | AND - | EOR - | ADC - | STA - | LDA - | CMP - | SBC - |
| \$0e ASL - | ROL - | LSR - | ROR - | STX - | LDX - | DEC - | INC - |
| \$0f SLO - | RLA - | SRE - | RRA - | SAX - | LAX - | DCC - | ICS - |
| \$10 BPL REL | BMI REL | BVC REL | BVS REL | BCC REL | BCS REL | BNE REL | BEQ REL |
| \$11 ORA (,Y) | AND (,Y) | EOR (,Y) | ADC (,Y) | STA (,Y) | LDA (,Y) | CMP (,Y) | SBC (,Y) |
| \$12 +++ | +++ | +++ | +++ | +++ | +++ | +++ | +++ |
| \$13 SLO (,Y) | RLA (,Y) | SRE (,Y) | RRA (,Y) | SAH (,Y) | LAX (,Y) | DCC (,Y) | ICS (,Y) |
| \$14 XXX Z | XXX Z | XXX Z | XXX Z | STY Z,X | LDY Z,X | XXX Z | XXX Z |
| \$15 ORA Z,X | AND Z,X | EOR Z,X | ADC Z,X | STA Z,X | LDA Z,X | CMP Z,X | SBC Z,X |
| \$16 ASL Z,X | ROL Z,X | LSR Z,X | ROR Z,X | STX Z,Y | LDX Z,Y | DEC Z,X | INC Z,X |
| \$17 SLO Z,X | RLA Z,X | SRE Z,X | RRA Z,X | SAX Z,Y | LAX Z,Y | DCC Z,X | ICS Z,X |
| \$18 CLC | SEC | CLI | SEI | TYA | CLV | CLD | SED |
| \$19 ORA -,Y | AND -,Y | EOR -,Y | ADC -,Y | STA -,Y | LDA -,Y | CMP -,Y | SBC -,Y |
| \$1a XXX | XXX | XXX | XXX | TXS | TSX | XXX | XXX |
| \$1b SLO -,Y | RLA -,Y | SRE -,Y | RRA -,Y | SSH -,Y | AST -,Y | DCC -,Y | ICS -,Y |
| \$1c XXX - | XXX - | XXX - | XXX - | SYH -,X | LDY -,X | XXX - | XXX - |
| \$1d ORA -,X | AND -,X | EOR -,X | ADC -,X | STA -,X | LDA -,X | CMP -,X | SBC -,X |
| \$1e ASL -,X | ROL -,X | LSR -,X | ROR -,X | SXH -,Y | LDX -,Y | DEC -,X | INC -,X |
| \$1f SLO -,X | RLA -,X | SRE -,X | RRA -,X | SAH -,Y | LAX -,Y | DCC -,X | ICS -,X |
| \$1 | \$3 | \$5 | \$7 | \$9 | \$b | \$d | \$f |

Bild 2: Alle legalen und illegalen Opcodes

LDA #3

LDX #5

LDY #7

SAH \$fe00,Y

Das High Byte ist \$fe. Es wird also #3 AND #5 AND #ff = #1 in Adresse \$fe00+7 = \$fe07 gespeichert. Wem dieser Befehl seltsam vorkommt, der wird sich erst recht bei der nächsten Gruppe von Opcodes wundern.

Chaoten: Die Befehle der letzten Gruppe scheinen die am wenigsten beabsichtigten Befehle zu sein. Sie führen komplexe Operationen durch, die in einigen Fällen recht undurchsichtige Resultate ergeben.

\$0b ANC # führt AND #xx aus und transportiert das Negativ-Flag ins Carry-Flag.

\$2b Wirkung wie \$0b.

\$4b ASR # führt AND #xx und LSR aus.

\$6b ARR # führt AND #xx und ROR aus.

\$8b AXA # führt TXA und AND #xx aus.

\$9b SSH -,Y

Das Ergebnis der UND-Funktion von Akku und X-Register wird im Stackpointer abgelegt. Danach wird dieser mit dem High Byte +1

der Adresse undiert und das Resultat absolut,Y-indiziert abgespeichert.

\$9c SYH -,X

Das Ergebnis der UND-Funktion zwischen dem Y-Register und dem High Byte + 1 der Adresse wird absolut,X-indiziert abgespeichert.

\$9e SXH -,Y

Wirkung wie bei \$9c (SYH), jedoch mit vertauschten Rollen für X- und Y-Register.

\$ab ??? #

Wirkung bislang nicht geklärt. Der Befehl kann jedoch als ??? #xx eingegeben werden.

\$bb AST -,Y

Das Ergebnis der UND-Funktion zwischen dem Stackpointer und dem Speicher wird im Stackpointer, im X-Register und im Akku abgelegt.

\$cb ASX #

Von dem Ergebnis der UND-Verknüpfung von Akku und X-Register werden #xx subtrahiert. Das Resultat wird im X-Register abgelegt.

\$eb SBC #

Wirkung wie Standard-Befehl \$e9. Bild 2 zeigt alle legalen und illegalen Opcodes auf einen Blick.

```

>D$9000: $AF $B9 $A3 ;LAX $A3B9 Pointer auf
>D$9003: $07 $BB ;SAX $BB Filename
>D$9005: $AF $39 $A3 ;LAX $A389 auf $A008
>D$9008: $07 $BC ;SAX $BC ("A")
>D$900A: $AF $90 $A3 ;LAX $A390 Laenge des
>D$900D: $07 $B7 ;SAX $B7 Namens =1
>D$900F: $A0 $B0 ;LDY #$B0 Sek.adr
>D$9011: $97 $09 ;SAX $09,Y =1
>D$9013: $AF $B9 $A3 ;LAX $A3B9 Geraete#
>D$9016: $97 $0A ;SAX $0A,Y =8
>D$9018: $AF $31 $03 ;LAX $0331 LOAD-Vektor
>D$901B: $F4 $E6 ;*** $E6 HB
>D$901D: $48 ;PHA auf Stack
>D$901E: $AF $30 $03 ;LAX $0330 LOAD-Vektor
>D$9021: $CA ;DEX LB - 1
>D$9022: $04 $C5 ;*** $C5
>D$9024: $8A ;TXA
>D$9025: $44 $25 ;*** $25
>D$9027: $48 ;PHA auf Stack
>D$9028: $D4 $AD ;*** $AD Flag fuer
>D$902A: $BF $AA $A2 ;LAX $A2AA,Y LOAD = 0
>D$902D: $60 ;RTS
    
```

```

,9000 AF 7??
,9001 B9 A3 87 LDA $87A3,Y
,9004 BB 7??
,9005 AF 7??
,9006 89 7??
,9007 A3 7??
,9008 87 7??
,9009 BC AF 90 LDY $90AF,X
,900C A3 7??
,900D 87 7??
,900E B7 7??
,900F A0 B0 LDY #$B0
,9011 97 7??
,9012 09 AF ORA #$AF
,9014 B9 A3 97 LDA $97A3,Y
,9017 0A ASL
,9018 AF 7??
,9019 31 03 AND ($03),Y
,901B F4 7??
,901C E6 48 INC $48
,901E AF 7??
,901F 30 03 BMI $9024
,9021 CA DEX
,9022 04 7??
,9023 C5 8A CMP $8A
,9025 44 7??
,9026 85 48 STA $48
,9028 D4 7??
,9029 AD BF AA LDA $AABF
,902C A2 60 LDX #$60
    
```

daß sie auf verschiedenen Varianten des Prozessors ohne Abweichungen funktionieren. Wahrscheinlich ist ihre Wirkung so eng mit der Struktur des Prozessors verknüpft, daß es erst bei grundlegenden architektonischen Änderungen zu Schwierigkeiten käme.

Der zweite Einwand wiegt schwerer: Möglicherweise funktionieren

Bild 3a und 3b: LOAD "A",8,1 mit Geheimmonitor (oben) und normalem Monitor (rechts) betrachtet.

die Geheimbefehle nicht in allen Situationen. Manche Befehle scheinen nicht korrekt zu arbeiten, wenn während ihrer Ausführung ein IRQ auftritt. Ungeklärt ist auch das Verhalten einiger Befehle im Dezimal-Modus.

Jeder muß also selbst entschei-

den, ob er seinen Computer mit illegalen Opcodes füttern will. Auf jeden Fall sollte man zweimal hinschauen, ob ein Programm mit Spezialbefehlen auch den gewünschten Verlauf nimmt.

(Klaus Burkert/Holger Kruse/pk)

```

10 FOR P=0 TO 1:FOR Z=100 TO 350:5=0 2484
20 FOR B=1 TO 16:READ D:IF P THEN PRINT# 2546
1,CHR# (D)
30 5=S+D:NEXT B:READ C:IF C<>S THEN 4042
PRINT "DATA ERROR IN ZEILE"Z:END
40 NEXT Z:IF P=0 THEN INPUT "FILENAME": 5484
F$:OPEN 1,8,1,F$:PRINT#1,CHR# (0)
CHR# (132):
50 RESTORE :NEXT P:CLOSE 1 655
100 DATA 169,0,141,175,207,141,176,207, 6891
141,177,207,141,180,207,141,181,2591
101 DATA 207,133,92,133,93,162,248,142, 6815
178,207,9,104,141,179,207,32,2266
102 DATA 19,194,169,68,141,22,3,189,196, 5915
141,23,3,169,8,141,185,1651
103 DATA 207,174,178,207,154,32,40,193, 6817
162,0,32,91,241,157,0,2,1870
104 DATA 232,201,13,208,245,169,0,157, 6486
255,1,169,255,141,182,207,32,2467
105 DATA 15,133,240,221,201,62,240,247, 6474
162,15,221,120,192,240,7,202,2578
106 DATA 16,248,232,76,187,192,138,10, 6840
170,32,111,192,76,49,192,189,2110
107 DATA 137,192,72,189,136,192,72,96,84, 5934
66,67,69,69,70,71,72,1633
108 DATA 76,77,81,82,83,84,86,88,110,196, 6752
199,198,113,199,189,202,2062
109 DATA 17,186,104,195,235,195,161,195, 6428
185,197,0,198,195,194,198,194,2659
110 DATA 207,197,220,206,196,197,188,194, 7176
32,210,255,72,32,225,255,208,2894
    
```

```

111 DATA 8,32,225,255,240,251,76,49,192, 6853
104,96,32,230,132,13,63,2058
112 DATA 32,0,138,10,170,189,222,192,32, 6771
168,192,189,223,192,32,168,2149
113 DATA 192,32,230,192,32,69,82,82,79, 5986
82,0,76,49,192,67,77,1533
114 DATA 67,82,79,86,73,79,133,2,104,133, 6277
94,104,133,95,152,72,1488
115 DATA 160,0,32,8,193,177,94,240,5,32, 5958
168,192,208,244,32,8,1793
116 DATA 193,104,168,165,2,108,94,0,230, 5443
94,208,2,230,95,96,32,1821
117 DATA 24,193,201,32,240,248,208,13, 6021
238,102,207,134,2,174,182,207,2486
118 DATA 189,0,2,166,2,72,104,96,32,230, 5323
192,13,62,0,96,72,1328
119 DATA 169,32,32,168,192,169,36,32,168, 6701
192,104,133,2,74,74,74,1651
120 DATA 74,32,62,193,32,168,192,165,2, 6385
41,15,32,82,193,32,168,1503
121 DATA 192,96,9,48,201,58,144,3,24,105, 5905
7,96,133,2,169,32,1319
122 DATA 32,168,182,169,37,32,168,192, 5388
162,8,169,0,6,2,42,32,1411
123 DATA 82,193,32,168,192,202,208,242, 5656
96,32,15,193,201,36,240,17,2149
124 DATA 201,37,240,75,32,250,193,16,5, 6807
162,1,76,107,192,206,182,2055
125 DATA 207,134,2,169,0,141,183,207,141, 6861
194,207,32,24,193,32,250,2186
126 DATA 193,48,41,14,183,207,46,184,207, 6721
    
```

| | | | |
|--|------|---|------|
| 176,74,14,183,207,46,184,2007 | | 207,32,1,196,141,175,207,140,2560 | |
| 127 DATA 207,176,66,14,183,207,46,184,207,176,58,14,183,207,46,184,2158 | 6628 | 167 DATA 176,207,142,177,207,186,142,178,207,6,184,141,179,207,32,33,2328 | 7445 |
| 128 DATA 207,176,50,13,183,207,141,183,207,76,155,193,166,2,96,134,2189 | 6735 | 168 DATA 194,76,49,192,180,168,184,178,184,141,175,207,140,176,207,142,2349 | 6995 |
| 129 DATA 2,169,0,141,183,207,141,184,207,32,24,193,32,250,193,48,2006 | 6526 | 169 DATA 177,207,184,141,179,207,216,88,184,56,233,2,141,180,207,184,2346 | 7312 |
| 130 DATA 17,14,183,207,46,184,207,176,12,13,183,207,141,183,207,76,2056 | 6296 | 170 DATA 233,0,141,181,207,166,142,178,207,32,33,194,76,49,192,32,2083 | 6895 |
| 131 DATA 217,193,166,2,96,162,2,76,187,192,201,58,48,7,201,65,1873 | 6551 | 171 DATA 15,193,240,73,201,36,240,103,201,35,240,38,174,182,207,160,2338 | 6303 |
| 132 DATA 48,14,56,233,7,201,48,48,7,201,64,16,3,41,15,96,1098 | 5309 | 172 DATA 2,32,24,193,208,251,134,2,173,192,207,56,229,2,32,189,1918 | 6485 |
| 133 DATA 169,255,96,169,6,141,32,208,141,33,208,32,230,192,147,154,2213 | 6636 | 173 DATA 255,169,7,174,185,207,180,15,32,186,255,32,192,255,32,231,2387 | 6863 |
| 134 DATA 0,32,230,192,14,8,13,204,201,206,197,193,211,211,45,54,2011 | 6749 | 174 DATA 255,96,32,121,183,173,184,207,208,14,173,183,207,41,246,208,2541 | 6827 |
| 135 DATA 52,32,87,82,73,84,84,69,78,32,49,57,56,53,32,66,986 | 5959 | 175 DATA 7,173,183,207,141,185,207,96,162,1,76,187,192,169,13,32,2031 | 6643 |
| 136 DATA 89,32,203,76,65,85,83,32,194,85,82,75,69,82,84,13,1349 | 5777 | 176 DATA 168,192,173,185,207,133,186,32,180,255,169,111,133,185,32,150,2491 | 6914 |
| 137 DATA 32,32,32,32,73,82,81,32,32,32,80,87,32,32,32,65,768 | 5962 | 177 DATA 255,32,165,255,32,168,192,201,13,208,246,32,171,255,96,169,2490 | 6903 |
| 138 DATA 32,32,32,89,32,32,32,88,32,32,32,83,32,32,32,78,720 | 5607 | 178 DATA 13,32,168,192,174,182,207,160,2,32,24,193,208,251,134,2,1974 | 5602 |
| 139 DATA 86,45,66,68,73,90,87,13,62,82,58,0,169,32,32,168,1111 | 5861 | 179 DATA 173,182,207,56,229,2,32,189,255,169,127,174,185,207,168,96,2443 | 7584 |
| 140 DATA 192,173,21,3,32,59,193,173,20,3,32,59,193,169,32,32,1386 | 5973 | 180 DATA 32,186,255,32,192,255,165,186,32,180,255,165,185,32,150,255,2557 | 7013 |
| 141 DATA 168,192,173,181,207,32,59,193,173,180,207,32,59,193,173,175,2397 | 7611 | 181 DATA 169,0,133,144,160,3,132,94,32,165,255,133,95,164,144,208,2031 | 6691 |
| 142 DATA 207,32,47,193,173,176,207,32,47,193,173,177,207,32,47,193,2136 | 7331 | 182 DATA 47,32,165,255,164,144,208,40,164,94,138,208,233,166,95,32,2183 | 6884 |
| 143 DATA 173,178,207,32,47,193,173,179,207,32,92,193,96,169,55,133,2159 | 6901 | 183 DATA 205,189,169,32,32,188,192,32,165,255,166,144,208,18,178,240,2385 | 7217 |
| 144 DATA 1,78,116,164,188,252,255,32,15,193,208,12,76,33,194,201,1936 | 6882 | 184 DATA 6,32,168,192,76,55,197,169,13,32,168,192,160,2,208,198,1868 | 6238 |
| 145 DATA 58,240,5,162,1,78,187,192,32,121,193,120,173,184,207,141,2092 | 6422 | 185 DATA 169,73,133,94,169,1,162,94,180,0,32,189,255,169,7,174,1881 | 6969 |
| 146 DATA 21,3,173,183,207,141,20,3,88,32,121,193,173,183,207,141,1889 | 7119 | 186 DATA 185,207,160,15,32,186,255,32,192,255,32,231,255,96,32,15,2180 | 6890 |
| 147 DATA 180,207,173,184,207,141,181,207,32,121,193,173,184,207,208,211,2089 | 7148 | 187 DATA 193,240,37,201,34,208,247,160,2,174,182,207,232,134,2,32,2285 | 6174 |
| 148 DATA 173,183,207,141,175,207,32,121,193,173,184,207,208,197,173,183,2757 | 7205 | 188 DATA 15,193,240,20,201,34,208,247,173,182,207,56,229,2,32,189,2228 | 6394 |
| 149 DATA 207,141,176,207,32,121,193,173,184,207,208,183,173,183,207,141,2736 | 6964 | 189 DATA 255,32,15,193,201,44,240,5,162,1,76,187,192,32,121,193,1949 | 6702 |
| 150 DATA 177,207,32,121,193,173,184,207,208,169,173,183,207,141,178,207,2768 | 6831 | 190 DATA 173,184,207,248,5,162,2,76,187,192,173,183,207,201,20,176,2388 | 5948 |
| 151 DATA 32,121,193,173,184,207,208,155,173,183,207,141,179,207,96,32,2491 | 6302 | 191 DATA 244,170,160,1,169,127,32,186,255,96,32,110,197,169,0,32,1980 | 7044 |
| 152 DATA 121,193,173,183,207,133,92,173,184,207,133,93,206,182,207,32,2519 | 7272 | 192 DATA 213,255,76,243,197,32,110,197,169,1,32,213,255,76,243,197,2589 | 6856 |
| 153 DATA 15,193,201,44,240,5,162,1,76,197,192,32,121,193,173,183,2018 | 6625 | 193 DATA 32,110,197,206,182,207,32,15,193,201,44,240,5,162,1,76,1983 | 6168 |
| 154 DATA 207,133,94,173,184,207,133,95,96,32,63,195,206,182,207,32,2239 | 6999 | 194 DATA 187,192,169,13,32,188,192,32,63,195,169,92,166,94,164,95,2023 | 7055 |
| 155 DATA 15,193,201,44,240,5,162,1,76,197,192,32,121,193,173,184,2019 | 6514 | 195 DATA 32,216,255,176,7,165,144,41,63,208,1,96,162,3,76,187,1832 | 6303 |
| 156 DATA 207,240,5,162,2,76,187,192,160,0,173,183,207,145,92,230,2261 | 6177 | 196 DATA 192,32,15,193,240,16,206,182,207,32,121,193,173,183,207,133,2325 | 7657 |
| 157 DATA 92,208,2,230,93,165,92,197,94,165,93,229,95,144,233,96,2228 | 6462 | 197 DATA 92,173,184,207,133,93,206,182,207,32,15,193,208,16,165,92,2198 | 6227 |
| 158 DATA 234,234,32,63,195,206,182,207,32,15,193,201,44,240,5,162,2245 | 6899 | 198 DATA 24,105,1,133,94,165,93,185,0,133,95,76,82,138,201,58,1563 | 6162 |
| 159 DATA 1,76,187,192,32,121,193,173,184,207,240,5,162,2,76,187,2038 | 6345 | 199 DATA 240,109,201,44,240,5,162,1,76,187,192,206,182,207,32,15,2099 | 6488 |
| 160 DATA 132,160,0,173,183,207,209,92,206,15,169,13,32,168,192,165,2178 | 6819 | 200 DATA 193,201,44,208,241,32,121,193,173,183,207,133,94,173,184,207,2587 | 6688 |
| 161 DATA 93,32,47,193,165,92,32,59,193,230,92,208,2,230,93,165,1926 | 6009 | 201 DATA 133,95,169,13,32,168,192,169,62,32,168,192,169,77,32,168,1671 | 6573 |
| 162 DATA 92,197,94,165,93,229,95,144,216,96,234,234,32,121,193,173,2488 | 6429 | 202 DATA 192,169,36,32,168,192,165,93,32,59,193,165,92,32,59,193,1872 | 6889 |
| 163 DATA 179,207,72,40,174,178,207,154,174,177,207,172,176,207,173,175,2672 | 6952 | 203 DATA 163,58,32,188,192,160,0,169,32,32,168,192,177,92,32,59,1732 | 6572 |
| 164 DATA 207,72,173,183,207,141,180,207,173,184,207,141,181,207,104,188,2675 | 6353 | 204 DATA 193,200,192,8,48,241,165,92,24,185,0,133,92,165,93,185,1864 | 6343 |
| 165 DATA 180,207,184,184,32,121,193,173,179,207,72,40,174,178,207,154,2325 | 7064 | 205 DATA 0,133,93,165,92,56,229,94,165,93,229,95,144,188,96,32,1886 | 6183 |
| 166 DATA 174,177,207,172,176,207,173,175,207,32,1,196,141,175,207,140,2560 | 8086 | 206 DATA 121,193,173,184,207,240,5,162,2, | 6601 |

Super-Tool

| | | | |
|--|------|--|------|
| 76,187,192,173,183,207,160,2465 | | 76,60,75,43,4,651 | |
| 207 DATA 0,145,92,230,92,208,2,230,93,206,182,207,32,15,193,8,1935 | 6174 | 247 DATA 60,19,43,75,60,75,43,4,60,36,2,76,48,12,2,49,664 | 5013 |
| 208 DATA 206,182,207,40,208,217,96,32,15,193,240,16,206,182,207,32,2279 | 6437 | 248 DATA 48,47,2,49,1,12,2,49,48,13,2,76,48,75,2,49,523 | 4876 |
| 209 DATA 121,193,173,183,207,133,92,173,184,207,133,93,206,182,207,32,2519 | 7272 | 249 DATA 48,57,2,75,46,75,2,49,48,52,30,76,61,75,30,41,769 | 4921 |
| 210 DATA 15,193,208,16,165,92,24,105,1,133,94,165,93,105,0,133,1542 | 6317 | 250 DATA 61,44,30,41,5,35,30,41,61,17,30,76,61,75,30,41,678 | 5801 |
| 211 DATA 95,76,14,139,201,58,240,82,201,44,240,5,162,1,76,187,1881 | 6099 | 251 DATA 61,21,30,75,61,75,30,41,61,53,0,76,51,75,0,50,760 | 4829 |
| 212 DATA 192,32,121,193,173,183,207,133,94,173,184,207,133,95,169,13,2382 | 6924 | 252 DATA 51,46,0,50,3,35,0,50,51,18,0,76,51,75,0,50,556 | 5080 |
| 213 DATA 32,168,192,169,62,32,168,182,169,66,32,168,192,169,36,32,1879 | 6396 | 253 DATA 51,59,0,75,51,75,0,50,51,75,63,75,55,65,63,64,872 | 5162 |
| 214 DATA 168,192,165,93,32,59,193,165,92,32,59,193,169,58,32,168,1870 | 6749 | 254 DATA 55,29,75,71,8,65,63,64,55,9,63,76,54,65,63,64,879 | 5511 |
| 215 DATA 192,160,0,177,92,32,92,193,230,92,208,2,230,93,165,92,2050 | 6006 | 255 DATA 55,73,63,72,62,67,63,66,54,40,38,39,37,40,38,39,846 | 6134 |
| 216 DATA 56,229,94,165,93,229,95,144,197,96,32,121,193,173,184,207,2308 | 6582 | 256 DATA 37,69,38,68,74,40,38,39,37,10,38,76,37,40,38,39,718 | 6325 |
| 217 DATA 240,5,162,2,76,187,192,173,183,207,160,0,145,92,230,92,2146 | 6494 | 257 DATA 37,22,38,70,6,40,38,39,37,25,23,75,26,25,23,27,551 | 5733 |
| 218 DATA 208,2,230,93,206,182,207,32,15,193,8,206,182,207,40,208,2219 | 6641 | 258 DATA 26,34,23,28,7,25,23,27,26,14,23,76,26,75,23,27,483 | 5925 |
| 219 DATA 217,96,32,15,193,240,16,206,182,207,32,121,193,173,183,207,2313 | 6689 | 259 DATA 26,20,23,75,26,75,23,27,26,24,56,75,31,24,56,32,619 | 5762 |
| 220 DATA 133,92,173,184,207,133,93,206,182,207,32,15,193,208,16,165,2239 | 6851 | 260 DATA 31,33,56,42,56,24,56,32,31,11,56,76,31,75,56,32,688 | 5956 |
| 221 DATA 92,24,105,1,133,94,165,93,105,0,133,95,76,185,193,201,1701 | 6195 | 261 DATA 31,58,56,75,31,75,56,32,31,1,2,2,2,2,2,2,458 | 4837 |
| 222 DATA 56,240,106,201,44,240,5,162,1,76,187,192,32,121,193,173,2031 | 6617 | 262 DATA 2,3,3,3,3,141,186,207,201,188,208,3,169,12,96,201,1546 | 5661 |
| 223 DATA 183,207,133,94,173,184,207,133,95,169,13,32,168,192,169,62,2214 | 6501 | 263 DATA 137,208,3,169,3,96,41,31,170,173,186,187,201,0,16,10,1651 | 5750 |
| 224 DATA 32,168,192,169,67,32,168,192,169,36,32,168,192,165,93,32,1907 | 6465 | 264 DATA 201,192,16,6,189,142,202,76,77,202,189,110,202,224,0,240,2268 | 6259 |
| 225 DATA 59,193,165,32,32,59,193,169,58,32,168,192,160,0,177,82,1841 | 6453 | 265 DATA 15,224,2,240,18,96,173,186,207,74,74,74,74,74,170,96,1797 | 6085 |
| 226 DATA 133,2,41,96,208,4,169,32,133,2,165,2,32,168,192,200,1579 | 6064 | 266 DATA 32,86,202,189,174,202,96,32,86,202,189,182,202,96,0,7,1977 | 5938 |
| 227 DATA 192,16,48,234,165,92,24,105,16,133,92,165,93,105,0,133,1613 | 6535 | 267 DATA 0,7,3,3,3,3,1,2,1,2,8,9,9,9,6,8,75 | 3986 |
| 228 DATA 93,165,92,56,229,94,165,93,229,95,144,173,96,160,0,32,1916 | 6275 | 268 DATA 1,8,3,4,4,4,1,11,1,11,9,10,10,10,0,7,94 | 3861 |
| 229 DATA 24,193,240,29,133,2,201,32,208,10,177,92,41,96,208,4,1690 | 5694 | 269 DATA 0,7,3,3,3,3,1,2,1,2,9,9,9,9,6,8,75 | 3986 |
| 230 DATA 177,92,133,2,165,2,145,92,230,92,208,225,230,93,76,13,1975 | 6031 | 270 DATA 1,8,4,4,5,5,1,11,1,11,10,10,11,11,1,9,103 | 4300 |
| 231 DATA 200,96,65,65,65,65,65,65,65,65,65,66,66,66,66,66,1211 | 6006 | 271 DATA 1,1,3,2,2,2,1,1,1,1,3,2,3,3,32,15,73 | 4021 |
| 232 DATA 66,66,66,66,66,67,67,67,67,67,67,67,67,68,68,68,68,1071 | 6336 | 272 DATA 193,240,16,206,182,207,32,121,193,173,183,207,133,92,173,184,2535 | 6831 |
| 233 DATA 69,73,73,73,73,74,74,76,76,76,76,76,76,78,79,80,80,1206 | 6288 | 273 DATA 207,133,93,206,182,207,32,15,193,208,12,169,23,133,255,32,2100 | 6641 |
| 234 DATA 80,80,82,82,82,82,82,82,83,83,83,83,83,83,83,83,1316 | 5442 | 274 DATA 19,203,198,255,208,249,96,201,58,208,3,76,182,204,201,44,2405 | 5930 |
| 235 DATA 83,83,83,83,83,83,84,84,84,84,84,84,63,42,94,68,1269 | 5998 | 275 DATA 240,5,162,1,76,187,192,32,121,193,173,183,207,133,94,173,2172 | 6892 |
| 236 DATA 78,78,82,83,83,83,83,83,83,83,83,83,83,83,83,83,1251 | 5664 | 276 DATA 184,207,133,95,32,19,203,165,92,56,229,94,165,93,229,95,2081 | 6741 |
| 237 DATA 96,86,76,76,76,76,77,80,80,80,80,80,80,80,80,80,80,1211 | 6354 | 277 DATA 144,242,96,169,13,32,168,192,169,62,32,169,192,169,68,32,1948 | 6274 |
| 238 DATA 78,78,77,83,65,68,68,68,83,79,82,72,72,76,76,76,1201 | 6030 | 278 DATA 168,192,169,36,32,168,192,165,93,32,59,193,165,92,32,59,1847 | 6883 |
| 239 DATA 79,79,82,84,84,65,65,66,69,69,69,76,82,83,84,84,1220 | 5230 | 279 DATA 193,169,58,32,168,192,160,0,177,92,32,47,193,177,92,32,1814 | 6860 |
| 240 DATA 84,88,89,65,65,83,88,88,89,63,42,94,67,67,68,82,1222 | 5786 | 280 DATA 37,202,170,202,183,25,202,170,224,1,240,16,200,177,92,32,2179 | 5529 |
| 241 DATA 76,82,84,88,65,67,83,81,84,73,69,76,75,67,83,67,1220 | 6009 | 281 DATA 47,193,224,2,240,6,200,177,92,32,47,193,163,32,32,168,1854 | 6120 |
| 242 DATA 68,73,86,88,88,89,67,67,88,89,92,83,67,88,89,80,1284 | 5892 | 282 DATA 192,165,211,201,21,48,245,169,59,32,168,192,160,0,177,92,2132 | 6723 |
| 243 DATA 82,88,65,88,89,82,80,65,65,80,65,80,65,76,82,65,1217 | 5741 | 283 DATA 170,189,25,201,170,189,50,200,32,169,192,189,127,200,32,168,2302 | 6634 |
| 244 DATA 73,83,72,88,67,67,68,73,79,69,72,65,88,89,72,72,1197 | 6388 | 284 DATA 192,189,204,200,32,168,192,169,32,32,168,192,173,186,207,32,2368 | 6932 |
| 245 DATA 88,89,88,65,63,63,63,42,94,16,43,76,60,75,43,4,994 | 5239 | 285 DATA 37,202,170,224,1,208,3,76,147,204,224,2,208,5,169,35,1915 | 5936 |
| 246 DATA 60,45,43,4,1,75,43,4,60,15,43,76,60,75,43,4,651 | 5475 | 286 DATA 32,168,192,224,7,240,8,224,8,76,48,12,2,49,664 | 5119 |

Super-Tool

| | |
|--|------|
| 240,4,224,12,208,5,169,1965 | |
| 287 DATA 40,32,168,192,169,36,32,168,192, 224,9,48,3,76,90,204,1603 | 6137 |
| 288 DATA 224,6,240,88,160,1,177,92,32,59, 193,224,4,16,3,76,1595 | 6243 |
| 289 DATA 147,204,224,4,208,13,169,44,32, 169,192,169,88,32,168,192,2054 | 6698 |
| 290 DATA 76,147,204,224,5,208,13,169,44, 32,169,192,169,88,32,168,1940 | 6219 |
| 291 DATA 192,76,147,204,224,7,208,18,169, 44,32,168,192,169,88,32,1970 | 6848 |
| 292 DATA 168,192,169,41,32,168,192,76, 147,204,169,41,32,168,192,169,2168 | 6684 |
| 293 DATA 44,32,168,192,169,88,32,168,192, 76,147,204,165,92,24,105,1899 | 6723 |
| 294 DATA 2,141,163,207,165,93,105,0,141, 184,207,160,1,177,92,16,1874 | 6205 |
| 295 DATA 4,169,255,133,2,177,92,48,4,169, 0,133,2,177,92,24,1481 | 5556 |
| 296 DATA 109,183,207,141,183,207,165,2, 183,184,207,141,184,207,32,59,2328 | 7328 |
| 297 DATA 193,173,183,207,32,59,193,76, 147,204,160,2,177,92,32,59,1989 | 7813 |
| 298 DATA 193,160,1,177,92,32,59,193,224, 9,240,39,224,10,208,13,1874 | 6423 |
| 299 DATA 169,44,32,168,192,169,88,32,168, 192,76,147,204,224,11,208,2124 | 6062 |
| 300 DATA 13,169,44,32,168,192,169,89,32, 168,192,76,147,204,169,41,1985 | 6293 |
| 301 DATA 32,168,192,160,0,177,92,32,37, 202,170,202,189,25,202,24,1804 | 5777 |
| 302 DATA 101,92,133,92,165,93,105,0,133, 93,169,32,32,168,192,165,1785 | 6709 |
| 303 DATA 211,201,39,48,245,96,169,0,141, 190,207,32,15,193,240,6,2033 | 6862 |
| 304 DATA 201,59,208,242,240,5,162,1,76, 197,192,32,24,193,208,3,2033 | 6524 |
| 305 DATA 76,148,206,141,187,207,32,24, 193,208,3,76,148,206,141,188,2184 | 5912 |
| 306 DATA 207,32,24,193,208,3,76,148,206, 141,189,207,32,15,193,208,2082 | 6451 |
| 307 DATA 8,162,1,142,190,207,76,153,205, 201,35,208,6,162,2,142,1902 | 6158 |
| 308 DATA 190,207,76,36,205,201,40,208,27, 174,187,207,224,74,208,15,2279 | 6576 |
| 309 DATA 174,188,207,224,77,208,8,162,12, 142,190,207,76,36,205,162,2278 | 6672 |
| 310 DATA 126,142,190,207,173,190,207,208, 3,206,182,207,32,121,193,206,2595 | 7221 |
| 311 DATA 182,207,32,15,193,208,10,174, 190,207,208,5,162,9,142,190,2134 | 6691 |
| 312 DATA 207,201,41,208,34,174,190,207, 224,12,240,77,44,190,207,16,2272 | 6485 |
| 313 DATA 72,32,15,193,201,44,208,65,32, 15,193,201,89,208,58,169,1795 | 6358 |
| 314 DATA 8,141,190,207,76,153,205,201,44, 208,46,32,15,193,201,89,2089 | 6547 |
| 315 DATA 208,6,169,11,141,190,207,76,153, 205,201,88,208,27,32,15,1939 | 6595 |
| 316 DATA 193,208,6,169,10,141,190,207,76, 153,205,201,41,208,10,44,2064 | 6573 |
| 317 DATA 190,207,16,5,169,7,141,190,207, 173,190,207,41,127,141,190,2201 | 5839 |
| 318 DATA 207,173,190,207,201,9,208,70, 173,187,207,201,66,208,63,173,2543 | 7464 |
| 319 DATA 183,207,201,73,240,56,169,6,141, 190,207,234,234,234,234,234,2848 | 7810 |
| 320 DATA 165,92,24,183,2,133,94,165,93, 105,0,133,95,173,183,207,1769 | 6622 |
| 321 DATA 56,229,94,141,183,207,170,173, 184,207,229,95,224,129,105,0,2425 | 6626 |
| 322 DATA 234,234,240,5,169,0,141,190,207, 169,0,141,184,207,173,184,2478 | 7223 |
| 323 DATA 207,208,17,173,190,207,201,9,48, 10,201,12,16,6,56,233,1794 | 5928 |
| 324 DATA 6,141,190,207,173,187,207,201, 42,240,4,201,94,208,5,169,2275 | 6186 |
| 325 DATA 0,141,190,207,162,77,202,48,77, 173,197,207,221,50,200,208,2358 | 6399 |
| 326 DATA 245,173,189,207,221,127,200,208, 237,173,189,207,221,204,200,208,3208 | 8244 |
| 327 DATA 223,138,141,191,207,160,0,240,3, 200,240,22,173,191,207,217,2559 | 6966 |
| 328 DATA 25,201,208,245,152,32,37,202, 174,186,207,205,190,207,240,27,2538 | 6817 |
| 329 DATA 208,231,173,190,207,201,3,48,13, 201,6,16,9,24,105,6,1641 | 6792 |
| 330 DATA 141,190,207,76,20,206,169,0,141, 190,207,142,186,207,173,190,2445 | 7391 |
| 331 DATA 207,240,33,170,202,169,25,202, 170,173,186,207,160,0,145,92,2481 | 6288 |
| 332 DATA 224,1,240,16,173,183,207,200, 145,92,224,2,240,6,173,184,2310 | 6165 |
| 333 DATA 207,200,145,92,198,214,32,19, 203,169,13,32,168,192,169,62,2115 | 6425 |
| 334 DATA 32,168,192,169,68,32,168,192, 169,36,32,168,192,165,93,32,1988 | 6244 |
| 335 DATA 59,193,165,92,32,59,193,169,58, 32,168,192,169,59,32,168,1848 | 6593 |
| 336 DATA 192,169,32,32,168,192,165,211, 201,39,48,245,198,214,182,7,2275 | 6662 |
| 337 DATA 169,29,157,119,2,202,16,258,169, 0,133,198,96,32,121,193,1894 | 6759 |
| 338 DATA 173,183,207,133,92,173,184,207, 133,93,206,182,207,32,15,193,2413 | 7106 |
| 339 DATA 201,44,240,5,162,1,76,187,192, 32,121,193,173,183,207,141,2158 | 6505 |
| 340 DATA 192,207,173,184,207,141,193,207, 206,182,207,32,15,193,201,44,2584 | 7345 |
| 341 DATA 208,226,32,121,193,173,183,207, 133,94,173,184,207,133,95,165,2527 | 6988 |
| 342 DATA 94,56,229,92,165,95,229,93,176, 32,160,0,177,92,145,94,1929 | 6465 |
| 343 DATA 230,32,208,2,230,93,230,94,208, 2,230,95,165,92,56,237,2264 | 6362 |
| 344 DATA 192,207,165,93,237,193,207,144, 225,96,173,192,207,24,229,92,2676 | 7889 |
| 345 DATA 133,2,173,193,207,229,93,72,24, 165,2,101,94,133,94,184,1819 | 6221 |
| 346 DATA 101,95,133,95,173,192,207,166, 92,133,92,142,192,207,173,193,2386 | 7264 |
| 347 DATA 207,166,93,133,93,142,193,207, 165,92,56,233,1,133,92,165,2171 | 6599 |
| 348 DATA 93,233,0,133,93,160,0,177,92, 145,94,165,92,56,233,1,1767 | 6285 |
| 349 DATA 133,92,176,2,198,93,165,94,56, 233,1,133,94,176,2,198,1846 | 6678 |
| 350 DATA 95,165,92,56,237,192,207,165,93, 237,193,207,176,215,96,255,2681 | 7482 |

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|
| C000 | A9 | 00 | 8D | AF | CF | 8D | B0 | CF | A9 | C038 | A2 | 00 | 20 | 5B | F1 | 9D | 00 | 02 | B9 |
| C008 | 8D | B1 | CF | 8D | B4 | CF | 8D | B5 | DB | C040 | E8 | C9 | 0D | D0 | F5 | A9 | 00 | 9D | 0C |
| C010 | CF | 85 | 5C | 85 | 5D | A2 | F8 | 8E | 46 | C048 | FF | 01 | A9 | FF | 8D | B6 | CF | 20 | 2D |
| C018 | B2 | CF | 08 | 68 | 8D | B3 | CF | 20 | 0C | C050 | 0F | C1 | F0 | DD | C9 | 3E | F0 | F7 | E5 |
| C020 | 13 | C2 | A9 | 44 | 8D | 16 | 03 | A9 | 87 | C058 | A2 | 0F | DD | 78 | C0 | F0 | 07 | CA | 05 |
| C028 | C4 | 8D | 17 | 03 | A9 | 08 | 8D | B9 | BF | C060 | 10 | F8 | E8 | 4C | BB | C0 | 8A | 0A | 2F |
| C030 | CF | AE | B2 | CF | 9A | 20 | 28 | C1 | E5 | C068 | AA | 20 | 6F | C0 | 4C | 31 | C0 | BD | 56 |
| | | | | | | | | | | C070 | 89 | C0 | 48 | BD | 88 | C0 | 48 | 80 | 05 |
| | | | | | | | | | | C078 | 40 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | FB |

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| C080 | 4C | 4D | 51 | 52 | 53 | 54 | 56 | 58 | FB | C268 | 20 | 20 | 20 | 53 | 20 | 20 | 20 | 4E | A0 |
| C088 | 6E | C4 | C6 | C6 | 71 | C7 | BD | CA | 47 | C270 | 56 | 2D | 42 | 44 | 49 | 5A | 43 | 0D | 4C |
| C090 | 11 | C4 | 68 | C3 | EB | C3 | A1 | C3 | F9 | C278 | 3E | 52 | 3A | 00 | A9 | 20 | 20 | A8 | 27 |
| C098 | B9 | C5 | 00 | C6 | C3 | C2 | C6 | C2 | 34 | C280 | C0 | AD | 15 | 03 | 20 | 3B | C1 | AD | 1E |
| C0A0 | CF | C5 | DC | CE | C4 | C5 | BC | C2 | 11 | C282 | 14 | 03 | 20 | 3B | C1 | A9 | 20 | 20 | 0E |
| C0A8 | 20 | D2 | FF | 48 | 20 | E1 | FF | D0 | 73 | C290 | A8 | C0 | AD | B5 | CF | 20 | 3B | C1 | 09 |
| C0B0 | 08 | 20 | E1 | FF | F0 | FB | 4C | 31 | DA | C298 | AD | B4 | CF | 20 | 3B | C1 | AD | AF | 20 |
| C0B8 | C0 | 68 | 60 | 20 | E6 | C0 | 0D | 3F | 8C | C2A0 | CF | 20 | 2F | C1 | AD | B0 | CF | 20 | C9 |
| C0C0 | 20 | 00 | 8A | 0A | AA | BD | DE | C0 | AA | C2A8 | 2F | C1 | AD | B1 | CF | 20 | 2F | C1 | 87 |
| C0C8 | 20 | A8 | C0 | BD | DF | C0 | 20 | A8 | 13 | C2B0 | AD | B2 | CF | 20 | 2F | C1 | AD | B3 | 12 |
| C0D0 | C0 | 20 | E6 | C0 | 20 | 45 | 52 | 52 | A1 | C2B8 | CF | 20 | 5C | C1 | 60 | A9 | 37 | 85 | B3 |
| C0D8 | 4F | 52 | 00 | 4C | 31 | C0 | 43 | 4D | 19 | C2C0 | 01 | 4C | 74 | A4 | 6C | FC | FF | 20 | D5 |
| C0E0 | 43 | 52 | 4F | 56 | 49 | 4F | 85 | 02 | 67 | C2C8 | 0F | C1 | D0 | 0C | 4C | 21 | C2 | C9 | 36 |
| C0E8 | 68 | 85 | 5E | 68 | 85 | 5F | 98 | 48 | 4F | C2D0 | 3A | F0 | 05 | A2 | 01 | 4C | BB | C0 | 5E |
| C0F0 | A0 | 00 | 20 | 08 | C1 | B1 | 5E | F0 | 35 | C2D8 | 20 | 79 | C1 | 78 | AD | B8 | CF | 8D | 27 |
| C0F8 | 05 | 20 | A8 | C0 | D0 | F4 | 20 | 08 | C2 | C2E0 | 15 | 03 | AD | B7 | CF | 8D | 14 | 03 | 5B |
| C100 | C1 | 68 | A8 | A5 | 02 | 6C | 5E | 00 | 50 | C2E8 | 58 | 20 | 79 | C1 | AD | B7 | CF | 8D | 28 |
| C108 | E6 | 5E | D0 | 02 | E6 | 5F | 60 | 20 | 1B | C2F0 | B4 | CF | AD | B8 | CF | 8D | B5 | CF | 02 |
| C110 | 18 | C1 | C9 | 20 | F0 | F9 | D0 | 0D | BA | C2F8 | 20 | 79 | C1 | AD | B8 | CF | D0 | D3 | A1 |
| C118 | EE | B6 | CF | 86 | 02 | AE | B6 | CF | BC | C300 | AD | B7 | CF | 8D | AF | CF | 20 | 79 | BF |
| C120 | BD | 00 | 02 | A6 | 02 | 48 | 68 | 60 | DB | C308 | C1 | AD | B8 | CF | D0 | C5 | AD | B7 | FE |
| C128 | 20 | E6 | C0 | 0D | 3E | 00 | 60 | 48 | 43 | C310 | CF | 8D | B0 | CF | 20 | 79 | C1 | AD | DE |
| C130 | A9 | 20 | 20 | A8 | C0 | A9 | 24 | 20 | 1C | C318 | B8 | CF | D0 | B7 | AD | B7 | CF | 8D | 3A |
| C138 | A6 | C0 | 68 | 85 | 02 | 4A | 4A | 4A | 84 | C320 | B1 | CF | 20 | 79 | C1 | AD | B8 | CF | 86 |
| C140 | 4A | 20 | 52 | C1 | 20 | A8 | C0 | A5 | 4F | C328 | D0 | A9 | AD | B7 | CF | 8D | B2 | CF | 41 |
| C148 | 02 | 29 | 0F | 20 | 52 | C1 | 20 | A8 | D2 | C330 | 20 | 79 | C1 | AD | B8 | CF | D0 | 9B | D9 |
| C150 | C0 | 60 | 09 | 30 | C9 | 3A | 90 | 03 | 54 | C338 | AD | B7 | CF | 8D | B3 | CF | 60 | 20 | 5C |
| C158 | 18 | 69 | 07 | 60 | 85 | 02 | A9 | 20 | 62 | C340 | 79 | C1 | AD | B7 | CF | 85 | 5C | AD | A7 |
| C160 | 20 | A8 | C0 | A9 | 25 | 20 | A8 | C0 | 7C | C348 | B8 | CF | 85 | 5D | CE | B6 | CF | 20 | D8 |
| C168 | A2 | 08 | A9 | 00 | 06 | 02 | 2A | 20 | 50 | C350 | 0F | C1 | C9 | 2C | F0 | 05 | A2 | 01 | 76 |
| C170 | 52 | C1 | 20 | A8 | C0 | CA | D0 | F2 | DE | C358 | 4C | BB | C0 | 20 | 79 | C1 | AD | B7 | DF |
| C178 | 60 | 20 | 0F | C1 | C9 | 24 | F0 | 11 | 11 | C360 | CF | 85 | 5E | AD | B8 | CF | 85 | 5F | 0A |
| C180 | C9 | 25 | F0 | 4B | 20 | FA | C1 | 10 | 1F | C368 | 60 | 20 | 3F | C3 | CE | B6 | CF | 20 | 83 |
| C188 | 05 | A2 | 01 | 4C | BB | C0 | CE | B6 | 2C | C370 | 0F | C1 | C9 | 2C | F0 | 05 | A2 | 01 | 76 |
| C190 | CF | 86 | 02 | A9 | 00 | 8D | B7 | CF | FE | C378 | 4C | BB | C0 | 20 | 79 | C1 | AD | B8 | DE |
| C198 | 8D | B8 | CF | 20 | 18 | C1 | 20 | FA | 02 | C380 | CF | F0 | 05 | A2 | 02 | 4C | BB | C0 | F4 |
| C1A0 | C1 | 30 | 23 | 0E | B7 | CF | 2E | B8 | 0A | C388 | A0 | 00 | AD | B7 | CF | 91 | 5C | E6 | 49 |
| C1A8 | CF | B0 | 4A | 0E | B7 | CF | 2E | B8 | B9 | C390 | 5C | D0 | 02 | E6 | 5D | A5 | 5C | C5 | F9 |
| C1B0 | CF | B0 | 42 | 0E | B7 | CF | 2E | B8 | B1 | C398 | 5E | A5 | 5D | E5 | 5F | 90 | E9 | 60 | 8A |
| C1B8 | CF | B0 | 3A | 0E | B7 | CF | 2E | B8 | A9 | C3A0 | EA | EA | 20 | 3F | C3 | CE | B6 | CF | BD |
| C1C0 | CF | B0 | 32 | 0D | B7 | CF | 8D | B7 | 02 | C3A8 | 20 | 0F | C1 | C9 | 2C | F0 | 05 | A2 | A9 |
| C1C8 | CF | 4C | 9B | C1 | A6 | 02 | 60 | 66 | DA | C3B0 | 01 | 4C | BB | C0 | 20 | 79 | C1 | AD | 6B |
| C1D0 | 02 | A9 | 00 | 8D | B7 | CF | 8D | B8 | 8A | C3B8 | B8 | CF | F0 | 05 | A2 | 02 | 4C | BB | 04 |
| C1D8 | CF | 20 | 18 | C1 | 20 | FA | C1 | 30 | BE | C3C0 | C0 | A0 | 00 | AD | B7 | CF | D1 | 5C | D0 |
| C1E0 | 11 | 0E | B7 | CF | 2E | B8 | CF | B0 | 80 | C3C8 | D0 | 0F | A9 | 0D | 20 | A8 | C0 | A5 | EF |
| C1E8 | 0C | 0D | B7 | CF | 8D | B7 | CF | 4C | 40 | C3D0 | 5D | 20 | 2F | C1 | A5 | 5C | 20 | 3B | D9 |
| C1F0 | D9 | C1 | A6 | 02 | 60 | A2 | 02 | 4C | 30 | C3D8 | C1 | E6 | 5C | D0 | 02 | E6 | 5D | A5 | 3D |
| C1F8 | BB | C0 | C9 | 3A | 30 | 07 | C9 | 41 | 3A | C3E0 | 5C | C5 | 5E | A5 | 5D | E5 | 5F | 90 | 98 |
| C200 | 30 | 0E | 38 | E9 | 07 | C9 | 30 | 30 | B1 | C3E8 | D8 | 60 | EA | EA | 20 | 79 | C1 | AD | 33 |
| C208 | 07 | C9 | 40 | 10 | 03 | 29 | 0F | 60 | F8 | C3F0 | B3 | CF | 48 | 28 | AE | B2 | CF | 9A | 35 |
| C210 | A9 | FF | 60 | A9 | 06 | 8D | 20 | D0 | 2C | C3F8 | AE | B1 | CF | AC | B0 | CF | AD | AF | FF |
| C218 | 8D | 21 | D0 | 20 | E6 | C0 | 93 | 9A | 3A | C400 | CF | 48 | AD | B7 | CF | 8D | B4 | CF | A3 |
| C220 | 00 | 20 | E6 | C0 | 0E | 08 | 0D | CC | 4D | C408 | AD | B9 | CF | 8D | B5 | CF | 68 | 6C | 19 |
| C228 | C9 | CE | C5 | C1 | D3 | D3 | 2D | 36 | F6 | C410 | B4 | CF | 68 | 68 | 20 | 79 | C1 | AD | A0 |
| C230 | 34 | 20 | 57 | 52 | 49 | 54 | 54 | 45 | 1D | C418 | B3 | CF | 48 | 28 | AE | B2 | CF | 9A | 35 |
| C238 | 4E | 20 | 31 | 39 | 38 | 35 | 20 | 42 | 07 | C420 | AE | B1 | CF | AC | B0 | CF | AD | AF | FF |
| C240 | 59 | 20 | CB | 4C | 41 | 55 | 53 | 20 | D7 | C428 | CF | 20 | 01 | C4 | 8D | AF | CF | 8C | 0D |
| C248 | C2 | 55 | 52 | 4B | 45 | 52 | 54 | 0D | AE | C430 | B0 | CF | 8E | B1 | CF | BA | 8E | B2 | AF |
| C250 | 20 | 20 | 20 | 20 | 49 | 52 | 51 | 20 | 28 | C438 | CF | 08 | 68 | 8D | B3 | CF | 20 | 21 | 85 |
| C258 | 20 | 20 | 50 | 43 | 20 | 20 | 20 | 41 | EC | C440 | C2 | 4C | 31 | C0 | 68 | A8 | 68 | AA | 65 |
| C260 | 20 | 20 | 20 | 58 | 20 | 20 | 20 | 58 | 90 | C448 | 68 | 8D | AF | CF | 8C | B0 | CF | 8E | D8 |

Super-Tool

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| C450 | B1 | CF | 68 | 8D | B3 | CF | D8 | 58 | 21 | C638 | 4C | BB | C0 | CE | B6 | CF | 20 | 0F | 7C |
| C458 | 68 | 38 | E9 | 02 | 8D | B4 | CF | 68 | 56 | C640 | C1 | C9 | 2C | D0 | F1 | 20 | 79 | C1 | DD |
| C460 | E9 | 00 | 8D | B5 | CF | BA | 8E | B2 | B1 | C648 | AD | B7 | CF | 85 | 5E | AD | B8 | CF | DA |
| C468 | CF | 20 | 21 | C2 | 4C | 31 | C0 | 20 | C9 | C650 | 85 | 5F | A9 | 0D | 20 | A8 | C0 | A9 | 51 |
| C470 | 0F | C1 | F0 | 49 | C9 | 24 | F0 | 67 | 22 | C658 | 3E | 20 | A8 | C0 | A9 | 4D | 20 | A8 | DA |
| C478 | C9 | 23 | F0 | 26 | AE | B6 | CF | A0 | 95 | C660 | C0 | A9 | 24 | 20 | A8 | C0 | A5 | 5D | 4B |
| C480 | 02 | 20 | 18 | C1 | D0 | FB | 86 | 02 | 93 | C668 | 20 | 3B | C1 | A5 | 5C | 20 | 3B | C1 | B7 |
| C488 | AD | B6 | CF | 38 | E5 | 02 | 20 | BD | D3 | C670 | A9 | 3A | 20 | A8 | C0 | A0 | 00 | A9 | 5E |
| C490 | FF | A9 | 07 | AE | B9 | CF | A0 | 0F | 2A | C678 | 20 | 20 | A8 | C0 | B1 | 5C | 20 | 3B | 22 |
| C498 | 20 | BA | FF | 20 | C0 | FF | 20 | E7 | 3F | C680 | C1 | C8 | C0 | 08 | 30 | F1 | A5 | 5C | 39 |
| C4A0 | FF | 60 | 20 | 79 | C1 | AD | B8 | CF | 42 | C688 | 18 | 69 | 09 | 95 | 5C | A5 | 5D | 69 | DE |
| C4A8 | D0 | 0E | AD | B7 | CF | 29 | F6 | D0 | 82 | C690 | 00 | 85 | 5D | A5 | 5C | 38 | E5 | 5E | DE |
| C4B0 | 07 | AD | B7 | CF | 8D | B9 | CF | 60 | 86 | C698 | A5 | 5D | E5 | 5F | 90 | B4 | 60 | 20 | E9 |
| C4B8 | A2 | 01 | 4C | BB | C0 | A9 | 0D | 20 | 36 | C6A0 | 79 | C1 | AD | B8 | CF | F0 | 05 | A2 | F0 |
| C4C0 | A8 | C0 | AD | B9 | CF | 85 | BA | 20 | C0 | C6A8 | 02 | 4C | BB | C0 | AD | B7 | CF | A0 | D6 |
| C4C8 | B4 | FF | A9 | 6F | 85 | B9 | 20 | 96 | 46 | C6B0 | 00 | 91 | 5C | E6 | 5C | D0 | 02 | E6 | 8F |
| C4D0 | FF | 20 | A5 | FF | 20 | A8 | C0 | C9 | F4 | C6B8 | 5D | CE | B6 | CF | 20 | 0F | C1 | 08 | 40 |
| C4D8 | 0D | D0 | F6 | 20 | AB | FF | 60 | A9 | 76 | C6C0 | CE | B6 | CF | 28 | D0 | D9 | 60 | 20 | F5 |
| C4E0 | 0D | 20 | A3 | C0 | AE | B6 | CF | A0 | FC | C6C8 | 0F | C1 | F0 | 10 | CE | B6 | CF | 20 | F4 |
| C4E8 | 02 | 20 | 18 | C1 | D0 | FB | 86 | 02 | 93 | C6D0 | 79 | C1 | AD | B7 | CF | 85 | 5C | AD | A7 |
| C4F0 | AD | B6 | CF | 38 | E5 | 02 | 20 | BD | D3 | C6D8 | B8 | CF | 85 | 5D | CE | B6 | CF | 20 | D8 |
| C4F8 | FF | A9 | 7F | AE | B9 | CF | A0 | 60 | 51 | C6E0 | 0F | C1 | D0 | 10 | A5 | 5C | 18 | 69 | 06 |
| C500 | 20 | BA | FF | 20 | C0 | FF | A5 | BA | F1 | C6E8 | 01 | 85 | 5E | A5 | 5D | 69 | 00 | 85 | A5 |
| C508 | 20 | B4 | FF | A5 | B9 | 20 | 96 | FF | F6 | C6F0 | 5F | 4C | 0E | C7 | C9 | 3A | F0 | 52 | 87 |
| C510 | A9 | 00 | 85 | 90 | A0 | 03 | 84 | 5E | 60 | C6F8 | C9 | 2C | F0 | 05 | A2 | 01 | 4C | BB | B8 |
| C518 | 20 | A5 | FF | 85 | 5F | A4 | 90 | D0 | 70 | C700 | C0 | 20 | 79 | C1 | AD | B7 | CF | 85 | 97 |
| C520 | 2F | 20 | A5 | FF | A4 | 90 | D0 | 20 | 71 | C708 | 5E | AD | B8 | CF | 85 | 5F | A9 | 0D | 5C |
| C528 | A4 | 5E | 88 | D0 | E9 | A6 | 5F | 20 | 80 | C710 | 20 | A8 | C0 | A9 | 3E | 20 | A8 | C0 | 95 |
| C530 | CD | BD | A9 | 20 | 20 | A8 | C0 | 20 | B1 | C718 | A9 | 42 | 20 | A8 | C0 | A9 | 24 | 20 | FA |
| C538 | A5 | FF | A6 | 90 | D0 | 12 | AA | F0 | 33 | C720 | A8 | C0 | A5 | 5D | 20 | 3B | C1 | A5 | 31 |
| C540 | 06 | 20 | A8 | C0 | 4C | 37 | C5 | A9 | FF | C728 | 5C | 20 | 3B | C1 | A9 | 3A | 20 | A8 | 9D |
| C548 | 0D | 20 | A9 | C0 | A0 | 02 | D0 | C6 | 7C | C730 | C0 | A0 | 00 | B1 | 5C | 20 | 5C | C1 | 46 |
| C550 | A9 | 49 | 85 | 5E | A9 | 01 | A2 | 5E | 72 | C738 | E6 | 5C | D0 | 02 | E6 | 5D | A5 | 5C | 28 |
| C558 | A0 | 00 | 20 | BD | FF | A9 | 07 | AE | B2 | C740 | 38 | E5 | 5E | A5 | 5D | E5 | 5F | 90 | 55 |
| C560 | B9 | CF | A0 | 0F | 20 | BA | FF | 20 | C0 | C748 | C5 | 60 | 20 | 79 | C1 | AD | B8 | CF | 09 |
| C568 | C0 | FF | 20 | E7 | FF | 60 | 20 | 0F | AB | C750 | F0 | 05 | A2 | 02 | 4C | BB | C0 | AD | 2E |
| C570 | C1 | F0 | 25 | C9 | 22 | D0 | F7 | A0 | D7 | C758 | B7 | CF | A0 | 00 | 91 | 5C | E6 | 5C | 46 |
| C578 | 02 | AE | B6 | CF | E8 | 86 | 02 | 20 | 80 | C760 | D0 | 02 | E6 | 5D | CE | B6 | CF | 20 | 1C |
| C580 | 0F | C1 | F0 | 14 | C9 | 22 | D0 | F7 | A9 | C768 | 0F | C1 | 08 | CE | B6 | CF | 28 | D0 | C9 |
| C588 | AD | B6 | CF | 38 | E5 | 02 | 20 | BD | D3 | C770 | D9 | 60 | 20 | 0F | C1 | F0 | 10 | CE | 9D |
| C590 | FF | 20 | 0F | C1 | C3 | 2C | F0 | 05 | B4 | C778 | B6 | CF | 20 | 79 | C1 | AD | B7 | CF | 8A |
| C598 | A2 | 01 | 4C | BB | C0 | 20 | 79 | C1 | 89 | C780 | 85 | 5C | AD | B8 | CF | 85 | 5D | CE | F7 |
| C5A0 | AD | B8 | CF | F0 | 05 | A2 | 02 | 4C | EE | C788 | B6 | CF | 20 | 0F | C1 | D0 | 10 | A5 | 55 |
| C5A8 | BB | C0 | AD | B7 | CF | C9 | 14 | B0 | 5B | C790 | 5C | 18 | 69 | 01 | 85 | 5E | A5 | 5D | 1A |
| C5B0 | F4 | AA | A0 | 01 | A9 | 7F | 20 | BA | 78 | C798 | 69 | 00 | 85 | 5F | 4C | B9 | C7 | C9 | 20 |
| C5B8 | FF | 60 | 20 | 6E | C5 | A9 | 00 | 20 | 4D | C7A0 | 3A | F0 | 8A | C9 | 2C | F0 | 05 | A2 | 8C |
| C5C0 | D5 | FF | 4C | F3 | C5 | 20 | 6E | C5 | 7D | C7A8 | 01 | 4C | BB | C0 | 20 | 79 | C1 | AD | 6B |
| C5C8 | A9 | 01 | 20 | D5 | FF | 4C | F3 | C5 | D3 | C7B0 | B7 | CF | 95 | 5E | AD | B8 | CF | 85 | 4E |
| C5D0 | 20 | 6E | C5 | CE | B6 | CF | 20 | 0F | A2 | C7B8 | 5F | A9 | 0D | 20 | A8 | C0 | A9 | 3E | F6 |
| C5D8 | C1 | C9 | 2C | F0 | 05 | A2 | 01 | 4C | 4E | C7C0 | 20 | A8 | C0 | A9 | 43 | 20 | A8 | C0 | 9A |
| C5E0 | BB | C0 | A9 | 0D | 20 | A8 | C0 | 20 | AF | C7C8 | A9 | 24 | 20 | A8 | C0 | A5 | 5D | 20 | 55 |
| C5E8 | 3F | C3 | A9 | 5C | A6 | 5E | A4 | 5F | 56 | C7D0 | 3B | C1 | A5 | 5C | 20 | 3B | C1 | A9 | C0 |
| C5F0 | 20 | D8 | FF | 80 | 07 | A5 | 90 | 29 | 61 | C7D8 | 3A | 20 | A8 | C0 | A0 | 00 | B1 | 5C | F6 |
| C5F8 | 3F | D0 | 01 | 60 | A2 | 03 | 4C | BB | 41 | C7E0 | 85 | 02 | 29 | 60 | D0 | 04 | A9 | 20 | A0 |
| C600 | C0 | 20 | 0F | C1 | F0 | 10 | CE | B6 | E5 | C7E8 | 85 | 02 | A5 | 02 | 20 | A8 | C0 | C8 | 95 |
| C608 | CF | 20 | 79 | C1 | AD | B7 | CF | 85 | A6 | C7F0 | C0 | 10 | 30 | EA | A5 | 5C | 18 | 69 | EE |
| C610 | 5C | AD | B8 | CF | 85 | 5D | CE | B6 | D8 | C7F8 | 10 | 85 | 5C | A5 | 5D | 69 | 00 | 65 | B2 |
| C618 | CF | 20 | 0F | C1 | D0 | 10 | A5 | 5C | 05 | C800 | 5D | A5 | 5C | 38 | E5 | 5E | A5 | 5D | AA |
| C620 | 18 | 69 | 01 | 85 | 5E | A5 | 5D | 69 | D9 | C808 | E5 | 5F | 90 | AD | 60 | A0 | 00 | 20 | 09 |
| C628 | 00 | 85 | 5F | 4C | 52 | C6 | C9 | 3A | AA | C810 | 18 | C1 | F0 | 1D | 85 | 02 | C9 | 20 | 55 |
| C630 | F0 | 6D | C9 | 2C | F0 | 05 | A2 | 01 | AA | C818 | D0 | 0A | B1 | 5C | 29 | 60 | D0 | 04 | AF |

Super-Tool

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| C820 | B1 | 5C | 85 | 02 | A5 | 02 | 91 | 5C | AE | CA08 | 1F | 0B | 38 | 4C | 1F | 4B | 38 | 20 | EC |
| C828 | E6 | 5C | D0 | E1 | E6 | 5D | 4C | 0D | 40 | CA10 | 1F | 3A | 38 | 4B | 1F | 4B | 38 | 20 | BF |
| C830 | C8 | 60 | 41 | 41 | 41 | 41 | 41 | 41 | 68 | CA18 | 1F | 01 | 02 | 02 | 02 | 02 | 02 | 02 | 1E |
| C838 | 41 | 41 | 41 | 42 | 42 | 42 | 42 | 42 | FF | CA20 | 02 | 03 | 03 | 03 | 03 | 8D | BA | CF | 60 |
| C840 | 42 | 42 | 42 | 42 | 42 | 43 | 43 | 43 | FF | CA28 | C9 | 6C | D0 | 03 | A9 | 0C | 60 | C9 | 5C |
| C848 | 43 | 43 | 43 | 43 | 44 | 44 | 44 | 44 | 00 | CA30 | 89 | D0 | 03 | A9 | 03 | 60 | 29 | 1F | C2 |
| C850 | 45 | 49 | 49 | 49 | 49 | 4A | 4A | 4C | F9 | CA38 | AA | AD | BA | CF | C9 | 00 | 10 | 0A | B7 |
| C858 | 4C | 4C | 4C | 4C | 4E | 4F | 50 | 50 | FF | CA40 | C9 | 00 | 10 | 06 | BD | 8E | CA | 4C | BF |
| C860 | 50 | 50 | 52 | 52 | 52 | 52 | 52 | 52 | 00 | CA48 | 4D | CA | BD | 6E | CA | E0 | 00 | F0 | CD |
| C868 | 53 | 53 | 53 | 53 | 53 | 53 | 53 | 53 | 00 | CA50 | 0F | E0 | 02 | F0 | 12 | 60 | AD | BA | E8 |
| C870 | 53 | 53 | 53 | 53 | 53 | 53 | 54 | 54 | 00 | CA58 | CF | 4A | 4A | 4A | 4A | 4A | AA | 60 | CE |
| C878 | 54 | 54 | 54 | 54 | 3F | 2A | 5E | 44 | 2F | CA60 | 20 | 56 | CA | BD | AE | CA | 60 | 20 | FB |
| C880 | 4E | 4E | 52 | 53 | 53 | 53 | 53 | 58 | FA | CA68 | 56 | CA | BD | B6 | CA | 60 | 00 | 07 | F7 |
| C888 | 43 | 43 | 45 | 49 | 4D | 4E | 50 | 52 | F9 | CA70 | 00 | 07 | 03 | 03 | 03 | 03 | 01 | 02 | F9 |
| C890 | 56 | 56 | 4C | 4C | 4C | 4C | 4D | 50 | FD | CA78 | 01 | 02 | 09 | 09 | 09 | 09 | 06 | 08 | FD |
| C898 | 50 | 43 | 45 | 45 | 45 | 4F | 43 | 4E | F8 | CA80 | 01 | 08 | 03 | 04 | 04 | 04 | 01 | 0B | EF |
| C8A0 | 4E | 4E | 4D | 53 | 41 | 44 | 44 | 44 | F7 | CA88 | 01 | 0B | 09 | 0A | 0A | 0A | 00 | 07 | EF |
| C8A8 | 53 | 4F | 52 | 48 | 48 | 4C | 4C | 4C | 0A | CA90 | 00 | 07 | 03 | 03 | 03 | 03 | 01 | 02 | F9 |
| C8B0 | 4F | 4F | 52 | 54 | 54 | 41 | 41 | 42 | 10 | CA98 | 01 | 02 | 09 | 09 | 09 | 09 | 06 | 08 | FD |
| C8B8 | 45 | 45 | 45 | 4C | 52 | 53 | 54 | 54 | F8 | CAA0 | 01 | 08 | 04 | 04 | 05 | 05 | 01 | 0B | F0 |
| C8C0 | 54 | 58 | 59 | 41 | 41 | 53 | 58 | 58 | 02 | CAA8 | 01 | 0B | 0A | 0A | 0B | 0B | 01 | 09 | EF |
| C8C8 | 59 | 3F | 2A | 5E | 43 | 43 | 44 | 52 | D8 | CAB0 | 01 | 01 | 03 | 02 | 02 | 02 | 01 | 01 | 01 |
| C8D0 | 4C | 52 | 54 | 58 | 41 | 43 | 53 | 51 | F6 | CAB8 | 01 | 01 | 03 | 02 | 03 | 03 | 20 | 0F | 12 |
| C8D8 | 54 | 49 | 45 | 4C | 4B | 43 | 53 | 43 | 1C | CAC0 | C1 | F0 | 10 | CE | B6 | CF | 20 | 79 | A2 |
| C8E0 | 44 | 49 | 56 | 50 | 58 | 59 | 43 | 43 | 00 | CAC8 | C1 | AD | B7 | CF | 85 | 5C | AD | B8 | 1A |
| C8E8 | 58 | 59 | 52 | 53 | 43 | 58 | 59 | 50 | F2 | CAD0 | CF | 85 | 5D | CE | B6 | CF | 20 | 0F | D2 |
| C8F0 | 52 | 58 | 41 | 58 | 59 | 52 | 50 | 41 | F9 | CAD8 | C1 | D0 | 0C | A9 | 17 | 85 | FF | 20 | C6 |
| C8F8 | 41 | 50 | 41 | 50 | 41 | 4C | 52 | 41 | E8 | CAE0 | 13 | CB | C6 | FF | D0 | F9 | 60 | C9 | 7E |
| C900 | 49 | 53 | 48 | 58 | 43 | 43 | 44 | 49 | E1 | CAE8 | 3A | D0 | 03 | 4C | B6 | CC | C9 | 2C | A9 |
| C908 | 4F | 45 | 48 | 41 | 58 | 59 | 48 | 48 | 10 | CAF0 | F0 | 05 | A2 | 01 | 4C | BB | C0 | 20 | BC |
| C910 | 58 | 59 | 58 | 41 | 53 | 41 | 3F | 2A | 3D | CAF8 | 79 | C1 | AD | B7 | CF | 85 | 5E | AD | A9 |
| C918 | 5E | 10 | 2B | 4C | 3C | 4B | 2B | 04 | 45 | CB00 | B8 | CF | 85 | 5F | 20 | 13 | CB | A5 | 42 |
| C920 | 3C | 2D | 2B | 04 | 01 | 4B | 2B | 04 | 13 | CB08 | 5C | 38 | E5 | 5E | A5 | 5D | E5 | 5F | 78 |
| C928 | 3C | 0F | 2B | 4C | 3C | 4B | 2B | 04 | 24 | CB10 | 90 | F2 | 60 | A9 | 0D | 20 | A8 | C0 | 2B |
| C930 | 3C | 13 | 2B | 4B | 3C | 4B | 2B | 04 | 21 | CB18 | A9 | 3E | 20 | A8 | C0 | A9 | 44 | 20 | 1E |
| C938 | 3C | 24 | 02 | 4C | 30 | 0C | 02 | 31 | C4 | CB20 | A8 | C0 | A9 | 24 | 20 | A8 | C0 | A5 | 00 |
| C940 | 30 | 2F | 02 | 31 | 01 | 0C | 02 | 31 | 99 | CB28 | 5D | 20 | 3B | C1 | A5 | 5C | 20 | 3B | E5 |
| C948 | 30 | 0D | 02 | 4C | 30 | 4B | 02 | 31 | 90 | CB30 | C1 | A9 | 3A | 20 | A8 | C0 | A0 | 00 | BA |
| C950 | 30 | 39 | 02 | 4B | 30 | 4B | 02 | 31 | 65 | CB38 | B1 | 5C | 20 | 2F | C1 | B1 | 5C | 20 | 92 |
| C958 | 30 | 34 | 1E | 4C | 3D | 4B | 1E | 29 | B6 | CB40 | 25 | CA | AA | CA | BD | 19 | CA | AA | FF |
| C960 | 3D | 2C | 1E | 23 | 05 | 23 | 1E | 29 | DD | CB48 | E0 | 01 | F0 | 10 | C8 | B1 | 5C | 20 | 10 |
| C968 | 3D | 11 | 1E | 4C | 3D | 4B | 1E | 29 | E5 | CB50 | 2F | C1 | E0 | 02 | F0 | 06 | C8 | B1 | 4C |
| C970 | 3D | 15 | 1E | 4B | 3D | 4B | 1E | 29 | E2 | CB58 | 5C | 20 | 2F | C1 | A9 | 20 | 20 | A8 | AB |
| C978 | 3D | 35 | 00 | 4C | 33 | 4B | 00 | 32 | 73 | CB60 | C0 | A5 | D3 | C9 | 15 | 30 | F5 | A9 | 56 |
| C980 | 33 | 2E | 00 | 32 | 03 | 23 | 00 | 32 | 82 | CB68 | 3B | 20 | A8 | C0 | A0 | 00 | B1 | 5C | F7 |
| C988 | 33 | 12 | 00 | 4C | 33 | 4B | 00 | 32 | 8C | CB70 | AA | BD | 19 | C9 | AA | BD | 32 | C8 | 95 |
| C990 | 33 | 3B | 00 | 4B | 33 | 4B | 00 | 32 | 64 | CB78 | 20 | A8 | C0 | BD | 7F | C8 | 20 | A8 | AB |
| C998 | 33 | 4B | 3F | 4B | 37 | 41 | 3F | 40 | D1 | CB80 | C0 | BD | CC | C8 | 20 | A8 | C0 | A9 | 96 |
| C9A0 | 37 | 1D | 4B | 47 | 08 | 41 | 3F | 40 | E4 | CB88 | 20 | 20 | A8 | C0 | AD | BA | CF | 20 | 8A |
| C9A8 | 37 | 09 | 3F | 4C | 36 | 41 | 3F | 40 | 15 | CB90 | 25 | CA | AA | E0 | 01 | D0 | 03 | 4C | 0F |
| C9B0 | 37 | 49 | 3F | 48 | 3E | 43 | 3F | 42 | DD | CB98 | 93 | CC | E0 | 02 | D0 | 05 | A9 | 23 | F4 |
| C9B8 | 36 | 28 | 26 | 27 | 25 | 28 | 26 | 27 | 09 | CBA0 | 20 | A8 | C0 | E0 | 07 | F0 | 08 | E0 | 99 |
| C9C0 | 25 | 45 | 26 | 44 | 4A | 28 | 26 | 27 | E3 | CBA8 | 08 | F0 | 04 | E0 | 0C | D0 | 05 | A9 | D7 |
| C9C8 | 25 | 0A | 26 | 4C | 25 | 28 | 26 | 27 | F1 | CBB0 | 28 | 20 | A8 | C0 | A9 | 24 | 20 | A8 | ED |
| C9D0 | 25 | 16 | 26 | 46 | 06 | 28 | 26 | 27 | CD | CBB8 | C0 | E0 | 09 | 30 | 03 | 4C | 5A | CC | FF |
| C9D8 | 25 | 19 | 17 | 4B | 1A | 19 | 17 | 1B | D6 | CBC0 | E0 | 06 | F0 | 58 | A0 | 01 | B1 | 5C | 64 |
| C9E0 | 1A | 22 | 17 | 1C | 07 | 19 | 17 | 1B | DE | CBC8 | 20 | 3B | C1 | E0 | 04 | 10 | 03 | 4C | 72 |
| C9E8 | 1A | 0E | 17 | 4C | 1A | 4B | 17 | 1B | A3 | CBD0 | 93 | CC | E0 | 04 | D0 | 0D | A9 | 2C | E1 |
| C9F0 | 1A | 14 | 17 | 4B | 1A | 4B | 17 | 1B | 9E | CBD8 | 20 | A8 | C0 | A9 | 58 | 20 | A8 | C0 | AF |
| C9F8 | 1A | 18 | 38 | 4B | 1F | 18 | 38 | 20 | 0E | CBE0 | 4C | 93 | CC | E0 | 05 | D0 | 0D | A9 | 40 |
| CA00 | 1F | 21 | 38 | 2A | 38 | 18 | 38 | 20 | 44 | CBE8 | 2C | 20 | A8 | C0 | A9 | 59 | 20 | A8 | BC |

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| CBF0 | C8 | 4C | 93 | CC | E0 | 07 | D0 | 12 | D1 | CDD0 | 38 | E5 | 5E | 8D | B7 | CF | AA | AD | 0A |
| CBF8 | A9 | 2C | 20 | A8 | C0 | A9 | 58 | 20 | 44 | CDD8 | B8 | CF | E5 | 5F | E0 | 80 | 69 | 00 | 37 |
| CC00 | A8 | C0 | A9 | 29 | 20 | A8 | C0 | 4C | 54 | CDE0 | EA | EA | F0 | 05 | A9 | 00 | 8D | BE | 61 |
| CC08 | 93 | CC | A9 | 29 | 20 | A8 | C0 | A9 | D6 | CDE8 | CF | A9 | 00 | 8D | B8 | CF | AD | B8 | 77 |
| CC10 | 2C | 20 | A8 | C0 | A9 | 59 | 20 | A8 | BC | CDF0 | CF | D0 | 11 | AD | BE | CF | C9 | 09 | 12 |
| CC18 | C0 | 4C | 93 | CC | A5 | 5C | 18 | 69 | 33 | CDF8 | 30 | 0A | C9 | 0C | 10 | 06 | 38 | E9 | 38 |
| CC20 | 02 | 8D | B7 | CF | A5 | 5D | 69 | 00 | 0E | CE00 | 06 | 8D | BE | CF | AD | BB | CF | C9 | 60 |
| CC28 | 8D | B8 | CF | A0 | 01 | B1 | 5C | 10 | A1 | CE08 | 2A | F0 | 04 | C9 | 5E | D0 | 05 | A9 | 61 |
| CC30 | 04 | A9 | FF | 85 | 02 | B1 | 5C | 30 | 53 | CE10 | 00 | 8D | BE | CF | A2 | 4D | CA | 30 | 51 |
| CC38 | 04 | A9 | 00 | 85 | 02 | B1 | 5C | 18 | 6D | CE18 | 4D | AD | BB | CF | DD | 32 | C8 | D0 | 2F |
| CC40 | 6D | B7 | CF | 8D | B7 | CF | A5 | 02 | 83 | CE20 | F5 | AD | BC | CF | DD | 7F | C8 | D0 | 8A |
| CC48 | 6D | B8 | CF | 8D | B8 | CF | 20 | 3B | C5 | CE28 | ED | AD | BD | CF | DD | CC | C8 | D0 | 36 |
| CC50 | C1 | AD | B7 | CF | 20 | 3B | C1 | 4C | 56 | CE30 | E5 | 8A | 8D | BF | CF | A0 | 00 | F0 | 68 |
| CC58 | 93 | CC | A0 | 02 | B1 | 5C | 20 | 3B | 9F | CE38 | 03 | C8 | F0 | 16 | AD | BF | CF | D9 | F9 |
| CC60 | C1 | A0 | 01 | B1 | 5C | 20 | 38 | C1 | 28 | CE40 | 19 | C9 | D0 | F5 | 38 | 20 | 25 | CA | FF |
| CC68 | E0 | 09 | F0 | 27 | E0 | 0A | D0 | 0D | 36 | CE48 | AE | BA | CF | CD | BE | CF | F0 | 1B | 8A |
| CC70 | A9 | 2C | 20 | A8 | C0 | A9 | 58 | 20 | 44 | CE50 | D0 | E7 | AD | BE | CF | C9 | 03 | 30 | B2 |
| CC78 | A8 | C0 | 4C | 93 | CC | E0 | 0B | D0 | C9 | CE58 | 0D | C9 | 06 | 10 | 09 | 18 | 69 | 06 | 8F |
| CC80 | 0D | A9 | 2C | 20 | A8 | C0 | A9 | 59 | A8 | CE60 | 8D | BE | CF | 4C | 14 | CE | A9 | 00 | 41 |
| CC88 | 20 | A8 | C0 | 4C | 93 | CC | A9 | 29 | 33 | CE68 | 8D | BE | CF | 8E | BA | CF | AD | BE | EA |
| CC90 | 20 | A8 | C0 | A0 | 00 | B1 | 5C | 20 | 24 | CE70 | CF | F0 | 21 | AA | CA | BD | 19 | CA | B3 |
| CC98 | 25 | CA | AA | CA | BD | 19 | CA | 18 | 91 | CE78 | AA | AD | BA | CF | A0 | 00 | 91 | 5C | BC |
| CCA0 | 65 | 5C | 85 | 5C | A5 | 5D | 69 | 00 | E3 | CE80 | E0 | 01 | F0 | 10 | AD | B7 | CF | C8 | 8A |
| CCA8 | 95 | 5D | A3 | 20 | 20 | A8 | C0 | A5 | 44 | CE88 | 91 | 5C | E0 | 02 | F0 | 06 | AD | B8 | F0 |
| CCB0 | D3 | C9 | 27 | 30 | F5 | 60 | A9 | 00 | 3E | CE90 | CF | C8 | 91 | 5C | C6 | D6 | 20 | 13 | 39 |
| CCB8 | 8D | BE | CF | 20 | 0F | C1 | F0 | 06 | B6 | CE98 | CB | A9 | 0D | 20 | A8 | C0 | A9 | 3E | 62 |
| CCC0 | C9 | 3B | D0 | F2 | F0 | 05 | A2 | 01 | F7 | CEA0 | 20 | A8 | C0 | A9 | 44 | 20 | A8 | C0 | 9B |
| CCC8 | 4C | BB | C0 | 20 | 18 | C1 | D0 | 03 | 55 | CEA8 | A9 | 24 | 20 | A8 | C0 | A5 | 5D | 20 | 55 |
| CCD0 | 4C | 94 | CE | 8D | BB | CF | 20 | 18 | ED | CEB0 | 3B | C1 | A5 | 5C | 20 | 3B | C1 | A9 | C0 |
| CCD8 | C1 | D0 | 03 | 4C | 94 | CE | 8D | BC | 40 | CEB8 | 3A | 20 | A8 | C0 | A9 | 3B | 20 | A8 | E8 |
| CCE0 | CF | 20 | 18 | C1 | D0 | 03 | 4C | 94 | 8A | CEC0 | C0 | A9 | 20 | 20 | A8 | C0 | A5 | D3 | D1 |
| CCE8 | CE | 8D | BD | CF | 20 | 0F | C1 | D0 | 30 | CEC8 | C9 | 27 | 30 | F5 | C6 | D6 | A2 | 07 | 68 |
| CCF0 | 08 | A2 | 01 | 8E | BE | CF | 4C | 99 | 7C | CED0 | A9 | 1D | 9D | 77 | 02 | CA | 10 | FA | 01 |
| CCF8 | CD | C9 | 23 | D0 | 08 | A2 | 02 | 8E | 33 | CED8 | A9 | 08 | 85 | C6 | 60 | 20 | 79 | C1 | 57 |
| CD00 | BE | CF | 4C | 24 | CD | C9 | 28 | D0 | 73 | CEE0 | AD | B7 | CF | 85 | 5C | AD | B8 | CF | D8 |
| CD08 | 1B | AE | BB | CF | E0 | 4A | D0 | 0F | B0 | CEE8 | 85 | 5D | CE | B6 | CF | 20 | 0F | C1 | 3D |
| CD10 | AE | BC | CF | E0 | 4D | D0 | 08 | A2 | C5 | CEF0 | C9 | 2C | F0 | 05 | A2 | 01 | 4C | BB | B8 |
| CD18 | 0C | 8E | BE | CF | 4C | 24 | CD | A2 | C0 | CEF8 | C0 | 20 | 79 | C1 | AD | B7 | CF | 8D | 8F |
| CD20 | 80 | 8E | BE | CF | AD | BE | CF | D0 | CF | CF00 | C0 | CF | AD | B8 | CF | 8D | C1 | CF | 1A |
| CD28 | 03 | CE | B6 | CF | 20 | 79 | C1 | CE | B7 | CF08 | CE | B6 | CF | 20 | 0F | C1 | C9 | 2C | B2 |
| CD30 | B6 | CF | 20 | 0F | C1 | D0 | 0A | AE | 46 | CF10 | D0 | E2 | 20 | 79 | C1 | AD | B7 | CF | 91 |
| CD38 | BE | CF | D0 | 05 | A2 | 09 | 8E | BE | 22 | CF18 | 85 | 5E | AD | B8 | CF | 85 | 5F | A5 | 20 |
| CD40 | CF | C9 | 29 | D0 | 22 | AE | BE | CF | C3 | CF20 | 5E | 38 | E5 | 5C | A5 | 5F | E5 | 5D | 7C |
| CD48 | E0 | 0C | F0 | 4D | 2C | BE | CF | 10 | A3 | CF28 | B0 | 20 | A0 | 00 | B1 | 5C | 91 | 5E | B6 |
| CD50 | 48 | 20 | 0F | C1 | C9 | 2C | D0 | 41 | A2 | CF30 | E6 | 5C | D0 | 02 | E6 | 5D | E6 | 5E | 67 |
| CD58 | 20 | 0F | C1 | C9 | 59 | D0 | 3A | A9 | 24 | CF38 | D0 | 02 | E6 | 5F | A5 | 5C | 38 | ED | E8 |
| CD60 | 08 | 8D | BE | CF | 4C | 99 | CD | C9 | 22 | CF40 | C0 | CF | A5 | 5D | ED | C1 | CF | 90 | A3 |
| CD68 | 2C | D0 | 2E | 20 | 0F | C1 | C9 | 59 | 29 | CF48 | E1 | 60 | AD | C0 | CF | 18 | E5 | 5C | AC |
| CD70 | D0 | 08 | A9 | 08 | 8D | BE | CF | 4C | B6 | CF50 | 95 | 02 | AD | C1 | CF | E5 | 5D | 48 | 6E |
| CD78 | 99 | CD | C9 | 58 | D0 | 1B | 20 | 0F | 02 | CF58 | 18 | A5 | 02 | 65 | 5E | 85 | 5E | 68 | E0 |
| CD80 | C1 | D0 | 08 | A9 | 0A | 8D | BE | CF | 8D | CF60 | 65 | 5F | 85 | 5F | AD | C0 | CF | A6 | 42 |
| CD88 | 4C | 99 | CD | C9 | 29 | D0 | 0A | 2C | EF | CF68 | 5C | 95 | 5C | 8E | C0 | CF | AD | C1 | 82 |
| CD90 | BE | CF | 10 | 05 | A9 | 07 | 8D | BE | 6A | CF70 | CF | A6 | 5D | 85 | 5D | 8E | C1 | CF | C2 |
| CD98 | CF | AD | BE | CF | 29 | 7F | 8D | BE | 8A | CF78 | A5 | 5C | 38 | E9 | 01 | 85 | 5C | A5 | CC |
| CDA0 | CF | AD | BE | CF | C9 | 09 | D0 | 46 | 5A | CF80 | 5D | E9 | 00 | 85 | 5D | A0 | 00 | R1 | FN |
| CDA8 | AD | BB | CF | C9 | 42 | D0 | 3F | AD | FD | CF88 | 5C | 91 | 5E | A5 | 5C | 38 | E9 | 01 | 90 |
| CDB0 | BC | CF | C9 | 49 | F0 | 38 | A9 | 08 | C7 | CF90 | 85 | 5C | B0 | 02 | C6 | 5D | A5 | 5E | 86 |
| CDB8 | 8D | BE | CF | EA | EA | EA | EA | EA | B4 | CF98 | 38 | E9 | 01 | 85 | 5E | B0 | 02 | C6 | B7 |
| CDC0 | A5 | 5C | 18 | 69 | 02 | 85 | 5E | A5 | 2F | CFA0 | 5F | A5 | 5C | 38 | ED | C0 | CF | A5 | 35 |
| CDC8 | 5D | 69 | 00 | 85 | 5F | AD | B7 | CF | 0A | CFA8 | 5D | ED | C1 | CF | B0 | D7 | 60 | 0D | 8F |

```

.inpt 'startadresse=? ',start
.offb %c000 - start
*= start
status=%90
sa  =%b9
ga  =%ba
tbuf =%0277
irqv =%0314
brkv =%0316
border=%d020
screen=%d021
gchar =%f15b
sectik=%ff96
iecin =%ffa5
untik =%ffab
talk  =%ffb4
setpar=%ffba
setnam=%ffbd
open  =%ffc0
close =%ffc3
bsout =%ffd2
cload =%ffd5
osave =%ffd8
stkey =%ffe1
ciall =%ffe7
    lda #000      ;Register setzen
    sta ac
    sta yr
    sta xr
    sta pc
    sta pc+1
    sta %5c
    sta %5d
    ldx #%f6
    stx sp
    php
    pia
    sta ps
    jsr mess      ;Startmeldung
    lda < < break ;Break setzen
    sta brkv
    lda > > break
    sta brkv+1
    lda #08
    sta device
comm  ldx sp
    txs
    jsr prmpt     ;Prompt drucken
    ldx #00       ;Eingabe vom Bild-
in    jsr gchar   ;schirm in Puffer
    sta %0200,x
    inx
    cmp #0d
    bne in
    lda #00
    sta %01ff,x
    lda #ff      ;Test auf leere
    sta epoin    ;Zeile
get1  jsr eget     ;Zeichen aus Puffer
    beq comm
    cmp #'>      ;Test auf Prompt
    beq get1
    ldx #0f      ;Befehisauswertung
com1  cmp com1,x
    beq com2
    dex
    bpl com1
    inx
    jmp error    ;CM-Eroor
com2  txa         ;Befehisadresse
    asl a        ;holen
    tax
    jsr com3
    jmp comm
com3  lda comma+1,x ;Befehl ausfuehren
    pha
    lda comma,x
    pha
    rts
commt .text "0bdefghimqrstvx"
comma .word dos-1,bin-1,chr-1,dass-1
        .word exec-1,fill-1,go-1,hunt-1
        .word load-1,mem-1,quit-1,reg-1
        .word save-1,tran-1,veri-1,exit-1
chout jsr bsout   ;Zeichen ausgeben
    pha
    jsr stkey
    bne st2
st1   jsr stkey
    beq st1
    jmp comm
st2   pla
    rts
error jsr print   ;Error ausgeben
    .byte $0d,'?',$20,$00
    txa
    asl a
    tax
    lda erra,x
    jsr chout
    lda erra+1,x
    jsr chout
    jsr print
    .text " error"
    .byte $00
    jmp comm
erra .text "macrovio"
print sta %02     ;Text ausgeben
    sta %5e
    pia
    sta %5f
    tya
    pha
    ldy #00
    jsr incr
    lda (%5e),y
    beq p2
    jsr chout
    bne p1
p2    jsr incr
    pia
    tay
    lda %02
    jmp (%5e)
incr  inc %5e
    bne i1
    inc %5f
i1    rts
eget  jsr get      ;Zeichen aus Puffer
    cmp #20       ;lesen <> Space
    beq eget
    bne get2
get   inc epoin    ;Zeichen aus Puffer
    stx %02       ;lesen
    ldx epoin
    lda %0200,x
    ldx %02
get2  pha
    pia
    rts
prmpt jsr print   ;Prompt drucken
    .byte $0d,'>',$00
    rts
hout  pha         ;i Byte in hex aus-
    lda #20       ;ben mit $
    jsr chout
    lda #'$
    jsr chout
    pia
hout1 sta %02     ;dto. ohne $
    lsr a
    lsr a
    lsr a
    jsr asc
    jsr chout
    lda %02
    and #0f
    jsr asc
    jsr chout
    rts
asc   ora #30     ;Nibble > ASCII
    cmp #3a
    bcc asci
    clc
    adc #07
asc1  rts
bout  sta %02     ;i Byte binaer aus-
    lda #20       ;geben mit %
    jsr chout
    lda #'%
    jsr chout
    ldx #08
    lda #00
    asl %02
    rol a
    jsr asc
    jsr chout
    dex
    bne bi
    jmp error
gnum  jsr eget     ;16-Bit-Zahl aus
    cmp #'$
    beq hexz      ;Puffer holen und
    cmp #'X       ;in numl/numh ab-
    beq binz      ;legen
    jsr hex
    bpl gn1
    ldx #01
    jmp error
gn1   dec epoin
hexz  stx %02     ;Hex-Zahl
    lda #00
    sta numl
    sta numh
he1   jsr get
    jsr hex
    bpl he2
    asl numl
    rol numh
    bcs gn2
    asl numl
    rol numh
    bcs gn2
    asl numl
    rol numh
    bcs gn2
    ora numl
    sta numl
    jmp he1
he2   ldx %02
    rts
binz  stx %02     ;Binaer-Zahl
    lda #00
    sta numl
    sta numh
bi1   jsr get
    jsr hex
    bmi bi2
    asl numl
    rol numh
    bcs gn2
    ora numl
    sta numl
    jmp bi1
bi2   ldx %02
    rts
gn2   ldx #02     ;0V-Error
    jmp error
hex   cmp #3a
    bmi hex1
    cmp #41
    bmi hex2
    sec
    sbc #07
    cmp #30
    bmi hex2
    cmp #40
    bpl hex2
    and #0f
    rts
hex2  lda #ff
    rts
mess  lda #06     ;Startmeldung drucken
    sta border
    sta screen
    jsr print
    .byte %93,%9a,$00
mess1 jsr print
    .byte %0e,%08,$0d
    .text "LINEASS-64 written 1985 by"
    .text " Klaus Burkert"
    .byte %0d
    .text "      irq  pc  a  y  x"
    .text "      s  nv-bdizc"
    .byte %0d
    .text ">":
    .byte %00
    lda #20      ;IRQ-Vektor drucken
    jsr chout
    lda irqv+1
    jsr hout1
    lda irqv
    jsr hout1
    lda #20      ;PC drucken
    jsr chout

```

Super-Tool

| | | |
|---|---|--|
| <pre> lda pc+1 jsr hout1 lda pc jsr hout1 lda ac ;Akku drucken jsr hout lda yr ;Y drucken jsr hout lda xr ;X drucken jsr hout lda sp ;Stackpointer drucken jsr hout lda ps ;Statusreg. drucken jsr bout rts exit lda #37 ;BASIC- und KERNAL- sta #01 ;ROM ein jmp #a474 ;Basic Warmstart quit jmp (\$ffc) ;Software-Reset reg jsr eget bne reg1 ;wenn nur r ,dann jmp mess1 ;Einsprungmeldung cmp #' ;wenn r: ,dann Ein- beq reg1 ;gabe der Register reger idx #01 ;sonst CR-Error jmp error reg1 jsr gnum ;IRQ-Vektor sei lda numh sta irqv+1 lda numl sta irqv cli jsr gnum ;PC lda numl sta pc lda numh sta pc+1 jsr gnum ;Akku lda numh bne reger lda numl sta ac jsr gnum ;Y-Register lda numh bne reger lda numl sta yr jsr gnum ;X-Register lda numh bne reger lda numl sta sp jsr gnum ;Statusregister lda numh bne reger lda numl sta ps rts gadr jsr gnum ;Startadresse lda numl sta \$5c lda numh sta \$5d dec epoin jsr eget ;Test auf Komma cmp #' beq gad idx #01 jmp error gad jsr gnum ;Endadresse+1 lda numl sta \$5e lda numh sta \$5f rts fill jsr gadr ;Adressen holen dec epoin jsr eget ;Test auf Komma cmp #' beq fill1 idx #01 ;CR-Error jmp error fill1 jsr gnum ;Fill-Byte </pre> | <pre> lda numh beq fill2 ldx #02 ;OV-Error jmp error fill2 ldy #00 lda numl sta (\$5c),y ;Fill inc \$5c bne fill3 inc \$5d fill3 lda \$5c ;akt. Adr. = Endadr.? cmp \$5e lda \$5d sbc \$5f bcc fill2 rts nop nop hunt jsr gadr ;Adressen holen dec epoin jsr eget cmp #' ;Test auf Komma beq hunt1 ldx #01 jmp error hunt1 jsr gnum ;Such-Byte lda numh beq hunt2 ldx #02 jmp error hunt2 ldy #00 lda numl cmp (\$5c),y bne hunt3 lda #0d ;gefunden Adressen jsr chout ;ausgeben lda \$5d jsr hout lda \$5c jsr hout1 inc \$5c bne hunt4 inc \$5d hunt4 lda \$5c ;akt. Adr. = Endadr.? cmp \$5e lda \$5d sbc \$5f bcc hunt2 rts nop nop go jsr gnum ;Adresse holen lda ps ;Register belegen pha plp ldx sp txs ldx xr ldy yr lda ac go1 pha lda numl ;Adresse nach PC sta pc ;kopieren lda numh sta pc+1 pla jmp (pc) ;ausfuehren exec pla pla jsr gnum ;Adresse holen lda ps ;Register belegen pha plp ldx sp txs ldx xr ldy yr lda ac jsr go1 ;ausfuehren sta ac ;Register retten sty yr stx xr tsx stx sp php pla sta ps jsr mess1 ;Register anzeigen jmp comm </pre> | <pre> break pia ;YR holen tay pia ;XR holen tax pia ;AC holen sta ac ;und abspeichern sty yr stx xr pia ;PS holen sta ps ;und speichern cli cli pia ;PC holen und sec ;speichern sbc #02 sta pc pia sbc #00 sta pc+1 tsx ;SP speichern stx sp jsr mess1 ;Register anzeigen jmp comm dos jsr eget ;Status? beq stat cmp #' ;Directory? beq dir cmp #' ;Geratenummer? beq dev ldx epoin ;Befehl senden ldy #02 dos1 jsr get bne dos1 stx #02 lda epoin sec sbc #02 jsr setnam lda #7f idx device ldy #0f jsr setpar jsr open jsr clall dev jsr gnum ;Geratenummer lda numh ;aendern bne dever lda numl and #11110110 bne dever lda numl sta device rts dever ldx #01 jmp error stat lda #0d ;Status auslesen jsr chout lda device sta ga jsr talk lda #6f sta sa jsr sectlk stat1 jsr iecin jsr chout cmp #0d bne stat1 jsr untlk rts dir lda #0d ;Directory anzeigen jsr chout ldx epoin ldy #02 dir0 jsr get bne dir0 stx #02 lda epoin sec sbc #02 jsr setnam lda #7f idx device ldy #60 jsr setpar jsr open lda ga jsr talk lda sa </pre> |
|---|---|--|


```

jsr sectik
lda #00
sta status
dir1 idy #03
sty $5e
jsr iecin
sta $5f
ldy status
bne dir4
jsr iecin
ldy status
bne dir4
ldy $5e
dey
bne dir1
ldx $5f
jsr %bddd
lda #20
dir3 jsr chout
jsr iecin
ldx status
bne dir4
tax
beq dir2
jsr chout
jmp dir3
dir2 lda #0d
jsr chout
ldy #02
bne dir1
dir4 lda #'i
sta $5e
lda #01
ldx #5e
ldy #00
jsr setnam
lda #7e
ldx device
ldy #0f
jsr setpar
jsr open
jsr clall
rts
getp jsr eget ;Dateinamen holen und
beq geter
cmp #' ' ;setzen
bne getp
ldy #02
ldx epoin
inx
stx #02
getp1 jsr eget
beq geter
cmp #' '
bne getp1
lda epoin
sec
sbc #02
jsr setnam
jsr eget ;Test auf Komma
cmp #' ,
beq getp2
geter ldx #01
jmp error
getp2 jsr gnum ;Geratenummer holen
lda numh ;und Fileparameter
beq getp4 ;setzen
getp3 ldx #02
jmp error
getp4 lda num1
cmp #20
bcs getp3
tax
ldy #01
lda #7f
jsr setpar
rts
load jsr getp ;Parameter setzen
lda #00 ;Load
jsr cload
jmp io
veri jsr getp ;Parameter setzen
lda #01 ;Verify
jsr cload
jmp io
save jsr getp ;Parameter setzen
dec epoin
jsr eget ;Test auf Komma
cmp #' ,
beq save1
ldx #01
jmp error
save1 lda #0d
jsr chout
jsr gadr ;Start- und Endadr.
lda #5c ;setzen
ldx $5e
ldy $5f
jsr csave ;Save
io bcs io1
lda status
and #00111111
bne io1
rts
io1 ldx #03
jmp error
mem jsr eget
beq mem.1
dec epoin
jsr gnum ;Startadresse holen
lda num1 ;und setzen
sta $5c
lda numh
sta $5d
mem.1 dec epoin
jsr eget
bne mem.2
lda $5c
clc
adc #01
sta $5e
lda $5d
adc #00
sta $5f
jmp mout1
mem.2 cmp #' : ;Eingabe?
beq memin ;Ausgabe?
cmp #' ,
beq mout
memer ldx #01
jmp error
mout dec epoin
jsr eget ;Test auf Komma
cmp #' ,
bne memer
jsr gnum ;Endadresse holen
lda num1 ;und setzen
sta $5e
lda numh
sta $5f
mout1 lda #0d ;Prompt und Adresse
jsr chout ;ausgeben
lda #' >
jsr chout
lda #' b
jsr chout
lda #' $
jsr chout
lda $5d
jsr hout1
lda $5c
jsr hout1
lda #' :
jsr chout
ldy #00 ;Byte holen
lda ($5c),y
jsr bout
inc $5c ;naechstes Byte
bne bnt2
inc $5d
bnt2 lda $5c ;Ende?
sec
sbc $5e
lda $5d
sbc $5f
bcc bnt1
rts
binin jsr gnum ;Byte holen
lda numh
beq bin11
ldx #02
jmp error
bin11 lda num1
ldy #00 ;und in Speicher
sta ($5c),y ;schreiben
inc $5c ;naechstes Byte
bne bin12
inc $5d
bin12 dec epoin
jsr eget
php
dec epoin
pip
beq min1
ldx #02
jmp error
min1 lda num1
ldy #00 ;und in Speicher
sta ($5c),y ;schreiben
inc $5c ;naechste Adresse
bne min2
inc $5d
min2 dec epoin
jsr eget
php
dec epoin
pip
bne memin ;Ende?
rts
bin jsr eget
beq bin.1
dec epoin
jsr gnum ;Startadresse holen
lda num1 ;und setzen
sta $5c
lda numh
sta $5d
bin.1 dec epoin
jsr eget
bne bin.2
lda $5c
clc
adc #01
sta $5e
lda $5d
adc #00
sta $5f
jmp bnt1
bin.2 cmp #' : ;Eingabe?
beq binin
cmp #' , ;Ausgabe?
beq bnout
ldx #01
jmp error
bnout jsr gnum ;Endadresse holen
lda num1 ;und setzen
sta $5e
lda numh
sta $5f
bnt1 lda #0d ;Prompt und Adresse
jsr chout ;ausgeben
lda #' >
jsr chout
lda #' b
jsr chout
lda #' $
jsr chout
lda $5d
jsr hout1
lda $5c
jsr hout1
lda #' :
jsr chout
ldy #00 ;Byte holen
lda ($5c),y
jsr bout
inc $5c ;naechstes Byte
bne bnt2
inc $5d
bnt2 lda $5c ;Ende?
sec
sbc $5e
lda $5d
sbc $5f
bcc bnt1
rts
binin jsr gnum ;Byte holen
lda numh
beq bin11
ldx #02
jmp error
bin11 lda num1
ldy #00 ;und in Speicher
sta ($5c),y ;schreiben
inc $5c ;naechstes Byte
bne bin12
inc $5d
bin12 dec epoin
jsr eget
php
dec epoin
pip

```

```

    bne binin      ;Ende?
chr   rts
      jsr eget
      beq chr.1
      dec epoin
      jsr gnum      ;Startadresse holen
      lda num1      ;und setzen
      sta %5c
      lda numh
      sta %5d
chr.1 dec epoin
      jsr eget
      bne chr.2
      lda %5c
      cbc
      adc #001
      sta %5e
      lda %5d
      adc #00
      sta %5f
      jmp chrol
chr.2 cmp #'      ;Eingabe?
      beq chrin
      cmp #'      ;Ausgabe?
      beq crout
      ldx #001
      jmp error
crout jsr gnum      ;Endadresse holen
      lda num1      ;und setzen
      sta %5e
      lda numh
      sta %5f
chrol lda #0d      ;Prompt und Adresse
      jsr chout      ;ausgeben
      lda #'>
      jsr chout
      lda #'c
      jsr chout
      lda #'$
      jsr chout
      lda %5d

      lda #'
      jsr chout
      ldy #00      ;einzelne Zeichen
chr2  lda (%5c),y ;holen
      sta %02
      and #01100000
      bne chro3
      lda #020
      sta %02
chr3  lda %02      ;und ausgeben
      jsr chout
      iny
      cpy #10
      bml chro2
      lda %5c      ;naechste Zeile
      cbc
      adc #10
      sta %5c
      lda %5d
      adc #00
      sta %5d
      lda %5c      ;Ende?
      sec
      sbc %5e
      lda %5d
      sbc %5f
      bcc chrol
      rts
chrin ldy #00      ;Zeichen holen
      jsr get
      beq chr12      ;Ende?
      sta %02
      cmp #020
      bne chr11
      lda (%5c),y
      and #01100000
      bne chr11
      lda (%5c),y
      sta %02
chr11 lda %02      ;und in Speicher
      sta (%5c),y ;schreiben
      inc %5c      ;naechstes Byte
      bne chrin
      inc %5d
      jmp chrin
chr12 rts

mnem1 .text "aaaaaaaaabbbbbbb";erste
      .text "bbccccccccddddd";Stelle
      .text "iiiijjlllllnopppp";der
      .text "rrrrrrrrrrrrrrrrrr";Mnemo-
      .text "ssstttttt?*" ;inics
mnem2 .text "dnrrrrrrrrrrrrrrrr";zweite
      .text "rvvlllllmppeeeoc";Stelle
      .text "nnnsaddddsorhhll";der
      .text "loorttaabeeelrst";Mnemo-
      .text "ttxyaasxy?*" ;inics
mnem3 .text "ccdrirtkacsqtie";dritte
      .text "kcccdvpxyccoxrs";Stelle
      .text "oxyprxaxyrrpaapap";der
      .text "alraishxcccioeha";Mnemo-
      .text "xyhhkxasa?*" ;inics
mnem4 .byte $10,$2b,$4c,$3c ;Code
      .byte $4b,$2b,$04,$3c ;fuer
      .byte $2d,$2b,$04,$01 ;Befehl
      .byte $4b,$2b,$04,$3c
      .byte $0f,$2b,$4c,$3c
      .byte $4b,$2b,$04,$3c
      .byte $13,$2b,$4b,$3c
      .byte $4b,$2b,$04,$3c
      .byte $24,$02,$4c,$30
      .byte $0c,$02,$31,$30
      .byte $2f,$02,$31,$01
      .byte $0c,$02,$31,$30
      .byte $0d,$02,$4c,$30
      .byte $4b,$02,$31,$30
      .byte $39,$02,$4b,$30
      .byte $4b,$02,$31,$30
      .byte $34,$1e,$4c,$3d
      .byte $4b,$1e,$29,$3d
      .byte $15,$1e,$4b,$3d
      .byte $4b,$1e,$29,$3d
      .byte $35,$00,$4c,$33
      .byte $4b,$00,$32,$33
      .byte $2e,$00,$32,$03
      .byte $23,$00,$32,$33
      .byte $12,$00,$4c,$33
      .byte $4b,$00,$32,$33
      .byte $3b,$00,$4b,$33
      .byte $4b,$00,$32,$33
      .byte $4b,$3f,$4b,$37
      .byte $41,$3f,$40,$37
      .byte $1d,$4b,$47,$08
      .byte $41,$3f,$40,$37
      .byte $09,$3f,$4c,$36
      .byte $41,$3f,$40,$37
      .byte $49,$3f,$48,$3e
      .byte $43,$3f,$42,$36
      .byte $28,$26,$27,$25
      .byte $28,$26,$27,$25
      .byte $45,$26,$44,$4a
      .byte $28,$26,$27,$25
      .byte $0a,$26,$4c,$25
      .byte $28,$26,$27,$25
      .byte $18,$26,$46,$06
      .byte $26,$26,$27,$25
      .byte $19,$17,$4b,$1a
      .byte $19,$17,$1b,$1a
      .byte $22,$17,$1c,$07
      .byte $19,$17,$1b,$1a
      .byte $0e,$17,$4c,$1a
      .byte $4b,$17,$1b,$1a
      .byte $14,$17,$4b,$1a
      .byte $4b,$17,$1b,$1a
      .byte $18,$38,$4b,$1f
      .byte $18,$38,$20,$1f
      .byte $21,$38,$2a,$38
      .byte $18,$38,$20,$1f
      .byte $0b,$38,$4c,$1f
      .byte $4b,$38,$20,$1f
      .byte $3a,$38,$4b,$1f
      .byte $4b,$38,$20,$1f
lang  .byte $01,$02,$02,$02 ;Anzahl
      .byte $02,$02,$02,$02 ;der Bytes
      .byte $03,$03,$03,$03
adr   sta opc      ;Adressier-
      cmp #06c      ;lungsart
      bne a0
      lda #00c      ;aus Opcode
      rts            ;ermitteln
a0    cmp #089
      bne a1
      lda #003

a1    rts
      and #01f
      tax
      lda opc
      cmp #00
      bpl a2
      cmp #0c
      bpl a2
      lda adr1,x
      jmp a3
a2    lda adr0,x
a3    cpx #00
      beq a5
      cpx #02
      beq a6
      rts
a4    lda opc
      lsr a
      lsr a
      lsr a
      lsr a
      tax
      rts
a5    jsr a4
      lda adr2,x
      rts
a6    jsr a4
      lda adr3,x
      rts
adr0  .byte $00,$07,$00,$07 ;Tabelle
      .byte $03,$03,$03,$03 ;fuer
      .byte $01,$02,$01,$02 ;Adressier-
      .byte $09,$09,$09,$09 ;lungsarten
      .byte $06,$08,$01,$08
      .byte $03,$04,$04,$04
      .byte $01,$0b,$01,$0b
      .byte $09,$0a,$0a,$0a
adr1  .byte $00,$07,$00,$07 ;Sonder-
      .byte $03,$03,$03,$03
      .byte $01,$02,$01,$02 ;fall 1
      .byte $09,$09,$09,$09
      .byte $06,$08,$01,$08
      .byte $04,$04,$05,$05
      .byte $01,$0b,$01,$0b
      .byte $0a,$0a,$0b,$0b
adr2  .byte $01,$09,$01,$01 ;Sonder-
      .byte $03,$02,$02,$02 ;fall 2
adr3  .byte $01,$01,$01,$01 ;Sonder-
      .byte $03,$02,$03,$03 ;fall 3
dass  jsr eget
      beq das.1
      dec epoin
      jsr gnum      ;Startadresse holen
      lda num1      ;und setzen
      sta %5c
      lda numh
      sta %5d
das.1 dec epoin
      jsr eget
      bne das.2
      lda #23
      sta %ff
das.. jsr diss1
      dec %ff
      bne das..
      rts
das.2 cmp #'      ;Eingabe?
      bne dass1
      jmp ass
dass1 cmp #'      ;Ausgabe?
      beq diss
      ldx #001
      jmp error
diss  jsr gnum      ;Endadresse holen
      lda num1      ;und setzen
      sta %5e
      lda numh
      sta %5f
dissa jsr diss1      ;1 Zeile drucken
      lda %5c      ;Ende?
      sec
      sbc %5e
      lda %5d
      sbc %5f
      bcc dissa
      rts
diss1 lda #0d      ;Prompt und Adresse
      jsr chout      ;ausgeben
      lda #'>

```

| | | |
|--|---|---|
| <pre> jsr chout lda #'d jsr chout lda #'# jsr chout lda \$5d jsr hout1 lda \$5c jsr hout1 lda #' jsr chout ldy #00 ;opcode holen lda (\$5c),y jsr hout ;und drucken lda (\$5c),y jsr adr ;Adressierungsart tax ;holen dex lda lang,x ;Befehlslaenge holen tax cpx #01 beq diss2 iny lda (\$5c),y jsr hout cpx #02 beq diss2 iny lda (\$5c),y jsr hout diss2 lda #20 ;auf Spalte 21 posi- jsr chout ;tionieren lda \$d3 cmp #15 bmi diss2 lda #' jsr chout ldy #00 ;Mnemonic drucken lda (\$5c),y tax lda mne4,x tax lda mne1,x jsr chout lda mne2,x jsr chout lda mne3,x jsr chout lda #20 jsr chout lda opc jsr adr tax cpx #01 bne dis1 jmp dise dis1 cpx #02 ;Operanden bestimmen bne diss3 ;und ausgeben lda #' ;immediate? jsr chout diss3 cpx #07 ;indiziert indirekt? beq diss4 cpx #08 ;indirekt indiziert? beq diss4 cpx #0c ;absolut indirekt? bne diss5 diss4 lda #' jsr chout diss5 lda #' jsr chout cpx #09 ;absolut? bmi dis2 jmp dis12 dis2 cpx #06 ;relativ? beq diss9 ldy #01 ;impliziert? lda (\$5c),y jsr hout1 cpx #04 bpl dis3 jmp dise dis3 cpx #04 ;0-page x-indiziert? bne diss6 lda #' jsr chout lda #'x jsr chout jmp dise diss6 cpx #05 ;0-page y-indiziert? bne diss7 </pre> | <pre> lda #' jsr chout lda #'y jsr chout jmp dise diss7 cpx #07 ;indiziert indirekt? bne diss8 lda #' jsr chout lda #'x jsr chout lda #' jsr chout jmp dise diss8 lda #' ;indirekt indiziert jsr chout lda #' jsr chout lda #'y jsr chout jmp dise diss9 lda \$5c ;Operand fuer ;relative Adressier- ;ung bestimmen clc adc #02 sta num1 lda \$5d adc #00 sta numh ldy #01 lda (\$5c),y bpl dis10 lda #ff sta \$02 dis10 lda (\$5c),y bmi dis11 lda #00 sta \$02 dis11 lda (\$5c),y clc adc num1 sta num1 lda \$02 adc numh sta numh jsr hout1 lda num1 jsr hout1 jmp dise dis12 ldy #02 ;16-Bit Adresse aus- lda (\$5c),y ;geben jsr hout1 ldy #01 lda (\$5c),y jsr hout1 cpx #09 ;absolut? beq dise cpx #0a ;ix-indiziert? bne dis14 lda #' jsr chout lda #'x jsr chout jmp dise dis14 cpx #0b ;iy-indiziert? bne dis15 lda #' jsr chout lda #'y jsr chout jmp dise dis15 lda #' ;absolut indirekt? jsr chout dise ldy #00 ;naechste Zeile lda (\$5c),y jsr adr ;Adressierungsart tax dex lda lang,x ;Befehlslaenge clc adc \$5c sta \$5c lda \$5d adc #00 sta \$5d disel lda #20 ;Restzeile loeschen jsr chout lda \$d3 cmp #27 bmi dise1 rts </pre> | <pre> ass lda #00 ;nichtexistenten Be- sta mne4 ;fehl voraussetzen jsr eget ;nach Mnemonic-Anfang beq ass1 ;suchen cmp #' bne ass beq ass2 ass1 ldx #01 ;CR-Error jmp error ass2 jsr get ;Mnemonic in mne1- bne ass3 ;mne3 ablegen jmp assel ass3 sta mne1 jsr get bne ass4 jmp assel ass4 sta mne2 jsr get bne ass5 jmp assel ass5 sta mne3 jsr eget ;Adressierungsart und bne ass6 ;operand holen ldx #01 ;impliziert stx mne4 jmp asse ass6 cmp #' ;immediate bne ass7 ldx #02 stx mne4 jmp ass9 ass7 cmp #' ;indirekt bne ass9 ldx mne1 ;absolut indirekt? cpx #'j bne ass8 ldx mne2 cpx #'m bne ass8 ldx #0c ;absolut indirekt stx mne4 jmp ass9 ass8 ldx #80 ;indirekt-flag setzen stx mne4 ass9 lda mne4 ;Adresse holen bne ass10 dec epoin ass10 jsr gnum dec epoin jsr eget bne ass11 ldx mne4 bne ass11 ldx #09 ;absolut stx mne4 ass11 cmp #' bne ass12 ldx mne4 cpx #0c beq asse bit mne4 bpl asse jsr eget cmp #' bne asse jsr eget cmp #'y bne asse lda #08 ;indirekt indiziert sta mne4 jmp asse ass12 cmp #' bne asse jsr eget cmp #'y bne ass13 lda #0b ;iy-indiziert sta mne4 jmp asse ass13 cmp #'x bne asse jsr eget bne ass14 lda #0a ;ix-indiziert sta mne4 jmp asse ass14 cmp #' bne asse bit mne4 bpl asse </pre> |
|--|---|---|

Super-Tool

| | | |
|-------------------------------------|-----------------------------------|---------------------------------|
| lda #07 ;indiziert indirekt | cmp #03 ;moeglich mit abso- | sub #5c |
| asse sta mne4 | bmi ass22 ;lut versuchen | lda #5f |
| lda mne4 | cmp #06 | sub #5d |
| and #7f | bpl ass22 | bcs down |
| sta mne4 | cic | up0 ldy #00 ;Verschieben |
| lda mne4 | adc #06 | lda (\$c),y |
| cmp #09 ;absolut? | sta mne4 | sta (\$e),y |
| bne ass16 | jap ass19 | inc #5c |
| lda mne1 | ass22 lda #00 | bne up1 |
| cmp #'b ;Branch-Befehl? | sta mne4 | inc #5d ;inaechstes Byte |
| bne ass16 | ass23 stx opc ;Opcode sichern | up1 inc #5e |
| lda mne2 | lda mne4 ;Adressierungsart | bne up2 |
| cmp #'i | beq assel | up2 inc #5f |
| beq ass16 | tax | lda #5c |
| lda #06 ;relativ | dex | sec |
| sta mne4 | lda lang,x ;Befehlslaenge holen | sub end1 ;schon fertig ? |
| nop | tax | lda #5d |
| nop | lda opc ;Opcode holen | sub endh |
| nop | ldy #00 | bcc up0 |
| nop | sta (\$c),y ;und speichern | rts |
| nop | cpx #01 | down lda end1 ;Adressen fuer |
| lda #5c | beq assel ;impliziert? | cic |
| cic | lda numl ;low-Byte holen | sub #5c |
| adc #02 | iny | sta #02 ;Verschiebung um- |
| sta #5e | sta (\$c),y ;und speichern | lda endh |
| lda #5d | cpx #02 ;2-Byte-Befehl? | sub #5d ;rechnen |
| adc #00 | beq assel | pha |
| sta #5f | lda numh ;high-Byte holen | cic |
| lda numl | iny | lda #02 |
| sec | sta (\$c),y ;und speichern | adc #5e |
| sub #5e | assel dec #d6 ;1 Zeile hoch | sta #5e |
| sta numl | jsr diss1 ;Befehl ausgeben | pla |
| tax | lda #0d ;neue Zeile (cr) | adc #5f |
| lda numh | jsr chout | sta #5f |
| sub #5f | lda #'> ;Prompt | lda end1 |
| cpx #80 | jsr chout | ldx #5c |
| adc #00 | lda #'d ;Befehl | sta #5c |
| nop | jsr chout | stx end1 |
| nop | lda #'\$;Zahlsystem | lda endh |
| beq ass15 | jsr chout | ldx #5d |
| lda #00 | lda #5d ;high-Byte der aktu- | sta #5d |
| sta mne4 | jsr hout1 ;ellen Adresse | stx endh |
| ass15 lda #00 | lda #5c ;low-Byte der aktu- | lda #5c |
| sta numh | jsr hout1 ;ellen Adresse | sec |
| ass16 lda numh ;Adresse < #0100? | lda #' ;Eingabe-Zeichen | sub #01 |
| bne ass17 | jsr chout | sta #5c |
| lda mne4 | lda #' ;Mnemonic-Zeichen | lda #5d |
| cmp #09 ;absolut oder | jsr chout | sub #00 |
| bmi ass17 ;absolut indiziert? | ass24 lda #20 ;Zeile bis Position | sta #5d |
| cmp #0c | jsr chout ;39 mit Leerzeichen | dow0 ldy #00 ;Verschieben |
| bpl ass17 | lda #d3 ;fuehlen | lda (\$c),y |
| sec | cmp #27 | sta (\$e),y |
| sub #06 ;0-page oder 0-page | bmi ass24 | lda #5c |
| sta mne4 ;indiziert | dec #d6 ;1 Zeile hoch | sec |
| ass17 lda mne1 | ldx #07 ;Tastaturpuffer mit | sub #01 ;inaechstes |
| cmp #'* ;Mehr-Byte NOP? | lda #1d ;8 * Cursor-Rechts | sta #5c |
| beq ass18 | ass25 sta tbuff,x ;fuehlen | bcs dow1 |
| cmp #'t ;"Aufhaenger"? | dex | dec #5d |
| bne ass19 | bpl ass25 | dow1 lda #5e ;Byte |
| ass18 lda #00 ;nicht beruecksichti- | lda #08 ;8 Tasten setzen | sec |
| sta mne4 ;gen | sta #c6 | sub #01 |
| ass19 ldx #4d ;Opcode suchen | rts ;Ende | sta #5e |
| ass20 dex | tran jsr gnum ;Startadresse alt | bcs dow2 |
| bmi ass22 ;schon alle probiert? | lda numl | dec #5f |
| lda mne1 | sta #5c | dow2 lda #5c ;schon fertig ? |
| cmp mne1,x ;1. Stelle richtig? | lda numh | sec |
| bne ass20 | sta #5d | sub end1 |
| lda mne2 | dec epoin | lda #5d |
| cmp mne2,x ;2. Stelle richtig? | jsr eget | sub endh |
| bne ass20 | cmp #' ;Komma ? | bcs dow0 |
| lda mne3 | beq tra1 | rts |
| cmp mne3,x ;3. Stelle richtig? | tra0 ldx #01 ;CR-Error | ac #++ +1 ;Akku-Speicher |
| bne ass20 | jmp error | yr #++ +1 ;Y-Speicher |
| txa | tra1 jsr gnum ;Endadresse alt | xr #++ +1 ;X-Speicher |
| sta opt | lda numl | sp #++ +1 ;S-Speicher |
| ldy #00 ;Opcode fuer | sta end1 | ps #++ +1 ;P-Speicher |
| beq as1 | lda numh | pc #++ +2 ;PC-Speicher (lo/hi) |
| as0 iny ;Befehl suchen | sta endh | epoin #++ +1 ;Eingabepointer |
| beq ass21 | dec epoin | numl #++ +1 ;Argument low |
| as1 lda opt | jsr eget | numh #++ +1 ;Argument high |
| cmp mne4,y ;Adressierungsart | cmp #' ;Komma ? | device #++ +1 ;Floppynummer |
| bne as0 | bne tra0 | opc #++ +1 ;Opcode |
| tya ;holen | jsr gnum ;Startadresse neu | mne1 #++ +1 ;1. Stelle Mnemonic |
| jsr adr | lda numl | mne2 #++ +1 ;2. Stelle Mnemonic |
| ldx opc | sta #5e | mne3 #++ +1 ;3. Stelle Mnemonic |
| cmp mne4 ;Adressierungsart | lda numh | mne4 #++ +1 ;Adressierungsart |
| beq ass23 ;richtig? | sta #5f | opt #++ +1 ;Opcode-Test |
| bne as0 | lda #5e | end1 #++ +1 ;Adresse fuer |
| ass21 lda mne4 ;falls 0-page nicht | sec | endh #++ +1 ;Transfer-Befehl |

Zuverlässigkeit macht Schule

SYS SYSTEMS 85
München
28.10. - 1.11.
Halle 20 - Stand D3

No. _____
maxell MINI-FLOPPY DISK
MINI-DISKETTE
DISQUE MINI-DISKETTE
MD1-D JAPAN JAPON



Schul-Arbeiten zuverlässig vorbereiten und erledigen: Stunden- und Arbeitspläne auf einen Blick, Unterrichtsinhalte mit direktem Zugriff. Da müssen auch die Datenträger Klasse sein.

Maxell-Datenträger bekommen die besten Zeugnisse - in jedem Fach. Und in Zuverlässigkeit immer eine Eins. Ihre Mitarbeit im Unterricht ist vorbildlich. Maxell-Datenträger - die Zuverlässigen.

maxell[®]
Datenträger
die Zuverlässigen

(PLZ 4 und 5) **KOMP**, Heinrich-Spahn-Straße 12-14, 4019 Monheim, Tel.: 02173/52071/2
(PLZ 6) **ART 2000**, Altstraße 2, 6450 Hanau, Tel.: 06181/24786-7 - (PLZ 7 und 8)
SYNELEC Datensysteme GmbH, Postfach 151727, 8000 München 2, Tel.: 089/5179733.

Spitzen-Grafik ohne Mühe

Der Commodore 64 ist ein Widerspruch in sich: er hat ein schlechtes Basic und ist doch der meistverkaufte Heimcomputer. Er unterstützt weder Grafik noch Musik und läßt doch Konzerte erklingen und Gemälde entstehen. Aber dieses Paradoxon ist auch seine Stärke. Da er (fast) alles kann, aber wenig Mittel zur Verwirklichung seiner Fähigkeiten anbietet, sind der Phantasie und dem Geschick der Benutzer keine Grenzen gesetzt. RUN-Programmierer Frank Riemenschneider hat sich diese Tatsache zunutze gemacht und eine Erweiterung geschrieben, die das magere Basic um 50 neue Grafikbefehle bereichert. Diese 50 Befehle lassen sich in sechs Gruppen einteilen:

- 1) Befehle zur Programmierung der hochauflösenden Grafik (2 Grafikseiten mit je 320x200 Grafikpunkten)
- 2) Befehle zur Spriteprogrammierung
- 3) Befehle zur Diskettenprogrammierung
- 4) Befehle zur Erstellung eines eigenen Zeichensatzes
- 5) Befehle zur Programmierung der Blockgrafik (40x25 Zeichen)
- 6) Allgemeine Programmierhilfen.

Den bei weitem größten Anteil hat die Programmierung der hochauflösenden Grafik mit 21 Befehlen. Es stehen zwei Grafikseiten zur Verfügung. Das bedeutet, daß man gleichzeitig zwei unabhängige Grafiken darstellen kann. Es ist aber auch möglich, beide Seiten zu verknüpfen, so daß sich eine Grafik über zwei Seiten erstrecken kann. Damit erhält man eine Auflösung von 320x400 beziehungsweise von 640x200 Punkten, was den Möglichkeiten von erheblich größeren Computern entspricht.

Hier die Befehle zur Programmierung der hochauflösenden Grafik:

- 1) Einschalten der Grafik/Wahl des Grafikmodus

Spielefreaks wissen es schon lange: Der C64 hat exzellente Grafikmöglichkeiten. Doch man braucht solide Programmierkenntnisse, um sie zu nutzen.

Mit den 50 neuen Grafikbefehlen kommt man der Perfektion einen Riesenschritt näher.

Befehl: GMODE AS,BS

AS: Angezeigte Grafikseite (1 oder 2)

BS: Bearbeitete Grafikseite (1 oder 2)

Durch einen unterschiedlichen Wert von AS und BS ist es möglich, „unsichtbar“ zu zeichnen.

2) Ausschalten der Grafik

Befehl: GMODE 0

Es wird auf den Textschirm umgeschaltet und dieser dabei gelöscht.

3) Grafikfarbe setzen

Befehl: GCOL Px16+H

P: Punktfarbe

H: Hintergrundfarbe

Man kann für die beiden Grafikseiten eine unterschiedliche Farbkombination festlegen.

4) Grafikfarbe ändern

Befehl: PCOL Px16+H

P: Punktfarbe

H: Hintergrundfarbe

Hiermit kann man während des Zeichnens die Farbkombination wechseln.

5) Grafikseite löschen

Befehl: GCLEAR

Die mit GMODE angesprochene Grafikseite wird gelöscht.

6) Grafikpunkt setzen

Befehl: SPLOT X,Y

X: X-Koordinate (0—319)

Y: Y-Koordinate (0—199)

7) Grafikpunkt löschen

Befehl: CPLOT X,Y

X- und Y-Werte siehe unter 6).

8) Linie ziehen

Befehl: SLINE X1,Y1,X2,Y2

X1/2: X-Koordinate des Start-/Endpunktes (0—319)

Y1/2: Y-Koordinate des Start-/Endpunktes (0—199)

9) Linie löschen

Befehl: CLINE X1,Y1,X2,Y2

X1,Y1,X2,Y2 siehe unter 8).

10) Ausgefülltes Rechteck zeichnen

Befehl: SBLOC X1,Y1,X2,Y2

X1/2: X-Koordinate der linken, oberen/rechten, unteren Ecke des Rechtecks (0—319)

Y1/2: Y-Koordinate der linken, oberen/rechten, unteren Ecke des Rechtecks (0—199)

Auf jeden Fall muß Y2 größer als Y1 sein, da sonst eine Fehlermeldung erzeugt wird.

11) Ausgefülltes Rechteck löschen

Befehl: CBLOC X1,Y1,X2,Y2

X1,Y1,X2 und Y2 siehe unter 10).

12) Rahmen zeichnen

Befehl: SFRAME X1,Y1,X2,Y2

X1,Y1, X2 und Y2 siehe unter 10).

13) Rahmen löschen

Befehl: CFRAME X1,Y1,X2,Y2

X1,Y1, X2 und Y2 siehe unter 10).

14) Kreis oder Ellipse zeichnen

Befehl: SCIR XM,YM,XR,YR, AW,EW

XM: X-Koordinate des Mittelpunktes (0—319)

YM: Y-Koordinate des Mittelpunktes (0—199)

XR: Radius in X-Richtung (0—319)

YR: Radius in Y-Richtung (0—199)

AW: Anfangswinkel (0—2*Pi)

EW: Endwinkel (0—2*Pi)

Der Winkel „Null“ liegt dem kleinen Zeiger einer Uhr entsprechend, die drei Uhr anzeigt.

Pi/2 liegt bei sechs Uhr, Pi bei neun Uhr und 3*Pi/2 bei zwölf Uhr. Ein Kreis wird gezeichnet, wenn XR und YR gleich groß sind.

15) Kreis oder Ellipse löschen
 Befehl: CCIR XM,YM,XR,YR,AW,EW
 XM,YM,XR,YR,AW und EW siehe unter 14)

16) Text auf Grafikschiem schreiben

Befehl: STTEXT M,X,Y,"Text"
 M: Zeichenmodus (0=Großschriftmodus, 1=Kleinschriftmodus)
 X: X-Koordinate des Textanfangs (0-319)
 Y: Y-Koordinate des Textanfangs (0-199)

"Text": Ausdruck, der geschrieben werden soll. Dabei kann es sich auch um eine Stringvariable handeln.

Dieser Befehl ignoriert alle Steuerzeichen bis auf Revers on und Revers off. Wird das Zeilenende erreicht, so wird in der nächsten Zeile weitergeschrieben.

17) Text vom Grafikschiem löschen

Befehl: CTEXT M,X,Y,"Text"
 M,X,Y und "Text" siehe unter 16).

18) Grafikschiem invertieren

Befehl: GINV
 G: Geräteadresse des Ausgabe-gerätes (1=Kassette, 8=Floppy)

"Filename": Name, unter dem der Grafikbildschirm abgespeichert werden soll. Dabei kann es sich auch um eine Stringvariable handeln.

20) Grafikschiem laden

Befehl: GLOAD "Filename",G
 G und "Filename" siehe unter 19).

21) Grafik ausdrucken

Befehl: GHARD
 Drückt eine Hardcopy auf einem 7-Nadeldrucker (z.B. MPS 801). Funktioniert aber auch auf Epson mit Interface etc.

Die Spriteprogrammierung wird vom normalen Basic leider überhaupt nicht unterstützt. GRAFIK-BASIC vereinfacht den Einsatz von Sprites mit fünf neuen Befehlen.

1) Definition eines Sprites

Befehl: SPRDEF N,B,F,P,A,X,Y
 N: Nummer des Sprites (0-7)
 B: Block, aus dem der Sprite seine Informationen holt (0-255)
 F: Farbe des Sprites (0-15)
 P: Priorität (0=Sprite vor, 1=Sprite hinter Bildschirmzeichen)

A: Art des Sprites (0=Einfarbensprite 24x21 Punkte Auflösung, 1=Multicolorsprite 12x21 Punkte Auflösung)

X: Ausdehnung in X-Richtung (0=Normalgröße, 1=doppelte Größe)

Y: Ausdehnung in Y-Richtung (0=Normalgröße, 1=doppelte Größe)

2) Einschalten eines Sprites

Befehl: SPRIN N
 N: Spritenummer (0-7)

3) Sprite auf Bildschirm setzen

Befehl: SPRSET N,X,Y
 N: Spritenummer (0-7)
 X: X-Koordinate des Sprites (0-350)
 Y: Y-Koordinate des Sprites (0-250)

4) Sprite ausschalten

Befehl: SPROFF N
 N: Spritenummer (0-7)

5) Farbe für Multicolorsprite setzen

Befehl: SRPCOL F1,F2
 F1: erste Farbe (0-15)
 F2: zweite Farbe (0-15)

Ein Multicolorsprite kann drei Farben annehmen: Zwei sind für alle Sprites gleich und werden mit SPRCOL festgelegt. Darüber hinaus hat jedes Sprite eine individuelle dritte Farbe. Diese wird mit SPRDEF festgelegt.

Für die Daten, die das Aussehen der Sprites bestimmen, bieten sich die folgenden Blöcke an; ohne eigenen Zeichensatz gibt es keine Probleme, da die Blöcke 32-63 (\$0800 bis \$0FFF) benutzt werden können. Bei einem neu definierten Zeichensatz wird es jedoch eng: Vor dem Basic-Start können nur die Blöcke 11 und 15 genutzt werden. Sollen mehr Sprites dargestellt werden, greift man am besten auf die höheren Blöcke wie 250,251 usw. zurück. In diesem Fall ist aber die Länge des Basic-Programms auf ca. 12 K begrenzt, da die Spritedefinitionen sonst überschrieben würden.

Für die Diskettenprogrammierung stellt GRAFIK-BASIC drei neue Befehle zur Verfügung.

1) Auslesen und Anzeigen des Fehlerkanals

Befehl: ERROR
 Tritt keine Fehlermeldung auf, wird "0,OK,0,0" angezeigt.

2) Befehl an Floppy senden
 Befehl: DISK "Anweisung"
 DISK "N:Floppy,01" zum Beispiel formatiert eine Diskette.

3) Directory einlesen und anzeigen

Befehl: DIR
 Ein eventuell im Speicher stehendes BASIC-Programm wird nicht gelöscht.

Der dritte Befehlsbereich befaßt sich mit der Programmierung eines individuellen Zeichensatzes. Eine Stärke von Grafik-Basic liegt darin, daß mit dem TEXT-Befehl (siehe oben) außer dem normalen Zeichensatz auch selbstdefinierte Zeichen auf dem hochauflösenden Grafikschiem dargestellt werden können. Weiterhin ist ein deutscher Zeichensatz in GRAFIK-BASIC bereits integriert, so daß dieser mit einem Befehl sofort zur Verfügung steht. Hier nun die sechs neuen Befehle:

1) Zeichensatz ins RAM kopieren

Befehl: ZCOPY
 Durch diesen Befehl wird der Original-Commodore Zeichensatz ins RAM von \$800 bis \$FFF kopiert. Dies ist notwendig, um Änderungen am Zeichensatz durchzuführen.

2) Deutschen Zeichensatz ins RAM kopieren

Befehl: ZGER
 Durch diesen Befehl wird nicht der Original-Commodore Zeichensatz, sondern der in GRAFIK-BASIC integrierte deutsche Zeichensatz ins RAM gelegt. Tabelle

| Taste | neue Belegung |
|-----------------------|---------------|
| Shift und 7 | / |
| + | ß |
| Shift und * | + |
| Klammeraffe | ü |
| Shift und Klammeraffe | Û |
| : | ö |
| Shift und : | Ö |
| ; | ä |
| Shift und ; | Ä |
| / | - |
| Shift und , | : |
| Shift und . | : |
| z | y |
| Z | Y |
| y | z |
| Y | Z |

Tabelle 1: Die Tastaturbelegung unter dem deutschen Zeichensatz

1 führt die neubelegten Tasten auf.

3) Zeichensatz abspeichern
Befehl: ZSAVE "Filename", G
G: Geräteadresse des Ausgabegerätes (1 = Kassette, 8 = Floppy)
"Filename": Name, unter dem der Zeichensatz abgespeichert werden soll.

Mit diesem Befehl kann man einen neu definierten Zeichensatz abspeichern.

4) Zeichensatz laden
Befehl: ZLOAD "Filename", G
G und "Filename" siehe unter 3). Hiermit wird ein abgespeicherter Zeichensatz wieder eingeladen. Es ist auch möglich, Zeichensätze aus anderen Programmen zu verwenden, da sie automatisch an die richtige Stelle geladen werden.

5) Zeichensatz auf RAM schalten
Befehl: ZIN

Hiermit wird der von \$800 bis \$FFF liegende Zeichensatz aktiviert. Bevor dieser Befehl verwendet werden kann, muß entweder mit ZCOPY oder ZGER ein Zeichensatz in diesen Bereich kopiert oder ein Zeichensatz mit ZLOAD geladen werden.

6) Zeichensatz auf ROM schalten
Befehl: ZOFF

Hierdurch wird wieder der Original-Zeichensatz im ROM aktiviert.

Auch die Programmierung der Low-Resolution Grafik wird von GRAFIK-BASIC unterstützt. Acht Befehle erleichtern das Erstellen von Blockgrafiken.

1) Bildschirm invertieren
Befehl: LINV

2) Einen bestimmten Bildschirmbereich mit einem Zeichen füllen
Befehl: LFILL Z, X1, Y1, X2, Y2
Z: Bildschirmcode des Zeichens (0-255)

X1/2: Zeile der linken, oberen/rechten, unteren Ecke des Bereichs (0-24)

Y1/2: Spalte der linken, oberen/rechten unteren Ecke des Bereichs (0-39).

Achtung: Der Bildschirmcode ist nicht mit dem ASCII-Code identisch. Man findet die Codes der einzelnen Zeichen auf den Seiten 132-134 im Handbuch.

3) Einen bestimmten Bildschirmbereich mit einer Farbe füllen
Befehl: LCOL F, X1, Y1, X2, Y2

F: Farbcode (0-15)

X1, Y1, X2 und Y2 siehe unter 2).

4) Bildschirm abspeichern
Befehl: LSAVE "Filename", G

G: Geräteadresse des Ausgabegerätes (1 = Kassette, 8 = Floppy)

"Filename": Name, unter dem der Bildschirm abgespeichert werden soll.

5) Bildschirm laden
Befehl: LLOAD "Filename", G
G und "Filename" siehe unter 4).

6) Positionierte Ausgabe
Befehl: LPRI Z, S, "Ausdruck"

Z: Bildschirmzeile (0-24)

S: Bildschirmspalte (0-39)

"Ausdruck": Auszugebender Ausdruck wie beim PRINT-Befehl.

7) Ausdruck des Bildschirms
Befehl: LHard

Druckt auf beliebigem Drucker eine Bildschirmhardcopy.

8) Scrollen des Bildschirms
Befehl: LSCROL R, A, E

R: Richtung des Scrollens (0=links, 1=rechts)

A: Erste Zeile, die gescrollt wird (0-24)

E: Letzte Zeile, die gescrollt wird (0-24)

Die letzte Befehlsgruppe umfaßt sechs allgemeine Programmierhilfen:

1) Zurückholen eines gelöschten Programms

Befehl: RENEW

RENEW hilft sowohl bei Hardware- oder Software-Reset als auch beim Befehl „NEW“.

2) Programm neu durchnummerieren

Befehl: RENUM AZ, SW

AZ: Anfangszeilennummer

SW: Schrittweite der Zeilennummern.

Zeilenreferenzen (GOTO, GO-SUB) werden nicht geändert und müssen per Hand korrigiert werden.

3) Ausgabe aller Variablen
Befehl: DUMP

Funktionen werden mit einer spitzen Klammer gekennzeichnet.

4) Auslesen des „versteckten“ RAMs

Befehl: RAM S

S: auszulesende Speicherstelle

Es ist von Basic aus nicht möglich, auf bestimmte Teile des RAM zuzugreifen. Dabei handelt es sich um die Bereiche unter dem Betriebssystem-ROM (\$A000-\$BFFF und \$E000-\$FFFF) sowie um die 4 K unter dem Charakter-ROM (\$D000-\$DFFF). Der normale PEEK-Befehl gibt immer den Inhalt des ROMs aus. Der Befehl RAM überträgt den Wert aus dem RAM in die Speicherstelle 250, wo er mit PRINT PEEK (250) leicht

| | | | | | |
|--------|-------|-----------|--------|-----|-----------|
| CBLOCK | C | (SHIFT) B | LSAVE | L | (SHIFT) S |
| CCIR | C | (SHIFT) C | LSCROL | LS | (SHIFT) C |
| CFRAME | C | (SHIFT) F | MERGE | M | (SHIFT) E |
| CLINE | CL | (SHIFT) I | PCOL | P | (SHIFT) C |
| CPLOT | C | (SHIFT) P | RAM | R | (SHIFT) A |
| CTEXT | C | (SHIFT) T | RENEW | RE | (SHIFT) N |
| DISK | DI | (SHIFT) S | RENUM | REN | (SHIFT) U |
| DIR | keine | Abkürzung | SBLOCK | S | (SHIFT) B |
| DUMP | D | (SHIFT) U | SCIR | S | (SHIFT) C |
| ERROR | E | (SHIFT) R | SFRAME | S | (SHIFT) F |
| GCLEAR | G | (SHIFT) C | SLINE | S | (SHIFT) L |
| GCOL | GC | (SHIFT) O | SPLIT | SP | (SHIFT) L |
| GHARD | G | (SHIFT) H | SPRCOL | SPR | (SHIFT) C |
| GINV | G | (SHIFT) I | SPRDEF | SP | (SHIFT) R |
| GLOAD | G | (SHIFT) L | SPRIN | SPR | (SHIFT) I |
| GMODE | G | (SHIFT) M | SPROFF | SPR | (SHIFT) O |
| GSAVE | G | (SHIFT) S | SPRSET | SPR | (SHIFT) S |
| HEXA | H | (SHIFT) E | STEXT | STE | (SHIFT) X |
| LCOLOR | L | (SHIFT) C | ZCOPY | Z | (SHIFT) C |
| LFILL | L | (SHIFT) F | ZGER | Z | (SHIFT) G |
| LHARD | L | (SHIFT) H | ZIN | Z | (SHIFT) I |
| LINV | LI | (SHIFT) N | ZLOAD | Z | (SHIFT) L |
| LLOAD | L | (SHIFT) L | ZOFF | Z | (SHIFT) O |
| LPRI | L | (SHIFT) P | ZSAVE | Z | (SHIFT) S |

Tabelle 2: Alphabetische Auflistung der Befehlsabkürzung

auslesbar ist.

5) Basic-Programme aneinanderhängen

Befehl: MERGE "Filename",G

G: Geräteadresse des Eingabegerätes (1 = Kassette, 8 = Floppy)

"Filename": Name des nachzuladenden Programms.

Es können beliebig viele Programme angehängt werden. Die Zeilennummern des nachzuladenden Programms müssen größer sein als die des Programms im Speicher. Im Zweifelsfall sollte vorher mit RENUM neu durchnummeriert werden.

6) Hexadezimale Ein- und Ausgaben erlauben

Befehl: HEXA

Nach Eingabe dieses Befehls kann man sämtliche Ein- und Ausgaben in hexadezimaler Form vornehmen. So kann man zum Beispiel statt POKE 53248,120 schreiben: POKE \$D000,120.

Alle Befehle können auch abgekürzt eingegeben werden (Tabelle 2). Da die neuen Befehle voll ins Betriebssystem eingebunden sind, erscheint beim Listen wie gewohnt das ausgeschriebene Befehlswort.

Die Erweiterung nimmt „nur“ 7 K des Basic-Speichers in Anspruch, so daß für Programme noch ca. 32 K zur Verfügung stehen. Die Spei-

cherbelegung von GRAFIK-BASIC sieht so aus: ein Teil liegt von \$9000 bis \$9FFF, der zweite von \$C400 bis \$CFFF. Damit sind 4096 Basic-Bytes abgezweigt worden. Die beiden Grafikseiten liegen unter dem ROM versteckt, das heißt, Seite 1 liegt bei \$A000-\$BFFF und Seite 2 bei \$E000-\$FFFF. Das Videoram der zweiten Seite liegt bei \$C000-\$C3FF und nimmt deshalb ebenfalls keinen Basic-Platz in Anspruch. Nur das Videoram der ersten Grafikseite zwingt zwangsläufig noch einmal 1 K ab, da es im Bereich von \$8C00 bis \$8FFF liegt. Damit sind 5 K abhanden gekommen. Zu guter Letzt braucht ein eigens erstellter Zeichensatz noch einmal 2 K, so daß zusammen 7 K Speicher unbenutzbar sind. Dies ist jedoch im Vergleich zu anderen Erweiterungen minimal: Schließlich gibt es keine Doppelbelegungen, also sowohl Zeichensatz als auch beide Grafikseiten können gleichzeitig benutzt werden. Der Zeichensatz liegt im Bereich von \$0800 bis \$0FFF, dort, wo normalerweise Basic-Programme stehen. Der Basic-Speicher unter GRAFIK-BASIC beginnt daher erst bei \$1000 oder dezimal 4096. Teil 1 und Teil 2 von GRAFIK-BASIC müssen mit dem Checker (Listing 1 und 2) oder als Basic-La-

der (Listing 3 und 4) eingegeben werden. Als Startadresse für den ersten Teil ist beim Checker C400, für den zweiten Teil 9000 anzugeben. Der erste Teil ist auch ohne den zweiten lauffähig. Er enthält die 21 Befehle für die hochauflösende Grafik. Geladen und gestartet wird er mit

```
LOAD "G-BASIC TEIL 1",8,1
```

```
(...,1,1 bei Kassette)
```

```
SYS 50176
```

```
NEW
```

Die komplette Erweiterung wird folgendermaßen aktiviert:

```
LOAD "G-BASIC TEIL 1",8,1
```

```
(...,1,1)
```

```
NEW
```

```
LOAD "G-BASIC TEIL 2",8,1
```

```
(...,1,1)
```

```
NEW
```

```
SYS 50176
```

```
NEW
```

GRAFIK-BASIC schützt sich selbst vor dem Überschreiben durch Basic-Programme. Die Erweiterung bleibt so lange aktiv, bis ein Reset ausgelöst wird. Danach kann sie mit SYS 50176 neu gestartet werden.

Listing 5 bietet einen Autostart-Erzeuger zu GRAFIK-BASIC. Er speichert nach dem Start mit RUN auf Diskette ein Ladeprogramm ab. Es wird mit

```
LOAD "LADER G-BASIC",8,1
```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|----|--------|
| C400 | 4C | 9E | CB | 4C | 58 | CC | 4C | 96 | 6F |
| C408 | C9 | 4C | 42 | C4 | 4C | 5A | C4 | 4C | 65 |
| C410 | 84 | C4 | 4C | 1B | C6 | 4C | 18 | C6 | BD |
| C418 | 4C | 2B | C6 | 4C | 28 | C6 | 4C | 20 | 29 |
| C420 | C9 | 4C | 1D | C9 | 4C | E9 | C7 | 4C | B0 |
| C428 | E6 | C7 | 4C | 62 | C6 | 4C | 5F | C6 | 1C |
| C430 | 4C | 6E | C9 | 4C | 2B | CA | 4C | 08 | 00 |
| C438 | CB | 4C | 38 | CA | 4C | F6 | CE | 4C | C5 |
| C440 | F3 | CE | EA | A9 | A0 | 85 | FC | A9 | D3 |
| C448 | 00 | 85 | FB | A2 | 20 | A0 | 00 | 91 | C4 |
| C450 | FB | C8 | D0 | FB | E6 | FC | CA | D0 | EC |
| C458 | F4 | 60 | 20 | 9E | B7 | 8E | F0 | CF | 5F |
| C460 | A2 | 03 | A9 | 8C | 95 | FC | A0 | 00 | E5 |
| C468 | 84 | FB | 84 | 97 | AD | F0 | CF | 91 | 71 |
| C470 | FB | C8 | C4 | 97 | D0 | F9 | E6 | FC | 20 |
| C478 | CA | F0 | 03 | 10 | F2 | 60 | A2 | E8 | 18 |
| C480 | 86 | 97 | D0 | EB | 20 | 9E | B7 | 8E | 7F |
| C488 | F0 | CF | 60 | 20 | EB | B7 | 8A | A8 | 76 |
| C490 | A6 | 15 | C0 | C8 | B0 | 0D | A5 | 14 | BC |
| C498 | E0 | 01 | 90 | 06 | D0 | 05 | C9 | 40 | BB |
| C4A0 | B0 | 01 | 60 | 4C | 48 | B2 | 8C | FD | E8 |
| C4A8 | CF | 8D | FC | CF | 8E | FE | CF | 85 | 49 |
| C4B0 | 14 | 86 | 15 | 98 | 4A | 4A | 4A | AA | AC |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| C4B8 | BD | 41 | C6 | 85 | AD | 8A | 29 | 03 | 05 |
| C4C0 | AA | BD | 5B | C6 | 85 | AC | 98 | 29 | CB |
| C4C8 | 07 | 18 | 65 | AC | 85 | AC | A5 | 14 | 12 |
| C4D0 | 29 | F9 | 85 | 63 | A9 | A0 | 05 | AD | B5 |
| C4D8 | 85 | AD | 18 | A5 | AC | 65 | 63 | 85 | 71 |
| C4E0 | AC | A5 | AD | 65 | 15 | 85 | AD | A5 | E7 |
| C4E8 | 14 | 29 | 07 | 49 | 07 | AA | BD | 39 | 8B |
| C4F0 | C6 | 85 | AB | 60 | 48 | AD | FE | CF | 55 |
| C4F8 | 4A | AD | FC | CF | 6A | 4A | 4A | 85 | AF |
| C500 | 6F | 68 | 48 | 38 | ED | FC | CF | 48 | 8F |
| C508 | 8A | ED | FE | CF | 85 | 6C | B0 | 0A | 8B |
| C510 | 68 | 49 | FF | 69 | 01 | 48 | A9 | 00 | 16 |
| C518 | E5 | 6C | 85 | 6A | 85 | 6E | 68 | 85 | 8D |
| C520 | 69 | 85 | 6D | 68 | 8D | FC | CF | 8E | BB |
| C528 | FE | CF | 98 | 18 | ED | FD | CF | 90 | DD |
| C530 | 04 | 49 | FF | 69 | FE | 85 | 68 | 8C | A8 |
| C538 | FD | CF | 66 | 6C | 38 | E5 | 69 | AA | 3B |
| C540 | A9 | FF | E5 | 6A | 85 | 70 | A4 | 6F | 6F |
| C548 | B0 | 05 | 0A | 20 | C7 | C5 | 38 | A5 | 2A |
| C550 | 6D | 65 | 8B | 85 | 6D | A5 | 6E | E9 | 3B |
| C558 | 00 | 85 | 6E | 84 | 6F | 20 | DC | C5 | CB |
| C560 | E8 | D0 | 05 | E6 | 70 | D0 | 01 | 60 | 7A |
| C568 | A5 | 6C | B0 | DE | 20 | 98 | C5 | 18 | 40 |
| C570 | A5 | 6D | 65 | 69 | 85 | 6D | A5 | 6E | 83 |
| C578 | 65 | 6A | 50 | DD | A5 | AC | 29 | 07 | 8A |

Listing 1: Hexdump Teil 1, mit Checker eingeben

geladen, startet sich selbst und lädt und initialisiert dann die beiden Teile von GRAFIK-BASIC. Damit das funktioniert, müssen die zwei Teile unbedingt mit den oben angegebenen Namen abgespeichert worden sein. Kassettenbesitzer müssen einige Zeilen in Listing 5 durch die Zeilen in Listing 6 ersetzen.

Listing 7 enthält ein Programm, mit dem es sehr leicht möglich ist, den Zeichensatz zu verändern oder einen völlig neuen zu erstellen. Bevor man das Programm eintippt und startet, muß GRAFIK-BASIC bereits aktiviert sein. Da die Erweiterung nicht mit dem Korrektor zusammenarbeitet, können bei Listing 7 leider keine Prüfsummen

angegeben werden. Zuerst wird das Arbeitsfeld aufgebaut, in dem die neuen Zeichen entworfen werden. Zuerst wird nach dem Zeichen gefragt, welches geändert oder neugestaltet werden soll. Danach wird das alte Zeichen gelesen und im Arbeitsfeld angezeigt. Jedes Zeichen besteht ja aus einer 8x8-Punkt-Matrix. Dabei wird jeder gesetzte Punkt durch einen Stern, jeder nichtgesetzte Punkt durch einen Punkt dargestellt. In der linken oberen Ecke des Arbeitsfeldes blinkt nun ein Cursor, der genau wie im normalen Basic-Editor mit den Cursortasten bewegt wird. Alle anderen Funktionen wie INST, DEL, HOME, CLR können ebenfalls wie gewohnt be-

nutzt werden. Zum Ändern eines Punktes bewegt man den Cursor an die gewünschte Position und drückt dann Stern oder Punkt, je nachdem, ob der Punkt gesetzt werden soll oder nicht.

Soll ein Zeichen völlig neu gestaltet werden, ist es empfehlenswert, vor der Erstellung das Arbeitsfeld mit CLR zu löschen. Mit der RETURN-Taste wird das neue Zeichen in den Zeichensatz übernommen. Danach können weitere Zeichen geändert werden. Beendet wird die Eingabe mit der Taste „Pfeil links“. In diesem Fall kann man den neuen Zeichensatz noch auf Floppy beziehungsweise Kassette abspeichern.

(Frank Riemenschneider)

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| C590 | C9 | 07 | F0 | 05 | 38 | A9 | 00 | B0 | 8B | C6C8 | 90 | 09 | C8 | CC | 88 | 03 | D0 | EC | EB |
| C598 | 04 | A9 | 38 | E6 | AD | 65 | AC | 85 | 1D | C6D0 | 4C | D9 | C6 | CE | 87 | 03 | 4C | CA | 71 |
| C590 | AC | A9 | 00 | 65 | AD | 85 | AD | 60 | 13 | C6D8 | C6 | A9 | 00 | 8D | 86 | 03 | A9 | 00 | BC |
| C598 | 30 | E2 | A5 | AC | 29 | 07 | F0 | 05 | 54 | C6E0 | 8D | 3D | 03 | AE | 86 | 03 | BD | 00 | E5 |
| C5A0 | 18 | A9 | FF | 90 | 04 | A9 | C7 | C6 | 3A | C6E8 | 04 | EC | 87 | 03 | D0 | 03 | 4C | E5 | D0 |
| C5A8 | AD | 65 | AC | 85 | AC | A5 | AD | E9 | 39 | C6F0 | C7 | 8D | 3C | 03 | 0E | 3C | 03 | 2E | 1A |
| C5B0 | 00 | 85 | AD | 60 | 46 | AB | 90 | 0E | E6 | C6F8 | 3D | 03 | 0E | 3C | 03 | 2E | 3D | 03 | 1B |
| C5B8 | 66 | AB | A5 | AC | C8 | 18 | 69 | 08 | C5 | C700 | 0E | 3C | 03 | 2E | 3D | 03 | A9 | 00 | 8A |
| C5C0 | 85 | AC | 90 | 02 | E6 | AD | 60 | 10 | EF | C708 | 8D | 3E | 03 | A9 | D0 | 8D | 3F | 03 | 28 |
| C5C8 | EB | 06 | AB | 90 | 0E | 26 | AB | A5 | ED | C710 | A9 | 00 | 8D | 40 | 03 | 8D | 41 | 03 | AA |
| C5D0 | AC | 90 | 38 | E9 | 08 | 85 | AC | B0 | F3 | C718 | AD | E7 | 03 | 29 | 02 | C9 | 00 | F0 | EB |
| C5D8 | 02 | C6 | AD | 60 | 78 | A0 | 35 | 84 | 13 | C720 | 05 | A9 | 08 | 8D | 41 | 03 | A0 | 00 | B6 |
| C5E0 | 01 | A0 | 00 | 08 | 49 | FF | A5 | AB | 9E | C728 | 84 | FA | 18 | AD | 3C | 03 | 6D | 3E | 5E |
| C5E8 | 24 | 97 | 30 | 05 | 49 | FF | 31 | AC | 88 | C730 | 03 | 8D | 42 | 03 | AD | 3D | 03 | 6D | BB |
| C5F0 | 2C | 11 | AC | 91 | AC | A9 | 37 | 85 | EB | C738 | 3F | 03 | 8D | 43 | 03 | 18 | AD | 42 | DB |
| C5F8 | 01 | 58 | A5 | AC | 85 | FB | A5 | AD | 25 | C740 | 03 | 6D | 40 | 03 | 8D | 42 | 03 | AD | 74 |
| C600 | 4A | 66 | FB | 4A | 66 | FB | 4A | 66 | E4 | C748 | 43 | 03 | 6D | 41 | 03 | 8D | 43 | 03 | 22 |
| C608 | FB | 29 | 03 | 09 | 8C | 85 | FC | AD | 21 | C750 | 18 | AD | 42 | 03 | 65 | FA | 85 | FD | 9E |
| C610 | F0 | CF | 91 | FB | 28 | A4 | 6F | 60 | 4B | C758 | AD | 43 | 03 | 69 | 00 | 85 | FE | 78 | 05 |
| C618 | A2 | 00 | 2C | A2 | 80 | 86 | 97 | 20 | 9D | C760 | A9 | 31 | 95 | 01 | A0 | 00 | B1 | FD | 4E |
| C620 | 8B | C4 | 20 | A6 | C4 | 4C | DC | C5 | D0 | C768 | 8D | 84 | 03 | A9 | 37 | 85 | 01 | 58 | BF |
| C628 | A2 | 00 | 2C | A2 | 80 | 20 | 1D | C6 | E3 | C770 | A2 | 08 | AD | FC | CF | 18 | 69 | 01 | 69 |
| C630 | 20 | FD | AE | 20 | 8B | C4 | 4C | F4 | D1 | C778 | 8D | FC | CF | AD | FE | CF | 69 | 00 | 4B |
| C638 | C4 | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 99 | C780 | 8D | FE | CF | AD | 84 | 03 | 3D | 38 | 37 |
| C640 | 80 | 00 | 01 | 02 | 03 | 05 | 06 | 07 | 7C | C788 | C6 | DD | 38 | C6 | D0 | 18 | 8A | 48 | 55 |
| C648 | 08 | 0A | 0B | 0C | 0D | 0F | 10 | 11 | FA | C790 | 98 | 48 | 18 | AC | FD | CF | AD | FC | 9B |
| C650 | 12 | 14 | 15 | 16 | 17 | 19 | 1A | 1B | FA | C798 | CF | AE | FE | CF | 20 | A6 | C4 | 20 | 6E |
| C658 | 1C | 1E | 1F | 00 | 40 | 80 | C0 | A2 | FB | C7A0 | DC | C5 | 68 | A8 | 68 | AA | CA | E0 | 7F |
| C660 | 00 | 2C | A2 | 80 | 86 | 97 | 20 | 9E | 67 | C7A8 | 01 | B0 | C7 | E6 | FA | EE | FD | CF | 6C |
| C668 | B7 | E0 | 00 | F0 | 07 | E0 | 01 | F0 | 21 | C7B0 | AD | 45 | 03 | 8D | FC | CF | AD | 46 | 72 |
| C670 | 0B | 4C | 48 | B2 | A9 | 19 | 8D | E7 | 8B | C7B8 | 03 | 8D | FE | CF | A4 | FA | C0 | 08 | 07 |
| C678 | 03 | 4C | 81 | C6 | A9 | 17 | 8D | E7 | AA | C7C0 | 90 | 8E | 18 | AD | FC | CF | 69 | 08 | FB |
| C680 | 03 | 20 | FD | AE | 20 | 8B | C4 | 8C | FF | C7C8 | 8D | FC | CF | 8D | 45 | 03 | AD | FE | C4 |
| C688 | FD | CF | 8D | FC | CF | 8E | FE | CF | 2F | C7D0 | CF | 69 | 00 | 8D | FE | CF | 8D | 46 | 4F |
| C690 | 8C | 47 | 03 | 8D | 45 | 03 | 8E | 46 | 45 | C7D8 | 03 | EE | 86 | 03 | AD | 47 | 03 | 8D | 74 |
| C698 | 03 | A9 | 13 | 20 | D2 | FF | 20 | FD | 44 | C7E0 | FD | CF | 4C | DE | C6 | 60 | A9 | 00 | AB |
| C6A0 | AE | 20 | 9E | AD | 24 | 0D | 30 | 06 | C0 | C7E8 | 2C | A9 | 80 | 95 | 97 | 20 | 8B | C4 | BC |
| C6A8 | 20 | DD | BD | 20 | 87 | 84 | 20 | A6 | 2E | C7F0 | 8C | A7 | 03 | 8D | A5 | 03 | 8E | A6 | E5 |
| C6B0 | 86 | 8D | 87 | 03 | 8D | 88 | 03 | 20 | 95 | C7F8 | 03 | 20 | FD | AE | 20 | 8A | AD | A0 | D5 |
| C6B8 | 24 | AB | A0 | 00 | B1 | 22 | C9 | 20 | 50 | C800 | 03 | A2 | 79 | 20 | D4 | BB | A9 | 00 | 7C |
| C6C0 | 90 | 11 | C9 | 80 | 90 | 04 | C9 | A1 | 7A | C808 | 85 | AF | 20 | FD | AE | 20 | 8A | AD | 64 |

C810 A0 03 A2 7E 20 D4 BB A9 1F
 C818 F5 A0 E2 20 A2 BB A9 D1 D5
 C820 A0 B9 20 0F BB A2 88 A0 F9
 C828 03 20 D4 BB A9 20 8D 84 8D
 C830 03 20 FD AE 20 8A AD A0 D5
 C838 03 A2 83 20 D4 BB 20 FD 01
 C840 AE 20 8A AD A0 03 A2 8D 1C
 C848 20 D4 BB A9 83 A0 03 20 25
 C850 A2 BB 20 64 E2 A9 79 A0 B5
 C858 03 20 28 BA A9 11 A0 BF CA
 C860 20 67 B8 20 9B BC 19 A5 A3
 C868 65 6D A5 03 85 14 A5 64 4B
 C870 6D A6 03 85 15 A9 83 A0 95
 C878 03 20 A2 BB 20 6B E2 A9 B8
 C880 7E A0 03 20 28 BA A9 11 C8
 C888 A0 BF 20 67 B8 20 9B BC 11
 C890 18 A5 65 6D A7 03 A8 A5 12
 C898 14 A6 15 8D 3F 03 8E 40 81
 C8A0 03 8C 41 03 20 A6 C4 20 D4
 C8A8 DC C5 A5 AF C9 00 F0 0F B6
 C8B0 AD 3C 03 AE 3D 03 AC 3E 6E
 C8B8 03 20 F4 C4 4C C1 C8 E6 80
 C8C0 AF AD 3F 03 8D 3C 03 AD E5
 C8C8 40 03 8D 3D 03 AD 41 03 21
 C8D0 8D 3E 03 A9 83 A0 03 20 70
 C8D8 A2 BB A9 88 A0 03 20 67 5E
 C8E0 B8 A2 83 A0 03 20 D4 BB F5
 C8E8 A9 8D A0 03 20 50 B8 20 20
 C8F0 2B BC 30 03 4C 4B C8 A9 BC
 C8F8 8D A0 03 20 A2 BB A2 83 D6
 C900 A0 03 20 D4 BB A9 12 8D 80
 C908 BD C8 A9 C9 8D BE C8 4C 20
 C910 4B C8 A9 C1 8D BD C8 A9 5A
 C918 C8 8D BE C8 60 A2 00 2C C4
 C920 A2 80 86 97 20 8B C4 8D DD
 C928 3C 03 8E 3D 03 8C 3E 03 3C
 C930 20 FD AE 20 8B C4 CC 3E 06
 C938 03 90 30 8D 3F 03 8E 40 A1
 C940 03 8C 41 03 AD 3C 03 AE 7B
 C948 3D 03 AC 3E 03 20 A6 C4 6C
 C950 20 DC C5 AD 3F 03 AE 40 06
 C958 03 AC 3E 03 20 F4 C4 EE 95
 C960 3E 03 AD 3E 03 CD 41 03 1E
 C968 D0 DA 60 4C 48 B2 A9 00 49
 C970 A2 A0 85 FB 86 FC A2 20 99
 C978 78 A9 35 85 01 A0 00 B1 31
 C980 FB 49 FF 91 FB C8 D0 F7 2A
 C988 E6 FC CA D0 F0 A9 37 85 DD
 C990 01 58 60 4C 58 CC EA 20 13
 C998 9E B7 E0 00 F0 F5 E0 01 A0
 C9A0 F0 07 E0 02 F0 15 4C 48 A4
 C9A8 B2 A9 3B 8D 11 D0 A9 38 6A
 C9B0 8D 18 D0 A9 95 8D 00 DD C7
 C9B8 4C CA C9 A9 3B 8D 11 D0 92
 C9C0 A9 08 8D 18 D0 A9 00 8D AF
 C9C8 00 DD 20 FD AE 20 9E B7 BC
 C9D0 E0 01 F0 07 E0 02 F0 2B 68
 C9D8 4C 48 B2 A9 A0 8D 44 C4 A0
 C9E0 8D D5 C4 8D 71 C9 8D 30 F4
 C9E8 CA 8D 70 CB A9 8C 8D 63 29
 C9F0 C4 8D 0C C6 A9 F0 8D 5E 66
 C9F8 C4 8D 6D C4 8D 88 C4 8D 1C
 CA00 10 C6 60 A9 E0 8D 44 C4 D5
 CA08 8D D5 C4 8D 71 C9 8D 30 F4

CA10 CA 8D 70 CB A9 C0 8D 63 F5
 CA18 C4 8D 0C C6 A9 F1 8D 5E 65
 CA20 C4 8D 6D C4 8D 88 C4 8D 1C
 CA28 10 C6 60 EA 20 D4 E1 A0 4F
 CA30 A0 A2 00 A9 00 4C D5 FF E0
 CA38 A9 01 A2 04 A0 00 20 BA 4A
 CA40 FF A9 00 20 BD FF 20 C0 54
 CA48 FF A2 01 20 C9 FF A9 FF B2
 CA50 85 61 A9 07 85 FD A9 1C DB
 CA58 85 97 A9 00 8D 3C 03 A9 42
 CA60 28 8D 3D 03 A2 04 BD 03 2C
 CA68 CB 20 D2 FF CA 10 F7 A9 84
 CA70 00 85 63 85 64 AD 3C 03 4A
 CA78 85 65 A9 00 85 FE A5 63 92
 CA80 A6 64 A4 65 20 A6 C4 A0 1F
 CA88 00 78 A9 35 85 01 B1 AC 84
 CA90 85 FA A9 37 85 01 58 A5 34
 CA98 FA A6 FE 9D 40 03 E6 65 72
 CAA0 E8 86 FE E4 FD D0 D7 A9 D6
 CAA8 00 A0 07 A6 FD 1E 40 03 DE
 CAB0 2A CA 10 F9 25 61 09 80 C6
 CAB8 20 D2 FF 88 10 ED A5 63 28
 CAC0 18 69 08 85 63 90 02 E6 22
 CAC8 64 CE 3D 03 D0 A7 A9 0D 95
 CAD0 20 D2 FF AD 3C 03 18 69 89
 CAD8 07 8D 3C 03 C6 97 F0 03 CF
 CAE0 4C 5F CA A9 04 C5 FD F0 5A
 CAE8 0C 85 FD A9 01 85 97 A9 46
 CAF0 0F 85 61 D0 EB A9 0F 20 4D
 CAF8 D2 FF 20 CC FF A9 01 20 5F
 CB00 C3 FF 60 50 00 10 18 08 D8
 CB08 EA 20 9E AD 24 0D 30 06 FB
 CB10 20 DD BD 20 87 B4 20 A6 2E
 CB18 B6 85 FD A9 03 85 BC 85 3A
 CB20 FC A9 3C 85 8B 85 FB A0 9A
 CB28 00 B1 22 91 FB C8 C4 FD DB
 CB30 D0 F7 A9 2C 91 FB C8 A9 08
 CB38 50 91 FB C8 A9 2C 91 FB 04
 CB40 C8 A9 57 91 FB C8 84 B7 E5
 CB48 20 FD AE 20 9E B7 E0 08 70
 CB50 F0 07 E0 01 F0 03 4C 48 B7
 CB58 B2 A9 01 A0 02 20 BA FF 07
 CB60 20 C0 FF A2 01 20 C9 FF 68
 CB68 A9 00 85 FB 20 D2 FF A9 D7
 CB70 A0 85 FC 20 D2 FF A2 20 4B
 CB78 A0 00 78 A9 35 85 01 B1 6F
 CB80 FB 85 FD A9 37 85 01 58 25
 CB88 A5 FD 20 D2 FF C8 D0 EA 14
 CB90 E6 FC CA D0 E3 20 CC FF 73
 CB98 A9 01 20 C3 FF 60 AD 11 3F
 CBA0 D0 8D E3 CF AD 18 D0 8D 34
 CBA8 EA CF AD 00 DD 8D EB CF 32
 CBB0 A9 6D 8D 04 03 A9 CC 8D 5E
 CBB8 05 03 A9 0E 8D 20 D0 8D 4C
 CBC0 21 D0 A9 30 8D 06 03 A9 AB
 CBC8 CD 8D 07 03 20 5B CE A2 35
 CBD0 02 A0 03 86 D6 84 D3 20 E4
 CBD8 6C E5 A9 0A A0 CC 20 1E FC
 CBE0 AB A9 86 8D 00 03 A9 CF D2
 CBE8 8D 01 03 A2 04 A0 04 86 D0
 CBF0 D6 84 D3 20 6C E5 A9 35 FF
 CBF8 A0 CC 20 1E AB A9 10 85 64
 CC00 2C AD 00 08 8D 00 10 4C C8
 CC08 D0 CF 93 1F 11 20 2A 2A 66

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| CC10 | 2A | 2A | 20 | 47 | 52 | 41 | 46 | 49 | E7 | CE08 | 50 | D9 | 52 | 45 | 4E | 45 | D7 | 52 | 12 |
| CC18 | 4B | 20 | 2D | 20 | 42 | 41 | 53 | 49 | 43 | CE10 | 45 | 4E | 55 | CD | 44 | 55 | 4D | D0 | EC |
| CC20 | 43 | 20 | 20 | 20 | 56 | 45 | 52 | 53 | 33 | CE18 | 52 | 41 | CD | 4D | 45 | 52 | 47 | C5 | 06 |
| CC28 | 49 | 4F | 4E | 20 | 32 | 2E | 32 | 20 | 3E | CE20 | 48 | 45 | 58 | C1 | 53 | 50 | 52 | 44 | AC |
| CC30 | 2A | 2A | 2A | 2A | 00 | 28 | 43 | 29 | F2 | CE28 | 45 | C6 | 53 | 50 | 52 | 49 | CE | 53 | 06 |
| CC38 | 20 | 31 | 39 | 38 | 35 | 20 | 20 | 20 | 05 | CE30 | 50 | 52 | 4F | 46 | C6 | 53 | 50 | 52 | 78 |
| CC40 | 46 | 52 | 41 | 4E | 48 | 20 | 52 | 49 | 1B | CE38 | 53 | 45 | D4 | 53 | 50 | 52 | 43 | 4F | 81 |
| CC48 | 45 | 4D | 45 | 4E | 53 | 43 | 48 | 4E | F9 | CE40 | CC | 5A | 47 | 45 | D2 | 5A | 49 | CE | 86 |
| CC50 | 45 | 49 | 44 | 45 | 52 | 11 | 11 | 00 | 4D | CE48 | 5A | 4F | 46 | C6 | 5A | 4C | 4F | 41 | A8 |
| CC58 | AD | E9 | CF | 8D | 11 | D0 | AD | EA | 0B | CE50 | C4 | 5A | 53 | 41 | 56 | C5 | 47 | 4F | 05 |
| CC60 | CF | 8D | 18 | D0 | AD | EB | CF | 8D | 8E | CE58 | 46 | C6 | 00 | A9 | 66 | A0 | CE | 8D | DF |
| CC68 | 00 | DD | 4C | 44 | E5 | A6 | 7A | A0 | 45 | CE60 | 08 | 03 | 8C | 09 | 03 | 60 | 20 | 73 | D8 |
| CC70 | 04 | 84 | 0F | BD | 00 | 02 | 10 | 07 | DB | CE68 | 00 | 20 | 6F | CE | 4C | AE | A7 | C9 | FE |
| CC78 | C9 | FF | F0 | 3E | E8 | D0 | F4 | C9 | BE | CE70 | CC | 90 | 04 | C9 | FE | 90 | 06 | 20 | CC |
| CC80 | 20 | F0 | 37 | 85 | 08 | C9 | 22 | F0 | 55 | CE78 | 79 | 00 | 4C | ED | A7 | 38 | E9 | CC | 63 |
| CC88 | 55 | 24 | 0F | 70 | 2D | C9 | 3F | D0 | A4 | CE80 | 0A | AA | BD | 8E | CE | 48 | BD | 8D | 45 |
| CC90 | 04 | A9 | 99 | D0 | 25 | C9 | 30 | 90 | 22 | CE88 | CE | 48 | 4C | 73 | 00 | 00 | 30 | 00 | 8F |
| CC98 | 04 | C9 | 3C | 90 | 1D | 84 | 71 | A0 | 53 | CE90 | 30 | 05 | C4 | 08 | C4 | 0B | C4 | 0E | 54 |
| CCA0 | 00 | 84 | 0B | 88 | 86 | 7A | CA | C8 | 0E | CE98 | C4 | 11 | C4 | 14 | C4 | 17 | C4 | 1A | B8 |
| CCA8 | E8 | BD | 00 | 02 | 38 | F9 | 9E | A0 | 66 | CEA0 | C4 | 1D | C4 | 20 | C4 | 23 | C4 | 26 | 88 |
| CCB0 | F0 | F5 | C9 | 00 | D0 | 2F | 05 | 0B | DF | CEA8 | C4 | 29 | C4 | 2C | C4 | 2F | C4 | 32 | 58 |
| CCB8 | A4 | 71 | E8 | C8 | 99 | FB | 01 | C9 | 2A | CEB0 | C4 | 35 | C4 | 38 | C4 | 3B | C4 | 3E | 28 |
| CCC0 | 00 | F0 | 38 | 38 | E9 | 3A | F0 | 04 | AB | CEB8 | C4 | FF | 8F | 05 | 90 | 02 | 90 | 08 | 64 |
| CCC8 | C9 | 49 | D0 | 02 | 85 | 0F | 38 | E9 | 12 | CEC0 | 90 | 0B | 90 | 0E | 90 | 11 | 90 | 14 | 00 |
| CCD0 | 55 | D0 | A0 | 85 | 08 | BD | 00 | 02 | EB | CEC8 | 90 | 17 | 90 | 1A | 90 | 1D | 90 | 20 | D1 |
| CCD8 | F0 | E0 | C5 | 08 | F0 | DC | C8 | 99 | 0F | CED0 | 90 | 23 | 90 | 26 | 90 | 29 | 90 | 2C | A1 |
| CCF0 | FB | 01 | E8 | D0 | F0 | A6 | 7A | E6 | EF | CED8 | 90 | 2F | 90 | 32 | 90 | 35 | 90 | 38 | 71 |
| CCE8 | 0B | C8 | B9 | 9D | A0 | 10 | FA | B9 | 30 | CEE0 | 90 | 3B | 90 | 3E | 90 | 41 | 90 | 44 | 41 |
| CCF0 | 9E | A0 | D0 | B5 | F0 | 0F | BD | 00 | B6 | CEE8 | 90 | 47 | 90 | 4A | 90 | 4D | 90 | 50 | 11 |
| CCF8 | 02 | 10 | BD | 99 | FD | 01 | C6 | 7B | 5C | CEF0 | 90 | 00 | 30 | A2 | 00 | 2C | A2 | 80 | 14 |
| CD00 | A9 | FF | 85 | 7A | 60 | A0 | 00 | B9 | BD | CEF8 | 86 | 97 | 20 | 8B | C4 | 8D | 3C | 03 | F5 |
| CD08 | 65 | CD | D0 | 02 | C8 | E8 | BD | 00 | 02 | CF00 | 8E | 3D | 03 | 8C | 3E | 03 | 20 | FD | 26 |
| CD10 | 02 | 38 | F9 | 65 | CD | F0 | F5 | C9 | 66 | CF08 | AE | 20 | 8B | C4 | CC | 3E | 03 | 90 | 56 |
| CD18 | 80 | D0 | 04 | 05 | 08 | D0 | 99 | A6 | DE | CF10 | 49 | 8D | 3F | 03 | 8E | 40 | 03 | 8C | BD |
| CD20 | 7A | E6 | 08 | C8 | B9 | 64 | CD | 10 | EA | CF18 | 41 | 03 | AD | 3C | 03 | AE | 3D | 03 | 3E |
| CD28 | FA | B9 | 65 | CD | D0 | E0 | F0 | C6 | F3 | CF20 | AC | 3E | 03 | 20 | A6 | C4 | 20 | DC | 77 |
| CD30 | 10 | 0F | 24 | 0F | 30 | 0B | C9 | FF | 04 | CF28 | C5 | AD | 3F | 03 | AE | 40 | 03 | AC | 19 |
| CD38 | F0 | 07 | C9 | CC | B0 | 06 | 4C | 24 | B7 | CF30 | 3E | 03 | 20 | F4 | C4 | AD | 3F | 03 | BB |
| CD40 | A7 | 4C | F3 | A6 | 38 | E9 | CB | AA | 18 | CF38 | AE | 40 | 03 | AC | 41 | 03 | 20 | F4 | 2F |
| CD48 | 84 | 49 | A0 | FF | CA | F0 | 08 | C8 | F7 | CF40 | C4 | AD | 3C | 03 | AE | 3D | 03 | AC | 18 |
| CD50 | B9 | 65 | CD | 10 | FA | 30 | F5 | C8 | 06 | CF48 | 41 | 03 | 20 | F4 | C4 | AD | 3C | 03 | BB |
| CD58 | B9 | 65 | CD | 30 | 05 | 20 | 47 | AB | 71 | CF50 | AE | 3D | 03 | AC | 3E | 03 | 20 | F4 | 2F |
| CD60 | D0 | F5 | 4C | EF | A6 | 57 | 45 | 52 | 7B | CF58 | C4 | 60 | 4C | 48 | B2 | 13 | 20 | 20 | 06 |
| CD68 | 4E | 45 | D2 | 47 | 4F | 54 | 45 | 58 | 7C | CF60 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 |
| CD70 | D4 | 47 | 4D | 4F | 44 | C5 | 47 | 43 | 0E | CF68 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 |
| CD78 | 4C | 45 | 41 | D2 | 47 | 43 | 4F | CC | FE | CF70 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 |
| CD80 | 50 | 43 | 4F | CC | 53 | 50 | 4C | 4F | 91 | CF78 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 |
| CD88 | D4 | 43 | 50 | 4C | 4F | D4 | 53 | 4C | 17 | CF80 | 20 | 20 | 20 | 20 | 20 | 00 | 86 | FB | AB |
| CD90 | 49 | 4E | C5 | 43 | 4C | 49 | 4E | C5 | 09 | CF88 | A5 | D6 | 85 | FC | A5 | D3 | 85 | FD | B3 |
| CD98 | 53 | 42 | 4C | 4F | C3 | 43 | 42 | 4C | 84 | CF90 | A2 | 01 | A0 | 01 | 86 | D6 | 84 | D3 | A0 |
| CDA0 | 4F | C3 | 53 | 43 | 49 | D2 | 43 | 43 | 14 | CF98 | 20 | 6C | E5 | A9 | 5D | A0 | CF | 20 | 5C |
| CDA8 | 49 | D2 | 53 | 54 | 45 | 58 | D4 | 43 | F4 | CFA0 | 1E | AB | A5 | FC | 85 | D6 | A5 | FD | 74 |
| CDB0 | 54 | 45 | 58 | D4 | 47 | 49 | 4E | D6 | 0A | CFA8 | 85 | D3 | 20 | 6C | E5 | AD | E9 | CF | B8 |
| CDB8 | 47 | 4C | 4F | 41 | C4 | 47 | 53 | 41 | 98 | CFB0 | 8D | 11 | D0 | AD | EA | CF | 8D | 18 | 2E |
| CDC0 | 56 | C5 | 47 | 48 | 41 | 52 | C4 | 53 | F0 | CFB8 | D0 | AD | EB | CF | 8D | 00 | DD | A6 | 02 |
| CDC8 | 46 | 52 | 41 | 4D | C5 | 43 | 46 | 52 | 5E | CFC0 | FB | 8A | 30 | 03 | 4C | 3A | A4 | 4C | 07 |
| CDD0 | 41 | 4D | C5 | 45 | 52 | 52 | 4F | D2 | F1 | CFC8 | 74 | A4 | 02 | FA | 00 | FF | 02 | DF | FF |
| CDD8 | 44 | 49 | 53 | CB | 44 | 49 | D2 | 4C | 04 | CFD0 | A9 | 9C | 85 | 38 | 4C | 86 | E3 | DF | 33 |
| CDE0 | 49 | 4E | D6 | 4C | 46 | 49 | 4C | CC | 02 | CFD8 | 00 | DF | 00 | DF | 04 | 6F | 02 | BF | 1D |
| CDE8 | 4C | 43 | 4F | CC | 4C | 4C | 4F | 41 | 9B | CFE0 | 20 | 2E | 6F | 2F | 02 | 06 | 08 | 0F | 27 |
| CDF0 | C4 | 4C | 53 | 41 | 56 | C5 | 4C | 50 | 17 | CFE8 | 00 | 1B | 15 | C7 | 00 | 0E | 32 | 2C | 2C |
| CDF8 | 52 | C9 | 4C | 48 | 41 | 52 | C4 | 4C | F4 | CFF0 | 6E | 61 | 00 | 00 | 00 | 00 | 06 | 0C | 07 |
| CE00 | 53 | 43 | 52 | 4F | CC | 5A | 43 | 4F | 79 | CFF8 | 0E | 0D | 00 | 0C | 38 | C7 | 01 | 0D | 58 |

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|
| 9000 | 4C | 54 | 90 | 4C | 7D | 90 | 4C | E6 | 8F | 91D9 | FA | A5 | FB | 69 | 00 | 85 | FB | A0 | BC |
| 9008 | 90 | 4C | 08 | 91 | 4C | 22 | 91 | 4C | 2A | 91E0 | 00 | CC | 3E | 03 | F0 | 13 | A5 | FC | F5 |
| 9010 | 2F | 91 | 4C | 6B | 92 | 4C | 4A | 92 | 7D | 91E8 | 18 | 69 | 28 | 85 | FC | A5 | FD | 69 | 3D |
| 9018 | 4C | B1 | 97 | 4C | 9C | 92 | 4C | B7 | 85 | 91F0 | 00 | 85 | FD | C8 | CC | 3E | 03 | D0 | 71 |
| 9020 | 93 | 4C | F5 | 92 | 4C | 2F | 92 | 4C | 0C | 91F8 | ED | 18 | A5 | FC | 6D | 3F | 03 | 85 | 2A |
| 9028 | 5E | 93 | 4C | B4 | 94 | 4C | 74 | 95 | 8A | 9200 | FC | A5 | FD | 69 | 00 | 85 | FD | 18 | 4A |
| 9030 | 4C | AB | 95 | 4C | ED | 95 | 4C | 44 | 4A | 9208 | A5 | FB | 65 | FE | 95 | FB | AE | 45 | 05 |
| 9038 | 96 | 4C | 11 | 97 | 4C | 26 | 97 | 4C | 35 | 9210 | 03 | A0 | 00 | AD | 40 | 03 | 91 | FA | 8B |
| 9040 | 43 | 97 | 4C | 8F | 97 | 4C | 24 | 93 | 46 | 9218 | C8 | CC | 46 | 03 | D0 | F5 | A5 | FA | C5 |
| 9048 | 4C | 32 | 93 | 4C | 84 | 92 | 4C | 3E | 61 | 9220 | 18 | 69 | 28 | 85 | FA | A5 | FB | 69 | 39 |
| 9050 | 93 | 4C | 4A | 93 | A9 | 01 | A2 | 08 | 3F | 9228 | 00 | 85 | FB | CA | D0 | E3 | 60 | A9 | 51 |
| 9058 | A0 | 0F | 20 | BA | FF | A9 | 00 | 20 | 2D | 9230 | 01 | A0 | 01 | 91 | 2B | 20 | 33 | A5 | 6B |
| 9060 | BD | FF | 20 | C0 | FF | A2 | 01 | 20 | 5D | 9238 | A5 | 22 | 69 | 02 | 85 | 2D | A5 | 23 | C3 |
| 9068 | C6 | FF | 20 | CF | FF | 20 | D2 | FF | CA | 9240 | 69 | 00 | 85 | 2E | 20 | 63 | A6 | 4C | D6 |
| 9070 | 24 | 90 | 50 | F6 | 20 | CC | FF | A9 | 99 | 9248 | 74 | A4 | A5 | 9D | 95 | FD | A9 | 00 | 09 |
| 9078 | 01 | 20 | C3 | FF | 60 | A9 | 24 | 85 | FC | 9250 | 85 | 9D | 20 | D4 | E1 | A2 | E8 | A0 | BB |
| 9080 | FB | A9 | FB | 85 | BB | A9 | 00 | 85 | 55 | 9258 | 07 | A9 | 00 | 85 | FB | A9 | 04 | 85 | AB |
| 9088 | BC | A9 | 01 | 85 | B7 | A9 | 08 | 85 | 21 | 9260 | FC | A9 | FB | 20 | D8 | FF | A5 | FD | AE |
| 9090 | BA | A9 | 60 | 85 | B9 | 20 | D5 | F3 | 66 | 9268 | 85 | 9D | 60 | A5 | 9D | 85 | FD | A9 | 0F |
| 9098 | A5 | BA | 20 | B4 | FF | A5 | B9 | 20 | 4A | 9270 | 00 | 85 | 9D | 20 | D4 | E1 | A2 | 00 | 8D |
| 90A0 | 96 | FF | A9 | 00 | 85 | 90 | A0 | 03 | D2 | 9278 | A0 | 04 | A9 | 00 | 20 | D5 | FF | A5 | E9 |
| 90A8 | 84 | FB | 20 | A5 | FF | 85 | FC | A4 | D6 | 9280 | FD | 85 | 9D | 60 | A9 | 15 | 8D | 18 | BD |
| 90B0 | 90 | D0 | 2F | 20 | A5 | FF | A4 | 90 | 89 | 9288 | D0 | 8D | EA | CF | EA | A9 | D0 | 8D | E1 |
| 90B8 | D0 | 28 | A4 | FB | 88 | D0 | E9 | A6 | 4C | 9290 | 0C | C7 | 60 | 4C | A4 | AA | 4C | A4 | FC |
| 90C0 | FC | 20 | CD | BD | A9 | 20 | 20 | D2 | C2 | 9298 | AA | 20 | FD | AA | A9 | 04 | 85 | BA | 4B |
| 90C8 | FF | 20 | A5 | FF | A6 | 90 | D0 | 12 | 58 | 92A0 | A9 | 7E | 85 | 88 | A9 | 00 | A0 | 04 | 3C |
| 90D0 | AA | F0 | 06 | 20 | D2 | FF | 4C | C9 | F7 | 92A8 | 85 | 71 | 84 | 72 | 85 | B7 | 85 | B9 | C0 |
| 90D8 | 90 | A9 | 0D | 20 | D2 | FF | A0 | 02 | 45 | 92B0 | 20 | C0 | FF | A6 | B8 | 20 | C9 | FF | 1A |
| 90E0 | D0 | C6 | 20 | 42 | F6 | 60 | A9 | 01 | 25 | 92B8 | A2 | 19 | A9 | 0D | 20 | D2 | FF | A0 | D1 |
| 90E8 | A2 | 08 | A0 | 0F | 20 | BA | FF | A9 | E6 | 92C0 | 00 | B1 | 71 | 85 | 67 | 29 | 3F | 06 | B3 |
| 90F0 | 00 | 20 | BD | FF | 20 | C0 | FF | A2 | 5C | 92C8 | 67 | 24 | 67 | 10 | 02 | 09 | 80 | 70 | A2 |
| 90F8 | 01 | 20 | C9 | FF | 20 | A4 | AA | 20 | B2 | 92D0 | 02 | 09 | 40 | 20 | D2 | FF | C8 | C0 | F4 |
| 9100 | CC | FF | A9 | 01 | 20 | C3 | FF | 60 | 71 | 92D8 | 28 | D0 | E6 | 98 | 18 | 65 | 71 | 85 | 46 |
| 9108 | A9 | 00 | 85 | FB | A9 | 04 | 85 | FC | 60 | 92E0 | 71 | 90 | 02 | E6 | 72 | CA | D0 | D2 | A4 |
| 9110 | AA | A0 | 00 | B1 | FB | 49 | 80 | 91 | FA | 92E8 | A9 | 0D | 20 | D2 | FF | 20 | CC | FF | 95 |
| 9118 | FB | C8 | D0 | F7 | E6 | FC | CA | D0 | F0 | 92F0 | A9 | 7E | 4C | C3 | FF | 78 | A5 | 01 | DF |
| 9120 | F0 | 60 | A9 | 04 | 85 | FE | 20 | 9E | 3E | 92F8 | 85 | FE | 29 | FB | 85 | 01 | A9 | D0 | 13 |
| 9128 | B7 | 8E | 40 | 03 | 4C | 3C | 91 | A9 | 5D | 9300 | 85 | FB | A9 | 08 | 85 | FD | A9 | 00 | 5C |
| 9130 | D8 | 85 | FE | 20 | 9E | B7 | 8A | 29 | 78 | 9308 | 85 | FA | 85 | FC | A2 | 08 | A0 | 00 | 4E |
| 9138 | 0F | 8D | 40 | 03 | 20 | FD | AE | 20 | 71 | 9310 | B1 | FA | 91 | FC | C8 | D0 | F9 | E6 | 57 |
| 9140 | 9E | B7 | E0 | 19 | 90 | 03 | 4C | 48 | 3E | 9318 | FB | E6 | FD | CA | D0 | F0 | A5 | FE | CF |
| 9148 | B2 | 8E | 3C | 03 | 20 | FD | AE | 20 | 0E | 9320 | 85 | 01 | 58 | 60 | A9 | 98 | 8D | FF | 1A |
| 9150 | 9E | B7 | E0 | 28 | 90 | 03 | 4C | 48 | 2F | 9328 | 92 | 20 | F5 | 92 | A9 | D0 | 8D | FF | 3B |
| 9158 | B2 | 8E | 3D | 03 | 20 | FD | AE | 20 | 0F | 9330 | 92 | 60 | A9 | 12 | 8D | 18 | D0 | 8D | 7F |
| 9160 | 9E | B7 | E0 | 19 | 90 | 03 | 4C | 48 | 3E | 9338 | EA | CF | 4C | CA | 97 | 60 | 20 | D4 | 21 |
| 9168 | B2 | 8E | 3E | 03 | 20 | FD | AE | 20 | 10 | 9340 | E1 | A0 | 08 | A2 | 00 | A9 | 00 | 4C | B3 |
| 9170 | 9E | B7 | E0 | 28 | 90 | 03 | 4C | 48 | 2F | 9348 | D5 | FF | 20 | D4 | E1 | A0 | 0F | A2 | D1 |
| 9178 | B2 | 8E | 3F | 03 | AD | 3C | 03 | CD | 07 | 9350 | FF | A9 | 00 | 85 | FA | A9 | 08 | 85 | A5 |
| 9180 | 3E | 03 | 90 | 0F | D0 | F0 | AD | 3D | 08 | 9358 | FB | A9 | FA | 4C | D8 | FF | EA | 20 | A2 |
| 9188 | 03 | CD | 3F | 03 | 90 | 05 | F0 | 03 | EA | 9360 | 8A | AD | 20 | F7 | B7 | A5 | 14 | 85 | A8 |
| 9190 | 4C | 48 | B2 | AD | 3F | 03 | 38 | ED | 90 | 9368 | FB | A5 | 15 | 85 | FC | 20 | FD | AE | 10 |
| 9198 | 3D | 03 | 18 | 69 | 01 | 8D | 46 | 03 | A1 | 9370 | 20 | 8A | AD | 20 | F7 | B7 | A5 | 2B | DC |
| 91A0 | AD | 3E | 03 | 38 | ED | 3C | 03 | 18 | D6 | 9378 | A4 | 2C | 85 | 22 | 84 | 23 | A0 | 03 | D8 |
| 91A8 | 69 | 01 | 8D | 45 | 03 | A9 | 00 | 85 | 85 | 9380 | A5 | FC | 91 | 22 | 88 | A5 | FB | 91 | 65 |
| 91B0 | FA | 85 | FB | 85 | FC | 85 | FD | A0 | BD | 9388 | 22 | 88 | B1 | 22 | F0 | 25 | 85 | 25 | 53 |
| 91B8 | 00 | CC | 3C | 03 | F0 | 13 | A5 | FA | F5 | 9390 | 88 | B1 | 22 | 85 | 24 | A0 | 02 | 18 | E4 |
| 91C0 | 18 | 69 | 28 | 85 | FA | A5 | FB | 69 | 39 | 9398 | B1 | 22 | 65 | 14 | 91 | 24 | C8 | B1 | 62 |
| 91C8 | 00 | 85 | FB | C8 | CC | 3C | 03 | D0 | 71 | 93A0 | 22 | 65 | 15 | 91 | 24 | B0 | 0D | A5 | 1F |
| 91D0 | ED | 18 | A5 | FA | 6D | 3D | 03 | 85 | 2E | 93A8 | 24 | 85 | 22 | A5 | 25 | 85 | 23 | A0 | 41 |
| | | | | | | | | | | 93B0 | 01 | D0 | D7 | 60 | 4C | 48 | B2 | EA | 74 |
| | | | | | | | | | | 93B8 | 20 | 9E | B7 | 8A | 4A | 08 | 20 | FD | 14 |

Listing 2: Hexdump Teil 2, mit Checker eingeben

| | | | |
|------|----------------------------|------|----------------------------|
| 93C0 | AE 20 9E B7 E0 19 90 03 C8 | 95A8 | 85 FE 60 A5 2B 85 FB A5 3F |
| 93C8 | 4C 48 B2 86 FA 20 FD AE 57 | 95B0 | 2C 85 FC 20 04 E1 A5 2D ED |
| 93D0 | 20 9E B7 E0 19 90 03 4C 9B | 95B8 | 38 E9 02 85 2B A5 2E 85 FD |
| 93D8 | 48 B2 86 FB 8A A6 FA A4 5B | 95C0 | 2C B0 02 C6 2C A6 2B A4 C7 |
| 93E0 | FB 39 E5 FA B0 06 49 FF A1 | 95C8 | 2C A9 00 20 D5 FF 86 2D 93 |
| 93E8 | A6 FB A4 FA 85 FE 28 09 FD | 95D0 | 84 2E A5 FB 85 2B A5 FC 03 |
| 93F0 | 90 03 C8 98 AA BD 80 94 95 | 95D8 | 85 2C A9 76 A0 A3 20 1E 8B |
| 93F8 | 85 62 BD 9A 94 85 61 28 8E | 95E0 | AB 20 2A A5 20 8E A6 20 28 |
| 9400 | 08 90 04 E9 01 85 61 A5 CD | 95E8 | 33 A5 4C 77 A6 A9 F8 A2 B6 |
| 9408 | 62 29 03 09 04 28 08 20 F7 | 95F0 | 95 8D 0A 03 8E 0B 03 60 35 |
| 9410 | 3B 94 28 08 A5 61 90 0A 91 | 95F8 | A9 00 85 0D 20 73 00 C9 05 |
| 9418 | 69 27 85 61 90 0C E6 62 8D | 9600 | 24 F0 06 20 79 00 4C 8D 53 |
| 9420 | B0 08 E9 27 85 61 B0 02 3A | 9608 | AE 20 3A 96 20 73 00 90 50 |
| 9428 | C6 62 A5 62 29 03 09 D8 FE | 9610 | 0B C9 41 90 1F C9 47 80 E2 |
| 9430 | 28 08 20 3B 94 C6 FE 10 C1 | 9618 | 1B 38 E9 07 38 E9 30 48 FC |
| 9438 | CE 28 60 85 62 90 03 4C 0A | 9620 | A5 61 F0 07 18 69 04 B0 30 |
| 9440 | 60 94 A0 00 B1 61 AA A0 C5 | 9628 | 0E 85 61 68 F0 DE 20 7E 37 |
| 9448 | 27 B1 61 48 8A 91 61 68 82 | 9630 | BD 4C 0C 96 4C 79 00 4C 6F |
| 9450 | AA 88 10 F5 18 A5 61 69 A9 | 9638 | 48 B2 A9 00 A2 0A 95 5D 0E |
| 9458 | 28 85 61 90 02 E6 62 60 94 | 9640 | CA 10 FB 60 EA 20 9E B7 04 |
| 9460 | 38 A5 61 E9 28 85 61 B0 60 | 9648 | E0 08 90 03 4C 48 B2 86 93 |
| 9468 | 02 C6 62 A0 28 B1 61 AA 2E | 9650 | FD 20 FD AE 20 9E B7 A9 BB |
| 9470 | A0 81 B1 61 48 8A 91 61 DC | 9658 | 07 85 FB A9 F8 18 65 FD 1B |
| 9478 | 68 AA C8 C0 29 D0 F3 60 B2 | 9660 | 85 FA A0 00 8A 91 FA 20 FD |
| 9480 | 04 04 04 04 04 04 04 05 FF | 9668 | FD AE 20 9E B7 E0 10 90 29 |
| 9488 | 05 05 05 05 05 06 06 06 FF | 9670 | 03 4C 48 B2 A9 D0 85 FB B1 |
| 9490 | 06 06 06 06 07 07 07 00 00 | 9678 | A9 27 18 65 FD 85 FA A0 06 |
| 9498 | 07 07 00 28 50 78 A0 C8 88 | 9680 | 00 8A 91 FA 20 FD AE 20 C0 |
| 94A0 | F0 18 40 68 90 B8 E0 08 5F | 9688 | 9E B7 E0 02 90 03 4C 48 55 |
| 94A8 | 30 58 80 A8 D0 F8 20 48 61 | 9690 | B2 86 FA AD 1B D0 A6 FD 6D |
| 94B0 | 70 98 C0 E8 A5 2D A4 2E 9E | 9698 | 1D 3B 97 A4 FA C0 01 F0 20 |
| 94B8 | 85 14 84 15 C4 30 D0 02 40 | 96A0 | 03 50 3B 97 8D 1B D0 20 6C |
| 94C0 | C5 2F B0 18 69 02 90 01 22 | 96A8 | FD AE 20 9E B7 E0 02 90 1B |
| 94C8 | C8 85 22 84 23 20 FA 94 4A | 96B0 | 03 4C 48 B2 86 FA AD 1C 6B |
| 94D0 | 20 2E 95 8A 10 07 20 37 EF | 96B8 | D0 A6 FD 1D 3B 97 A4 FA 57 |
| 94D8 | 95 4C E9 94 60 98 30 06 90 | 96C0 | C0 01 F0 03 5D 3B 97 8D D6 |
| 94E0 | 20 47 95 4C E9 94 20 50 47 | 96C8 | 1C D0 20 FD AE 20 9E B7 E5 |
| 94E8 | 95 A9 0D 20 D2 FF A5 14 3D | 96D0 | E0 02 90 03 4C 48 B2 86 99 |
| 94F0 | A4 15 18 69 07 90 C1 C8 AE | 96D8 | FA AD 17 D0 A6 FD 1D 3B 20 |
| 94F8 | B0 BE A0 00 B1 14 AA 29 AF | 96E0 | 97 A4 FA C0 01 F0 03 5D E5 |
| 9500 | 7F 20 D2 FF C8 B1 14 A8 B5 | 96E8 | 3B 97 8D 17 D0 20 FD AE 18 |
| 9508 | 29 7F F0 03 20 D2 FF 8A 5A | 96F0 | 20 9E B7 E0 02 90 03 4C 84 |
| 9510 | 10 11 98 30 0A A9 3E 20 E6 | 96F8 | 48 B2 86 FA AD 1D D0 A6 DC |
| 9518 | D2 FF 68 68 4C E9 94 A9 22 | 9700 | FD 1D 3B 97 A4 FA C0 01 E0 |
| 9520 | 25 D0 4E 98 10 04 A9 24 9D | 9708 | F0 03 5D 3B 97 8D 1D D0 65 |
| 9528 | D0 47 60 20 D2 FF A9 20 24 | 9710 | 60 EA 20 9E B7 E0 08 90 49 |
| 9530 | 20 D2 FF A9 3D D0 3A A0 AC | 9718 | 03 4C 48 B2 AD 15 D0 1D 98 |
| 9538 | 00 B1 22 AA C8 B1 22 A8 59 | 9720 | 3B 97 8D 15 D0 60 EA 20 55 |
| 9540 | 8A 20 95 B3 4C 4A 95 20 C3 | 9728 | 9E B7 E0 08 90 03 4C 48 4F |
| 9548 | A6 BB 20 DD BD 4C 1E AB 13 | 9730 | B2 AD 15 D0 5D 3B 97 8D 76 |
| 9550 | 20 6F 95 A0 02 B1 22 85 95 | 9738 | 15 D0 60 01 02 04 08 10 9B |
| 9558 | 25 88 B1 22 85 24 88 B1 63 | 9740 | 20 40 80 EA 20 9E B7 E0 D0 |
| 9560 | 22 85 26 F0 0A B1 24 20 32 | 9748 | 08 90 03 4C 48 B2 86 FE 4E |
| 9568 | D2 FF C8 C4 26 D0 F6 A9 7A | 9750 | A9 D0 85 FB 85 FD A9 00 95 |
| 9570 | 22 4C D2 FF EA 20 8A AD 50 | 9758 | 18 7D 87 97 85 FA 18 69 C6 |
| 9578 | 20 F7 B7 A5 14 85 FD A5 23 | 9760 | 01 85 FC 20 FD AE 20 EB DC |
| 9580 | 15 85 FE A6 01 86 FC C9 96 | 9768 | B7 A0 00 8A 91 FC A5 14 B4 |
| 9588 | D0 90 07 C9 E0 B0 03 A9 09 | 9770 | 91 FA AD 10 D0 A6 FE 1D 3E |
| 9590 | 31 2C A9 34 78 85 01 A0 CE | 9778 | 3B 97 A4 15 C0 01 F0 03 DE |
| 9598 | 00 B1 FD 85 FA A5 FC 85 92 | 9780 | 5D 3B 97 8D 10 D0 60 00 CD |
| 95A0 | 01 58 A9 00 85 FC 85 FD 63 | 9788 | 02 04 06 08 0A 0C 0E EA 1E |

9790 20 9E B7 E0 10 90 03 4C 92
 9798 48 B2 8E 25 D0 20 FD AE FD
 97A0 20 9E B7 E0 10 90 03 4C 92
 97A8 48 B2 8E 26 D0 60 20 FD 91
 97B0 AE 20 9E B7 8A 48 20 FD DA
 97B8 AE 20 9E B7 8A A8 68 AA 15
 97C0 18 20 F0 FF 20 FD AE 4C 6F
 97C8 A4 AA A9 08 8D 0C C7 60 82
 97D0 6F FF 20 DB 00 DF 20 FF FA
 97D8 00 DF 20 DF 22 DF 02 DF CB
 97E0 00 DF 00 DF 00 DF 07 77 F6
 97E8 00 FF 00 B6 00 FF 00 FF 50
 97F0 00 FF 00 00 06 FF 00 77 93
 97F8 40 77 40 55 02 D6 FF 17 C9
 9800 66 00 66 66 66 66 3E 00 A4
 9808 00 00 3C 06 3E 66 3E 00 4C
 9810 00 60 60 7C 66 66 7C 00 00
 9818 00 00 3C 60 60 60 3C 00 18
 9820 00 06 06 3E 66 66 3E 00 00
 9828 00 00 3C 66 7E 60 3C 00 30
 9830 00 0E 18 3E 18 18 18 00 E5
 9838 00 00 3E 66 66 3E 06 7C 8A
 9840 00 60 60 7C 66 66 66 00 EB
 9848 00 18 00 38 18 18 3C 00 ED
 9850 00 06 00 06 06 06 06 3C BF
 9858 00 60 60 6C 78 6C 66 00 06
 9860 00 38 18 18 18 18 3C 00 04
 9868 00 00 66 7F 7F 6B 63 00 5E
 9870 00 00 7C 66 66 66 66 00 7C
 9878 00 00 3C 66 66 66 3C 00 12
 9880 00 00 7C 66 66 7C 60 60 00
 9888 00 00 3E 66 66 3E 06 06 00
 9890 00 00 7C 66 60 60 60 00 76
 9898 00 00 3E 60 3C 06 7C 00 90
 98A0 00 18 7E 18 18 18 0E 00 5C
 98A8 00 00 66 66 66 66 3E 00 3E
 98B0 00 00 66 66 66 3C 18 00 42
 98B8 00 00 63 6B 7F 3E 36 00 6F
 98C0 00 00 66 3C 18 3C 66 00 6C
 98C8 00 00 7E 0C 18 30 7E 00 D8
 98D0 00 00 66 66 66 3E 0C 78 BC
 98D8 66 3C 66 66 66 66 3C 00 66
 98E0 00 00 7E 0C 18 30 7E 00 D8
 98E8 DB 3C 66 7E 66 66 66 00 ED
 98F0 00 18 3C 7E 18 18 18 18 A7
 98F8 00 10 30 7F 7F 30 10 00 00
 9900 00 00 00 00 00 00 00 00 00
 9908 18 18 18 18 00 00 18 00 18
 9910 66 66 66 00 00 00 00 00 66
 9918 3C 60 38 6C 38 0C 78 00 4C
 9920 18 3E 60 3C 06 7C 18 00 A1
 9928 62 66 0C 18 30 66 46 00 00
 9930 3C 66 3C 38 67 66 3F 00 1A
 9938 00 03 06 0C 18 30 60 00 3F
 9940 0C 18 30 30 30 18 0C 00 18
 9948 30 18 0C 0C 0C 18 30 00 3C
 9950 00 66 3C FF 3C 66 00 00 AF
 9958 3C 66 66 6C 66 66 7C 60 EC
 9960 00 00 00 00 00 18 18 30 D0
 9968 00 00 00 7E 00 00 00 00 83
 9970 00 00 00 00 00 18 18 00 00

9978 00 00 00 7E 00 00 00 00 83
 9980 3C 66 6E 76 66 66 3C 00 0A
 9988 18 18 38 18 18 18 7E 00 9E
 9990 3C 66 06 0C 30 60 7E 00 1E
 9998 3C 66 06 1C 06 66 3C 00 9D
 99A0 06 0E 1E 66 7F 06 06 00 2F
 99A8 7E 60 7C 06 06 66 3C 00 70
 99B0 3C 66 60 7C 66 66 3C 00 F7
 99B8 7E 66 0C 18 18 18 18 00 24
 99C0 3C 66 66 3C 66 66 3C 00 3C
 99C8 3C 66 66 3E 06 66 3C 00 DB
 99D0 66 00 3C 66 66 66 3C 00 78
 99D8 36 00 3C 06 3E 66 3E 00 82
 99E0 00 00 18 00 00 18 18 30 E8
 99E8 00 00 7E 00 7E 00 00 00 FC
 99F0 00 00 18 00 00 18 00 00 00
 99F8 3C 66 06 0C 18 00 18 00 00
 9AA0 00 18 18 7E 18 18 00 00 83
 9AA8 18 3C 66 7E 66 66 66 00 2A
 9A10 7C 66 66 7C 66 66 7C 00 7C
 9A18 3C 66 60 60 60 66 3C 00 0C
 9A20 78 6C 66 66 66 6C 78 00 7E
 9A28 7E 60 60 78 60 60 7E 00 84
 9A30 7E 60 60 78 60 60 60 00 66
 9A38 3C 66 60 6E 66 66 3C 00 04
 9A40 66 66 66 7E 66 66 66 00 4E
 9A48 3C 18 18 18 18 18 3C 00 60
 9A50 1E 0C 0C 0C 0C 6C 38 00 EB
 9A58 66 6C 78 70 78 6C 66 00 74
 9A60 60 60 60 60 60 60 7E 00 7E
 9A68 63 77 7F 6B 63 63 63 00 63
 9A70 66 76 7E 7E 6E 66 66 00 5E
 9A78 3C 66 66 66 66 66 3C 00 12
 9A80 7C 66 66 7C 60 60 60 00 60
 9A88 3C 66 66 66 66 3C 0E 00 0E
 9A90 7C 66 66 7C 78 6C 66 00 72
 9A98 3C 66 60 3C 06 66 3C 00 D7
 9AA0 7E 18 18 18 18 18 18 00 7E
 9AA8 66 66 66 66 66 66 3C 00 3C
 9AB0 66 66 66 66 66 3C 18 00 42
 9AB8 63 63 63 6B 7F 77 63 00 63
 9AC0 66 66 3C 18 3C 66 66 00 60
 9AC8 7E 06 0C 18 30 60 7E 00 BA
 9AD0 66 66 66 3C 18 18 18 00 42
 9ADB 3C 66 06 0C 18 00 18 00 00
 9AE0 C0 C0 30 30 C0 C0 30 30 00
 9AEB 66 66 66 3C 18 18 18 00 42
 9AF0 33 33 CC CC 33 33 CC CC 00
 9AF8 00 00 18 00 00 18 00 00 00
 9B00 00 00 00 00 00 66 66 3C C4
 9B08 F0 F0 F0 F0 F0 F0 F0 F0 00
 9B10 00 00 00 00 FF FF FF FF 00
 9B18 C3 99 9F C3 F9 99 C3 FF 29
 9B20 00 00 00 18 00 00 00 00 E9
 9B28 C0 C0 C0 C0 C0 C0 C0 C0 00
 9B30 CC CC 33 33 CC CC 33 33 00
 9B38 03 03 03 03 03 03 03 03 00
 9B40 00 00 66 66 66 3E 0C 78 BC
 9B48 7E 06 0C 18 30 60 7E 00 BA
 9B50 03 03 03 03 03 03 03 03 00
 9B58 18 18 18 1F 1F 18 18 18 00

| | |
|---------------------------------|---------------------------------|
| 9F30 33 33 CC CC 33 33 CC CC 00 | 9F98 E7 E7 E7 07 07 E7 E7 E7 00 |
| 9F38 FC FC FC FC FC FC FC FC 00 | 9FA0 3F 3F 3F 3F 3F 3F 3F 00 |
| 9F40 FF FF 99 99 99 C1 F3 87 44 | 9FAB 1F 1F 1F 1F 1F 1F 1F 00 |
| 9F48 81 F9 F3 E7 CF 9F 81 FF 46 | 9FB0 F8 F8 F8 F8 F8 F8 F8 F8 00 |
| 9F50 FC FC FC FC FC FC FC FC 00 | 9FB8 00 00 FF FF FF FF FF FF 00 |
| 9F58 E7 E7 E7 E0 E0 E7 E7 E7 00 | 9FC0 00 00 00 FF FF FF FF FF 01 |
| 9F60 FF FF FF FF F0 F0 F0 F0 00 | 9FC8 FF FF FF FF FF 00 00 00 FF |
| 9F68 E7 E7 E7 E0 E0 FF FF FF E8 | 9FD0 E7 8D 99 99 99 99 C3 FF EE |
| 9F70 FF FF FF 07 07 E7 E7 E7 18 | 9FD8 FF FF FF FF 0F 0F 0F 0F 00 |
| 9F78 FF FF FF FF FF FF 00 00 00 | 9FE0 F0 F0 F0 F0 FF FF FF FF 00 |
| 9F80 FF FF FF E0 E0 E7 E7 E7 18 | 9FE8 E7 E7 E7 07 07 FF FF FF E8 |
| 9F88 E7 E7 E7 00 00 FF FF FF E8 | 9FF0 0F 0F 53 3A 54 54 54 54 19 |
| 9F90 FF FF FF 00 00 E7 E7 E7 18 | 9FF8 54 2C 46 52 41 4E 4B 0D 4D |

Listing 3: Basic-Lader Teil 1 ▼

```

90 FOR I=50176 TO 53247:READ A:S=S+A:POKE I,A:NEXT          3552
91 IF S<>363048 THEN PRINT "[CLR]FEHLER IN DATAS!":GOTO 500  3814
92 SYS 57812"G-BASIC TEIL 1",8                               2094
93 POKE 193,000:POKE 194,196                                  1854
94 POKE 174,255:POKE 175,207                                  2046
95 SYS 62954:NEW                                              1107
96 REM
100 FOR I=50176 TO 53247:READ A:S=S+A:POKE I,A:NEXT        3552
101 IF S<>363048 THEN PRINT "FEHLER IN DATAS!":END           3369
102 DATA 076,158,203,076,088,204,076,150,201,076,066,196,076,090,196,076  7204
103 DATA 132,196,076,027,198,076,024,198,076,043,198,076,040,198,076,032  7460
104 DATA 201,076,029,201,076,233,199,076,230,199,076,098,198,076,095,198  7778
105 DATA 076,110,201,076,043,202,076,008,203,076,056,202,076,246,206,076  6977
106 DATA 243,206,234,169,160,133,252,169,000,133,251,162,032,160,000,145  7726
107 DATA 251,200,208,251,230,252,202,208,244,096,032,158,183,142,240,207  6875
108 DATA 162,003,169,140,133,252,160,000,132,251,132,151,173,240,207,145  7043
109 DATA 251,200,196,151,208,249,230,252,202,240,003,016,242,096,162,232  7061
110 DATA 134,151,208,235,032,158,183,142,240,207,096,032,235,183,138,168  7563
111 DATA 166,021,192,200,176,013,165,020,224,001,144,006,208,005,201,064  7142
112 DATA 176,001,096,076,072,178,140,253,207,141,252,207,142,254,207,133  7732
113 DATA 020,134,021,152,074,074,074,170,189,065,198,133,173,138,041,003  7806
114 DATA 170,189,091,198,133,172,152,041,007,024,101,172,133,172,165,020  7316
115 DATA 041,248,133,099,169,160,005,173,133,173,024,165,172,101,099,133  8035
116 DATA 172,165,173,101,021,133,173,165,020,041,007,073,007,170,189,057  8026
117 DATA 198,133,171,096,072,173,254,207,074,173,252,207,106,074,074,133  8181
118 DATA 111,104,072,056,237,252,207,072,138,237,254,207,133,108,176,010  7211
119 DATA 104,073,255,105,001,072,169,000,229,108,133,106,133,110,104,133  7259
120 DATA 105,133,109,104,141,252,207,142,254,207,152,024,237,253,207,144  7260
121 DATA 004,073,255,105,254,133,107,140,253,207,102,108,056,229,105,170  7699
122 DATA 169,255,229,106,133,112,164,111,176,005,010,032,199,197,056,165  7659
123 DATA 109,101,107,133,109,165,110,233,000,133,110,132,111,032,220,197  8096
124 DATA 232,208,005,230,112,208,001,096,165,108,176,222,032,152,197,024  6949
125 DATA 165,109,101,105,133,109,165,110,101,106,080,221,165,172,041,007  7273
126 DATA 201,007,240,005,056,169,000,176,004,169,056,230,173,101,172,133  7807
127 DATA 172,169,000,101,173,133,173,096,048,226,165,172,041,007,240,005  7564
128 DATA 024,169,255,144,004,169,199,198,173,101,172,133,172,165,173,233  8021
129 DATA 000,133,173,096,070,171,144,014,102,171,165,172,200,024,105,008  7479
130 DATA 133,172,144,002,230,173,096,016,235,006,171,144,014,038,171,165  7192
131 DATA 172,136,056,233,009,133,172,176,002,198,173,096,120,160,053,132  7761
132 DATA 001,160,000,008,073,255,165,171,036,151,048,005,073,255,049,172  7686
133 DATA 044,017,172,145,172,169,055,133,001,088,165,172,133,251,165,173  7888
134 DATA 074,102,251,074,102,251,074,102,251,041,003,009,140,133,252,173  7433
135 DATA 240,207,145,251,040,164,111,036,162,000,044,162,128,134,151,032  7230
136 DATA 139,196,032,166,196,076,220,197,162,000,044,162,128,032,029,198  7266
137 DATA 032,253,174,032,139,196,076,244,136,001,002,004,008,016,032,064  7056
138 DATA 128,000,001,002,003,005,006,007,008,010,011,012,013,015,016,017  6753
139 DATA 018,020,021,022,023,025,026,027,028,030,031,000,064,128,192,162  7329
140 DATA 000,044,162,128,134,151,032,158,183,224,000,240,007,224,001,240  6912

```

| | | | |
|-----|------|---|------|
| 141 | DATA | 011,076,072,178,169,025,141,231,003,076,129,198,169,023,141,231 | 8079 |
| 142 | DATA | 003,032,253,174,032,139,196,140,253,207,141,252,207,142,254,207 | 7402 |
| 143 | DATA | 140,071,003,141,069,003,142,070,003,169,019,032,210,255,032,253 | 7778 |
| 144 | DATA | 174,032,158,173,036,013,048,006,032,221,189,032,135,180,032,166 | 7621 |
| 145 | DATA | 182,141,135,003,141,136,003,032,036,171,160,000,177,034,201,032 | 7451 |
| 146 | DATA | 144,017,201,129,144,004,201,161,144,009,200,204,136,003,208,236 | 7207 |
| 147 | DATA | 076,217,198,206,135,003,076,202,198,169,000,141,134,003,169,000 | 7320 |
| 148 | DATA | 141,061,003,174,134,003,189,000,004,236,135,003,208,003,076,229 | 7549 |
| 149 | DATA | 199,141,060,003,014,060,003,046,061,003,014,060,003,046,061,003 | 6795 |
| 150 | DATA | 014,060,003,046,061,003,169,000,141,062,003,169,208,141,063,003 | 6994 |
| 151 | DATA | 169,000,141,064,003,141,065,003,173,231,003,041,002,201,000,240 | 7160 |
| 152 | DATA | 005,169,008,141,065,003,160,000,132,250,024,173,060,003,109,062 | 7350 |
| 153 | DATA | 003,141,066,003,173,061,003,109,063,003,141,067,003,024,173,066 | 7351 |
| 154 | DATA | 003,109,064,003,141,066,003,173,067,003,109,065,003,141,067,003 | 7508 |
| 155 | DATA | 024,173,066,003,101,250,133,253,173,067,003,105,000,133,254,120 | 7762 |
| 156 | DATA | 169,049,133,001,160,000,177,253,141,132,003,169,055,133,001,088 | 7425 |
| 157 | DATA | 162,008,173,252,207,024,105,001,141,252,207,173,254,207,105,000 | 7116 |
| 158 | DATA | 141,254,207,173,132,003,061,056,198,221,056,198,208,024,138,072 | 7555 |
| 159 | DATA | 152,072,024,172,253,207,173,252,207,174,254,207,032,166,196,032 | 7597 |
| 160 | DATA | 220,197,104,168,104,170,202,224,001,176,199,230,250,238,253,207 | 7705 |
| 161 | DATA | 173,069,003,141,252,207,173,070,003,141,254,207,164,250,192,008 | 7326 |
| 162 | DATA | 144,142,024,173,252,207,105,008,141,252,207,141,069,003,173,254 | 7209 |
| 163 | DATA | 207,105,000,141,254,207,141,070,003,238,134,003,173,071,003,141 | 7535 |
| 164 | DATA | 253,207,076,222,198,096,169,000,044,169,128,133,151,032,139,196 | 7622 |
| 165 | DATA | 140,167,003,141,165,003,142,166,003,032,253,174,032,138,173,160 | 7514 |
| 166 | DATA | 003,162,121,032,212,187,169,000,133,175,032,253,174,032,138,173 | 7713 |
| 167 | DATA | 160,003,162,126,032,212,187,169,245,160,226,032,162,187,169,209 | 7538 |
| 168 | DATA | 160,185,032,015,187,162,136,160,003,032,212,187,169,032,141,132 | 7498 |
| 169 | DATA | 003,032,253,174,032,138,173,160,003,162,131,032,212,187,032,253 | 7502 |
| 170 | DATA | 174,032,138,173,160,003,162,141,032,212,187,169,131,160,003,032 | 7648 |
| 171 | DATA | 162,187,032,100,226,169,121,160,003,032,040,186,169,017,160,191 | 7277 |
| 172 | DATA | 032,103,184,032,155,188,024,165,101,109,165,003,133,020,165,100 | 7175 |
| 173 | DATA | 109,166,003,133,021,169,131,160,003,032,162,187,032,107,226,169 | 7738 |
| 174 | DATA | 126,160,003,032,040,186,169,017,160,191,032,103,184,032,155,188 | 7245 |
| 175 | DATA | 024,165,101,109,167,003,168,165,020,166,021,141,063,003,142,064 | 7451 |
| 176 | DATA | 003,140,065,003,032,166,196,032,220,197,165,175,201,000,240,015 | 7272 |
| 177 | DATA | 173,060,003,174,061,003,172,062,003,032,244,196,076,193,200,230 | 7367 |
| 178 | DATA | 175,173,063,003,141,060,003,173,064,003,141,061,003,173,065,003 | 7836 |
| 179 | DATA | 141,062,003,169,131,160,003,032,162,187,169,136,160,003,032,103 | 7412 |
| 180 | DATA | 184,162,131,160,003,032,212,187,169,141,160,003,032,080,184,032 | 7183 |
| 181 | DATA | 043,188,048,003,076,075,200,169,141,160,003,032,162,187,162,131 | 7651 |
| 182 | DATA | 160,003,032,212,187,169,018,141,189,200,169,201,141,190,200,076 | 7371 |
| 183 | DATA | 075,200,169,193,141,189,200,169,200,141,190,200,096,162,000,044 | 7125 |
| 184 | DATA | 162,129,134,151,032,139,196,141,060,003,142,061,003,140,062,003 | 7441 |
| 185 | DATA | 032,253,174,032,139,196,204,062,003,144,048,141,063,003,142,064 | 7199 |
| 186 | DATA | 003,140,065,003,173,060,003,174,061,003,172,062,003,032,166,196 | 7076 |
| 187 | DATA | 032,220,197,173,063,003,174,064,003,172,062,003,032,244,196,238 | 7561 |
| 188 | DATA | 062,003,173,062,003,205,065,003,208,218,096,076,072,178,169,000 | 7278 |
| 189 | DATA | 162,160,133,251,134,252,162,032,120,169,053,133,001,160,000,177 | 7797 |
| 190 | DATA | 251,073,052,145,251,200,208,247,230,252,202,208,240,169,055,133 | 7564 |
| 191 | DATA | 001,098,096,076,088,204,234,032,158,183,224,000,240,245,224,001 | 7040 |
| 192 | DATA | 240,007,224,002,240,021,076,072,178,169,059,141,017,208,169,056 | 7339 |
| 193 | DATA | 141,024,208,169,149,141,000,221,076,202,201,169,059,141,017,208 | 7578 |
| 194 | DATA | 169,008,141,024,208,169,000,141,000,221,032,253,174,032,158,183 | 7553 |
| 195 | DATA | 224,001,240,007,224,002,240,043,076,072,178,169,160,141,068,196 | 7214 |
| 196 | DATA | 141,213,196,141,113,201,141,048,202,141,112,203,169,140,141,099 | 7197 |
| 197 | DATA | 196,141,012,198,169,240,141,094,196,141,109,196,141,136,196,141 | 7211 |
| 198 | DATA | 016,198,096,163,224,141,068,196,141,213,196,141,113,201,141,048 | 7371 |
| 199 | DATA | 202,141,112,203,169,192,141,099,196,141,012,198,169,241,141,094 | 7234 |
| 200 | DATA | 196,141,109,196,141,136,196,141,016,198,096,234,032,212,225,160 | 7451 |
| 201 | DATA | 160,162,000,169,000,076,213,255,169,001,162,004,160,000,032,186 | 6992 |
| 202 | DATA | 255,169,000,032,189,255,032,192,255,162,001,032,201,255,169,255 | 7760 |
| 203 | DATA | 133,097,169,007,133,253,169,028,133,151,169,000,141,060,003,169 | 7486 |
| 204 | DATA | 040,141,061,003,162,004,189,003,203,032,210,255,202,016,247,169 | 7224 |
| 205 | DATA | 000,133,099,133,100,173,060,003,133,101,169,000,133,254,165,099 | 7715 |
| 206 | DATA | 166,100,164,101,032,166,196,160,000,120,169,053,133,001,177,172 | 7356 |

| | | |
|----------|---|------|
| 207 DATA | 133,250,169,055,133,001,000,165,250,166,254,157,064,003,230,101 | 7509 |
| 208 DATA | 232,134,254,228,253,208,215,169,000,160,007,166,253,030,064,003 | 7436 |
| 209 DATA | 042,202,016,249,037,097,009,128,032,210,255,136,016,237,165,099 | 7660 |
| 210 DATA | 024,105,008,133,099,144,002,230,100,206,061,003,203,167,169,013 | 7578 |
| 211 DATA | 032,210,255,173,060,003,024,105,007,141,060,003,198,151,240,003 | 7438 |
| 212 DATA | 076,095,202,169,004,197,253,240,012,133,253,169,001,133,151,169 | 8020 |
| 213 DATA | 015,133,097,208,235,169,015,032,210,255,032,204,255,169,001,032 | 7725 |
| 214 DATA | 195,255,096,080,000,016,027,008,234,032,158,173,036,013,048,006 | 7238 |
| 215 DATA | 032,221,189,032,135,180,032,166,182,133,253,169,003,133,180,133 | 8162 |
| 216 DATA | 252,169,060,133,187,133,251,160,000,177,034,145,251,200,196,253 | 8012 |
| 217 DATA | 208,247,169,044,145,251,200,169,080,145,251,200,169,044,145,251 | 7347 |
| 218 DATA | 200,169,087,145,251,200,132,183,032,253,174,032,158,183,224,008 | 7338 |
| 219 DATA | 240,007,224,001,240,003,076,072,178,169,001,160,002,032,186,255 | 7511 |
| 220 DATA | 032,192,255,162,001,032,201,255,169,000,133,251,032,210,255,169 | 7394 |
| 221 DATA | 160,133,252,032,210,255,162,032,160,000,120,169,053,133,001,177 | 7867 |
| 222 DATA | 251,133,253,169,055,133,001,088,165,253,032,210,255,200,208,234 | 7750 |
| 223 DATA | 230,252,202,208,227,032,204,255,169,001,032,195,255,036,173,017 | 7317 |
| 224 DATA | 208,141,233,207,173,024,208,141,234,207,173,000,221,141,235,207 | 7269 |
| 225 DATA | 169,109,141,004,003,169,204,141,005,003,169,014,141,032,208,141 | 7248 |
| 226 DATA | 033,208,169,048,141,006,003,169,205,141,007,003,032,091,206,162 | 7042 |
| 227 DATA | 002,160,003,134,214,132,211,032,108,229,169,010,160,204,032,030 | 7428 |
| 228 DATA | 171,169,134,141,000,003,169,207,141,001,003,162,004,160,004,134 | 7414 |
| 229 DATA | 214,132,211,032,108,229,169,053,160,204,032,030,171,169,016,133 | 7867 |
| 230 DATA | 044,173,000,008,141,000,016,076,209,207,147,031,017,032,042,042 | 7287 |
| 231 DATA | 042,042,032,071,082,065,070,073,075,032,045,032,066,065,063,073 | 7922 |
| 232 DATA | 067,032,032,032,086,069,082,083,073,079,078,032,050,046,050,032 | 7576 |
| 233 DATA | 042,042,042,042,000,040,067,041,032,049,057,056,053,032,032,032 | 6704 |
| 234 DATA | 070,082,065,078,075,032,082,073,069,077,069,078,083,067,072,078 | 7856 |
| 235 DATA | 069,073,068,069,082,017,017,000,173,233,207,141,017,208,173,234 | 7833 |
| 236 DATA | 207,141,024,208,173,235,207,141,000,221,076,068,229,166,122,160 | 7617 |
| 237 DATA | 004,132,015,189,000,002,016,070,201,255,240,062,232,208,244,201 | 7079 |
| 238 DATA | 032,240,055,133,008,201,034,240,085,036,015,112,045,201,063,208 | 7306 |
| 239 DATA | 004,169,153,208,037,201,048,144,004,201,060,144,029,132,113,160 | 7256 |
| 240 DATA | 000,132,011,136,134,122,202,200,232,189,000,002,056,249,158,160 | 7129 |
| 241 DATA | 240,245,201,128,208,047,005,011,164,113,232,200,153,251,001,201 | 7368 |
| 242 DATA | 000,240,056,056,233,058,240,004,201,073,208,002,133,015,056,233 | 7083 |
| 243 DATA | 035,208,160,133,008,189,000,002,240,224,197,008,240,220,200,153 | 7167 |
| 244 DATA | 251,001,232,208,240,166,122,230,011,200,185,157,160,016,250,185 | 7048 |
| 245 DATA | 158,160,208,181,240,015,189,000,002,016,189,153,253,001,198,123 | 7442 |
| 246 DATA | 169,255,133,122,096,160,000,185,101,205,208,002,200,232,189,000 | 7268 |
| 247 DATA | 002,056,249,101,205,240,245,201,128,208,004,005,011,208,153,166 | 6954 |
| 248 DATA | 122,230,011,200,185,100,205,016,250,185,101,205,208,224,240,198 | 7014 |
| 249 DATA | 016,015,036,015,048,011,201,255,240,007,201,204,176,006,076,036 | 7287 |
| 250 DATA | 167,076,243,166,056,233,203,170,132,073,160,255,202,240,008,200 | 7705 |
| 251 DATA | 185,101,205,016,250,048,245,200,185,101,205,048,005,032,071,171 | 7066 |
| 252 DATA | 208,245,076,239,166,087,069,082,078,069,210,071,079,084,069,088 | 7584 |
| 253 DATA | 212,071,077,079,068,197,071,067,076,069,065,210,071,067,079,204 | 8021 |
| 254 DATA | 080,067,079,204,083,080,076,079,212,067,080,076,079,212,083,076 | 7573 |
| 255 DATA | 073,078,197,067,076,073,078,197,083,066,076,079,195,067,066,076 | 8180 |
| 256 DATA | 079,195,083,067,073,210,067,067,073,210,083,084,069,088,212,067 | 7453 |
| 257 DATA | 084,069,088,212,071,073,078,214,071,076,079,065,196,071,083,065 | 8106 |
| 258 DATA | 086,197,071,072,065,082,196,083,070,082,065,077,197,067,070,082 | 7290 |
| 259 DATA | 065,077,197,069,082,082,079,210,068,073,083,203,068,073,210,076 | 7958 |
| 260 DATA | 073,078,214,076,070,073,076,204,076,067,079,204,076,076,079,065 | 7773 |
| 261 DATA | 196,076,083,065,086,197,076,080,082,201,076,072,065,082,196,076 | 7506 |
| 262 DATA | 083,067,082,079,204,090,067,079,080,217,082,069,078,069,215,082 | 7863 |
| 263 DATA | 069,078,085,205,068,085,077,208,082,065,205,077,069,082,071,197 | 7633 |
| 264 DATA | 072,069,088,193,083,080,082,068,069,198,083,080,082,073,206,083 | 7339 |
| 265 DATA | 060,082,079,070,198,083,080,082,083,069,212,083,080,082,067,079 | 7171 |
| 266 DATA | 204,090,071,069,210,090,073,206,090,079,070,198,090,076,079,065 | 7337 |
| 267 DATA | 196,090,083,065,086,197,071,079,070,198,000,169,182,160,206,141 | 7615 |
| 268 DATA | 008,003,140,003,003,096,032,115,000,032,111,206,076,174,167,201 | 7204 |
| 269 DATA | 204,144,004,201,254,144,006,032,121,000,076,237,167,056,233,204 | 7027 |
| 270 DATA | 010,170,139,142,206,072,189,141,206,072,076,115,000,000,048,000 | 7173 |
| 271 DATA | 048,005,196,008,196,011,196,014,196,017,196,020,196,023,196,026 | 7448 |
| 272 DATA | 196,029,196,032,196,035,196,038,196,041,196,044,196,047,196,050 | 7651 |

```

273 DATA 136,053,136,056,136,059,136,062,136,255,143,005,144,002,144,008 7218
274 DATA 144,011,144,014,144,017,144,020,144,023,144,026,144,029,144,032 7485
275 DATA 144,035,144,038,144,041,144,044,144,047,144,050,144,053,144,056 7172
276 DATA 144,059,144,062,144,065,144,068,144,071,144,074,144,077,144,080 7624
277 DATA 144,000,048,162,000,044,162,128,134,151,032,139,196,141,060,003 7219
278 DATA 142,061,003,140,062,003,032,253,174,032,139,196,204,062,003,144 7133
279 DATA 073,141,063,003,142,064,003,140,065,003,173,060,003,174,061,003 7063
280 DATA 172,062,003,032,166,196,032,220,197,173,063,003,174,064,003,172 7519
281 DATA 062,003,032,244,196,173,063,003,174,064,003,172,065,003,032,244 7275
282 DATA 196,173,060,003,174,061,003,172,065,003,032,244,196,173,060,003 7706
283 DATA 174,061,003,172,062,003,032,244,196,096,076,072,178,019,032,032 7432
284 DATA 032,032,032,032,032,032,032,032,032,032,032,032,032,032,032 7431
285 DATA 032,032,032,032,032,032,032,032,032,032,032,032,032,032,032 7431
286 DATA 032,032,032,032,032,000,134,251,165,214,133,252,165,211,133,253 7418
287 DATA 162,001,160,001,134,214,132,211,032,108,229,169,093,160,207,032 7364
288 DATA 030,171,165,252,133,214,165,253,133,211,032,108,229,173,233,207 7747
289 DATA 141,017,208,173,234,207,141,024,208,173,235,207,141,000,221,166 7588
290 DATA 251,138,048,003,076,058,164,076,116,164,002,250,000,255,002,223 7489
291 DATA 169,140,133,056,076,134,227,223,000,223,000,223,004,111,002,191 7731
292 DATA 032,046,111,047,002,006,009,015,000,027,021,199,000,014,050,044 6712
293 DATA 110,097,000,000,000,000,006,012,014,013,000,012,056,199,001,015 6773
300 DATA 35375,33012,26315,30155,29353,32818 4270
301 DATA 32983,34266,27704,29991,26680,24396 4358
400 REM
401 REM
402 REM FEHLERFINDER
403 REM
500 PRINT :PRINT :AA=49920:EZ=86 2072
501 FOR K=1 TO 12:AA=AA+256:EA=AA+255:EZ=EZ+16:LZ=EZ+15:READ CS:GOSUB 600:NEXT 6053
502 END 128
600 S=0:FOR I=AA TO EA:S=S+PEEK (I):NEXT 2675
601 IF S<>CS THEN PRINT "FEHLER IN ZEILE";EZ;"BIS";LZ 4497
602 RETURN 142
    
```

Listing 4: Basic-Lader Teil 2 ▼

```

90 FOR I=36864 TO 40959:READ A:S=S+A:POKE I,A:NEXT 3403
91 IF S<>511790 THEN PRINT "[CLR]FEHLER IN DATAS!":GOTO 500 3678
92 SYS 57812"G-BASIC TEIL 2",8 2110
93 POKE 193,000:POKE 194,144 1722
94 POKE 174,255:POKE 175,159 2114
95 SYS 62954:NEW 1107
96 REM
102 DATA 076,084,144,076,125,144,076,230,144,076,008,145,076,034,145,076 7403
103 DATA 047,145,076,107,146,076,074,146,076,177,151,076,156,146,076,183 7498
104 DATA 147,076,245,146,076,047,146,076,094,147,076,180,148,076,116,149 7338
105 DATA 076,171,149,076,237,149,076,068,150,076,017,151,076,038,151,076 7825
106 DATA 067,151,076,143,151,076,036,147,076,050,147,076,132,146,076,062 7339
107 DATA 147,076,074,147,169,001,162,009,160,015,032,186,255,169,000,032 7264
108 DATA 189,255,032,192,255,162,001,032,198,255,032,207,255,032,210,255 7690
109 DATA 036,144,080,246,032,204,255,169,001,032,195,255,096,169,036,133 7637
110 DATA 251,169,251,133,187,169,000,133,188,169,001,133,183,169,008,133 8629
111 DATA 186,169,096,133,185,032,213,243,165,186,032,180,255,165,185,032 7755
112 DATA 150,255,169,000,133,144,160,003,132,251,032,165,255,133,252,164 7496
113 DATA 144,208,047,032,165,255,164,144,208,040,164,251,136,208,233,166 7182
114 DATA 252,032,205,189,169,032,032,210,255,032,165,255,166,144,208,018 7380
115 DATA 170,240,006,032,210,255,076,201,144,169,013,032,210,255,160,002 7392
116 DATA 208,198,032,066,246,096,169,001,162,008,160,015,032,186,255,169 7044
117 DATA 000,032,189,255,032,192,255,162,001,032,201,255,032,164,170,032 7460
118 DATA 204,255,169,001,032,195,255,096,169,000,133,251,169,004,133,252 7234
119 DATA 170,160,000,177,251,073,128,145,251,200,208,247,230,252,202,208 7395
120 DATA 240,096,169,004,133,254,032,158,183,142,064,003,076,060,145,169 6983
121 DATA 216,133,254,032,158,183,138,041,015,141,064,003,032,253,174,032 7708
122 DATA 158,183,224,025,144,003,076,072,178,142,060,003,032,253,174,032 7485
123 DATA 158,183,224,040,144,003,076,072,178,142,061,003,032,253,174,032 7236
    
```

| | | |
|----------|---|------|
| 124 DATA | 158,183,224,025,144,003,076,072,178,142,062,003,032,253,174,032 | 7501 |
| 125 DATA | 158,183,224,040,144,003,076,072,178,142,063,003,173,060,003,205 | 7112 |
| 126 DATA | 062,003,144,015,208,240,173,061,003,205,063,003,144,005,240,003 | 7244 |
| 127 DATA | 076,072,178,173,063,003,056,237,061,003,024,105,001,141,070,003 | 7702 |
| 128 DATA | 173,062,003,056,237,060,003,024,105,001,141,069,003,169,000,133 | 7421 |
| 129 DATA | 250,133,251,133,252,133,253,160,000,204,060,003,240,019,165,250 | 7657 |
| 130 DATA | 024,105,040,133,250,165,251,105,000,133,251,200,204,060,003,208 | 7449 |
| 131 DATA | 237,024,165,250,109,061,003,133,250,165,251,105,000,133,251,160 | 7823 |
| 132 DATA | 000,204,062,003,240,019,165,252,024,105,040,133,252,165,253,105 | 7422 |
| 133 DATA | 000,133,253,200,204,062,003,208,237,024,165,252,109,063,003,133 | 7493 |
| 134 DATA | 252,165,253,105,000,133,253,024,165,251,101,254,133,251,174,069 | 7817 |
| 135 DATA | 003,160,000,173,064,003,145,250,200,204,070,003,208,245,165,250 | 7259 |
| 136 DATA | 024,105,040,133,250,165,251,105,000,133,251,202,208,227,096,169 | 7961 |
| 137 DATA | 001,160,001,145,043,032,051,165,165,034,105,002,133,045,165,035 | 7415 |
| 138 DATA | 105,000,133,046,032,099,166,076,116,164,165,157,133,253,169,000 | 7148 |
| 139 DATA | 133,157,032,212,225,162,232,160,007,169,000,133,251,169,004,133 | 7905 |
| 140 DATA | 252,169,251,032,216,255,165,253,133,157,096,165,157,133,253,169 | 8273 |
| 141 DATA | 000,133,157,032,212,225,162,000,160,004,169,000,032,213,255,165 | 7441 |
| 142 DATA | 253,133,157,096,169,021,141,024,208,141,234,207,234,169,208,141 | 7763 |
| 143 DATA | 012,199,096,076,164,170,076,164,170,032,253,170,169,004,133,186 | 7386 |
| 144 DATA | 169,126,133,184,169,000,160,004,133,113,132,114,133,183,133,185 | 6918 |
| 145 DATA | 032,192,255,166,184,032,201,255,162,025,169,013,032,210,255,160 | 7520 |
| 146 DATA | 000,177,113,133,103,041,063,006,103,036,103,016,002,009,128,112 | 7241 |
| 147 DATA | 002,009,064,032,210,255,200,192,040,208,230,152,024,101,113,133 | 7211 |
| 148 DATA | 113,144,002,230,114,202,208,210,169,013,032,210,255,032,204,255 | 7298 |
| 149 DATA | 169,126,076,195,255,120,165,001,133,254,041,251,133,001,169,208 | 7428 |
| 150 DATA | 133,251,169,008,133,253,169,000,133,250,133,252,162,008,160,000 | 6861 |
| 151 DATA | 177,250,145,252,200,208,249,230,251,230,253,202,208,240,165,254 | 7217 |
| 152 DATA | 133,001,088,096,169,152,141,255,146,032,245,146,169,208,141,255 | 7281 |
| 153 DATA | 146,096,169,018,141,024,208,141,234,207,076,202,151,096,032,212 | 6989 |
| 154 DATA | 225,160,008,162,000,169,000,076,213,255,032,212,225,160,015,162 | 7458 |
| 155 DATA | 255,169,000,133,250,169,000,133,251,169,250,076,216,255,234,032 | 8208 |
| 156 DATA | 138,173,032,247,183,165,020,133,251,165,021,133,252,032,253,174 | 8229 |
| 157 DATA | 032,138,173,032,247,183,165,043,164,044,133,034,132,035,160,003 | 7401 |
| 158 DATA | 165,252,145,034,136,165,251,145,034,136,177,034,240,037,133,037 | 7914 |
| 159 DATA | 136,177,034,133,036,160,002,024,177,034,101,020,145,036,200,177 | 7876 |
| 160 DATA | 034,101,021,145,036,176,013,165,036,133,034,165,037,133,035,160 | 8035 |
| 161 DATA | 001,208,215,096,076,072,178,234,032,158,183,138,074,008,032,253 | 7243 |
| 162 DATA | 174,032,158,183,224,025,144,003,076,072,178,134,250,032,253,174 | 7730 |
| 163 DATA | 032,158,183,224,025,144,003,076,072,178,134,251,138,166,250,164 | 7352 |
| 164 DATA | 251,056,229,250,176,006,073,255,166,251,164,250,133,254,040,008 | 7075 |
| 165 DATA | 144,003,200,152,170,189,128,148,133,098,189,154,148,133,037,040 | 7092 |
| 166 DATA | 008,144,004,233,001,133,097,165,098,041,003,009,004,040,008,032 | 7598 |
| 167 DATA | 059,148,040,008,165,097,144,010,105,039,133,097,144,012,230,098 | 6991 |
| 168 DATA | 176,008,233,039,133,097,176,002,198,098,165,098,041,003,009,216 | 7087 |
| 169 DATA | 040,008,032,059,148,198,254,016,206,040,096,133,098,144,003,076 | 7060 |
| 170 DATA | 096,148,160,000,177,097,170,160,039,177,097,072,138,145,097,104 | 7377 |
| 171 DATA | 170,136,016,245,024,165,097,105,040,133,097,144,002,230,098,096 | 7689 |
| 172 DATA | 056,165,097,233,040,133,097,176,002,198,098,160,040,177,097,170 | 8256 |
| 173 DATA | 160,001,177,097,072,138,145,097,104,170,200,192,041,208,243,096 | 7194 |
| 174 DATA | 004,004,004,004,004,004,004,005,005,005,005,005,005,006,006,006 | 6361 |
| 175 DATA | 006,006,006,006,007,007,007,007,007,007,000,040,080,120,160,200 | 6609 |
| 176 DATA | 240,024,064,104,144,184,224,008,048,088,128,168,208,248,032,072 | 6721 |
| 177 DATA | 112,152,192,232,165,045,164,046,133,020,132,021,196,048,208,002 | 6965 |
| 178 DATA | 197,047,176,024,105,002,144,001,200,133,034,132,035,032,250,148 | 7250 |
| 179 DATA | 032,046,149,138,016,007,032,055,149,076,233,148,096,152,040,006 | 7047 |
| 180 DATA | 032,071,149,076,233,148,032,080,149,169,013,032,210,255,165,020 | 7470 |
| 181 DATA | 164,021,024,105,007,144,193,200,176,190,160,000,177,020,170,041 | 6971 |
| 182 DATA | 127,032,210,255,200,177,020,168,041,127,240,003,032,210,255,138 | 7634 |
| 183 DATA | 016,017,152,048,010,169,062,032,210,255,104,104,076,233,148,169 | 7654 |
| 184 DATA | 037,208,078,152,016,004,169,036,208,071,096,032,210,255,169,032 | 7560 |
| 185 DATA | 032,210,255,169,061,208,058,160,000,177,034,170,200,177,034,169 | 7715 |
| 186 DATA | 138,032,149,179,076,074,149,032,166,187,032,221,189,076,030,171 | 8189 |
| 187 DATA | 032,111,149,160,002,177,034,133,037,136,177,034,133,036,136,177 | 8161 |
| 188 DATA | 034,133,038,240,010,177,036,032,210,255,200,196,038,208,246,169 | 7691 |
| 189 DATA | 034,076,210,255,234,032,138,173,032,247,183,165,020,133,253,165 | 8116 |

| | | | |
|-----|------|---|------|
| 190 | DATA | 021,133,254,166,001,134,252,201,208,144,007,201,224,176,003,163 | 7658 |
| 191 | DATA | 049,044,169,052,120,133,001,160,000,177,253,133,250,165,252,133 | 7915 |
| 192 | DATA | 001,000,169,000,133,252,133,253,133,254,096,165,043,133,251,165 | 7499 |
| 193 | DATA | 044,133,252,032,212,225,165,045,056,233,002,133,043,165,046,133 | 8410 |
| 194 | DATA | 044,176,002,198,044,166,043,164,044,169,000,032,213,255,134,045 | 7474 |
| 195 | DATA | 132,046,165,251,133,043,165,252,133,044,169,118,160,163,032,030 | 7159 |
| 196 | DATA | 171,032,042,165,032,142,166,032,051,165,076,119,166,169,248,162 | 7782 |
| 197 | DATA | 149,141,010,003,142,011,003,096,169,000,133,013,032,115,000,201 | 7141 |
| 198 | DATA | 036,240,006,032,121,000,076,141,174,032,058,150,032,115,000,144 | 7009 |
| 199 | DATA | 011,201,065,144,031,201,071,176,027,056,233,007,056,233,048,072 | 7472 |
| 200 | DATA | 165,097,240,007,024,105,004,176,014,133,097,104,240,222,032,126 | 7458 |
| 201 | DATA | 189,076,012,150,076,121,000,076,072,178,169,000,162,010,149,093 | 7404 |
| 202 | DATA | 202,016,251,096,234,032,158,183,224,008,144,003,076,072,178,134 | 7158 |
| 203 | DATA | 253,032,253,174,032,158,183,169,007,133,251,169,248,024,101,253 | 7875 |
| 204 | DATA | 133,250,160,000,138,145,250,032,253,174,032,158,183,224,016,144 | 7057 |
| 205 | DATA | 003,076,072,178,169,208,133,251,169,039,024,101,253,133,250,160 | 7825 |
| 206 | DATA | 000,138,145,250,032,253,174,032,158,183,224,002,144,003,076,072 | 7362 |
| 207 | DATA | 178,134,250,173,027,208,166,253,029,059,151,164,250,192,001,240 | 7491 |
| 208 | DATA | 003,093,059,151,141,027,208,032,253,174,032,158,183,224,002,144 | 7452 |
| 209 | DATA | 003,076,072,178,134,250,173,028,208,166,253,029,059,151,164,250 | 7561 |
| 210 | DATA | 192,001,240,003,093,059,151,141,028,208,032,253,174,032,158,183 | 7372 |
| 211 | DATA | 224,002,144,003,076,072,178,134,250,173,023,208,166,253,029,059 | 7564 |
| 212 | DATA | 151,164,250,192,001,240,003,093,059,151,141,023,208,032,253,174 | 7407 |
| 213 | DATA | 032,158,183,224,002,144,003,076,072,178,134,250,173,029,208,166 | 7309 |
| 214 | DATA | 253,029,059,151,164,250,192,001,240,003,093,059,151,141,029,208 | 7402 |
| 215 | DATA | 096,234,032,158,183,224,008,144,003,076,072,178,173,021,208,029 | 7616 |
| 216 | DATA | 059,151,141,021,208,096,234,032,158,183,224,008,144,003,076,072 | 7386 |
| 217 | DATA | 178,173,021,208,093,059,151,141,021,208,096,001,002,004,008,016 | 7088 |
| 218 | DATA | 032,064,128,234,032,158,183,224,008,144,003,076,072,178,134,254 | 7304 |
| 219 | DATA | 169,208,133,251,133,253,169,000,024,125,135,151,133,250,024,105 | 7496 |
| 220 | DATA | 001,133,252,032,253,174,032,235,183,160,000,138,145,252,165,020 | 7775 |
| 221 | DATA | 145,250,173,016,208,166,254,029,059,151,164,021,192,001,240,003 | 7481 |
| 222 | DATA | 093,059,151,141,016,208,096,000,002,004,006,008,010,012,014,234 | 6748 |
| 223 | DATA | 032,158,183,224,016,144,003,076,072,178,142,037,208,032,253,174 | 7517 |
| 224 | DATA | 032,158,183,224,016,144,003,076,072,178,142,038,208,096,032,253 | 7277 |
| 225 | DATA | 174,032,158,183,138,072,032,253,174,032,158,183,138,168,104,170 | 7731 |
| 226 | DATA | 024,032,240,255,032,253,174,076,164,170,169,008,141,012,139,096 | 7374 |
| 227 | DATA | 111,255,032,219,000,223,032,255,000,223,032,223,034,223,002,223 | 8204 |
| 228 | DATA | 000,223,000,223,000,223,007,119,000,255,000,182,000,255,000,255 | 7722 |
| 229 | DATA | 000,255,000,000,006,255,000,119,064,119,064,085,002,214,255,023 | 7396 |
| 230 | DATA | 102,000,102,102,102,102,062,000,000,000,060,006,062,102,062,000 | 5963 |
| 231 | DATA | 000,096,096,124,102,102,124,000,000,000,060,096,096,060,000 | 6341 |
| 232 | DATA | 000,006,006,062,102,102,062,000,000,000,060,102,126,096,060,000 | 6143 |
| 233 | DATA | 000,014,024,062,024,024,024,000,000,000,062,102,102,062,006,124 | 6540 |
| 234 | DATA | 000,096,096,124,102,102,102,000,000,024,000,056,024,024,060,000 | 6439 |
| 235 | DATA | 000,006,000,006,006,006,006,060,000,096,096,108,120,108,102,000 | 6150 |
| 236 | DATA | 000,056,024,024,024,024,060,000,000,000,102,127,127,107,099,000 | 6741 |
| 237 | DATA | 000,000,124,102,102,102,102,000,000,000,060,102,102,102,060,000 | 5939 |
| 238 | DATA | 000,000,124,102,102,124,096,096,000,000,062,102,102,062,006,006 | 6343 |
| 239 | DATA | 000,000,124,102,096,096,096,000,000,000,062,096,060,006,124,000 | 6153 |
| 240 | DATA | 000,024,126,024,024,024,014,000,000,000,102,102,102,102,062,000 | 6329 |
| 241 | DATA | 000,000,102,102,102,060,024,000,000,000,099,107,127,062,054,000 | 6420 |
| 242 | DATA | 000,000,102,060,024,060,102,000,000,000,126,012,024,048,126,000 | 6223 |
| 243 | DATA | 000,000,102,102,102,062,012,120,102,060,102,102,102,102,060,000 | 6315 |
| 244 | DATA | 000,000,126,012,024,048,126,000,219,060,102,126,102,102,102,000 | 6348 |
| 245 | DATA | 000,024,060,126,024,024,024,024,000,016,048,127,127,048,016,000 | 6881 |
| 246 | DATA | 000,000,000,000,000,000,000,000,024,024,024,024,000,000,024,000 | 6043 |
| 247 | DATA | 102,102,102,000,000,000,000,000,060,096,056,108,056,012,120,000 | 6072 |
| 248 | DATA | 024,062,096,060,006,124,024,000,098,102,012,024,048,102,070,000 | 6720 |
| 249 | DATA | 060,102,060,056,103,102,063,000,000,003,006,012,024,048,096,000 | 6290 |
| 250 | DATA | 012,024,048,048,048,024,012,000,048,024,012,012,012,024,048,000 | 6653 |
| 251 | DATA | 000,102,060,255,060,102,000,000,060,102,102,108,102,102,124,096 | 6309 |
| 252 | DATA | 000,000,000,000,024,024,048,000,000,000,126,000,000,000,000 | 6001 |
| 253 | DATA | 000,000,000,000,000,024,024,000,000,000,126,000,000,000,000 | 5996 |
| 254 | DATA | 060,102,110,110,102,102,060,000,024,024,056,024,024,024,126,000 | 6497 |
| 255 | DATA | 060,102,006,012,048,096,126,000,060,102,006,028,006,102,060,000 | 6382 |

| | | | |
|-----|------|---|------|
| 256 | DATA | 006,014,030,102,127,006,006,000,126,096,124,006,006,102,060,000 | 6216 |
| 257 | DATA | 060,102,096,124,102,102,060,000,126,102,012,024,024,024,024,000 | 6492 |
| 258 | DATA | 060,102,102,060,102,102,060,000,060,102,102,062,006,102,060,000 | 6252 |
| 259 | DATA | 102,000,060,102,102,102,060,000,054,000,060,006,062,102,062,000 | 6005 |
| 260 | DATA | 000,000,024,000,000,024,024,048,000,000,126,000,126,000,000,000 | 5994 |
| 261 | DATA | 000,000,024,000,000,024,000,000,060,102,006,012,024,000,024,000 | 6083 |
| 262 | DATA | 000,024,024,126,024,024,000,000,024,060,102,126,102,102,102,000 | 6585 |
| 263 | DATA | 124,102,102,124,102,102,124,000,060,102,096,096,096,102,060,000 | 6362 |
| 264 | DATA | 120,108,102,102,102,108,120,000,126,096,096,120,096,096,126,000 | 6406 |
| 265 | DATA | 126,096,096,120,096,096,096,000,060,102,096,110,102,102,060,000 | 6583 |
| 266 | DATA | 102,102,102,126,102,102,102,000,060,024,024,024,024,024,060,000 | 6482 |
| 267 | DATA | 030,012,012,012,012,108,056,000,102,108,120,112,120,108,102,000 | 6174 |
| 268 | DATA | 096,096,096,096,096,096,126,000,099,119,127,107,099,099,099,000 | 7159 |
| 269 | DATA | 102,118,126,126,110,102,102,000,060,102,102,102,102,102,060,000 | 6242 |
| 270 | DATA | 124,102,102,124,096,096,096,000,060,102,102,102,102,060,014,000 | 6446 |
| 271 | DATA | 124,102,102,124,120,108,102,000,060,102,096,060,006,102,060,000 | 6357 |
| 272 | DATA | 126,024,024,024,024,024,024,000,102,102,102,102,102,102,060,000 | 6408 |
| 273 | DATA | 102,102,102,102,102,060,024,000,099,099,099,107,127,119,099,000 | 6931 |
| 274 | DATA | 102,102,060,024,060,102,102,000,126,006,012,024,048,096,126,000 | 6361 |
| 275 | DATA | 102,102,102,060,024,024,024,000,060,102,006,012,024,000,024,000 | 6321 |
| 276 | DATA | 192,192,048,048,192,192,048,048,102,102,102,060,024,024,024,000 | 6547 |
| 277 | DATA | 051,051,204,204,051,051,204,204,000,000,024,000,000,024,000,000 | 6479 |
| 278 | DATA | 000,000,000,000,000,102,102,060,240,240,240,240,240,240,240,240 | 6183 |
| 279 | DATA | 000,000,000,000,255,255,255,255,195,153,159,195,249,153,195,255 | 7590 |
| 280 | DATA | 000,000,000,024,000,000,000,000,192,192,192,192,192,192,192,192 | 6405 |
| 281 | DATA | 204,204,051,051,204,204,051,051,003,003,003,003,003,003,003,003 | 6921 |
| 282 | DATA | 000,000,102,102,102,062,012,120,126,006,012,024,048,096,126,000 | 6460 |
| 283 | DATA | 003,003,003,003,003,003,003,003,024,024,031,031,024,024,024 | 7125 |
| 284 | DATA | 000,000,000,000,015,015,015,015,024,024,024,031,031,000,000,000 | 6689 |
| 285 | DATA | 000,000,000,248,248,024,024,024,000,000,000,000,000,000,255,255 | 6439 |
| 286 | DATA | 000,000,000,031,031,024,024,024,024,024,024,255,255,000,000,000 | 6836 |
| 287 | DATA | 000,000,000,255,255,024,024,024,024,024,024,248,248,024,024,024 | 6949 |
| 288 | DATA | 192,192,192,192,192,192,192,192,224,224,224,224,224,224,224,224 | 7227 |
| 289 | DATA | 007,007,007,007,007,007,007,007,255,255,000,000,000,000,000,000 | 6734 |
| 290 | DATA | 255,255,255,000,000,000,000,000,000,000,000,000,255,255,255 | 6632 |
| 291 | DATA | 024,066,102,102,102,102,060,000,000,000,000,000,240,240,240,240 | 6131 |
| 292 | DATA | 015,015,015,015,000,000,000,000,024,024,024,248,248,000,000,000 | 6502 |
| 293 | DATA | 240,240,240,240,000,000,000,000,240,240,240,240,015,015,015,015 | 6479 |
| 294 | DATA | 153,255,153,153,153,153,193,255,255,255,195,249,193,153,193,255 | 8040 |
| 295 | DATA | 255,159,159,131,153,153,131,255,255,255,195,159,159,159,195,255 | 8245 |
| 296 | DATA | 255,249,249,193,153,153,193,255,255,255,195,153,129,159,195,255 | 8111 |
| 297 | DATA | 255,241,231,193,231,231,231,255,255,255,193,153,153,193,249,131 | 8190 |
| 298 | DATA | 255,159,159,131,153,153,153,255,255,231,255,199,231,231,195,255 | 8466 |
| 299 | DATA | 255,249,255,249,249,249,249,195,255,159,159,147,135,147,153,255 | 7868 |
| 300 | DATA | 255,199,231,231,231,231,195,255,255,255,153,128,128,148,156,255 | 8083 |
| 301 | DATA | 255,255,131,153,153,153,153,255,255,255,195,153,153,153,195,255 | 8047 |
| 302 | DATA | 255,255,131,153,153,131,159,159,255,255,193,153,153,193,249,249 | 8128 |
| 303 | DATA | 255,255,131,153,159,159,255,255,255,255,193,159,195,249,131,255 | 8080 |
| 304 | DATA | 255,231,129,231,231,231,241,255,255,255,153,153,153,153,193,255 | 8460 |
| 305 | DATA | 255,255,153,153,153,195,231,255,255,255,156,148,128,193,201,255 | 7817 |
| 306 | DATA | 255,255,153,195,231,195,153,255,255,255,129,243,231,207,129,255 | 7976 |
| 307 | DATA | 255,255,153,153,153,193,243,135,153,195,153,153,153,153,195,255 | 8084 |
| 308 | DATA | 255,255,129,243,231,207,129,255,036,195,153,129,153,153,153,255 | 8025 |
| 309 | DATA | 255,231,195,129,231,231,231,231,255,239,207,128,128,207,239,255 | 8589 |
| 310 | DATA | 255,255,255,255,255,255,255,255,255,231,231,231,231,255,231,255 | 8417 |
| 311 | DATA | 153,153,153,255,255,255,255,255,195,159,199,147,199,243,135,255 | 8084 |
| 312 | DATA | 231,193,159,195,249,131,231,255,157,153,243,231,207,153,185,255 | 8314 |
| 313 | DATA | 195,153,195,199,152,153,192,255,249,243,231,255,255,255,255,255 | 8067 |
| 314 | DATA | 243,231,207,207,207,231,243,255,207,231,243,243,243,231,207,255 | 8522 |
| 315 | DATA | 255,153,195,000,195,153,255,255,195,153,153,147,153,153,131,159 | 7695 |
| 316 | DATA | 255,255,255,255,255,231,231,207,255,255,255,129,255,255,255,255 | 8301 |
| 317 | DATA | 255,255,255,255,255,231,231,255,255,255,255,129,255,255,255,255 | 8365 |
| 318 | DATA | 195,153,145,137,153,153,195,255,231,231,199,231,231,231,129,255 | 8640 |
| 319 | DATA | 195,153,249,243,207,159,129,255,195,153,249,227,249,153,195,255 | 8257 |
| 320 | DATA | 249,241,225,153,128,249,249,255,129,159,131,249,249,153,195,255 | 8033 |
| 321 | DATA | 195,153,159,131,153,153,195,255,129,153,243,231,231,231,255 | 8449 |

```

322 DATA 195,153,153,195,153,153,195,255,195,153,153,193,249,153,195,255 8082
323 DATA 153,255,195,153,153,195,255,201,255,195,249,193,153,193,255 8036
324 DATA 255,255,231,255,255,231,231,207,255,255,129,255,129,255,255,255 8264
325 DATA 255,255,231,255,255,231,255,255,195,153,249,243,231,255,231,255 8196
326 DATA 255,231,231,129,231,231,255,255,231,195,153,129,153,153,153,255 8325
327 DATA 131,153,153,131,153,153,131,255,195,153,159,159,159,153,195,255 8146
328 DATA 135,147,153,153,153,147,135,255,129,159,159,135,159,159,129,255 8099
329 DATA 129,159,159,135,159,159,159,255,195,153,159,145,153,153,195,255 8268
330 DATA 153,153,153,129,153,153,153,255,195,231,231,231,231,231,195,255 8465
331 DATA 225,243,243,243,243,147,199,255,153,147,135,143,135,147,153,255 7673
332 DATA 159,159,159,159,159,159,129,255,156,136,128,148,156,156,156,255 7880
333 DATA 153,137,129,129,145,153,153,255,195,153,153,153,153,153,195,255 8212
334 DATA 131,153,153,131,159,159,159,255,195,153,153,153,153,195,241,255 8122
335 DATA 131,153,153,131,135,147,153,255,195,153,159,195,249,153,195,255 8113
336 DATA 129,231,231,231,231,231,231,255,153,153,153,153,153,153,195,255 8381
337 DATA 153,153,153,153,153,195,231,255,156,156,156,148,128,136,156,255 7671
338 DATA 153,153,195,231,195,153,153,255,129,249,243,231,207,159,129,255 8343
339 DATA 153,153,153,195,231,231,231,255,195,153,249,243,231,255,231,255 8056
340 DATA 063,063,207,207,063,063,207,207,153,153,153,195,231,231,231,255 8022
341 DATA 204,204,051,051,204,204,051,051,255,255,231,255,255,231,255,255 7686
342 DATA 255,255,255,255,255,153,153,195,015,015,015,015,015,015,015,015 7825
343 DATA 255,255,255,255,000,000,000,000,060,102,096,060,006,102,060,000 6684
344 DATA 255,255,255,231,255,255,255,255,063,063,063,063,063,063,063,063 8304
345 DATA 051,051,204,204,051,051,204,204,252,252,252,252,252,252,252,252 6989
346 DATA 255,255,153,153,153,193,243,135,129,249,243,231,207,159,129,255 8266
347 DATA 252,252,252,252,252,252,252,252,231,231,231,224,224,231,231,231 7805
348 DATA 255,255,255,255,240,240,240,240,240,231,231,231,224,224,255,255 7731
349 DATA 255,255,255,007,007,231,231,231,255,255,255,255,255,255,000,000 7998
350 DATA 255,255,255,224,224,231,231,231,231,231,231,231,000,000,255,255,255 8094
351 DATA 255,255,255,000,000,231,231,231,231,231,231,231,007,007,231,231,231 8253
352 DATA 063,063,063,063,063,063,063,063,063,031,031,031,031,031,031,031 8247
353 DATA 248,248,248,248,248,248,248,248,000,000,255,255,255,255,255,255 7244
354 DATA 000,000,000,255,255,255,255,255,255,255,255,255,255,000,000,000 7278
355 DATA 231,189,153,153,153,153,195,255,255,255,255,255,015,015,015,015 7856
356 DATA 240,240,240,240,255,255,255,255,231,231,231,007,007,255,255,255 7680
357 DATA 015,015,083,058,084,084,084,084,084,084,044,070,082,065,078,075,050 6654
360 DATA 33332,31122,33959,35115,24810,31557,31818,29831
361 DATA 13634,10855,18924,21290,51646,54596,46356,42945 5548
400 REM 5444
401 REM
402 REM FEHLERFINDER
403 REM
500 PRINT :PRINT :AA=36608:EZ=86 2246
501 FOR K=1 TO 16:AA=AA+256:EA=AA+255:EZ=EZ+16: LZ=EZ+15:READ CS:GOSUB 6054 6054
600: NEXT
502 END 128
600 S=0:FOR I=AA TO EA:S=S+PEEK (I):NEXT 2675
601 IF S<>CS THEN PRINT "FEHLER IN ZEILE";EZ;"BIS";LZ 4497
602 RETURN 142

```

Listing 5: Autostart-Erzeuger ▼

```

10 REM *****
11 REM *** AUTOSTART-LADER FUER GRAFIC-BASIC ***
12 REM ***
13 REM *** VON FRANK RIEMENSCHNEIDER ***
14 REM *****
15 REM
16 FOR I=828 TO 978:READ A:S=S+A:POKE I,A:NEXT 2538
17 IF S<>13303 THEN PRINT "[2DOWN]FEHLER IN DATAS 2699
!":END
18 POKE 770,060:POKE 771,003 1724
92 SYS 57812"LADER G-BASIC",8 2125
93 POKE 193,002:POKE 194,003 1656

```



```

94 POKE 174,211:POKE 175,003                1911
95 SYS 62954                                  884
96 POKE 770,131:POKE 771,164:NEW            2269
102 DATA 169,143,160,003,032,030,171,169,001,162 4708
103 DATA 008,160,001,032,186,255,169,014,162,181 4707
104 DATA 160,003,032,189,255,169,000,133,157,032 4902
105 DATA 213,255,169,014,162,196,160,003,032,189 4519
106 DATA 255,169,000,032,213,255,169,131,141,002 4890
107 DATA 003,169,164,141,003,003,169,078,141,119 4810
108 DATA 002,169,069,141,120,002,169,087,141,121 4694
109 DATA 002,169,013,141,122,002,169,005,133,198 4465
110 DATA 076,000,196,017,017,071,082,065,070,073 4967
111 DATA 067,045,066,065,083,073,067,032,086,032 5038
112 DATA 050,046,050,032,087,073,082,068,032,071 4749
113 DATA 069,076,065,068,069,078,032,046,046,046 4454
114 DATA 000,071,045,066,065,083,073,067,032,084 4620
115 DATA 069,073,076,032,049,000,071,045,066,065 4845
116 DATA 083,073,067,032,084,069,073,076,032,050 4889
117 DATA 000,000,000,000,000,000,000,000,000,000 3572
    
```

Listing 6: Änderungen für Datensette ▼

```

17 IF S<>13296 THEN PRINT "[2DOWN]FEHLER IN DATAS 2770
    !":END
92 SYS 57812"LADER G-BASIC",1                2187
103 DATA 001,160,001,032,186,255,169,014,162,181 4651
    
```

Listing 7: Zeichensatzgenerator ▼

```

100 REM *****
110 REM ** PROGRAMM ZUR DEFINITION EINES **
120 REM ** INDIVIDUELLEN ZEICHENSATZES **
130 REM ** MIT G R A F I C - B A S I C **
140 REM **
141 REM ** VON FRANK RIEMENSCHNEIDER **
143 REM *****
156 REM
157 REM *** MASCHINENPROGRAMM LESEN ***
158 REM
160 FOR I=828 TO 838:READ A:POKE I,A:NEXT
170 DATA 165,207,201,00,208,250,169,1,133,204,96
180 ZCOPY:Z IN
181 REM
182 REM *** ZEICHEN EINGEBEN ***
183 REM
190 PRINT"[CLR]":GOSUB 580
200 LPRI 8,12,"ZEICHEN: ";:POKE 204,0
210 POKE 211,21:POKE 214,8:SYS 58732
220 GET A$:IF A$="" THEN 220
230 SYS 828:IF A$="+ " THEN 740
240 PRINT A$:A=PEEK(1365):GOSUB 620
250 X=0:Y=0
251 REM
252 REM *** CURSOR SETZEN ***
253 REM
260 X1=X+13:Y1=Y+12:POKE 211,X1:POKE 214,Y1:SYS 58732:POKE 204,0
262 REM
    
```

```

263 REM *** EINGABESCHLEIFE ***
264 REM
270 GETI$: IF I$="" THEN 270
280 SYS828: IF I$(">") [RIGHT] THEN 320
281 REM
282 REM *** CURSOR RECHTS ***
283 REM
290 IF X<7 THEN X=X+1:GOTO 260
300 IF Y<7 THEN X=0:Y=Y+1:GOTO 260
310 GOTO 250
320 IF I$("<") [LEFT] THEN 360
321 REM
322 REM *** CURSOR LINKS ***
323 REM
330 IF X>0 THEN X=X-1:GOTO 260
340 IF Y>0 THEN Y=Y-1:X=7:GOTO 260
350 GOTO 250
360 IF I$(">") [DOWN] THEN 390
361 REM
362 REM *** CURSOR RUNTER ***
363 REM
370 IF Y<7 THEN Y=Y+1
380 GOTO 260
390 IF I$("<") [UP] THEN 420
391 REM
392 REM *** CURSOR RAUF ***
393 REM
400 IF Y>0 THEN Y=Y-1
410 GOTO 260
420 IF I$("<") [CLR] THEN 440
421 REM
422 REM *** ARBEITSFELD LOESCHEN ***
423 REM
430 GOSUB 580:GOTO 250
440 IF I$("<") [HOME] THEN 460
441 REM
442 REM *** CURSOR HOME ***
443 REM
450 GOTO 250
460 IF I$(">") CHR$(148) THEN 490
461 REM
462 REM *** ZEICHEN EINFUEGEN ***
463 REM
470 FOR U=1024+Y1*40+20 TO 1024+Y1*40+X1STEP-1
480 POKE U,PEEK(U-1):NEXT U:POKE 1024+Y1*40+X1,46:GOTO 260
490 IF I$(">") CHR$(20) THEN 520
491 REM
492 REM *** ZEICHEN LOESCHEN ***
493 REM
500 FOR U=1024+Y1*40+X1 TO 1024+Y1*40+20
510 POKE U,PEEK(U+1):NEXT U:POKE 1024+Y1*40+20,46:GOTO 260
520 IF I$("<") "*" THEN 540
521 REM
522 REM *** STERN SETZEN ***
523 REM
530 LPRINT Y1,X1,"*":GOTO 290
540 IF I$("<") "." THEN 560

```

```

541 REM
542 REM *** PUNKT SETZEN ***
543 REM
550 LPRI Y1,X1,".":GOTO290
560 IF I$(<)CHR$(13)THEN#60
561 REM
562 REM *** ARBEITSFELD FERTIG ***
563 REM
570 GOSUB680:GOTO190
579 REM
580 REM *** ARBEITSFELD AUFBAUEN ***
581 REM
590 LPRI 11,12," [ ] "
600 FORQ=12TO19:LPRI Q,12," |.....|":NEXTQ
610 LPRI 20,12," [ ] ":RETURN
611 REM
620 REM *** ZEICHENBYTES LESEN ***
621 REM
630 FORU=0TO7:B=PEEK(2048+8*A+U)
640 FORU1=7TO0STEP-1:B1= BAND2↑U1
650 IFB1=2↑U1THEN:LPRI 12+U,13+7-U1,"*":GOTO670
660 LPRI 12+U,13+7-U1,"."
670 NEXTU1,U:RETURN
671 REM
680 REM *** ARBEITSFELD AUSLESEN ***
681 REM
690 BA=2048+8*A.
700 FORU=0TO7:D=0:FORU1=7TO0STEP-1
710 IFPEEK(1517+U*40+(7-U1))=42THEND=D+2↑U1
720 NEXTU1:POKEBA+U,D:NXTU:RETURN
739 REM
740 REM *** PROGRAMM BEENDEN ***
741 REM
750 PRINT"[CLR][DOWN]ZEICHENSATZ SPEICHERN (J/N)"
760 GETA$:IFA$(<)"J"ANDA$(<)"N"THEN760
770 IFA$="N"THEN#20
780 INPUT"[2DOWN]FILENAME :";F$
790 INPUT"[2DOWN]AUSGEBEGERAET (1=KASSETTE,8=FLOPPY):";G
800 IFG(<)&ANDG(<)&8THENPRINT"[4UP]":GOTO790
810 ZSAVE F$,G
820 PRINT"[2DOWN]":END
    
```

**Modul 90 = +144 KB ROM
(bis 288 KB erweiterbar)**

Im 64'er eingebaut = 14 normale Eprom.
Gesamter Adress- und Datenbus voll
gepuffert.
Steckplätze für 4 verschiedene Betriebs-
systeme.
Menügesteuert - Programme auf Tasten-
druck.
6 Super-Programierhilfen im Preis ent-
halten.
Preis incl. Disk. u. Anleitung DM 298,-
Modul 90 Einbauservice nur DM 50,-

KELLERMANN Computer-Shop
Konrad-Adenauer-Platz 10, 4030 Ratingen-4
☎ 02102/33918

C-64 POWER

Weltneuheit !!!

z.B.
fabrikneue
Eproms 2764
nur
DM 7,50

Modul - Master

Ein absolutes Muß für jeden, der Eprom
benutzt.
MM ist ein Programmgenerator für Eproms
- beliebig viele Programme auf ein Eprom
- frei erstellbares Menü (wie DATAMAT)
- geeignet für alle Epromkarten
- auch schaltb., wie z.B. Modul 90, DELA etc.
Auch für Profis, da so optimale Speicheraus-
nutzung "von Hand" kaum möglich.
Preis incl. aut. Handbuch DM 99,-

Modul - 6

- 32 KB, softwaremäßig schaltbare Super-
Epromkarte.
Für bis zu 128 Block langen Programmen.
Preis incl. Software auf Diskette DM 55,-

Betriebssystem-, abtastfrei f. C-64+1541
mit Mode-Control nur DM 39,-, div. Eprom-
Karten ab DM 10,-, Kühltrubine DM 55,-
Steckplatzerw. ab DM 148,-, Reset ab DM 5,-
div. Floppy-Beschleunigungen ab DM 49,-
Turbotape II im Geh. DM 49,-, Eprommer

**hochauflösender
Lightpen**

Sofort einsatzbereit - läuft mit allen Lightpen-
programmen.
Preis incl. Anleitung DM 29,-
Programmpaket auf Steckmodul nur DM 49,-
Komplettpreis Lightpen + Modul DM 100,-

COMPUTERTECHNIK
Z. Zaporowski
Vincenzstr. 4, 5800 Hagen-i. Westf.
☎ 02331/4144

GRATISKATALOG anfordern

Computergrafik für Bilanzen

Zwei unterschiedlich große Tortenstücke erkennt man auf einen Blick. Bei Zahlen muß man schon genauer hinschauen. Im Lehrbuch wie in der Konzernbilanz ist deshalb die Tortengrafik so beliebt. Das Programm „Business Grafik“ zeichnet Tortendiagramme und andere Schaubilder, die das Haushaltsbudget wie die Jahresbilanz eines Multis aussehen lassen. „Business Grafik“ ist für den C64 mit Simon's Basic geschrieben. Sechs verschiedene Darstellungsformen stehen zur Verfügung (siehe Abbildungen):

- Balken + — maximal 24 Werte
- Balken 3D maximal 12 Werte
- Pareto % maximal 12 Werte
- Münzen maximal 6 Werte
- Linien + — maximal 24 Werte
- Tortengrafik maximal 6 Werte

Alle Diagramme können als Hardcopy ausgedruckt werden.

Die Druckroutinen wurden für den EPSON RX 80 mit Görlitz-Interface und für den STAR SG-10C geschrieben, können jedoch leicht für andere Drucker — Interfaces — angepaßt werden (Unterprogramm 3940 — 4300).

Beim Start des Programms fordert eine Laufschrift auf, über die Funktionstasten eine Darstellungsart auszuwählen.

Mittelwert und

Prozente

Im folgenden Unterprogramm „Eingaberoutine“ werden die Anzahl der Posten, der Titel der Grafik, die Werte und Namen der Posten (Werte maximal fünfstellig), die Postendimension (DM, Stk., etc.) sowie mögliche Anzeigen des Mittelwertes, der Summe und des Prozentgehaltes abgefragt. In der Wertaufzählung kann ein Kommentar vorgegeben werden. Dann wird die Anzahl der gewünschten Ausdrücke abgefragt.

Bilanz ohne Bilder ist nicht denkbar. Doch früher mußten Grafiker zeichnen, was heute der Homecomputer schafft.

Nach Bestätigung der Kontrollabfrage: „Drucker ein — Papier auf Blattanfang“ wird das Diagramm auf dem Bildschirm gezeichnet, wobei die Grafik sehr flexibel an eine optimale Größe, abhängig von der Werteanzahl, angepaßt wird. Natürlich wird auch die Werteskala umgerechnet und angeglichen.

Bei den Unterprogrammen „Balken 3D“ und „DM-Münzen“, die

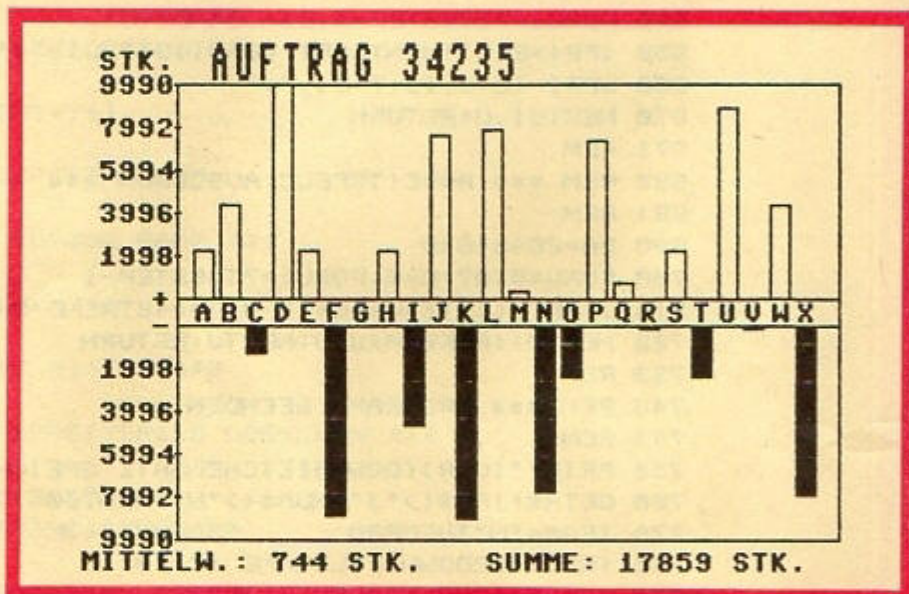


Bild 1

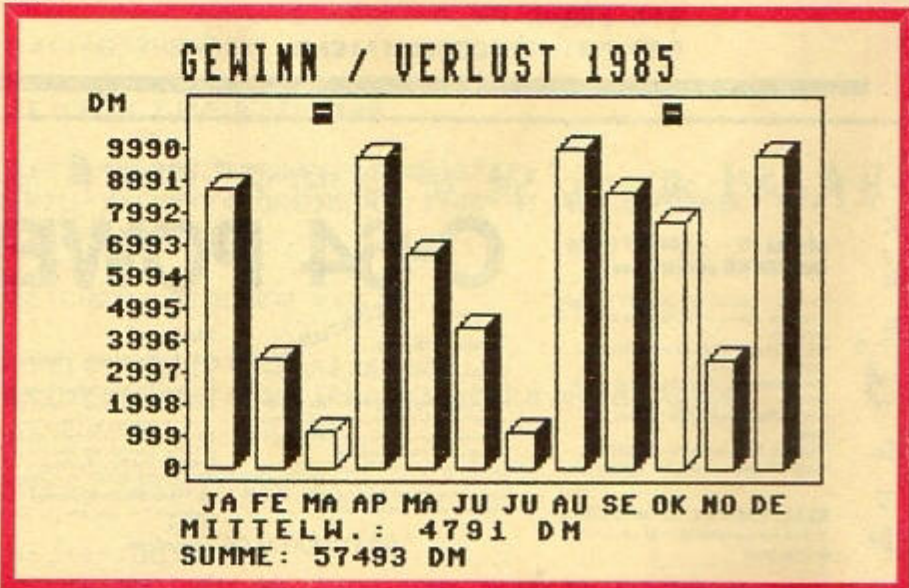


Bild 2

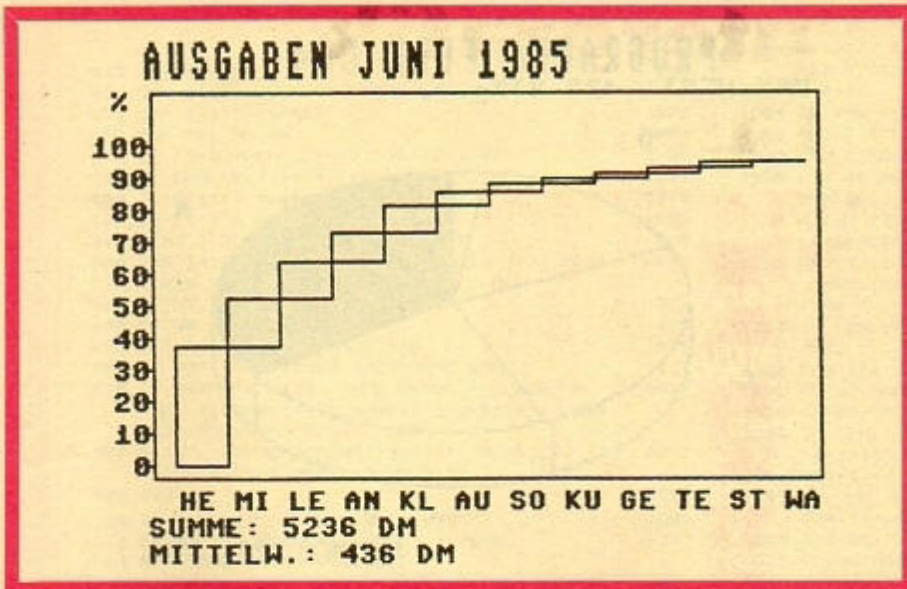


Bild 3

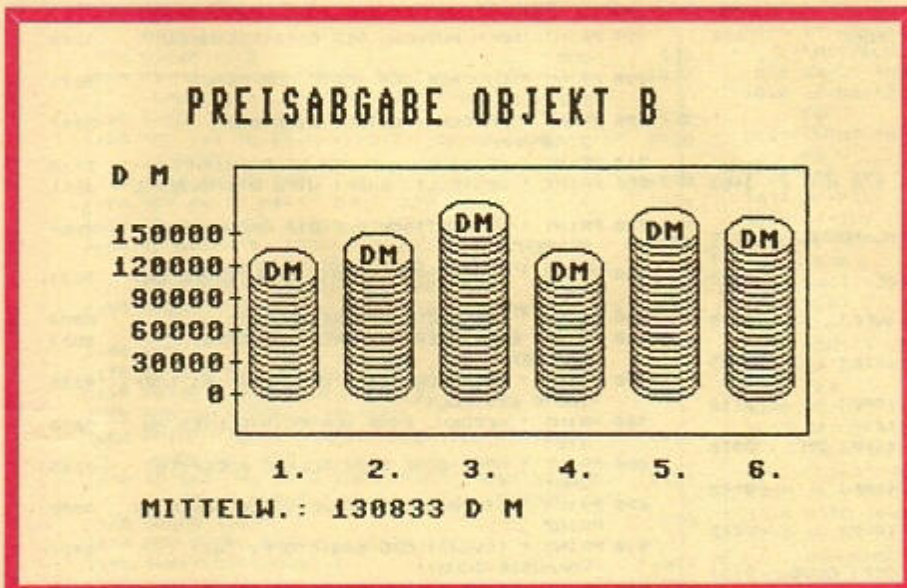


Bild 4

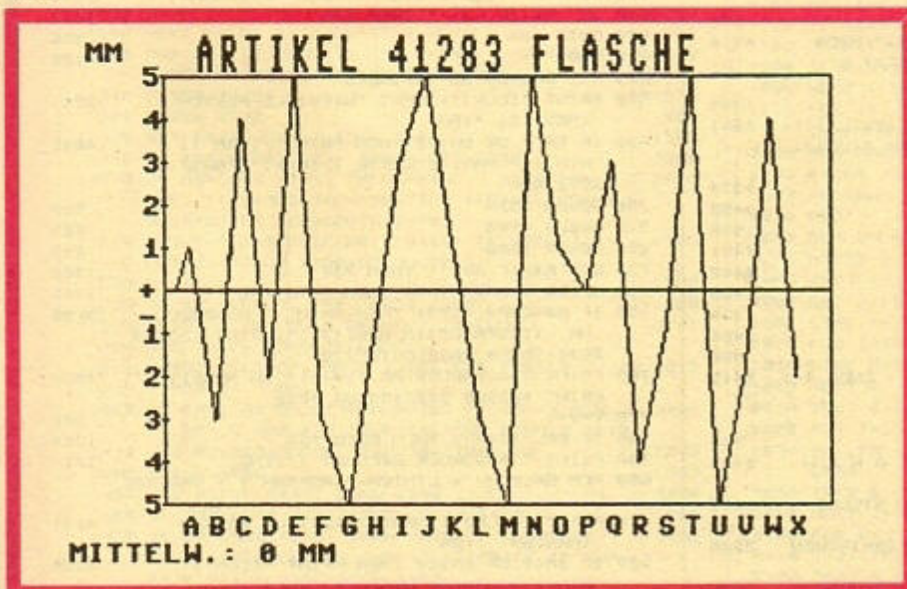


Bild 5

vorwiegend für positive Werte gedacht sind, werden negative Werte mit einem reversen Minus in der Grafik gekennzeichnet. Beim Balken-Diagramm wird zusätzlich der Paint-Befehl (Schwärzung der seitlichen Balkenseite) bei negativen Werten unterdrückt. Das Münzdiagramm baut sich durch die Anzahl der Arc-Befehle als einzige Grafik zeitintensiv auf. Hier ist etwas Geduld erforderlich.

Tortenstücke statt

Zahlen

Bei der Tortengrafik wird der Max-Wert angezeigt, und im Diagramm der „Belag“ des Tortenstückes durch den Paint-Befehl ausgefüllt. Sind mehrere Max-Werte vorhanden, werden sie nur ausgefüllt, wenn sie nicht direkt angrenzend nebeneinander liegen. Torten-Grafiken mit mehr als fünf oder sechs Werten sind unübersichtlich. Daher wurde dieses Diagramm auf maximal sechs Werte beschränkt. Diese Grafik-Form eignet sich besonders zur Darstellung von Teilmengen (Vergleich mit Gesamtmenge).

Die Pareto-Darstellung (schrittweise Kumulation der Werte vom größten zum kleinsten Wert) ist vor allem für die Darstellung von Schwerpunkten gedacht, wie zum Beispiel Fehleranteile, Zeiträume etc. (ABC-Analyse und Teilmengen).

Ist die Grafik zufriedenstellend ausgefallen, wird mit der F1-Taste der Druckvorgang ausgelöst. Es wird eine Hardcopy des Bildschirms erstellt und im Anschluß daran eine Werteauflistung mit allen Parametern ausgegeben. Der Prozentanteil der einzelnen Posten wird wahlweise angezeigt. Ebenso kann ein Kommentar (Überschrift) vorangestellt werden. Nach Druckende wird ein Papiervorschub auf den nächsten Blattanfang ausgelöst. Je nach Wahl werden weitere Kopien der gleichen Grafik ausgedruckt.

Um das Listing übersichtlich zu ►

| | | | |
|--|------|--|------|
| GOTO 810 | | | |
| 830 GOSUB 1520 | 599 | 1480 PRINT " AUSDRUCK NR.":V+1;" IN ARBEIT !": | 5017 |
| 840 REM STAPELGRÖSSE OPTIMIEREN | | PRINT :GOSUB 3930:GOSUB 4170 | |
| 850 G=0:T=200000:SS=1 | 1976 | 1490 V=V+1 | 390 |
| 860 FOR I=1 TO BA | 840 | 1500 IF VAL (C#)>V THEN GOTO 1400 | 1398 |
| 870 IF SS=1 THEN G=P(I):T=P(I):SS=2 | 2790 | 1510 PRINT " AUSDRUCK BEENDET !":END | 1777 |
| 880 IF SGN (P(I))=-1 THEN LET P(I)=P(I)*-1 | 2531 | 1520 REM EINGABE-ROUTINE | |
| 890 IF P(I)<T THEN LET T=P(I) | 1910 | 1530 DIM BB(BA),Z*(BA),N(BA),P(BA),B(BA), | 6014 |
| 900 IF P(I)>G THEN LET G=P(I) | 2000 | BB*(BA),J*(BA):A=12:H=100:HI=155:C=6:X=37 | |
| 910 NEXT I:REM G= GRÖSSTER WERT P | 500 | 1540 PRINT :INPUT " WELCHER TITEL FUER GRAFIK | 3593 |
| 920 FOR I=1 TO BA:N(I)=INT (80+BB(I)/G):NEXT | 2647 | [12SPACES]":T# | |
| I:REM MASS-STAB BALKEN OPTIMIEREN | | 1550 PRINT :INPUT "POSTEN-DIMENSION | 5350 |
| 930 F=INT (G/5) | 972 | (MAX.4STELLEN !)":P#:IF LEN (P#)>4 THEN | |
| 940 GOSUB 3090 | 639 | 1550 | |
| 950 GET A#:IF A#="" THEN 950 | 1516 | 1555 IF LEN (P#)<4 THEN P#=P#+CHR# (32):GOTO | 3099 |
| 960 IF A#<>CHR# (133) THEN GOTO 950 | 1677 | 1555 | |
| 970 IF A#<>CHR# (133) THEN PRINT " AUSDRUCK | 5917 | 1560 FOR I=1 TO BA:O=0+I:PRINT O".": | 2033 |
| NR. [12SPACES]IN ARBEIT !":PRINT :GOSUB | | 1570 INPUT "WERT ":BB(I):PRINT "[UP]" SPC(10): | 2030 |
| 3930:GOSUB 4170:GOTO 930 | | 1580 PRINT "POSTEN-NAME ":Z*(I) | 1934 |
| 980 PRINT " AUSDRUCK NR.":V+1;" IN ARBEIT !": | 5017 | 1590 IF LEN (Z*(I))<2 THEN Z*(I)=CHR# (32)+ | 3561 |
| PRINT :GOSUB 3930:GOSUB 4170 | | Z*(I):GOTO 1590 | |
| 990 V=V+1 | 390 | 1595 BB*(I)=STR# (BB(I)) | 1367 |
| 1000 IF VAL (C#)>V THEN GOTO 800 | 1328 | 1597 IF LEN (BB*(I))<10 THEN BB*(I)=CHR# (32)+ | 3685 |
| 1010 PRINT " AUSDRUCK BEENDET !":END | 1777 | BB*(I):GOTO 1597 | |
| 1020 REM PARETO (%) VARIABLEL | | 1600 S=S+BB(I) | 669 |
| 1030 PRINT "[CLR]":INPUT "WIEVIELE POSTEN | 3211 | 1605 SU=SU+ABS (BB(I)) | 1507 |
| (MAX.12) ":BA | | 1610 P(I)=ABS (BB(I)) | 972 |
| 1040 IF BA<1 OR BA>12 THEN PRINT "[CBM 1] | 4357 | 1620 NEXT I | 276 |
| MIN.1 - MAX.12 WERTE ! [BLK]":PAUSE 2: | | 1623 INPUT "MITTELWERT IN GRAFIK ZEIGEN (J/N)" | 3253 |
| GOTO 1030 | | :F# | |
| 1050 GOSUB 1520 | 599 | 1625 IF F#="J" THEN LET MW=INT (S/BA):M#=" | 5671 |
| 1060 PRINT :PRINT "[RVS] SORTIER - ROUTINE | 3077 | "MITTELW.":STR# (MW)+" "+P# | |
| LAEUFFT ! [OFF]":GOSUB 4400 | | 1630 INPUT "SUMME IN GRAFIK ZEIGEN (J/N)":O# | 3400 |
| 1090 G=SU | 419 | 1630 IF G#="J" THEN S#="SUMME":STR# (S)+" "+ | 3293 |
| 1100 FOR I=1 TO BA:N(I)=INT (120+VAL (BB*(I))/ | 3510 | P# | |
| G):NEXT I | | 1631 INPUT "% ERRECHNEN +IN TAB.DRUCK.(J/N)": | 3533 |
| 1110 F=10:GOSUB 2360 | 1279 | F# | |
| 1120 GET A#:IF A#="" THEN 1120 | 1464 | 1632 FOR I=1 TO BA | 840 |
| 1130 IF A#<>CHR# (133) THEN GOTO 1120 | 1724 | 1633 J1=P(I) | 749 |
| 1140 IF A#<>CHR# (133) THEN PRINT " AUSDRUCK | 5026 | 1635 J1=INT ((J1*100/SU)+.5):J#*STR# (J1) | 3609 |
| NR. [12SPACES]IN ARBEIT !":PRINT :GOSUB | | 1636 IF LEN (J#)<3 THEN J#=CHR# (32)+J#:GOTO | 2648 |
| 3930:GOSUB 4020:GOTO 1160 | | 1636 | |
| 1150 PRINT " AUSDRUCK NR.":V+1;" IN ARBEIT !": | 4023 | 1637 J*(I)=J# | 776 |
| PRINT :GOSUB 3930:GOSUB 4020 | | 1638 NEXT I | 276 |
| 1160 V=V+1 | 390 | 1640 INPUT "KOMMENTAR ZUR TABELLE (J/N) | 3566 |
| 1170 IF VAL (C#)>V THEN GOTO 1150 | 1326 | [SHIFT-SPACE]?"":B# | |
| 1180 PRINT " AUSDRUCK BEENDET !":END | 1777 | 1660 IF B#="J" THEN INPUT "BITTE KOMMENTAR !": | 3116 |
| 1190 REM DM MUENZEN-VARIABLEL | | :K# | |
| 1200 PRINT "[CLR]":INPUT "WIEVIELE POSTEN | 3210 | 1670 PRINT :INPUT "WIEVIEL AUSDRUCKE ":C# | 2162 |
| (MAX.6) ":BA | | 1680 PRINT "[INT]DRUCKER EIN ? PAPIER AUF | 5162 |
| 1210 IF BA<1 OR BA>6 THEN PRINT "[CBM 1] MIN.1 | 5191 | BLATTANFANG ?":INPUT "(2SPACES)(J/N) | |
| - MAX.6 WERTE ! [BLK]":PAUSE 2:GOTO 1200 | | [BLK]":O# | |
| 1220 GOSUB 1520 | 599 | 1700 IF O#="J" THEN 1720 | 1536 |
| 1230 REM STAPELGRÖSSE OPTIMIEREN | | 1710 GOTO 1690 | 647 |
| 1240 G=0:T=200000:SS=1 | 1976 | 1720 PRINT "[CLR]":PRINT :PRINT "[5SPACES] | 3907 |
| 1250 FOR I=1 TO BA | 840 | GEWÜNSCHTE AUSDRUCKE !":PRINT :RETURN | |
| 1260 IF SS=1 THEN G=P(I):T=P(I):SS=2 | 2790 | 1730 END | 120 |
| 1270 IF P(I)<T THEN LET T=P(I) | 1910 | 1740 REM MAX.KLEINSTER + GRÖSSTER WERT | |
| 1280 IF P(I)>G THEN LET G=P(I) | 2000 | 1750 G=0:T=200000:SS=1 | 1920 |
| 1290 NEXT I:REM G= GRÖSSTER WERT P | 503 | 1760 FOR I=1 TO BA | 840 |
| 1300 FOR I=1 TO BA:N(I)=INT (20+P(I)/G):NEXT | 2523 | 1770 IF SS=1 THEN G=P(I):T=P(I):SS=2 | 2790 |
| I:REM MASSSTAB BALKEN OPTIMIEREN | | 1780 IF P(I)<T THEN LET T=P(I) | 1910 |
| 1310 F=INT (G/5) | 972 | 1790 IF P(I)>G THEN LET G=P(I) | 2000 |
| 1320 GOSUB 2740 | 625 | 1800 NEXT I:REM G= GRÖSSTER WERT P | 500 |
| 1330 GET A#:IF A#="" THEN 1330 | 1520 | 1810 FOR I=1 TO BA:N(I)=INT (120+P(I)/G):NEXT | 3061 |
| 1340 IF A#<>CHR# (133) THEN GOTO 1330 | 1980 | I:REM MASSSTAB BALKEN OPTIMIEREN | |
| 1350 IF A#<>CHR# (133) THEN PRINT " AUSDRUCK | 5950 | 1820 REM MASSSTAB-Y-ACHSE AUSRECHNEN | |
| NR. [12SPACES]IN ARBEIT !":PRINT :GOSUB | | 1830 F=INT (G/10) | 800 |
| 3930:GOSUB 4020:GOTO 1370 | | 1840 RETURN | 142 |
| 1360 PRINT " AUSDRUCK NR.":V+1;" IN ARBEIT !": | 4023 | 1850 END | 120 |
| PRINT :GOSUB 3930:GOSUB 4020 | | 1860 REM BALKENDIAGRAMM ZEICHNEN | |
| 1370 V=V+1 | 390 | 1870 HIRES 0,15 | 576 |
| 1380 IF VAL (C#)>V THEN GOTO 1360 | 1358 | 1880 TEXT 47,1,"[CTRL A]"+T#,1,2,10 | 2136 |
| 1390 PRINT " AUSDRUCK BEENDET !":END | 1777 | 1890 REC X+13,H-140,(BA*21)+15,H-15,1 | 3179 |
| 1400 REM TORTEN-DARSTELLUNG | | 1900 LINE X+13,H,X+11,H,1 | 1224 |
| 1410 PRINT "[CLR]":INPUT "WIEVIELE POSTEN | 3210 | 1910 FOR I=12 TO 120 STEP 12 | 1370 |
| (MAX.6) ":BA | | 1950 LINE X+13,H-1,X+11,H-1,1 | 1761 |
| 1420 IF BA<1 OR BA>6 THEN PRINT "[CBM 1] MIN.1 | 5192 | 2000 NEXT I | 276 |
| - MAX.6 WERTE ! [BLK]":PAUSE 2:GOTO 1410 | | 2010 TEXT 0,H-3,"[5SPACES]0".1,0,0 | 1996 |
| 1430 GOSUB 1520:GOSUB 1740:G#*STR# (G) | 2345 | 2020 FOR I=15 TO 123 STEP 12:W=W+F:A#*STR# (W) | 3103 |
| 1440 GOSUB 3000 | 599 | 2030 IF LEN (A#)<6 THEN A#*CHR# (32)+A#:GOTO | 2445 |
| 1450 GET A#:IF A#="" THEN 1450 | 1600 | 2030 | |
| 1460 IF A#<>CHR# (133) THEN GOTO 1450 | 1790 | 2050 TEXT 0,H-1,"[CTRL A]"+A#,1,0,0 | 2045 |
| 1470 IF A#<>CHR# (133) THEN PRINT " AUSDRUCK | 5995 | 2090 NEXT I:TEXT 0,H-140,"[CTRL A]"+P#,1,0,0 | 2430 |
| NR. [12SPACES]IN ARBEIT !":PRINT :GOSUB | | 2100 FOR O=X+19 TO (BA-1)*21:X+19 STEP 21 | 2003 |
| 3930:GOSUB 4170:GOTO 1490 | | 2110 J=J+1:X#*LEFT# (Z*(J),2) | 1797 |
| | | 2120 TEXT 0,H+10,"[CTRL A]"+X#,1,0,0 | 1707 |

| | |
|---|--|
| <pre> 2130 NEXT Q 2140 FOR I=1 TO BA 2150 X=X+21 2160 B=N(I):Y=H-B:IE=X+A:F=H1-B:G=X+C:A:K=X+C 2170 IF B=0 THEN 2240 2180 REC X,Y,A,B,1 2190 LINE X,Y,K,F,1 2200 LINE K,F,G,F,1 2210 LINE G,F,E,Y,1 2220 LINE G,F,G,H1,1 2230 LINE G,H1,E,H,1 2240 NEXT I 2250 I=0 2260 FOR X=72 TO (BA-1)*21+72 STEP 21 2270 I=I+1:B=N(I):IF B=0 THEN 2290 2274 IF SGN (BB(I))=-1 THEN TEXT X-11,H1-132," [CTRL A,RVSI-[OFF]]",1,1,7:GOTO 2290 2280 PAINT X,H1,1 2290 NEXT X 2300 TEXT 47,H+20,"[CTRL A]"+M\$,1,1,10 2310 TEXT 47,H+30,"[CTRL A]"+S\$,1,1,8 2340 RETURN 2350 END 2360 REM PARETO (%) DIAGRAMM ZEICHNEN 2370 HIRES 0,15:K*20 2380 TEXT 30,1,"[CTRL A]"+T\$,1,2,10 2390 REC X+13,H-140,(BA*22)+15,H-15,1 2400 FOR I=12 TO 120 STEP 12 2450 LINE X+13,H-I,X+11,H-I,1 2500 NEXT I 2510 TEXT 0,H-3,"[3SPACES]0",1,0,8 2520 FOR I=15 TO 123 STEP 12:W=W+F:A\$=STR\$(W) 2530 IF LEN (A\$)<4 THEN A\$=CHR\$(32)+A\$:GOTO 2530 2550 TEXT 0,H-1,"[CTRL A]"+A\$,1,0,8 2570 NEXT I:TEXT 0,H-140,"[2SPACES]%",1,0,8 2580 FOR Q=K+23 TO (BA-1)*23+K+23 STEP 23 2590 J=J+1:K\$=LEFT\$(2*(J),2) 2600 TEXT 0,H+10,"[CTRL A]"+K\$,1,0,8 2610 NEXT Q:A=22 2620 FOR I=1 TO BA 2630 X=X+22 2640 B=N(I):Y=H-B:IE=X+A:F=H1-B:G=X+C:A:K=X+C 2650 REC X,Y,A,B,1 2660 H=Y 2670 NEXT I 2680 H=100:A=12:I=0 2690 TEXT 30,H+20,"[CTRL A]"+S\$,1,1,8 2700 TEXT 30,H+30,"[CTRL A]"+M\$,1,1,8 2720 RETURN 2730 END 2740 REM MUENZGRAFIK ZEICHNEN 2750 HIRES 0,15:Y=130:A=0:C=0:K*X=0 2760 FOR X=80 TO ((BA-1)*40)+80 STEP 40 2770 A=A+1:B=N(A):C=Y-(3*B) 2780 CIRCLE X,C,15,0,1 2790 FOR I=1 TO N(A) 2800 ARC X,C+(3*I),90,270,5,15,0,1 2810 NEXT I 2820 TEXT X-0,C-2,"DM",1,0,8 2825 IF SGN (BB(A))=-1 THEN TEXT X-6,C-17," [CTRL A,RVSI-[OFF]]",1,0,8 2830 NEXT X 2840 REC 60,50,(40*BA),100,1 2850 FOR I=3 TO 63 STEP 12 2860 LINE 62,130-I,50,130-I,1 2870 NEXT I 2880 TEXT 0,132,"[6SPACES]0",1,0,8 2890 TEXT 0,50,"[CTRL A] "+P\$,1,0,8 2900 FOR I=1 TO 5 2910 W=W+F:A\$=STR\$(W) 2920 IF LEN (A\$)<7 THEN A\$=CHR\$(32)+A\$:GOTO 2920 2950 TEXT 0,132-(I*12),"[CTRL A]"+A\$,1,0,8 2970 NEXT I 2980 FOR Q=75 TO (BA-1)*40+75 STEP 40 2990 J=J+1:K\$=LEFT\$(2*(J),2) 3000 TEXT 0,160,"[CTRL A]"+K\$,1,0,8 3010 NEXT Q 3020 TEXT 20,175,"[CTRL A]"+M\$,1,1,8 3030 TEXT 20,185,"[CTRL A]"+S\$,1,1,8 3060 TEXT 40,20,"[CTRL A]"+T\$,1,2,10 3070 RETURN 3080 END 3090 REM BALKEN- + LINIEN-DIAGRAMM ZEICHNEN 3100 H=95:K=37 3110 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R, </pre> | <pre> 292 840 944 4444 1088 1044 1168 1066 941 1179 1076 276 366 2257 2021 4805 861 306 2312 2526 142 128 1232 2076 3193 1370 1761 276 2056 3103 2574 2045 2391 2244 1797 1707 787 840 476 4444 1044 274 276 1520 2338 2337 142 128 2791 2499 2055 1310 1100 2743 276 2457 3595 306 1917 1467 2302 276 1724 1823 854 1442 2704 2592 276 2297 1797 1933 292 2306 2052 2039 142 128 1124 5991 S,T,U,V,W,X,Y,Z 3120 IF BA>18 THEN A=8:C=310=0 1130 IF BA>12 AND BA<19 THEN A=10:C=310=1 3140 IF BA>6 AND BA<13 THEN A=16:C=310=4 3150 IF BA<7 THEN A=24:C=310=8 3160 HIRES 0,15 3170 TEXT 55,0,"[CTRL A]"+T\$,1,2,10 3180 FOR I=15 TO 79 STEP 16 3200 LINE X+5,1,X+3,1,1 3220 NEXT I 3230 IF X\$=CHR\$(133) THEN GOTO 3300 3240 FOR I=111 TO 175 STEP 16 3260 LINE X+5,1,X+3,1,1 3280 NEXT I 3290 GOTO 3350 3300 FOR I=121 TO 185 STEP 16 3320 LINE X+5,1,X+3,1,1 3340 NEXT I 3350 IF X\$=CHR\$(133) THEN LINE X+5,105,X+18+ ((A+C)*BA)+(2*0),105,1 3360 LINE X+5,95,X+18+((A+C)*BA)+(2*0),95,1 3370 TEXT X-7,92,"+",1,0,7 3380 TEXT X-7,102,"-",1,0,7 3390 IF X\$=CHR\$(137) THEN REC X+5,15,((A+C)* BA)+14+(2*0),100,1:GOTO 3410 3400 REC X+5,15,((A+C)*BA)+14+(2*0),170,1 3410 K=X+5:J=H:I IF N(I)<1 THEN J=95 3420 FOR I=1 TO BA 3430 X=X+A+C 3440 IF X\$=CHR\$(137) THEN READ O\$:TEXT X+0, 100,"[CTRL A]"+O\$,1,0,7 3450 IF X\$=CHR\$(137) THEN GOSUB 3720:GOTO 3500 3460 READ O\$:TEXT X+0,97,"[CTRL A]"+O\$,1,0,7 3470 B=N(I):IF SGN (B)=-1 THEN LET B=B*-1:GOTO 3760 3480 Y=H-8 3490 REC X,Y,A,B,1 3500 NEXT I 3510 FOR I=76 TO 12 STEP -16:W=W+F:A\$=STR\$(W) 3520 IF LEN (A\$)<5 THEN A\$=CHR\$(32)+A\$:GOTO 3520 3550 TEXT 0,1,"[CTRL A]"+A\$,1,0,8 3560 NEXT I:TEXT 0,3,"[CTRL A] "+P\$,1,0,8 3585 TEXT 0,190,"[CTRL A]"+M\$,1,1,8 3597 TEXT 170,190,"[CTRL A]"+S\$,1,1,8 3598 W=0 3600 IF X\$=CHR\$(133) THEN GOTO 3620 3610 FOR I=100 TO 172 STEP 16:W=W+F:A\$=STR\$(W):GOTO 3630 3620 FOR I=118 TO 182 STEP 16:W=W+F:A\$=STR\$(W) 3630 IF LEN (A\$)<5 THEN A\$=CHR\$(32)+A\$:GOTO 3630 3650 TEXT 0,1,"[CTRL A]"+A\$,1,0,8 3660 NEXT I 3700 RETURN 3710 END 3720 B=N(I):Y=H-B:IF SGN (B)=-1 THEN LET B=B*- 1:Y=95+B 3730 LINE K+0+3,J,X+0+3,Y,1:K=X:J=Y 3740 RETURN 3750 END 3760 Y=105 3770 REC X,Y,A,B,1 3780 PAINT X+1,Y+1,1 3790 GOTO 3500 3795 END 3800 REM TORTENGRAFIK ZEICHNEN 3805 HIRES 0,15 3810 TEXT 40,1,"[CTRL A]"+T\$,1,2,10 3815 TEXT 0,17,"[CTRL A]MAX.WERT: "+G\$+" "+P\$, 1,1,8 3820 CIRCLE 150,100,100,50,1 3830 ARC 150,120,90,270,5,100,50,1 3840 R/360/SU:ANGL 150,100,360,100,50,1 3850 LINE 50,100,50,120,1:LINE 250,100,250, 120,1 3860 FOR I=1 TO BA 3865 Q(I)=ABS (BB(I)) 3870 P=P+Q(I):ANGL 150,100,R*P,100,50,1 3880 M=(R*P/180)*[PI]:N=((R*Q(I)/2)/180)*[PI]: IF (R*P)>90 AND (R*P)<270 THEN GOSUB 4310 3890 READ X\$:TEXT 145-COS ((M-N)+(PI/2))* 130,100-SIN ((M-N)+(PI/2))*80,X\$,1,1,7 3895 IF Q(BA)*ABS (BB(1)) THEN 3910 3900 IF Q(1)=0 AND H<>G THEN PAINT 150-COS </pre> |
|---|--|


```

((M-N)+(P1)/2)*20,100-SIN ((M-N)+(P1)/2)*
20,1
3905 H=0(I) 492
3910 NEXT I 276
3915 TEXT 0,190,"[CTRL A]*M*,1,1,7 1970
3917 TEXT 170,190,"[CTRL A]*S*,1,1,7 2579
3920 RETURN 142
3925 END 120
3930 REM BILDSCHIRM-COPY
3940 OPEN 1,4:PRINT#1,CHR*(27) CHR*(100) 3433
CHR*(7):CLOSE 1:REM LINKER RAND SETZEN
3950 OPEN 1,4 594
3960 OPEN 9,4,9,"7" 994
3970 PRINT#1,CHR*(27)" V" 1245
3980 PRINT#1,CHR*(08) 706
3990 COPY 339
4000 CLOSE 1:CLOSE 9 642
4010 RETURN 142
4015 END 120
4020 REM LISTE DRUCKEN
4030 NRM 306
4040 OPEN 1,4:CMD 1 1012
4050 PRINT :PRINT :PRINT "[8SPACES]WERTE - 3413
AUFLISTUNG : "
4060 PRINT :PRINT K* 527
4070 PRINT :PRINT "ZEICHEN[7SPACES]WERT 3470
[3SPACES]BEZ.[4SPACES]X[3SPACES]POSTEN":
PRINT
4080 IS="#####.##" 1755
4090 FOR I=1 TO BA:J*=J*(I) 1600
4100 X*=LEFT$(Z*(I),2):IF F*(X)"J" THEN 4115 2024
4110 PRINT "[2SPACES]*X*[3SPACES]":USE I*, 3929
BB*(I):PRINT "[2SPACES]*P*[2SPACES]*J*"
[4SPACES]:Z*(I):GOTO 4120
4115 PRINT "[2SPACES]*X*[3SPACES]":USE I*, 3099
BB*(I):PRINT "[2SPACES]*P*[8SPACES]*
Z*(I)
4120 NEXT I 276
4130 PRINT#1:CLOSE 1 506
4140 OPEN 1,4:PRINT#1,CHR*(27) CHR*(12): 2553
CLOSE 1:REM BLATTENDE-VORSCHUB
4150 RETURN 142
4160 END 120
4170 REM LISTE DRUCKEN (A-2)
4180 NRM 306
4190 OPEN 1,4:CMD 1 1012
4200 PRINT :PRINT :PRINT "[8SPACES]WERTE - 3413
AUFLISTUNG : "
4210 PRINT :PRINT K* 527
4220 PRINT :PRINT "ZEICHEN[7SPACES]WERT 3362
[3SPACES]BEZ.[4SPACES]X[4SPACES]POSTEN":
PRINT
4230 RESTORE :IS="#####.##" 1666
4240 FOR I=1 TO BA:J*=J*(I) 1079
4250 READ X*:IF F*(X)"J" THEN 4265 2053
4260 PRINT "[3SPACES]*X*[3SPACES]":USE I*, 4461
BB*(I):PRINT "[2SPACES]*P*[2SPACES]*J*"
[4SPACES]:Z*(I):GOTO 4270
4265 PRINT "[3SPACES]*X*[3SPACES]":USE I*, 3553
BB*(I):PRINT "[2SPACES]*P*[8SPACES]*
Z*(I)
4270 NEXT I 276
4280 PRINT#1:CLOSE 1 506
4290 OPEN 1,4:PRINT#1,CHR*(27) CHR*(12): 2553
CLOSE 1:REM BLATTENDE-VORSCHUB
4300 RETURN 142
4305 END 120
4310 REM ARC TEILEN
4320 MM=(150-COS (M+(P1)/2)*100):NN=(100-SIN 4921
(M+(P1)/2)*50)
4330 LINE MM,NN,MM,NN+20,1 1969
4340 RETURN 142
4345 END 120
4400 REM SORTIER - ROUTINE PARETO
4420 I=1 370
4450 XX=I+1 748
4460 IF BB*(I)BB*(XX) THEN 4480 2119
4470 BB*(0)=BB*(I)+BB*(I)=BB*(XX):BB*(XX)= 3667
BB*(0)
4475 Z*(0)=Z*(I)+Z*(I)+Z*(X)+Z*(X)=Z*(0) 3823
4477 J*(0)=J*(I)+J*(I)+J*(X)+J*(X)=J*(0) 3690
4480 XX=XX+1 954
4490 IF XX=BA THEN 4460 1476
4500 I=I+1 315
4510 IF I>BA THEN 4450 1317
4520 RETURN 142

```

Wenn mal was
nicht funktioniert...

... an Ihrem Commodore Computer

... rufen Sie uns ungeniert -
die Service-Profis von
»Rat und Tat«.

Wir warten und reparieren
schnell · preiswert · gut

 **commodore**

● VC 20 ● C 64

und die Peripherie

● Floppy-Disk ● Drucker

● Datasette



**Technischer
Kundendienst**

Bundesweit - an 70 Standorten:

Augsburg, Tel. (08 21) 46 50 33
 Berlin, Tel. (03 0) 6 84 60 57-59
 Bielefeld, Tel. (05 21) 2 08 04 40
 Bocholt, Tel. (0 29 71) 18 21 95
 Braunschweig, Tel. (05 31) 4 46 71/84 50 99
 Bremen, Tel. (04 21) 41 43 50
 Bremerhaven, Tel. (04 71) 4 91 88
 Celle, Tel. (05 141) 87 67
 Cuxhaven, Tel. (04 721) 5 15 40 / 3 70 33
 Darmstadt, Tel. (0 61 51) 10 92 52
 Deggendorf, Tel. (09 91) 3 32 92
 Dortmund-Holzwickede, Tel. (0 23 01) 8 74 15-16
 Düsseldorf-Eller, Tel. (02 11) 21 30 45 / 22 29 58-59
 Essen-Vogelheim, Tel. (02 01) 3 59 23-27
 Frankfurt/Main, Tel. (0 69) 41 60 11-13
 Freiburg-Gundelfingen, Tel. (07 61) 5 88 01-02
 Fulda-Petersberg, Tel. (06 61) 6 19 10
 Gießen, Tel. (06 41) 59 44-45
 Göttingen, Tel. (05 51) 78 24 40
 Goslar-Baßgeige, Tel. (05 321) 5 05 31/5 03 45
 Hamburg 74, Tel. (0 40) 73 16 65-69
 Hannover 1, Tel. (05 11) 3 10 46 39/32 77 55-56
 Heilbronn, Tel. (0 71 31) 4 49 32
 Hof/Saale, Tel. (09 2 81) 99 41
 Idar-Oberstein, Tel. (0 67 81) 2 78 00
 Ingolstadt, Tel. (08 41) 5 80 80
 Iserlohn, Tel. (0 23 71) 2 41 51
 Kaiserslautern, Tel. (06 31) 85 92 58 / 6 18 12
 Karlsruhe, Tel. (07 21) 13 72 57 / 2 18 21
 Kassel, Tel. (05 61) 7 89 52 51 / 10 31 01
 Kempten, Tel. (08 31) 2 41 10
 Kiel, Tel. (04 31) 68 00 49
 Koblenz-Lützel, Tel. (02 61) 8 20 44-45
 Köln-Rodenkirchen, Tel. (0 22 36) 6 40 56-57
 Landshut, Tel. (08 71) 2 67 60
 Limburg/Lahn, Tel. (0 64 31) 2 57 06
 Lübeck 1, Tel. (04 51) 89 80 40
 Lüneburg, Tel. (0 41 31) 3 66 86
 Mannheim, Tel. (06 21) 1 68 33 30 / 29 14 75
 Memmingen, Tel. (0 83 31) 43 35
 Minden, Tel. (05 71) 2 80 25-26 / 88 12 49
 Mönchengladbach-Rheydt, Tel. (0 21 66) 42 08 80
 München-Eching, Tel. (0 81 65) 7 42 56-57
 Münster, Tel. (02 51) 62 40 10
 Neumünster, Tel. (0 43 21) 4 20 61-68
 Neu-Ulm, Tel. (07 31) 8 40 70
 Nürnberg-Eibach, Tel. (09 11) 21 38 16-18 / 63 20 02
 Passau, Tel. (08 51) 5 21 77
 Pforzheim, Tel. (0 72 31) 2 40 21-22
 Ravensburg, Tel. (07 51) 2 51 16
 Recklinghausen, Tel. (0 23 61) 20 95 51/37 22 79
 Regensburg, Tel. (09 41) 5 34 46
 Rendsburg, Tel. (0 43 31) 20 43 04
 Rosenheim, Tel. (0 80 31) 4 22 05
 Saarbrücken, Tel. (06 81) 3 01 72 78 / 3 70 93
 Siegen/Haiger, Tel. (0 27 73) 24 46
 Singen, Tel. (0 77 31) 6 78 70
 Solingen 1, Tel. (02 12) 20 08 80
 Stuttgart-Leonberg, Tel. (0 71 52) 7 22 38-39
 Trier, Tel. (06 51) 7 32 09
 Villingen, Tel. (0 77 21) 5 41 90
 Wiesbaden-Deikheim, Tel. (0 61 22) 5 22 71-72
 Wilhelmshaven, Tel. (0 44 21) 4 23 99
 Würzburg, Tel. (09 31) 5 02 89

Spezial 1 **RUN 57**

ZED setzt Zeichen

Von deutschen Umlauten bis zu griechischen Buchstaben, alles kann der C64 erzeugen. Doch es kostet Zeit und Mühe: Endlose Zahlenkolonnen sind zu berechnen, bevor man seine Kreationen am Bildschirm bewundern kann. Ein Zeicheneditor erspart diesen langen und fehlerträchtigen Weg. Die neuen Zeichen werden direkt am Bildschirm entworfen. Der Computer macht die Drecksarbeit. So bleibt der Kopf frei für die künstlerische Inspiration.

Doch Zeicheneditor ist nicht gleich Zeicheneditor. Der einfachste stellt nur die Matrix auf dem Bildschirm dar und errechnet die Datas. „ZED“ geht weit darüber hinaus. Mit mehr als 30 Befehlen bietet er außergewöhnlichen Komfort. Er rotiert und kopiert,

Wer seinem C64 einen Satz neue Zeichen verpassen will, hat zwei Möglichkeiten: Er rechnet tagelang — oder benutzt „ZED“.

sendet Floppykommandos und verwaltet zwei unabhängige Zeichensätze — RUN-Programmierer Andreas Wellie hat an alles gedacht.

„ZED“ unterteilt den Bildschirm in mehrere Bereiche (siehe Bild);

— Das Zeichensatzfeld im oberen

Teil enthält den kompletten Zeichensatz in acht Reihen zu je 32 Zeichen.

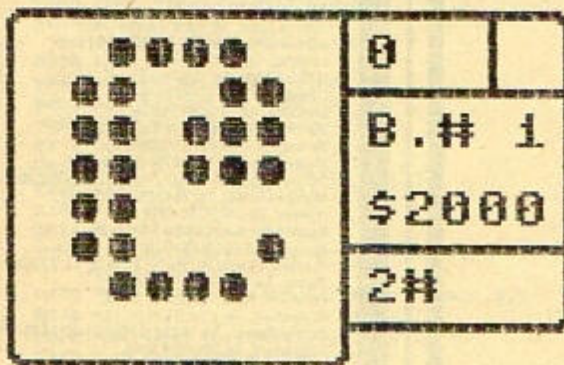
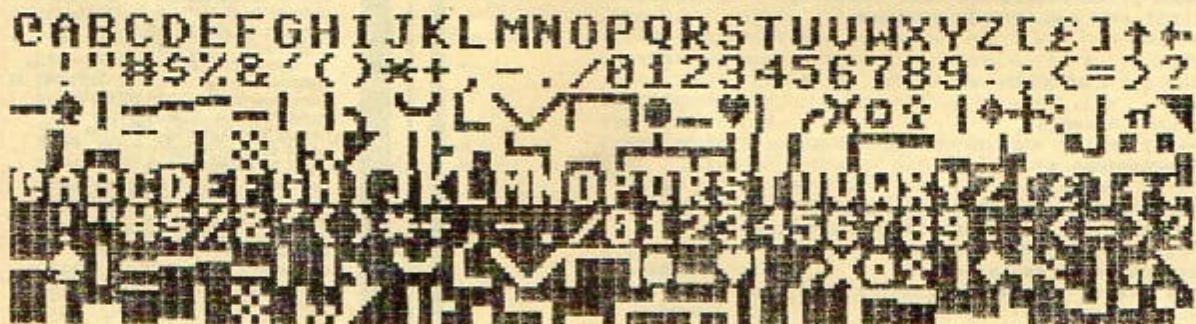
— Im Arbeitsfeld in der linken Bildschirmseite ist das aktuelle Zeichen in achtfacher Vergrößerung zu sehen. Hier werden Zeichen editiert.

— Daneben stehen rechts oben das aktuelle Zeichen in Originalgröße und seine Bildschirmcodezahl.

— Darunter werden die Nummer des Zeichensatzes (1 oder 2) und seine Startadresse für den SAVE-Befehl (siehe unten) angezeigt.

— Im Feld darunter wird die Nummer des zweiten Zeichens für die Befehle Tauschen, Kopieren und Oder-Verknüpfung eingegeben.

— Im Menüfeld auf der rechten



CRSR/CLR/HOME
CR: P. SE/LOE

0/: Z. WAHLEN
,: NR. DEZ. EINGEBEN
,: ZEICHEN EINGEBEN
>: INV. Z. EINGEBEN
SPACE: P. LOESCHEN
SH. SPACE: P. SETZEN
↑: HOCHROTIEREN
←: LINKSROTIEREN
I: INVENTIEREN
H: HORIZONTAL SPIEG.
V: VERTIKAL SPIEG.
D: 90° RECHTSDREHEN
C: Z. COPIEREN
T: Z. TAUSCHEN

Bildschirmhardcopy: „ZED“ bearbeitet den Klammeraffen

Seite stehen die Editor-Kommandos. Da auf dem Bildschirm kein Platz ist, um alle Befehle auf einmal zu zeigen, ist das Menü auf zwei Seiten verteilt.

= schaltet zwischen den Seiten hin und her.

Einige Befehle, für die trotz des doppelseitigen Menüs kein Eckchen mehr frei war, sind unter dem Arbeitsfeld angegeben.

Zeichen bearbeiten

CRSR-Tasten: Mit den Cursor-Tasten wird der blinkende Cursor durch das Arbeitsfeld gefahren. Stößt er an den Rand, erscheint er wieder auf der gegenüberliegenden Seite.

HOME: Der Cursor wird in die linke obere Ecke des Arbeitsfeldes gesetzt.

CLR: Das Arbeitsfeld wird gelöscht und der Cursor in die linke obere Ecke gesetzt.

SPACE: Der Punkt unter dem Cursor wird gelöscht und der Cursor um einen Punkt nach rechts bewegt.

SHIFT-SPACE: Der Punkt an der Cursorposition wird gesetzt, der Cursor wandert einen Punkt nach rechts.

RETURN: Der Punkt, auf dem der Cursor steht, wird invertiert, das heißt, ein gelöschter Punkt wird gesetzt und ein gesetzter Punkt wird gelöscht. Der Cursor bleibt an derselben Stelle.

Die nächsten Befehle verändern das gesamte Zeichen. Aus einem einzigen Zeichen lassen sich mit ihnen 1024 verschiedene Muster erzeugen!

I: Das aktuelle Zeichen wird invertiert.

t: Das Zeichen rotiert um eine Reihe nach oben. Die Reihe, die oben verschwindet, wird unten wieder hineingeschoben.

-: Das Zeichen rotiert um eine Spalte nach links.

D: Das Zeichen wird um 90 Grad nach rechts gedreht.

H: Das Zeichen wird horizontal gespiegelt.

V: Das Zeichen wird vertikal gespiegelt.

Ist ein Zeichen zufriedenstellend umgestaltet, kann das nächste ausgewählt werden. Dazu gibt es gleich drei Möglichkeiten:

;;@/: Mit diesen vier Tasten steuert man einen zweiten, schwarzen Cursor durch das Zeichensatzfeld. Das Zeichen unter dem Cursor erscheint automatisch im Arbeitsfeld und im kleinen Zeichenfeld.

,:: Ein Zeichen wird über seinen Bildschirmcode angewählt. Er muß in dezimaler Form eingegeben werden.

.:: Das gewünschte Zeichen wird direkt über die entsprechende Taste auf der Tastatur ausgesucht. Dabei kann die Taste auch zusammen mit Shift oder Commodore gedrückt werden. Das Zeichen erscheint zunächst nur im kleinen Zeichenfeld. Erst nach Return wird es ins Arbeitsfeld übernommen.

>: Wie unter '.', nur für reverse Zeichen.

Bei den folgenden Befehlen muß ein zweites Zeichen über seinen Bildschirmcode ausgewählt werden. Er wird wieder in dezimaler Form eingegeben.

T: Das aktuelle Zeichen tauscht mit dem zweiten Zeichen den Platz im Speicher.

C: Das zweite Zeichen wird auf das aktuelle Zeichen kopiert.

O: Das Zeichen im Arbeitsfeld wird mit dem zweiten Zeichen oder- verknüpft. Ein Punkt im ersten Zeichen wird also gesetzt, wenn er in mindestens einem der beiden Zeichen gesetzt war.

Zwei Zeichensätze

Mit dem Zeichen-Editor können zwei Zeichensätze unabhängig voneinander bearbeitet werden.

1 2: Mit diesen Tasten wird zwischen den beiden Zeichensätzen hin- und hergesprungen. Damit man nicht durcheinanderkommt, wird neben dem Arbeitsfeld angezeigt, welcher Satz gerade in Arbeit ist.

F2: Der normale Zeichensatz für Großschrift/Grafik wird aus dem ROM in den aktuellen Zeichensatz übernommen. Achtung: Der gera-

de bearbeitete Satz wird dabei gelöscht!

F4: Der ROM-Zeichensatz für Großschrift/Kleinschrift wird in den aktuellen Zeichensatz geholt. Auch hier geht der alte Zeichensatz verloren.

F5: Mit diesem Kommando werden Zeichen vom aktuellen Zeichensatz in den anderen Satz kopiert. Dazu ist der Bildschirmcode des letzten zu übertragenden Zeichens einzugeben. Der gesamte Bereich vom aktuellen Zeichen bis zu diesem Zeichen wechselt hinüber. Will man nur das gerade bearbeitete Zeichen in den anderen Satz übertragen, so reicht es, wenn man statt der Eingabe des Codes des letzten Zeichens nur Return drückt.

F7: Der aktuelle Zeichensatz wird in DATA-Zeilen übertragen. Es werden die Zeilennummern 63000-63255 benutzt. Jede Zeile enthält die acht Bytes für ein Zeichen.

f: Mit diesem Befehl wird der neue Zeichensatz erprobt. Der Bildschirm wird gelöscht, und ein beliebiger Text kann geschrieben werden. Mit der Stop-Taste geht's zurück in den Editor.

Diskettenbefehle

Commodore D: Das Directory wird eingelesen. Nach einem beliebigen Tastendruck wird am unteren Bildschirmrand jeweils ein File angezeigt. Mit der Stop-Taste wird die Ausgabe abgebrochen.

Commodore C: Senden eines Diskbefehls

Commodore E: Lesen des Fehlerkanals

Commodore L: Ein Zeichensatz wird von Diskette in den aktuellen Bereich geladen. Er muß als PRG-File abgelegt sein und im 15. und 16. Zeichen des Namens die Kennung 'z' haben. Er darf nicht länger als neun Blöcke sein, weil sonst der Zeichen-Editor überschrieben wird.

Commodore S: Der aktuelle Zeichensatz wird auf Diskette abgespeichert. Er belegt neun Blöcke und hat im 15. und 16. Zeichen des Namens die Kennung 'z'. Rechts ▶

neben dem Arbeitsfeld steht die Startadresse des erzeugten Programmfiles. Wird der Zeichensatz außerhalb des Editor-Programms mit LOAD .,8,1 geladen, so wird er ab dieser Adresse im Speicher abgelegt. Beim Saven muß daher auf die richtige Startadresse geachtet werden, um böse Überraschungen zu vermeiden. Für den Ladebefehl des Editors hat die Adresse keine Bedeutung.

+ -: Mit diesen Tasten kann die Startadresse für den Save-Befehl neu festgesetzt werden. Die Einstellung beim Start ist \$2000.

Bei den Befehlen für Laden, Speichern und Directory einlesen wird vorübergehend auf den Commodore-Zeichensatz zurückgeschaltet.

RUN/STOP: Alle Befehle können mit der RUN/STOP-Taste abge-

brochen werden, soweit sie nicht nach dem Aufruf sofort erledigt sind. Im normalen Editier-Modus wird nach Drücken von RUN/STOP der Standard-Zeichensatz aktiviert und das Programm verlassen. Es kann mit SYS 49152 neu gestartet werden. Die geänderten Zeichensätze bleiben dabei erhalten.

Die Befehle, die die Eingabe einer Dezimalzahl, eines Filenamens oder eines Diskbefehls verlangen, benutzen eine spezielle Input-Routine: Mit den Tasten Cursor Right (für rechts) und Cursor Down (für links) kann man sich innerhalb des Eingabefeldes frei bewegen. Gelöscht wird mit der Space-Taste. Die Eingabe wird mit RETURN beendet oder mit RUN/STOP ohne weitere Bearbeitung abgebrochen.

Der Editor verbraucht keinen Basic-Speicher, da er die Zeichensätze in dem RAM-Bereich unter dem Basic-ROM ablegt. Dazu läßt man den Videochip auf BANK#2 (\$8000-\$BFFF) zugreifen. Beim Programmstart ist dieser RAM-Bereich noch nicht initialisiert. Deshalb erscheint im Zeichenfeld ein mehr oder weniger zufälliges Punktemuster. Mit F2 oder F4 läßt sich das Durcheinander durch die vertrauten Commodore-Zeichen ersetzen.

Das Programm ist vollständig in Maschinensprache geschrieben und belegt den Speicherbereich von \$C000 bis \$CD75. Es kann als Basic-Lader oder als Hexdump eingegeben werden. Für den Hexdump ist der Checker zu verwenden. Als Startadresse ist C000 anzugeben. *(Andreas Wellie)*

| | | | |
|----|-------|---------------------------------|------|
| 1 | REM | ZEICHEN-EDITOR | |
| 2 | REM | | |
| 3 | REM | VON | |
| 4 | REM | ANDREAS WELLIE' | |
| 5 | REM | SONDERFELD 8 | |
| 6 | REM | 4300 ESSEN 14 | |
| 7 | REM | | |
| 8 | PRINT | "[CLR,2DOWN] ZEICHEN-EDITOR" | 2040 |
| 9 | PRINT | "[DOWN] DATAS WERDEN GELESEN | 1991 |
| | | " " | |
| 10 | FOR | A=49152 TO 52597 | 1925 |
| 11 | READ | B | 267 |
| 12 | POKE | A,B | 475 |
| 13 | C=C+B | | 302 |
| 14 | NEXT | | 130 |
| 15 | IF | C<>393635 THEN PRINT "[2DOWN] | 2062 |
| | | FEHLER IN DATAS !":END | |
| 16 | SYS | 49152 | 780 |
| 17 | DATA | 32,178,192,169,8,141,32,208 | 3003 |
| 18 | DATA | 141,33,208,169,1,141,21,208 | 3347 |
| 19 | DATA | 169,0,141,39,208,169,152,141 | 3473 |
| 20 | DATA | 0,208,169,130,141,1,208,169 | 2748 |
| 21 | DATA | 144,141,248,163,162,63,169,0 | 3672 |
| 22 | DATA | 157,0,164,202,16,250,169,128 | 3071 |
| 23 | DATA | 141,138,2,169,0,133,157,169 | 3190 |
| 24 | DATA | 0,133,66,169,1,133,67,169 | 2736 |
| 25 | DATA | 176,133,248,169,0,133,247,169 | 3544 |
| 26 | DATA | 140,141,213,202,169,49,141,215 | 3340 |
| 27 | DATA | 202,169,160,141,136,2,169,149 | 3518 |
| 28 | DATA | 141,0,221,32,90,200,160,200 | 2884 |
| 29 | DATA | 169,218,32,169,200,169,0,141 | 3038 |
| 30 | DATA | 134,2,173,215,202,141,242,161 | 2924 |
| 31 | DATA | 162,1,134,249,160,0,132,250 | 3131 |
| 32 | DATA | 169,128,133,2,32,226,197,32 | 3412 |
| 33 | DATA | 110,197,32,163,200,32,32,198 | 3100 |
| 34 | DATA | 76,203,192,32,163,199,162,40 | 2977 |
| 35 | DATA | 221,82,202,240,6,202,16,248 | 2964 |
| 36 | DATA | 76,139,192,168,139,123,202,141 | 2961 |
| 37 | DATA | 170,192,169,164,202,141,171,192 | 3591 |
| 38 | DATA | 152,76,0,0,32,178,192,76 | 2799 |
| 39 | DATA | 123,227,169,55,133,1,169,199 | 3555 |
| 40 | DATA | 141,0,221,169,4,141,136,2 | 2540 |
| 41 | DATA | 32,163,253,32,21,253,32,91 | 2860 |
| 42 | DATA | 255,96,234,169,10,133,214,169 | 3033 |
| 43 | DATA | 5,133,211,32,16,229,162,1 | 2571 |
| 44 | DATA | 134,249,160,0,132,250,160,128 | 3106 |
| 45 | DATA | 132,2,32,226,197,76,139,192 | 3600 |
| 46 | DATA | 168,249,224,8,208,15,169,128 | 3206 |
| 47 | DATA | 133,2,169,1,133,249,169,5 | 3315 |
| 48 | DATA | 133,211,76,139,192,230,249,230 | 3025 |
| 49 | DATA | 211,70,2,76,139,192,166,249 | 3617 |
| 50 | DATA | 224,1,208,15,169,1,133,2 | 2407 |
| 51 | DATA | 169,0,133,249,169,12,133,211 | 3126 |
| 52 | DATA | 76,139,192,198,249,198,211,6 | 2442 |
| 53 | DATA | 2,76,139,192,166,250,224,7 | 3038 |
| 54 | DATA | 208,14,169,0,133,250,169,10 | 2947 |
| 55 | DATA | 133,214,32,16,229,76,139,192 | 3211 |
| 56 | DATA | 230,214,32,16,229,230,250,76 | 2813 |
| 57 | DATA | 139,192,166,250,224,0,208,14 | 3150 |
| 58 | DATA | 169,7,133,250,169,17,133,214 | 3460 |
| 59 | DATA | 32,16,229,76,139,192,198,214 | 3259 |
| 60 | DATA | 32,16,229,198,250,76,139,192 | 2998 |
| 61 | DATA | 230,66,32,110,197,76,203,192 | 3266 |
| 62 | DATA | 198,66,76,98,193,32,80,197 | 3347 |
| 63 | DATA | 164,250,177,78,69,2,145,78 | 3244 |
| 64 | DATA | 76,169,196,32,90,197,160,7 | 2747 |
| 65 | DATA | 177,78,10,8,106,40,42,145 | 2774 |
| 66 | DATA | 78,136,16,244,76,169,196,32 | 3209 |
| 67 | DATA | 80,197,160,0,177,78,170,200 | 3112 |
| 68 | DATA | 177,79,136,145,79,200,192,7 | 2988 |
| 69 | DATA | 208,245,138,145,78,76,169,196 | 3762 |
| 70 | DATA | 169,129,32,210,255,169,19,32 | 3033 |
| 71 | DATA | 210,255,165,66,141,0,160,169 | 3474 |
| 72 | DATA | 221,141,208,206,169,1,141,217 | 3471 |
| 73 | DATA | 202,32,197,199,173,0,160,133 | 2899 |
| 74 | DATA | 66,169,0,141,208,206,141,134 | 3038 |
| 75 | DATA | 2,169,146,32,210,255,76,98 | 3129 |
| 76 | DATA | 193,169,10,32,210,255,76,168 | 3286 |
| 77 | DATA | 193,169,14,133,211,169,10,133 | 3302 |
| 78 | DATA | 214,32,16,229,32,250,196,133 | 3179 |
| 79 | DATA | 66,76,98,193,173,213,202,133 | 3350 |
| 80 | DATA | 2,169,132,141,213,202,169,23 | 2698 |
| 81 | DATA | 133,214,169,9,133,211,32,16 | 3113 |
| 82 | DATA | 229,169,250,160,201,32,169,200 | 3509 |
| 83 | DATA | 169,34,160,202,32,169,200,169 | 2859 |
| 84 | DATA | 14,141,217,202,32,197,199,169 | 3079 |
| 85 | DATA | 1,162,8,160,97,32,186,255 | 2685 |
| 86 | DATA | 169,16,162,219,232,160,202,32 | 2932 |
| 87 | DATA | 189,255,32,192,255,169,8,32 | 3087 |
| 88 | DATA | 177,255,169,97,32,147,255,169 | 3555 |
| 89 | DATA | 0,32,166,255,173,209,202,32 | 3186 |
| 90 | DATA | 188,255,165,247,133,78,165,248 | 3700 |
| 91 | DATA | 133,79,162,8,160,0,177,78 | 2999 |
| 92 | DATA | 134,96,32,168,255,166,96,200 | 3006 |
| 93 | DATA | 208,244,230,79,202,208,237,169 | 3204 |
| 94 | DATA | 1,32,195,255,165,2,141,213 | 2828 |
| 95 | DATA | 202,76,94,192,173,213,202,133 | 3258 |
| 96 | DATA | 2,169,132,141,213,202,169,23 | 2698 |

| | | | | | |
|----------|---------------------------------|------|----------|--------------------------------|------|
| 97 DATA | 133,214,169,9,133,211,32,16 | 3113 | 180 DATA | 216,202,136,48,47,185,158,161 | 3342 |
| 98 DATA | 229,169,16,160,202,32,169,200 | 3092 | 181 DATA | 41,15,170,224,0,240,12,173 | 2790 |
| 99 DATA | 169,34,160,202,32,169,200,169 | 2859 | 182 DATA | 216,202,24,105,10,141,216,202 | 2628 |
| 100 DATA | 14,141,217,202,32,197,199,169 | 3079 | 183 DATA | 202,208,244,136,48,216,158 | 3150 |
| 101 DATA | 2,162,8,160,0,32,186,255 | 2709 | 184 DATA | 161,41,15,170,224,0,240,12 | 2735 |
| 102 DATA | 169,16,162,219,232,160,202,32 | 2932 | 185 DATA | 173,216,202,24,105,100,141,216 | 2922 |
| 103 DATA | 189,255,169,0,166,247,164,248 | 3567 | 186 DATA | 202,202,208,244,173,216,202,96 | 3438 |
| 104 DATA | 32,213,255,165,2,141,213,202 | 3052 | 187 DATA | 165,247,133,78,165,240,133,79 | 3282 |
| 105 DATA | 169,0,133,66,76,94,192,32 | 2813 | 188 DATA | 169,0,162,0,228,66,240,11 | 2715 |
| 106 DATA | 80,197,160,7,169,0,145,78 | 2467 | 189 DATA | 24,105,8,208,2,230,79,232 | 2341 |
| 107 DATA | 136,16,251,76,98,193,32,80 | 3236 | 190 DATA | 76,92,197,133,78,96,32,00 | 2892 |
| 108 DATA | 197,160,7,177,78,73,255,145 | 3548 | 191 DATA | 197,169,10,133,214,169,5,133 | 3268 |
| 109 DATA | 78,136,16,247,76,169,196,32 | 3305 | 192 DATA | 211,32,16,229,160,0,132,65 | 3017 |
| 110 DATA | 80,197,160,7,162,0,161,78 | 2523 | 193 DATA | 169,128,133,60,177,78,37,60 | 3221 |
| 111 DATA | 72,177,78,129,78,104,145,78 | 3059 | 194 DATA | 209,5,169,32,76,145,197,169 | 3579 |
| 112 DATA | 230,78,136,136,16,240,76,169 | 3151 | 195 DATA | 209,132,65,32,210,255,164,65 | 3175 |
| 113 DATA | 196,165,66,72,32,184,186,133 | 3484 | 196 DATA | 165,60,201,1,240,5,70,60 | 2449 |
| 114 DATA | 66,32,80,197,160,7,177,78 | 2692 | 197 DATA | 76,132,197,200,192,0,240,16 | 2791 |
| 115 DATA | 145,73,136,16,249,104,133,66 | 3357 | 198 DATA | 132,65,230,214,169,5,133,211 | 3040 |
| 116 DATA | 76,94,182,165,66,72,32,194 | 3107 | 199 DATA | 32,16,229,164,65,76,128,197 | 3024 |
| 117 DATA | 196,133,66,32,80,197,160,7 | 3088 | 200 DATA | 160,201,169,218,32,169,200,169 | 3419 |
| 118 DATA | 177,78,170,177,73,145,78,138 | 3511 | 201 DATA | 55,133,1,166,66,169,0,32 | 2776 |
| 119 DATA | 145,73,136,16,243,104,133,66 | 3345 | 202 DATA | 205,189,169,54,133,1,32,210 | 3250 |
| 120 DATA | 76,94,192,32,80,197,160,7 | 2796 | 203 DATA | 197,96,160,7,162,23,177,78 | 3296 |
| 121 DATA | 177,78,162,7,74,133,73,165 | 3545 | 204 DATA | 157,0,164,202,202,202,136,16 | 2853 |
| 122 DATA | 74,42,133,74,165,73,202,16 | 2836 | 205 DATA | 245,96,169,4,133,61,133,63 | 3352 |
| 123 DATA | 243,165,74,145,78,136,16,232 | 3252 | 206 DATA | 169,160,133,62,169,216,133,64 | 3499 |
| 124 DATA | 76,169,196,165,66,72,32,194 | 3114 | 207 DATA | 162,0,160,0,130,145,61,197 | 3264 |
| 125 DATA | 196,133,66,32,80,197,160,7 | 3088 | 208 DATA | 66,208,5,169,0,76,2,198 | 2279 |
| 126 DATA | 177,73,17,78,145,73,136,16 | 3505 | 209 DATA | 169,8,145,63,232,200,192,32 | 3352 |
| 127 DATA | 247,184,133,66,76,94,192,32 | 3407 | 210 DATA | 208,234,169,40,24,101,63,133 | 3116 |
| 128 DATA | 80,197,160,7,162,7,177,78 | 2712 | 211 DATA | 63,133,61,144,4,230,62,230 | 2965 |
| 129 DATA | 74,62,224,206,202,16,249,136 | 2915 | 212 DATA | 64,169,68,197,61,208,211,96 | 3319 |
| 130 DATA | 16,242,160,7,185,224,206,145 | 3183 | 213 DATA | 169,123,133,73,169,203,133,74 | 3770 |
| 131 DATA | 78,136,16,248,76,169,196,32 | 3082 | 214 DATA | 169,7,141,134,2,166,67,160 | 3420 |
| 132 DATA | 80,197,164,250,165,2,224,180 | 2781 | 215 DATA | 0,177,73,208,3,202,240,8 | 2709 |
| 133 DATA | 240,8,73,255,49,78,224,100 | 2780 | 216 DATA | 32,125,198,76,49,198,32,125 | 3098 |
| 134 DATA | 208,2,17,78,145,78,165,211 | 3133 | 217 DATA | 198,169,9,133,214,169,21,133 | 3016 |
| 135 DATA | 133,69,165,214,133,70,32,110 | 2863 | 218 DATA | 211,32,16,229,160,0,177,73 | 3142 |
| 136 DATA | 197,165,69,133,211,165,70,133 | 3367 | 219 DATA | 240,9,32,210,255,32,125,198 | 3449 |
| 137 DATA | 214,32,16,229,76,232,192,32 | 3146 | 220 DATA | 76,76,198,32,125,198,203,214 | 3432 |
| 138 DATA | 80,197,162,100,76,154,195,165 | 2927 | 221 DATA | 169,23,197,214,208,223,169,8 | 3007 |
| 139 DATA | 66,56,233,32,133,66,76,98 | 3036 | 222 DATA | 141,134,2,165,249,24,105,4 | 2930 |
| 140 DATA | 193,165,66,24,105,32,133,66 | 3346 | 223 DATA | 133,211,165,250,24,105,10,133 | 3154 |
| 141 DATA | 76,98,193,160,216,132,96,160 | 2760 | 224 DATA | 214,32,16,229,96,230,73,208 | 3283 |
| 142 DATA | 224,132,91,76,140,199,165,66 | 3370 | 225 DATA | 2,230,74,96,160,208,132,96 | 3081 |
| 143 DATA | 133,95,32,194,196,197,66,176 | 3341 | 226 DATA | 160,216,132,91,162,0,134,95 | 3119 |
| 144 DATA | 3,76,94,192,133,96,230,96 | 2941 | 227 DATA | 162,0,134,98,162,0,134,88 | 2475 |
| 145 DATA | 32,80,197,173,242,161,201,49 | 2831 | 228 DATA | 165,240,24,105,0,133,89,120 | 2810 |
| 146 DATA | 240,10,165,79,56,233,8,133 | 2689 | 229 DATA | 169,51,133,1,32,191,163,169 | 3529 |
| 147 DATA | 89,76,27,196,165,79,24,105 | 3010 | 230 DATA | 54,133,1,88,76,106,192,169 | 3093 |
| 148 DATA | 8,133,89,165,78,133,88,160 | 2900 | 231 DATA | 23,133,214,169,0,133,211,32 | 3096 |
| 149 DATA | 7,177,78,145,88,136,16,249 | 3124 | 232 DATA | 16,229,162,214,134,187,160,202 | 2645 |
| 150 DATA | 230,66,165,96,197,66,200,200 | 3613 | 233 DATA | 132,188,169,1,133,193,169,8 | 3205 |
| 151 DATA | 165,95,133,66,76,94,192,173 | 3242 | 234 DATA | 133,196,169,96,133,185,32,213 | 2984 |
| 152 DATA | 209,202,24,105,8,141,209,202 | 2824 | 235 DATA | 243,165,196,32,180,255,165,185 | 3439 |
| 153 DATA | 32,183,200,76,139,192,173,209 | 3071 | 236 DATA | 32,150,255,169,0,133,144,160 | 2818 |
| 154 DATA | 202,56,233,8,141,209,202,32 | 3307 | 237 DATA | 3,132,251,32,165,255,133,252 | 2952 |
| 155 DATA | 183,200,76,139,192,173,213,202 | 3159 | 238 DATA | 164,144,208,69,32,165,255,164 | 3368 |
| 156 DATA | 141,212,202,169,0,141,21,208 | 3140 | 239 DATA | 144,208,62,164,251,136,208,233 | 3144 |
| 157 DATA | 169,147,32,210,255,32,163,199 | 3080 | 240 DATA | 166,252,160,55,132,1,32,205 | 3268 |
| 158 DATA | 201,3,240,6,32,210,255,76 | 2751 | 241 DATA | 189,169,54,133,1,169,32,32 | 3350 |
| 159 DATA | 101,196,169,132,141,212,202,169 | 3777 | 242 DATA | 210,255,32,165,255,166,144,208 | 3168 |
| 160 DATA | 1,141,21,208,76,94,192,141 | 2677 | 243 DATA | 32,170,240,6,32,210,255,76 | 3051 |
| 161 DATA | 215,202,56,233,49,170,189,205 | 3146 | 244 DATA | 10,199,160,23,132,214,32,163 | 2773 |
| 162 DATA | 202,133,248,189,207,202,141,213 | 3906 | 245 DATA | 199,201,3,208,3,76,49,199 | 2634 |
| 163 DATA | 202,76,106,192,165,67,201,1 | 3123 | 246 DATA | 169,13,32,210,255,160,2,208 | 2779 |
| 164 DATA | 240,5,169,1,76,161,196,169 | 3215 | 247 DATA | 176,32,66,246,76,94,192,169 | 3423 |
| 165 DATA | 15,133,67,32,32,198,76,139 | 3304 | 248 DATA | 23,133,214,169,0,133,211,32 | 3096 |
| 166 DATA | 192,165,211,133,69,165,214,133 | 3766 | 249 DATA | 16,229,169,62,32,210,255,169 | 3128 |
| 167 DATA | 70,32,110,197,165,69,133,211 | 3329 | 250 DATA | 14,141,217,202,32,197,199,169 | 3079 |
| 168 DATA | 165,70,133,214,32,16,229,76 | 3035 | 251 DATA | 32,205,220,202,240,27,169,1 | 2664 |
| 169 DATA | 139,192,32,80,197,165,78,133 | 3167 | 252 DATA | 162,8,160,15,32,186,255,173 | 3415 |
| 170 DATA | 73,165,79,133,74,169,16,133 | 3054 | 253 DATA | 217,202,162,219,232,160,202,32 | 3719 |
| 171 DATA | 211,169,16,133,214,32,16,229 | 3266 | 254 DATA | 189,255,32,192,255,169,0,133 | 3130 |
| 172 DATA | 169,32,141,220,202,141,221,202 | 3007 | 255 DATA | 152,76,94,192,169,23,133,214 | 3472 |
| 173 DATA | 141,222,202,169,3,141,217,202 | 3166 | 256 DATA | 169,0,133,211,32,16,229,169 | 3334 |
| 174 DATA | 32,207,199,162,3,189,219,202 | 3274 | 257 DATA | 8,133,186,32,180,255,169,111 | 2958 |
| 175 DATA | 157,157,161,202,208,247,32,2 | 3325 | 258 DATA | 133,185,32,150,255,32,165,255 | 3127 |
| 176 DATA | 197,96,169,3,141,217,202,32 | 3179 | 259 DATA | 201,13,240,6,32,210,255,76 | 3256 |
| 177 DATA | 197,199,165,66,141,216,202,160 | 3544 | 260 DATA | 141,199,32,171,255,32,163,199 | 3234 |
| 178 DATA | 2,185,158,161,201,32,208,5 | 2818 | 261 DATA | 76,94,192,169,0,133,212,165 | 3032 |
| 179 DATA | 136,48,57,16,244,41,15,141 | 3013 | 262 DATA | 198,133,204,141,146,2,240,243 | 3598 |

| | | | | | |
|----------|---------------------------------|------|----------|---------------------------------|------|
| 263 DATA | 120,165,207,240,12,165,206,174 | 3383 | 345 DATA | 90,157,157,157,157,157,157,157 | 3576 |
| 264 DATA | 135,2,160,0,132,207,32,19 | 3087 | 346 DATA | 157,157,157,157,157,157,157,157 | 4099 |
| 265 DATA | 234,32,180,229,96,169,32,162 | 3035 | 347 DATA | 157,0,3,19,29,157,17,145 | 2573 |
| 266 DATA | 14,157,219,202,202,200,250,162 | 2944 | 348 DATA | 59,50,172,180,13,95,94,46 | 2540 |
| 267 DATA | 1,142,218,202,32,163,199,201 | 2854 | 349 DATA | 62,44,174,182,147,73,72,67 | 3045 |
| 268 DATA | 3,200,17,169,0,141,200,206 | 2442 | 350 DATA | 84,86,79,60,32,160,177,64 | 3135 |
| 269 DATA | 165,2,201,130,144,3,141,213 | 2960 | 351 DATA | 47,137,130,135,43,45,92,49 | 2770 |
| 270 DATA | 202,76,94,192,201,13,200,1 | 3029 | 352 DATA | 50,136,61,172,202,232,6,36 | 2486 |
| 271 DATA | 96,201,29,200,16,174,219,202 | 2070 | 353 DATA | 66,96,104,175,55,109,123,143 | 3365 |
| 272 DATA | 236,217,202,240,215,230,210,202 | 3720 | 354 DATA | 160,217,225,244,116,199,214,231 | 4051 |
| 273 DATA | 230,211,76,212,199,201,17,200 | 2875 | 355 DATA | 1,27,59,91,119,151,199,116 | 2600 |
| 274 DATA | 15,173,210,202,201,1,240,196 | 3074 | 356 DATA | 207,217,132,227,238,55,70,85 | 3756 |
| 275 DATA | 206,218,202,190,211,76,212,199 | 3432 | 357 DATA | 127,127,236,140,192,192,192,193 | 3767 |
| 276 DATA | 201,32,144,8,201,96,144,7 | 2071 | 358 DATA | 193,193,193,193,190,199,193,193 | 4074 |
| 277 DATA | 201,160,178,3,76,212,199,174 | 3430 | 359 DATA | 193,193,193,193,193,194,194,194 | 3705 |
| 278 DATA | 210,202,157,213,202,230,210,202 | 3767 | 360 DATA | 194,195,195,195,195,195,195,195 | 4091 |
| 279 DATA | 32,210,255,169,0,133,212,174 | 2966 | 361 DATA | 199,195,195,198,195,195,196,196 | 3086 |
| 280 DATA | 217,202,232,236,210,202,200,5 | 3624 | 362 DATA | 196,196,196,202,196,176,184,140 | 3626 |
| 281 DATA | 206,218,202,190,211,169,221,205 | 3704 | 363 DATA | 142,32,1,120,132,140,36,234 | 3254 |
| 282 DATA | 200,206,200,11,173,0,160,133 | 3013 | 364 DATA | 0,0,0,36,32,32,32,32 | 2360 |
| 283 DATA | 66,32,80,197,32,210,197,76 | 3026 | 365 DATA | 32,32,32,32,32,32,32,32 | 2637 |
| 284 DATA | 212,199,120,169,127,141,13,220 | 3522 | 366 DATA | 32,32,46,90,32,178,192,165 | 2714 |
| 285 DATA | 169,1,141,26,200,141,18,200 | 3194 | 367 DATA | 247,133,69,165,240,133,70,169 | 3430 |
| 286 DATA | 141,240,200,169,27,141,17,200 | 3119 | 368 DATA | 246,133,74,169,24,133,73,169 | 3426 |
| 287 DATA | 169,124,141,20,3,169,200,141 | 3297 | 369 DATA | 119,160,203,32,30,171,166,73 | 3320 |
| 288 DATA | 21,3,80,96,173,25,200,141 | 2496 | 370 DATA | 165,74,32,205,199,169,60,32 | 3029 |
| 289 DATA | 25,200,41,1,240,26,174,240 | 2460 | 371 DATA | 210,255,169,193,32,210,255,160 | 3317 |
| 290 DATA | 206,109,212,202,141,24,200,109 | 3145 | 372 DATA | 0,169,54,133,1,177,69,72 | 2444 |
| 291 DATA | 210,202,141,18,200,202,142,240 | 2913 | 373 DATA | 169,55,133,1,104,170,152,72 | 3199 |
| 292 DATA | 200,16,0,162,1,142,240,206 | 2793 | 374 DATA | 169,0,32,205,169,104,169,200 | 3046 |
| 293 DATA | 76,49,234,104,160,104,170,104 | 2844 | 375 DATA | 192,8,240,0,169,44,32,210 | 2640 |
| 294 DATA | 64,72,169,55,133,1,104,32 | 2493 | 376 DATA | 255,76,25,203,169,109,160,203 | 3206 |
| 295 DATA | 30,171,169,54,133,1,96,173 | 2956 | 377 DATA | 32,30,171,169,13,141,122,2 | 2604 |
| 296 DATA | 209,202,162,0,74,74,74,74 | 2933 | 378 DATA | 141,123,2,169,5,133,190,96 | 3070 |
| 297 DATA | 32,199,200,232,173,209,202,41 | 2571 | 379 DATA | 290,73,200,2,230,74,165,73 | 3544 |
| 298 DATA | 15,201,10,176,6,24,105,40 | 2223 | 380 DATA | 201,24,240,14,165,69,24,105 | 3000 |
| 299 DATA | 76,214,200,56,233,9,157,63 | 2764 | 381 DATA | 0,133,69,144,2,230,70,76 | 2393 |
| 300 DATA | 162,96,13,8,129,147,17,17 | 3107 | 382 DATA | 255,202,76,0,192,13,03,217 | 3030 |
| 301 DATA | 17,17,17,17,17,17,17,29 | 2957 | 383 DATA | 53,50,48,52,56,19,0,147 | 2556 |
| 302 DATA | 29,29,29,20,213,192,192,192 | 3062 | 384 DATA | 17,17,0,0,64,47,50,59 | 2204 |
| 303 DATA | 192,192,192,192,192,170,192,192 | 3617 | 385 DATA | 32,90,46,32,87,65,69,72 | 2662 |
| 304 DATA | 192,178,192,201,13,29,29,29 | 3109 | 386 DATA | 76,69,78,32,32,32,0,44 | 2606 |
| 305 DATA | 29,221,32,32,32,32,32,32 | 2414 | 387 DATA | 50,78,82,46,32,60,69,90 | 2601 |
| 306 DATA | 32,32,221,32,32,32,221,32 | 2350 | 388 DATA | 46,69,73,78,71,69,66,69 | 2970 |
| 307 DATA | 221,13,29,29,29,29,221,32 | 2910 | 389 DATA | 70,0,46,50,90,69,73,67 | 2515 |
| 308 DATA | 32,32,32,32,32,32,32,171 | 2600 | 390 DATA | 72,69,78,32,69,73,70,71 | 2966 |
| 309 DATA | 192,192,192,177,192,179,13,29 | 3669 | 391 DATA | 69,66,69,70,0,62,50,73 | 2066 |
| 310 DATA | 29,29,29,221,32,32,32,32 | 2315 | 392 DATA | 70,86,46,32,90,46,32,69 | 2640 |
| 311 DATA | 32,32,32,32,221,66,46,35 | 2005 | 393 DATA | 73,70,71,69,66,69,70,0 | 2744 |
| 312 DATA | 32,49,221,13,29,29,29,29 | 2249 | 394 DATA | 83,80,65,67,69,50,80,46 | 2463 |
| 313 DATA | 221,32,32,32,32,32,32,32 | 2601 | 395 DATA | 32,76,79,69,83,67,72,69 | 2900 |
| 314 DATA | 32,221,32,32,32,32,32,221 | 2350 | 396 DATA | 70,0,83,72,46,83,80,65 | 2501 |
| 315 DATA | 13,29,29,29,29,221,32,32 | 2320 | 397 DATA | 67,69,50,80,46,32,83,69 | 2040 |
| 316 DATA | 32,32,32,32,32,32,221,36 | 2523 | 398 DATA | 84,90,69,78,0,94,50,72 | 2773 |
| 317 DATA | 48,48,48,48,221,13,29,29 | 2370 | 399 DATA | 79,67,72,82,79,84,73,69 | 3094 |
| 318 DATA | 29,29,221,32,32,32,32,32 | 2310 | 400 DATA | 92,69,78,32,32,32,32,0 | 2455 |
| 319 DATA | 32,32,32,171,192,192,192,192 | 3223 | 401 DATA | 95,50,76,73,78,75,83,82 | 2530 |
| 320 DATA | 192,179,13,29,29,29,29,221 | 3004 | 402 DATA | 79,84,73,69,82,63,78,32 | 2030 |
| 321 DATA | 32,32,32,32,32,32,32,32 | 2637 | 403 DATA | 32,32,0,73,50,73,78,86 | 2291 |
| 322 DATA | 221,50,35,32,32,32,221,13 | 2642 | 404 DATA | 69,78,84,73,69,82,63,70 | 2752 |
| 323 DATA | 29,29,29,29,221,32,32,32 | 2546 | 405 DATA | 32,32,32,0,72,50,72,79 | 2603 |
| 324 DATA | 32,32,32,32,32,171,192,192 | 2795 | 406 DATA | 82,73,90,79,78,84,65,76 | 2902 |
| 325 DATA | 192,192,192,203,13,29,29,29 | 2900 | 407 DATA | 32,83,80,73,69,46,0,86 | 2610 |
| 326 DATA | 29,202,192,192,192,192,192,192 | 2777 | 408 DATA | 50,86,69,82,84,73,75,65 | 2704 |
| 327 DATA | 192,192,203,13,17,29,29,29 | 3312 | 409 DATA | 76,32,83,80,73,69,71,46 | 2577 |
| 328 DATA | 29,150,67,82,83,82,47,67 | 2400 | 410 DATA | 0,60,50,57,40,39,32,82 | 2400 |
| 329 DATA | 76,82,47,72,79,77,69,13 | 2051 | 411 DATA | 69,67,72,84,93,60,62,69 | 3009 |
| 330 DATA | 29,29,29,29,67,82,50,32 | 2025 | 412 DATA | 72,69,78,0,67,50,90,46 | 2739 |
| 331 DATA | 80,46,32,83,69,47,76,79 | 3084 | 413 DATA | 32,67,79,80,73,69,82,69 | 3175 |
| 332 DATA | 69,0,19,17,17,17,17,17 | 2764 | 414 DATA | 70,0,84,50,90,46,32,84 | 2104 |
| 333 DATA | 17,17,17,17,17,29,29,29 | 3002 | 415 DATA | 65,85,83,67,72,69,78,0 | 2015 |
| 334 DATA | 29,29,29,29,29,29,29,29 | 2637 | 416 DATA | 79,50,79,82,45,77,73,80 | 2515 |
| 335 DATA | 29,29,29,32,32,32,157,157 | 2721 | 417 DATA | 69,76,32,32,32,32,32,32 | 2672 |
| 336 DATA | 157,0,90,69,73,67,72,69 | 3106 | 418 DATA | 32,32,0,92,50,90,46,45 | 2324 |
| 337 DATA | 70,83,65,84,90,32,83,80 | 2360 | 419 DATA | 80,82,79,66,69,32,32,32 | 2006 |
| 338 DATA | 69,73,67,72,69,82,70,0 | 2612 | 420 DATA | 32,32,32,32,32,0,43,45 | 2662 |
| 339 DATA | 90,69,73,67,72,69,78,83 | 2924 | 421 DATA | 50,90,45,65,60,82,69,83 | 2650 |
| 340 DATA | 65,84,90,32,76,65,60,69 | 3126 | 422 DATA | 83,69,32,65,69,70,60,46 | 2605 |
| 341 DATA | 70,0,13,29,29,29,29,29 | 2394 | 423 DATA | 0,49,50,50,90,45,83,65 | 2593 |
| 342 DATA | 29,29,29,29,78,65,77,69 | 2730 | 424 DATA | 84,90,32,84,65,85,83,67 | 2090 |
| 343 DATA | 50,32,32,32,32,32,32,32 | 2605 | 425 DATA | 72,69,70,0,70,50,50,78 | 2540 |
| 344 DATA | 32,32,32,32,32,32,32,46 | 2440 | 426 DATA | 79,82,46,90,46,32,45,62 | 2664 |

427 DATA 32,02,65,77,32,32,0,70
 428 DATA 52,58,75,78,69,46,90,46
 429 DATA 32,45,62,32,82,65,77,32
 430 DATA 0,70,53,58,90,46,73,78
 431 DATA 32,65,78,46,66,69,82,69
 432 DATA 73,67,72,0,70,55,58,68
 433 DATA 65,84,65,90,69,73,76,69
 434 DATA 78,32,69,82,83,46,0,67
 435 DATA 61,32,69,58,68,73,83,75
 436 DATA 83,84,65,84,85,83,32,32
 437 DATA 32,0,67,61,32,67,58,68

2665
 2533
 2649
 2568
 3047
 2702
 2888
 2724
 2738
 2557
 2520

438 DATA 73,83,75,66,69,70,69,72
 439 DATA 76,32,32,32,0,67,61,32
 440 DATA 68,58,68,73,82,69,67,84
 441 DATA 79,62,69,32,32,0,67
 442 DATA 61,32,83,58,90,46,32,83
 443 DATA 80,69,73,67,72,69,82,78
 444 DATA 32,0,67,61,32,76,58,90
 445 DATA 46,32,76,65,68,69,78,0
 446 DATA 32,32,32,32,32,32,32,32
 447 DATA 32,32,32,32,32,0

2981
 2496
 2377
 2811
 2327
 2980
 2486
 2595
 2637
 1888

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|----|--------|
| C000 | 20 | B2 | C0 | A9 | 08 | 8D | 20 | D0 | 51 |
| C008 | 8D | 21 | D0 | A9 | 01 | 8D | 15 | D0 | 4C |
| C010 | A9 | 00 | 8D | 27 | D0 | A9 | 98 | 8D | 40 |
| C018 | 00 | D0 | A9 | 82 | 8D | 01 | D0 | A9 | 0A |
| C020 | 90 | 8D | F8 | A3 | A2 | 3F | A9 | 00 | 63 |
| C028 | 9D | 00 | A4 | CA | 10 | FA | A9 | 80 | B6 |
| C030 | 8D | 8A | 02 | A9 | 00 | 85 | 9D | A9 | CC |
| C038 | 00 | 85 | 42 | A9 | 01 | 85 | 43 | A9 | 2C |
| C040 | B0 | 85 | F8 | A9 | 00 | 85 | F7 | A9 | 43 |
| C048 | 8C | 8D | D5 | CA | A9 | 31 | 8D | D7 | 37 |
| C050 | CA | A9 | A0 | 8D | 88 | 02 | A9 | 95 | CD |
| C058 | 8D | 00 | DD | 20 | 5A | C8 | A0 | C8 | B3 |
| C060 | A9 | DA | 20 | A9 | C8 | A9 | 00 | 8D | D9 |
| C068 | 86 | 02 | AD | D7 | CA | 8D | F2 | A1 | E7 |
| C070 | A2 | 01 | 86 | F9 | A0 | 00 | 84 | FA | 57 |
| C078 | A9 | 80 | 85 | 02 | 20 | E2 | C5 | 20 | 8F |
| C080 | 6E | C5 | 20 | B7 | C8 | 20 | 20 | C6 | 15 |
| C088 | 4C | CB | C0 | 20 | A3 | C7 | A2 | 28 | 77 |
| C090 | DD | 52 | CA | F0 | 06 | CA | 10 | F8 | BA |
| C098 | 4C | 8B | C0 | A8 | BD | 7B | CA | 8D | 58 |
| C0A0 | AA | C0 | BD | A4 | CA | 8D | AB | C0 | 28 |
| C0A8 | 98 | 4C | 00 | 00 | 20 | B2 | C0 | 4C | 2E |
| C0B0 | 7B | E3 | A9 | 37 | 85 | 01 | A9 | C7 | 6F |
| C0B8 | 8D | 00 | DD | A9 | 04 | 8D | 88 | 02 | BE |
| C0C0 | 20 | A3 | FD | 20 | 15 | FD | 20 | 5B | 38 |
| C0C8 | FF | 60 | EA | A9 | 0A | 85 | D6 | A9 | 91 |
| C0D0 | 05 | 85 | D3 | 20 | 10 | E5 | A2 | 01 | FF |
| C0D8 | 86 | F9 | A0 | 00 | 84 | FA | A0 | 80 | D7 |
| C0E0 | 84 | 02 | 20 | E2 | C5 | 4C | 8B | C0 | 04 |
| C0E8 | A6 | F9 | E0 | 08 | D0 | 0F | A9 | 80 | 6E |
| C0F0 | 85 | 02 | A9 | 01 | 85 | F9 | A9 | 05 | 5A |
| C0F8 | 85 | D3 | 4C | 8B | C0 | E6 | F9 | E6 | 60 |
| C100 | D3 | 46 | 02 | 4C | 8B | C0 | A6 | F9 | BB |
| C108 | E0 | 01 | D0 | 0F | A9 | 01 | 85 | 02 | C9 |
| C110 | A9 | 08 | 85 | F9 | A9 | 0C | 85 | D3 | 7B |
| C118 | 4C | 8B | C0 | C6 | F9 | C6 | D3 | 06 | BB |
| C120 | 02 | 4C | 8B | C0 | A6 | FA | E0 | 07 | 06 |
| C128 | D0 | 0E | A9 | 00 | 85 | FA | A9 | 0A | 94 |
| C130 | 85 | D6 | 20 | 10 | E5 | 4C | 8B | C0 | 23 |
| C138 | E6 | D6 | 20 | 10 | E5 | E6 | FA | 4C | CC |
| C140 | 8B | C0 | A6 | FA | E0 | 00 | D0 | 0E | 18 |
| C148 | A9 | 07 | 85 | FA | A9 | 11 | 85 | D6 | 73 |
| C150 | 20 | 10 | E5 | 4C | 8B | C0 | C6 | D6 | 63 |
| C158 | 20 | 10 | E5 | C6 | FA | 4C | 8B | C0 | A7 |
| C160 | E6 | 42 | 20 | 8E | C5 | 4C | CB | C0 | D9 |
| C168 | C6 | 42 | 4C | 62 | C1 | 20 | 50 | C5 | 99 |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| C170 | A4 | FA | B1 | 4E | 45 | 02 | 91 | 4E | 93 |
| C178 | 4C | A9 | C4 | 20 | 50 | C5 | A0 | 07 | 6B |
| C180 | B1 | 4E | 0A | 08 | 6A | 28 | 2A | 91 | 40 |
| C188 | 4E | 88 | 10 | F4 | 4C | A9 | C4 | 20 | 2A |
| C190 | 50 | C5 | A0 | 00 | B1 | 4E | AA | C8 | 6F |
| C198 | B1 | 4E | 88 | 91 | 4E | C8 | C0 | 07 | 99 |
| C1A0 | D0 | F5 | 8A | 91 | 4E | 4C | A9 | C4 | BB |
| C1A8 | A9 | 81 | 20 | D2 | FF | A9 | 13 | 20 | C0 |
| C1B0 | D2 | FF | A5 | 42 | 8D | 00 | A0 | A9 | B9 |
| C1B8 | DD | 8D | D0 | CE | A9 | 01 | 8D | D9 | AD |
| C1C0 | CA | 20 | C5 | C7 | AD | 00 | A0 | 85 | 6F |
| C1C8 | 42 | A9 | 00 | 8D | D0 | CE | 8D | 86 | 16 |
| C1D0 | 02 | A9 | 92 | 20 | D2 | FF | 4C | 62 | 89 |
| C1D8 | C1 | A9 | 12 | 20 | D2 | FF | 4C | A8 | 81 |
| C1E0 | C1 | A9 | 0E | 85 | D3 | A9 | 0A | 85 | 51 |
| C1E8 | D6 | 20 | 10 | E5 | 20 | FA | C4 | 85 | 47 |
| C1F0 | 42 | 4C | 62 | C1 | AD | D5 | CA | 85 | B4 |
| C1F8 | 02 | A9 | 84 | 8D | D5 | CA | A9 | 17 | ED |
| C200 | 85 | D6 | A9 | 09 | 85 | D3 | 20 | 10 | 11 |
| C208 | E5 | A9 | FA | A0 | C9 | 20 | A9 | C8 | 1F |
| C210 | A9 | 22 | A0 | CA | 20 | A9 | C8 | A9 | F3 |
| C218 | 0E | 8D | D9 | CA | 20 | C5 | C7 | A9 | 0A |
| C220 | 01 | A2 | 08 | A0 | 61 | 20 | BA | FF | C4 |
| C228 | A9 | 10 | A2 | DB | E8 | A0 | CA | 20 | 51 |
| C230 | BD | FF | 20 | C0 | FF | A9 | 08 | 20 | 5D |
| C238 | B1 | FF | A9 | 61 | 20 | 93 | FF | A9 | DD |
| C240 | 00 | 20 | A8 | FF | AD | D1 | CA | 20 | 0F |
| C248 | A8 | FF | A5 | F7 | 85 | 4E | A5 | F8 | 3B |
| C250 | 85 | 4F | A2 | 08 | A0 | 00 | B1 | 4E | D1 |
| C258 | 86 | 60 | 20 | A8 | FF | A6 | 60 | C8 | 8F |
| C260 | D0 | F4 | E6 | 4F | CA | D0 | ED | A9 | B0 |
| C268 | 01 | 20 | C3 | FF | A5 | 02 | 8D | D5 | 00 |
| C270 | CA | 4C | 5E | C0 | AD | D5 | CA | 85 | 39 |
| C278 | 02 | A9 | 84 | 8D | D5 | CA | A9 | 17 | ED |
| C280 | 85 | D6 | A9 | 09 | 85 | D3 | 20 | 10 | 11 |
| C288 | E5 | A9 | 10 | A0 | CA | 20 | A9 | C8 | 37 |
| C290 | A9 | 22 | A0 | CA | 20 | A9 | C8 | A9 | F3 |
| C298 | 0E | 8D | D9 | CA | 20 | C5 | C7 | A9 | 0A |
| C2A0 | 02 | A2 | 08 | A0 | 00 | 20 | BA | FF | 64 |
| C2A8 | A9 | 10 | A2 | DB | E8 | A0 | CA | 20 | 51 |
| C2B0 | BD | FF | A9 | 00 | A6 | F7 | A4 | F8 | C2 |
| C2B8 | 20 | D5 | FF | A5 | 02 | 8D | D5 | CA | 26 |
| C2C0 | A9 | 00 | 85 | 42 | 4C | 5E | C0 | 20 | 79 |
| C2C8 | 50 | C5 | A0 | 07 | A9 | 00 | 91 | 4E | 0F |
| C2D0 | 88 | 10 | FB | 4C | 62 | C1 | 20 | 50 | 98 |
| C2D8 | C5 | A0 | 07 | B1 | 4E | 49 | FF | 91 | EE |
| C2E0 | 4E | 88 | 10 | F7 | 4C | A9 | C4 | 20 | 27 |
| C2E8 | 50 | C5 | A0 | 07 | A2 | 00 | A1 | 4E | 18 |

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| C2F0 | 48 | B1 | 4E | 81 | 4E | 68 | 91 | 4E | 8E | C4C0 | 8B | C0 | 20 | 50 | C5 | A5 | 4E | 85 | 84 |
| C2F8 | E6 | 4E | 88 | 88 | 10 | F0 | 4C | A9 | 5B | C4C8 | 49 | A5 | 4F | 85 | 4A | A9 | 10 | 85 | 9B |
| C300 | C4 | A5 | 42 | 48 | 20 | C2 | C4 | 85 | B6 | C4D0 | D3 | A9 | 10 | 85 | D6 | 20 | 10 | E5 | 96 |
| C308 | 42 | 20 | 50 | C5 | A0 | 07 | B1 | 4E | A9 | C4D8 | A9 | 20 | 8D | DC | CA | 8D | DD | CA | 89 |
| C310 | 91 | 49 | 88 | 10 | F9 | 68 | 85 | 42 | 93 | C4E0 | 8D | DE | CA | A9 | 03 | 8D | D9 | CA | 55 |
| C318 | 4C | 5E | C0 | A5 | 42 | 48 | 20 | C2 | 61 | C4E8 | 20 | CF | C7 | A2 | 03 | BD | DB | CA | CE |
| C320 | C4 | 85 | 42 | 20 | 50 | C5 | A0 | 07 | 85 | C4F0 | 9D | 9D | A1 | CA | D0 | F7 | 20 | 02 | CF |
| C328 | B1 | 4E | AA | B1 | 49 | 91 | 4E | 8A | D8 | C4F8 | C5 | 60 | A9 | 03 | 8D | D9 | CA | 20 | 68 |
| C330 | 91 | 49 | 88 | 10 | F3 | 68 | 85 | 42 | 8D | C500 | C5 | C7 | A5 | 42 | 8D | D8 | CA | A0 | 40 |
| C338 | 4C | 5E | C0 | 20 | 50 | C5 | A0 | 07 | B2 | C508 | 02 | B9 | 9E | A1 | C9 | 20 | D0 | 05 | BA |
| C340 | B1 | 4E | A2 | 07 | 4A | 85 | 49 | A5 | 66 | C510 | 88 | 30 | 39 | 10 | F4 | 29 | 0F | 8D | CD |
| C348 | 4A | 2A | 85 | 4A | A5 | 49 | CA | 10 | 70 | C518 | D8 | CA | 88 | 30 | 2F | B9 | 9E | A1 | D9 |
| C350 | F3 | A5 | 4A | 91 | 4E | 88 | 10 | E8 | F6 | C520 | 29 | 0F | AA | E0 | 00 | F0 | 0C | AD | 54 |
| C358 | 4C | A9 | C4 | A5 | 42 | 48 | 20 | C2 | 1B | C528 | D8 | CA | 18 | 69 | 0A | 8D | D8 | CA | 48 |
| C360 | C4 | 85 | 42 | 20 | 50 | C5 | A0 | 07 | 85 | C530 | CA | D0 | F4 | 88 | 30 | 16 | B9 | 9E | 9A |
| C368 | B1 | 49 | 11 | 4E | 91 | 49 | 88 | 10 | EB | C538 | A1 | 29 | 0F | AA | E0 | 00 | F0 | 0C | A0 |
| C370 | F7 | 68 | 85 | 42 | 4C | 5E | C0 | 20 | 5F | C540 | AD | D8 | CA | 18 | 69 | 64 | 8D | D8 | 40 |
| C378 | 50 | C5 | A0 | 07 | A2 | 07 | B1 | 4E | 21 | C548 | CA | CA | D0 | F4 | AD | D8 | CA | 60 | 1B |
| C380 | 4A | 3E | E0 | CE | CA | 10 | F9 | 88 | 48 | C550 | A5 | F7 | 85 | 4E | A5 | F8 | 85 | 4F | C8 |
| C388 | 10 | F2 | A0 | 07 | B9 | E0 | CE | 91 | CD | C558 | A9 | 00 | A2 | 00 | E4 | 42 | F0 | 0B | D0 |
| C390 | 4E | 88 | 10 | F8 | 4C | A9 | C4 | 20 | 26 | C560 | 18 | 69 | 08 | D0 | 02 | E6 | 4F | E8 | 6C |
| C398 | 50 | C5 | A4 | FA | A5 | 02 | E0 | 64 | 54 | C568 | 4C | 5C | C5 | 85 | 4E | 60 | 20 | 50 | EE |
| C3A0 | F0 | 08 | 49 | FF | 31 | 4E | E0 | 64 | 91 | C570 | C5 | A9 | 0A | 85 | D6 | A9 | 05 | 85 | 4F |
| C3A8 | D0 | 02 | 11 | 4E | 91 | 4E | A5 | D3 | A5 | C578 | D3 | 20 | 10 | E5 | A0 | 00 | 84 | 41 | C0 |
| C3B0 | 85 | 45 | A5 | D6 | 85 | 46 | 20 | 6E | 00 | C580 | A9 | 80 | 85 | 3C | B1 | 4E | 25 | 3C | BE |
| C3B8 | C5 | A5 | 45 | 85 | D3 | A5 | 46 | 85 | CF | C588 | D0 | 05 | A9 | 20 | 4C | 91 | C5 | A9 | 2A |
| C3C0 | D6 | 20 | 10 | E5 | 4C | E8 | C0 | 20 | E5 | C590 | D1 | 84 | 41 | 20 | D2 | FF | A4 | 41 | A4 |
| C3C8 | 50 | C5 | A2 | 64 | 4C | 9A | C3 | A5 | 99 | C598 | A5 | 3C | C9 | 01 | F0 | 05 | 46 | 3C | 24 |
| C3D0 | 42 | 38 | E9 | 20 | 85 | 42 | 4C | 62 | FF | C5A0 | 4C | 84 | C5 | C8 | C0 | 08 | F0 | 10 | 5C |
| C3D8 | C1 | A5 | 42 | 18 | 69 | 20 | 85 | 42 | D1 | C5A8 | 84 | 41 | E6 | D6 | A9 | 05 | 85 | D3 | A8 |
| C3E0 | 4C | 62 | C1 | A0 | D8 | 84 | 60 | A0 | 1F | C5B0 | 20 | 10 | E5 | A4 | 41 | 4C | 80 | C5 | 01 |
| C3E8 | E0 | 84 | 5B | 4C | 8C | C6 | A5 | 42 | 94 | C5B8 | A0 | C9 | A9 | DA | 20 | A9 | C8 | A9 | 3D |
| C3F0 | 85 | 5F | 20 | C2 | C4 | C5 | 42 | B0 | 16 | C5C0 | 37 | 85 | 01 | A6 | 42 | A9 | 00 | 20 | 88 |
| C3F8 | 03 | 4C | 5E | C0 | 85 | 60 | E6 | 60 | 00 | C5C8 | CD | BD | A9 | 36 | 85 | 01 | 20 | D2 | 54 |
| C400 | 20 | 50 | C5 | AD | F2 | A1 | C9 | 31 | D0 | C5D0 | C5 | 60 | A0 | 07 | A2 | 17 | B1 | 4E | EA |
| C408 | F0 | 0A | A5 | 4F | 38 | E9 | 08 | 85 | 0E | C5D8 | 9D | 00 | A4 | CA | CA | CA | 88 | 10 | EF |
| C410 | 59 | 4C | 1B | C4 | A5 | 4F | 18 | 69 | 6A | C5E0 | F5 | 60 | A9 | 04 | 85 | 3D | 85 | 3F | C6 |
| C418 | 08 | 85 | 59 | A5 | 4E | 85 | 58 | A0 | B9 | C5E8 | A9 | A0 | 85 | 3E | A9 | D8 | 85 | 40 | 66 |
| C420 | 07 | B1 | 4E | 91 | 58 | 88 | 10 | F9 | FC | C5F0 | A2 | 00 | A0 | 00 | 8A | 91 | 3D | C5 | B2 |
| C428 | E6 | 42 | A5 | 60 | C5 | 42 | D0 | D0 | 6A | C5F8 | 42 | D0 | 05 | A9 | 00 | 4C | 02 | C6 | C0 |
| C430 | A5 | 5F | 85 | 42 | 4C | 5E | C0 | AD | 89 | C600 | A9 | 09 | 91 | 3F | E8 | C8 | C0 | 20 | B1 |
| C438 | D1 | CA | 18 | 69 | 08 | 8D | D1 | CA | 38 | C608 | D0 | EA | A9 | 28 | 18 | 65 | 3F | 85 | D4 |
| C440 | 20 | B7 | C8 | 4C | 8B | C0 | AD | D1 | 8C | C610 | 3F | 85 | 3D | 90 | 04 | E6 | 3E | E6 | DF |
| C448 | CA | 38 | E9 | 08 | 8D | D1 | CA | 20 | D8 | C618 | 40 | A9 | 44 | C5 | 3D | D0 | D3 | 60 | F7 |
| C450 | B7 | C8 | 4C | 8B | C0 | AD | D5 | CA | CE | C620 | A9 | 7B | 85 | 49 | A9 | CB | 85 | 4A | 83 |
| C458 | 8D | D4 | CA | A9 | 00 | 8D | 15 | D0 | 93 | C628 | A9 | 07 | 8D | 86 | 02 | A6 | 43 | A0 | A8 |
| C460 | A9 | 93 | 20 | D2 | FF | 20 | A3 | C7 | 1F | C630 | 00 | B1 | 49 | D0 | 03 | CA | F0 | 06 | ED |
| C468 | C9 | 03 | F0 | 06 | 20 | D2 | FF | 4C | B0 | C638 | 20 | 7D | C6 | 4C | 31 | C6 | 20 | 7D | 2C |
| C470 | 65 | C4 | A9 | 84 | 8D | D4 | CA | A9 | A0 | C640 | C6 | A9 | 09 | 85 | D6 | A9 | 15 | 85 | 5F |
| C478 | 01 | 8D | 15 | D0 | 4C | 5E | C0 | 8D | DB | C648 | D3 | 20 | 10 | E5 | A0 | 00 | B1 | 49 | E5 |
| C480 | D7 | CA | 38 | E9 | 31 | AA | BD | CD | D4 | C650 | F0 | 09 | 20 | D2 | FF | 20 | 7D | C6 | CA |
| C488 | CA | 85 | F8 | BD | CF | CA | 8D | D5 | 3C | C658 | 4C | 4C | C6 | 20 | 7D | C6 | E6 | D6 | 6C |
| C490 | CA | 4C | 6A | C0 | A5 | 43 | C9 | 01 | 51 | C660 | A9 | 17 | C5 | D6 | D0 | DF | A9 | 00 | 1A |
| C498 | F0 | 05 | A9 | 01 | 4C | A1 | C4 | A9 | 57 | C668 | 8D | 86 | 02 | A5 | F9 | 18 | 69 | 04 | AA |
| C4A0 | 0F | 85 | 43 | 20 | 20 | C6 | 4C | 8B | C9 | C670 | 85 | D3 | A5 | FA | 18 | 69 | 0A | 85 | 92 |
| C4A8 | C0 | A5 | D3 | 85 | 45 | A5 | D6 | 85 | 5A | C678 | D6 | 20 | 10 | E5 | 60 | E6 | 49 | D0 | D5 |
| C4B0 | 46 | 20 | 6E | C5 | A5 | 45 | 85 | D3 | E1 | C680 | 02 | E6 | 4A | 60 | A0 | D0 | 84 | 60 | FB |
| C4B8 | A5 | 46 | 85 | D6 | 20 | 10 | E5 | 4C | B6 | C688 | A0 | D8 | 84 | 5B | A2 | 00 | 86 | 5F | B9 |

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| C690 | A2 | 00 | 86 | 5A | A2 | 00 | 86 | 58 | 9D | C860 | A9 | 01 | 8D | 1A | D0 | 8D | 12 | D0 | 9F |
| C698 | A5 | F8 | 18 | 69 | 08 | 85 | 59 | 78 | C1 | C868 | 8D | F0 | CE | A9 | 1B | 8D | 11 | D0 | 92 |
| C6A0 | A9 | 33 | 85 | 01 | 20 | BF | A3 | A9 | 55 | C870 | A9 | 7C | 8D | 14 | 03 | A9 | C8 | 8Q | 3B |
| C6A8 | 36 | 85 | 01 | 58 | 4C | 6A | C0 | A9 | 54 | C878 | 15 | 03 | 58 | 60 | AD | 19 | D0 | 8D | E0 |
| C6B0 | 17 | 85 | D6 | A9 | 00 | 85 | D3 | 20 | ED | C880 | 19 | D0 | 29 | 01 | F0 | 1A | AE | F0 | 05 |
| C6B8 | 10 | E5 | A2 | D6 | 86 | BB | A0 | CA | 99 | C888 | CE | BD | D4 | CA | 8D | 18 | D0 | BD | A2 |
| C6C0 | 84 | BC | A9 | 01 | 85 | B7 | A9 | 08 | DF | C890 | D2 | CA | 8D | 12 | D0 | CA | 8E | F0 | 26 |
| C6C8 | 85 | BA | A9 | 60 | 85 | B9 | 20 | D5 | 2B | C898 | CE | 10 | 08 | A2 | 01 | 8E | F0 | CE | B9 |
| C6D0 | F3 | A5 | BA | 20 | B4 | FF | A5 | B9 | 88 | C8A0 | 4C | 31 | EA | 68 | A8 | 68 | AA | 68 | 1E |
| C6D8 | 20 | 96 | FF | A9 | 00 | 85 | 90 | A0 | 4C | C8A8 | 40 | 48 | A9 | 37 | 85 | 01 | 68 | 20 | 35 |
| C6E0 | 03 | 84 | FB | 20 | A5 | FF | 85 | FC | 89 | C8B0 | 1E | AB | A9 | 36 | 85 | 01 | 60 | AD | 1D |
| C6E8 | A4 | 90 | D0 | 45 | 20 | A5 | FF | A4 | 74 | C8B8 | D1 | CA | A2 | 00 | 4A | 4A | 4A | 4A | A9 |
| C6F0 | 90 | D0 | 3E | A4 | FB | 88 | D0 | E9 | B4 | C8C0 | 20 | C7 | C8 | E8 | AD | D1 | CA | 29 | B7 |
| C6F8 | A6 | FC | A0 | 37 | 84 | 01 | 20 | CD | E9 | C8C8 | 0F | C9 | 0A | B0 | 06 | 18 | 69 | 30 | C9 |
| C700 | BD | A9 | 36 | 85 | 01 | A9 | 20 | 20 | 1E | C8D0 | 4C | D6 | C8 | 38 | E9 | 09 | 9D | 3F | 43 |
| C708 | D2 | FF | 20 | A5 | FF | A6 | 90 | D0 | 67 | C8D8 | A2 | 60 | 0D | 08 | 81 | 93 | 11 | 11 | 35 |
| C710 | 20 | AA | F0 | 06 | 20 | D2 | FF | 4C | 61 | C8E0 | 11 | 11 | 11 | 11 | 11 | 11 | 11 | 1D | F4 |
| C718 | 0A | C7 | A0 | 17 | 84 | D6 | 20 | A3 | F8 | C8E8 | 1D | 1D | 1D | 1C | D5 | C0 | C0 | C0 | 16 |
| C720 | C7 | C9 | 03 | D0 | 03 | 4C | 31 | C7 | 53 | C8F0 | C0 | C0 | C0 | C0 | C0 | B2 | C0 | C0 | 0E |
| C728 | A9 | 0D | 20 | D2 | FF | A0 | 02 | D0 | 7B | C8F8 | C0 | B2 | C0 | C9 | 0D | 1D | 1D | 1D | F5 |
| C730 | B0 | 20 | 42 | F6 | 4C | 5E | C0 | A9 | E1 | C900 | 1D | DD | 20 | 20 | 20 | 20 | 20 | 20 | 41 |
| C738 | 17 | 85 | D6 | A9 | 00 | 85 | D3 | 20 | ED | C908 | 20 | 20 | DD | 20 | 20 | 20 | DD | 20 | 79 |
| C740 | 10 | E5 | A9 | 3E | 20 | D2 | FF | A9 | 3B | C910 | DD | 0D | 1D | 1D | 1D | DD | 20 | 8C | 8C |
| C748 | 0E | 8D | D9 | CA | 20 | C5 | C7 | A9 | 0A | C918 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | AB | 75 |
| C750 | 20 | CD | DC | CA | F0 | 1B | A9 | 01 | E2 | C920 | C0 | C0 | C0 | B1 | C0 | B3 | 0D | 1D | 0C |
| C758 | A2 | 08 | A0 | 0F | 20 | BA | FF | AD | E2 | C928 | 1D | 1D | 1D | DD | 20 | 20 | 20 | 20 | 41 |
| C760 | D9 | CA | A2 | DB | E8 | A0 | CA | 20 | C8 | C930 | 20 | 20 | 20 | 20 | DD | 42 | 2E | 23 | A6 |
| C768 | BD | FF | 20 | C0 | FF | A9 | 00 | 85 | F0 | C938 | 20 | 31 | DD | 0D | 1D | 1D | 1D | 1D | BF |
| C770 | 98 | 4C | 5E | C0 | A9 | 17 | 85 | D6 | 2A | C940 | DD | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 8D |
| C778 | A9 | 00 | 85 | D3 | 20 | 10 | E5 | A9 | A6 | C948 | 20 | DD | 20 | 20 | 20 | 20 | 20 | DD | 87 |
| C780 | 08 | 85 | BA | 20 | B4 | FF | A9 | 6F | 0C | C950 | 0D | 1D | 1D | 1D | DD | 20 | 20 | 31 | 31 |
| C788 | 85 | B9 | 20 | 96 | FF | 20 | A5 | FF | DB | C958 | 20 | 20 | 20 | 20 | 20 | 20 | DD | 24 | B9 |
| C790 | C9 | 0D | F0 | 06 | 20 | D2 | FF | 4C | A6 | C960 | 30 | 30 | 30 | 30 | DD | 0D | 1D | 1D | D0 |
| C798 | 8D | C7 | 20 | AB | FF | 20 | A3 | C7 | F6 | C968 | 1D | 1D | DD | 20 | 20 | 20 | 20 | 20 | 8D |
| C7A0 | 4C | 5E | C0 | A9 | 00 | 85 | D4 | A5 | AF | C970 | 20 | 20 | 20 | AB | C0 | C0 | C0 | C0 | 75 |
| C7A8 | C6 | 85 | CC | 8D | 92 | 02 | F0 | F3 | 0C | C978 | C0 | B3 | 0D | 1D | 1D | 1D | DD | 3D | 3D |
| C7B0 | 78 | A5 | CF | F0 | 0C | A5 | CE | AE | 3A | C980 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 |
| C7B8 | 87 | 02 | A0 | 00 | 84 | CF | 20 | 13 | E7 | C988 | DD | 32 | 23 | 20 | 20 | 20 | DD | 0D | 7D |
| C7C0 | EA | 20 | B4 | E5 | 60 | A9 | 20 | A2 | CE | C990 | 1D | 1D | 1D | 1D | DD | 20 | 20 | 20 | 8D |
| C7C8 | 0E | 9D | DB | CA | CA | D0 | FA | A2 | D4 | C998 | 20 | 20 | 20 | 20 | 20 | AB | C0 | C0 | 75 |
| C7D0 | 01 | 8E | DA | CA | 20 | A3 | C7 | C9 | FF | C9A0 | C0 | C0 | C0 | CB | 0D | 1D | 1D | 1D | E5 |
| C7D8 | 03 | D0 | 11 | A9 | 00 | 8D | D0 | CE | 12 | C9A8 | 1D | CA | C0 | C0 | C0 | C0 | C0 | C0 | 53 |
| C7E0 | A5 | 02 | C9 | 82 | 90 | 03 | 8D | D5 | 2D | C9B0 | C0 | C0 | CB | 0D | 11 | 1D | 1D | 1D | B2 |
| C7E8 | CA | 4C | 5E | C0 | C9 | 0D | D0 | 01 | A6 | C9B8 | 1D | 9E | 43 | 52 | 53 | 52 | 2F | 43 | 5E |
| C7F0 | 60 | C9 | 1D | D0 | 10 | AE | DA | CA | 57 | C9C0 | 4C | 52 | 2F | 48 | 4F | 4D | 45 | 0D | 1B |
| C7F8 | EC | D9 | CA | F0 | D7 | EE | DA | CA | E6 | C9C8 | 1D | 1D | 1D | 1D | 43 | 52 | 3A | 20 | 0B |
| C800 | E6 | D3 | 4C | D4 | C7 | C9 | 11 | D0 | CB | C9D0 | 50 | 2E | 20 | 53 | 45 | 2F | 4C | 4F | 02 |
| C808 | 0F | AD | DA | CA | C9 | 01 | F0 | C4 | 65 | C9D8 | 45 | 00 | 13 | 11 | 11 | 11 | 11 | 11 | 47 |
| C810 | CE | DA | CA | C6 | D3 | 4C | D4 | C7 | 8B | C9E0 | 11 | 11 | 11 | 11 | 11 | 1D | 1D | 1D | F4 |
| C818 | C9 | 20 | 90 | 08 | C9 | 60 | 90 | 07 | 21 | C9E8 | 1D | 1D | 1D | 1D | 1D | 1D | 1D | 1D | 00 |
| C820 | C9 | A0 | B0 | 03 | 4C | D4 | C7 | AE | 66 | C9F0 | 1D | 1D | 1D | 20 | 20 | 20 | 9D | 9D | FD |
| C828 | DA | CA | 9D | DB | CA | EE | DA | CA | BE | C9F8 | 9D | 00 | 5A | 45 | 49 | 43 | 48 | 45 | 8A |
| C830 | 20 | D2 | FF | A9 | 00 | 85 | D4 | AE | 46 | CA00 | 4E | 53 | 41 | 54 | 5A | 20 | 53 | 50 | 25 |
| C838 | D9 | CA | E8 | EC | DA | CA | D0 | 05 | E6 | CA08 | 45 | 49 | 43 | 48 | 45 | 52 | 4E | 00 | 38 |
| C840 | CE | DA | CA | C6 | D3 | A9 | DD | CD | 32 | CA10 | 5A | 45 | 49 | 43 | 48 | 45 | 4E | 53 | 19 |
| C848 | D0 | CE | D0 | 0B | AD | 00 | A0 | 85 | 8D | CA18 | 41 | 54 | 5A | 20 | 4C | 41 | 44 | 45 | 31 |
| C850 | 42 | 20 | 50 | C5 | 20 | D2 | C5 | 4C | 75 | CA20 | 4E | 00 | 0D | 1D | 1D | 1D | 1D | 1D | 3E |
| C858 | D4 | C7 | 78 | A9 | 7F | 8D | 0D | DC | 00 | CA28 | 1D | 1D | 1D | 1D | 4E | 41 | 4D | 45 | 15 |

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| CA30 | 3A | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 1A | CB60 | 08 | 85 | 45 | 90 | 02 | E6 | 46 | 4C | 50 |
| CA38 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 2E | F2 | CB68 | FF | CA | 4C | 00 | C0 | 0D | 53 | D9 | AD |
| CA40 | 5A | 9D | 9D | 9D | 9D | 9D | 9D | 9D | 8D | CB70 | 35 | 32 | 30 | 34 | 38 | 13 | 00 | 93 | 91 |
| CA48 | 9D | 9D | 9D | 9D | 9D | 9D | 9D | 9D | 00 | CB78 | 11 | 11 | 00 | 00 | 40 | 2F | 3A | 3B | 10 |
| CA50 | 9D | 00 | 03 | 13 | 1D | 9D | 11 | 91 | 8D | CB80 | 20 | 5A | 2E | 20 | 57 | 41 | 45 | 48 | E7 |
| CA58 | 3B | 3A | AC | BC | 0D | 5F | 5E | 2E | D0 | CB88 | 4C | 45 | 4E | 20 | 20 | 20 | 00 | 2C | 09 |
| CA60 | 3E | 2C | AE | B6 | 93 | 49 | 48 | 43 | 59 | CB90 | 3A | 4E | 52 | 2E | 20 | 44 | 45 | 5A | D7 |
| CA68 | 54 | 56 | 4F | 44 | 20 | A0 | B1 | 40 | FA | CB98 | 2E | 45 | 49 | 4E | 47 | 45 | 42 | 45 | E3 |
| CA70 | 2F | 89 | 8A | 87 | 2B | 2D | 5C | 31 | D2 | CBA0 | 4E | 00 | 2E | 3A | 5A | 45 | 49 | 43 | 5D |
| CA78 | 32 | 88 | 3D | AC | CA | E8 | 06 | 24 | 00 | CBA8 | 48 | 45 | 4E | 20 | 45 | 49 | 4E | 47 | 34 |
| CA80 | 42 | 60 | 68 | AF | 37 | 6D | 7B | 8F | 52 | CBB0 | 45 | 42 | 45 | 4E | 00 | 3E | 3A | 49 | AE |
| CA88 | A8 | D9 | E1 | F4 | 74 | C7 | D6 | E7 | 58 | CBB8 | 4E | 56 | 2E | 20 | 5A | 2E | 20 | 45 | 0D |
| CA90 | 01 | 1B | 3B | 5B | 77 | 97 | C7 | 74 | F9 | CBC0 | 49 | 4E | 47 | 45 | 42 | 45 | 4E | 00 | 48 |
| CA98 | CF | D9 | 84 | E3 | EE | 37 | 46 | 55 | 3F | CBC8 | 53 | 50 | 41 | 43 | 45 | 3A | 50 | 2E | 2E |
| CAA0 | 7F | 7F | EC | 94 | C0 | C0 | C0 | C1 | 56 | CBD0 | 20 | 4C | 4F | 45 | 53 | 43 | 48 | 45 | F1 |
| CAA8 | C1 | C1 | C1 | C1 | C6 | C7 | C1 | C1 | FF | CBD8 | 4E | 00 | 53 | 48 | 2E | 53 | 50 | 41 | 43 |
| CAB0 | C1 | C1 | C1 | C1 | C1 | C2 | C2 | C2 | FF | CBE0 | 43 | 45 | 3A | 50 | 2E | 20 | 53 | 45 | 04 |
| CAB8 | C2 | C3 | C3 | C3 | C3 | C3 | C3 | C3 | FF | CBE8 | 54 | 5A | 45 | 4E | 00 | 5E | 3A | 48 | 86 |
| CAC0 | C7 | C3 | C3 | C6 | C3 | C3 | C4 | C4 | 01 | CBF0 | 4F | 43 | 48 | 52 | 4F | 54 | 49 | 45 | 01 |
| CAC8 | C4 | C4 | C4 | CA | C4 | B0 | B8 | 8C | 3A | CBF8 | 52 | 45 | 4E | 20 | 20 | 20 | 20 | 00 | 5B |
| CAD0 | 8E | 20 | 01 | 78 | 84 | 8C | 24 | EA | 29 | CC00 | 5F | 3A | 4C | 49 | 4E | 4B | 53 | 52 | 2C |
| CAD8 | 00 | 00 | 00 | 24 | 20 | 20 | 20 | 20 | DD | CC08 | 4F | 54 | 49 | 45 | 52 | 45 | 4E | 20 | 3A |
| CAE0 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 | CC10 | 20 | 20 | 00 | 49 | 3A | 49 | 4E | 56 | A1 |
| CAE8 | 20 | 20 | 2E | 5A | 20 | B2 | C0 | A5 | 5D | CC18 | 45 | 4E | 54 | 49 | 45 | 52 | 45 | 4E | EC |
| CAF0 | F7 | 85 | 45 | A5 | F8 | 85 | 46 | A9 | 22 | CC20 | 20 | 20 | 20 | 00 | 48 | 3A | 48 | 4F | 27 |
| CAF8 | F6 | 85 | 4A | A9 | 18 | 85 | 49 | A9 | 46 | CC28 | 52 | 49 | 5A | 4F | 4E | 54 | 41 | 4C | 03 |
| CB00 | 77 | A0 | CB | 20 | 1E | AB | A6 | 49 | 52 | CC30 | 20 | 53 | 50 | 49 | 45 | 2E | 00 | 56 | 96 |
| CB08 | A5 | 4A | 20 | CD | BD | A9 | 44 | 20 | E6 | CC38 | 3A | 56 | 45 | 52 | 54 | 49 | 48 | 41 | EC |
| CB10 | D2 | FF | A9 | C1 | 20 | D2 | FF | A0 | 68 | CC40 | 4C | 20 | 53 | 50 | 49 | 45 | 47 | 2E | 4C |
| CB18 | 00 | A9 | 36 | 85 | 01 | B1 | 45 | 48 | 57 | CC48 | 00 | 44 | 3A | 39 | 30 | 27 | 20 | 52 | 95 |
| CB20 | A9 | 37 | 85 | 01 | 68 | AA | 98 | 48 | 03 | CC50 | 45 | 43 | 48 | 54 | 53 | 44 | 52 | 45 | 12 |
| CB28 | A9 | 00 | 20 | CD | BD | 68 | A8 | C8 | 31 | CC58 | 48 | 45 | 4E | 00 | 43 | 3A | 5A | 2E | 86 |
| CB30 | C0 | 08 | F0 | 08 | A9 | 2C | 20 | D2 | 69 | CC60 | 20 | 43 | 4F | 50 | 49 | 45 | 52 | 45 | ED |
| CB38 | FF | 4C | 19 | CB | A9 | 6D | A0 | CB | 12 | CC68 | 4E | 00 | 54 | 3A | 5A | 2E | 20 | 54 | 60 |
| CB40 | 20 | 1E | AB | A9 | 0D | 8D | 7A | 02 | FD | CC70 | 41 | 55 | 53 | 43 | 48 | 45 | 4E | 00 | 4D |
| CB48 | 8D | 7B | 02 | A9 | 05 | 85 | C6 | 60 | 52 | CC78 | 4F | 3A | 4F | 52 | 2D | 4D | 49 | 58 | E3 |
| CB50 | E6 | 49 | D0 | 02 | E6 | 4A | A5 | 49 | 61 | CC80 | 45 | 4E | 20 | 20 | 20 | 20 | 20 | 20 | F7 |
| CB58 | C9 | 18 | F0 | 0E | A5 | 45 | 18 | 69 | A0 | | | | | | | | | | |

SOFT-TEAM-BERLIN

Software und Zubehör für

C-64 Atari Schneider

Spectrum MSX

Über 2000 Programme immer auf Lager.
Ständig die neuesten Programme aus England und den USA.

Gratiskatalog anfordern:

**Anton Peter & Partner, Kamminerstraße 9
1000 Berlin 10, Telefon 453 27 11**

AS-A 2480 Akustikkoppler

- 300 Baud
- LED Anzeige
- Voll duplex
- Ohne FTZ-Nr.
- CCITT V. 21 Norm
- Answer-Originate
- Sichere Technik
- Made in Germany

AS-A 2480 mit V. 24 Schnittstelle, inkl. Netzteil **DM 184,-**

Telekommunikations-Komplettpaket
Akustikkoppler AS-A 2480 anschlussfertig + Netzteil und Terminalsoftware.

Für Commodore VC20/C64/PC128 **DM 188,-**

Für Apple IIe/II+ **DM 228,-**

Für Schneider CPC 464 **DM 248,-**

In der BRD einschließlich W-Berlin ohne fernmelderechtliche Genehmigung.

Preise inkl. MwSt. zzgl. NN plus Porto.
Bestellungen direkt an uns. Weitere In-
formationen kostenlos erhältlich; nur
anrufen oder Postkarte.



STOCKEM Computertechnik
Borghausen 13 - 5778 Meschede
Telefon (0291) 1221

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| CC88 | 20 | 20 | 00 | 5C | 3A | 5A | 2E | 2D | 86 | CD00 | 41 | 54 | 41 | 5A | 45 | 49 | 4C | 45 | D7 |
| CC90 | 50 | 52 | 4F | 42 | 45 | 20 | 20 | 20 | 30 | CD08 | 4E | 20 | 45 | 52 | 53 | 2E | 00 | 43 | 03 |
| CC98 | 20 | 20 | 20 | 20 | 20 | 00 | 2B | 2D | 1E | CD10 | 3D | 20 | 45 | 3A | 44 | 49 | 53 | 4B | 2B |
| CCA0 | 3A | 5A | 2D | 41 | 44 | 52 | 45 | 53 | B0 | CD18 | 53 | 54 | 41 | 54 | 55 | 53 | 20 | 20 | EE |
| CCA8 | 53 | 45 | 20 | 41 | 45 | 4E | 44 | 2E | FA | CD20 | 20 | 00 | 43 | 3D | 20 | 43 | 3A | 44 | F9 |
| CCB0 | 00 | 31 | 32 | 3A | 5A | 2D | 53 | 41 | 06 | CD28 | 49 | 53 | 4B | 42 | 45 | 46 | 45 | 4B | FB |
| CCB8 | 54 | 5A | 20 | 54 | 41 | 55 | 53 | 43 | C2 | CD30 | 4C | 20 | 20 | 20 | 00 | 43 | 3D | 20 | 06 |
| CCC0 | 48 | 45 | 4E | 00 | 46 | 32 | 3A | 4E | 51 | CD38 | 44 | 3A | 44 | 49 | 52 | 45 | 43 | 54 | 01 |
| CCC8 | 4F | 52 | 2E | 5A | 2E | 20 | 2D | 3E | CE | CD40 | 4F | 52 | 59 | 20 | 20 | 20 | 00 | 43 | F3 |
| CCD0 | 20 | 52 | 41 | 4D | 20 | 20 | 00 | 46 | 7D | CD48 | 3D | 20 | 53 | 3A | 5A | 2E | 20 | 53 | 2F |
| CCD8 | 34 | 3A | 4B | 4C | 45 | 2E | 5A | 2E | 3C | CD50 | 50 | 45 | 49 | 43 | 48 | 45 | 52 | 4E | 18 |
| CCE0 | 20 | 2D | 3E | 20 | 52 | 41 | 4D | 20 | 4F | CD58 | 20 | 00 | 43 | 3D | 20 | 4C | 3A | 5A | DA |
| CCE8 | 00 | 46 | 35 | 3A | 5A | 2E | 49 | 4E | DC | CD60 | 2E | 20 | 4C | 41 | 44 | 45 | 4E | 00 | 66 |
| CCF0 | 20 | 41 | 4E | 2E | 42 | 45 | 52 | 45 | 09 | CD68 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 00 |
| CCF8 | 49 | 43 | 48 | 00 | 46 | 37 | 3A | 44 | 53 | CD70 | 20 | 20 | 20 | 20 | 20 | 00 | 0D | 0D | 20 |

```

100 SYS9*4096
110 .OPT 00
120 *=#C000
130 ;
140 ;
150 ; ZEICHEN - EDITOR
160 ;
170 ; VON
180 ;
190 ; ANDREAS WELLIE'
200 ; SONDERFELD 8
210 ; 4300 ESSEN 14
220 ;
230 ; BELEGUNG DES SPEICHERS
240 ;
250 ; PROGRAMM #C000-#CD40
260 ;
270 ; BENUTZTE SPEICHERSTELLEN
280 ; IN DER ZERO-PAGE
290 ;
300 ; #02 AKTUELLES BIT
310 ; #3C VERSCHIEDEN
320 ; #3D- ZEICHENSATZ AUF BILDSCHIRM
330 ; #40
340 ; #41 VERSCHIEDEN
350 ; #42 NR. DES AKTUELLEN ZEICHENS
360 ; #43 STELLUNG DES MENUES
370 ; #49- ZEIGER ZUM Z.-COP,Z.-TAUSCH
380 ; #4A OR-VERKNUE.,VERTIKALSPIEGE.
390 ; & MENU AUSDRUCKEN
400 ; #4E- ADRESSE DES AKTUELLEN
410 ; #4F ZEICHENS
420 ; #F7- ADRESSE DES ZEICHENSATZES
430 ; #F8 IM SPEICHER
440 ; #F9 X-POSITION DES CURSORS
450 ; #FA Y-POSITION DES CURSORS
460 ; #FB IM SPEICHER
470 ;
480 ;
490 ;
500 ; INITIALISIEREN
510 ;
520 JSR RESET
530 LDA #*08 : STA #D020 ; FARBEN
540 ; STA #D021
550 LDA #*01 : STA #D015 ; SPRITE-
560 LDA #*00 : STA #D027 ; DATEN
570 LDA #152 : STA #D000 ; SETZEN
580 LDA #130 : STA #D001
590 LDA #*90 : STA #A3FB
600 LDX #63 : LDA #*00 ; SPRITE
610 MOBCLR STA #A400,X ; LOSCHEN
620 DEX : BPL MOBCLR

```

```

630 LDA #*80 : STA 650 ; KEYREPEAT
640 LDA #*00 : STA #9D ; PRGKONTROLLE
650 LDA #*00 : STA #42 ; ZEIGER
660 LDA #*01 : STA #43 ; ZEIGER
670 LDA #176 : STA #F8 ; FUER
680 LDA #*00 : STA #F7 ; Z-EDITOR
690 LDA #140 : STA SATZ+1
700 LDA #'1' : STA BEREI
710 LDA #160 : STA 640 ; VIDEOCHIP &
720 LDA #149 : STA #DD00 ; AUSGABE
730 ; AUF #8000
740 ;
750 JSR IRQIN ; RASTERINTERRUPT SETZEN
760 ;
770 ;
780 ; EDITIEREN
790 ;
800 ;
810 EDIT LDY #>TEXT2 ; BILD AUFBAUEN
820 LDA #<TEXT2
830 JSR ABDR
840 LDA #*00 : STA 646
850 EDIT1 LDA BEREI : STA #A1F2
860 LDX #*01 : STX #F9 ;CU, X-ACHSE
870 LDY #*00 : STY #FA ;CU, Y-ACHSE
880 LDA #*00 : STA #02
890 JSR ZSB ; ZEICHENS. AUSDRUCKEN
900 JSR ZDRUCKEN ; VERG. ZEICHEN AUSGE.
910 JSR HEX ; HEXZAHL AUSGEBEN
920 JSR MENU ; MENU AUSDRUCKEN
930 JMP HOME ; CURSOR HOME
940 ;
950 ; ZEICHEN VON TASTATUR HOLEN
960 ; & AUSWERTEN
970 ;
980 GETKEY JSR GETCHAR
990 LDX #40
1000 AUSW1 CMP BEFEHLE,X
1010 BEQ AUSW2 ;"BEFEHL ?
1020 DEX
1030 BPL AUSW1 ;"BEFEHLSLISTE ENDE ?
1040 JMP GETKEY
1050 AUSW2 TAY
1060 LDA LOW,X ;SPRUNGADRESSE LOW-BYTE
1070 STA JUMP+1
1080 LDA HI,X ;SPRUNGADRESSE HI-BYTE
1090 STA JUMP+2
1100 TYA
1110 JUMP JMP #0000 ; BEFEHL ANSPRINGEN
1120 ;
1130 ; ENDE
1140 ;
1150 B1 JSR RESET:JMP #E37B

```

```

1160 RESET LDA #55 : STA #01 ;ROM AN
1170 LDA #199 : STA #0000 ;AUSGABE AUF
1180 LDA #4 : STA #48 ;#0000
1190 JSR #FDA3;JSR #FD15;JSR #FF5B
1200 RTS
1210 ;
1220 ; CURSOR AUF HOMEPOSITION
1230 ;
1240 B2 NOP
1250 HOME LDA #*0A : STA #06
1260 LDA #*05 : STA #D3
1270 JSR 58640
1280 LDX #*01 : STX #F9
1290 LDY #*00 : STY #FA
1300 LDY #*00 : STY #02
1310 JSR ZSB
1320 JMP GETKEY
1330 ;
1340 ; CURSOR RECHTS
1350 ;
1360 B3 LDX #F9
1370 CPX #*08
1380 BNE CURE
1390 LDA #*00 : STA #02
1400 LDA #*01 : STA #F9
1410 LDA #*05 : STA #D3
1420 JMP GETKEY
1430 CURE INC #F9
1440 INC #D3
1450 LSR #02
1460 JMP GETKEY
1470 ;
1480 ; CURSOR LINKS
1490 ;
1500 B4 LDX #F9
1510 CPX #*01
1520 BNE CULI
1530 LDA #*01 : STA #02
1540 LDA #*08 : STA #F9
1550 LDA #*0C : STA #D3
1560 JMP GETKEY
1570 CULI DEC #F9
1580 DEC #D3
1590 ASL #02
1600 JMP GETKEY
1610 ;
1620 ; CURSOR UNTEN
1630 ;
1640 B5 LDX #FA
1650 CPX #*07
1660 BNE CUUN
1670 LDA #*00 : STA #FA
1680 LDA #*0A : STA #D6 : JSR 58640
1690 JMP GETKEY
1700 CUUN INC #D6;JSR 58640
1710 INC #FA
1720 JMP GETKEY
1730 ;
1740 ; CURSOR OBEN
1750 ;
1760 B6 LDX #FA
1770 CPX #*08
1780 BNE CUOB
1790 LDA #*07 : STA #FA
1800 LDA #*11 : STA #D6 : JSR 58640
1810 JMP GETKEY
1820 CUOB DEC #D6;JSR 58640
1830 DEC #FA
1840 JMP GETKEY
1850 ;
1860 ; ZEICHEN PLUS
1870 ;
1880 B7 INC #42
1890 ZAB JSR ZDRUCKEN
1900 JMP HOME
1910 ;
1920 ; ZEICHEN MINUS
1930 ;
1940 B8 DEC #42
1950 JMP ZAB
1960 ;
1970 ; PUNKT SETZEN B.Z.W. LOESCHEN
1980 ;
1990 B11 JSR ZEICHEN
2000 LDY #FA
2010 LDA (#4E),Y
2020 EOR #02
2030 STA (#4E),Y
2040 JMP ERNEU
2050 ;
2060 ; ROTIEREN NACH LINKS
2070 ;
2080 B12 JSR ZEICHEN
2090 LDY #*07
2100 ROR LDA (#4E),Y
2110 ASL
2120 PHP
2130 ROR
2140 PLP
2150 ROL
2160 STA (#4E),Y
2170 DEY
2180 BPL RORE
2190 JMP ERNEU
2200 ;
2210 ; ROTIEREN NACH OBEN
2220 ;
2230 B13 JSR ZEICHEN
2240 LDY #*00
2250 LDA (#4E),Y
2260 TAX
2270 ROUP INY
2280 LDA (#4E),Y
2290 DEY
2300 STA (#4E),Y
2310 INY
2320 CPY #*07
2330 BNE ROUP
2340 TXA
2350 STA (#4E),Y
2360 JMP ERNEU
2370 ;
2380 ; ZEICHEN AENDERN
2390 ;
2400 B14 LDA #"□" : JSR #FFD2
2410 LDA #"■" : JSR #FFD2
2420 LDA #42 : STA #A000
2430 LDA #*0D : STA #CED0
2440 LDA #*01 : STA LAENGE
2450 JSR INPUT
2460 LDA #A000 : STA #42
2470 LDA #*00 : STA #CED0
2480 STA #48
2490 LDA #"■" : JSR #FFD2
2500 JMP ZAB
2510 ;
2520 ; INVERSES ZEICHEN AENDERN
2530 ;
2540 B15 LDA #"■" : JSR #FFD2 : JMP B14
2550 ;
2560 ; ZEICHEN NR. DEZIMAL EINGEBEN
2570 ;
2580 B16 LDA #*0E : STA #D3
2590 LDA #*0A : STA #D6
2600 JSR 58640
2610 JSR DEZIMAL
2620 STA #42
2630 JMP ZAB
2640 ;
2650 ; ZEICHENSATZ SPEICHERN
2660 ;
2670 B17 LDA SATZ+1 : STA #02
2680 LDA #132 : STA SATZ+1
2690 LDA #*17 : STA #D6
2700 LDA #*09 : STA #D3
2710 JSR 58640

```

```

2720 LDA #<TEXT5
2730 LDY #>TEXT5
2740 JSR ABDR
2750 LDA #<TEXT7
2760 LDY #>TEXT7
2770 JSR ABDR
2780 LDA #14 : STA LAENGE
2790 JSR INPUT ; NAME EINGEBEN
2800 LDA #*01:LDX #*08:LDY #*61
2810 JSR #FFBA ; FILEPARAMETER
2820 LDA #*10
2830 LDX #<STRING: INK
2840 LDY #>STRING
2850 JSR #FFBD ; FILENAMENPARAMETER
2860 JSR #FFC0 ; OPEN
2870 LDA #*06 : JSR #FFB1 ; LISTEN
2880 LDA #*61 : JSR #FF93 ; SEK.ADRESSE
2890 LDA #*00 : JSR #FFA8
2900 LDA ORG1 : JSR #FFA8
2910 LDA #F7 : STA #4E
2920 LDA #F8 : STA #4F
2930 LDX #*08
2940 SV2 LDY #*00
2950 SV1 LDA (#4E),Y
2960 STX #60:JSR #FFA8 :LDX #60
2970 INY
2980 BNE SV1
2990 INC #4F
3000 DEX
3010 BNE SV2
3020 LDA #*01 : JSR #FFC3 ; CLOSE
3030 LDA #02 : STA SATZ+1
3040 JMP EDIT
3050 ;
3060 ; ZEICHENSATZ LADEN
3070 ;
3080 B18 LDA SATZ+1 : STA #02
3090 LDA #132 : STA SATZ+1
3100 LDA #*17 : STA #06
3110 LDA #*09 : STA #03
3120 JSR 58640
3130 LDA #<TEXT6
3140 LDY #>TEXT6
3150 JSR ABDR
3160 LDA #<TEXT7
3170 LDY #>TEXT7
3180 JSR ABDR
3190 LDA #14 : STA LAENGE
3200 JSR INPUT ; NAME EINGEBEN
3210 LDA #*02 : LDX #*08 : LDY #*00
3220 JSR #FFBA ; FILEPARAMETER
3230 LDA #*10
3240 LDX #<STRING : INK
3250 LDY #>STRING
3260 JSR #FFBD ; FILENAMENPARAMETER
3270 LDA #*00
3280 LDX #F7 : LDY #F8
3290 JSR #FFD5 ; LOAD
3300 LDA #02 : STA SATZ+1
3310 LDA #*00 : STA #42
3320 JMP EDIT
3330 ;
3340 ; ZEICHEN LOESCHEN
3350 ;
3360 B19 JSR ZEICHEN
3370 LDY #*07
3380 LDA #*00
3390 ZELO STA (#4E),Y
3400 DEY
3410 BPL ZELO
3420 JMP ZAB
3430 ;
3440 ; ZEICHEN INVENTIEREN
3450 ;
3460 B20 JSR ZEICHEN
3470 LDY #*07
3480 ZEII LDA (#4E),Y
3490 EOR #FF

3500 STA (#4E),Y
3510 DEY
3520 BPL ZEII
3530 JMP ERNEU
3540 ;
3550 ; ZEICHEN HORIZONTAL SPIEGELN
3560 ;
3570 B21 JSR ZEICHEN
3580 LDY #*07 : LDX #*00
3590 HOSPI LDA (#4E),X)
3600 PHA
3610 LDA (#4E),Y
3620 STA (#4E),X)
3630 PLA
3640 STA (#4E),Y
3650 INC #4E
3660 DEY
3670 DEY
3680 BPL HOSPI
3690 JMP ERNEU
3700 ;
3710 ; ZEICHEN COPIEREN
3720 ;
3730 B22 LDA #42 : PHA
3740 JSR ZWEIIN
3750 STA #42
3760 JSR ZEICHEN
3770 LDY #*07
3780 COPY LDA (#4E),Y
3790 STA (#49),Y
3800 DEY
3810 BPL COPY
3820 PLA
3830 STA #42
3840 JMP EDIT
3850 ;
3860 ; *ZEICHEN TAUSCHEN
3870 ;
3880 B23 LDA #42 : PHA
3890 JSR ZWEIIN
3900 STA #42
3910 JSR ZEICHEN
3920 LDY #*07
3930 TAUSCHEN LDA (#4E),Y : TAX
3940 LDA (#49),Y
3950 STA (#4E),Y
3960 TXA : STA (#49),Y
3970 DEY
3980 BPL TAUSCHEN
3990 PLA
4000 STA #42
4010 JMP EDIT
4020 ;
4030 ; ZEICHEN VERTIKAL SPIEGELN
4040 ;
4050 B24 JSR ZEICHEN
4060 LDY #*07
4070 VESPII LDA (#4E),Y
4080 LDX #*07
4090 VESPI LSR
4100 STA #49
4110 LDA #4A
4120 ROL
4130 STA #4A
4140 LDA #49
4150 DEX
4160 BPL VESPI
4170 LDA #4A
4180 STA (#4E),Y
4190 DEY
4200 BPL VESPII
4210 JMP ERNEU
4220 ;
4230 ; ZEICHEN OR VERKNUEPFEN
4240 ;
4250 B25 LDA #42 : PHA
4260 JSR ZWEIIN
4270 STA #42

```

| | |
|--|--|
| <pre> 4280 JSR ZEICHEN 4290 LDY #07 4300 ORVER LDA (#49),Y 4310 ORA (#4E),Y 4320 STA (#49),Y 4330 DEY 4340 BPL ORVER 4350 PLA 4360 STA #42 4370 JMP EDIT 4380 ; 4390 ; UM 90° NACH RECHTS DREHEN 4400 ; 4410 B26 JSR ZEICHEN 4420 LDY #07 4430 REDRE2 LDX #07 4440 LDA (#4E),Y 4450 REDRE1 LSR 4460 ROL #CEE0,X ;ZWISCHEN SPEICHERN 4470 DEX 4480 BPL REDRE1 4490 DEY 4500 BPL REDRE2 4510 LDY #07 4520 REDRE3 LDA #CEE0,Y ;ZURUECK- 4530 STA (#4E),Y ;COPIEREN 4540 DEY 4550 BPL REDRE3 4560 JMP ERNEU 4570 ; 4580 ; PUNKT LOESCHEN & CURSOR WEITER 4590 ; 4600 B27 JSR ZEICHEN 4610 PUME LDY #FA 4620 LDA #02 4630 CPX #100 4640 BEQ LOWE 4650 EOR #FF 4660 AND (#4E),Y 4670 CPX #100 4680 BNE LOWE1 4690 LOWE ORA (#4E),Y 4700 LOWE1 STA (#4E),Y 4710 LDA #03 ; STA #45 4720 LDA #06 ; STA #46 4730 JSR ZDRUCKEN 4740 LDA #45 ; STA #03 4750 LDA #46 ; STA #06 4760 JSR 58640 4770 JMP B3 4780 ; 4790 ; PUNKT SETZEN & CURSOR WEITER 4800 ; 4810 B28 JSR ZEICHEN 4820 LDX #100 4830 JMP PUME 4840 ; 4850 ; ZEICHEN EINE ZEILE HOCH 4860 ; 4870 B30 LDA #42 4880 SEC ; SBC #32 ; STA #42 4890 JMP ZAB 4900 ; 4910 ; ZEICHEN EINE ZEILE NACH UNTEN 4920 ; 4930 B31 LDA #42 4940 CLC ; ADC #32 ; STA #42 4950 JMP ZAB 4960 ; 4970 ; KLEINE ZEICHEN IN EDITORBEREICH 4980 ; COPIEREN 4990 ; 5000 B33 LDY #08 ; STY #60 5010 LDY #E0 ; STY #5B 5020 JMP ROMRAM1 5030 ; 5040 ; ZS IN ANDEREN BEREICH COPIEREN 5050 ; </pre> | <pre> 5060 B34 LDA #42 ; STA #5F 5070 JSR ZWEIIN 5080 CMP #42 5090 BCS ZC7 5100 JMP EDIT 5110 ZC7 STA #60 ; INC #60 5120 ZC5 JSR ZEICHEN:LDA #A1F2 5130 CMP #'1' 5140 BEQ ZC2 5150 LDA #4F ; SEC ; SBC #08 ; STA #59 5160 JMP ZC3 5170 ZC2 LDA #4F:CLC:ADC #08:STA #59 5180 ZC3 LDA #4E ; STA #58 5190 LDY #07 5200 ZC4 LDA (#4E),Y ; STA (#58),Y 5210 DEY ; BPL ZC4 5220 INC #42 ; LDA #60 ; CMP #42 5230 BNE ZC5 5240 LDA #5F ; STA #42 5250 JMP EDIT 5260 ; 5270 ; ZS STARTADRESSE + 5280 ; 5290 B35 LDA ORG1 5300 CLC ; ADC #08 5310 STA ORG1 5320 JSR HEX 5330 JMP GETKEY 5340 ; 5350 ; ZS STARTADRESSE - 5360 ; 5370 B36 LDA ORG1 5380 SEC ; SBC #08 5390 STA ORG1 5400 JSR HEX 5410 JMP GETKEY 5420 ; 5430 ; ZEICHENSATZ TEXTPROBE 5440 ; 5450 B37 LDA SATZ+1:STA SATZ 5460 LDA #08 ; STA 53248+21 5470 LDA #'J' ; JSR #FFD2 5480 TE2 JSR GETCHAR 5490 CMP #' ' 5500 BEQ TE1 5510 JSR #FFD2 5520 JMP TE2 5530 TE1 LDA #132 ; STA SATZ 5540 LDA #01 ; STA 53248+21 5550 JMP EDIT 5560 ; 5570 ; ANDEREN BEREICH WAELLEN 5580 ; 5590 B38 STA BERE1 5600 SEC ; SBC #31 5610 TAX 5620 LDA MEM,X 5630 STA #F8 5640 LDA POS1,X 5650 STA SATZ+1 5660 JMP EDIT1 5670 ; 5680 ; MENU-SEITE WECHSELN 5690 ; 5700 B40 LDA #43 5710 CMP #01 5720 BEQ MESC 5730 LDA #01 5740 JMP MESC1 5750 MESC LDA #0F 5760 MESC1 STA #43 5770 JSR MENU 5780 JMP GETKEY 5790 ; 5800 ; UNTERPROGRAMME 5810 ; "_____ 5820 ; 5830 ; VERGROESSERTES ZEICHEN DRUCKEN </pre> |
|--|--|

```

5840 ; & CURSORPOSITION ZWISCHEN-
5850 ; SPEICHERN
5860 ;
5870 ERNEU LDA #D3 : STA #45
5880 LDA #D6 : STA #46
5890 JSR ZDRUCKEN
5900 LDA #45 : STA #D3
5910 LDA #46 : STA #D6
5920 JSR 58640
5930 JMP GETKEY
5940 ;
5950 ; NR. EINES ZWEITEN ZEICHENS EIN
5960 ;
5970 ZWEIIN JSR ZEICHEN
5980 LDA #4E : STA #49
5990 LDA #4F : STA #4A
6000 ZWEIINI LDA #16 : STA #D3
6010 LDA #16 : STA #D6
6020 JSR 58640
6030 LDA # " " : STA STRING+1
6040 STA STRING+2 : STA STRING+3
6050 LDA #*03 : STA LAENGE
6060 JSR INPUT1
6070 LDX #*03
6080 TRANS LDA STRING,X
6090 STA 41373,X
6100 DEX
6110 BNE TRANS
6120 JSR DEZIMAL2
6130 RTS
6140 ;
6150 ; DEZIMALZAHL EINGEBEN
6160 ; & NACH HEX UMRECHNEN
6170 ;
6180 DEZIMAL LDA #*03 : STA LAENGE
6190 JSR INPUT
6200 DEZIMAL2 LDA #42 : STA COUNTER
6210 LDY #*02
6220 DEZ11 LDA 41374,Y
6230 CMP #*20
6240 BNE DEZ12
6250 DEY
6260 BMI TAU
6270 BPL DEZ11
6280 DEZ12 AND #*0F
6290 STA COUNTER
6300 DEY
6310 BMI TAU
6320 LDA 41374,Y
6330 AND #*0F
6340 TAX
6350 CPX #*00
6360 BEQ DEZ14
6370 DEZ13 LDA COUNTER
6380 CLC
6390 ADC #*0A
6400 STA COUNTER
6410 DEX
6420 BNE DEZ13
6430 DEZ14 DEY
6440 BMI TAU
6450 LDA 41374,Y
6460 AND #*0F
6470 TAX
6480 CPX #*00
6490 BEQ TAU
6500 WAHLS LDA COUNTER
6510 CLC
6520 ADC #100
6530 STA COUNTER
6540 DEX
6550 BNE WAHLS
6560 TAU LDA COUNTER
6570 RTS
6580 ;
6590 ; ADRESSE DES AKTUELLEN
6600 ; ZEICHENS ERRECHNEN
6610 ;

```

```

6620 ZEICHEN LDA #F7 : STA #4E
6630 LDA #F8 : STA #4F
6640 LDA #*00
6650 LDX #*00
6660 ZE11 CPX #42
6670 BEQ ZE13
6680 CLC : ADC #*08
6690 BNE ZE12
6700 INC #4F
6710 ZE12 INX
6720 JMP ZE11
6730 ZE13 STA #4E
6740 RTS
6750 ;
6760 ; ZEICHEN VERGROESSERT AUSGEBEN
6770 ;
6780 ZDRUCKEN JSR ZEICHEN
6790 LDA #*0A : STA #D6
6800 LDA #*05 : STA #D3
6810 JSR 58640
6820 LDY #*00
6830 STY #41
6840 SETBIT LDA #*80
6850 STA #3C
6860 GETBIT LDA (&4E),Y
6870 AND #3C
6880 BNE PUNKT
6890 LDA # " "
6900 JMP DRUCKE
6910 PUNKT LDA #*"#"
6920 DRUCKE STY #41
6930 JSR #FFD2
6940 LDY #41
6950 LDA #3C
6960 CMP #*01
6970 BEQ CR
6980 LSR #3C
6990 JMP GETBIT
7000 CR INY
7010 CPY #*00
7020 BEQ STOPP
7030 STY #41
7040 INC #D6
7050 LDA #*05 : STA #D3
7060 JSR 58640
7070 LDY #41
7080 JMP SETBIT
7090 STOPP LDY #>TEXT4 ;HI-BYTE TEXT4
7100 LDA #<TEXT4
7110 JSR ABDR
7120 LDA #55 : STA #01
7130 LDX #42 : LDA #*00
7140 JSR #BDCD
7150 LDA #54 : STA #01
7160 JSR SPRITON
7170 RTS
7180 ;
7190 ; ZEICHEN IN SPRITE COPIEREN
7200 ;
7210 SPRITON LDY #*07 : LDX #23
7220 SPRITSE LDA (&4E),Y : STA #A400,X
7230 DEX : DEX : DEX
7240 DEY
7250 BPL SPRITSE
7260 RTS
7270 ;
7280 ; ZEICHENSATZ AUF BILDSCHIRM
7290 ; DRUCKEN
7300 ;
7310 ZSB LDA #*04 : STA #3D
7320 : STA #3F
7330 LDA #*A0 : STA #3E
7340 LDA #*D8 : STA #40
7350 LDX #*00
7360 ZSB1 LDY #*00
7370 ZSB2 TXA
7380 STA (&3D),Y
7390 CMP #42

```

```

7400 BNE ZSB4
7410 LDA ##00
7420 JMP ZSB5
7430 ZSB4 LDA ##09
7440 ZSB5 STA (*3F),Y
7450 INX
7460 INY
7470 CPY #32
7480 BNE ZSB2
7490 LDA ##28
7500 CLC
7510 ADC #3F
7520 STA #3F
7530 STA #3D
7540 BCC ZSB3
7550 INC #3E
7560 INC #40
7570 ZSB3 LDA ##44
7580 CMP #3D
7590 BNE ZSB1
7600 RTS
7610 ;
7620 ; MENUE DRUCKEN
7630 ;
7640 MENU LDA #(TEXT1 ; STA #49
7650 LDA #)TEXT1 ; STA #4A
7660 LDA ##07 ; STA #46
7670 LDX #43
7680 LDY ##00
7690 MENU1 LDA (*49),Y
7700 BNE MENU2
7710 DEX
7720 BEQ MENU4
7730 MENU2 JSR MENU3
7740 JMP MENU1
7750 MENU4 JSR MENU3
7760 LDA ##09 ; STA #06
7770 MENU5 LDA ##15 ; STA #03
7780 JSR 58640
7790 MENU6 LDY ##00
7800 LDA (*49),Y
7810 BEQ MENU7
7820 JSR #FFD2
7830 JSR MENU3
7840 JMP MENU6
7850 MENU7 JSR MENU3
7860 INC #06
7870 LDA #23
7880 CMP #06
7890 BNE MENU5
7900 LDA ##00 ; STA #46
7910 LDA #F9 ; CLC ; ADC ##04
7920 STA #03
7930 LDA #FA ; CLC ; ADC ##0A
7940 STA #06
7950 JSR 58640
7960 RTS
7970 MENU3 INC #49
7980 BNE MENU3
7990 INC #4A
8000 MENU3 RTS
8010 ;
8020 ; ZEICHENSATZ ROM -> RAM
8030 ;
8040 B32 LDY ##00 ; STY #60
8050 LDY ##08 ; STY #5B
8060 ROMRAM1 LDX ##00 ; STX #5F
8070 LDX ##00 ; STX #5A
8080 LDX ##00 ; STX #58
8090 LDA #FB
8100 CLC ; ADC ##08
8110 STA #59
8120 SEI
8130 LDA ##33 ; STA #01
8140 JSR #A3BF
8150 LDA #54 ; STA #01
8160 CLI
8170 JMP EDIT1

8180 ;
8190 ; DIRECTORY LESEN
8200 ;
8210 B9 LDA ##17 ; STA #06
8220 LDA ##00 ; STA #03
8230 JSR 58640
8240 LDX #<DIRE:STX #8B
8250 LDY #>DIRE:STY #8C
8260 LDA ##01:STA #87
8270 LDA ##08
8280 STA #8A
8290 LDA ##60
8300 STA #89
8310 JSR #F3D5
8320 LDA #8A
8330 JSR #FFB4
8340 LDA #89
8350 JSR #FF96
8360 LDA ##00
8370 STA #90
8380 LDY ##03
8390 DIR1 STY #FB
8400 JSR #FFA5
8410 STA #FC
8420 LDY #90
8430 BNE DIR4
8440 JSR #FFA5
8450 LDY #90
8460 BNE DIR4
8470 LDY #FB
8480 DEY
8490 BNE DIR1
8500 LDX #FC
8510 LDY #55 ; STY #01
8520 JSR #80CD
8530 LDA #54 ; STA #01
8540 LDA ##20
8550 JSR #FFD2
8560 DIR3 JSR #FFA5
8570 LDX #90
8580 BNE DIR4
8590 TAX
8600 BEQ DIR2
8610 JSR #FFD2
8620 DIR7 JMP DIR3
8630 DIR2 LDY ##17 ; STY #06
8640 JSR GETCHAR
8650 CMP ##03
8660 BNE DIR5
8670 JMP DIR4
8680 DIR5 LDA #13
8690 JSR #FFD2
8700 LDY ##02
8710 BNE DIR1
8720 DIR4 JSR #F642
8730 JMP EDIT
8740 ;
8750 ; DISK - BEFEHL
8760 ;
8770 B10 LDA ##17 ; STA #06
8780 LDA ##00 ; STA #03
8790 JSR 58640
8800 LDA #")" ; JSR #FFD2
8810 LDA #14 ; STA LAENGE
8820 JSR INPUT
8830 LDA # " "
8840 CMP STRING+1
8850 BEQ COMEND
8860 LDA ##01:LDX ##08:LDY ##0F
8870 JSR #FFBA ; FILEPARAMETER
8880 LDA LAENGE
8890 LDX #<STRING:INX
8900 LDY #>STRING
8910 JSR #FFBD ; FILENAMENPARAMETER
8920 JSR #FFC0 ; OPEN
8930 LDA ##00 ; STA #152 ; KANAL ZU
8940 COMEND JMP EDIT
8950 ;

```



```

8960 ;" DISK STATUS
8970 ;
8980 B29 LDA #17 : STA #06
8990 LDA #00 : STA #03
9000 JSR 58640
9010 LDA #00 : STA #8A
9020 JSR #FFB4
9030 LDA #6F : STA #89
9040 JSR #FF96
9050 STATUS1 JSR #FFA5
9060 CMP #00
9070 BEQ STATUS2
9080 JSR #FFD2
9090 JMP STATUS1
9100 STATUS2 JSR #FFAB
9110 JSR GETCHAR ;"TASTE ?
9120 JMP EDIT
9130 ;
9140 ; CUSOR BLINKT UND WARTET
9150 ; AUF EINGABE EINES ZEICHENS
9160 ; ZEICHEN WIRD IM AKKU UEBERGEBEN
9170 ;
9180 GETCHAR LDA #00:STA 212
9190 LDA #C6
9200 STA #CC
9210 STA #0292
9220 BEQ GETCHAR ;"TASTE GEDRUECKT ?
9230 SEI
9240 LDA #CF
9250 BEQ PUFFER ;"CURSOR IN BLINKPHASE?
9260 LDA #CE
9270 LDX #0207
9280 LDY #00
9290 STY #CF
9300 JSR #EA13
9310 PUFFER JSR #E5B4
9320 RTS
9330 ;
9340 ; INPUT - ROUTINE
9350 ;
9360 INPUT LDA # " :LDX #14
9370 FILLST STA STRING,X
9380 DEX : BNE FILLST
9390 INPUT1 LDX #01:STX STELLUNG
9400 TASTE JSR GETCHAR
9410 CMP #3 ;" RUN STOP ?
9420 BNE RETURN
9430 LDA #00 : STA #CED0
9440 LDA #02 : CMP #130
9450 BCC INSTOP
9460 STA SATZ+1
9470 INSTOP JMP EDIT
9480 RETURN CMP #13 ;" RETURN ?
9490 BNE RECHTS
9500 RTS
9510 RECHTS CMP # " ;" RECHTS ?
9520 BNE LINKS
9530 LDX STELLUNG
9540 CPX LAENGE
9550 BEQ TASTE
9560 INC STELLUNG
9570 INC #03
9580 JMP TASTE
9590 LINKS CMP # " ;" LINKS ?
9600 BNE NUM
9610 LDA STELLUNG
9620 CMP #01
9630 BEQ TASTE
9640 DEC STELLUNG
9650 DEC #03
9660 JMP TASTE
9670 NUM CMP #32 : BCC ALL
9680 CMP #96 : BCC AN
9690 CMP #160 : BCS AN
9700 ALL JMP TASTE
9710 AN LDX STELLUNG
9720 STA STRING,X
9730 INC STELLUNG
9740 JSR #FFD2
9750 LDA #00 : STA 212
9760 LDX LAENGE : INK
9770 CPX STELLUNG:BNE TW
9780 DEC STELLUNG:DEC #03
9790 TW LDA #00
9800 CMP #CED0
9810 BNE TW2
9820 LDA #A000 : STA #42
9830 JSR ZEICHEN
9840 JSR SPRITON
9850 TW2 JMP TASTE
9860 ;
9870 ;
9880 ; INTERRUPT AENDERN
9890 ;
9900 IRQEIN SEI
9910 LDA #7F
9920 STA #DC00
9930 LDA #01
9940 STA #001A
9950 STA #0012
9960 STA #CEFO
9970 LDA #1B
9980 STA #0011
9990 LDA #<START
10000 STA #0314
10010 LDA #>START
10020 STA #0315
10030 CLI
10040 RTS
10050 ;
10060 ; INTERRUPTROUTINE
10070 ;
10080 START LDA #0019:STA #0019
10090 AND #01
10100 BEQ ROM
10110 LDX #CEFO
10120 LDA SATZ,X
10130 STA #0010
10140 LDA ZEILE,X
10150 STA #0012
10160 DEX
10170 STX #CEFO
10180 BPL END
10190 LDX #01
10200 STX #CEFO
10210 ROM JMP #EA31
10220 END PLA
10230 TAY
10240 PLA
10250 TAX
10260 PLA
10270 RTI
10280 ;
10290 ; ROM AN & TEXT DRUCKEN
10300 ;
10310 ABDR PHA: LDA #55 : STA #01:PLA
10320 JSR #AB1E
10330 LDA #54 : STA #01
10340 RTS
10350 ;
10360 ; HEXADEZIMAL AUSDRUCKEN
10370 ;
10380 HEX LDA ORGI
10390 LDX #00
10400 LSR :LSR :LSR :LSR
10410 JSR MAKE
10420 INK
10430 LDA ORGI
10440 MAKE AND #0F
10450 CMP #0A
10460 BCS HEX1
10470 CLC
10480 ADC #40
10490 JMP HEX2
10500 HEX1 SEC
10510 SBC #09

```

```

10520 HEX2 STA #A23F,X
10530 RTS
10540 ;
10550 ; TEXTE
10560 ;
10570 TEXT2 .BYTE 13,01.ASC "IXM"
10580 .ASC "XXXXXXXXXXXX"
10590 .BYTE 13:ASC "XXXXX"
10600 .BYTE 13:ASC "XXXXX"
10610 .BYTE 13:ASC "XXXXX"
10620 .BYTE 13:ASC "XXXXX"
10630 .BYTE 13:ASC "XXXXX"
10640 .BYTE 13:ASC "XXXXX"
10650 .BYTE 13:ASC "XXXXX"
10660 .BYTE 13:ASC "XXXXX"
10670 .BYTE 13:ASC "XXXXX"
10680 .BYTE 13:ASC "XXXXXXXXCRSR/CLR/HOME"
10690 .BYTE 13:ASC "XXXXXCR: P. SE/LOE"
10700 .BYTE 0
10710 ;
10720 ;
10730 TEXT4 .ASC "XXXXXXXXXXXXXXXXXXXX"
10740 .ASC "XXXX" : .BYTE 0
10750 ;
10760 TEXT5 .ASC "ZEICHENSATZ SPEICHERN"
10770 .BYTE 0
10780 ;
10790 TEXT6 .ASC "ZEICHENSATZ LADEN"
10800 .BYTE 0
10810 ;
10820 TEXT7 .BYTE 13:ASC "XXXXXXXXXXXXNAME:"
10830 .ASC " " .Z"
10840 .ASC "XXXXXXXXXXXX"
10850 .BYTE 0
10860 ;
10870 ;
10880 ; DATEN
10890 ;
10900 BEFEHLE .ASC "XXXXXXXXX":.BYTE 13
10910 .ASC "++.>.,- IJHCTVOD"
10920 .ASC " -0/XXXX+-E12"
10930 LOW .BYTE <B1,<B2,<B3,<B4,<B5
10940 .BYTE <B6,<B7,<B8,<B9,<B10
10950 .BYTE <B11,<B12,<B13,<B14,<B15
10960 .BYTE <B16,<B17,<B18,<B19,<B20
10970 .BYTE <B21,<B22,<B23,<B24,<B25
10980 .BYTE <B26,<B27,<B28,<B29,<B30
10990 .BYTE <B31,<B32,<B33,<B34,<B35
11000 .BYTE <B36,<B37,<B38,<B38,<B39
11010 .BYTE <B40
11020 HI .BYTE >B1,>B2,>B3,>B4,>B5
11030 .BYTE >B6,>B7,>B8,>B9,>B10
11040 .BYTE >B11,>B12,>B13,>B14,>B15
11050 .BYTE >B16,>B17,>B18,>B19,>B20
11060 .BYTE >B21,>B22,>B23,>B24,>B25
11070 .BYTE >B26,>B27,>B28,>B29,>B30
11080 .BYTE >B31,>B32,>B33,>B34,>B35
11090 .BYTE >B36,>B37,>B38,>B38,>B39
11100 .BYTE >B40
11110 MEM .BYTE 176,184
11120 POS1 .BYTE 140,142
11130 ORG1 .BYTE 32
11140 ZEILE .BYTE 1,120
11150 SATZ .BYTE 132,140
11160 DIRE .ASC "*"
11170 BERE1 NOP
11180 COUNTER .BYTE 0
11190 LAENGE .BYTE 0
11200 STELLUNG .BYTE 0
11210 STRING .ASC "*" .Z"
11220 ;
11230 ; DATAZEILEN VON ZEICHENSATZ
11240 ;
11250 B39 JSR RESET
11260 LDA #F7 : STA #45
11270 LDA #F8 : STA #46
11280 LDA #F6 : STA #4A
11290 LDA #18 : STA #49
11300 DAMA1 LDA #<CUHO:LDY #>CUHO
11310 JSR #ABIE
11320 LDX #49 : LDA #4A
11330 JSR #BDCD
11340 LDA #D" : JSR #FFD2
11350 LDA #*# : JSR #FFD2
11360 LDY #30
11370 DAMA2 LDA #54:STA #01
11380 LDA (#45),Y:
11390 PHA:LDA #55:STA #01:PLA
11400 TAX
11410 TYA : PHA
11420 LDA #30
11430 JSR #BDCD
11440 PLA : TAY
11450 INY : CPY #30
11460 BEQ DAMA3
11470 LDA #", " : JSR #FFD2
11480 JMP DAMA2
11490 DAMA3 LDA #<SYS : LDY #>SYS
11500 JSR #ABIE
11510 LDA #13 : STA 634 : STA 635
11520 LDA #305: STA 198
11530 RTS
11540 NEXT INC #49
11550 BNE DAMA4
11560 INC #4A
11570 DAMA4 LDA #49
11580 CMP #18
11590 BEQ DAMA6
11600 LDA #45
11610 CLC : ADC #30
11620 STA #45
11630 BCC DAMA5
11640 INC #46
11650 DAMA5 JMP DAMA1
11660 DAMA6 JMP #C000
11670 ;
11680 SYS .BYTE 13:ASC "S 52048" : .BYTE 0
11690 CUHO .ASC "XXXX" : .BYTE 0
11700 ;
11710 ;
11720 ;
11730 TEXT1 .BYTE 0
11740 .ASC "@/:; Z. WAELHEN " : .BYTE 0
11750 .ASC ",:NR. DEZ.EINGEBEN" : .BYTE 0
11760 .ASC " : ZEICHEN EINGEBEN" : .BYTE 0
11770 .ASC ">:INV. Z. EINGEBEN" : .BYTE 0
11780 .ASC "SPACE:P. LOESCHEN" : .BYTE 0
11790 .ASC "SH.SPACE:P. SETZEN" : .BYTE 0
11800 .ASC "↑:HOCHROTIEREN " : .BYTE 0
11810 .ASC "←:LINKSROTIEREN " : .BYTE 0
11820 .ASC "I:INVENTIEREN " : .BYTE 0
11830 .ASC "H:HORIZONTAL SPIEG." : .BYTE 0
11840 .ASC "V:VERTIKAL SPIEG." : .BYTE 0
11850 .ASC "D:90' RECHTSDREHEN" : .BYTE 0
11860 .ASC "C:Z. COPIEREN" : .BYTE 0
11870 .ASC "T:Z. TAUSCHEN" : .BYTE 0
11880 .ASC "O:OR-MIXEN " : .BYTE 0
11890 .ASC "E:Z.-PROBE " : .BYTE 0
11900 .ASC "+-:Z-ADRESSE AEND." : .BYTE 0
11910 .ASC "12:Z-SATZ TAUSCHEN" : .BYTE 0
11920 .ASC "F2:NR.Z. -> RAM " : .BYTE 0
11930 .ASC "F4:KLE.Z. -> RAM " : .BYTE 0
11940 .ASC "F5:Z. IN AN.BEREICH" : .BYTE 0
11950 .ASC "F7:DATAZEILEN ERS." : .BYTE 0
11960 .ASC "C= E:DISKSTATUS " : .BYTE 0
11970 .ASC "C= C:DISKBEFEHL " : .BYTE 0
11980 .ASC "C= D:DIRECTORY " : .BYTE 0
11990 .ASC "C= S:Z. SPEICHERN " : .BYTE 0
12000 .ASC "C= L:Z. LADEN" : .BYTE 0
12010 .ASC " " : .BYTE 0
READY.

```



Das RUN Geschenk- Abo!

Macht Spaß, ist nützlich und kommt 12 mal pro Jahr ins Haus. Ein Geschenk mit bestem Erinnerungswert.



Eine Erste Adresse für die Computerwelt.

Am einfachsten bestellen Sie mit diesem Coupon. Ausfüllen, abtrennen und ausreichend frankiert im Kuvert ab die Post an:

CW-Publikationen
Vertrieb RUN
Postfach 40 04 29
D-8000 München 40

RUN Geschenk-Abo Bestellcoupon

Ich verschenke RUN zum Bezugspreis von DM 58,- (12 Ausgaben), inkl. MwSt. und Lieferung „frei Haus“. Auslandsendpreis: DM/sfr 66,-. Luftpostversand auf Anfrage.

Adresse des Bestellers (Rechnungsanschrift):

Name Vorname

Straße, Nr./Postfach

PLZ Wohnort

Datum Unterschrift

Adresse des Beschenkten:

Name Vorname

Straße, Nr./Postfach

PLZ Wohnort

Laufzeit des Geschenkabonnements:

- Das Abonnement soll sich nach einem Jahr automatisch um weitere 12 Monate verlängern, wenn es nicht acht Wochen vor Ablauf gekündigt wird.
- Das Abonnement soll auf ein Jahr (12 Ausgaben) begrenzt sein.

Garantie:

Ich weiß, daß ich diese Vereinbarung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs. Ich bestätige dies durch meine zweite Unterschrift.

Gewünschte Zahlungsweise:

- Bequem und bargeldlos durch Bankabbuchung

Bankleitzahl (vom Scheck abschreiben) Geldinstitut

Kontonummer/Inhaber

- Gegen Rechnung, zahlbar sofort nach Erhalt

Hardcopy groß rausgebracht

Das Problem ist sicherlich schon jedem einmal begegnet, der eine hochauflösende Grafik ausdrucken wollte: In bezug auf den Mal-Komfort bieten die heutzutage auf dem Markt befindlichen Malprogramme höchste Leistungen. Doch kaum ein Malprogramm besitzt auch eine gute Hardcopy-Routine. Bei Dia-Shows liegt die Sache ähnlich. Auch hier: Weit und breit keine für diesen speziellen Fall passende Druckroutine zur Hand. Etwas besser hatten es da nur die Besitzer von SIMON's Basic mittels des implementierten COPY-Befehls. Doch abgesehen von der bescheidenen Bildgröße: Dia-Show-Bilder auszudrucken war immer noch kompliziert genug: Originalbild auf andere Disk kopieren, mit einem Diskmonitor die Startadresse auf \$E000 ändern, SIMON's Basic laden ... und so weiter.

Laden und Drucken von Diskette

Doch damit ist jetzt Schluß: das Programm ENLARGED PRINT lädt und druckt jede als Programmfile abgespeicherte Grafik (von der Diskette) aus. Mit den Druckern MPS801, VC1525 und kompatible (Seikosha) sowie mit Epson-kompatiblen Druckern, die mittels Merlin-Interface angeschlossen wurden, gibt es kein Problem. Außerdem arbeitet ENLARGED PRINT mit den meisten Basic-Erweiterungen zusammen, da die Maschinenprogramme im Kassettenpuffer und im freien Basic-Speicher ab dez.10496 untergebracht werden und kein Zwischenspeicher benötigt wird. Das ist nur möglich, weil die errechneten Bytes unmittelbar zum Drucker gesendet werden. Vor dem ersten Lauf sollte das Programm unbedingt abgespeichert werden,

*Malprogramme haben
eines gemeinsam: Sie
kümmern sich wenig
um die Hardcopy. EN-
LARGED PRINT gleicht
diesen Mangel aus und
setzt noch eins drauf:
die Vergrößerung.*

da es die Zeiger auf das Programm-Ende verändert. Das ist wichtig, um die Print-Routine vor dem Überschreiben durch Variablen zu schützen.

Nach dem Start liest der Computer die DATAs, die gleichzeitig durch Prüfsummen auf ihre Korrektheit untersucht werden, und blendet dann ein Menü ein. Man hat nun die Möglichkeit, entweder F1 (INVERT), F3 (LOAD), F5 (DIRECTORY) oder RETURN (PRINT) zu drücken.

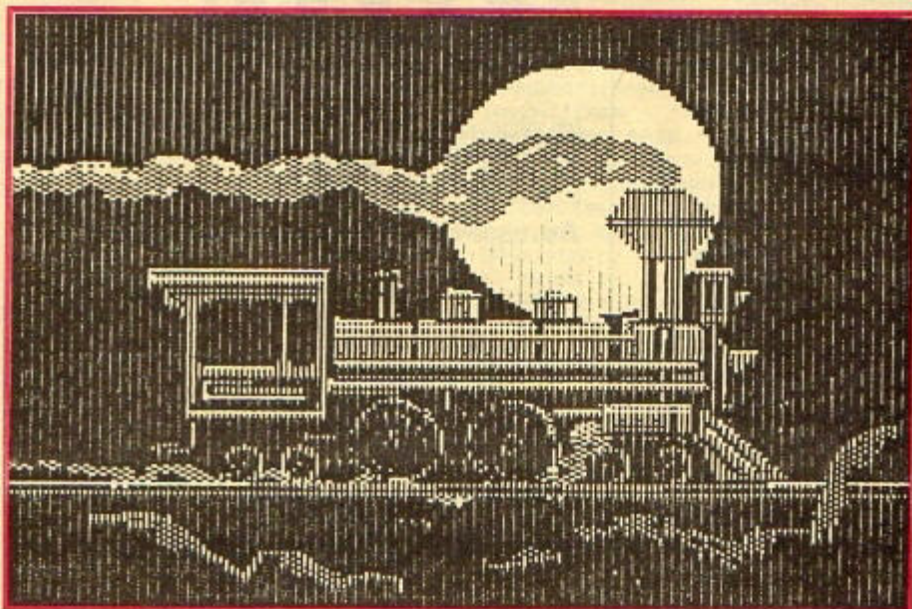
Drückt man F1, so wird eine ab

\$E000 stehende Grafik invertiert. Dies geschieht innerhalb einer halben Sekunde. Nach dem ersten Invertieren erscheint in der linken oberen Bildschirmecke ein Ausrufezeichen, um die erfolgte Invertierung anzuzeigen. Beim zweiten Invertieren verschwindet es wieder.

Directory

Nach Drücken von F3 erscheint die Aufforderung, die Diskette mit den Grafikfiles einzulegen und RETURN zu drücken. Jetzt erscheinen nacheinander die Directory-Einträge, und durch Drücken von N beziehungsweise Y kann man ein geeignetes File auswählen.

Danach kann es die Meldung „No Program File“ geben, die anzeigt, daß das ausgesuchte File nicht geladen werden kann. Im positiven Fall wird nach der Anzahl der zu überspringenden Bytes gefragt. Diese Frage ist notwendig, um eine hohe Kompatibilität zu bestehenden Grafiken zu gewährleisten. Meistens muß hier die vor-eingestellte „2“ eingegeben wer-



Hires-Grafik mit „Paint Magic“

den; man sollte also lediglich RETURN eingeben. Nur bei Grafiken des Programms PAINT MAGIC muß „116“ eingegeben werden, da hier nicht die zwei Bytes übersprungen werden müssen, die die Startadresse der Grafik angeben, sondern ein 116 Bytes langes Ladeprogramm.

Übrigens ist die Länge des Grafikfiles unwichtig; ist es kürzer als 8000 Bytes, so füllt das Ladeprogramm den Rest des Grafikspeichers automatisch mit Nullen auf. Ist es dagegen länger, dann bricht das Ladeprogramm nach 8000 Bytes automatisch ab.

Doch trotz aller Kompatibilität sollte man vorsichtig sein bei Grafiken, die mit „Blazing Paddles“ oder dem „Koalapainter“ erstellt wurden. Diese Programme speichern die Grafiken wegen ihres „Multicolor-Modus“ etwas anders ab, so daß sie zwar geladen werden können, jedoch beim Ausdruck nicht das gewünschte Ergebnis erzielt wird. Relativ gut lassen sich einfarbige Grafiken ausdrucken. Die wenigsten Probleme machen die Bilder von Dia-Shows und von Paint Magic (siehe dazu auch die Bilder 1 und 2). Übrigens funktioniert ENLARGED PRINT auch einwandfrei mit Bildern, die mittels der Routine GRAFIK SPEICHERN aus der RUN, Heft 10/1984, abgespeichert wurden. Somit bietet sich ENLARGED PRINT

als (fast) ideale Ergänzung zu dieser Routine an. Um mit dem Ausdruck beginnen zu können, ist RETURN zu betätigen. Vorher ist auf jeden Fall der Drucker einzuschalten, da sonst das Programm mit DEVICE NOT PRESENT abstürzt. Die Print-Routine, die in Zeile 36 mit SYS10496 aufgerufen wird, druckt eine im RAM ab \$E000 liegende Grafik innerhalb von 13 Minuten (mit MPS 801; auf Epson-Kompatiblen mit Merlin-Interface schneller) im Format DIN A4 quer aus. Da dieser Bereich mit dem Grafikspeicher von SIMON's Basic übereinstimmt, können außer den von Disk eingelesenen Grafiken auch Grafiken ausgedruckt werden, die vorher mit SIMON's Basic erstellt wurden. Dazu muß lediglich unmittelbar nach dem Start von ENLARGED PRINT die RETURN-Taste betätigt werden.

Druck in der Mitte

Das Programm rückt die Grafik auf dem Drucker automatisch um 32 Punkte nach rechts ein, um das Bild annähernd in Blattmitte zu drucken. Um diese Einrückung zu verändern, ist lediglich das 11. Datum in Zeile 98 von 32 auf einen anderen Wert zu ändern. Zulässig sind Werte von 1 bis 79, wobei 1 das Bild an den linken und 79 es

an den rechten Blattrand druckt. Aber Vorsicht: Bei Ändern der DATAs sollte gleichzeitig auch die Prüfsumme in Zeile 18 verändert werden!

Nun noch einen Tip für die Besitzer von SIMON's Basic: Mit SIMON's Basic kann die eingeladene Grafik auch während des Druckvorganges auf dem Bildschirm gezeigt werden. Dazu sind folgende Zeilen einzufügen:

```
19 HIRES0,1
35 CSET2
37 CSET0
```

Aber Achtung: Diese Zeilen müssen mit SIMON's Basic eingegeben werden, da sie sonst nicht in Tokens umgewandelt werden und bei einem Lauf zusammen mit SIMON's Basic ein SYNTAX ERROR in diesen Zeilen angezeigt würde. Außerdem sollte die Zeile 19 weggelassen werden, wenn nur der Ausdruck von mit SIMON's Basic erstellten Grafiken beabsichtigt ist, da eine vorhandene Grafik sonst gelöscht würde.

MPS801-Besitzern, die sich keinen PRINT SHOP leisten können, sei hier eine Art Notlösung empfohlen: Da ENLARGED PRINT die Bilder im Querformat ausdruckt, ist es möglich, meterlange Transparente zu erstellen. Wenn die Buchstaben des Textes mit einem beliebigen Malprogramm auf mehreren Grafiken hintereinander erstellt werden (zirka zwei Buchstaben pro Grafik; Hintergrundfarbe: weiß) und dann nacheinander geladen und ausgedruckt werden (von hinten nach vorne; die letzten Buchstaben zuerst!), dann kann es klappen. Siehe dazu auch Bild 3.

Zeilen 12 bis 18:

Maschinenprogrammabereich vor Überschreiben schützen; Maschinenprogramme einlesen, Prüfsummen

Zeilen 21 bis 33:

Anzeige des Menüs; Verzweigung in die einzelnen Unterprogramme
Zeile 31:

Aufruf der Invertierungsroutine

Zeilen 34 bis 41:

Aufruf der Druckroutine und Abfrage auf nochmaligen Programm-durchlauf

Zeilen 42 bis 81:

Directory-Routine (wenn D=-1, ▶



Sargon II stammt aus einer Dia-Show

dann Directory-Funktion; ansonsten Verzweigung zur Laderoutine)
 Zeilen 82 bis 84:
 Aufruf der Laderoutine
 Zeilen 86 bis 88:
 Daten für die Invertierungsroutine

Zeilen 90 bis 93:
 Daten für die Laderoutine
 Zeilen 95 bis 110:
 Daten für die Druckroutine
 Die Invertierungsroutine befindet sich im Kassettenpuffer von dez.828 bis dez.880 und wird mit

SYS 828 aufgerufen. Die Laderoutine liegt ebenfalls im Kassettenpuffer, und zwar von 881 bis 952. Sie wird mit SYS 881 aufgerufen. Die Druckroutine wurde im Basic-Speicher von 10496 bis 10815 untergebracht. (Guido Coenen)

| | |
|--|---|
| <pre> 1 REM ***** 2 REM ** ENLARGED PRINT ** 3 REM ** FOR THE MPS 801/VC1525 ** 4 REM ** (C) 7.7. 1985 BY: ** 5 REM ** GUIDO COENEN ** 6 REM ** ALLENSTEINERSTRASSE 11 ** 7 REM ** 4048 GREVENBROICH 1 ** 8 REM ** TEL.: (02181)42206 ** 9 REM ***** 10 ; 11 REM **** M - ROUTINEN EINLESEN **** 12 POKE 53280,4:POKE 53281,14:POKE 45,0: 3698 POKE 46,43:CLR:SB=0 13 A=0:FOR X=828 TO 880:READ S:A=A+S: 3376 POKE X,S:NEXT 14 IF A<>5635 THEN PRINT " [BLK,RVS] 4980 WRONG DATA(S) IN INVERT ROUTINE ! ": STOP 15 A=0:FOR X=881 TO 952:READ S:A=A+S: 3392 POKE X,S:NEXT 16 IF A<>11487 THEN PRINT " [BLK,RVS] 4895 WRONG DATA(S) IN LOAD ROUTINE ! ": STOP 17 A=0:FOR X=10496 TO 10815:READ S:A=A+ 3186 S:POKE X,S:NEXT 18 IF A<>39533 THEN PRINT " [BLK,RVS] 5045 WRONG DATA(S) IN PRINT ROUTINE ! ": STOP 20 REM ***** INVERTIEREN, DRUCKEN ***** 21 D=0:PRINT "[CLR,BLK] * TAB(5)*GUIDO 5289 COENEN PROUDLY PRESENTS : " 22 PRINT TAB(7)*[DOWN]ENLARGED GRAPHICS 3185 PRINTING" 23 PRINT TAB(7)*[F1 = INVERT GRAPHIC 2877 PAGE)*" 24 PRINT TAB(8)*[F3 = CHOOSE A PICTURE 3840 FILE)*" 25 PRINT TAB(12)*[F5 = DIRECTORY)*" 2369 26 PRINT TAB(5)*FOR THE MPS-801 MATRIX 3247 PRINTER" 27 PRINT "[DOWN,3SPACES]GRAPHICS START 3193 ADDRESS : 57344)" 28 PRINT TAB(9)*[DOWN]TURN ON THE 2732 PRINTER !" 29 PRINT TAB(5)*[RETURN] TO CONFIRM 3609 PRINTING" 30 GET A\$:IF A\$<>CHR\$(13) THEN IF A\$<>" 5171 [F1]" THEN IF A\$<>[F3]" THEN IF A\$<> [F5]" GOTO 30 31 IF A\$=CHR\$(133) THEN SYS 828:GOTO 30 2130 32 IF A\$=[F3]" GOTO 43 1290 33 IF A\$=[F5]" THEN D=-1:GOTO 43 2041 34 PRINT TAB(18)*[DOWN]OK !" 1350 36 OPEN 4,4:OPEN 6,4,6:PRINT#6,CHR# 4775 (18):CLOSE 6:PRINT#4,CHR# (8):SYS 10496 38 OPEN 4,4:PRINT#4,CHR# (15):CLOSE 4: 4247 PRINT "[CLR,DOWN] ANOTHER PICTURE (Y/N) ?" 39 GET A\$:IF A\$<>"Y" THEN IF A\$<>"N" 2761 GOTO 39 40 IF A\$="Y" GOTO 21 1119 41 END 128 42 REM *** DIRECTORY + FILES LESEN *** 43 PRINT "[CLR,5SPACES]INSERT DISK & 2623 PRESS [RETURN)*" </pre> | <pre> 44 PRINT "[3SPACES] 3617 *****" 45 GET A\$:IF A\$<>CHR\$(13) GOTO 45 1976 46 PRINT TAB(6)*[DOWN]M TO RETURN TO 3449 MAIN MENU)" 47 OPEN 1,8,15,"10":INPUT#1,A\$:IF A\$<> 3544 "00" THEN CLOSE 1:RETURN 48 I\$="":OPEN 2,8,2,"#":PRINT#1,"U1:2 0 3369 18 0":PRINT :PRINT 49 PRINT#1,"B-P1:2:144:PRINT "[7SPACES, 2326 RVS]"); 50 FOR X=144 TO 161:GET #2,B\$:PRINT B\$:: 2247 NEXT 51 FOR X=1 TO 5:GET #2,B\$:I\$=I\$+B\$:PRINT 4703 B\$::NEXT :PRINT :PRINT :TX=18:SY=1 52 PRINT#1,"U1:2:0:TX:SY:PRINT#1,"B-P1:" 4768 2:0:GET #2,T\$,S\$ 53 TS=TS+CHR\$(0):S\$=S\$+CHR\$(0):TX=ASC 4360 (T\$):SY=ASC (S\$) 54 N\$="":FOR PO=2 TO 226 STEP 32:TYP\$="" 3917 :PRINT#1,"B-P1:2:PO 55 GET #2,TYP\$:IF TYP\$="" GOTO 72 1786 56 PRINT#1,"B-P1:2:PO+28:GET #2,LB\$,HB\$ 2823 57 LB\$=LB\$+CHR\$(0):HB\$=HB\$+CHR\$(0):L= 4318 ASC (LB\$)+ASC (HB\$)+256 58 IF L<100 THEN PRINT " ":IF L<10 THEN 2681 PRINT " "; 59 PRINT L"[2SPACES]" CHR# (34):BO=0: 4521 N\$="":PRINT#1,"B-P1:2:PO+3:C\$="" 60 FOR Z=1 TO 16:GET #2,C\$:IF C\$=CHR\$ 5506 (16) AND NOT BO THEN BO=-1:C\$=CHR\$ (34) 61 PRINT C\$::IF C\$<>CHR\$(34) THEN IF 4118 C\$<>CHR\$(160) THEN N\$=N\$+C\$ 62 NEXT 130 63 IF LEN (N\$)=0 THEN PO=226:TX=0:PRINT 3673 CHR# (34):GOTO 72 64 IF NOT BO THEN PRINT CHR# (34): 1321 65 IF BO THEN PRINT CHR# (32): 1325 66 IF D THEN PRINT :GOTO 72 1079 67 PRINT "[3SPACES](Y/N)":A\$="" 1510 68 GET A\$:IF A\$<>"N" THEN IF A\$<>"Y" 3841 THEN IF A\$<>"M" GOTO 68 69 IF A\$="M" THEN PO=226:TX=0 2444 70 IF A\$="N" THEN PRINT "[RVS]NO":GOTO 2356 72 71 IF A\$="Y" THEN N=-1:D=-1:N\$=N\$:TYP= 5579 ASC (TYP\$):PRINT :PO=226:TX=0 72 NEXT 130 73 IF TX<>18 GOTO 75 1048 74 GOTO 52 443 75 PRINT#1,"M-R" CHR# (250) CHR# (2):GET 3481 #1,A\$:A\$=A\$+CHR\$(0) 76 PRINT#1,"M-R" CHR# (252) CHR# (2):GET 5697 #1,C\$:C\$=C\$+CHR\$(0):X=ASC (A\$)+ASC (C\$)+256 77 PRINT TAB(7)*[DOWN,RVS]"X"[LEFT] 2389 BLOCKS FREE " 78 IF D THEN POKE 198,0:WAIT 198,1:POKE 2185 198,0 79 IF NOT N GOTO 84 618 80 IF TYP=130 OR TYP=194 GOTO 82 2056 81 PRINT "[RVS] SORRY, NO PROGRAM-FILE ! 5963 ":POKE 198,0:WAIT 198,1:POKE 198,0: GOTO 84 </pre> |
|--|---|

| | | | |
|--|------|--|------|
| 82 INPUT *(2DOW!,5SPACES)SKIP OVER HOW MANY BYTES (3RIGHT)2(3LEFT)*:BY:IF BY<0 OR BY>255 GOTO 82 | 5674 | 208,246,169,248,141,240,3,169,223,141 | |
| 83 POKE 895,BY:PRINT TAB(5)*LOADING*:: CLOSE 2:OPEN 2,0,2,NN\$+*,P,R*:POKE 144,0:SYS 881 | 5614 | 97 DATA 241,3,169,56,141,242,3,169,255,141,243,3,169,0,133,39,133,72,133,18 | 7367 |
| 84 SYS 65404:CLOSE 2:CLOSE 1:GOTO 21 | 2063 | 98 DATA 162,4,32,201,255,169,91,133,253,160,32,169,129,32,210,255,136,208,250 | 7424 |
| 85 REM ** DATA'S FOR INVERT-ROUTINE ** | | 99 DATA 173,240,3,133,251,173,241,3,133,252,230,252,169,64,24,101,251,133,251 | 7494 |
| 86 DATA 169,224,141,78,3,141,83,3,165,1,72,169,52,120,133,1,173,0,224,73,255 | 7606 | 100 DATA 144,2,230,252,169,0,133,71,165,252,133,70,165,251,133,69,24,101,71 | 6979 |
| 87 DATA 141,0,224,238,77,3,238,82,3,208,240,238,78,3,238,83,3,208,232,104,133 | 7080 | 101 DATA 133,69,144,2,230,70,32,45,42,169,128,133,38,165,72,133,39,169,0,133 | 7696 |
| 88 DATA 1,88,173,0,4,73,1,141,0,4,96 | 3298 | 102 DATA 254,165,2,164,39,57,208,3,240,10,164,254,185,224,3,24,101,38,133,38 | 7724 |
| 89 REM *** DATA'S FOR LOAD-ROUTINE *** | | 103 DATA 230,39,165,39,41,16,240,31,169,0,133,39,169,0,56,229,71,133,2,165,252 | 7758 |
| 90 DATA 162,2,32,198,255,169,0,133,251,169,224,133,252,160,2,240,6,32,228,255 | 7245 | 104 DATA 133,70,165,251,56,229,2,133,69,176,2,199,70,32,45,42,230,18,230,254 | 7705 |
| 91 DATA 136,208,250,32,228,255,145,251,208,208,2,230,252,165,144,73,64,240,12 | 7123 | 105 DATA 169,7,69,254,208,190,165,38,32,210,255,32,210,255,230,71,169,8,69,71 | 7103 |
| 92 DATA 165,252,73,255,208,234,152,73,64,208,229,96,169,0,145,251,200,208,2,230 | 7270 | 106 DATA 240,3,76,106,41,165,252,77,243,3,240,3,76,89,41,165,251,77,242,3,208 | 7913 |
| 93 DATA 252,165,252,73,255,208,241,152,73,64,208,236,96 | 5310 | 107 DATA 246,165,39,133,72,169,13,32,210,255,165,18,240,32,169,0,133,18,173 | 7067 |
| 94 REM ** DATA'S FOR PRINT-ROUTINE ** | | 108 DATA 240,3,56,233,8,141,240,3,176,3,206,241,3,173,242,3,56,233,8,141,242 | 7139 |
| 95 DATA 160,0,169,1,153,208,3,153,209,3,200,208,10,56,192,16,208,242,162,0 | 7162 | 109 DATA 3,176,3,206,243,3,198,253,240,3,76,69,41,32,204,255,169,4,32,195,255 | 7282 |
| 96 DATA 169,1,157,224,3,10,232,56,224,7, | 7093 | 110 DATA 96,165,1,72,160,0,169,52,120,133,1,177,69,133,2,104,133,1,88,96 | 6915 |

Alles für CBM-64/C-128 VC-20

mükra
DATEN-TECHNIK

Schöneberger Str. 5
(Am Berlinicke Platz)
1000 Berlin 42/W
☎ 030-752 91 50/60

Öffnungszeiten:
Mo.-Fr.: 10-18 Uhr
Sa.: 10-13 Uhr

HARDWARE

- Staubschutzhäuben** C-64/20 o. Floppy 1541 je: **139,-**
 - Eprom-Karte "2x8K"** C-64 (die Preiswerte!) nur: **187,-**
 - 4-Fach Betriebssystem-Adapter** C-64 nur: **36,-**
 - Turbo Floppy** C-64 (Stockmodul) nur: **49,-**
 - Floppy-Kühler** C-64/20 (Ohne Einbau!) nur: **59,-**
 - Super + Extra Tool** C-64 (Stockmodul) nur: **99,-**
 - 3-Fach Steckplatzerweiterung** C-64/20 nur: **99,-**
 - Zusatz-Tastatur** C-64/20 nur: **159,-**
 - Akustikkoppler** C-64/20 (Ohne FTZ Nummer) nur: **189,-**
 - PP-64 Eprommer** C-64/20 (Testgerät "64er" 9/85) **279,-**
 - Merlin-Face** C-64 (Centronics-Hardwareinterface) **218,-**
 - Die neuen C-64 Drucker!** (Test "64er" 9/85) **1148,-**
 - Panasonic 1090** mit MERLIN-Interface nur: **1248,-**
 - Panasonic 1091** mit MERLIN-Interface und NMQ nur: **1498,-**
 - Panasonic 1092** mit MERLIN-Interface und NMQ nur: **1498,-**
- 81 K RAM-Speicher für den C-64, 80 Zeichenkarten, Eprombank, Epromlötlaster, Userportadapter, VC-20 Speichererweiterungen + Zubehör, IEEE-488, RS 232 C-Schnittstellen und Interface, Kabel, Joysticks, Super Sketch, Monitore, Light-pen, Resetstator, Bauteile, Stecker, Eproms, Bücher, Geräte und ... und ...

SOFTWARE

- Luxus-Telefon** C-64 (Supportierpreis!) Disk: **197,-**
 - Lotto-Berechnung** C-64 (Spiel 6 aus 49) Disk: **49,-**
 - Krankheits-Diagnose** C-64/20 Cass: **29,-**
 - Psycho-Test** C-64 (nach Löscher) Disk: **36,-**
 - Astrologie** C-64 (Häuser nach Koch) Disk: **85,-**
 - Biorhythmus** C-64/20 (Mit Partnervergleich!) Cass: **29,-**
 - Eprom-Maker V1.0** C-64 (Modulgenerator) Disk: **36,-**
 - Boeing 727** C-64/20 (Joystick) Cass: **29,-**
 - Mükra-Datei** C-64/20 (Universelles Dateiprogr.) Disk: **36,-**
 - Star-Texter** C-64 (Test "64er" 9/85) Disk: **42,-**
 - Faktura + Lager** C-64 Disk: **64,-**
 - Sex-Trainer** C-64 (Keine Spielerei!) Disk: **49,-**
- Automatisches Wählen jeder Nummer. Praktischer Gebührenzähler integrierte Adressdatei. Auch Anschlußhardware lieferbar.
Analyse aller gespeicherten Ziehungen seit 1985.
Stellt fest, welche Krankheit Sie haben könnten.
Über Ja/Nein-Abfragen. Auch alle Kinderkrankheiten.
Wer bin ich? Wer ist mein Nächster? Hier bekommen Sie Antwort!
Genauere Horoskopberechnung mit allen Daten und Auswertungen.
2-seitiger Ausdruck eines Persönlichkeits-Profil.
Erstellt Diagramm mit allen Biokurven. Auch Geburtsmonatphasen!
Ausgabe: Bildschirm o. Commodore-Drucker.
Alle Programme Basic o. MA-Sprache bis 16 K im Eprom schießen.
Echtzeit-Flugsimulator mit Flugprotokoll.
Der neueste Spitzen-Textverarbeiter zum Tiefpreis von Sybex.
Unbedingt Sonderinformation mit Ausdrucken anfordern.
Wissenschaftliche Tests rund um Sex und Partnerschaft.
- ☛ Immer die allerneuesten Spiele am Lager!

Hier sind wir:



GRATIS!

Zu jeder Katalogbestellung die "PEEK + POKE MAGICS". Über 300 Tips + Tricks für C-64 als kompaktes Nachschlagewerk.



JETZT NEU!

Katalog "Winter 85/86" (64 Seiten)
Schnell anfordern für 2.50 DM (bar oder Briefmarken)
Versandpauschale 6.- DM
Händleranfragen erwünscht/Preise inkl. MwSt.

Quick-Information:

☎ 030/752 91 50/60

C 64 mit besonderer Note

Die eingebauten Zeichen des C64 sind gut lesbar und erfüllen ihren Zweck. Oft sind aber ausdrucksvollere Schriftzüge gewünscht. RUN-Autor Martin Bittner hat einen Zeichensatz für den besonderen Geschmack entworfen. Er ersetzt die eckigen Lettern der Commodore-Druckschrift durch die sanft geschwungenen Bögen der Schreibschrift.

Schreibschrift mit Simon's Basic

Das Programm ist in Simon's Basic geschrieben. Es verlegt mit dem MEM-Befehl den Zeichensatz in den RAM-Bereich ab \$E000 und nimmt dort an ihm die nötigen Änderungen vor. Nun stehen die kleinen Schreibbuchstaben zur Verfügung (siehe Bild). Man erhält sie durch Drücken der entsprechenden Buchstabentaste zusammen mit der Shift-Taste. Auch an die deutschen Sonderzeichen wurde gedacht:

;=ä / :=ö / @=ü / +=ß

Die Grafikzeichen, die sonst mit

*Wer's stilecht mag,
braucht seinen C64
nicht gleich mit Brokat
einzuschlagen. Er kann
ihm auch durch
lateinische Schrift
eine besondere
Note verleihen.*



der Shift-Taste zu erreichen sind, sind nicht mehr verfügbar. Das Programm kann mit dem Befehl MERGE "SCHREIBSCHRIFT",8 zu einem anderen Programm dazugeladen und als Unterprogramm eingebaut werden. Die handgemalten Zeichen können auch ohne Simon's Basic verwendet werden. Dazu müssen die Daten der neuen Buchstaben in Data-Zeilen untergebracht werden.

Zeichenmatrix mit DESIGN

Am einfachsten lassen sich die Data-Zeilen mit dem Zeicheneditor ZED erzeugen (Seite 58 in diesem Heft). Dabei kann man die Zeichenmatrix für jeden Buchstaben direkt den Programmzeilen nach den DESIGN-Befehlen entnehmen. Nach dem Simon's Basic-Befehl "@" wird das Punktemuster für eine Zeichenzeile angegeben. Dabei entspricht "B" einem gesetzten, "." einem freien Punkt.

(Martin Bittner)

| | | | |
|----------------------------|------|----------------------------|------|
| 10050 MEM | 344 | 10250 e..... | 1084 |
| 10060 DESIGN 2,\$E000+65*8 | 1590 | 10260 e..... | 1084 |
| 10070 e..... | 1084 | 10270 e..... | 1084 |
| 10080 e..... | 1084 | 10280 e..BBBB.. | 679 |
| 10090 e..... | 1084 | 10290 e.B..... | 909 |
| 10100 e.BBBBB.. | 504 | 10300 eB.....B | 1144 |
| 10110 eB...B.. | 1129 | 10310 e.B...B. | 919 |
| 10120 eB...B.. | 1129 | 10320 e..BBBB.. | 679 |
| 10130 eB...B.B | 1149 | 10330 DESIGN 2,\$E000+68*8 | 1614 |
| 10140 e.BBBB.B. | 509 | 10340 e....B.. | 1089 |
| 10150 DESIGN 2,\$E000+66*8 | 1598 | 10350 e....B.. | 1089 |
| 10160 e..BBBB.. | 679 | 10360 e....B.. | 1089 |
| 10170 e..B...B. | 999 | 10370 e.BBBBB.. | 504 |
| 10180 e..B..BB. | 1004 | 10380 eB...B.. | 1129 |
| 10190 e..BBB... | 674 | 10390 eB...B.. | 1129 |
| 10200 e.BB...B. | 824 | 10400 eB...B.B | 1149 |
| 10210 eB.B...BB | 1059 | 10410 e.BBBB.B. | 509 |
| 10220 e..B...B. | 999 | 10420 DESIGN 2,\$E000+69*8 | 1622 |
| 10230 e...BBB.. | 774 | 10430 e..... | 1084 |
| 10240 DESIGN 2,\$E000+67*8 | 1606 | 10440 e..... | 1084 |

RUN

11. November 1985
5,50 DM

UNABHÄNGIGES
COMMODORE
COMPUTERMAGAZIN

C64
TESTET
INTELLIGENZ

C 128 IM VISIER
SOFTWARE, LISTINGS,
TIPS & TRICKS

MITMACHEN -
GEWINNEN -
20 000 MARK -
WETTBEWERB

COCKPIT
SIMON'S BASIC
DISKETTENSTATION
1541-ALTERNATIVE
MIT TURBO-ACCESS

KURS: BLOCKGRAFIK • VC-20 SICHTET DATEN •
KARATESPIEL: EXPLODING FIST • EPROM-
BANK IM RECHNER • NEUES VON DER
FUNKAUSSTELLUNG UND, UND, UND ...

RUN stürmt die Commodore- Welten. Und besonders die des C64. RUN bringt alles, was Commodore-Anwender wissen wollen. Ob „Einsteiger“ oder „Fortgeschrittener“; ob „Computer als Hobby“ oder „Computer im Beruf“ — RUN hat die richtigen Themen.

Wenn Sie das nächste Mal beim Zeitschriften-Händler sind, oder bei Ihrem Computer-Händler: Bringen Sie (sich) das neue RUN mit — Sie bekommen es dort jeden Monat. Aktuell.

Falls vergriffen — schreiben Sie an den Verlag. Wir senden Ihnen ein kostenloses Probe-Exemplar.



PUBLIKATIONEN

Eine Erste Adresse für die Computerwelt.

Friedrichstraße 31, D-8000 München 40

| | | | | | |
|-------|----------------------|------|-------|----------------------|------|
| 10450 | e..... | 1084 | 11030 | e..B..B.B | 1014 |
| 10460 | e..BBB... | 674 | 11040 | e..B..BB. | 1004 |
| 10470 | e.B...B.. | 914 | 11050 | DESIGN 2,\$E000+76*8 | 1602 |
| 10480 | eBBBBB..B | 559 | 11060 | e..BBBBB. | 689 |
| 10490 | e.B....B. | 919 | 11070 | e..B...B. | 999 |
| 10500 | e..BBBB.. | 679 | 11080 | e..B..BB. | 1004 |
| 10510 | DESIGN 2,\$E000+70*8 | 1554 | 11090 | e..B.BB.. | 869 |
| 10520 | e....BBB. | 974 | 11100 | e..BBB... | 674 |
| 10530 | e...B..B. | 904 | 11110 | eBBB..... | 854 |
| 10540 | e...B.B.. | 899 | 11120 | e...B...B | 914 |
| 10550 | e...BB... | 769 | 11130 | e...BBBB. | 784 |
| 10560 | e..BB.... | 799 | 11140 | DESIGN 2,\$E000+77*8 | 1610 |
| 10570 | eBB.BBBBB | 669 | 11150 | e..... | 1084 |
| 10580 | e..BB.... | 799 | 11160 | e..... | 1084 |
| 10590 | e.B.B.... | 719 | 11170 | e..... | 1084 |
| 10600 | DESIGN 2,\$E000+71*8 | 1562 | 11180 | e.BB.BB.. | 694 |
| 10610 | e..... | 1084 | 11190 | eB..B..B. | 944 |
| 10620 | e.BBBB... | 499 | 11200 | eB..B..B. | 944 |
| 10630 | e.B...B.. | 914 | 11210 | eB..B..B. | 944 |
| 10640 | e.BBBBB.. | 504 | 11220 | eB..B..BB | 964 |
| 10650 | eB....B.. | 1129 | 11230 | DESIGN 2,\$E000+78*8 | 1618 |
| 10660 | e..BBBBBB | 709 | 11240 | e..... | 1084 |
| 10670 | e.B...B.. | 914 | 11250 | e..... | 1084 |
| 10680 | e.BBBBB.. | 504 | 11260 | e..... | 1084 |
| 10690 | DESIGN 2,\$E000+72*8 | 1570 | 11270 | e.BB.BB.. | 694 |
| 10700 | e..BBBB.. | 679 | 11280 | eB.BB.B.. | 844 |
| 10710 | e..B..B.. | 994 | 11290 | eB.B..B.. | 1034 |
| 10720 | e..B.B... | 864 | 11300 | e..B..B.B | 1014 |
| 10730 | e..BB.... | 799 | 11310 | e..B..BB. | 1004 |
| 10740 | e.BBBBB.. | 504 | 11320 | DESIGN 2,\$E000+79*8 | 1626 |
| 10750 | eB.B..B.. | 1034 | 11330 | e..... | 1084 |
| 10760 | e..B..B.B | 1014 | 11340 | e..... | 1084 |
| 10770 | e..B..BB. | 1004 | 11350 | e..... | 1084 |
| 10780 | DESIGN 2,\$E000+73*8 | 1578 | 11360 | e..BBBB.. | 679 |
| 10790 | e..... | 1084 | 11370 | e.BB..BB. | 829 |
| 10800 | e..... | 1084 | 11380 | eBB....BB | 979 |
| 10810 | e...B.... | 894 | 11390 | e.BB..BB. | 829 |
| 10820 | e..... | 1084 | 11400 | e..BBBB.. | 679 |
| 10830 | e..BB.... | 799 | 11410 | DESIGN 2,\$E000+80*8 | 1558 |
| 10840 | eBB.B.... | 759 | 11420 | e..... | 1084 |
| 10850 | e...B...B | 914 | 11430 | e..B..... | 989 |
| 10860 | e....BBB. | 974 | 11440 | e.BBBB... | 499 |
| 10870 | DESIGN 2,\$E000+74*8 | 1586 | 11450 | e.BB..B.. | 819 |
| 10880 | e.....B.. | 1089 | 11460 | eB.B..B.. | 1034 |
| 10890 | e..... | 1084 | 11470 | eB.B..B.B | 1054 |
| 10900 | e....BB.. | 964 | 11480 | e..B..BB. | 1004 |
| 10910 | e..BB.B.. | 804 | 11490 | e..B.... | 989 |
| 10920 | e.B...B.. | 914 | 11500 | DESIGN 2,\$E000+81*8 | 1566 |
| 10930 | eB.BBBBBB | 749 | 11510 | e..... | 1084 |
| 10940 | e.B...B.. | 914 | 11520 | e..BBBB.. | 679 |
| 10950 | e.BBBBB.. | 504 | 11530 | e.B...B.. | 914 |
| 10960 | DESIGN 2,\$E000+75*8 | 1594 | 11540 | e.B...B.. | 914 |
| 10970 | e..BBBB.. | 679 | 11550 | e.BBBBB.. | 504 |
| 10980 | e..B..B.. | 994 | 11560 | eB....BBB | 1159 |
| 10990 | e..B.B... | 864 | 11570 | e...BBB.. | 774 |
| 11000 | e..BBBB.. | 679 | 11580 | e.....B.. | 1089 |
| 11010 | e.BB..B.. | 819 | 11590 | DESIGN 2,\$E000+82*8 | 1574 |
| 11020 | eB.BBBB.. | 719 | 11600 | e..... | 1084 |

| | | | | | |
|-------|----------------------|------|-------|-----------------------|------|
| 11610 | e..... | 1084 | 12190 | e...BB... | 769 |
| 11620 | e..... | 1084 | 12200 | e..B..B.. | 994 |
| 11630 | e...BB... | 769 | 12210 | @BB....BB | 979 |
| 11640 | e.BBB.B.. | 629 | 12220 | DESIGN 2,\$E000+89*8 | 1630 |
| 11650 | @BB.B.BBB | 794 | 12230 | e..... | 1084 |
| 11660 | e...B.... | 894 | 12240 | e.B...B.. | 914 |
| 11670 | e...B.... | 894 | 12250 | e.B...B.. | 914 |
| 11680 | DESIGN 2,\$E000+83*8 | 1582 | 12260 | @B.BBBB.. | 719 |
| 11690 | e..... | 1084 | 12270 | @B....B.. | 1129 |
| 11700 | e..... | 1084 | 12280 | e..BBBBBB | 709 |
| 11710 | e..... | 1084 | 12290 | e.B...B.. | 914 |
| 11720 | e..BBBB.. | 679 | 12300 | e.BBBB.. | 504 |
| 11730 | e.B...BB. | 924 | 12310 | DESIGN 2,\$E000+90*8 | 1562 |
| 11740 | @B....B. | 1134 | 12320 | e..... | 1084 |
| 11750 | e....BBB | 1119 | 12330 | e..... | 1084 |
| 11760 | e...BBB.. | 774 | 12340 | e..... | 1084 |
| 11770 | DESIGN 2,\$E000+84*8 | 1590 | 12350 | e.BBBBBB. | 514 |
| 11780 | e....B... | 959 | 12360 | @B....BB. | 1139 |
| 11790 | e....B... | 959 | 12370 | @B..BB... | 809 |
| 11800 | e....B... | 959 | 12380 | e.BB....B | 834 |
| 11810 | e..BBBBB. | 689 | 12390 | e..BBBBB. | 514 |
| 11820 | e...BB... | 769 | 12400 | DESIGN 2,\$E000+91*8 | 1570 |
| 11830 | @BBB.B... | 729 | 12410 | e..BBBBB. | 689 |
| 11840 | e....B..B | 979 | 12420 | e..B....B | 1009 |
| 11850 | e....BB. | 1099 | 12430 | e..B....B | 1009 |
| 11860 | DESIGN 2,\$E000+85*8 | 1598 | 12440 | e..BBBBB. | 689 |
| 11870 | e..... | 1084 | 12450 | e.BB....B | 834 |
| 11880 | e..... | 1084 | 12460 | @B.B....B | 1049 |
| 11890 | e..... | 1084 | 12470 | e..BBBBB. | 689 |
| 11900 | e..B..B.. | 994 | 12480 | e..B..... | 989 |
| 11910 | e.BB..B.. | 819 | 12490 | DESIGN 2,\$E000+27*8 | 1590 |
| 11920 | @B.B..B.. | 1034 | 12500 | e..... | 1084 |
| 11930 | e..B..B.B | 1014 | 12510 | e..B..B.. | 994 |
| 11940 | e...BB.B. | 779 | 12520 | e..... | 1084 |
| 11950 | DESIGN 2,\$E000+86*8 | 1606 | 12530 | e..BBBB.. | 679 |
| 11960 | e..... | 1084 | 12540 | e.BB..BB. | 829 |
| 11970 | e..... | 1084 | 12550 | @BB....BB | 979 |
| 11980 | e..... | 1084 | 12560 | e.BB..BB. | 829 |
| 11990 | e..B..B.. | 994 | 12570 | e..BBBB.. | 679 |
| 12000 | e.BB..BB. | 829 | 12580 | DESIGN 2,\$E000+29*8 | 1606 |
| 12010 | @B.B..B.B | 1054 | 12590 | e..... | 1084 |
| 12020 | e..B..B.. | 994 | 12600 | e..B..B.. | 994 |
| 12030 | e...BB... | 769 | 12610 | e..... | 1084 |
| 12040 | DESIGN 2,\$E000+87*8 | 1614 | 12620 | e..BBBB.. | 504 |
| 12050 | e..... | 1084 | 12630 | @B....B.. | 1129 |
| 12060 | e..... | 1084 | 12640 | @B....B.. | 1129 |
| 12070 | e..... | 1084 | 12650 | @B....B.B | 1149 |
| 12080 | @B..BB..B | 829 | 12660 | e..BBBB.B. | 509 |
| 12090 | @B..BB..B | 829 | 12670 | DESIGN 2,\$E000+122*8 | 1535 |
| 12100 | @B..BB..B | 829 | 12680 | e..... | 1084 |
| 12110 | @B..BB..B | 829 | 12690 | e..B..B.. | 994 |
| 12120 | e.BB..BB. | 829 | 12700 | e..... | 1084 |
| 12130 | DESIGN 2,\$E000+88*8 | 1622 | 12710 | e..B..B.. | 994 |
| 12140 | e..... | 1084 | 12720 | e.BB..B.. | 819 |
| 12150 | e..... | 1084 | 12730 | @B.B..B.. | 1034 |
| 12160 | e..... | 1084 | 12740 | e..B..B.B | 1014 |
| 12170 | @BB....BB | 979 | 12750 | e...BB.B. | 779 |
| 12180 | e..B..B.. | 994 | | | |

Plotten ohne Plotter

Ein Drucker wird hauptsächlich zum Ausdrucken von Listings oder zum Schreiben von Texten verwendet. Das ist aber nicht alles, was man mit ihm machen kann. Mit der geeigneten Software kann man ihn in einen Plotter verwandeln. Das Druckprinzip kann jedoch nicht geändert werden. Die nicht unterbrochene Linie des Plotters kann mit einem Nadeldrucker nicht erzeugt werden. Wen jedoch mehr die Aussage einer Darstellung interessiert als deren „Reinzeichnung“, der ist mit dem „Plotter-Emulator“ bestens bedient (siehe Probeausdrucke).

Das Programm wird mit LOAD „PLOTTER-EMULATOR“,8,1 geladen und mit SYS22528 gestartet. Nach der Einschaltmeldung ist NEW einzugeben, um die Betriebssystemzeiger richtig zu setzen. Es stehen noch 18 429 Bytes für Basic-Programme zur Verfügung.

Das Programm wird wie ein Peripheriegerät am IEC-Bus angesprochen. Die Geräteadresse für das virtuelle Gerät ist 12. Der Gerätekanal wird mit OPEN 12,12 geöffnet. Die Plotterkommandos werden dann mit PRINT # 12 dem Programm übergeben. Dabei stellt das erste Zeichen den Befehl dar. Der Befehlssatz des Programms entspricht im wesentlichen den Befehlen der Watanabe-Plotter. Lediglich die Auflösung ist größer. Der Befehlssatz ist um einige Kommandos erweitert, die in dieser Form auf einem Plotter nur sehr schwer realisierbar wären.

Die neuen Befehle sind mit (*) markiert. Der Befehl „X“ (Koordinatenachsen zeichnen) ist im Programm nicht implementiert und muß selbst nachgebildet werden. Wird der Befehl trotzdem benutzt, so wird er überlesen.

Die Grafik wird im Speicher erstellt und dann per Befehl auf dem Drucker ausgedruckt. Dabei hat die Grafikseite eine Auflösung von 567 mal 480 Punkten.

Plotter und Drucker sind zwei grundverschiedene Maschinen. Die eine zeichnet, die andere schreibt. Wer es ändern will, kann das nur softwareseitig tun. Das Programm „Plotter-Emulator“ bringt den Druckern MPS 801 und GP 100 VC das Plotten bei.

Beschreibung

der Befehle

Folgende Befehle stehen zur Verfügung:

* initialisiert die Grafik. Die Grafik wird im Speicher gelöscht. Alle Parameter werden auf ihren Anfangswert gesetzt. Die Eingabe dieses Befehls ist bei Programmstart unbedingt erforderlich.

C
(*) löscht die Grafikseite im Speicher. Der Stift bleibt jedoch auf der aktuellen Position.

Aufruf: PRINT # 12, "C"

H
setzt den Stift auf die Position 0/0, ohne eine Linie zu ziehen.

Aufruf: PRINT # 12, "H"

M
setzt den Stift auf die angegebene Position, ohne eine Linie zu ziehen.

Aufruf: PRINT # 12, "M"; x1; ", "; y1

R
bewegt den Stift relativ zur aktuellen Position, ohne eine Linie zu zeichnen.

Aufruf: PRINT # 12, "R"; dx; ", "; dy

(*) bewegt den Stift an die Position x,y und setzt dort einen Punkt.

Aufruf: PRINT # 12, ", "; x; ", "; y

D
zieht eine Linie von der aktuellen Stiftposition zur angegebenen Position.

Aufruf: PRINT # 12, "D"; x1; ", "; y1

I
zieht eine Linie relativ zu der aktuellen Position.

Aufruf: PRINT # 12, "I"; dx; ", "; dy

L
stellt den Typ der gezogenen Linien ein. Der Typ t muß sich zwischen 0 und 3 bewegen:

0: die Linien werden durchgezogen (Voreinstellung)

1: die Linien werden gestrichelt mit der durch den Befehl B vorgegebenen Länge gezeichnet

2: alle Linien werden punktiert gezeichnet, wobei der Abstand fest vorgegeben ist

3: die Linien werden doppelt punktiert gezeichnet

Aufruf: PRINT # 12, "L"; t

Die letzten beiden Linientypen sind im Original-Befehlssatz nicht implementiert.

B
stellt die Länge der Teilabschnitte für die gestrichelten Linien ein.

Das Argument b muß größer oder gleich 1 sein. (Voreinstellung: 1)

Aufruf: PRINT # 12, "B"; b

/

(*) schaltet den Reset-Modus an beziehungsweise aus. Ist der Modus abgeschaltet, so wird der Zähler, der bestimmt, wann eine unterbrochene Linie gezeichnet wird, zu Beginn jedes Befehls zurückgesetzt. Der Parameter p bestimmt, ob der Modus an- (1) beziehungsweise ausgeschaltet (0) wird. (Voreinstellung: 0)

Aufruf: PRINT # 12, "/" ; p

F
(*) zeichnet ein Rechteck mit den Eckkoordinaten x1, y1 und x2, y2

Aufruf: PRINT # 12, "F"; x1; ", "; y1; ", "; x2; ", "; y2

P
schreibt einen Text in die Grafik. Dazu stehen 64 ASCII-Zeichen zur

PLOTTER-DEMO

BY MATTHIAS KOEFFERLEIN
(C) 1985

ES GIBT UNTERSCHIEDLICHE

SCHRIFTEN

0 1 2 3 4 5 6 7 8

...UND UNTERSCHIEDLICHE

BEFEHL...

ROTATE- BEFEHL...
SCHRIFTRICH-
TUNGEN DURCH DEN

UNTERSCHIEDLICHE LINIENTYPEN

GESTRICHELT (1)

(2)

(3)

DURCHGEZOGEN

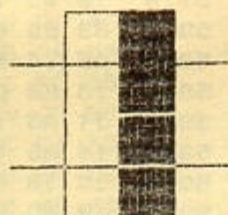
PUNKTIERT (1)

(2)

LINIE TYP 0 (SET)

LINIE TYP 1 (RES)

LINIE TYP 2 (EOR)



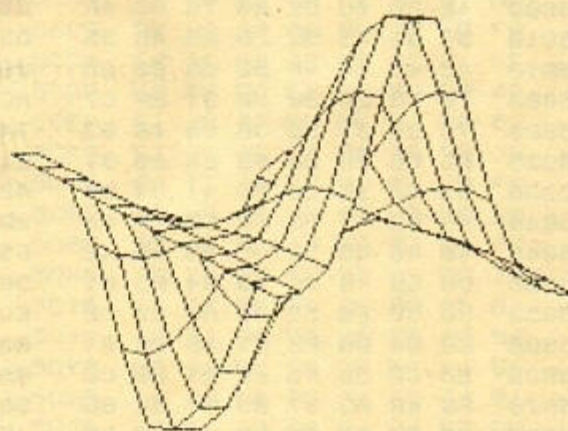
ZEICHENVORRAT

@ABCDEFGHIJKLMNPOQRSTUVWXYZ[-]!+
!"#\$%&'()*+,-./0123456789:;<=>?

MARKEN

* ◇ □ ▲ ▣ ○

✦ ◊ ◻ △ ∑ ○



HIDDEN-LINE-DEMO

Verfügung. Die Größe der Zeichen und die Richtung wird mit dem Befehl S beziehungsweise Q eingestellt.

Die ASCII-Zeichen 96 bis 128 sind nicht implementiert. Die Kleinbuchstaben werden deshalb groß wiedergegeben.

Aufruf: PRINT # 12, "P"; TEXT\$

S stellt die Größe der Buchstaben ein. Für 0 sind die Buchstaben 6

mal 9 Punkte groß. Für andere Größen sind die Zeichen jeweils um den Faktor s+1 (angegebene Größe) größer.

Das Argument s muß größer als 0 sein. (Voreinstellung: 0)

Aufruf: PRINT # 12, "S";s

Q stellt die Richtung der Schrift ein. Das Argument q muß zwischen 0 und 3 liegen:

0: die Schrift wird von links nach

rechts geschrieben (Voreinstellung)

1: die Schrift wird von unten nach oben geschrieben

2: die Schrift wird von rechts nach links auf dem Kopf geschrieben

3: die Schrift wird von oben nach unten geschrieben

Aufruf: PRINT # 12, "Q";q

N zeichnet an der aktuellen Position eine der sechs Marken (vordefi-

nierte Grafiksymbole). Nach Ausführung des Befehls steht der Stift wieder an derselben Position wie vor dem Befehl.

Das Argument n muß sich zwischen 1 und 6 bewegen. (Marke #2 ist nicht implementiert!)

Aufruf: PRINT # 12, "N";n

O

(*) gibt die Grafikseite auf dem Drucker aus. Das Programm wurde für die Drucker MPS 801 oder GP 100 VC geschrieben und ist auf die Geräteadresse 4 eingestellt.

A

(*) stellt den Typ der gesetzten Punkte ein. Der Typ t muß zwischen 0 und 2 liegen:

0: alle Punkte werden gesetzt (Voreinstellung)

1: alle Punkte werden gelöscht

2: alle Punkte werden invertiert
Aufruf: PRINT # 12, "A";t

E

(*) schaltet den Echo-Modus ein oder aus. In diesem Modus wird bei jedem Befehl eine Meldung zur Protokollierung ausgegeben. Das Argument p entscheidet, ob der Modus aus- oder angeschaltet wird:

0: Modus ausschalten (Voreinstellung)

1: Modus anschalten

Aufruf: PRINT # 12, "E";p

#

(*) schaltet den Minimum-Maximum-Modus an beziehungsweise aus. Ist der Modus angeschaltet, so wird ein Punkt nur dann gezeichnet, wenn er höher als der höchste bisher gezeichnete Punkt oder niedriger als der niedrigste

bisher gezeichnete Punkt liegt. Diese Funktion wird benötigt, wenn hinter einem Objekt liegende Linien nicht gezeichnet werden sollen. Der Parameter p bestimmt, ob der Modus an- (1) oder ausgeschaltet (0, Voreinstellung) werden soll. Bei jedem Aufruf wird die Minimum-Maximum-Tabelle gelöscht.

Aufruf: PRINT # 12, "#";p

Die Argumente der Befehle werden als Zahlen oder als Variablen übergeben. Die Strichpunkte zur Abtrennung der Argumente können weggelassen werden. Bei Zahlen sind auch die Anführungszeichen überflüssig. Beispiele:

PRINT # 12, "R";A;";";B

PRINT # 12, "R"A";"B

PRINT # 12, "R5,6"

(M. Köfferlein/W. Wintersberger)

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|--|
| 5800 | 20 | A0 | 5D | A9 | 0A | A0 | 58 | 4C | AC | 5908 | 06 | 14 | 26 | 15 | A5 | 15 | 85 | B0 | 67 | |
| 5808 | 1E | AB | 0D | 0E | 20 | 70 | 4C | 4F | 20 | 5910 | A5 | 14 | 0A | 26 | B0 | 0A | 26 | B0 | 90 | |
| 5810 | 54 | 54 | 45 | 52 | 2D | 45 | 4D | 55 | D3 | 5918 | 18 | 65 | 14 | 85 | 14 | 08 | BD | 80 | 8B | |
| 5818 | 4C | 41 | 54 | 4F | 52 | 0D | 20 | 6D | 08 | 5920 | 03 | 29 | 0F | 18 | 65 | 14 | 85 | 14 | 93 | |
| 5820 | 70 | 73 | 20 | 38 | 30 | 31 | 2F | 67 | AC | 5928 | A5 | 15 | 69 | 00 | 28 | 65 | B0 | 85 | E6 | |
| 5828 | 70 | 20 | 31 | 30 | 30 | 20 | 76 | 63 | 74 | 5930 | 15 | E8 | D0 | C7 | 18 | 24 | B4 | 10 | CF | |
| 5830 | 0D | 0D | 20 | 28 | 63 | 29 | 20 | 31 | 21 | 5938 | 11 | A5 | 14 | 49 | FF | 69 | 01 | 85 | 4A | |
| 5838 | 39 | 38 | 35 | 20 | 6D | 41 | 54 | 54 | 42 | 5940 | 14 | A5 | 15 | 49 | FF | 69 | 00 | 85 | 4D | |
| 5840 | 48 | 49 | 41 | 53 | 20 | 68 | 4F | 45 | AD | 5948 | 15 | 18 | 24 | 38 | A4 | 14 | A5 | 15 | 08 | |
| 5848 | 46 | 46 | 45 | 52 | 4C | 45 | 49 | 4E | F5 | 5950 | 60 | BD | 80 | 03 | F0 | F5 | C9 | 2C | B8 | |
| 5850 | 0D | 00 | 48 | 78 | A9 | 34 | 85 | 01 | D6 | 5958 | F0 | DA | E8 | D0 | F4 | A5 | FC | 30 | 48 | |
| 5858 | 68 | 60 | 20 | 52 | 58 | A0 | 00 | A2 | ED | 5960 | EA | F0 | 0A | C9 | 01 | D0 | E4 | A5 | AC | |
| 5860 | 68 | 84 | B0 | 86 | B1 | A9 | 80 | 91 | 05 | 5968 | FB | C9 | E0 | B0 | DE | A5 | FE | 30 | 68 | |
| 5868 | B0 | C8 | D0 | FB | E6 | B1 | E8 | D0 | 0A | 5970 | DA | A4 | FD | C9 | 02 | 90 | 06 | D0 | 13 | |
| 5870 | F6 | 48 | A9 | 37 | 85 | 01 | 58 | 68 | 93 | 5978 | D2 | C0 | 37 | B0 | CE | 84 | 0D | 85 | 6C | |
| 5878 | 60 | A9 | 00 | 85 | FB | 85 | FC | 85 | 1F | 5980 | 0E | 2C | 59 | 03 | 10 | 53 | A5 | FB | 9F | |
| 5880 | FD | 85 | FE | 60 | 48 | A5 | 9A | C9 | 89 | 5988 | 85 | B0 | A5 | FC | 06 | B0 | 2A | 06 | F9 | |
| 5888 | 0C | D0 | 56 | 68 | 85 | 0D | 48 | 98 | 53 | 5990 | B0 | 2A | 18 | 69 | 50 | 85 | B1 | A5 | 0C | |
| 5890 | 48 | 8A | 48 | A4 | 02 | F0 | 07 | AD | D0 | 5998 | FD | 38 | A0 | 00 | F1 | B0 | A5 | FE | 4B | |
| 5898 | 80 | 03 | C9 | 50 | F0 | 08 | A5 | 0D | 74 | 59A0 | C8 | F1 | B0 | B0 | 0D | A5 | FD | 88 | B4 | |
| 58A0 | 29 | 7F | C9 | 20 | F0 | 16 | A5 | 0D | C4 | 59A8 | 91 | B0 | A5 | FE | C8 | 91 | B0 | A9 | C6 | |
| 58A8 | 29 | 7F | C9 | 0E | 90 | 15 | C9 | 20 | 88 | 59B0 | 80 | 2C | A9 | 00 | 85 | B5 | C8 | 38 | 5C | |
| 58B0 | 90 | 0A | 99 | 80 | 03 | C8 | C0 | 50 | 4A | 59B8 | B1 | B0 | E5 | FD | C8 | B1 | B0 | E5 | CB | |
| 58B8 | F0 | 09 | 84 | 02 | 68 | AA | 68 | A8 | E6 | 59C0 | FE | B0 | 0D | 88 | A5 | FD | 91 | B0 | 5C | |
| 58C0 | 68 | 18 | 60 | A9 | 0D | 99 | 80 | 03 | F9 | 59C8 | C8 | A5 | FE | 91 | B0 | A9 | 80 | 2C | EA | |
| 58C8 | C8 | A9 | 00 | 85 | 02 | 99 | 80 | 03 | 81 | 59D0 | A9 | 00 | 05 | B5 | D0 | 03 | 4C | 4B | C6 | |
| 58D0 | A9 | 03 | 85 | 9A | 20 | CB | 5D | 20 | 23 | 59D8 | 59 | A0 | 01 | 84 | B5 | A9 | C0 | 85 | 7D | |
| 58D8 | 06 | 5B | A9 | 0C | 85 | 9A | 4C | BC | C3 | 59E0 | B4 | 88 | 84 | B0 | A0 | 07 | A5 | 0D | 30 | |
| 58E0 | 58 | 4C | CB | F1 | A9 | 00 | 85 | 14 | FF | 59E8 | 38 | E5 | B4 | 48 | A5 | 0E | E5 | B5 | 85 | |
| 58E8 | 85 | 15 | BD | 80 | 03 | F0 | 5C | C9 | 53 | 59F0 | 90 | 06 | 85 | 0E | 68 | 85 | 0D | 24 | CD | |
| 58F0 | 2D | 00 | 04 | E8 | A9 | 80 | 2C | A9 | 27 | 59F8 | 68 | 26 | B0 | 46 | B5 | 66 | B4 | 88 | 26 | |
| 58F8 | 00 | 85 | B4 | BD | 80 | 03 | F0 | 4B | 94 | 5A00 | D0 | E4 | A5 | B0 | 4A | 66 | B4 | 4A | 2F | |
| 5900 | C9 | 30 | 90 | 30 | C9 | 3A | B0 | 2C | 0A | 5A08 | 66 | B4 | 4A | 66 | B4 | 85 | B5 | A9 | D1 | |
| | | | | | | | | | | 5A10 | 00 | 85 | B0 | 85 | B1 | A0 | 04 | A5 | 17 | |
| | | | | | | | | | | 5A18 | B4 | 18 | 65 | B0 | 85 | B0 | A5 | B5 | 16 | |

5A20 65 B1 85 B1 06 B4 26 B5 4C
 5A28 88 D0 EC A5 B0 18 65 FB 01
 5A30 85 B0 A5 B1 65 FC 69 68 34
 5A38 85 B1 20 52 58 A4 0D B9 AB
 5A40 6C 5A AC 48 03 F0 1C C0 E6
 5A48 01 F0 0D C0 02 D0 14 A0 07
 5A50 00 51 B0 91 B0 4C 71 58 4B
 5A58 A0 00 49 FF 31 B0 91 B0 4D
 5A60 4C 71 58 A0 00 11 B0 91 A1
 5A68 B0 4C 71 58 01 02 04 08 78
 5A70 10 20 40 A9 00 8D 49 03 41
 5A78 60 AD 51 03 F0 43 C9 01 75
 5A80 F0 2A C9 02 F0 17 C9 04 28
 5A88 B0 37 AD 49 03 C9 05 90 8C
 5A90 2D F0 36 C9 07 F0 32 C9 2D
 5A98 03 B0 29 90 21 AD 49 03 AE
 5AA0 C9 05 90 1A F0 23 C9 07 C7
 5AA8 B0 1A 90 12 AD 49 03 CD AD
 5AB0 4A 03 90 0A ED 4A 03 CD A5
 5AB8 4A 03 B0 08 90 0B EE 49 17
 5AC0 03 4C 5D 59 A9 FF 8D 49 AA
 5AC8 03 EE 49 03 60 2C 55 03 E2
 5AD0 10 30 85 B0 8A 48 A5 B0 EC
 5AD8 20 91 B3 20 DD BD AE 00 F0
 5AE0 01 E0 20 D0 02 69 00 20 EC
 5AE8 1E AB 68 AA 60 2C 55 03 B8
 5AF0 10 10 8D 56 03 8A 48 AD 4C
 5AF8 56 03 20 1E AB 68 AA 4C F5
 5B00 2C 64 60 A9 2C 2C A9 0D 1B
 5B08 2C A9 20 2C A9 3B 2C 55 BC
 5B10 03 10 EF 4C 0C E1 A5 A7 BF
 5B18 85 B0 A5 A8 49 80 0A A5 01
 5B20 A8 85 B1 05 B0 08 A5 A9 71
 5B28 85 B4 A5 AA 49 80 0A A5 FB
 5B30 AA 85 B5 05 B4 4C 61 5B 42
 5B38 A5 A7 38 E5 FB 85 B0 A5 D2
 5B40 A8 E5 FC 85 B1 49 80 0A 17
 5B48 A5 B1 05 B0 08 A5 A9 38 1E
 5B50 E5 FD 85 B4 A5 AA E5 FE 9B
 5B58 85 B5 49 80 0A A5 B5 05 AF
 5B60 B4 F0 07 B0 03 A9 FF 2C 49
 5B68 A9 01 8D 46 03 8D 42 03 A4
 5B70 6A 49 80 8D 47 03 8D 43 A2
 5B78 03 A9 00 28 F0 07 B0 03 C8
 5B80 A9 FF 2C A9 01 8D 44 03 E4
 5B88 8D 40 03 6A 49 80 8D 45 F7
 5B90 03 8D 41 03 A5 B0 8D 4B EB
 5B98 03 A4 B1 8C 4C 03 10 10 CE
 5BA0 49 FF 18 69 01 8D 4B 03 B7
 5BA8 98 49 FF 69 00 8D 4C 03 A1
 5BB0 A5 B4 8D 4D 03 A4 B5 8C B9
 5BB8 4E 03 10 10 49 FF 18 69 45
 5BC0 01 8D 4D 03 98 49 FF 69 A2
 5BC8 00 8D 4E 03 AD 4B 03 38 EB
 5BD0 ED 4D 03 AD 4C 03 ED 4E DD
 5BD8 03 B0 27 A9 00 8D 40 03 83
 5BE0 8D 41 03 AD 4D 03 48 AD 87
 5BE8 4E 03 48 AD 4B 03 8D 4D 6E

5BF0 03 AD 4C 03 8D 4E 03 68 7A
 5BF8 8D 4C 03 68 8D 4B 03 4C D5
 5C00 0A 5C A9 00 8D 42 03 8D 18
 5C08 43 03 AD 4C 03 8D 53 03 67
 5C10 8D 50 03 AD 4B 03 8D 52 16
 5C18 03 4E 50 03 6A 8D 4F 03 2B
 5C20 20 79 5A AD 52 03 D0 03 70
 5C28 CE 53 03 CE 52 03 2C 53 D8
 5C30 03 30 6B AD 4F 03 18 6D 89
 5C38 4D 03 8D 4F 03 AD 50 03 2B
 5C40 6D 4E 03 8D 50 03 AD 4F 40
 5C48 03 38 ED 4B 03 A8 AD 50 25
 5C50 03 ED 4C 03 90 27 8D 50 05
 5C58 03 8C 4F 03 A5 FB 18 6D 19
 5C60 44 03 85 FB A5 FC 6D 45 9D
 5C68 03 85 FC A5 FD 18 6D 46 E0
 5C70 03 85 FD A5 FE 6D 47 03 AB
 5C78 85 FE 4C 20 5C A5 FB 18 4D
 5C80 6D 40 03 85 FB A5 FC 6D 90
 5C88 41 03 85 FC A5 FD 18 6D 1B
 5C90 42 03 85 FD A5 FE 6D 43 99
 5C98 03 85 FE 4C 20 5C 60 A9 AB
 5CA0 84 8D 26 03 A9 58 8D 27 D1
 5CA8 03 60 20 F4 5C A9 0D 20 71
 5CB0 0C E1 20 0C E1 20 0C E1 2B
 5CB8 A9 08 20 0C E1 A9 00 85 68
 5CC0 B0 A9 68 85 B1 A9 51 8D B6
 5CC8 52 03 20 0A 5D 20 ED F6 98
 5CD0 F0 1A A9 0D 20 0C E1 CE 97
 5CD8 52 03 D0 EE A9 0F 20 0C DF
 5CE0 E1 A9 0D 20 0C E1 20 0C 65
 5CE8 E1 20 0C E1 20 33 F3 A9 23
 5CF0 04 4C 91 F2 A9 00 20 F9 27
 5CF8 FD A9 04 A2 04 A0 00 20 FB
 5D00 00 FE 20 4A F3 A2 04 4C E2
 5D08 AD E4 20 52 58 A5 B0 18 E3
 5D10 69 E0 85 B4 85 0D A5 B1 C6
 5D18 69 01 85 B5 85 0E A5 B4 9F
 5D20 D0 02 C6 B5 C6 B4 A4 B4 E0
 5D28 C4 B0 D0 06 A4 B5 C4 B1 DF
 5D30 F0 08 A0 00 B1 B4 C9 80 CC
 5D38 F0 E4 E6 B4 D0 02 E6 B5 3C
 5D40 A2 00 A0 00 B1 B0 A4 B0 36
 5D48 C4 B4 D0 06 A4 B1 C4 B5 DB
 5D50 F0 11 E6 B0 D0 02 E6 B1 16
 5D58 A0 00 E8 E0 FF F0 04 D1 EA
 5D60 B0 F0 DF E0 04 B0 0E 20 02
 5D68 71 58 20 0C E1 20 52 58 E7
 5D70 CA D0 F4 F0 14 48 20 71 7A
 5D78 58 A9 1A 20 0C E1 8A 20 3F
 5D80 0C E1 68 20 0C E1 20 52 6E
 5D88 58 A4 B0 C4 B4 D0 B1 A4 91
 5D90 B1 C4 B5 D0 AB A5 0D 85 61
 5D98 B0 A5 0E 85 B1 4C 71 58 12
 5DA0 20 5A 58 20 9F 5C 20 79 E8
 5DA8 58 8D 48 03 8D 51 03 8D C2
 5DB0 55 03 8D 54 03 8D 59 03 57
 5DB8 85 02 8D 58 03 A9 00 85 8D

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| 5DC0 | 37 | A9 | 50 | 85 | 38 | A9 | 01 | 8D | 5E | 5F98 | 4F | 4D | 45 | 00 | A9 | A6 | A0 | 5F | 8B |
| 5DC8 | 57 | 03 | 60 | A2 | 00 | 2C | 54 | 03 | 37 | 5FA0 | 20 | ED | 5A | 4C | AA | 5C | 0E | 70 | 2E |
| 5DD0 | 10 | 03 | 8E | 49 | 03 | AD | 80 | 03 | 25 | 5FA8 | 52 | 49 | 4E | 54 | 20 | 47 | 52 | 41 | ED |
| 5DD8 | E8 | A0 | 13 | D9 | 17 | 5E | F0 | 04 | 27 | 5FB0 | 50 | 48 | 49 | 43 | 00 | A9 | 8F | A0 | 55 |
| 5DE0 | 88 | 10 | F8 | 60 | 98 | 0A | A8 | B9 | 8B | 5FB8 | 60 | 20 | ED | 5A | 8D | 80 | 03 | F0 | 22 |
| 5DE8 | 2B | 5E | 85 | B0 | B9 | 2C | 5E | 85 | 08 | 5FC0 | D0 | C9 | 0D | F0 | CC | 20 | 0C | E1 | FC |
| 5DF0 | B1 | 6C | B0 | 00 | 20 | E4 | 58 | B0 | D9 | 5FC8 | 29 | 3F | 8D | 52 | 03 | A9 | 9B | 85 | 95 |
| 5DF8 | 1D | 85 | A8 | 84 | A7 | 20 | CD | 5A | B5 | 5FD0 | B0 | A9 | 60 | 85 | B1 | AD | 52 | 03 | 35 |
| 5E00 | 20 | 51 | 59 | B0 | 11 | 20 | 03 | 5B | 12 | 5FD8 | F0 | 19 | A0 | 00 | B1 | B0 | F0 | 08 | 5E |
| 5E08 | E8 | 20 | E4 | 58 | B0 | 08 | 85 | AA | D5 | 5FE0 | E6 | B0 | D0 | F8 | E6 | B1 | D0 | F4 | 1E |
| 5E10 | 84 | A9 | 20 | CD | 5A | 18 | 60 | 43 | 8E | 5FE8 | CE | 52 | 03 | E6 | B0 | D0 | E6 | E6 | 79 |
| 5E18 | 2A | 48 | 44 | 4D | 2E | 41 | 4C | 45 | CD | 5FF0 | B1 | D0 | E2 | 8A | 48 | A0 | 00 | B1 | 31 |
| 5E20 | 4F | 50 | 53 | 51 | 49 | 52 | 2F | 23 | 04 | 5FF8 | B0 | F0 | 7E | 4A | 4A | 4A | 4A | 8D | B1 |
| 5E28 | 46 | 4E | 42 | 58 | 5F | 71 | 5F | 87 | A8 | 6000 | 56 | 03 | B1 | B0 | 29 | 07 | 38 | ED | C1 |
| 5E30 | 5F | 53 | 5E | 7B | 5E | BC | 5E | D4 | 1C | 6008 | 58 | 03 | 38 | ED | 58 | 03 | 29 | 07 | 17 |
| 5E38 | 5E | F6 | 5E | 39 | 5F | 9C | 5F | B5 | FB | 6010 | AA | 84 | A8 | 84 | AA | AC | 57 | 03 | 9C |
| 5E40 | 5F | 39 | 63 | 4D | 63 | 71 | 63 | 9E | F3 | 6018 | 84 | A7 | 29 | 04 | F0 | 0B | 98 | 49 | 35 |
| 5E48 | 63 | BF | 63 | E2 | 63 | 3B | 64 | 9B | 17 | 6020 | FF | A8 | C8 | 84 | A7 | A9 | FF | 85 | 12 |
| 5E50 | 64 | CB | 64 | A9 | 6D | A0 | 5E | 20 | 60 | 6028 | A8 | 8A | 29 | 03 | D0 | 04 | 85 | A7 | ED |
| 5E58 | ED | 5A | 20 | F4 | 5D | B0 | 38 | 20 | 85 | 6030 | 85 | A8 | 8A | 38 | E9 | 02 | 29 | 07 | 37 |
| 5E60 | 38 | 5B | 20 | 51 | 59 | B0 | 30 | 20 | 66 | 6038 | AA | AC | 57 | 03 | 84 | A9 | 29 | 04 | 52 |
| 5E68 | 0C | 5B | E8 | D0 | ED | 0E | 64 | 52 | B9 | 6040 | F0 | 0B | 98 | 49 | FF | A8 | C8 | 84 | CD |
| 5E70 | 41 | 57 | 20 | 4C | 49 | 4E | 45 | 20 | DF | 6048 | A9 | A9 | FF | 85 | AA | 8A | 29 | 03 | C0 |
| 5E78 | 54 | 4F | 00 | A9 | B3 | A0 | 5E | 20 | AE | 6050 | D0 | 04 | 85 | A9 | 85 | AA | A0 | 00 | 22 |
| 5E80 | ED | 5A | 20 | F4 | 5D | B0 | 10 | A5 | D8 | 6058 | B1 | B0 | 29 | 08 | D0 | 0B | 20 | 7F | 87 |
| 5E88 | A7 | 85 | FB | A5 | A8 | 85 | FC | A5 | F1 | 6060 | 60 | CE | 56 | 03 | D0 | F8 | 4C | 71 | 98 |
| 5E90 | A9 | 85 | FD | A5 | AA | 85 | FE | 60 | 3E | 6068 | 60 | 20 | 98 | 5E | CE | 56 | 03 | D0 | 25 |
| 5E98 | A5 | FB | 18 | 65 | A7 | 85 | FB | A5 | D5 | 6070 | F8 | E6 | B0 | D0 | 80 | E6 | B1 | D0 | 6D |
| 5EA0 | FC | 65 | A8 | 85 | FC | A5 | FD | 18 | F4 | 6078 | FA | 68 | AA | E8 | 4C | BC | 5F | A5 | 9E |
| 5EA8 | 65 | A9 | 85 | FD | A5 | FE | 65 | AA | A7 | 6080 | B0 | 48 | A5 | B1 | 48 | 20 | 16 | 5B | 3F |
| 5EB0 | 85 | FE | 60 | 0E | 6D | 4F | 56 | 45 | 08 | 6088 | 68 | 85 | B1 | 68 | 85 | B0 | 60 | 0E | 53 |
| 5EB8 | 20 | 54 | 4F | 00 | A9 | CB | A0 | 5E | 3B | 6090 | 70 | 52 | 49 | 4E | 54 | 20 | 54 | 45 | 5C |
| 5EC0 | 20 | ED | 5A | 20 | 82 | 5E | B0 | CF | 72 | 6098 | 58 | 54 | 00 | 3A | 2C | 16 | 24 | 22 | E2 |
| 5EC8 | 4C | 79 | 5A | 0E | 70 | 4C | 4F | 54 | 3E | 60A0 | 30 | 17 | 26 | 15 | 54 | 13 | 22 | 11 | 7C |
| 5ED0 | 20 | 41 | 54 | 00 | A9 | 00 | 8D | 48 | 20 | 60A8 | 2A | 68 | 00 | 7C | 60 | 11 | 22 | 13 | A5 |
| 5ED8 | 03 | A9 | EB | A0 | 5E | 20 | ED | 5A | 76 | 60B0 | 64 | 48 | 46 | 39 | 3A | 00 | 12 | 74 | 01 |
| 5EE0 | 20 | E4 | 58 | B0 | B2 | 8C | 48 | 03 | 50 | 60B8 | 16 | 32 | 11 | 20 | 17 | 26 | 2A | 11 | E0 |
| 5EE8 | 4C | CD | 5A | 0E | 73 | 45 | 54 | 20 | 2D | 60C0 | 10 | 17 | 26 | 5A | 00 | 1A | 15 | 54 | 6D |
| 5EF0 | 54 | 59 | 50 | 45 | 00 | 60 | A9 | 00 | 4F | 60C8 | 13 | 22 | 11 | 58 | 17 | 26 | 5A | 00 | F6 |
| 5EF8 | 8D | 51 | 03 | A9 | 01 | 8D | 4A | 03 | 52 | 60D0 | 32 | 13 | 54 | 15 | 36 | 12 | 70 | 5A | 98 |
| 5F00 | A9 | 12 | A0 | 5F | 20 | ED | 5A | 20 | 45 | 60D8 | 00 | 42 | 4E | 74 | 42 | 28 | 2F | 26 | 8C |
| 5F08 | E4 | 58 | B0 | E9 | 8C | 51 | 03 | 4C | 45 | 60E0 | 39 | 3A | 00 | 42 | 4E | 74 | 48 | 22 | 8E |
| 5F10 | CD | 5A | 0E | 73 | 45 | 54 | 20 | 44 | DB | 60E8 | 39 | 1A | 00 | 4A | 1C | 17 | 26 | 15 | EB |
| 5F18 | 4F | 54 | 20 | 4D | 4F | 44 | 45 | 00 | 1E | 60F0 | 54 | 13 | 22 | 11 | 30 | 26 | 39 | 1A | 7B |
| 5F20 | 2C | 55 | 03 | 10 | D0 | 48 | A9 | 4F | AC | 60F8 | 00 | 74 | 40 | 42 | 44 | 70 | 2A | 00 | 89 |
| 5F28 | 20 | 0C | E1 | 68 | 30 | 06 | A9 | 46 | 19 | 6100 | 1A | 22 | 16 | 74 | 16 | 22 | 39 | 48 | 80 |
| 5F30 | 20 | 0C | E1 | 2C | A9 | 4E | 4C | 0C | 63 | 6108 | 00 | 2A | 22 | 16 | 64 | 15 | 16 | 17 | 30 |
| 5F38 | E1 | 20 | E4 | 58 | B0 | B7 | 98 | 4A | 93 | 6110 | 69 | 00 | 74 | 48 | 31 | 3D | 43 | 29 | A3 |
| 5F40 | 6A | 29 | 80 | 8D | 55 | 03 | A9 | 4D | E1 | 6118 | 58 | 00 | 74 | 42 | 29 | 58 | 00 | 7C | DF |
| 5F48 | A0 | 5F | 4C | ED | 5A | 0E | 65 | 43 | 0E | 6120 | 70 | 23 | 21 | 74 | 29 | 58 | 00 | 7C | 50 |
| 5F50 | 48 | 4F | 20 | 4D | 4F | 44 | 45 | 00 | 1C | 6128 | 70 | 43 | 34 | 70 | 2A | 00 | 1A | 15 | 20 |
| 5F58 | A9 | 62 | A0 | 5F | 20 | ED | 5A | 4C | C9 | 6130 | 54 | 13 | 22 | 11 | 50 | 17 | 26 | 5A | 57 |
| 5F60 | 5A | 58 | 0E | 63 | 4C | 45 | 41 | 52 | A4 | 6138 | 00 | 74 | 30 | 32 | 11 | 20 | 17 | 36 | 5D |
| 5F68 | 20 | 47 | 52 | 41 | 50 | 48 | 49 | 43 | F8 | 6140 | 6A | 00 | 1A | 15 | 54 | 13 | 12 | 21 | A1 |
| 5F70 | 00 | A9 | 7B | A0 | 5F | 20 | ED | 5A | 04 | 6148 | 40 | 17 | 26 | 4C | 1B | 23 | 2A | 78 | AD |
| 5F78 | 4C | A0 | 5D | 0E | 69 | 4E | 49 | 54 | 0B | 6150 | 00 | 32 | 13 | 24 | 15 | 36 | 12 | 33 | 7C |
| 5F80 | 49 | 41 | 4C | 49 | 5A | 45 | 00 | A9 | 77 | 6158 | 4E | 70 | 6A | 00 | 4A | 1C | 17 | 26 | 67 |
| 5F88 | 92 | A0 | 5F | 20 | ED | 5A | 4C | 79 | 96 | 6160 | 15 | 24 | 13 | 22 | 13 | 14 | 15 | 26 | D1 |
| 5F90 | 58 | 60 | 0E | 70 | 45 | 4E | 20 | 48 | 66 | 6168 | 17 | 69 | 00 | 42 | 26 | 74 | 49 | 38 | 30 |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| 6170 | 00 | 64 | 13 | 22 | 11 | 60 | 2A | 00 | 69 |
| 6178 | 54 | 23 | 21 | 50 | 2A | 00 | 74 | 21 | 7F |
| 6180 | 23 | 70 | 2A | 00 | 14 | 43 | 24 | 4E | 85 |
| 6188 | 20 | 41 | 10 | 2A | 00 | 14 | 23 | 44 | 91 |
| 6190 | 40 | 21 | 10 | 2A | 00 | 42 | 14 | 45 | 93 |
| 6198 | 24 | 42 | 2A | 78 | 00 | 3A | 26 | 74 | 0D |
| 61A0 | 22 | 3A | 78 | 00 | 4C | 42 | 2A | 48 | 4C |
| 61A8 | 00 | 1A | 22 | 74 | 26 | 5A | 78 | 00 | D9 |
| 61B0 | 2A | 23 | 4E | 21 | 74 | 49 | 38 | 00 | 97 |
| 61B8 | 4C | 21 | 4C | 27 | 42 | 2A | 48 | 00 | B0 |
| 61C0 | 6A | 00 | 2A | 44 | 2C | 14 | 49 | 38 | 79 |
| 61C8 | 00 | 1A | 24 | 2A | 20 | 3A | 00 | 1A | AD |
| 61D0 | 74 | 2A | 70 | 1A | 2C | 46 | 3C | 42 | 80 |
| 61D8 | 2A | 58 | 00 | 2A | 74 | 2E | 28 | 13 | 03 |
| 61E0 | 22 | 11 | 10 | 17 | 26 | 17 | 11 | 22 | 08 |
| 61E8 | 13 | 2A | 28 | 00 | 4A | 1C | 14 | 45 | 0E |
| 61F0 | 14 | 3A | 20 | 1B | 26 | 1E | 38 | 20 | FF |
| 61F8 | 1B | 26 | 28 | 6A | 00 | 1A | 15 | 14 | 9B |
| 6200 | 13 | 12 | 24 | 23 | 28 | 25 | 16 | 17 | 04 |
| 6208 | 20 | 11 | 12 | 11 | 10 | 17 | 16 | 5A | C5 |
| 6210 | 00 | 2A | 14 | 15 | 5A | 28 | 00 | 3A | CD |
| 6218 | 15 | 54 | 13 | 3A | 78 | 00 | 1A | 13 | 19 |
| 6220 | 54 | 15 | 5A | 78 | 00 | 4C | 42 | 28 | EF |
| 6228 | 1E | 15 | 24 | 15 | 48 | 13 | 24 | 13 | 5E |
| 6230 | 3A | 68 | 00 | 4C | 42 | 2F | 44 | 4A | 94 |
| 6238 | 68 | 00 | 5C | 2A | 14 | 15 | 5A | 78 | 7B |
| 6240 | 00 | 4C | 42 | 2A | 48 | 00 | 5C | 2A | 46 |
| 6248 | 14 | 4A | 68 | 00 | 1C | 4A | 14 | 45 | D3 |
| 6250 | 14 | 6A | 78 | 00 | 4A | 1C | 17 | 26 | 41 |
| 6258 | 15 | 54 | 13 | 22 | 11 | 50 | 45 | 6A | 4F |
| 6260 | 58 | 00 | 2A | 74 | 12 | 26 | 68 | 11 | 51 |
| 6268 | 4A | 00 | 1C | 11 | 22 | 13 | 24 | 15 | 73 |
| 6270 | 26 | 15 | 24 | 42 | 2A | 78 | 00 | 1C | 0A |
| 6278 | 11 | 22 | 13 | 14 | 15 | 16 | 12 | 13 | EC |
| 6280 | 24 | 15 | 26 | 17 | 69 | 00 | 4A | 74 | 5D |
| 6288 | 20 | 46 | 10 | 41 | 2A | 00 | 4A | 46 | D7 |
| 6290 | 34 | 32 | 13 | 24 | 15 | 26 | 17 | 69 | 8F |
| 6298 | 00 | 3A | 26 | 15 | 54 | 13 | 22 | 11 | 29 |
| 62A0 | 20 | 17 | 36 | 39 | 3A | 00 | 42 | 24 | 5E |
| 62A8 | 25 | 34 | 4A | 78 | 00 | 1A | 22 | 13 | B9 |
| 62B0 | 14 | 15 | 26 | 15 | 24 | 13 | 22 | 11 | 32 |
| 62B8 | 20 | 17 | 26 | 17 | 10 | 18 | 6A | 00 | 7A |
| 62C0 | 1A | 22 | 13 | 54 | 15 | 26 | 17 | 2A | 94 |
| 62C8 | 29 | 36 | 17 | 20 | 11 | 5A | 00 | 2A | 78 |
| 62D0 | 2C | 14 | 2C | 14 | 49 | 28 | 00 | 2A | 27 |
| 62D8 | 2C | 14 | 2C | 14 | 15 | 59 | 28 | 00 | 14 |
| 62E0 | 4A | 1C | 16 | 35 | 33 | 12 | 2A | 78 | E2 |
| 62E8 | 00 | 3C | 42 | 2C | 46 | 59 | 1A | 00 | E2 |
| 62F0 | 1C | 12 | 33 | 35 | 16 | 18 | 69 | 00 | 6F |
| 62F8 | 1C | 11 | 22 | 13 | 14 | 15 | 26 | 14 | 2B |
| 6300 | 2C | 14 | 69 | 28 | 00 | 18 | 11 | 2E | 24 |
| 6308 | 13 | 24 | 15 | 2A | 17 | 18 | 2E | 42 | C5 |
| 6310 | 2E | 00 | 28 | 33 | 35 | 37 | 31 | 2C | 26 |
| 6318 | 00 | 29 | 44 | 46 | 40 | 42 | 2D | 00 | 00 |
| 6320 | 28 | 35 | 62 | 37 | 2C | 00 | 29 | 46 | 2D |
| 6328 | 43 | 46 | 41 | 2D | 00 | 28 | 12 | 13 | E9 |
| 6330 | 24 | 15 | 26 | 17 | 20 | 11 | 12 | 2C | 13 |
| 6338 | 00 | A0 | 63 | A9 | 5F | 20 | ED | 5A | EC |
| 6340 | 20 | E4 | 58 | B0 | 07 | C8 | 8C | 57 | 5A |
| 6348 | 03 | 20 | CD | 5A | 60 | A0 | 63 | A9 | D0 |
| 6350 | 69 | 20 | ED | 5A | 20 | E4 | 58 | B0 | C0 |
| 6358 | F3 | 8C | 58 | 03 | 4C | CD | 5A | 0E | 87 |
| 6360 | 73 | 45 | 54 | 20 | 53 | 49 | 5A | 45 | 81 |
| 6368 | 00 | 0E | 72 | 4F | 54 | 41 | 54 | 45 | 37 |
| 6370 | 00 | A0 | 63 | A9 | 8C | 20 | ED | 5A | 19 |
| 6378 | 20 | F4 | 5D | B0 | 0E | 20 | 16 | 5B | 84 |
| 6380 | 20 | 51 | 59 | B0 | 06 | 20 | 0C | 5B | 10 |
| 6388 | E8 | D0 | ED | 60 | 0E | 64 | 52 | 41 | 60 |
| 6390 | 57 | 20 | 52 | 45 | 4C | 41 | 54 | 49 | 5A |
| 6398 | 56 | 45 | 20 | 54 | 4F | 00 | A0 | 63 | 69 |
| 63A0 | A9 | AD | 20 | ED | 5A | 20 | F4 | 5D | 00 |
| 63A8 | B0 | E1 | 4C | 98 | 5E | 0E | 6D | 4F | F1 |
| 63B0 | 56 | 45 | 20 | 52 | 45 | 4C | 41 | 54 | C5 |
| 63B8 | 49 | 56 | 45 | 20 | 54 | 4F | 00 | A0 | 7D |
| 63C0 | 63 | A9 | D6 | 20 | ED | 5A | 20 | E4 | 3E |
| 63C8 | 58 | B0 | C0 | 98 | 4A | A9 | 00 | 6A | 08 |
| 63D0 | 8D | 54 | 03 | 4C | 20 | 5F | 0E | 72 | 4E |
| 63D8 | 45 | 53 | 45 | 54 | 20 | 4D | 4F | 44 | C1 |
| 63E0 | 45 | 00 | A0 | 64 | A9 | 1F | 20 | ED | 3D |
| 63E8 | 5A | 20 | E4 | 58 | B0 | 9D | 98 | 4A | 26 |
| 63F0 | A9 | 00 | 6A | 8D | 59 | 03 | 48 | 20 | 03 |
| 63F8 | 20 | 5F | 68 | 10 | 21 | A0 | 00 | A2 | F9 |
| 6400 | 50 | 84 | B0 | 86 | B1 | A9 | 10 | 91 | 7D |
| 6408 | B0 | C8 | 91 | B0 | C8 | A9 | 00 | 91 | 58 |
| 6410 | B0 | C8 | 91 | B0 | C8 | D0 | EE | E6 | C9 |
| 6418 | B1 | E8 | E0 | 58 | D0 | E7 | 60 | 6D | 2D |
| 6420 | 49 | 4E | 2D | 6D | 41 | 58 | 2D | 4D | 85 |
| 6428 | 4F | 44 | 45 | 00 | 2C | 55 | 03 | 10 | 1A |
| 6430 | 09 | 20 | 09 | 5B | A5 | D3 | C9 | 14 | 1E |
| 6438 | 90 | F7 | 60 | A9 | 92 | A0 | 64 | 20 | 87 |
| 6440 | ED | 5A | 20 | 82 | 5E | B0 | F3 | 20 | B2 |
| 6448 | 51 | 59 | B0 | EE | E8 | 20 | 0C | 5B | 33 |
| 6450 | 20 | F4 | 5D | B0 | E5 | A5 | FD | 48 | CE |
| 6458 | A5 | FE | 48 | A5 | FB | 48 | A5 | FC | A6 |
| 6460 | 48 | A5 | A9 | 48 | A5 | AA | 48 | A5 | A2 |
| 6468 | FD | 48 | A5 | FE | 48 | A9 | 02 | 8D | 71 |
| 6470 | 5A | 03 | 68 | 85 | AA | 68 | 85 | A9 | 57 |
| 6478 | 20 | 38 | 5B | CE | 5A | 03 | D0 | F2 | AA |
| 6480 | 68 | 85 | A8 | 68 | 85 | A7 | 20 | 38 | E9 |
| 6488 | 5B | 68 | 85 | AA | 68 | 85 | A9 | 4C | 0E |
| 6490 | 38 | 5B | 64 | 52 | 41 | 57 | 20 | 42 | B8 |
| 6498 | 4F | 58 | 00 | A9 | C2 | A0 | 64 | 20 | B5 |
| 64A0 | ED | 5A | 20 | E4 | 58 | B0 | 93 | 8C | 7E |
| 64A8 | 5A | 03 | 20 | CD | 5A | AD | 5A | 03 | AF |
| 64B0 | F0 | 88 | C9 | 07 | B0 | 84 | 69 | 3F | 7F |
| 64B8 | 48 | A9 | 00 | 9D | 7F | 03 | 68 | 4C | 9B |
| 64C0 | CA | 5F | 73 | 45 | 54 | 20 | 4D | 41 | D8 |
| 64C8 | 52 | 4B | 00 | A9 | DE | A0 | 64 | 20 | E0 |
| 64D0 | ED | 5A | 20 | E4 | 58 | B0 | 06 | 8C | F2 |
| 64D8 | 4A | 03 | 20 | CD | 5A | 60 | 73 | 45 | C2 |
| 64E0 | 54 | 20 | 4C | 49 | 4E | 45 | 20 | 53 | 0D |
| 64E8 | 43 | 41 | 4C | 45 | 00 | FF | FF | FF | 0A |
| 64F0 | 00 | FF | 00 | FF | 00 | FF | 00 | FF | 07 |
| 64F8 | 00 | FF | 00 | FF | 90 | 7E | 80 | 00 | 96 |

Listing 1: Hexdump mit Checker eingeben. Startadresse ist 5800

| | | | |
|--|------|--|------|
| 10 OPEN 12,12 | 643 | 700 PRINT#12,"P ROTATE- " | 1328 |
| 20 PRINT#12,"*" | 786 | 710 PRINT#12,"Q0" | 652 |
| 40 : | | 720 PRINT#12,"P BEFEHL..." | 1517 |
| 50 PRINT#12,"#1":AL=30/180*[PI] | 2061 | 730 : | |
| 60 FOR X=0 TO 110 STEP 10 | 1307 | 740 PRINT#12,"M250,190" | 1648 |
| 70 FOR Y=0 TO 110 STEP 10 | 1309 | 750 PRINT#12,"PUNTERSCHIEDLICHE | 3259 |
| 75 PRINT X,Y | 707 | LINIENTYPEN" | |
| 80 : | | 760 PRINT#12,"M250,215":PRINT#12,"L0" | 2566 |
| 90 X1=350+X-COS (AL)*Y | 1758 | 770 PRINT#12,"PGESTRICHELT (1)" | 2199 |
| 100 X2=X1+10 | 788 | 780 PRINT#12,"M360,220" | 1533 |
| 110 X3=350+X-COS (AL)*(Y+10) | 2255 | 790 PRINT#12,"L1":PRINT#12,"B1" | 1851 |
| 120 Y1=440-SIN (X/110*[PI])+4*COS (Y/ | 5107 | 800 PRINT#12,"D460,220":PRINT#12,"L0" | 2207 |
| 110*[PI])*110-SIN (AL)*Y | | 810 PRINT#12,"M250,225" | 1676 |
| 130 Y2=440-SIN ((X+10)/110*[PI])+4*COS | 5370 | 820 PRINT#12,"P[12SPACES](2)" | 1485 |
| (Y/110*[PI])*110-SIN (AL)*Y | | 830 PRINT#12,"M360,230" | 1541 |
| 140 Y3=440-SIN (X/110*[PI])+4*COS ((Y+ | 6054 | 840 PRINT#12,"L1":PRINT#12,"B2" | 1724 |
| 10)/110*[PI])*110-SIN (AL)*(Y+10) | | 850 PRINT#12,"D460,230":PRINT#12,"L0" | 2215 |
| 150 PRINT#12,"M*X1",Y1 | 1427 | 860 PRINT#12,"M250,235" | 1684 |
| 160 IF X<>110 THEN PRINT#12,"D*X2",Y2 | 2369 | 870 PRINT#12,"P[12SPACES](3)" | 1493 |
| 170 PRINT#12,"M*X1",Y1 | 1427 | 880 PRINT#12,"M360,240" | 1549 |
| 180 IF Y<>110 THEN PRINT#12,"D*X3",Y3 | 2404 | 890 PRINT#12,"L1":PRINT#12,"B3" | 1852 |
| 190 : | | 900 PRINT#12,"D460,240" | 1325 |
| 200 NEXT Y,X | 678 | 905 PRINT#12,"M250,245":PRINT#12,"L0" | 2590 |
| 205 PRINT#12,"E1" | 842 | 910 PRINT#12,"PDURCHGEZOGEN[3SPACES]" | 2042 |
| 210 : | | 920 PRINT#12,"M360,250" | 1557 |
| 220 PRINT#12,"#0" | 710 | 930 PRINT#12,"L0" | 747 |
| 230 PRINT#12,"L1":PRINT#12,"B1" | 1851 | 940 PRINT#12,"D460,250" | 1333 |
| 240 PRINT#12,"M0,60" | 1070 | 950 PRINT#12,"M250,255":PRINT#12,"L0" | 2598 |
| 250 PRINT#12,"D479,60,479,540,0,540,0, | 3790 | 960 PRINT#12,"PPUNKTIERT[3SPACES](1)" | 1966 |
| 60" | | 970 PRINT#12,"M360,260" | 1565 |
| 260 PRINT#12,"M0,180" | 1154 | 980 PRINT#12,"L2" | 875 |
| 270 PRINT#12,"D479,180" | 1490 | 990 PRINT#12,"D460,260" | 1341 |
| 280 PRINT#12,"M0,300" | 1140 | 1000 PRINT#12,"M250,265":PRINT#12,"L0" | 2606 |
| 290 PRINT#12,"D479,300" | 1434 | 1010 PRINT#12,"P[12SPACES](2)" | 1485 |
| 300 PRINT#12,"M240,180" | 1512 | 1020 PRINT#12,"M360,270" | 1573 |
| 310 PRINT#12,"D240,540" | 1463 | 1030 PRINT#12,"L3" | 939 |
| 320 : | | 1040 PRINT#12,"D460,270" | 1349 |
| 330 PRINT#12,"M160,0" | 1198 | 1050 : | |
| 340 PRINT#12,"S1":PRINT#12,"L0" | 1587 | 1060 PRINT#12,"M250,520":PRINT#12,"L0" | 2506 |
| 350 PRINT#12,"PLOTTER-DEMO" | 1771 | 1070 PRINT#12,"PHIDDEN-LINE-DEMO" | 2379 |
| 360 PRINT#12,"S0" | 716 | 1080 : | |
| 370 PRINT#12,"M160,30" | 1356 | 1090 PRINT#12,"M160,320" | 1409 |
| 380 PRINT#12,"PBY MATTHIAS KOEFFERLEIN" | 3110 | 1100 PRINT#12,"D200,320,200,400,160,400, | 3646 |
| 390 PRINT#12,"M200,40" | 1421 | 160,320" | |
| 400 PRINT#12,"P(C) 1985" | 1697 | 1110 FOR I=180 TO 200 | 961 |
| 410 : | | 1120 PRINT#12,"M" I ",320 | 1502 |
| 420 PRINT#12,"M10,70" | 1213 | 1130 PRINT#12,"D" I ",400 | 1445 |
| 430 PRINT#12,"PES GIBT UNTERSCHIEDLICHE" | 2740 | 1140 NEXT I | 276 |
| 440 PRINT#12,"M10,80" | 1215 | 1150 PRINT#12,"A0" | 650 |
| 450 PRINT#12,"PS" | 820 | 1160 PRINT#12,"M140,340" | 1424 |
| 460 PRINT#12,"S1":PRINT#12,"PC" | 1705 | 1170 PRINT#12,"D220,340" | 1454 |
| 470 PRINT#12,"S2":PRINT#12,"PH" | 1644 | 1180 PRINT#12,"A1" | 714 |
| 480 PRINT#12,"S3":PRINT#12,"PR" | 1713 | 1190 PRINT#12,"M140,360" | 1440 |
| 490 PRINT#12,"S4":PRINT#12,"PI" | 1645 | 1200 PRINT#12,"D220,360" | 1470 |
| 500 PRINT#12,"S5":PRINT#12,"PF" | 1580 | 1210 PRINT#12,"A2" | 770 |
| 510 PRINT#12,"S6":PRINT#12,"PT" | 1651 | 1220 PRINT#12,"M140,380" | 1456 |
| 520 PRINT#12,"S7":PRINT#12,"PE" | 1835 | 1230 PRINT#12,"D220,380":PRINT#12,"A0" | 2208 |
| 530 PRINT#12,"S8":PRINT#12,"PN" | 1521 | 1240 PRINT#12,"M10,330" | 1363 |
| 600 PRINT#12,"M10,160":PRINT#12,"S0" | 2131 | 1250 PRINT#12,"PLINIE TYP 0 (SET)" | 1907 |
| 610 PRINT#12,"P01 2[2SPACES]3[3SPACES]4 | 2647 | 1260 PRINT#12,"M10,350" | 1371 |
| [4SPACES]5[5SPACES]6[6SPACES]7 | | 1270 PRINT#12,"PLINIE TYP 1 (RES)" | 1868 |
| [7SPACES]8" | | 1280 PRINT#12,"M10,370" | 1379 |
| 620 : | | 1290 PRINT#12,"PLINIE TYP 2 (EOR)" | 1893 |
| 630 PRINT#12,"M10,190":PRINT#12,"S0" | 2143 | 1300 : | |
| 640 PRINT#12,"P...UND UNTERSCHIEDLICHE " | 2570 | 1310 PRINT#12,"L1":PRINT#12,"B1" | 1851 |
| 650 PRINT#12,"Q3" | 844 | 1320 PRINT#12,"M0,420" | 1145 |
| 660 PRINT#12,"P SCHRIFTRICH- " | 2102 | 1330 PRINT#12,"D240,420" | 1443 |
| 670 PRINT#12,"Q2" | 780 | 1340 : | |
| 680 PRINT#12,"P TUNGEN DURCH DEN " | 2055 | 1350 PRINT#12,"M10,450" | 1373 |
| 690 PRINT#12,"Q1" | 716 | 1360 PRINT#12,"L0" | 747 |

```

1365 FOR I=64 TO 95
1370 PRINT#12,"P" CHR# (I)
1375 NEXT I
1380 PRINT#12,"M10,460"
1390 FOR I=32 TO 63
1400 PRINT#12,"P" CHR# (I)
1410 NEXT I
1420 :
1430 PRINT#12,"M10,430"
1440 PRINT#12,"PZEICHENVORRAT"
1450 :
1460 PRINT#12,"M20,500"
1470 PRINT#12,"N1"
1480 PRINT#12,"M40,500"
1490 PRINT#12,"N2"
1500 PRINT#12,"M60,500"
1510 PRINT#12,"N3"
1520 PRINT#12,"M80,500"
1530 PRINT#12,"N4"
1540 PRINT#12,"M100,500"
1550 PRINT#12,"N5"
1560 PRINT#12,"M120,500"
1101
1304
276
1377
1107
1304
276
1365
2177
1419
875
1292
939
1420
1003
1293
812
1398
876
1399
1570 PRINT#12,"N6"
1600 PRINT#12,"S1"
1610 PRINT#12,"M20,520"
1620 PRINT#12,"N1"
1630 PRINT#12,"M40,520"
1640 PRINT#12,"N2"
1650 PRINT#12,"M60,520"
1660 PRINT#12,"N3"
1670 PRINT#12,"M80,520"
1680 PRINT#12,"N4"
1690 PRINT#12,"M100,520"
1700 PRINT#12,"N5"
1710 PRINT#12,"M120,520"
1720 PRINT#12,"N6"
1730 :
1740 PRINT#12,"M10,480":PRINT#12,"S0"
1750 PRINT#12,"PMARKEN"
1760 :
1770 PRINT#12,"0"
1780 CLOSE 12
1790 :
940
780
1427
875
1300
939
1428
1003
1301
812
1414
876
1415
940
2145
1235
950
458

```

Listing 2: Das Demo-Programm erzeugt das Bild auf Seite 85

```

10 REM *****
20 REM * *
30 REM * PLOTTER-EMULATOR *
40 REM * FUER MPS 801 / GP 100 VC *
50 REM * VON W.WINTERSBERGER *
60 REM * UND M.KOEFFERLEIN *
70 REM * *
80 REM *****
90 :
91 FOR A=22528 TO 25849:READ D:POKE A, 2912
D:NEXT
92 SYS 22528:NEW 892
93 :
100 DATA 32,160,93,169,10,160,88,76,30, 5013
171,13,14,32,112,76,79
101 DATA 84,84,69,82,45,69,77,85,76,65, 5237
84,79,82,13,32,109
102 DATA 112,115,32,56,48,49,47,103,112, 5813
32,49,48,48,32,118,99
103 DATA 13,13,32,40,99,41,32,49,57,56, 5057
53,32,109,65,84,84
104 DATA 72,73,85,83,32,107,79,69,70,70, 5228
69,82,76,69,73,78
105 DATA 13,0,72,120,169,52,133,1,104, 5184
96,32,82,88,160,0,162
106 DATA 104,132,176,134,177,169,128, 7781
145,176,200,208,251,230,177,232,208
107 DATA 246,72,169,55,133,1,88,104,96, 6991
169,0,133,251,133,252,133
108 DATA 253,133,254,96,72,165,154,201, 6476
12,208,86,104,133,13,72,152
109 DATA 72,138,72,164,2,240,7,173,128, 5184
3,201,80,240,0,165,13
110 DATA 41,127,201,32,240,22,165,13,41, 5251
127,201,14,144,21,201,32
111 DATA 144,10,153,128,3,200,192,80, 5553
240,9,132,2,104,170,104,168
112 DATA 104,24,96,169,13,153,128,3,200, 5184
169,0,133,2,153,128,3
113 DATA 169,3,133,154,32,203,93,32,6, 5978
91,169,12,133,154,76,188
114 DATA 88,76,203,241,169,0,133,20,133, 5652
21,189,128,3,240,92,201
115 DATA 45,208,4,232,169,128,44,169,0, 5533
133,180,189,128,3,240,75
116 DATA 201,48,144,48,201,58,176,44,6, 5793
20,38,21,165,21,133,176
117 DATA 165,20,10,38,176,10,38,176,24, 5673
101,20,133,20,8,189,128
118 DATA 3,41,15,24,101,20,133,20,165, 5315
21,105,0,40,101,176,133
119 DATA 21,232,208,199,24,36,180,16,17, 5491
165,20,73,255,105,1,133
120 DATA 20,165,21,73,255,185,0,133,21, 5310
24,36,56,164,20,165,21
121 DATA 96,189,128,3,240,245,201,44, 6311
240,218,232,208,244,165,252,48
122 DATA 234,240,10,201,1,208,228,165, 6652
251,201,224,176,222,165,254,48
123 DATA 218,164,253,201,2,144,6,208, 6080
210,192,55,176,206,132,13,133
124 DATA 14,44,89,3,16,83,165,251,133, 5792
176,165,252,6,176,42,6
125 DATA 176,42,24,105,80,133,177,165, 6305
253,56,160,0,241,176,165,254
126 DATA 200,241,176,176,13,165,253,136, 6514
145,176,165,254,200,145,176,169
127 DATA 128,44,169,0,133,181,200,56, 7257
177,176,229,253,200,177,176,229
128 DATA 254,176,13,136,165,253,145,176, 6486
200,165,254,145,176,169,128,44
129 DATA 169,0,5,181,208,3,76,75,89,160, 5966
1,132,181,169,192,133
130 DATA 180,136,132,176,160,7,165,13, 6282
56,229,180,72,165,14,229,181
131 DATA 144,6,133,14,104,133,13,36,104, 6018
38,176,70,181,102,180,136
132 DATA 208,228,165,176,74,182,180,74, 6478
182,180,74,182,180,133,181,169
133 DATA 0,133,176,133,177,160,4,165, 5601
180,24,181,176,133,176,165,181
134 DATA 181,177,133,177,6,180,38,181, 7303
136,208,236,165,176,24,181,251
135 DATA 133,176,165,177,181,252,105, 7034
184,133,177,32,82,88,164,13,185
136 DATA 188,90,172,72,3,240,28,192,1, 5088

```

| | | |
|---|------|--|
| 240,13,192,2,208,20,160 | | |
| 137 DATA 0,81,176,145,176,76,113,88,160,8,73,255,49,176,145,176 | 6394 | |
| 138 DATA 76,113,88,160,0,17,176,145,176,76,113,88,1,2,4,8 | 4535 | |
| 139 DATA 16,32,64,169,0,141,73,3,96,173,81,3,240,67,201,1 | 4742 | |
| 140 DATA 240,42,201,2,240,23,201,4,176,55,173,73,3,201,5,144 | 5935 | |
| 141 DATA 45,240,54,201,7,240,50,201,9,176,41,144,33,173,73,3 | 5423 | |
| 142 DATA 201,5,144,26,240,35,201,7,176,26,144,18,173,73,3,205 | 6230 | |
| 143 DATA 74,3,144,10,237,74,3,205,74,3,176,8,144,11,238,73 | 5276 | |
| 144 DATA 3,76,93,89,169,255,141,73,3,238,73,3,96,44,85,3 | 5211 | |
| 145 DATA 16,48,133,176,138,72,165,176,32,145,179,32,221,189,174,0 | 6075 | |
| 146 DATA 1,224,32,208,2,185,0,32,30,171,184,170,96,44,85,3 | 5126 | |
| 147 DATA 16,16,141,86,3,138,72,173,86,3,32,30,171,184,170,76 | 5976 | |
| 148 DATA 44,180,96,169,44,44,169,13,44,169,32,44,169,58,44,85 | 5428 | |
| 149 DATA 3,16,239,76,12,225,165,167,133,176,165,168,73,128,10,165 | 6016 | |
| 150 DATA 168,133,177,5,176,8,165,169,133,180,165,170,73,128,10,165 | 7485 | |
| 151 DATA 170,133,181,5,180,76,97,91,165,167,56,229,251,133,176,165 | 6600 | |
| 152 DATA 168,229,252,133,177,73,128,10,165,177,5,176,8,165,169,56 | 6859 | |
| 153 DATA 229,253,133,180,165,170,229,254,133,181,73,128,10,165,181,5 | 6657 | |
| 154 DATA 180,240,7,176,3,169,255,44,169,1,141,70,3,141,68,3 | 5474 | |
| 155 DATA 186,73,128,141,71,3,141,67,3,169,0,40,240,7,176,3 | 5266 | |
| 156 DATA 169,255,44,169,1,141,68,3,141,64,3,186,73,128,141,69 | 5282 | |
| 157 DATA 3,141,65,3,165,176,141,75,3,164,177,140,76,3,16,16 | 4987 | |
| 158 DATA 73,255,24,185,1,141,75,3,152,73,255,185,0,141,76,3 | 5674 | |
| 159 DATA 165,180,141,77,3,164,181,140,78,3,16,16,73,255,24,185 | 5886 | |
| 160 DATA 1,141,77,3,152,73,255,185,0,141,78,3,173,75,3,56 | 5283 | |
| 161 DATA 237,77,3,173,76,3,237,78,3,176,39,169,0,141,64,3 | 5594 | |
| 162 DATA 141,65,3,173,77,3,72,173,78,3,72,173,75,3,141,77 | 5896 | |
| 163 DATA 3,173,76,3,141,78,3,184,141,76,3,184,141,75,3,76 | 5271 | |
| 164 DATA 10,92,169,0,141,66,3,141,67,3,173,76,3,141,83,3 | 5335 | |
| 165 DATA 141,80,3,173,75,3,141,82,3,78,80,3,186,141,79,3 | 4464 | |
| 166 DATA 32,121,90,173,82,3,208,3,206,83,3,206,82,3,44,83 | 4259 | |
| 167 DATA 3,48,187,173,79,3,24,189,77,3,141,79,3,173,80,3 | 5557 | |
| 168 DATA 189,78,3,141,80,3,173,79,3,56,237,75,3,168,173,80 | 5442 | |
| 169 DATA 3,237,76,3,144,39,141,80,3,140,79,3,165,251,24,189 | 5068 | |
| 170 DATA 68,3,133,251,165,252,189,69,3,133,252,165,253,24,189,70 | 6001 | |
| 171 DATA 3,133,253,165,254,189,71,3,133,254,76,32,92,165,251,24 | 5713 | |
| 172 DATA 189,64,3,133,251,165,252,189,65,3,133,252,165,253,24,189 | 6321 | |
| 173 DATA 66,3,133,253,165,254,189,67,3,133,254,76,32,92,96,169 | 5934 | |
| 174 DATA 132,141,38,3,169,88,141,39,3,96,32,244,92,169,13,32 | 5354 | |
| 175 DATA 12,225,32,12,225,32,12,225,169,8,32,12,225,169,0,133 | 5386 | |
| 176 DATA 176,169,184,133,177,169,81,141,82,3,32,10,93,32,237,246 | 6730 | |
| 177 DATA 240,26,169,13,32,12,225,206,82,3,208,238,169,15,32,12 | 6259 | |
| 178 DATA 225,169,13,32,12,225,32,12,225,32,12,225,32,51,243,169 | 5922 | |
| 179 DATA 4,76,145,242,169,0,32,249,253,169,4,162,4,160,0,32 | 4979 | |
| 180 DATA 0,254,32,74,243,162,4,76,173,228,32,82,88,165,176,24 | 6063 | |
| 181 DATA 105,224,133,180,133,13,165,177,105,1,133,181,133,14,165,180 | 6950 | |
| 182 DATA 208,2,198,181,198,180,164,180,196,176,208,6,164,181,196,177 | 6536 | |
| 183 DATA 240,8,160,0,177,180,201,128,240,228,230,180,208,2,230,181 | 6405 | |
| 184 DATA 162,0,160,0,177,176,164,176,196,180,208,6,164,177,196,181 | 6614 | |
| 185 DATA 240,17,230,176,208,2,230,177,160,0,232,224,255,240,4,209 | 6019 | |
| 186 DATA 176,240,223,224,4,176,14,32,113,88,32,12,225,32,82,88 | 6021 | |
| 187 DATA 202,208,244,240,20,72,32,113,88,169,26,32,12,225,138,32 | 5931 | |
| 188 DATA 12,225,184,32,12,225,32,82,88,164,176,196,180,208,177,164 | 5883 | |
| 189 DATA 177,196,181,208,171,165,13,133,176,165,14,133,177,76,113,88 | 6706 | |
| 190 DATA 32,90,88,32,159,92,32,121,88,141,72,3,141,81,3,141 | 5509 | |
| 191 DATA 85,3,141,84,3,141,89,3,133,2,141,88,3,169,0,133 | 5641 | |
| 192 DATA 55,169,80,133,56,169,1,141,87,3,96,162,0,44,84,3 | 5152 | |
| 193 DATA 16,3,142,73,3,173,128,3,232,160,19,217,23,94,240,4 | 5370 | |
| 194 DATA 136,16,248,96,152,10,168,185,43,94,133,176,185,44,94,133 | 6028 | |
| 195 DATA 177,188,176,0,32,228,88,176,29,133,168,132,167,32,205,90 | 5954 | |
| 196 DATA 32,81,89,176,17,32,3,91,232,32,228,88,176,8,133,170 | 5919 | |
| 197 DATA 132,169,32,205,90,24,96,67,42,72,68,77,46,65,76,69 | 5780 | |
| 198 DATA 79,80,83,81,73,82,47,35,70,78,66,88,95,113,95,135 | 5566 | |
| 199 DATA 95,83,94,123,94,188,94,212,94,246,94,57,95,156,95,181 | 5741 | |
| 200 DATA 95,57,99,77,99,113,99,158,99,191,99,226,99,59,100,155 | 6239 | |
| 201 DATA 180,203,180,169,189,160,94,32,237,90,32,244,93,176,56,32 | 6365 | |
| 202 DATA 56,91,32,81,89,176,48,32,12,91,232,208,237,14,100,82 | 5078 | |
| 203 DATA 65,87,32,76,73,78,69,32,84,79,0,169,179,160,94,32 | 5995 | |
| 204 DATA 237,90,32,244,93,176,16,165,167,133,251,165,168,133,252,165 | 6949 | |
| 205 DATA 169,133,253,165,170,133,254,96,165,251,24,181,167,133,251,165 | 7095 | |
| 206 DATA 252,181,168,133,252,165,253,24,181,169,133,253,165,254,181,170 | 6995 | |
| 207 DATA 133,254,96,14,189,79,86,69,32,5496 | 5496 | |

| | | | |
|--|------|--|--|
| 84,79,0,169,203,160,94 | | | |
| 208 DATA 32,237,90,32,130,94,176,207,76, 121,90,14,112,76,79,84 | 6245 | | |
| 209 DATA 32,65,84,0,169,0,141,72,3,169, 235,160,94,32,237,90 | 5292 | | |
| 210 DATA 32,228,88,176,178,140,72,3,76, 205,90,14,115,69,84,32 | 5427 | | |
| 211 DATA 84,89,80,69,0,96,169,0,141,81, 3,169,1,141,74,3 | 5154 | | |
| 212 DATA 169,18,160,95,32,237,90,32,228, 88,176,233,140,81,3,76 | 5421 | | |
| 213 DATA 205,90,14,115,69,84,32,68,79, 84,32,77,79,68,69,0 | 5431 | | |
| 214 DATA 44,85,3,16,208,72,169,79,32,12, 225,104,48,6,169,70 | 5104 | | |
| 215 DATA 32,12,225,44,169,78,76,12,225, 32,228,88,176,183,152,74 | 6094 | | |
| 216 DATA 106,41,128,141,85,3,169,77,160, 95,76,237,90,14,101,67 | 6070 | | |
| 217 DATA 72,79,32,77,79,68,69,0,169,98, 160,95,32,237,90,76 | 5490 | | |
| 218 DATA 90,88,14,99,76,69,65,82,32,71, 82,65,80,72,73,67 | 5156 | | |
| 219 DATA 0,169,123,160,95,32,237,90,76, 160,93,14,105,78,73,84 | 5983 | | |
| 220 DATA 73,65,76,73,90,69,0,169,146, 160,95,32,237,90,76,121 | 5690 | | |
| 221 DATA 88,96,14,112,69,78,32,72,79,77, 69,0,169,166,160,95 | 6286 | | |
| 222 DATA 32,237,90,76,170,92,14,112,82, 73,78,84,32,71,82,65 | 5313 | | |
| 223 DATA 80,72,73,67,0,169,143,160,96, 32,237,90,189,128,3,240 | 5501 | | |
| 224 DATA 208,201,13,240,204,32,12,225, 41,63,141,82,3,169,155,133 | 6119 | | |
| 225 DATA 176,169,96,133,177,173,82,3, 240,25,160,0,177,176,240,0 | 6063 | | |
| 226 DATA 230,176,208,248,230,177,208, 244,208,82,3,230,176,208,230,230 | 7018 | | |
| 227 DATA 177,208,226,138,72,160,0,177, 176,240,126,74,74,74,74,141 | 6214 | | |
| 228 DATA 86,3,177,176,41,7,56,237,88,3, 56,237,88,3,41,7 | 5562 | | |
| 229 DATA 170,132,168,132,170,172,87,3, 132,167,41,4,240,11,152,73 | 6547 | | |
| 230 DATA 255,168,200,132,167,169,255, 133,168,138,41,3,208,4,133,167 | 7102 | | |
| 231 DATA 133,168,138,56,233,2,41,7,170, 172,87,3,132,169,41,4 | 5836 | | |
| 232 DATA 240,11,152,73,255,168,200,132, 169,169,255,133,170,138,41,3 | 5301 | | |
| 233 DATA 208,4,133,169,133,170,160,0, 177,176,41,8,208,11,32,127 | 6338 | | |
| 234 DATA 96,206,86,3,208,248,76,113,96, 32,152,94,206,86,3,208 | 5899 | | |
| 235 DATA 248,230,176,208,128,230,177, 208,250,104,170,232,76,188,95,165 | 6854 | | |
| 236 DATA 176,72,165,177,72,32,22,91,104, 133,177,104,133,176,96,14 | 6239 | | |
| 237 DATA 112,82,73,78,84,32,84,69,88,84, 0,58,44,22,36,34 | 5144 | | |
| 238 DATA 48,23,38,21,84,19,34,17,42,104, 0,124,96,17,34,19 | 4648 | | |
| 239 DATA 100,72,70,57,58,0,18,116,22,50, 17,32,23,38,42,17 | 5582 | | |
| 240 DATA 16,23,38,90,0,26,21,84,19,34, 17,88,23,38,90,0 | 4697 | | |
| 241 DATA 50,19,84,21,54,18,112,90,0,66, 78,116,66,40,47,38 | 4944 | | |
| 242 DATA 57,58,0,66,78,116,72,34,57,26, 0,74,28,23,38,21 | 4493 | | |
| 243 DATA 84,19,34,17,48,38,57,26,0,116, 64,66,68,112,42,0 | 4863 | | |
| 244 DATA 26,34,22,116,22,34,57,72,0,42, 34,22,100,21,22,23 | 4673 | | |
| 245 DATA 105,0,116,72,49,61,67,41,88,0, 116,66,41,88,0,124 | 5209 | | |
| 246 DATA 112,35,33,116,41,88,0,124,112, 67,52,112,42,0,26,21 | 5117 | | |
| 247 DATA 84,19,34,17,80,23,38,90,0,116, 48,50,17,32,23,54 | 4783 | | |
| 248 DATA 106,0,26,21,84,19,18,33,64,23, 38,76,27,35,42,120 | 5247 | | |
| 249 DATA 0,50,19,36,21,54,18,51,78,112, 106,0,74,28,23,38 | 5175 | | |
| 250 DATA 21,36,19,34,19,20,21,38,23,105, 0,66,38,116,73,56 | 4955 | | |
| 251 DATA 0,100,19,34,17,96,42,0,84,35, 33,80,42,0,116,33 | 5079 | | |
| 252 DATA 35,112,42,0,20,67,36,78,32,65, 16,42,0,20,35,68 | 4991 | | |
| 253 DATA 64,33,16,42,0,66,20,69,36,66, 42,120,0,58,38,116 | 5050 | | |
| 254 DATA 34,58,120,0,76,66,42,72,0,26, 34,116,38,90,120,0 | 5038 | | |
| 255 DATA 42,35,78,33,116,73,56,0,76,33, 76,39,66,42,72,0 | 5103 | | |
| 256 DATA 106,0,42,68,44,28,73,56,0,26, 36,42,32,58,0,26 | 4861 | | |
| 257 DATA 116,42,112,26,44,70,68,66,42, 88,0,42,116,46,40,19 | 5355 | | |
| 258 DATA 34,17,16,23,38,23,17,34,19,42, 40,0,74,28,20,69 | 5014 | | |
| 259 DATA 20,58,32,27,38,30,56,32,27,38, 40,106,0,26,21,20 | 4660 | | |
| 260 DATA 19,18,36,35,40,37,22,23,32,17, 18,17,16,23,22,90 | 5082 | | |
| 261 DATA 0,42,20,21,90,40,0,58,21,84,19, 58,120,0,26,19 | 3997 | | |
| 262 DATA 84,21,90,120,0,76,66,40,30,21, 36,21,72,19,36,19 | 4913 | | |
| 263 DATA 58,104,0,76,66,47,68,74,104,0, 92,42,20,21,90,120 | 5177 | | |
| 264 DATA 0,76,66,42,72,0,92,42,20,74, 104,0,28,74,20,69 | 4595 | | |
| 265 DATA 20,106,120,0,74,28,23,38,21,84, 19,34,17,80,69,106 | 4821 | | |
| 266 DATA 88,0,42,116,18,38,104,17,74,0, 28,17,34,19,36,21 | 4929 | | |
| 267 DATA 38,21,36,66,42,120,0,28,17,34, 19,20,21,22,18,19 | 5160 | | |
| 268 DATA 36,21,38,23,105,0,74,116,32,70, 16,65,42,0,74,70 | 5250 | | |
| 269 DATA 52,50,19,36,21,38,23,105,0,58, 38,21,84,19,34,17 | 4766 | | |
| 270 DATA 32,23,54,57,58,0,66,36,37,52, 74,120,0,26,34,19 | 5182 | | |
| 271 DATA 20,21,38,21,36,19,34,17,32,23, 38,23,16,24,106,0 | 4959 | | |
| 272 DATA 26,34,19,84,21,38,23,42,41,54, 23,32,17,90,0,42 | 5057 | | |
| 273 DATA 44,20,44,20,73,40,0,42,44,20, 44,20,21,89,40,0 | 4559 | | |
| 274 DATA 74,28,22,53,51,18,42,120,0,60, 66,44,70,89,26,0 | 4624 | | |
| 275 DATA 28,18,51,53,22,24,105,0,28,17, 34,19,20,21,38,20 | 4960 | | |
| 276 DATA 44,20,105,40,0,24,17,46,19,36, 21,42,23,24,46,66 | 4783 | | |
| 277 DATA 46,0,40,51,53,55,49,44,0,41,68, 70,64,66,45,0 | 4668 | | |
| 278 DATA 40,53,98,55,44,0,41,70,67,70, 4810 | 4810 | | |

| | |
|--|------|
| 65,45,0,40,18,19 | |
| 279 DATA 36,21,38,23,32,17,10,44,0,160,99,169,95,32,237,90 | 5228 |
| 280 DATA 32,228,88,176,7,200,140,87,3,32,205,90,96,160,99,169 | 5438 |
| 281 DATA 105,32,237,90,32,228,88,176,243,140,88,3,76,205,90,14 | 5874 |
| 282 DATA 115,69,84,32,83,73,90,69,0,14,114,79,84,65,84,69 | 4961 |
| 283 DATA 0,160,99,169,140,32,237,90,32,244,93,176,14,32,22,91 | 5875 |
| 284 DATA 32,81,89,176,6,32,12,91,232,208,237,96,14,100,82,65 | 5284 |
| 285 DATA 87,32,82,69,76,65,84,73,86,69,32,84,79,0,160,99 | 5646 |
| 286 DATA 169,173,32,237,90,32,244,93,176,225,76,152,94,14,109,79 | 6946 |
| 287 DATA 86,69,32,82,69,76,65,84,73,86,69,32,84,79,0,160 | 5052 |
| 288 DATA 99,169,214,32,237,90,32,228,88,176,192,152,74,169,0,106 | 5474 |
| 289 DATA 141,84,3,76,32,95,14,114,69,83,69,84,32,77,79,68 | 5377 |
| 290 DATA 69,0,160,100,169,31,32,237,90,32,228,88,176,157,152,74 | 5361 |
| 291 DATA 169,0,106,141,89,3,72,32,32,95,104,16,33,160,0,162 | 5106 |
| 292 DATA 80,132,176,134,177,169,16,145,176,200,145,176,200,169,0,145 | 5727 |
| 293 DATA 176,200,145,176,200,208,238,230,177,232,224,88,208,231,96,109 | 6715 |
| 294 DATA 73,78,45,109,65,88,45,77,79,68,69,0,44,85,3,16 | 5310 |
| 295 DATA 9,32,9,91,165,211,201,20,144,247,96,169,146,160,100,32 | 5915 |
| 296 DATA 237,90,32,130,94,176,243,32,81,89,176,238,232,32,12,91 | 5670 |
| 297 DATA 32,244,93,176,229,165,253,72,165,254,72,165,251,72,165,252 | 6203 |
| 298 DATA 72,165,169,72,165,170,72,165,253,72,165,254,72,169,2,141 | 6400 |
| 299 DATA 90,3,104,133,170,104,133,169,32,56,91,206,90,3,208,242 | 5988 |
| 300 DATA 104,133,168,104,133,167,32,56,91,104,133,170,104,133,169,76 | 6875 |
| 301 DATA 56,91,100,82,65,87,32,66,79,88,0,169,194,160,100,32 | 5561 |
| 302 DATA 237,90,32,228,88,176,147,140,90,3,32,205,90,173,90,3 | 5430 |
| 303 DATA 240,136,201,7,176,132,105,63,72,169,0,157,127,3,104,76 | 6659 |
| 304 DATA 202,95,115,69,84,32,77,65,82,75,0,169,222,160,100,32 | 5595 |
| 305 DATA 237,90,32,228,88,176,6,140,74,3,32,205,90,96,115,69 | 5346 |
| 306 DATA 84,32,76,73,78,69,32,83,67,65,76,69,0,255,255,255 | 5261 |
| 307 DATA 0,255,0,255,0,255,0,255,0,255 | 3262 |

Das Neueste von TurboAccess (TA)

TA gibt es jetzt auch für den Commodore-Rechner 262/32 in Verbindung mit der Floppy 1541. Das bedeutet:

- keine zusätzlichen Kosten für die Neuanschaffung eines Floppy
- 10fache Geschwindigkeit auch im C-64-Modus
- CFM-Laufweg
- man braucht auf den Komfort des TA nicht zu verzichten

Lieferung des TA für den PC 128 voraussichtlich Ende September

TurboTrans, die neue Generation (TT)

Um wieder TurboTrans schneller zu wissen, wir nicht genau, so etwa 200mal schneller beim Laden und beim Speichern von Bytes - 200fache werden mit weniger als 1 (einem) Sekunde geladen oder gespeichert.

Wie das funktioniert?

Wir setzen 256 Kbytes Ram mit einem neuen Betriebssystem in die Floppy. Wenn der Anwender kurz mit der eingesteckten Diskette arbeiten möchte, wird wie beim TA bis zu 10mal schneller geladen. Soll die Diskette jedoch länger benutzt werden, so wird auf Befehl die komplette Diskette in ca. 10-20 Sekunden in das Ram geladen. Anschließend gilt es beim Laden und Speichern keine Wartezeiten mehr. Nach dem Drücken von RETURN sind Programme oder Daten da.

Beim Abspeichern werden die Daten oder Programme in das Floppy Ram abgelegt und wie dort als intern gelagert auf die Diskette geschrieben. Das bedeutet, daß der Rechner sofort wieder bereit für weitere Arbeiten. Da das Speichern intern erfolgt, ist keine Floppy beim Abspeichern unterbrochen, während, um z. B. ein Programm zu laden. Anschließend fährt die Floppy automatisch mit dem Abspeichern fort.

TurboTrans ist so kompatibel wie das TurboAccess und unterscheidet sich nur in der Floppyplatte vom TA. Das heißt, daß man z. B. die Platte für das 2. Laufwerk von TT bestellen kann, um so das TA aufzuschieben zum TurboTrans. Die TT-Platte kommt dann in die Floppy Nr. 8 und die TA-Platte in die Floppy Nr. 9, die ja ebenfalls nicht so häufig benutzt wird wie die Nr. 8.

TurboTrans wird voraussichtlich ab Oktober lieferbar sein. Bitte bestellen Sie schon jetzt, damit Sie eines von den ersten 1000 Exemplaren bekommen.

Die Post geht ab - der Traum wird wahr mit TURBOACCESS

TurboAccess, das Mail aller Dinge, ist köstlich - nie erreicht

- 10mal schnelleres Laden
- Abspeichern um 50% schneller
- was zeigt sich ein Programm, das nicht mit TurboAccess läuft?
- Drucken kommt am USER-Port + am IEC-Bus betriebsbereit
- komfortable Floppy - Bedienung per Tastendruck, auch für das 2. Laufwerk (Device-Nummer 9)
- Computer-Zeitstrahlen und Tonen sind mit Recht legiert

Mehr im Katalog - sofort anfordern
TurboAccess - LogoAccess 295 DM
Ausrüstung 2. Laufwerk 99 DM
Einbau in den C64 99 DM



| | |
|-------------------------|--------------------|
| TurboAccess | |
| TA 64 | 325 DM |
| TA 128 | 349 DM |
| Adapter TA 64 - TA 128 | 149 DM |
| TA 2. Laufwerk | 99 DM |
| TurboTrans | |
| TT 64 | 499 DM |
| TT 128 | 549 DM |
| Adapter TT 64 - TT 128 | 149 DM |
| TT 2. Laufwerk | 229 DM |
| Symbolischer | |
| Macro-Assembler (Modul) | 199 DM |
| EPROM 64 | 69 DM |
| EPROM 64 | Fertiggerät 199 DM |

| | |
|-----------------------|--------------|
| Leihengeschäft | |
| 5300 Bonn 1, Markt 52 | |
| Fal. (22.00.0502) 12 | |
| Ohrengehörten | |
| Mod-7 | 8,30 - 18,30 |
| SA | 8,30 - 14,00 |
| langer Sa | 8,30 - 17,30 |

Tagspreise für Rams, Eproms, TTL, IC's, Kondensatoren, Widerstände etc.

Das Maß: SCHARFMACHER

Weg mit dem schwammigen Soft

- komfortables, Moskit-Drift: klar wie bei einem 950-Monitor
- einseitige Verankerung des Monitors - kein Farbdrift
- Anschluss an den USER-Port
- V24-Ausgang, mit Normgerät + Normkabel ohne Kompromisse
- läuft mit allen Programmen

| | |
|--------------------------|--------|
| Leertafel ohne Stativ | 49 DM |
| bestückt mit V24 | 99 DM |
| Kompl. mit V24 + Gehäuse | 149 DM |

Der Alleskönner: UNIPROM

Vier Geräte in einem: ein Emulator, Computer für Commodore, MikroTQM kann an allen Rechnern über eine Commodore- oder V24-Schnittstelle betrieben werden. Beim VC20, C64 und C128 erfolgt der Anschluss am USER-Port ohne externe Stromversorgung.

1. Emul-Programmergerät
 - alle Emulationen: E-proms + CMOS und die neuen EPROMS (auch elektrisch lösbar)
 - alle herstellereigenen Formate, bei werden beachtet
 - keine Schalter, UNIPROM wird zur Fehlbedienungsicherheit völlig per Software gesteuert
 - programmiert ein 8 Kbytes E-prom in maximal 9 Sekunden
2. V24-Schnittstelle
 - ohne richtige Schnittstelle ohne Kompromisse
 - entsprechende Buchse und eine entsprechende Pegeländerung
3. Commodore-schnittstelle
 - Umkleintasche, Treiber zur Steuerung der internen CIA
 - natürlich eine Normkabel: Commodore-Buchse
4. Druckpuffer von 16 Kbytes
 - das Warten auf den langsamen Drucker ist vorbei
 - Der Druckpuffer verfügt über Kompressionsalgorithmen, mit denen bis zu 64 Kbytes Text gedruckt werden können.

Nehmen Sie das Super Ding in unserem Katalog. Sofort anfordern!

| | |
|---------------------------|--------|
| Einfrühungspreis komplet. | 349 DM |
| Ohne Gehäuse + Schnitt | |
| maximalisiert mit V24-Ram | 299 DM |
| Leertafel mit Software | 99 DM |

Geschäft für Katalog
Roßmüller GmbH
Finkenweg 1
5309 Mockenheim

ABACUS SOFTWARE-BOUTIQUE

SOFTWARE & BUECHER FUER
C64, C16, VC20, C128,
CPC 464, 664, 6128

ERST SCHAUEN -
DANN KAUFEN!

ESCHERSHEIMER LDSTR. 84
6000 FRANKFURT 1
TEL: 069/59 40 19
MO-FR 11-18.30, SA 10-13

Sieben mal schneller in den Speicher

Beim Kriechtempo der Floppy wird das Laden zur Geduldprobe. Der Auto Fast Loader Generator (AFLG) macht Schluß mit unfreiwilligen Kaffeepausen. Vor jedem Programm wird ein zwei Block langer Fast Loader angebracht. Ergebnis: Das Programm wird siebenmal so schnell geladen. Als kostenlose Dreingabe gibt's den Auto-Start.

Der Start mit RUN führt ins erste Menü: Taste 1 lädt ein Programm von Diskette. Taste 2 bringt das Directory auf den Bildschirm. Die SHIFT-Taste hält die Anzeige an, RUN/STOP bricht sie ganz ab. Drücken von 3 beendet das Programm. Der AFLG holt das alte Programm im Schnellverfahren in den Speicher und zeigt seine Anfangs- und Endadresse an. Wohin soll das modifizierte Programm geladen werden? Das zweite Menü stellt drei Möglichkeiten zur Auswahl: den bisherigen Programmstart auf der Diskette, den normalen Basic-Start, wie er im Zeiger in den Adressen 43/44 festgelegt ist, und eine frei wählbare Adresse. Sie muß in dezimaler Form oder als arithmetischer Ausdruck wie $12 * 4096 + 8 * 256$ eingegeben werden.

Den letzten Schliff gibt man seinem Programm mit dem Auto-

Michael Pall ist das Warten auf die lahme Floppy leid. Sein Schnellader bringt Programme im Blitztempo in den Speicher.

Start. Bei Menüpunkt 1 ist dazu ein Basic-Befehl von maximal neun Zeichen Länge einzugeben, z. B. RUN oder SYS XXXXX. Für Maschinenprogramme bringt Punkt 2 den Start durch Geisterhand. Einfach die Einsprungadresse dezimal oder als Rechenausdruck angeben. Nach dem Laden wird dann direkt mit einem JMP durchgestartet. Wer sein Programm wie üblich per Hand in Gang bringen will, wählt den dritten Menüpunkt. Zum Abspeichern des Programms wird im letzten Menü die Taste 1 gedrückt und der neue Name eingegeben. Nach dem Speichern landet man wieder im gleichen Menü, so daß mehrere Kopien hergestellt werden können. Ein Blick ins Directory läßt sich mit Taste 2 werfen. Mit der Taste 3 wird der Generator verlassen.

Das Programm wird nun mit LOAD "Name",8,1 in Sekundenschnelle geladen. Tritt dabei ein Fehler auf, so wird der Ladevorgang abgebrochen und ein LOAD ERROR ausgegeben. Schon bei Programmen mit drei Blöcken Länge läßt sich der Auto Fast Loader Generator nutzbringend anwenden. Auch mehrteilige Programme können schneller geladen werden. Dazu muß einfach jeder einzelne Teil mit einem Schnellader versehen werden. Natürlich darf dann nur der erste Teil einen Auto-Start erhalten.

Der zwei Blöcke lange Fast Loader wird im Bildschirmspeicher abgelegt. Wer seine Floppy mit einer anderen Gerätenummer als der üblichen Acht betreibt, muß sie durch Poke 2080,X neu einstellen. Die verbesserten Laderoutinen haben noch einen angenehmen Nebeneffekt. Statt der gewohnten häßlichen Knarrgeräusche gibt der Lesekopf jetzt nur noch ein leises Klicken von sich. Die Interruptzyklen für den Diskcontroller wurden nämlich erhöht, um die Lesekopfverstellung zu beschleunigen. Der gleiche Effekt läßt sich von Basic aus durch OPEN1,8,15,"M-W"+CHR\$(6)+CHR\$(28)+CHR\$(1)+CHR\$(15) erzielen.

Michael Pall

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSU |
|------|----|----|----|----|----|----|----|----|-------|
| 0001 | 1A | 08 | FF | FF | 9E | 32 | 30 | 37 | 77 |
| 0009 | 36 | 20 | 41 | 46 | 4C | 47 | 20 | 42 | F4 |
| 0011 | 59 | 20 | 40 | 2E | 50 | 41 | 4C | 4C | 67 |
| 0019 | 00 | 00 | 00 | A2 | FA | 9A | A9 | 08 | 5F |
| 0021 | 85 | BA | 58 | D8 | A2 | 09 | 0E | 20 | 52 |
| 0029 | D0 | 0E | 06 | 02 | A2 | 0E | 0E | 21 | C6 |
| 0031 | D0 | 20 | 44 | E5 | A2 | 77 | BD | 00 | F7 |
| 0039 | 04 | 49 | 80 | 9D | 00 | 04 | CA | 10 | 54 |
| 0041 | F5 | A9 | 08 | 8D | 21 | D0 | A9 | 05 | BD |
| 0049 | A0 | 0D | 20 | 1E | AB | A2 | 00 | A9 | F5 |
| 0051 | 02 | 85 | FE | 20 | D7 | AA | 20 | 3F | 69 |
| 0059 | AB | 38 | 66 | C7 | 20 | 3F | AB | A5 | F9 |
| 0061 | FE | 49 | 33 | 20 | D2 | FF | 20 | 3F | 7C |
| 0069 | AB | 06 | C7 | 20 | 3F | AB | BD | 52 | 4A |
| 0071 | 0C | E8 | 86 | FD | 20 | C2 | 08 | 20 | F5 |
| 0079 | D7 | AA | 20 | D7 | AA | A6 | FD | C6 | B1 |
| 0081 | FE | 10 | D3 | 20 | 3F | AB | 20 | 45 | 0F |
| 0089 | AB | 20 | E4 | FF | C9 | 31 | 90 | F9 | 9E |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| 0091 | C9 | 34 | B0 | F5 | 20 | D2 | FF | 38 | 65 |
| 0099 | E9 | 31 | 0A | 65 | FD | AA | BD | 52 | 1A |
| 00A1 | 0C | 8D | AC | 08 | BD | 53 | 0C | 8D | 0C |
| 00A9 | AD | 08 | 4C | FF | FF | 48 | A5 | BA | 93 |
| 00B1 | 20 | B1 | FF | 68 | 4C | 93 | FF | 48 | 76 |
| 00B9 | A5 | BA | 20 | B4 | FF | 68 | 4C | 96 | A4 |
| 00C1 | FF | 18 | 69 | 76 | 95 | 5F | A9 | 0C | 9C |
| 00C9 | 85 | 60 | A0 | 00 | B1 | 5F | 08 | 29 | F5 |
| 00D1 | 7F | 20 | D2 | FF | C8 | 20 | 10 | F4 | EE |
| 00D9 | 60 | 20 | 44 | E5 | A9 | F0 | 20 | AE | CB |
| 00E1 | 08 | A9 | 24 | 20 | AB | FF | 20 | AE | 7F |
| 00E9 | FF | A9 | 60 | 20 | B8 | 08 | A9 | 00 | EE |
| 00F1 | 95 | 90 | A0 | 02 | 20 | A5 | FF | 85 | 87 |
| 00F9 | C3 | 20 | A5 | FF | 85 | C4 | A5 | 90 | 1F |
| 0901 | D0 | 30 | 88 | 10 | EF | A6 | C3 | A5 | 7D |
| 0909 | C4 | 20 | CD | BD | 20 | 3F | AB | 20 | 1F |
| 0911 | A5 | FF | A6 | 90 | D0 | 1C | C9 | 00 | 38 |
| 0919 | F0 | 11 | 20 | D2 | FF | 20 | E1 | FF | ED |
| 0921 | F0 | 10 | AD | BD | 02 | 29 | 01 | D0 | 0A |
| 0929 | F9 | F0 | E4 | 20 | D7 | AA | A0 | 01 | 98 |

0931 D0 C2 08 20 D7 AA A9 E0 EC
 0939 20 AE 08 20 AE FF 28 F0 42
 0941 05 20 E4 FF F0 FB A5 FD 67
 0949 C9 03 D0 03 4C 23 08 4C 77
 0951 CB 0B 48 20 44 E5 A9 6A 86
 0959 20 C2 08 A9 02 85 BC A9 4F
 0961 00 85 8B A6 D3 A4 B7 F0 86
 0969 0D A8 B1 B8 20 D2 FF C8 E1
 0971 C4 B7 D0 F6 86 D3 A2 00 3C
 0979 20 12 E1 C9 0D F0 06 9D AD
 0981 00 02 E8 D0 F3 86 87 20 19
 0989 CA AA A9 08 8D 11 D0 68 A0
 0991 48 85 89 20 DD F3 68 48 66
 0999 A9 6F 20 88 08 20 A5 FF 30
 09A1 85 5F 20 A5 FF 48 05 5F FE
 09A9 C9 30 D0 0C 68 68 20 A5 D7
 09B1 FF C9 0D 08 F9 4C AB FF CC
 09B9 A9 1B 8D 11 D0 20 D7 AA E5
 09C1 A5 5F 20 D2 FF 68 20 D2 79
 09C9 FF 20 A5 FF 20 D2 FF C9 09
 09D1 0D D0 F6 A9 EF 20 AE 08 FE
 09D9 20 AE FF 20 E4 FF F0 FB 2A
 09E1 68 A8 68 68 98 D0 03 4C 40
 09E9 23 08 4C CB 0B 20 D7 AA B4
 09F1 20 D7 AA 20 60 A5 E8 86 F0
 09F9 7A 20 79 A5 E6 7A A9 02 40
 0A01 85 7B 20 8A AD 4C F7 B7 41
 0A09 A9 00 85 B7 20 53 09 A9 A4
 0A11 EF 20 AE 08 20 AE FF A9 3C
 0A19 F2 20 AE 08 A9 23 20 A8 74
 0A21 FF A9 30 20 A8 FF 20 AE 81
 0A29 FF A9 FF 20 AE 08 A2 06 75
 0A31 8D 30 0C 20 A8 FF CA D0 1C
 0A39 F7 20 AE FF A9 00 8D BC FF
 0A41 0E A9 60 8D 32 0F 8D C9 20
 0A49 0D A9 62 20 AE 08 BD 94 74
 0A51 0E 20 AE FF E8 E0 9F D0 6E
 0A59 F5 20 AE FF A9 02 8D 8C FB
 0A61 0E A9 6F 20 AE 08 A2 00 FC
 0A69 BD 37 0C 20 A8 FF E8 E0 22
 0A71 14 D0 F5 20 AE FF A0 00 68
 0A79 20 A6 0D A9 84 8D C9 8D 92
 0A81 78 A9 32 85 AE A9 0F 85 0C
 0A89 AF A9 FE 85 5F A0 FF 20 1C
 0A91 27 0E 2A B0 2D 20 27 0E BA
 0A99 AD 00 06 D0 0A AC 01 06 3D
 0AA1 C0 02 90 1E 88 84 5F A0 F2
 0AA9 00 89 02 06 91 AE C8 C4 2B
 0AB1 5F D0 F6 98 18 65 AE 85 C9
 0AB9 AE 90 02 E6 AF AD 00 06 37
 0AC1 D0 CB A9 00 20 98 09 A9 96
 0AC9 E2 20 AE 08 20 AE FF 20 88
 0AD1 D7 AA A9 1B 8D 11 D0 AE 57
 0AD9 32 0F AD 33 0F 20 CD BD 98
 0AE1 A9 2D 20 D2 FF AD 32 0F 3F
 0AE9 AC 33 0F 18 65 AE 90 01 86
 0AF1 C8 38 E9 34 AA 98 E9 0F 2F
 0AF9 18 65 AF 20 CD BD 20 E4 8E
 0B01 FF F0 FB A9 EA A0 0C 20 97
 0B09 1E AB A2 02 A0 10 20 8C B7
 0B11 E5 AE 32 0F AD 33 0F 20 C3
 0B19 CD BD A9 29 20 D2 FF A9 34
 0B21 91 20 D2 FF A2 4F BD 00 53
 0B29 02 9D 3C 03 CA 10 F7 A2 AC
 0B31 09 4C 50 08 20 EE 09 4C F5
 0B39 41 0B AC 32 0F AD 33 0F 36
 0B41 8C D3 0D 8D D7 8D A9 A9 83
 0B49 D0 0B A0 2B 8C D3 8D C8 37

0B51 8C D7 8D A9 A5 8D D2 8D F6
 0B59 8D D6 8D A9 34 85 5F A9 81
 0B61 0F 85 68 A9 00 85 5E 8D 8E
 0B69 16 0E A9 FE 18 65 5F 90 36
 0B71 02 E6 68 85 5F E6 5E C5 0B
 0B79 AE A5 60 E5 AF 90 E8 A5 E9
 0B81 AE 69 FD 38 E5 5F 8D F0 2B
 0B89 0D A6 5E CA 0E D8 0D A9 14
 0B91 F8 A0 8C 20 1E AB A2 12 47
 0B99 4C 50 08 20 D7 AA 20 D7 5A
 0BA1 AA 20 60 A5 E8 8D 00 02 6E
 0BA9 F0 08 9D 8A 0E E8 E0 09 F7
 0BB1 D0 F3 A9 0D 9D 8A 0E E8 B2
 0BB9 8E 16 0E A0 A9 A9 F5 D0 0B
 0BC1 03 20 EE 09 8C 41 8D 8D 92
 0BC9 42 0D A2 4F BD 3C 03 9D 6E
 0BD1 00 02 CA 10 F7 20 44 E5 ED
 0BD9 A2 1B 4C 50 08 A9 6F 20 31
 0BE1 AE 08 A2 06 8D 4B 8C 20 9F
 0BE9 A8 FF CA 10 F7 20 AE FF E8
 0BF1 A9 01 20 53 09 A9 60 8D A8
 0BF9 32 0F A9 38 85 AC A9 0D 08
 0C01 85 AD A9 61 20 AE 08 A9 F2
 0C09 36 85 01 20 D1 FC 80 0C 0B
 0C11 A0 00 81 AC 20 A8 FF 20 FB
 0C19 D8 FC D0 EF 20 46 F6 A9 E7
 0C21 37 85 01 A9 01 20 98 09 78
 0C29 A9 1B 8D 11 D0 4C CB 08 4C
 0C31 30 20 32 50 2D 42 4D 2D FD
 0C39 45 05 02 A5 3F 0A AA 85 C7
 0C41 06 85 06 B5 07 85 07 4C 11
 0C49 87 83 07 01 00 69 57 2D CC
 0C51 4D 08 08 11 09 0A DA 08 14
 0C59 74 A4 15 20 2B 3B 08 4B 76
 0C61 0B 35 08 42 4E 5C 9C 0B 22
 0C69 C2 0B CB 08 3A 08 11 DE 0B
 0C71 0B DA 88 74 A4 50 52 47 25
 0C79 20 4C 4F 41 C4 44 49 52 59
 0C81 45 43 54 4F 52 D9 45 4E 78
 0C89 44 C5 50 52 47 2D 53 54 97
 0C91 41 52 54 20 A8 42 41 53 77
 0C99 49 43 2D 53 54 41 52 D4 71
 0CA1 41 44 52 45 53 53 45 20 2F
 0CA9 41 4E 47 45 42 45 CE 50 70
 0CB1 52 47 2D 53 41 56 C5 42 46
 0CB9 41 53 49 43 2D 42 45 46 DE
 0CC1 45 48 CC 4A 4D 50 20 5A 42
 0CC9 55 20 41 44 52 45 53 53 3F
 0CD1 C5 48 45 49 4E 20 41 55 90
 0CD9 54 4F 53 54 41 52 D4 50 77
 0CE1 52 47 2D 4E 41 4D 45 20 83
 0CE9 BA 93 4C 41 44 45 41 44 2E
 0CF1 52 45 53 53 45 3A 00 93 85
 0CF9 41 55 54 4F 53 54 41 52 DF
 0D01 54 3A 8D 00 11 1D 12 41 EC
 0D09 92 55 54 4F 12 46 92 41 5F
 0D11 53 54 12 4C 92 4F 41 44 85
 0D19 45 52 12 47 92 45 4E 45 14
 0D21 52 41 54 4F 52 20 20 2E 40
 0D29 43 29 20 42 59 20 4D 2E 50
 0D31 50 41 4C 4C 0D 0D 00 E4 2B
 0D39 82 20 44 E5 20 93 FC 4C 7F
 0D41 A9 F5 A9 00 85 AE A9 F9 E4
 0D49 8D 28 83 E6 AF 68 A9 8B 6F
 0D51 8D 11 D0 A5 AE C9 14 D0 D0
 0D59 F4 A5 AF C9 85 D8 EE 4C 8C
 0D61 00 84 00 80 22 4C 48 82 68
 0D69 00 31 EA 66 FE 47 FE 4A BC


```

0D71 F3 91 F2 0E F2 50 F2 33 A5
0D79 F3 57 F1 CA F1 ED 02 68 61
0D81 68 20 15 FD 20 53 E4 20 F1
0D89 EF ED A5 BA 20 0C ED A9 45
0D91 6F 20 B9 ED A2 00 8D E2 97
0D99 04 20 DD ED E8 E0 28 D0 34
0DA1 F5 20 FE ED C8 A2 03 8A 84
0DA9 6A 6A 08 0A 2A 0A 28 2A 1C
0DB1 99 03 07 0A 99 04 07 0A 24
0DB9 0A 0A 99 05 07 0A 99 06 23
0DC1 07 98 69 40 A8 CA 10 DF A8
0DC9 84 A2 A9 06 C5 A2 D0 FC 7B
0DD1 78 A9 00 85 AE A9 C0 85 8A
0DD9 AF A9 10 85 90 A0 FF 20 60
0DE1 A7 04 2A 80 29 A0 02 20 89
0DE9 A7 04 A5 90 D0 06 A9 20 89
0DF1 8D 7D 04 98 08 B9 02 06 C9
0DF9 91 AE C8 00 FE D0 F6 98 76
0E01 18 65 AE 85 AE 90 02 E6 17
0E09 AF C6 90 28 D0 CF A9 10 EB
0E11 2A 0A 85 90 A2 00 F0 0B 9B
0E19 86 C6 8D 09 05 9D 76 02 50
0E21 CA D0 F7 4C E4 02 A9 0B 23
0E29 8D 00 DD A2 03 AD 00 DD 41
0E31 10 FB AD 00 DD 8E 00 DD 34
0E39 10 03 24 01 EA EA EA EA 2F
0E41 AE 00 DD 8D 03 07 EA AE 05
0E49 00 DD 1D 02 07 AE 00 DD 8C
0E51 1D 01 07 EA AE 00 DD 1D A7
    
```

```

0E59 00 07 99 00 06 C8 D0 C6 DA
0E61 60 4D 2D 45 05 02 A5 AF F4
0E69 69 02 8D 14 02 8D 19 02 6C
0E71 8D 1E 02 AD 00 FF 85 06 45
0E79 AD 01 FF 85 07 89 60 FF D5
0E81 99 00 03 C8 D0 F7 4C 07 F3
0E89 03 4D 2E 50 41 4C 4C 20 86
0E91 21 00 00 A9 04 85 31 4C DD
0E99 D1 F4 A9 0F 8D 07 1C A9 6F
0EA1 90 8D 98 02 A2 00 A9 E0 F3
0EA9 8D 58 02 85 00 20 99 D5 53
0EB1 C9 02 B0 2C 78 A2 FF 20 FE
0EB9 67 03 A2 02 20 64 03 58 6B
0EC1 AD 00 04 F0 17 C9 24 B0 84
0EC9 16 85 06 20 67 03 68 28 90
0ED1 04 F0 0C 90 0A AD 01 04 ED
0ED9 95 07 80 C3 4C EC DA 18 8D
0EE1 88 48 A2 00 20 56 D5 78 88
0EE9 CA 8A 20 67 03 68 28 90 2D
0EF1 03 4C 0A E6 4C 4D D5 B0 F3
0EF9 00 04 AC 00 18 10 FB A0 0A
0F01 10 8C 00 18 A0 02 85 95 0A
0F09 8C 00 18 4A 4A 4A 8D 17
0F11 00 18 EA 0A 29 0F 8D 00 6E
0F19 18 AD 85 00 29 0F 8D 00 97
0F21 18 EA 0A 29 0F 8D 00 18 7B
0F29 EA A0 0F 8C 00 18 EB D0 CD
0F31 C6 60 8D 0D 0D 0D 0D 66
    
```

```

10 FOR P=0 TO 1:FOR Z=100 TO 215:9=0 2423
20 FOR B=1 TO 16:READ D:IF P THEN 2546
PRINT#1,CHR$(D):
30 S=S+D:NEXT B:READ C:IF C<>S THEN 4042
PRINT "DATA ERROR IN ZEILE"Z:END
40 NEXT Z:IF P=0 THEN INPUT "FILENAME*": 5349
F*:OPEN 1,0,1,F*:PRINT#1,CHR$(1)
CHR$(8):
50 RESTORE :NEXT P:CLOSE 1 655
100 DATA 26,8,255,255,158,50,48,55,54, 6217
32,65,70,76,71,32,66,1321
101 DATA 89,32,77,46,80,65,76,76,0,0,0, 5387
162,250,154,169,8,1284
102 DATA 133,186,88,216,162,9,142,32, 6545
208,142,134,2,162,14,142,33,1805
103 DATA 208,32,68,229,162,119,189,0,4, 6250
73,128,157,0,4,202,16,1591
104 DATA 245,169,8,141,33,208,169,5,160, 6307
13,32,30,171,162,0,169,1715
105 DATA 2,133,254,32,215,170,32,63,171, 6610
56,102,199,32,63,171,165,1860
106 DATA 254,73,51,32,210,255,32,63,171, 7014
6,199,32,63,171,189,82,1883
107 DATA 12,232,134,253,32,194,8,32,215, 6015
170,32,215,170,166,253,188,2316
108 DATA 254,16,211,32,63,171,32,69,171, 6127
32,228,255,201,49,144,248,2177
109 DATA 201,52,176,245,32,210,255,56, 6216
233,49,10,101,253,170,189,82,2314
110 DATA 12,141,172,8,189,83,12,141,173, 6804
8,76,255,255,72,165,186,1948
111 DATA 32,177,255,104,76,147,255,72, 6911
165,186,32,180,255,104,76,150,2266
112 DATA 255,24,105,110,133,95,169,12, 6250
133,96,160,0,177,95,8,41,1621
113 DATA 127,32,210,255,200,40,16,244, 6110
96,32,68,229,169,240,32,174,2164
114 DATA 8,169,36,32,168,255,32,174,255, 6266
169,96,32,184,8,169,0,1787
115 DATA 133,144,160,2,32,165,255,133, 7964
195,32,165,255,133,196,165,144,2309
    
```

```

116 DATA 208,48,136,16,239,166,195,165, 6922
196,32,205,169,32,63,171,32,2093
117 DATA 165,255,166,144,208,28,201,0, 6759
240,17,32,210,255,32,225,255,2433
118 DATA 240,16,173,141,2,41,1,208,249, 5917
240,228,32,215,170,160,1,2117
119 DATA 208,194,8,32,215,170,169,224, 6865
32,174,8,32,174,255,40,240,2175
120 DATA 5,32,228,255,240,251,165,253, 5885
201,3,208,3,76,35,8,76,2039
121 DATA 203,11,72,32,68,229,169,106,32, 6745
194,8,169,2,133,188,169,1785
122 DATA 0,133,187,166,211,164,183,240, 7104
13,168,177,187,32,210,255,200,2526
123 DATA 196,183,208,246,134,211,162,0, 7262
32,18,225,201,13,240,6,157,2232
124 DATA 0,2,232,208,243,134,183,32,202, 6631
170,169,11,141,17,208,104,2056
125 DATA 72,133,185,32,221,243,104,72, 6365
169,111,32,184,8,32,165,255,2018
126 DATA 133,95,32,165,255,72,5,95,201, 5930
48,208,12,104,104,32,165,1726
127 DATA 255,201,13,208,249,76,171,255, 6461
169,27,141,17,208,32,215,170,2407
128 DATA 165,95,32,210,255,104,32,210, 6574
255,32,165,255,32,210,255,201,2508
129 DATA 13,208,246,169,239,32,174,8,32, 5903
174,255,32,228,255,240,251,2556
130 DATA 104,168,104,104,152,208,3,76, 6805
35,8,76,203,11,32,215,170,1669
131 DATA 32,215,170,32,96,165,232,134, 6944
122,32,121,165,230,122,169,2,2039
132 DATA 133,123,32,138,173,76,247,183, 7104
169,0,133,183,32,83,9,169,1883
133 DATA 239,32,174,8,32,174,255,169, 7032
242,32,174,8,169,95,32,168,1943
134 DATA 255,169,48,32,168,255,32,174, 7132
255,169,255,32,174,8,162,6,2194
135 DATA 169,48,12,32,168,255,202,208, 6903
247,32,174,255,169,0,141,189,2320
136 DATA 14,169,96,141,50,15,141,201,13, 6220
    
```

Floppy

| | | |
|---|--|---|
| 169,99,32,174,8,189,148,1658 | | 176 DATA 69,72,204,74,77,80,32,90,85,32, 5888 |
| 137 DATA 14,32,168,255,232,224,159,208, 6092 | | 65,68,82,69,83,83,1265 |
| 245,32,174,255,169,2,141,188,2498 | | 177 DATA 197,75,69,73,78,32,65,85,84,79, 6488 |
| 138 DATA 14,169,111,32,174,8,162,0,189, 6146 | | 83,84,65,82,212,80,1443 |
| 55,12,32,168,255,232,224,1837 | | 178 DATA 82,71,45,78,65,77,69,32,186, 6483 |
| 139 DATA 20,208,245,32,174,255,160,0,32, 6016 | | 147,76,65,68,69,85,68,1263 |
| 166,13,169,132,141,201,13,1961 | | 179 DATA 82,69,83,83,69,58,0,147,65,85, 5758 |
| 140 DATA 120,169,50,133,174,169,15,133, 6921 | | 84,79,83,84,65,82,1218 |
| 175,169,254,133,95,160,255,32,2236 | | 180 DATA 84,58,13,0,17,29,18,65,146,85, 5686 |
| 141 DATA 39,14,42,176,45,32,39,14,173,0, 5276 | | 84,79,18,70,146,65,977 |
| 6,208,10,172,1,6,977 | | 181 DATA 83,84,18,76,146,79,65,68,69,82, 5948 |
| 142 DATA 192,2,144,30,136,132,95,160,0, 6776 | | 18,71,146,69,78,69,1221 |
| 185,2,6,145,174,200,196,1799 | | 182 DATA 82,65,84,79,82,32,32,46,67,41, 5366 |
| 143 DATA 95,208,246,152,24,101,174,133, 6038 | | 32,66,89,32,77,46,946 |
| 174,144,2,230,175,173,0,6,2037 | | 183 DATA 80,65,76,76,13,13,0,228,2,32, 6106 |
| 144 DATA 208,203,169,0,32,152,9,169,226, 6095 | | 68,229,32,147,252,76,1389 |
| 32,174,8,32,174,255,32,1875 | | 184 DATA 169,245,169,0,133,174,169,249, 6743 |
| 145 DATA 215,170,169,27,141,17,208,174, 7050 | | 141,40,3,230,175,96,169,11,2173 |
| 50,15,173,51,15,32,205,189,1851 | | 185 DATA 141,17,208,165,174,201,20,208, 6218 |
| 146 DATA 169,45,32,210,255,173,50,15, 6228 | | 244,165,175,201,5,208,238,76,2446 |
| 172,51,15,24,101,174,144,1,1631 | | 186 DATA 0,4,0,0,34,76,72,178,0,49,234, 5684 |
| 147 DATA 200,56,233,52,170,152,233,15, 6363 | | 182,254,71,254,74,1402 |
| 24,101,175,32,205,189,32,228,2097 | | 187 DATA 243,145,242,14,242,80,242,51, 6792 |
| 148 DATA 255,240,251,169,234,160,12,32, 6432 | | 243,87,241,202,241,237,2,184,2616 |
| 30,171,162,2,160,16,32,12,1938 | | 188 DATA 104,32,21,253,32,83,228,32,239, 6513 |
| 149 DATA 229,174,50,15,173,51,15,32,205, 6431 | | 237,185,186,32,12,237,169,2062 |
| 189,169,41,32,210,255,169,2009 | | 189 DATA 111,32,185,237,162,0,189,226,4, 5886 |
| 150 DATA 145,32,210,255,162,79,189,0,2, 6208 | | 32,221,237,232,224,40,208,2340 |
| 157,60,3,202,16,247,162,1921 | | 190 DATA 245,32,254,237,200,162,3,138, 6147 |
| 151 DATA 9,76,80,8,32,238,9,76,65,11, 5779 | | 106,106,8,10,42,10,40,42,1635 |
| 172,50,15,173,51,15,1080 | | 191 DATA 153,3,7,10,153,4,7,10,10,10, 5209 |
| 152 DATA 140,211,13,141,215,13,169,169, 6285 | | 153,5,7,10,153,6,701 |
| 208,11,160,43,140,211,13,200,2057 | | 192 DATA 7,152,105,64,168,202,16,223, 6596 |
| 153 DATA 140,215,13,169,165,141,210,13, 6395 | | 132,162,169,8,197,162,208,252,2225 |
| 141,214,13,169,52,133,95,169,2052 | | 193 DATA 120,169,0,133,174,169,192,133, 6947 |
| 154 DATA 15,133,96,169,0,133,94,141,22, 6166 | | 175,169,16,133,144,160,255,32,2174 |
| 14,169,254,24,101,95,144,1604 | | 194 DATA 167,4,42,176,41,160,2,32,167,4, 6088 |
| 155 DATA 2,230,96,133,95,230,94,197,174, 6575 | | 165,144,208,6,169,32,1519 |
| 165,96,229,175,144,235,165,2460 | | 195 DATA 141,125,4,152,8,185,2,6,145, 6664 |
| 156 DATA 174,105,253,56,229,95,141,240, 7143 | | 174,200,192,254,208,246,152,2194 |
| 13,166,94,202,142,219,13,169,2311 | | 196 DATA 24,101,174,133,174,144,2,230, 6603 |
| 157 DATA 248,160,12,32,30,171,162,18,76, 6536 | | 175,198,144,40,208,207,169,16,2139 |
| 80,8,32,215,170,32,215,1661 | | 197 DATA 42,18,133,144,162,0,240,11,134, 5400 |
| 158 DATA 170,32,96,165,232,189,0,2,240, 5937 | | 198,189,9,5,157,118,2,1554 |
| 8,157,138,14,232,224,9,1988 | | 198 DATA 202,208,247,76,228,2,169,11, 6317 |
| 159 DATA 208,243,169,13,157,138,14,232, 7271 | | 141,0,221,162,3,173,0,221,2064 |
| 142,22,14,160,169,169,245,208,2303 | | 199 DATA 16,251,173,0,221,142,0,221,16, 6026 |
| 160 DATA 3,32,238,9,140,65,13,141,66,13, 5904 | | 3,36,1,234,234,234,234,2016 |
| 162,79,189,60,3,157,1378 | | 200 DATA 174,0,221,189,3,7,234,174,0, 5904 |
| 161 DATA 0,2,202,16,247,32,68,229,162, 6165 | | 221,29,2,7,174,0,221,1656 |
| 27,76,80,8,169,111,32,1461 | | 201 DATA 29,1,7,234,174,0,221,29,0,7, 5870 |
| 162 DATA 174,8,162,6,189,75,12,32,168, 6691 | | 153,0,6,200,208,198,1467 |
| 255,202,16,247,32,174,255,2007 | | 202 DATA 96,77,45,69,5,2,165,175,105,2, 5948 |
| 163 DATA 169,1,32,83,9,169,96,141,50,15, 6313 | | 141,20,2,141,25,2,1072 |
| 169,56,133,172,169,13,1477 | | 203 DATA 141,30,2,173,0,255,133,6,173,1, 5773 |
| 164 DATA 133,173,169,97,32,174,8,169,54, 7092 | | 255,133,7,185,96,255,1845 |
| 133,1,32,209,252,176,12,1824 | | 204 DATA 153,0,3,200,208,247,76,7,3,77, 6302 |
| 165 DATA 160,0,177,172,32,168,255,32, 7265 | | 46,80,65,76,76,32,1349 |
| 219,252,208,239,32,70,246,169,2431 | | 205 DATA 33,0,0,169,4,133,49,76,209,244, 5741 |
| 166 DATA 55,133,1,169,1,32,152,9,169,27, 6056 | | 169,15,141,7,28,169,1446 |
| 141,17,208,76,203,11,1404 | | 206 DATA 128,141,152,2,162,0,169,224, 6516 |
| 167 DATA 48,32,50,80,45,66,77,45,69,5,2, 5618 | | 141,91,2,133,0,32,153,213,1743 |
| 165,63,10,170,181,1108 | | 207 DATA 201,2,176,44,120,162,255,32, 5694 |
| 168 DATA 6,133,6,181,7,133,7,76,7,3,7,1, 5177 | | 103,3,162,2,32,100,3,88,1485 |
| 0,105,87,45,804 | | 208 DATA 173,0,4,240,23,201,36,176,22, 6031 |
| 169 DATA 77,0,8,17,9,10,218,8,116,164, 5666 | | 133,6,32,75,242,205,1,1569 |
| 21,32,43,59,11,75,868 | | 209 DATA 4,240,12,144,10,173,1,4,133,7, 6514 |
| 170 DATA 11,53,11,66,78,92,156,11,194, 6121 | | 176,195,76,236,218,24,1653 |
| 11,203,11,58,0,17,222,1202 | | 210 DATA 8,72,162,0,32,86,2,133,120,202, 5573 |
| 171 DATA 11,218,8,116,164,80,82,71,32, 6126 | | 138,32,103,3,104,40,144,1459 |
| 76,79,65,196,68,73,82,1421 | | 211 DATA 3,76,10,230,76,77,213,189,0,4, 5875 |
| 172 DATA 69,67,84,79,82,217,69,78,68, 6179 | | 172,0,24,16,251,160,1501 |
| 197,80,82,71,45,83,84,1455 | | 212 DATA 16,140,0,24,160,2,133,133,140, 5517 |
| 173 DATA 65,82,84,32,168,66,65,83,73,67, 5847 | | 0,24,74,74,74,74,141,1209 |
| 45,83,84,65,82,212,1356 | | 213 DATA 0,24,234,10,41,15,141,0,24,173, 5519 |
| 174 DATA 65,68,82,69,83,83,69,32,65,78, 6008 | | 133,0,41,15,141,0,992 |
| 71,69,66,69,206,80,1255 | | 214 DATA 24,234,10,41,15,141,0,24,234, 5924 |
| 175 DATA 82,71,32,83,65,86,197,66,65,83, 5921 | | 160,15,140,0,24,232,208,1502 |
| 73,67,45,66,69,70,1220 | | 215 DATA 198,96,0,0,0,0,0,0,0,0,0,0,0, 3343 |



Auszug aus unserem Lieferprogramm
(Orig. Commodore/Philips/usw. Neugeräte mit 6 Mon. Garantie)

| | | | |
|---------------------|--------|----------------------|-------|
| SFD 1001 | 795,— | Monitor Grün/Ton | 298,— |
| SFD 1002 | 935,— | C 64 | 598,— |
| CBM 710 | 1295,— | VC 1541 | 635,— |
| CBM 720 | 1795,— | VC 1530 | 99,— |
| CBM 8050 | 1800,— | Druckerständer | 58,— |
| CBM 9090 | 3990,— | Papier weiß 300BL | 9,95 |
| PC 10 | 5643,— | Disketten 1D 105T | 39,— |
| PC 20 | 9063,— | Farbdrucker zBVC1525 | 21,— |
| Erweiterung 512 KB | 495,— | Userportstecker | 9,86 |
| 640 KB | 795,— | Reinigungsdisk | 27,50 |
| Backuptape für PC20 | 2890,— | Tastaturabdeck C64 | 14,95 |

Ersatzteilversand per Nachnahme
(Original Commodore) z. B.: IC's

| | | | |
|----------------|-------|------|--------|
| 8501/7501 | 54,38 | 6502 | 37,48 |
| 325303-01 | 47,71 | 6504 | 33,65 |
| 325341-08 | 70,85 | 6510 | 43,75 |
| 325572-01 | 44,18 | 6520 | 34,88 |
| 901225-01 | 28,80 | 6522 | 36,66 |
| 901226-01 | 34,88 | 6526 | 43,64 |
| 901227-03 | 34,88 | 6532 | 48,84 |
| 901229-05 | 34,88 | 6561 | 52,45 |
| 808718 Mhz | 590,— | 6569 | 139,54 |
| VC20 Modulator | 70,48 | 6581 | 69,77 |
| Netzteil C64 | 91,— | 8701 | 18,47 |

Preise in DM inkl. MwSt. zzgl. Nachnahme + Verpackung DM 6,50. Ab DM 800,— Porto u. Verpackung frei.

HDS Prulftechnik GmbH, Maria-Eich-Str. 1, Telefon 089/83 70 21-22, Mailbox 089/83 70 23



Mailbox 64:

Ihr Weg, um Telefonkosten zu sparen, aber nicht am Spaß

Software inkl. Automodem DM 229,—
nur das Automodem DM 159,—
nur die Mailboxsoftware DM 129,—

Auf der Disk sind: Mailboxprg. 38 K, Editor 8 K, Terminalprg. 4 K, Wählautomat 8 K, Mailbox-Files 160 K.

Zum Betrieb werden benötigt: C 64/128 mit einer oder zwei Floppies 1541 oder ein IEC-Bus-Laufwerk, z. B. SFD 1001. Ein Automodemvorsatz und ein Akustikkoppler oder Modem. Ein Betrieb über Postmodem ist möglich. Das Automodem bietet Autoanswer, Autowahl, RS232 integriert, Resetaster, Schalter Automodem ein/aus, LEDs für Carrier, Ruf, Ltg. usw. Die Mailbox ist Btx-ähnlich seitenorientiert und hat z. B. folgenden Aufbau:

10 Inhalt 20 Usergroup 30 Pinnwand
31 persönliche Mail
40 Datex 50 Mailboxnr. 55 Sysopbox
60 Tips und Tricks
70 Fundg. 80 Datenbank 99 beenden
co consolecom

Btx-Decoder für C 64:

Fahren Sie Btx ohne Btx-Fernseher auf Ihrem C 64/128.

Das wird benötigt: C 64 mit Btx-Modul und normaler Fernseher ohne Decoder, wenn Sie Seiten drucken oder speichern wollen, brauchen Sie noch eine Floppy und einen normalen Drucker.

Preis nur DM 698,—

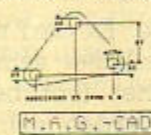
Wir liefern alles für die DFÜ

Rudolf Möllenbeck — Ihr Spezialist für DFÜ — Alte Linnerstr. 24, 4150 Krefeld 1, Tel. 0 21 51/2 01 30, Mailbox /801 339, Btx 920 163 025

MAG GRAPHISOFTWARE COMMODORE 64 MAG

3D-CONSTRUCTION-SET

Wenn Sie in die Welt der 3D-Grafik einsteigen, so ist dies genau das Richtige. Sie müssen nur Ihren Körper eingeben, dann stehen Ihnen folgende Möglichkeiten zur Verfügung: Abspeichern, Einladen, Zerren, Stauchen, Vergrößern, Verkleinern, Verschieben, Drehen, Sichtwinkel ändern, Hardcopies erstellen.

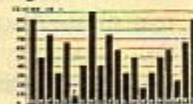
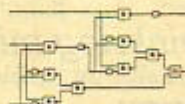


M.A.G.-CAD

Ein CAD-System für den VC-64. Mit Hilfe eines Joysticks und 29 neuen Befehlen können Sie Ihre Grafiken und technischen Zeichnungen auf dem Bildschirm erstellen und ausdrucken oder abspeichern.

M.A.G.-GRAFIK

Es stehen Ihnen 15 Grafikbefehle im Basic zur Verfügung. Spritessteuerung, Text-Grafik Mischung und ein kinderleichtes Programmieren von Grafiken wird Ihnen ermöglicht.



STATISTIK-PROFI

Alle Ihre Daten können Sie nun statistisch auswerten. Sie können mit 5 Datensätzen gleichzeitig arbeiten und auf verschiedene Arten grafisch auswerten: Balkendiagramm, Strichdiagramm und 3D-Kuchengrafik. Hardcopy dieser Grafiken ist möglich.



ALLE 4 PROGRAMME incl. Diskette, Porto, Verp. und deutschen Anleitungen nur **79,-**

M.A.G.-SOFTWARE, Schwarzwaldring 49
7505 Ettlingen 4, Tel.: 07243/28406

Computerprognosen Kabbalistik Programm

von Parapsychologie und Kabbalist erstellt und kinderleicht zu bedienen.

INHALT:

Persönlichkeitsanalyse: Charakter, Inneres Wesen, Ihr Intellekt, Ihre Persönlichkeit.

Monats- u. Jahreshoroskop: beliebiges Datum, Schicksalsweg, Lebensweg und Lebenshoroskop.

Biorhythmus jeder Tag in % und Monatskurve Glückssteine, Glückszahlen, Metalle und Farben Planetarische Ausstrahlung Ihrer Person und Planetenbeherrscher

Ihre Krankheitsdisposition

Für Schwangere: Junge oder Mädchen

Partnerwahl

mit EXPERTENTEIL

volle Diskette nur 260,00 DM. Nur per Nachnahme.

Für Apple nur auf Bestellung. Wir suchen noch einige Wiederverkäufer

Bestellung und Information

Chr. WEDEWER

Tel.: 0 29 41/2 12 62

Stirper Str. 101,

4780 LIPPSTADT



9" Monitor. Monochrom grün-entspiegelt. Auflösung über 20 MHz. BAS/Video-Eingang. Eigenes Netzteil 220 V. Passend für alle Apple u. Commodore und ähnliche.
Best.-Nr. 65032 NUR DM 199,90



Kostenlosen Katalog anfordern



Schmeißen Sie ihn weg! Ihren alten Commodore/ATARI-Joystick. Kaufen Sie bei uns den neuen Trickball. Gesteuert durch den Handballen. Außerst sensibel. Dazu ein Superpreis. Lieferung paarweise kpl. mit Anschlußkabel für alle Commodore und ATARI-Computer. Unser Paarpreis für your Joy. Best.-Nr. 94031 das Paar nur DM 24,50
ab 3 Paar Best.-Nr. 94032 nur DM 19,90

Bühler Computer, Postfach 32, 7570 Baden-Baden, Tel. (07221) 71004
Bühler Shop, Waldstraße 46, 7500 Karlsruhe

DISKORG gegen Diskettenchaos

Die Floppy 1541 ist ein intelligentes Laufwerk. Sie ist mit einem eigenen Prozessor ausgestattet. Ihr Betriebssystem ist doppelt so groß wie das des C 64.

Entsprechend groß ist auch die „Intelligenz“, mit der das Laufwerk seinen Besitzer zu ärgern versteht. Das Spektrum der Bosheiten reicht von verschwundenen Programmen bis zu vermurksten Directories.

Im Kleinkrieg mit der Floppy stellt DISKORG den idealen Bündnispartner dar. Durch einfache Menüwahl werden alle Floppyfunktionen zur Verfügung gestellt. Zusätzlich gibt es einige Extras, um Diskettenfehler zu beheben oder von vornherein zu verhindern.

Kleinkrieg mit der Floppy

Nach dem Start des Programms erscheint das Titeldisplay mit dem Menü:

Gerätenummer der Floppy ändern

New = Formatieren

Check Disk

Ändern Diskname/ID

Formatkennzeichen ändern

Schreibschutzbits ändern

@ Fehlerkanal der Floppy lesen

Directory mit Scrolling

Erweitertes Directory

Ordnen Directory (alphabetisch)

Initialisieren

Validieren

Kopieren

Umbenennen

Löschen

Programmblockcheck (Kettung prüfen)

Zurückholen gelöschter Files

Quit = Ende

Gerätenummer der Floppy ändern (Zeile 540)

Dieser Programmteil ist eine überarbeitete und verkürzte Version des Programms DISK ADDR CHANGE der Test/Demo-Diskette.

Beim Arbeiten mit der Floppy gibt's häufig Pannen. Mit DISKORG im Werkzeugkasten braucht man sich vor Diskettenproblemen nicht zu fürchten.

te. Es darf nur die Floppy angeschaltet sein, deren Gerätenummer (vorübergehend!) geändert werden soll. Als Gerätenummern sind die Zahlen 8 bis 15 zulässig.

New = Diskette neu formatieren (Zeile 670)

Formatiert eine Diskette wahlweise mit oder ohne ID. Das Formatieren ohne Angabe einer ID löscht das Directory. Es funktioniert nur bei vorher schon einmal formatierten Disketten.

Achtung: Wurde die ID per Programm geändert, so erscheint beim Neuformatieren ohne ID wieder die alte ID der ersten Formatierung!

Defekte Diskette

Im Gegensatz zum Programm PERFORMANCE TEST der Test/Demo-Diskette werden zwecks Zeitersparnis KEINE Schreib/Lesetests durchgeführt. Bei Verdacht auf defekte Sektoren kann die folgende Funktion eingesetzt werden:

Check Disk = Überprüfung der Diskette auf defekte Blöcke (Zeile 1260)

Dieser Programmteil entspricht von der Funktion her dem Programm CHECK DISK der Test/Demo-Diskette, ist aber wesentlich schneller. Es stehen zur Auswahl:

a) Leseprüfung (dauert etwa fünf Minuten)

b) Schreib/Leseprüfung (Schreiben und Verifizieren (dauert 10 Minuten))

Achtung: Bei der Schreib/Leseprüfung wird die Diskette gelöscht!

Ändern von Diskname/ID (Zeile 850)

Nach Ausgabe des bisherigen Eintrags können Diskettenname (16 Stellen) und/oder ID und Formatkennzeichen (fünf Stellen) im Block 18/0 geändert werden: Die entsprechenden Bytes in diesem Block sind:

Byte 144-161 = Diskettenname

162/163 = ID

164 = unbenutzt

165/166 = DOS- und Formatkennzeichen (normalerweise 2A)

Wird nur RETURN eingegeben, so bleibt der gegenwärtige Eintrag für Name beziehungsweise ID unverändert.

Schreibschutz

Formatkennzeichen ändern (Zeile 1670)

Hier läßt sich ein globaler Schreibschutz erzeugen. Bis auf das Neuformatieren mit ID sind keinerlei Schreibzugriffe mehr möglich. Der Schreibschutz kann unter demselben Menüpunkt auch wieder entfernt werden.

Zum Setzen oder Löschen wird in das Byte 2 in Block 18/0 der Buchstabe X beziehungsweise A (normaler Wert bei der Floppy VC1541) eingetragen.

Jeder Versuch, auf eine so geschützte Diskette zu schreiben, wird mit der Fehlermeldung „73 CBM DOS V2.6 1541“ abgewiesen.

Schreibschutzbits ändern (Zeile 2420)

Setzt/löscht den Schreibschutz für einzelne Files durch Setzen/Löschen von Bit 6 der Filetypkennung in Byte 0 des jeweiligen Directoryeintrags. Die so geschütz-

ten Files sind bei Ausgabe des Directory durch ein Kleiner-Zeichen hinter dem Filetyp markiert, also zum Beispiel:

12 "TEST" PRG <

ACHTUNG! Dieser Schreibschutz schützt zwar vor dem einfachen Scratch-Befehl (S:Name), aber nicht vor dem Überschreiben mit dem Klammeraffen (SAVE"@:Name")!

@ Fehlerkanal der Floppy lesen (Zeile 510)

Liest den Floppy-Fehlerkanal und gibt gegebenenfalls die Fehlermeldung auf den Bildschirm aus.

Directory ausgeben mit Scrollen (Zeile 3410)

Gibt das Directory (wahlweise unsortiert oder alphabetisch sortiert) auf Bildschirm oder Drucker aus:

| DISKNAME | ID |
|----------|---------------|
| FILENAME | Länge * Typ < |
| | |

BELEGT NNN ,FREI NNN V.664

Nach der vollständigen Ausgabe kann der Text durch die CRSR-UP/DOWN-Tasten zurück oder vorwärts gescrollt werden.

Sollten sich am Anfang des Directory unsinnige Zeilen ergeben, so liegt dies an einem manipulierten Block 18/0 (Fortsetzadresse ungleich 18/1, siehe dazu bei Ordnen des Directory).

Erweitertes Directory (Zeile 4160)

Auf Bildschirm/Drucker werden zusätzlich zur normalen Directoryausgabe weitere Angaben zu den Files gemacht:

- Ladeadresse bei PRG/USR-Files
- Track und Sektor des 1. Datenblocks

Weicht die Startadresse eines Programmfiles vom Basic-Start (2049) ab, so wird sie invers dargestellt.

Bei relativen Files erfolgt die Ausgabe aller Fileparameter:

- Anzahl der Datenblöcke
- Track und Sektor des ersten Datenblocks
- Anzahl der Side-Sector-Blöcke
- Track und Sektor des ersten Side-Sector-Blocks
- Anzahl der Records (Datensätze)
- Recordlänge in Zeichen

Weiterhin erfolgt ein Vergleich

der belegten Blöcke nach den Angaben in der BAM (Block Availability Map) und nach der Summe der belegten Blöcke in der Directory. Bei Abweichungen wird eine Warnung ausgegeben (siehe auch Funktionen Validate und Programmblockcheck).

Ordnen des Directory (Zeile 3200) Liest alle Directoryeinträge in den Rechenpeicher, ordnet sie in alphabetischer Reihenfolge und schreibt sie auf die Diskette zurück.

Normalerweise liegt das Directory auf Track 18, Sektor 0,1,4,7,10,13,16,2,5,8,11,14,17,3,6,9,12,15 und 18 in dieser Reihenfolge. Bei kopiergeschützten Disketten zum Beispiel ist diese normale Fileverkettung manchmal manipuliert. Insbesondere wird oft in 18/0 eine Folgeblocknummer ungleich 18/1 eingetragen. Bei solchen Disketten ist die Sortierfunktion nicht anwendbar. Das Programm bricht mit einer Warnung ab.

Initialisieren (Zeile 510)

Initialisiert die Floppy. Die BAM der eingelegten Diskette wird in den Speicher des Laufwerks eingelesen.

Nach Ausführung der normalen Validate-Funktion V steht die veränderte BAM noch nicht im Speicher des Laufwerks, das heißt, die Floppy hat noch die alte BAM geladen. Vor dem Weiterarbeiten sollte initialisiert werden.

Validieren (Zeile 1050)

Prüft entsprechend der Verkettung (Aufeinanderfolge der Blocks) der einzelnen Files im Directory die Belegungsbits in der BAM und korrigiert diese, falls nötig.

Achtung: Validate bringt keine Hilfe bei fehlerhafter Verkettung von Files, zum Beispiel bei einer Schleife in der Verkettung oder wenn mehrere Programme den gleichen Anfangssektor beanspruchen (siehe Funktion Programmblockcheck).

Kopieren von Files (Zeile 1090)

Kopiert Files nach Eingabe des alten und des neuen Namens.

Löschen von Files (Zeile 1200)

Löscht Files, auch mit Jokerzeichen.

Umbenennen von Files (Zeile 1170)

Ändert den Namen von Files nach Eingabe des neuen und des alten Namens.

Programmblockcheck = Kettung prüfen (Zeile 1860)

Gibt die Verkettung (Reihenfolge der einzelnen Blocks) von Files auf den Bildschirm aus mit den Optionen:

E = einzelne Files darstellen

B = Block suchen. Es wird nach dem File gesucht, der einen bestimmten Block enthält.

A = Alle Files darstellen

Z = Zurück zum Menü

Fehler in der Programmverkettung können durch fehlendes Initialisieren und wahrscheinlich durch einen versteckten Fehler im DOS beim Befehl SAVE "@:..." entstehen. Um sie zu beseitigen, braucht man einen Diskmonitor und viel Geduld.

Bei Doppeleinträgen im Directory sind die Files meist noch beide vorhanden.

Zurückholen gelöschter Files (Zeile 2590)

Liest im Directory die als gelöscht markierten Files (DEL-Files) und fragt für jedes, ob es wiederhergestellt werden soll. Wenn ja, ist noch der Filetyp anzugeben. Das File wird dann rekonstruiert. Die Nummern der belegten Blöcke werden auf den Bildschirm ausgegeben.

Quit = Ende

beendet das Programm. Es wird ins Basic zurückgekehrt. Mit RUN wird neu gestartet.

Das Programm verwendet drei Maschinenroutinen, die beim Start in den Speicher gepoked werden. Die erste liegt im Bereich von \$0334 bis \$03FF. Sie liest das Directory ein.

Die beiden anderen Routinen stehen im Speicherbereich \$C000 bis \$C0CB. Sie scrollen den Bildschirm- und Farbspeicher nach unten (für Cursor Down bei der Directory-Ausgabe) und sortieren ein Stringarray.

Das Programm ist in allen Funktionen mit DOS.HYPRALOAD und EXBASIC-Level II verträglich.

(Andreas Soik) ▶

Floppy

| | | | |
|--|------|--|------|
| 10 POKE 53280,0:POKE 53281,0:POKE 686,1:POKE 53272,21:G%=PEEK (186) | 4464 | 700 INPUT "[DOWN]MIT/OHNE ID(M/D/Z)[2SPACES]M [3LEFT]";Z* | 3282 |
| 20 GOSUB 1640:GOSUB 1578 | 1296 | 710 IF Z*(">"M" AND Z*(">"O" AND Z*(">"Z" THEN 700 | 3318 |
| 30 IF PEEK (828)=174 AND PEEK (49160)=133 THEN 100 | 2986 | 720 IF Z*="Z" THEN RUN | 1206 |
| 40 GOSUB 4800 | 601 | 730 INPUT "[2DOWN]DISKNAME :[2SPACES,CBM F, 3LEFT]";X* | 2068 |
| 100 DIM A*(147),S(20),T(20):A*(2)="*****" | 6679 | 740 IF LEN (X*)>16 THEN PRINT "[DOWN]MAX. 16 ZEICHEN!":GOTO 730 | 3477 |
| 110 PRINT "[RVS]G[OFF] ERAETENUMMER DER FLOPPY AENDERN" | 3588 | 750 IF Z*(">"M" THEN 700 | 1445 |
| 120 PRINT "[RVS]N[OFF] EW-FORMATIEREN", "0N" | 2398 | 760 INPUT "[DOWN,6SPACES]ID :[2SPACES,CBM F, 3LEFT]";Y* | 1474 |
| 130 PRINT "[RVS]C[OFF] HECK DISK" | 1404 | 770 IF LEN (Y*)>5 THEN PRINT "[DOWN]MAX. 5 ZEICHEN!":GOTO 760 | 3499 |
| 140 PRINT "[RVS]A[OFF] ENDERN[2SPACES]DISK-NAME/ID" | 2341 | 780 IF X*="[CBM F]" OR Y*="[CBM F]" THEN RUN | 1922 |
| 150 PRINT "[RVS]F[OFF] ORMATKENNZEICHEN [2SPACES]AENDERN" | 2714 | 790 X*="N":X* | 1059 |
| 160 PRINT "[RVS]S[OFF] CHREIBSCHUTZBITS [2SPACES]AENDERN" | 2518 | 800 IF Z*(">"M" THEN 1060 | 1452 |
| 170 PRINT "[RVS]0[OFF] FEHLERKANAL AUSLESEN" | 2681 | 810 PRINT "[DOWN]< BITTE CA. 80 SEC WARTEN! >" | 2568 |
| 180 PRINT "[RVS]D[OFF] IRECTORY M.SCROLL 0#" | 2214 | 820 OPEN 1,G%,15:PRINT#1,X*+",";Y*:GOTO 3740 | 3011 |
| 190 PRINT "[RVS]E[OFF] RWEIFTERES DIRECTORY" | 2585 | 850 GOSUB 1640:PRINT "[RVS]AENDERN VON DISK-NAME/ID":GOSUB 3050:F=0 | 4425 |
| 200 PRINT "[RVS]P[OFF] RDNEN DIRECTORY (ALPHABETISCH)" | 3125 | 860 OPEN 1,G%,15:OPEN 2,G%,2,"#":GOSUB 3830 | 2658 |
| 210 PRINT "[RVS]I[OFF] NITIALISIEREN", "0I" | 2324 | 870 X=144:GOSUB 3580:X*="":FOR U=1 TO 16:GET #2,Y*:X*=X*+Y*:NEXT | 4453 |
| 220 PRINT "[RVS]V[OFF] ALIDIERN", "0V" | 1616 | 880 PRINT "[2DOWN]ALTER DISK-NAME=[3SPACES, RVS]X* | 2124 |
| 230 PRINT "[RVS]K[OFF] OPIEREN", "0K" | 1526 | 890 PRINT "NEUER DISK-NAME=[3SPACES]X*[UP]" | 2410 |
| 240 PRINT "[RVS]J[OFF] MBENENNEN", "0R" | 1937 | 900 PRINT SPC(17):INPUT Y* | 1357 |
| 250 PRINT "[RVS]L[OFF] DESCHEN", "0S" | 1686 | 910 IF X*=Y* THEN 940 | 1426 |
| 260 PRINT "[RVS]P[OFF] RGBLOCKCHECK = KETTUNG PRUEFEN" | 3433 | 920 IF LEN (Y*)>16 THEN PRINT "[DOWN]MAX. 16 ZEICHEN!":GOTO 890 | 3498 |
| 270 PRINT "[RVS]Z[OFF] URUECKHOLEN[2SPACES] GELOESCHTER FILES" | 3515 | 925 Y*=LEFT\$ (Y*+"[16SHIFT-SPACE]",16) | 2271 |
| 280 PRINT "[RVS]Q[OFF] ENDE!" | 1155 | 930 X=144:GOSUB 3580:PRINT#2,Y*:F=1 | 2409 |
| 290 GET X*:IF X*="" THEN 290 | 1541 | 940 X=162:GOSUB 3580:X*="":FOR U=1 TO 5:GET #2,Y*:X*=X*+Y*:NEXT | 4101 |
| 300 IF X*="0" THEN X*="[RVS]FEHLERKANAL 0":GOTO 510 | 3365 | 950 PRINT "[DOWN]ALTE DISK-ID=[3SPACES,RVS]X*" | 1959 |
| 310 IF X*="G" THEN 540 | 1570 | 960 PRINT "NEUE DISK-ID=[3SPACES]X*" | 1875 |
| 320 IF X*="C" THEN 1260 | 1587 | 970 PRINT "[UP]" SPC(14):INPUT Y* | 1078 |
| 330 IF X*="A" THEN 850 | 1383 | 980 IF Y*=X* THEN 1010 | 1598 |
| 340 IF X*="I" THEN X*="[RVS]INITIALISIEREN 0I":GOTO 510 | 3681 | 990 IF LEN (Y*)>5 THEN PRINT "[DOWN]MAX. 5 ZEICHEN!":GOTO 960 | 3372 |
| 350 IF X*="N" THEN 670 | 1546 | 1000 X=162:GOSUB 3580:PRINT#2,Y*:F=1 | 2393 |
| 360 IF X*="V" THEN 1050 | 1677 | 1010 IF F=1 THEN PRINT#1,"U2 2 0 18 0" | 1781 |
| 370 IF X*="D" THEN 3410 | 1605 | 1020 GOTO 3740 | 623 |
| 380 IF X*="O" THEN 3200 | 1694 | 1050 GOSUB 1640:PRINT "[RVS]VALIDIEREN 0V":GOSUB 3050:X*="V" | 3349 |
| 390 IF X*="K" THEN 1090 | 1596 | 1060 OPEN 1,G%,15,X*:GOTO 3740 | 2842 |
| 400 IF X*="U" THEN 1170 | 1655 | 1090 GOSUB 1640:PRINT "[RVS]KOPIEREN 0C":GOSUB 3050:T*="C:" | 3584 |
| 410 IF X*="L" THEN 1200 | 1596 | 1100 INPUT "[DOWN]NAME ALT:[2SPACES,CBM F, 3LEFT]";X* | 2460 |
| 420 IF X*="F" THEN 1670 | 1695 | 1110 INPUT "[DOWN]NAME NEU:[2SPACES,CBM F, 3LEFT]";Y* | 2479 |
| 430 IF X*="P" THEN 1960 | 1505 | 1120 IF X*="[CBM F]" OR Y*="[CBM F]" THEN RUN | 1922 |
| 440 IF X*="S" THEN 2420 | 1578 | 1130 IF LEN (X*)>16 OR LEN (Y*)>16 THEN 1100 | 2282 |
| 450 IF X*="Z" THEN 2590 | 1577 | 1140 X*=T*+Y*+",";X*:GOTO 1060 | 2104 |
| 460 IF X*="0" THEN END | 1159 | 1170 GOSUB 1640:PRINT "[RVS]UMBENENNEN 0R":GOSUB 3050:T*="R":GOTO 1100 | 3927 |
| 470 IF X*="E" THEN 4160 | 1652 | 1200 GOSUB 1640:PRINT "[RVS]LOESCHEN[OFF]-ENDE MIT (RETURN)":GOSUB 3050 | 4164 |
| 480 GOTO 290 | 594 | 1210 INPUT "[DOWN]PROGRAMMNAME=[3SPACES,CBM F, 3LEFT]";X*:IF LEN (X*)>16 THEN 1210 | 4682 |
| 510 GOSUB 1640:PRINT X*:GOSUB 3050:GOSUB 3150:GOTO 3750 | 2691 | 1220 IF X*="[CBM F]" THEN RUN | 1246 |
| 540 GOSUB 1640:PRINT "[RVS]GERAETENR.DER FLOPPY AENDERN" | 3669 | 1230 X*="S":X*:GOTO 1060 | 1382 |
| 550 PRINT "[DOWN]BITTE ALLE FLOPPYS BIS AUF DIE, DEREN:PRINT "NR. GEAENDERT WERDEN SOLL:" | 6382 | 1260 GOSUB 1640:PRINT "[RVS]CHECK DISK[OFF] =UEBERPRUEFEN ALLER BLOECKE":GOSUB 3050 | 5064 |
| 560 PRINT "[RVS]AUSSCHALTEN !" | 1879 | 1270 PRINT "[DOWN,RVS]L[OFF]ESEN ALLER BLOECKE (CA 5 MIN DAUER)" | 3963 |
| 570 INPUT "[DOWN]GERAETEADR. ALT:[2SPACES, CBM F,3LEFT]";X*:G%=VAL (X*) | 4217 | 1280 PRINT "[2SPACES]FILES WERDEN [RVS]NICHT [OFF] ZERSTOERT !" | 2965 |
| 580 INPUT "[DOWN]GERAETEADR. NEU:[2SPACES, CBM F,3LEFT]";X*:Y=VAL (X*) | 3690 | 1290 PRINT "[DOWN,RVS]B[OFF]ESCHREIBEN UND VERIFIZIEREN (CA 10 MIN):" | 4381 |
| 590 IF G%<8 OR G%>15 OR Y<8 OR Y>15 THEN PRINT "[DOWN,RVS]UNZULAESSIGE GERAETEADRESSE !":GOTO 3750 | 6568 | 1300 PRINT "[2SPACES]DISKETTE[2SPACES]WIRD [RVS]GELOESCHT[OFF] !!!" | 3391 |
| 600 GOSUB 3070 | 623 | 1310 PRINT "[DOWN,RVS]Z[OFF]UM MENUE ZURUECK !" | 2277 |
| 610 OPEN 1,G%,15 | 936 | 1320 PRINT "[2DOWN]GEBEN SIE [RVS]L[OFF]/[RVS]B [OFF] ODER [RVS]Z[OFF] EIN ! [2SPACES]"; | 3773 |
| 620 PRINT#1,"M-W" CHR* (119) CHR* (8) CHR* (2) CHR* (Y+32) CHR* (Y+64):CLOSE 1 | 4583 | 1330 GET Z*:IF Z*(">"L" AND Z*(">"B" AND Z*(">"Z" THEN 1330 | 3795 |
| 630 G%=Y:POKE 106,Y | 1381 | 1340 PRINT "[RVS]Z* [OFF]":IF Z*="Z" THEN RUN | 1710 |
| 640 PRINT "[DOWN]< GERAETEADRESSE IN # " Y*GEAENDERT >":GOSUB 3060:GOTO 3750 | 5182 | 1350 PRINT "[DOWN]Z WIRD GEPRUEFT: SPUR | 3435 |
| 670 GOSUB 1640:PRINT "[RVS]FORMATIEREN 0N" | 2153 | | |
| 680 Z=1:GOSUB 3050:PRINT "[DOWN]L E E R E DISKETTE EINLEGEN!!!" | 4027 | | |
| 690 PRINT "[DOWN]ZURUECK INS MENUE MIT 'Z'" | 2785 | | |

| | | | |
|---|------|--|--|
| [29SPACES]SEKTOR[DOWN]* | | | |
| 1360 OPEN 1,8,15:OPEN 2,8,2,"H" | 1987 | | |
| 1370 F=0:Y=0:A\$(0)="17":A\$(1)="20":A\$(2)="24": A\$(3)="18":A\$(4)="30":A\$(5)="17" | 6768 | | |
| 1380 A\$(6)="35":A\$(7)="16":IF Z\$="B" THEN GOSUB 1480 | 3953 | | |
| 1390 FOR Z=0 TO 7 STEP 2:Y=X+1:Y=VAL (A\$(Z)):W= VAL (A\$(Z+1)) | 4395 | | |
| 1400 FOR T=X TO Y:FOR S=0 TO W:IF Z\$="B" THEN PRINT#1,"U2 2 0":T:S:GOTO 1420 | 4389 | | |
| 1410 GOSUB 3940 | 631 | | |
| 1420 GOSUB 1500:NEXT :NEXT :CLOSE 1 | 1356 | | |
| 1430 IF F=0 THEN PRINT "[DOWN]< DISK OK.ALLE BLOECKE LESBAR >":GOTO 1460 | 3947 | | |
| 1440 PRINT "[2DOWN]FOLGENDE BLOECKE SIND DEFEKT:[DOWN]" | 2942 | | |
| 1450 FOR U=1 TO F:PRINT "[2SPACES]" RIGHT\$ (" " +STR\$ (T(U)),2)," RIGHT\$ (" "+STR\$ (S(U)),2)" :NEXT | 5398 | | |
| 1460 IF Z\$="B" THEN X\$="N:CHECK DISK":Y\$="99": GOTO 810 | 3757 | | |
| 1470 X\$="I":GOTO 1860 | 1423 | | |
| 1480 Y\$="":FOR X=1 TO 51:Y\$=Y\$+CHR\$ (RND (162)*255):NEXT | 3482 | | |
| 1490 PRINT#2,Y\$Y\$Y\$Y\$:RETURN | 1862 | | |
| 1500 PRINT SPC(19)*[UP,8SPACES,UP]:PRINT SPC(19):T " S | 2340 | | |
| 1510 GOSUB 3170:IF X=0 THEN RETURN | 1516 | | |
| 1520 F=F+1:T(F)=T(S(F)):F:IF F<20 THEN RETURN | 2440 | | |
| 1530 CLOSE 2:CLOSE 1:PRINT "[2DOWN]MEHR ALS 20 BLOECKE DEFEKT !":GOTO 3780 | 3662 | | |
| 1540 PRINT "[DOWN]TEST ABGEBROCHEN !":GOTO 3760 | 2626 | | |
| 1570 PRINT "[UP]" | 5668 | | |
| * | | | |
| 1580 PRINT "[SHIFT -] (C) DIPL.ING.ANDREAS SOIK 84.06.1985[SHIFT -]" | 3881 | | |
| 1590 PRINT "[SHIFT -] ([10SPACES] MAKLRAINSTR.5[10SPACES,SHIFT -]" | 3269 | | |
| 1600 PRINT "[SHIFT -] A,[10SPACES]10000 MUENCHEN 90[7SPACES,SHIFT -]" | 2976 | | |
| 1610 PRINT "[SHIFT -] ([10SPACES])TEL.(089) 692 13 23[4SPACES,SHIFT -]" | 3344 | | |
| 1620 PRINT " | 5290 | | |
| 1630 RETURN | 142 | | |
| 1640 PRINT "[CLR,RVS,CBM 7]DISKORG5.3 (C) DIPL.ING.A.SOIK[2SPACES]4.6.1985[WHIT]: RETURN | 4938 | | |
| 1670 GOSUB 1640:PRINT "[RVS]FORMATKENNZEICHEN AENDERN":GOSUB 3050 | 4415 | | |
| 1680 PRINT "[DOWN]SCHREIBSCHUTZ SETZEN DURCH AENDERUNG" | 3975 | | |
| 1690 PRINT "DES FORMATKENNZEICHENS IN DER BAM !!!" | 3522 | | |
| 1700 PRINT "[DOWN]AB SOFORT KOENNEN KEINE SCHREIBVORGAENDE:" | 4615 | | |
| 1710 PRINT "AUSSER NEUFORMATIEREN DURCHGEFUEHRT:PRINT "WERDEN !:[DOWN]" | 5062 | | |
| 1720 PRINT "[RVS]S[OFF]=SCHREIBSCHUTZ SETZEN [DOWN]" | 2701 | | |
| 1730 PRINT "[RVS]L[OFF]=SCHREIBSCHUTZ LOESCHEN [DOWN]" | 3158 | | |
| 1740 PRINT "GEBEN SIE [RVS]S[OFF] ODER [RVS]L [OFF] EIN !" | 2266 | | |
| 1750 GET X\$:IF X\$(">S" AND X\$(">L" THEN 1750 | 2815 | | |
| 1760 OPEN 1,0%,15:OPEN 2,0%,2,"H" | 1939 | | |
| 1770 GOSUB 3930:X=2:GOSUB 3580 | 1200 | | |
| 1780 IF X\$="S" THEN PRINT#2,"X":X\$=" GESETZT": GOTO 1820 | 4466 | | |
| 1790 X\$=" GELOESCHT:" | 1482 | | |
| 1800 PRINT#1,"M-W" CHR\$ (1) CHR\$ (1) CHR\$ (1) CHR\$ (65) | 2964 | | |
| 1810 PRINT#2,"A": | 739 | | |
| 1820 PRINT "[DOWN]<SCHREIBSCHUTZ*X\$:PRINT#1,"U2 2 0 18 0" | 2788 | | |
| 1830 PRINT#1,"I":GOTO 3740 | 1495 | | |
| 1860 GOSUB 1640:PRINT "[RVS]PROGRAMMBLOCKCHECK [OFF]=FILEKETTUNG PRUEFEN":GOSUB 3050 | 5510 | | |
| 1880 GOSUB 4650 | 633 | | |
| 1890 PRINT "[DOWN,RVS]E[OFF] INZELNES FILE DARSTELLEN" | 3236 | | |
| 1900 PRINT "[RVS]B[OFF] LOCK SUCHEN IN FILE" | 2162 | | |
| 1910 PRINT "[RVS]A[OFF] LLE FILES DARSTELLEN" | 2389 | | |
| 1920 PRINT "[RVS]Z[OFF] URUECK INS MENUE[DOWN]" | 2018 | | |
| 1930 PRINT "GEBEN SIE [RVS]E[OFF]/[RVS]B[OFF]/ [RVS]A[OFF] ODER [RVS]Z[OFF] EIN !:" | 3349 | | |
| 1940 GET X\$:IF X\$(">E" AND X\$(">B" AND X\$(">A" AND X\$(">Z" THEN 1940 | 4872 | | |
| 1950 PRINT "[RVS]*X\$:S(20)=ASC (X\$):IF X\$="Z" THEN RUN | 2902 | | |
| 1960 IF X\$="B" THEN 2000 | 1528 | | |
| 1970 IF X\$="A" THEN 2020 | 1504 | | |
| 1980 INPUT "[DOWN]FILENAME:[3SPACES,CBM F, 3LEFT]":Z\$:IF Z\$="[CBM F]" THEN RUN | 3301 | | |
| 1990 GOTO 2020 | 577 | | |
| 2000 INPUT "[DOWN,RVS,2SPACES]SPUR[OFF,2SPACES] 0[3LEFT]":T(19) | 2329 | | |
| 2010 INPUT "[RVS]SEKTOR[OFF,2SPACES]0[3LEFT]": S(19) | 1969 | | |
| 2020 PRINT "[DOWN]<DIRECTORY EINLESEN" | 2623 | | |
| 2030 GOSUB 3970:U=T(19):V=S(19) | 2622 | | |
| 2040 IF Z=2 THEN 3780 | 1171 | | |
| 2050 IF S(20)<>69 THEN 2100 | 1610 | | |
| 2060 U=LEN (Z\$) | 637 | | |
| 2070 FOR F=3 TO 2:IF LEFT\$ (A\$(F),U)=Z\$ THEN S(20)=3:GOTO 2110 | 4020 | | |
| 2080 NEXT | 130 | | |
| 2090 PRINT "[2DOWN,RVS]FEHLER: " CHR\$ (34)Z\$ CHR\$ (34)" NICHT GEFUNDEN !":GOTO 3780 | 5139 | | |
| 2100 FOR F=3 TO 2-1 | 1159 | | |
| 2110 PRINT "[DOWN]" LEFT\$ (A\$(F),26) | 1297 | | |
| 2120 PRINT "[25CBM Y]":W=0 | 5643 | | |
| 2130 IF D=1 THEN PRINT#5,LEFT\$ (A\$(F),26): PRINT#5,"-----" | 5633 | | |
| 2140 T=ASC (MID\$ (A\$(F),28)):S=ASC (MID\$ (A\$(F),29)) | 2871 | | |
| 2150 IF T=U AND S=V THEN GOTO 2290 | 1934 | | |
| 2160 X\$=RIGHT\$ (STR\$ (T),2):Y\$=RIGHT\$ (STR\$ (S),2) | 2012 | | |
| 2170 GOSUB 2390:W=W+1 | 1390 | | |
| 2180 GET Z\$:IF Z\$(">)" THEN 3780 | 1940 | | |
| 2190 GOSUB 3840 | 631 | | |
| 2200 GET #2,X\$:GET #2,Y\$:T=ASC (X\$+CHR\$ (0)):S= ASC (Y\$+CHR\$ (0)) | 4844 | | |
| 2210 X\$=RIGHT\$ (STR\$ (T),2):Y\$=RIGHT\$ (STR\$ (S),2) | 2012 | | |
| 2220 IF T=U AND S=V THEN GOTO 2290 | 1934 | | |
| 2230 IF T(">0 THEN 2260 | 1216 | | |
| 2240 PRINT :IF S(20)=3 THEN 2330 | 1577 | | |
| 2250 NEXT :GOTO 2330 | 594 | | |
| 2260 W=W+1:IF W=5 THEN W=0:GOSUB 2360:GOTO 2180 | 3023 | | |
| 2270 GOSUB 2390 | 649 | | |
| 2280 GOTO 2180 | 629 | | |
| 2290 PRINT X\$,"Y\$" | 856 | | |
| 2300 PRINT "[DOWN,RVS]ENTHAELT GESUCHTEN BLOCK !:[UP]" | 2905 | | |
| 2310 IF D=1 THEN PRINT#5,X\$,"Y\$:PRINT#5, "ENTHAELT GESUCHTEN BLOCK !" | 4304 | | |
| 2320 GOTO 3780 | 655 | | |
| 2330 IF D=1 THEN PRINT#5 | 832 | | |
| 2340 GOTO 3780 | 655 | | |
| 2360 PRINT X\$,"Y\$" | 856 | | |
| 2370 IF D=1 THEN PRINT#5,X\$,"Y\$" | 1333 | | |
| 2380 RETURN | 142 | | |
| 2390 PRINT X\$,"Y\$"[3SPACES]:" | 1220 | | |
| 2400 IF D=1 THEN PRINT#5,X\$,"Y\$"[5SPACES]:" | 1851 | | |
| 2410 RETURN | 142 | | |
| 2420 GOSUB 1640:PRINT "[RVS]SCHREIBSCHUTZBITS AENDERN":GOSUB 3050 | 4231 | | |
| 2430 T(20)=3:GOSUB 3870 | 1613 | | |
| 2440 PRINT "[DOWN]LOESCHSCHUTZ-BIT AENDERN: [RVS]J/N[OFF,DOWN]" | 3739 | | |
| 2450 T(0)=T:S(0)=S | 1423 | | |
| 2460 T(20)=2:S(20)=2:GOSUB 3960 | 2438 | | |
| 2470 PRINT "[DOWN]<ENDE DIRECTORY>" | 2230 | | |
| 2480 GOTO 3740 | 623 | | |
| 2490 GET X\$:IF X\$(">J" AND X\$(">N" THEN 2490 | 2629 | | |
| 2500 PRINT SPC(30)*[UP,RVS]"X\$" | 1240 | | |
| 2510 IF X\$="N" THEN 4110 | 1665 | | |
| 2520 IF V AND 64 THEN V=V-64:Y\$="":GOTO 2540 | 2897 | | |
| 2530 V=V OR 64:Y\$="<" | 1488 | | |
| 2540 PRINT "[UP]" SPC(25)Y\$ | 870 | | |
| 2550 GOSUB 3560:PRINT#2,CHR\$ (V):PRINT#1,"U2 2 3851 | 3851 | | |

| | |
|--|---|
| <pre> 0*T(F-1);S(F-1) 2560 GOTO 4110 2590 GOSUB 1640:PRINT "[RVS]ZURUECKHOLEN GELOESCHTER FILES" 2610 PRINT "[DOWN]ERMUEGLICH DAS WIEDERHERSTELLEN" 2620 PRINT "VON VERSEHENTLICH GELOESCHTEN FILES" 2630 PRINT "(NATUERLICH NUR DANN, WENN DIESE NICHT" 2640 PRINT "SCHON UEBERSCHRIEBEN WORDEN SIND)" 2660 PRINT "[DOWN]BEIM SUCHEN WERDEN NUR DIE GELOESCHTEN[2SPACES]DATEIEN ABGEFRAGT" 2670 GOSUB 3050:GOSUB 1640 2680 PRINT "[DOWN]<BEGINN DER SUCHE>" 2690 T(20)=1:GOSUB 3070:PRINT "[DOWN]<ENDE DIRECTORY>" 2700 PRINT "[DOWN]KEINE WEITEREN GELOESCHTEN FILES!":GOTO 3740 2710 : 2720 PRINT LEFT*(X*,21) 2730 GOSUB 2740:GOTO 4120 2740 PRINT ",ZURUECKHOLEN J/N?" 2750 GET T*:IF T*((">"J" AND T*((">"N" THEN 2750 2760 IF T*="N" THEN RETURN 2770 PRINT "[DOWN]WAR ES EIN [RVS]P[OFF] ROGRAMM-[5SPACES,RVS]S[OFF]ESENTIELLES-" 2780 PRINT "[11SPACES,RVS]R[OFF]ELATIVES- [4SPACES,RVS]U[OFF]SER-F I L E ?" 2790 GET T*:IF T*((">"P" AND T*((">"S" AND T*((">"R" AND T*((">"U" THEN 2790 2800 IF T*="P" THEN Z=130 2810 IF T*="S" THEN Z=129 2820 IF T*="R" THEN Z=132 2830 IF T*="U" THEN Z=131 2840 OPEN 3,0%,3,"#":PRINT "[DOWN]<FILE WIEDERHERSTELLEN...>[DOWN]":PRINT "SPUR SEKTOR" 2850 T=ASC (MID*(X*,28)):S=ASC (MID*(X*, 29)) 2860 PRINT "[7SPACES,UP]":PRINT T;S"[UP]" 2870 PRINT#1,"B-A 0*T;S:GOSUB 3170 2880 IF X(>0 THEN 2900 2890 PRINT#1,"U1 3 0*T;S 2900 GET #3,Z#;GET #3,Y# 2910 T=ASC (Z#+CHR*(0)):S=ASC (Y#+CHR*(0)) 2920 IF T(>0 THEN 2960 2930 X=2+W*32:GOSUB 3500 2940 PRINT#2,CHR*(Z); 2950 PRINT#1,"U2 2 0*T(F-1);S(F-1) 2960 PRINT "[2DOWN]FILE WIEDERHERGESTELLT!" 2970 GOTO 3740 2980 IF X(>65 THEN 3150 2990 PRINT "[2DOWN]FILE WURDE BEREITS UEBERSCHRIEBEN!!" 3000 PRINT "WIEDERHERSTELLUNG[2SPACES]NICHT [2SPACES]MOEGLICH!!" 3010 PRINT "[DOWN]<VALIDIEREN>":PRINT#1,"V" 3020 GOTO 3780 3050 PRINT "[DOWN]FLOPPY-GERAETEADRESSE: "G% [4LEFT] "CHR*(INT(G%/10)+15)";INPUT G% 3060 IF G%<8 OR G%>15 THEN PRINT "[DOWN,RVS] UNZULAESSIGE GERAETEADRESSE !":GOTO 3750 3070 V=768:X=PEEK (V):POKE V,105:IF Z=1 THEN OPEN 1,0%,15:CLOSE 1:GOTO 3090 3080 OPEN 1,0%,15,"1" 3090 POKE V,X:IF PEEK (144)<>120 THEN 3110 3100 PRINT "[DOWN,RVS]FEHLER: FLOPPY#G%" [ASC20] NICHTeingeschaltet!":CLOSE 1:GOTO 3750 3110 IF Z=1 THEN RETURN 3120 GOSUB 3170:CLOSE 1 3130 IF X(>0 THEN 3150 3140 RETURN 3150 PRINT "[DOWN,RVS]FEHLERKANAL:":PRINT X* X* "U" "V"[DOWN]" 3160 GOTO 3750 3170 INPUT#1,X,X*,U,V:RETURN 3200 GOSUB 1640:PRINT "[RVS]ORDNEN DIRECTORY (ALPHABETISCH)" 3210 GOSUB 3050 3220 PRINT "[DOWN]<DIRECTORY EINLESEN>" 3230 T(20)=4:GOSUB 3070 </pre> | <pre> 3240 PRINT "[DOWN]<DIRECTORY WIRD SORTIERT>" 3250 SYS 49205:IF T(0)=10 AND S(0)=1 THEN 3260 3255 PRINT "[DOWN]FOLGEBLOCK VON T10,0 =" T(0)*,"S(0)";[RVS,RED]# 10,1[2SPACES]! [WHIT]":GOTO 3780 3260 PRINT "[DOWN]<DIRECTORY ZURUECKSCHREIBEN>" IF=0 3270 PRINT#1,"U1 2 0*T(F);S(F) 3280 FOR W=0 TO 7:X=2+W*32:GOSUB 3500 3290 IF ASC (MID*(A*(U),28))=0 THEN A*(U)= RIGHT*(A*(U),32) 3300 PRINT#2,MID*(A*(U),27,3);LEFT*(A*(U), 16);MID*(A*(U),30,3); 3310 X=X+28:GOSUB 3500 3320 Y=VAL (MID*(A*(U),17,3)):X=INT (Y/256): Y=Y-X*256 3330 PRINT#2,CHR*(Y);CHR*(X); 3340 U=U+1:IF U>2-1 THEN 3360 3350 NEXT 3360 PRINT#1,"U2 2 0*T(F);S(F) 3370 F=F+1:IF T(F)<0 THEN GOTO 3270 3380 GOTO 3740 3410 GOSUB 1640:PRINT "[RVS]DIRECTORY 0*[OFF] SCROLL:CURSOR UP/DOWN":PRINT SPC(13)"ENDE [2SPACES]":RETURN" 3420 GOSUB 3050:GOSUB 4590:GOSUB 4650 3430 IF A(>1 THEN T(20)=2 3440 GOSUB 3070:IF A(>1 THEN 3510 3450 A*(0)*="!A*(1)="[2SPACES]"+A*(1):A*(2)*=" "+A*(2) 3460 SYS 49205 3470 FOR I=3 TO 2-1:PRINT LEFT*(A*(1),26) 3480 IF D=1 THEN PRINT#5,LEFT*(A*(1),26) 3490 NEXT 3500 A*(1)=RIGHT*(A*(1),25):A*(2)=RIGHT* (A*(2),25) 3510 PRINT#1,"M-R" CHR*(250) CHR*(2):GET #1, X# 3520 PRINT#1,"M-R" CHR*(252) CHR*(2):GET #1, Y# 3530 F=ASC (X#+CHR*(0))+ASC (Y#+CHR*(0))*256 3540 A*(2)="BELEGT"+STR*(664-F)+"",FREI"+STR* (F)+" V.664[UP]" 3541 PRINT A*(2):PRINT A*(2) 3550 IF D=1 THEN PRINT#5,A*(2):PRINT#5,A*(2): CLOSE 5 3560 CLOSE 2:CLOSE 1:Z=2-1:U=2:GOTO 3600 3580 PRINT#1,"B-P";Z;X:RETURN 3600 GET X#:IF X*="" THEN 3600 3610 IF X*="[UP]" THEN 3670 3620 IF X*="[DOWN]" THEN 3650 3630 IF X#=CHR*(13) THEN RUN 3640 GOTO 3600 3650 IF U>2 THEN 3600 3660 U=U+1:PRINT :PRINT LEFT*(A*(U+1),26)"; GOTO 3600 3670 IF U=24<0 THEN 3600 3680 SYS 49152 3690 PRINT "[HOME]" LEFT*(A*(U-24),26)" [23DOWN]" 3700 U=U-1:GOTO 3600 3710 STOP 3740 CLOSE 2:CLOSE 3:GOSUB 3120:GOSUB 3150 3750 FOR U=1 TO 500:GET X#:IF X*((">"" THEN U= 5000 3760 NEXT :RUN 3770 : 3780 CLOSE 5:CLOSE 2:CLOSE 1:PRINT :PRINT SPC(10)"<<< TASTE >>" 3790 GET X#:IF X*="" THEN 3790 3800 RUN 3830 T=10:S=0 3840 PRINT#1,"U1 2 0*T;S:RETURN 3870 OPEN 1,0%,15:PRINT#1,"1":OPEN 2,0%,2,"#": Z=3 3880 GOSUB 3070:GET #2,X#:GET #2,Y#:T=ASC (X#+ CHR*(0)):S=ASC (Y#+CHR*(0)) 3890 T(0)=T:S(0)=S 3900 X=144:GOSUB 3500:X*=""!Y*="" 3910 FOR V=1 TO 23:GET #2,X#:Y#=Y#+X#:NEXT : A*(1)=LEFT*(Y*,16)+"[2SPACES]"+RIGHT* (Y*,7) 3920 PRINT "[DOWN]A*(1):PRINT A*(2) </pre> |
|--|---|

| | | | |
|--|---|---|--|
| <pre> 3930 IF D>0 THEN PRINT#5:PRINT#5,A*(1):PRINT#5, A*(2) 3940 IF T(20)=3 THEN RETURN 3950 IF T(20)=4 THEN Z=0 3960 GOSUB 3840:GET #2,X#1:GET #2,Y#1:T=ASC (X#+ CHR*(0)):S=ASC (Y#+CHR*(0)) 3970 F=F+1:T(F)=T(S)=S 3980 FOR W=0 TO 7:K=2+W*32:GOSUB 3580 3990 SYS 820,2,X# 4000 V=ASC (MID*(X#,27)):Y=ASC (MID*(X#, 28)) 4010 IF V=0 AND Y(<0) AND T(20)=1 THEN 2720 4020 IF V=0 AND T(20)=4 AND Y(<0) THEN X#=CHR* (255)+X#:GOTO 4040 4030 IF V=0 THEN 4120 4040 A*(2)=X# 4050 IF T(20)<>2 THEN 4090 4060 PRINT LEFT*(A*(2),26) 4070 IF D=1 THEN PRINT#5,LEFT*(A*(2),26) 4080 IF D=2 THEN PRINT#5,LEFT*(A*(2),26): 4090 IF S(20)=2 THEN 2490 4100 IF S(20)=3 THEN 4230 4110 Z=Z+1 4120 NEXT :IF T=0 THEN 4130 4122 IF T=18 AND S<21 THEN 3960 4124 PRINT "[DOWN],RVS,REDJUNZLAESSIGER FOLGEBLOCK [OFF,WHI]T",S:GOTO 3780 4130 RETURN 4160 GOSUB 1640:PRINT "[RVS]ERWEITERTES DIRECTORY" 4161 PRINT "[DOWN]=ANZEIGE SAEMTLICHER FILEPARAMETER:PRINT "(ABBRUCH MIT (RETURN))" 4170 GOSUB 3050 4190 GOSUB 4650:IF D=1 THEN D=2 4200 PRINT "[DOWN](DIRECTORY EINLESEN)" 4210 T(20)=2:S(20)=3:S(19)=664:GOSUB 3870:GOTO 4510 4230 OPEN 4,0%,4,"# 4240 V=V AND 15:S(19)=S(19)-VAL (MID*(A*(2), 18,3)) 4280 IF V=4 THEN 4405:REM REL 4290 X=ASC (MID*(A*(2),28)):Y=ASC (MID*(A*(2),29)) 4300 PRINT#1,"U1 4 0*X;Y 4310 GET #4,X#1:GET #4,X#1:GET #4,X#1:GET #4,Y#1 CLOSE 4 4320 X=ASC (X#+CHR*(0))+ASC (Y#+CHR*(0))*256 4330 Y=ASC (MID*(A*(2),28)):U=ASC (MID*(A*(2),29)) 4350 X#=RIGHT*("[3SPACES]" +STR*(X),5) 4351 IF V=1 THEN X#="[5SPACES]" 4360 Y#=RIGHT*(STR*(Y),2):Z#=RIGHT*(STR*(U),2) 4380 IF X(<2048) AND V(<1) THEN PRINT "[RVS]";:IF D=2 THEN PRINT#5,"[RVS]"; 4381 PRINT "[UP]";TAB(26);X#[OFF] "Y#",Z# 4390 IF D=2 THEN PRINT#5," X#[OFF] "Y#",Z# 4400 GOTO 4480 4405 X=ASC (RIGHT*(A*(2),1)):Y#=MID*(A*(2), 17,3):Y=VAL (Y#) 4420 U=Y/121:IF U(<1) THEN U=INT (U+1) 4430 V=INT ((Y-U)*254/X) 4450 PRINT "[UP]";TAB(26);Y-U:TAB(30);"D";ASC (MID*(A*(2),28)): "[LEFT]";ASC (MID*(A*(2),29)) 4460 PRINT TAB(26);U:TAB(30);"S";ASC (MID*(A*(2),30)): "[LEFT]";ASC (MID*(A*(2), 31)) 4461 PRINT TAB(26);V:TAB(30);"R.JE";X 4471 IF D<>2 THEN 4480 4472 PRINT#5,Y-U;"D"; 4473 PRINT#5,ASC (MID*(A*(2),28)): "[LEFT]"; ASC (MID*(A*(2),29)) 4474 PRINT#5,TAB(26);U;" S";ASC (MID*(A*(2), 30)): "[LEFT]";ASC (MID*(A*(2),31)) 4476 PRINT#5,TAB(26);V;"R.JE";X 4480 FOR V=1 TO 100:GET X#1:IF X#(<)" THEN 3780 4500 NEXT :CLOSE 4:GOTO 4110 </pre> | <pre> 2232 1173 1528 4773 1461 2882 954 2314 2434 4382 1047 779 1270 1795 2020 1526 1602 400 1105 2267 5049 142 2040 5715 607 1545 2623 4161 987 3254 1136 3007 1200 3476 2673 3000 1342 1250 2207 3914 1712 2407 641 3721 2950 2065 5536 5377 2802 1892 1597 4014 5069 2045 2295 1059 </pre> | <pre> 4510 PRINT#1,"M-R" CHR*(250) CHR*(2):GET #1, X# 4520 PRINT#1,"M-R" CHR*(252) CHR*(2):GET #1, Y# 4530 F=ASC (X#+CHR*(0))+ASC (Y#+CHR*(0))*256 4540 PRINT A*(2):PRINT "VON 664 BLOECKEN:PRINT "BELEG":664-F;" FREI":F; 4541 PRINT " NACH BAM-BELEGUNG" 4542 IF D<>2 THEN 4550 4545 PRINT#5,A*(2):PRINT#5,"VON 664 BLOECKEN": PRINT#5,"BELEG":664-F; 4546 PRINT#5," FREI":F;" NACH BAM-BELEGUNG" 4550 PRINT "[6SPACES]";664-S(19);"[5SPACES]"; S(19);" NACH FILESUMMEN" 4555 IF D=2 THEN PRINT#5,"[6SPACES]";664-S(19); "[5SPACES]";S(19);" NACH FILESUMMEN" 4556 IF F(<S(19)) THEN PRINT "[DOWN,CBM 2,RVS] A B W E I C H U N G , P R U E F E N !!!": POKE 646,1 4557 IF D=2 AND F(<S(19)) THEN PRINT#5,"[RVS] A B W E I C H U N G , P R U E F E N !!!" 4560 GOTO 3780 4590 PRINT "[DOWN]ALPHABETISCH GEORDNET J/N [3SPACES]N[3LEFT]"; 4600 INPUT X#:IF X#(<"N" AND X#(>"J" THEN 4590 4610 IF X#="J" THEN A=1 4620 RETURN 4630 PRINT "[DOWN]AUSGABE AUF[2SPACES]DRUCKER [2SPACES]J/N[3SPACES]N[3LEFT]"; 4660 INPUT X#:IF X#(<"N" AND X#(>"J" THEN 4650 4670 IF X#="J" THEN D=1:OPEN 5,4 4680 RETURN 4800 FOR I=820 TO 1023:READ X:POKE 1,X:NEXT 4801 FOR I=49152 TO 49355:READ X:POKE 1,X:NEXT :RETURN 4802 DATA 32,155,183,32,30,225,32,253,174,32, 139,176,133,73,132,74,32,163,182 4803 DATA 169,32,32,117,180,160,2,185,97,0,145, 73,136,16,248,165,98,133,73 4804 DATA 165,99,133,74,160,26,32,244,3,160,0, 32,18,225,145,73,200,192,16,208 4805 DATA 246,132,139,160,29,32,244,3,164,139, 162,7,32,18,225,202,208,250,133 4806 DATA 99,32,18,225,133,98,32,204,255,132, 139,162,144,56,32,73,188,32,223 4807 DATA 189,164,139,162,0,189,250,0,201,0, 240,6,145,73,200,232,208,243,169 4808 DATA 32,145,73,200,232,224,5,208,248,160, 28,177,73,136,72,10,16,4,169 4809 DATA 60,208,2,169,32,145,73,136,184,41,15, 170,189,238,3,145,73,136,189 4810 DATA 233,3,145,73,136,189,228,3,145,73, 136,176,4,169,42,208,2,169,32,145 4811 DATA 73,96,0,68,83,80,85,82,69,69,82,83, 69,76,81,71,82,76,0,162,3,32,18 4812 DATA 225,145,73,200,202,208,247,96 4813 DATA 169,191,133,98,133,180,169,219,133, 181,169,7,133,99,162,0,160,48 4814 DATA 161,98,145,98,161,180,145,100,198, 180,169,255,215,98,208,240,198 4815 DATA 99,198,181,165,99,201,3,208,238,169, 32,145,98,136,208,251,96,0,160 4816 DATA 0,177,47,48,13,200,177,47,16,0,160,4, 177,47,201,1,240,1,96,24,165 4817 DATA 47,185,7,133,118,165,48,185,0,133, 111,160,0,177,110,240,236,133,34 4818 DATA 200,177,110,153,34,0,192,2,208,246, 165,118,133,113,165,111,133,114 4819 DATA 24,165,113,185,3,133,113,144,2,230, 114,160,0,177,113,240,61,133,77 4820 DATA 197,34,144,2,185,34,133,85,200,177, 113,153,77,0,192,2,208,246,160 4821 DATA 0,177,35,209,78,240,4,176,11,144,287, 200,196,85,208,241,196,34,176 4822 DATA 198,160,2,177,110,178,177,113,145, 110,153,34,0,138,145,113,136,16 4823 DATA 240,48,178,24,165,110,105,3,133,110, 144,143,238,111,208,139,0 </pre> | <pre> 2498 2650 2655 5895 1750 1217 5172 3151 4531 5091 4254 5237 655 3186 2729 1424 142 3430 2791 2088 142 2448 3031 7225 7209 7674 7438 7255 7271 6318 7875 7624 7201 2865 6185 6227 6601 7881 7181 6876 6886 6886 7759 6586 6736 </pre> |
|--|---|---|--|

Directory in Reih und Glied

Wer kennt das nicht: Man hat ein mehrteiliges Programm auf Diskette abgespeichert. Um sich zu vergewissern, daß alles geklappt hat, listet man das Directory. Doch von der erhofften Ordnung ist dort nichts zu entdecken. Alles steht wie Kraut und Rüben durcheinander.

„Catsort“ bringt das Chaos in Ordnung. Man kann das Directory ganz nach Wunsch sortieren. Zur ansprechenden Gestaltung lassen sich Trennstriche einfügen. Veraltete Files werden gelöscht. Wichtige Programme können gegen Überschreiben geschützt werden.

Startet man das Programm mit RUN, erscheint das Titelbild mit der Aufforderung, die zu bearbeitende Diskette ins Laufwerk einzulegen. Im folgenden Menü hat man drei Wahlmöglichkeiten:

1. Directory anzeigen
2. Directory sortieren
3. Programm beenden

Mit '1' wird das Directory ausgegeben. Die Ausgabe kann mit der CBM-Taste unterbrochen und mit der SHIFT-Taste fortgesetzt werden.

Mit '2' gelangt man in das eigentliche Sortierprogramm. Auf dem Bildschirm erscheint eine Kopfzeile mit dem Diskettenamen. Im unteren Bereich steht das Menü mit den wichtigsten Funktionen. Der Bildschirmbereich dazwischen wird als Fenster für das Directory benutzt. Ein etwas breiterer 'Cursor' zeigt, welcher Fileeintrag gerade aktuell ist. Er wird mit den Tasten UP und DOWN bewegt. Passen nicht alle Einträge auf einmal auf den Bildschirm, wird gescrollt, sobald der Zeilencursor oben oder unten an den Fensterrand stößt. Mit HOME gelangt man an den Anfang des Directorys zurück.

Der Sortiervorgang wird über die Funktionstasten gesteuert: Mit 'F3' wird das File an der aktuellen Cursorposition aus dem Directory

Inhaltsverzeichnisse auf Disketten sind oft ein einziges Durcheinander. „Catsort“ bringt die Files in eine vernünftige Ordnung.

entfernt und in einen Puffer übernommen. Alle folgenden Einträge rücken nach oben auf. Die Zahl der Einträge im Puffer ist unbegrenzt. Mit 'F5' wird der zuletzt gelesene Puffereintrag an der Position des Zeilencursors wieder ins Directory eingefügt. Mit 'F7' wird das Directory auf die Diskette zurückgeschrieben. Solange diese Taste nicht betätigt wird, bleibt die Diskette unverändert. Wenn man sich einmal total ‚versortiert‘ hat, kann man mit der '←'-Taste ins Anfangsmenü zurückspringen, ohne daß an der Diskette etwas geändert wird.

Trennungsstriche einfügen

Was passiert, wenn man vier Files in den Puffer übernommen, aber nur zwei wieder zurückgeschrieben hat? Vor dem Zurückschreiben auf Diskette zeigt ‚Catsort‘ alle im Puffer verbliebenen Files an. Man kann sie entweder löschen (Scratch) oder hinten anhängen.

Mit der Taste 'L' (Lock) kann man ein File gegen Löschen schützen oder ein geschütztes File wieder freigeben. Dieser Schutz hilft natürlich nichts gegen versehentliches Formatieren der Diskette.

Mit der Insert-Taste wird an der Cursorposition ein Trennungsstrich von 16 '-'Zeichen eingefügt.

Er wird als USR-File von Null Block Länge angelegt.

Das DOS (Floppy-Betriebssystem) speichert Programme in Blöcken zu je 256 Byte auf der Diskette ab. Anschließend werden der Name dieses Programms, die Spur- und Sektornummer des Anfangsblocks und einige weitere Informationen im Directory eingetragen. Damit ein Programm geladen werden kann, braucht nur im Directory der Anfangsblock herausgesucht zu werden. Die weiteren Spur- und Sektornummern stehen am Anfang jedes weiteren Blocks.

Das Directory befindet sich immer in der Mitte der Diskette auf Spur 18 ab dem Sektor 1. Bild 1 zeigt ein Beispieldirectory. Bild 2 zeigt die Belegung von Spur 18, Sektor 1.

Die ersten beiden Zahlen in der ersten Reihe geben den nächsten Block an, der vom Directory belegt wird. Im Beispiel folgt kein Block mehr, was durch die unmögliche Koppeladresse in den beiden Bytes angezeigt wird:

00 FF 82 11 00 44 52 55

Nun folgt eine Kennziffer, die angibt, um welchen Filetyp es sich handelt.

00 FF 82 11 00 44 52 55

| FILETYP | ABKÜRZ. | HEX-CODE |
|------------|---------|----------|
| Deleted | DEL | \$80 |
| Sequential | SEQ | \$81 |
| Program | PRG | \$82 |
| User | USR | \$83 |
| Relative | REL | \$84 |

Es handelt sich bei den Files im Beispieldirectory ausnahmslos um Programme (alle haben eine 82-Kennung).

Die nächsten Zahlen geben an, wo das erste Programm 'DRUCKERTEST' auf der Diskette abgelegt ist. Dabei wird zuerst die Spur- und dann die Sektornummer angezeigt:

00 FF 82 11 00 44 52 55

Die 16 folgenden Zahlen enthalten den Namen des ersten Programms, und zwar als ASCII-

```

0 "LEER" " LE 2A
1 "DRUCKER-TEST" PRG
1 "PRINTER 1.0" PRG
1 "DRUCKER-TEST 2" PRG
6 "CENTRONICS" PRG
655 BLOCKS FREE.

```

READY.

Bild 1: Beispieldirectory

```

:BF00 00 FF 82 11 00 44 52 55 .....DRU
:BF08 43 4B 45 52 2D 54 45 53 CKER-TES
:BF10 54 A0 A0 A0 A0 00 00 00 T.....
:BF18 00 00 00 00 00 00 01 00 .....
:BF20 00 00 82 11 02 44 52 55 .....DRU
:BF28 43 4B 45 52 2D 54 45 53 CKER-TES
:BF30 54 20 32 A0 A0 00 00 00 T 2.....
:BF38 00 00 00 00 00 00 01 00 .....
:BF40 00 00 83 12 01 2D 2D 2D .....---
:BF48 2D 2D 2D 2D 2D 2D 2D 2D -----
:BF50 2D 2D 2D 2D 2D 00 00 00 -----...
:BF58 00 00 00 00 00 00 00 00 .....
:BF60 00 00 82 11 01 50 52 49 .....PRI
:BF68 4E 54 45 52 20 31 2E 30 NTER 1.0
:BF70 A0 A0 A0 A0 A0 00 00 00 .....
:BF78 00 00 00 00 00 00 01 00 .....
:BF80 00 00 83 12 01 2D 2D 2D .....---
:BF88 2D 2D 2D 2D 2D 2D 2D 2D -----
:BF90 2D 2D 2D 2D 2D 00 00 00 -----...
:BF98 00 00 00 00 00 00 00 00 .....
:BFA0 00 00 82 11 03 43 45 4E .....CEN
:BFAB 54 52 4F 4E 49 43 53 A0 TRONICS.
:BFBC A0 A0 A0 A0 A0 00 00 00 .....
:BFBB 00 00 00 00 00 00 06 00 .....
:BFC0 00 00 00 00 00 00 00 00 .....
:BFCB 00 00 00 00 00 00 00 00 .....
:BFDC 00 00 00 00 00 00 00 00 .....
:BFDB 00 00 00 00 00 00 00 00 .....
:BFEC 00 00 00 00 00 00 00 00 .....
:BFEB 00 00 00 00 00 00 00 00 .....
:BFFC 00 00 00 00 00 00 00 00 .....
:BFFB 00 00 00 00 00 00 00 00 .....

```

Bild 2: Beispieldirectory auf Spur 18, Sektor 1

```

0 "LEER" " LE 2A
1 "DRUCKER-TEST" PRG
1 "DRUCKER-TEST 2" PRG
0 "-----" USR
1 "PRINTER 1.0" PRG
0 "-----" USR
6 "CENTRONICS" PRG
655 BLOCKS FREE.

```

READY.

Bild 3: Sortiertes Directory

Zeichen. Da in diesem Fall der Name nur zwölf Zeichen lang ist, werden die vier restlichen Bytes mit \$A0 (Shift-Space) aufgefüllt:

00 FF 82 11 00 44 52 55

43 4B 45 52 2D 54 45 53

54 A0 A0 A0 00 00 00 00

Die folgenden Bytes sind unbenutzt und mit Nullen besetzt. Am Ende der vierten Zeile steht 00 00 00 00 00 00 01 00

Hierbei handelt es sich um die Anzahl der durch das Programm 'DRUCKER-TEST' belegten Blöcke, in diesem Fall ein Block.

'Catsort' liest das gesamte Directory wie ein Programm in den Speicher ein. Dann kann man, wie oben beschrieben, mit den Funktionstasten aufräumen und alles in der neuen Reihenfolge in das Directory zurückschreiben. Beim Beispiel könnte sich für Spur 18, Sektor 1 nach dem Ändern der Zustand in Bild 4 ergeben.

Die beiden Drucker-Programme sind nun hintereinander angeordnet und mit einem Strich vom Rest abgetrennt. Im Directory-Ausdruck (Bild 3) ist das noch deutlicher zu erkennen.

Catsort wurde mit dem Assembler AS-64 erstellt. Die verwendeten Pseudo-Opcodes (wie BA, DS, BY) entsprechen weitgehend denen des MAE-Assemblers.

Will man die Farben des Programms ändern, sind dafür in erster Linie die Routinen START und CLRSCR zuständig. Bei START steht in Speicherstelle \$0813 der Befehl 'LDA #6' für blauen Hintergrund und Rahmen, in Speicherstelle \$081B der Befehl 'LDA #3' für cyan als Schreibfarbe. Bei CLRSCR findet man ebenfalls diesen Befehl (\$087B). Leider ist es damit noch nicht getan; weitere Farbsteuerzeichen findet man im Textteil: Hier sind es die Bytes \$81 (orange) und \$9F (cyan), die am Anfang oder Ende der Ausgabertexte auftreten.

Eine Routine, die man in anderen Programmen gut einsetzen kann, ist die Universalausgabe TEXTOUT. TEXTOUT benötigt die Anfangsadresse des auszugebenden Textes, diese wird in den Speicherstellen TEXT und TEXT+1 (\$70 und \$71) abgelegt. TEXTOUT

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| 0931 | 08 | A9 | 08 | 20 | B4 | FF | A9 | 6F | 37 | 0AF9 | 8D | AA | 02 | 20 | 82 | 0B | 20 | 9D | BF |
| 0939 | 20 | 96 | FF | 20 | A5 | FF | C9 | 31 | A7 | 0B01 | 0B | A2 | 02 | 20 | C9 | FF | A0 | 00 | B6 |
| 0941 | 08 | 90 | 10 | 48 | 20 | 8D | 08 | 20 | BD | 0B09 | B1 | 76 | 20 | D2 | FF | 90 | 03 | 4C | B0 |
| 0949 | 8D | 08 | A9 | 02 | 8D | 20 | D0 | 68 | FF | 0B11 | 32 | 09 | C8 | D0 | F3 | 20 | CC | FF | C0 |
| 0951 | 20 | D2 | FF | 20 | A5 | FF | C9 | 0D | 8F | 0B19 | 20 | CF | 0B | AD | AA | 02 | 20 | E6 | 92 |
| 0959 | F0 | 0A | 28 | 08 | 90 | F5 | 20 | D2 | EF | 0B21 | 0B | 20 | AE | FF | 20 | 32 | 09 | A0 | F2 |
| 0961 | FF | 4C | 54 | 09 | 20 | AB | FF | A9 | CB | 0B29 | 01 | B1 | 76 | C9 | FF | F0 | AF | 8D | 2F |
| 0969 | 00 | 85 | 90 | 28 | B0 | 01 | 60 | 20 | D2 | 0B31 | AA | 02 | E6 | 77 | 4C | 02 | 0B | A9 | C2 |
| 0971 | 76 | 09 | 4C | 84 | 09 | 20 | E4 | FF | 02 | 0B39 | 04 | 8D | AA | 02 | A0 | 00 | A9 | 12 | 55 |
| 0979 | F0 | FB | 60 | 20 | 76 | 09 | C9 | 5F | 0B | 0B41 | 91 | 76 | C8 | AD | AA | 02 | 91 | 76 | F8 |
| 0981 | F0 | 01 | 60 | 20 | CC | FF | A9 | 02 | A2 | 0B49 | 98 | 18 | 69 | 1F | A8 | B0 | 0A | A9 | 23 |
| 0989 | 20 | C3 | FF | 4C | 10 | 08 | 20 | 65 | D3 | 0B51 | 00 | 91 | 76 | C8 | 91 | 76 | 4C | 49 | 3C |
| 0991 | 0A | 20 | C8 | 0F | 20 | 7E | 0A | 20 | 2F | 0B59 | 0B | AD | AA | 02 | 18 | 69 | 03 | C9 | F0 |
| 0999 | A1 | 0D | A9 | 00 | 8D | A8 | 02 | A9 | 7A | 0B61 | 13 | 90 | 02 | E9 | 11 | 8D | AA | 02 | CA |
| 09A1 | 50 | 8D | A9 | 02 | 20 | E1 | 0C | 20 | 96 | 0B69 | E6 | 77 | A5 | 72 | C5 | 76 | A5 | 73 | 22 |
| 09A9 | 6F | 0A | AD | A8 | 02 | 85 | 74 | AD | AE | 0B71 | E5 | 77 | B0 | C8 | C6 | 77 | A0 | 00 | 44 |
| 09B1 | A9 | 02 | 85 | 75 | 18 | A2 | 04 | A0 | 91 | 0B79 | 98 | 91 | 76 | C8 | A9 | FF | 91 | 76 | 7B |
| 09B9 | 00 | 20 | F0 | FF | 20 | DF | 0D | 20 | 00 | 0B81 | 60 | A9 | 02 | A2 | 08 | A8 | 20 | BA | DF |
| 09C1 | 7C | 09 | F0 | FB | C9 | 91 | D0 | 03 | 6C | 0B89 | FF | A9 | 23 | 85 | FF | A9 | 01 | A2 | A9 |
| 09C9 | 4C | ED | 0D | C9 | 11 | D0 | 03 | 4C | 9E | 0B91 | FF | A0 | 00 | 20 | BD | FF | 20 | C0 | 5D |
| 09D1 | 2B | 0E | C9 | 86 | D0 | 2B | A9 | 00 | B0 | 0B99 | FF | 4C | 32 | 09 | A9 | 08 | 20 | B1 | EB |
| 09D9 | C5 | 72 | A9 | 50 | E5 | 73 | B0 | DF | EE | 0BA1 | FF | A9 | 6F | 20 | 93 | FF | A0 | 00 | D9 |
| 09E1 | 20 | DF | 0D | A5 | 74 | C5 | 72 | A5 | 27 | 0BA9 | B9 | F7 | 0F | F0 | 06 | 20 | A8 | FF | 71 |
| 09E9 | 75 | E5 | 73 | 90 | 05 | A9 | 91 | 20 | 41 | 0BB1 | C8 | D0 | F5 | 4C | AE | FF | A9 | 08 | F1 |
| 09F1 | D2 | FF | 20 | 31 | 0D | 20 | E1 | 0C | 84 | 0BB9 | 20 | B1 | FF | A9 | 6F | 20 | 93 | FF | A8 |
| 09F9 | 20 | DF | 0D | 4C | C0 | 09 | C9 | 87 | FB | 0BC1 | A0 | 00 | B9 | E1 | 0F | F0 | 06 | 20 | 7E |
| 0A01 | D0 | 0F | 20 | DF | 0D | 20 | 6A | 0E | 4B | 0BC9 | A8 | FF | C8 | D0 | F5 | 60 | A9 | 08 | D7 |
| 0A09 | 20 | E1 | 0C | 20 | DF | 0D | 4C | C0 | 89 | 0BD1 | 20 | B1 | FF | A9 | 6F | 20 | 93 | FF | A8 |
| 0A11 | 09 | C9 | 88 | D0 | 03 | 4C | C2 | 0E | 64 | 0BD9 | A0 | 00 | B9 | EC | 0F | F0 | EE | 20 | 5A |
| 0A19 | C9 | 13 | D0 | 06 | 20 | DF | 0D | 4C | 82 | 0BE1 | A8 | FF | C8 | D0 | F5 | A2 | 30 | 38 | EC |
| 0A21 | 9B | 09 | C9 | 4C | D0 | 21 | 20 | DF | FE | 0BE9 | E9 | 0A | 90 | 03 | E8 | B0 | F9 | 69 | 32 |
| 0A29 | 0D | A0 | 02 | B1 | 74 | 49 | 40 | 91 | 99 | 0BF1 | 3A | 48 | 8A | F0 | 03 | 20 | A8 | FF | 18 |
| 0A31 | 74 | A5 | 74 | 85 | 76 | A5 | 75 | 85 | 7F | 0BF9 | 68 | 4C | A8 | FF | 8C | AD | 02 | A4 | 03 |
| 0A39 | 77 | 20 | 75 | 0C | A9 | 91 | 20 | D2 | 26 | 0C01 | D3 | 29 | 3F | 91 | D1 | E6 | D3 | AC | 69 |
| 0A41 | FF | 20 | DF | 0D | 4C | C0 | 09 | C9 | 7C | 0C09 | AD | 02 | 60 | A9 | 00 | 85 | 7E | 85 | D6 |
| 0A49 | 5F | D0 | 03 | 4C | 10 | 08 | C9 | 94 | 83 | 0C11 | 7F | 85 | 80 | A2 | 10 | F8 | 06 | B1 | 77 |
| 0A51 | D0 | F2 | 20 | DF | 0D | 20 | B2 | 0F | B0 | 0C19 | 26 | 82 | A5 | 80 | 65 | 80 | 85 | 80 | B3 |
| 0A59 | 20 | 6A | 0E | 20 | E1 | 0C | 20 | DF | BA | 0C21 | A5 | 7F | 65 | 7F | 85 | 7F | A5 | 7E | 39 |
| 0A61 | 0D | 4C | C0 | 09 | 20 | 74 | 08 | A0 | 8C | 0C29 | 65 | 7E | 85 | 7E | D8 | CA | D0 | E5 | E7 |
| 0A69 | 11 | A9 | 26 | 4C | 50 | 08 | A0 | 00 | 2A | 0C31 | A5 | 7E | 09 | 30 | 9D | 40 | 03 | A5 | BB |
| 0A71 | A2 | 17 | 18 | 20 | F0 | FF | A0 | 11 | 02 | 0C39 | 7F | 20 | 5E | 0C | A5 | 80 | 20 | 5E | 98 |
| 0A79 | A9 | 53 | 4C | 50 | 08 | 20 | 82 | 0B | B1 | 0C41 | 0C | A2 | 00 | BD | 40 | 03 | C9 | 30 | 84 |
| 0A81 | 20 | B7 | 0B | A9 | 31 | 20 | A8 | FF | 86 | 0C49 | D0 | 05 | E8 | E0 | 05 | 90 | F4 | 20 | 1B |
| 0A89 | 20 | AE | FF | 20 | 32 | 09 | A9 | 00 | 22 | 0C51 | D2 | FF | E8 | E0 | 05 | B0 | 05 | BD | 79 |
| 0A91 | 85 | 76 | A9 | 50 | 85 | 77 | A2 | 02 | 15 | 0C59 | 40 | 03 | D0 | F3 | 60 | 48 | 20 | 68 | EA |
| 0A99 | 20 | C6 | FF | A0 | 00 | 20 | CF | FF | 69 | 0C61 | 0C | 68 | 29 | 0F | 4C | 6E | 0C | 29 | 80 |
| 0AA1 | 90 | 03 | 4C | 32 | 09 | 91 | 76 | C8 | CD | 0C69 | F0 | 4A | 4A | 4A | 4A | 09 | 30 | E8 | 2E |
| 0AA9 | D0 | F3 | 20 | CC | FF | B1 | 76 | F0 | 06 | 0C71 | 9D | 40 | 03 | 60 | A9 | 00 | 85 | D3 | 5A |
| 0AB1 | 12 | 20 | B7 | 0B | A0 | 01 | B1 | 76 | 77 | 0C79 | A2 | 06 | 20 | 95 | 08 | A0 | 1E | B1 | FD |
| 0AB9 | 20 | E6 | 0B | 20 | AE | FF | E6 | 77 | 44 | 0C81 | 76 | 85 | 81 | C8 | B1 | 76 | 85 | 82 | EB |
| 0AC1 | 4C | 97 | 0A | A0 | 03 | B1 | 76 | F0 | F9 | 0C89 | D0 | 12 | A5 | 81 | C9 | 64 | B0 | 0C | EA |
| 0AC9 | 09 | 98 | 18 | 69 | 20 | A8 | 90 | F5 | 34 | 0C91 | 20 | 90 | 08 | A5 | 81 | C9 | 0A | B0 | 07 |
| 0AD1 | E6 | 77 | 98 | 38 | E9 | 23 | 85 | 72 | A6 | 0C99 | 03 | 20 | 90 | 08 | 20 | 0C | 0C | A2 | E9 |
| 0AD9 | A5 | 77 | E9 | 00 | 85 | 73 | A9 | 02 | CF | 0CA1 | 02 | 20 | 95 | 08 | A0 | 05 | B1 | 76 | 44 |
| 0AE1 | 4C | C3 | FF | A9 | 00 | 85 | 76 | A9 | 28 | 0CA9 | 20 | FD | 0B | C8 | C0 | 15 | 90 | F6 | AC |
| 0AE9 | 50 | 85 | 77 | 20 | 38 | 0B | A9 | 00 | FB | 0CB1 | 20 | 90 | 08 | A0 | 02 | B1 | 76 | 48 | 79 |
| 0AF1 | 85 | 76 | A9 | 50 | 85 | 77 | A9 | 01 | 1D | 0CB9 | 29 | 0F | AA | CA | BD | FF | 0F | 20 | AB |

Floppy

| | | | | | |
|------|-------------------------|----|------|-------------------------|----|
| 0CC1 | FD 0B BD 02 10 20 FD 0B | 8D | 0E89 | A0 1F B1 78 91 7A 88 10 | 48 |
| 0CC9 | BD 05 10 20 FD 0B 68 29 | D7 | 0E91 | F9 38 A5 78 85 7A E9 20 | C1 |
| 0CD1 | 40 F0 05 A9 3E 20 FD 0B | BD | 0E99 | 85 78 A5 79 85 7B E9 00 | 2B |
| 0CD9 | A2 02 20 95 08 4C 8D 08 | 6C | 0EA1 | 85 79 A5 78 C5 74 A5 79 | B5 |
| 0CE1 | 38 20 F0 FF 8E AE 02 8C | 60 | 0EA9 | E5 75 B0 DC 38 A5 7C E9 | 6A |
| 0CE9 | AD 02 18 A2 04 A0 00 20 | 66 | 0EB1 | 20 85 7C B0 02 C6 7D A0 | 81 |
| 0CF1 | F0 FF AD A8 02 85 76 AD | 3D | 0EB9 | 1F B1 7C 91 74 88 10 F9 | 5D |
| 0CF9 | A9 02 85 77 A9 12 8D AF | 29 | 0EC1 | 60 20 D0 0E B0 03 20 2E | A0 |
| 0D01 | 02 20 75 0C CE AF 02 F0 | 7C | 0EC9 | 0F 20 E4 0A 4C 10 08 A9 | 63 |
| 0D09 | 1D A5 76 C5 72 A5 77 E5 | 89 | 0ED1 | 00 85 76 C5 7C A9 60 85 | DB |
| 0D11 | 73 B0 0E A5 76 18 69 20 | D4 | 0ED9 | 77 E5 7D 90 01 60 20 65 | DC |
| 0D19 | 85 76 90 02 E6 77 4C 02 | 55 | 0EE1 | 0A A0 04 A2 17 18 20 F0 | FD |
| 0D21 | 0D A2 27 20 95 08 18 AE | 69 | 0EE9 | FF A0 00 B9 A7 11 F0 06 | 25 |
| 0D29 | AE 02 AC AD 02 4C F0 FF | 51 | 0EF1 | 20 D2 FF C8 D0 F5 A5 72 | 93 |
| 0D31 | A0 1F B1 74 91 7C 88 10 | 4A | 0EF9 | 8D AB 02 A5 73 8D AC 02 | D0 |
| 0D39 | F9 A5 7C 18 69 20 85 7C | 09 | 0F01 | 38 A5 7C E9 20 85 72 A5 | 8F |
| 0D41 | 90 02 E6 7D A5 74 85 78 | 34 | 0F09 | 7D E9 00 85 73 A2 04 8E | 58 |
| 0D49 | A5 75 85 79 A5 78 C5 72 | BB | 0F11 | AE 02 A0 00 8C AD 02 18 | 14 |
| 0D51 | A5 79 E5 73 B0 21 A5 78 | 59 | 0F19 | 20 F0 FF 20 FD 0C AD AB | 01 |
| 0D59 | 18 69 20 85 7A A5 79 69 | 30 | 0F21 | 02 85 72 AD AC 02 85 73 | FE |
| 0D61 | 00 85 7B A0 1F B1 7A 91 | AE | 0F29 | 20 DF 0D 18 60 A9 00 85 | 6A |
| 0D69 | 78 88 10 F9 A5 7A 85 78 | 40 | 0F31 | 74 A9 60 85 75 20 7C 09 | 6E |
| 0D71 | A5 7B 85 79 4C 4D 0D A0 | A2 | 0F39 | C9 53 F0 3A C9 0D D0 F5 | C1 |
| 0D79 | 1F A9 00 91 78 88 10 FB | EC | 0F41 | 18 A5 72 69 20 85 72 90 | FA |
| 0D81 | 38 A5 78 E9 20 85 72 A5 | 8B | 0F49 | 02 E6 73 A0 1F B1 74 91 | 42 |
| 0D89 | 79 E9 00 85 73 A5 72 C5 | 87 | 0F51 | 72 88 10 F9 18 A5 74 69 | 81 |
| 0D91 | 74 A5 73 E5 75 B0 08 A5 | 86 | 0F59 | 20 85 74 90 02 E6 75 C5 | 4C |
| 0D99 | 72 85 74 A5 73 85 75 60 | BF | 0F61 | 7C A5 75 E5 7D 90 01 60 | F6 |
| 0DA1 | A9 00 85 74 A9 50 85 75 | 22 | 0F69 | 20 DF 0D A9 11 20 D2 FF | 6A |
| 0DA9 | A9 00 85 7C A9 60 85 7D | 02 | 0F71 | 20 DF 0D 4C 36 0F A0 05 | C5 |
| 0DB1 | A0 02 B1 74 29 0F D0 06 | BE | 0F79 | B1 74 99 38 03 C8 C0 15 | 81 |
| 0DB9 | 20 31 0D 4C B1 0D A5 74 | 85 | 0F81 | 90 F6 A9 53 8D 3D 03 A9 | 9A |
| 0DC1 | C5 72 A5 75 E5 73 B0 0D | 97 | 0F89 | 30 8D 3E 03 A9 3A 8D 3F | 9B |
| 0DC9 | A5 74 69 20 85 74 90 02 | 18 | 0F91 | 03 A9 08 20 B1 FF A9 6F | 2F |
| 0DD1 | E6 75 4C B1 0D A9 00 85 | EC | 0F99 | 20 93 FF A0 00 B9 3D 03 | 6E |
| 0DD9 | 7C A9 60 85 7D 60 A0 04 | 67 | 0FA1 | 20 A8 FF C8 C0 13 90 F5 | F7 |
| 0DE1 | B1 D1 49 80 91 D1 C8 C0 | 71 | 0FA9 | 20 AE FF 20 32 09 4C 55 | 71 |
| 0DE9 | 22 90 F5 60 A5 75 C9 50 | CF | 0FB1 | 0F A0 1F B9 ED 11 91 7C | C6 |
| 0DF1 | D0 09 A5 74 C9 00 D0 03 | 8C | 0FB9 | 88 10 F8 18 A5 7C 69 20 | C9 |
| 0DF9 | 4C C0 09 38 A5 74 E9 20 | 57 | 0FC1 | 85 7C 90 02 E6 7D 60 A9 | B6 |
| 0E01 | 85 74 B0 02 C6 75 20 DF | 50 | 0FC9 | 00 A2 50 85 74 86 75 A9 | E4 |
| 0E09 | 0D A5 D6 C9 04 F0 0B A9 | ED | 0FD1 | 00 A8 91 74 C8 D0 FB E6 | 82 |
| 0E11 | 91 20 D2 FF 20 DF 0D 4C | 47 | 0FD9 | 75 A6 75 E0 60 90 F3 60 | C7 |
| 0E19 | C0 09 A5 74 8D A8 02 A5 | 2A | 0FE1 | 55 31 20 32 20 30 20 31 | F1 |
| 0E21 | 75 8D A9 02 20 E1 0C 4C | 8F | 0FE9 | 38 20 00 55 32 20 32 20 | E7 |
| 0E29 | AB 09 A5 74 C5 72 A5 75 | 55 | 0FF1 | 30 20 31 38 20 00 42 2D | 3E |
| 0E31 | E5 73 90 03 4C C0 09 A5 | EF | 0FF9 | 50 20 32 20 30 00 53 50 | 75 |
| 0E39 | 74 69 20 85 74 90 02 E6 | A7 | 1001 | 55 45 52 53 51 47 52 0D | 5E |
| 0E41 | 75 20 DF 0D A5 D6 C9 15 | A9 | 1009 | 0D 0D 06 23 23 23 23 23 | E3 |
| 0E49 | B0 0B A9 11 20 D2 FF 20 | 69 | 1011 | 23 23 23 23 23 23 23 23 | 00 |
| 0E51 | DF 0D 4C C0 09 18 AD A8 | 54 | 1019 | 23 23 23 23 23 23 23 23 | 00 |
| 0E59 | 02 69 20 8D A8 02 90 03 | 5F | 1021 | 23 23 23 23 23 23 0D 06 | 07 |
| 0E61 | EE A9 02 20 E1 0C 4C 50 | F7 | 1029 | 23 0A 0A 05 23 0D 06 23 | 17 |
| 0E69 | 0E A9 00 C5 7C A9 60 E5 | F0 | 1031 | 06 43 20 41 20 54 20 53 | 3C |
| 0E71 | 7D 90 01 60 A5 72 85 78 | CE | 1039 | 20 4F 20 52 20 54 06 23 | 4F |
| 0E79 | 69 20 85 72 85 7A A5 73 | 98 | 1041 | 0D 06 23 20 20 42 59 20 | 21 |
| 0E81 | 85 79 69 00 85 73 85 7B | 90 | 1049 | 4E 2E 4D 41 4E 4E 20 26 | 26 |

Floppy

| | | | | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| 1051 | 20 | 44 | 2E | 57 | 45 | 49 | 4E | 45 | B9 | 1131 | 54 | 20 | 53 | 20 | 4F | 20 | 52 | 20 | CB |
| 1059 | 43 | 4B | 20 | 20 | 23 | 0D | 06 | 23 | F1 | 1139 | 54 | 20 | 20 | 2A | 2A | 2A | 2A | 0D | 47 |
| 1061 | 0A | 0A | 05 | 23 | 0D | 06 | 23 | 23 | E9 | 1141 | 0D | 12 | 06 | 42 | 4C | 4B | 06 | 4E | 79 |
| 1069 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 00 | 1149 | 41 | 4D | 45 | 09 | 54 | 59 | 50 | 09 | 72 |
| 1071 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 00 | 1151 | 0D | 00 | 81 | 12 | 20 | 46 | 33 | 20 | 69 |
| 1079 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 23 | 00 | 1159 | 92 | 9F | 20 | 20 | 2D | 3E | 20 | 50 | B2 |
| 1081 | 23 | 0D | 0D | 0D | 20 | 20 | 4C | 45 | 1D | 1161 | 55 | 46 | 46 | 45 | 52 | 20 | 20 | 20 | 42 |
| 1089 | 47 | 45 | 4E | 20 | 53 | 49 | 45 | 20 | 5F | 1169 | 20 | 20 | 81 | 12 | 20 | 46 | 37 | 20 | 60 |
| 1091 | 44 | 49 | 45 | 20 | 52 | 49 | 43 | 48 | 24 | 1171 | 92 | 9F | 20 | 20 | 2D | 3E | 20 | 44 | BE |
| 1099 | 54 | 49 | 47 | 45 | 20 | 44 | 49 | 53 | DF | 1179 | 49 | 53 | 48 | 45 | 54 | 54 | 45 | 0D | 34 |
| 10A1 | 4B | 45 | 54 | 54 | 45 | 20 | 45 | 49 | 27 | 1181 | 81 | 12 | 20 | 46 | 35 | 20 | 92 | 9F | 51 |
| 10A9 | 4E | 0D | 0D | 07 | 55 | 4E | 44 | 20 | 72 | 1189 | 20 | 20 | 50 | 55 | 46 | 46 | 45 | 52 | EE |
| 10B1 | 44 | 52 | 55 | 45 | 43 | 4B | 45 | 4E | F1 | 1191 | 20 | 2D | 3E | 20 | 20 | 20 | 02 | 81 | 92 |
| 10B9 | 20 | 53 | 49 | 45 | 20 | 12 | 20 | 52 | AD | 1199 | 12 | 02 | 5F | 20 | 92 | 9F | 20 | 20 | 42 |
| 10C1 | 45 | 54 | 55 | 52 | 4E | 20 | 92 | 00 | B4 | 11A1 | 4D | 45 | 4E | 55 | 45 | 00 | 52 | 45 | 53 |
| 10C9 | 0D | 0D | 0A | 42 | 49 | 54 | 54 | 45 | CC | 11A9 | 53 | 54 | 4C | 49 | 43 | 48 | 45 | 20 | 22 |
| 10D1 | 20 | 57 | 41 | 45 | 48 | 4C | 45 | 4E | B8 | 11B1 | 50 | 52 | 4F | 47 | 52 | 41 | 4D | 4D | 17 |
| 10D9 | 20 | 53 | 49 | 45 | 20 | 3A | 0D | 0D | B8 | 11B9 | 45 | 20 | 49 | 4D | 20 | 50 | 55 | 46 | 00 |
| 10E1 | 09 | 31 | 2E | 20 | 44 | 49 | 52 | 45 | EE | 11C1 | 46 | 45 | 52 | 0D | 20 | 20 | 20 | 20 | 46 |
| 10E9 | 43 | 54 | 4F | 52 | 59 | 20 | 41 | 4E | 18 | 11C9 | 53 | 43 | 52 | 41 | 54 | 43 | 48 | 20 | 5A |
| 10F1 | 5A | 45 | 49 | 47 | 45 | 4E | 0D | 09 | 12 | 11D1 | 3A | 20 | 81 | 12 | 20 | 53 | 20 | 92 | E4 |
| 10F9 | 32 | 2E | 20 | 44 | 49 | 52 | 45 | 43 | D9 | 11D9 | 9F | 20 | 20 | 2D | 2D | 20 | 20 | 53 | 4C |
| 1101 | 54 | 4F | 52 | 59 | 20 | 53 | 4F | 52 | C8 | 11E1 | 4F | 4E | 53 | 54 | 20 | 52 | 45 | 54 | BF |
| 1109 | 54 | 49 | 45 | 52 | 45 | 4E | 0D | 09 | F9 | 11E9 | 55 | 52 | 4E | 00 | 00 | 00 | 83 | 12 | C2 |
| 1111 | 33 | 2E | 20 | 50 | 52 | 4F | 47 | 52 | CD | 11F1 | 01 | 2D | 2D | 2D | 2D | 2D | 2D | 2D | D4 |
| 1119 | 41 | 4D | 4D | 20 | 42 | 45 | 45 | 4E | 15 | 11F9 | 2D | 2D | 2D | 2D | 2D | 2D | 2D | 2D | 00 |
| 1121 | 44 | 45 | 4E | 0D | 00 | 06 | 2A | 2A | 3A | 1201 | 2D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 2D |
| 1129 | 2A | 2A | 20 | 20 | 43 | 20 | 41 | 20 | 44 | 1209 | 00 | 00 | 00 | 00 | 0D | 0D | 0D | 0D | 00 |

Hexdump mit Checker eingeben

| | | | |
|---|------|---|------|
| 10 FOR P=0 TO 1:FOR Z=100 TO 260:5=0 | 2293 | 113 DATA 9,169,8,32,180,255,169,96,32, | 6142 |
| 20 FOR B=1 TO 16:READ D:IF P THEN PRINT# | 2546 | 150,255,32,165,255,32,165,2004 | |
| 1,CHR#(D): | | 114 DATA 255,76,252,8,32,165,255,170,32, | 7269 |
| 30 S=S+D:NEXT B:READ C:IF C<>S THEN | 4042 | 165,255,32,205,189,32,144,2267 | |
| PRINT "DATA ERROR IN ZEILE"Z:END | | 115 DATA 8,32,165,255,240,5,32,210,255, | 6596 |
| 40 NEXT Z:IF P=0 THEN INPUT "FILENAME": | 5349 | 208,246,32,165,255,32,165,2305 | |
| F#:OPEN 1,8,1,F#:PRINT#1,CHR#(1) | | 116 DATA 255,165,144,208,19,32,141,8,173, | 7067 |
| CHR#(8): | | 141,2,201,2,208,213,173,2085 | |
| 50 RESTORE I:NEXT P:CLOSE 1 | 655 | 117 DATA 141,2,201,1,208,249,240,204,32, | 6807 |
| 100 DATA 13,8,10,0,158,40,50,48,54,52,41, | 4956 | 171,255,169,8,32,177,255,2345 | |
| 0,0,0,0,162,836 | | 118 DATA 169,224,32,147,255,32,174,255, | 6210 |
| 101 DATA 255,154,169,6,141,32,208,141,33, | 7155 | 32,50,8,32,118,9,76,16,1630 | |
| 208,169,3,141,134,2,32,1828 | | 119 DATA 8,169,8,32,180,255,169,111,32, | 5977 |
| 102 DATA 116,8,160,16,169,8,32,80,8,32, | 5753 | 150,255,32,165,255,201,49,2071 | |
| 228,255,201,13,208,249,1783 | | 120 DATA 8,144,16,72,32,141,8,32,141,8, | 5725 |
| 103 DATA 160,16,169,201,32,80,8,32,228, | 5648 | 169,2,141,32,208,104,1258 | |
| 255,201,49,208,3,76,156,1874 | | 121 DATA 32,210,255,32,165,255,201,13, | 6877 |
| 104 DATA 8,201,50,208,3,76,143,9,201,51, | 5933 | 240,10,40,8,144,245,32,210,2092 | |
| 208,235,76,226,252,132,2079 | | 122 DATA 255,76,84,9,32,171,255,169,0, | 6534 |
| 105 DATA 113,133,112,160,0,177,112,240, | 6889 | 133,144,40,176,1,96,32,1673 | |
| 25,201,13,176,5,170,202,32,1871 | | 123 DATA 118,9,76,132,9,32,228,255,240, | 5821 |
| 106 DATA 149,8,201,21,208,2,169,5,32,210, | 6722 | 251,96,32,118,9,201,95,1981 | |
| 255,208,208,231,230,113,2242 | | 124 DATA 240,1,96,32,204,255,169,2,32, | 6705 |
| 107 DATA 208,227,96,169,147,32,210,255, | 6848 | 195,255,76,16,8,32,101,1714 | |
| 162,0,169,3,157,0,216,157,2208 | | 125 DATA 10,32,200,15,32,126,10,32,161, | 5695 |
| 108 DATA 0,217,157,0,218,157,0,219,232, | 6244 | 13,169,0,141,168,2,169,1280 | |
| 208,241,96,169,13,44,169,2140 | | 126 DATA 80,141,169,2,32,225,12,32,111, | 6356 |
| 109 DATA 32,76,210,255,32,144,8,202,208, | 5587 | 18,173,168,2,133,116,173,1579 | |
| 250,96,32,116,8,169,0,1838 | | 127 DATA 169,2,133,117,24,162,4,160,0,32, | 6148 |
| 110 DATA 133,144,169,8,32,177,255,169, | 7276 | 240,255,32,223,13,32,1598 | |
| 111,32,147,255,169,73,32,168,2074 | | 128 DATA 124,9,240,251,201,145,208,3,76, | 5926 |
| 111 DATA 255,32,174,255,165,144,48,112, | 6583 | 237,13,201,17,208,3,76,2012 | |
| 169,8,32,177,255,169,240,32,2267 | | 129 DATA 43,14,201,134,208,40,169,0,197, | 5826 |
| 112 DATA 147,255,169,36,32,168,255,32, | 6864 | 114,169,80,229,115,176,223,2112 | |
| 174,255,165,144,16,3,76,25,1952 | | 130 DATA 32,223,13,165,116,197,114,165, | 6833 |

Floppy

| | | | |
|---|------|--|------|
| 117,229,115,144,5,169,145,32,1991 | | | |
| 131 DATA 210,255,32,49,13,32,225,12,32,223,13,76,192,9,201,135,1709 | 6208 | 170 DATA 12,104,41,15,76,110,12,41,240,74,74,74,74,9,48,232,1236 | 5717 |
| 132 DATA 208,15,32,223,13,32,106,14,32,225,12,32,223,13,76,192,1440 | 6139 | 171 DATA 157,64,3,96,169,0,133,211,162,6,32,149,8,160,30,177,1557 | 6077 |
| 133 DATA 9,201,136,208,3,76,194,14,201,19,208,6,32,223,13,76,1619 | 5453 | 172 DATA 118,133,129,200,177,118,133,130,208,18,165,129,201,100,176,12,2147 | 6812 |
| 134 DATA 155,9,201,76,208,33,32,223,13,160,2,177,116,73,64,145,1687 | 6957 | 173 DATA 32,144,8,165,129,201,10,176,3,32,144,8,32,12,12,162,1270 | 5818 |
| 135 DATA 116,165,116,133,118,165,117,133,119,32,117,12,169,145,32,210,1899 | 7888 | 174 DATA 2,32,149,8,160,5,177,118,32,253,11,200,192,21,144,246,1750 | 5996 |
| 136 DATA 253,32,223,13,76,192,9,201,95,208,3,76,16,8,201,148,1756 | 6621 | 175 DATA 32,144,8,160,2,177,118,72,41,15,170,202,189,255,15,32,1632 | 5921 |
| 137 DATA 208,242,32,223,13,32,178,15,32,106,14,32,225,12,32,223,1619 | 6441 | 176 DATA 253,11,189,2,18,32,253,11,189,5,16,32,253,11,104,41,1418 | 5618 |
| 138 DATA 13,76,192,9,32,116,8,160,17,169,38,76,80,8,160,0,1154 | 6884 | 177 DATA 64,240,5,169,62,32,253,11,162,2,32,149,8,76,141,8,1414 | 5463 |
| 139 DATA 162,23,24,32,240,255,160,17,169,83,76,80,8,32,130,11,1502 | 6307 | 178 DATA 56,32,240,255,142,174,2,140,173,2,24,162,4,160,0,32,1598 | 5875 |
| 140 DATA 32,183,11,169,49,32,168,255,32,174,255,32,50,9,169,0,1620 | 5658 | 179 DATA 240,255,173,168,2,133,118,173,169,2,133,119,169,18,141,175,2188 | 7049 |
| 141 DATA 133,118,169,80,133,119,162,2,32,198,255,160,0,32,207,255,2055 | 6813 | 180 DATA 2,32,117,12,206,175,2,240,29,165,118,197,114,165,119,229,1922 | 6741 |
| 142 DATA 144,3,76,50,9,145,118,200,208,243,32,204,255,177,118,240,2222 | 6246 | 181 DATA 115,176,14,165,118,224,105,32,133,118,144,2,230,119,76,2,1573 | 6358 |
| 143 DATA 18,32,183,11,160,1,177,118,32,230,11,32,174,255,230,119,1783 | 6496 | 182 DATA 13,162,39,32,149,8,24,174,174,2,172,173,2,76,240,255,1695 | 6215 |
| 144 DATA 76,151,10,160,3,177,118,240,8,152,24,105,32,168,144,245,1814 | 6501 | 183 DATA 160,31,177,118,145,124,136,16,249,165,124,24,105,32,133,124,1861 | 6553 |
| 145 DATA 230,119,152,56,233,35,133,114,165,119,233,0,133,115,169,2,2008 | 6803 | 184 DATA 144,2,230,125,165,116,133,120,165,117,133,121,165,120,197,114,2167 | 6555 |
| 146 DATA 76,195,255,169,0,133,118,169,80,133,119,32,56,11,169,0,1715 | 6423 | 185 DATA 185,121,229,115,176,33,165,120,24,105,32,133,122,165,121,105,1931 | 7254 |
| 147 DATA 133,118,169,80,133,119,169,1,141,170,2,32,130,11,32,157,1597 | 6181 | 186 DATA 0,133,123,160,31,177,122,145,120,136,18,249,165,122,133,120,1952 | 7230 |
| 148 DATA 11,162,2,32,201,255,160,0,177,118,32,210,255,144,3,76,1838 | 6771 | 187 DATA 165,123,133,121,76,77,13,160,31,169,0,145,120,136,16,251,1736 | 7583 |
| 149 DATA 50,9,200,208,243,32,204,255,32,207,11,173,170,2,32,230,2058 | 6113 | 188 DATA 56,165,120,233,32,133,114,165,121,233,0,133,115,165,114,197,2096 | 6861 |
| 150 DATA 11,32,174,255,32,50,9,160,1,177,118,201,255,240,175,141,2031 | 6850 | 189 DATA 118,165,115,229,117,176,8,165,114,133,116,165,115,133,117,96,2080 | 7628 |
| 151 DATA 170,2,230,119,76,2,11,169,4,141,170,2,160,0,169,18,1443 | 5807 | 190 DATA 169,0,133,116,169,80,133,117,169,0,133,124,169,96,133,125,1866 | 6865 |
| 152 DATA 145,118,200,173,170,2,145,118,152,24,105,31,168,176,18,169,1986 | 6816 | 191 DATA 160,2,177,116,41,15,208,6,32,49,13,76,177,13,165,116,1366 | 6138 |
| 153 DATA 0,145,118,200,145,118,76,73,11,173,170,2,24,105,3,201,1564 | 5778 | 192 DATA 197,114,165,117,229,115,176,13,165,116,105,32,133,116,144,2,1939 | 7237 |
| 154 DATA 19,144,2,233,17,141,170,2,230,119,165,114,197,118,165,115,1951 | 6950 | 193 DATA 230,117,76,177,13,169,0,133,124,169,96,133,125,96,160,4,1822 | 6265 |
| 155 DATA 229,119,176,200,198,119,160,0,152,145,118,200,169,255,145,118,2503 | 7115 | 194 DATA 177,209,73,128,145,209,200,192,34,144,245,96,165,117,201,80,2415 | 6337 |
| 156 DATA 96,169,2,162,8,168,32,186,255,169,35,133,255,169,1,162,2002 | 6318 | 195 DATA 208,9,165,116,201,0,208,3,76,192,9,56,165,116,233,32,1789 | 6148 |
| 157 DATA 255,160,0,32,189,255,32,192,255,76,50,9,169,0,32,177,1891 | 6392 | 196 DATA 133,116,176,2,198,117,32,223,13,165,214,201,4,240,11,169,2014 | 6751 |
| 158 DATA 255,169,111,32,147,255,160,0,185,247,15,240,6,32,168,255,2277 | 7447 | 197 DATA 145,32,210,255,32,223,13,76,192,9,165,116,141,168,2,165,1944 | 6693 |
| 159 DATA 200,208,245,76,174,255,169,0,32,177,255,169,111,32,147,255,2513 | 7482 | 198 DATA 117,141,169,2,32,225,12,76,171,9,165,116,197,114,165,117,1828 | 7116 |
| 160 DATA 160,0,185,225,15,240,6,32,168,255,208,208,245,96,169,0,2212 | 5919 | 199 DATA 229,115,144,3,76,192,9,165,116,105,32,133,116,144,2,230,1811 | 6635 |
| 161 DATA 32,177,255,169,111,32,147,255,160,0,185,236,15,240,238,32,2284 | 6261 | 200 DATA 117,32,223,13,165,214,201,21,176,11,169,17,32,210,255,32,1888 | 6487 |
| 162 DATA 160,255,200,208,245,162,48,56,233,10,144,3,232,176,249,105,2494 | 6519 | 201 DATA 223,13,76,192,9,24,173,168,2,105,32,141,168,2,144,3,1475 | 6375 |
| 163 DATA 58,72,138,240,3,32,168,255,104,76,168,255,140,173,2,164,2048 | 5679 | 202 DATA 238,169,2,32,225,12,76,80,14,169,0,197,124,169,96,229,1832 | 6371 |
| 164 DATA 211,41,63,145,209,230,211,172,173,2,96,169,0,133,126,133,2114 | 6483 | 203 DATA 125,144,1,96,165,114,133,120,105,32,133,114,133,122,165,115,1817 | 7483 |
| 165 DATA 127,133,128,162,16,248,6,129,38,130,165,128,181,128,133,128,1900 | 7765 | 204 DATA 133,121,105,0,133,115,133,123,160,31,177,120,145,122,136,16,1770 | 7174 |
| 166 DATA 165,127,101,127,133,127,165,126,101,126,133,126,216,202,208,229,2412 | 8290 | 205 DATA 249,56,165,120,133,122,233,32,133,120,165,121,133,123,233,0,2138 | 6217 |
| 167 DATA 165,126,9,48,157,64,3,165,127,32,94,12,165,128,32,94,1421 | 6373 | 206 DATA 133,121,165,120,197,116,165,121,229,117,176,220,56,165,124,233,2458 | 8104 |
| 168 DATA 12,162,0,189,64,3,201,48,208,5,232,224,5,144,244,32,1773 | 5843 | 207 DATA 32,133,124,176,2,198,125,160,31,177,124,145,116,136,16,249,1944 | 6942 |
| 169 DATA 210,255,232,224,5,176,5,189,64,3,208,243,96,72,32,104,2118 | 6945 | 208 DATA 96,32,208,14,176,3,32,46,15,32,170,104,41,15,76,110,12,41,240,74,74,74,74,9,48,232,1236 | 5717 |

| | | | |
|---|------|--|--|
| 228,10,76,16,8,169,1161 | | | |
| 209 DATA 0,133,118,197,124,169,96,133,119,229,125,144,1,96,32,101,1017 | 6749 | | |
| 210 DATA 10,160,4,162,23,24,32,240,255,160,0,105,167,17,240,6,1605 | 6027 | | |
| 211 DATA 32,210,255,200,200,245,165,114,141,171,2,165,115,141,172,2,2330 | 5903 | | |
| 212 DATA 56,165,124,233,32,133,114,165,125,233,0,133,115,162,4,142,1936 | 6032 | | |
| 213 DATA 174,2,160,0,140,173,2,24,32,240,255,32,253,12,173,171,1043 | 6200 | | |
| 214 DATA 2,133,114,173,172,2,133,115,32,223,13,24,96,169,0,133,1534 | 6550 | | |
| 215 DATA 116,169,96,133,117,32,124,9,201,83,240,58,201,13,200,245,2045 | 6607 | | |
| 216 DATA 24,165,114,105,32,133,114,144,2,230,115,160,31,177,116,145,1007 | 6002 | | |
| 217 DATA 114,136,16,249,24,165,116,105,32,133,116,144,2,230,117,197,1096 | 6567 | | |
| 218 DATA 124,165,117,229,125,144,1,96,32,223,13,169,17,32,210,255,1952 | 7400 | | |
| 219 DATA 32,223,13,76,54,15,160,5,177,116,153,59,3,200,192,21,1499 | 5074 | | |
| 220 DATA 144,246,169,83,141,61,3,169,48,141,62,3,169,50,141,63,1701 | 6449 | | |
| 221 DATA 3,169,8,32,177,255,169,111,32,147,255,160,0,105,61,3,1767 | 5922 | | |
| 222 DATA 32,168,255,200,192,19,144,245,32,174,255,32,50,9,76,05,1960 | 6109 | | |
| 223 DATA 15,160,31,185,237,17,145,124,136,16,240,24,165,124,105,32,1764 | 6272 | | |
| 224 DATA 133,124,144,2,230,125,96,169,0,162,00,133,116,134,117,169,1934 | 6675 | | |
| 225 DATA 0,160,145,116,200,200,251,230,117,168,117,224,96,144,243,96,2521 | 6300 | | |
| 226 DATA 85,49,32,50,32,40,32,49,56,32,0,85,50,32,50,32,714 | 5224 | | |
| 227 DATA 40,32,49,56,32,0,66,45,00,32,50,32,40,0,0,0,0,733 | 4077 | | |
| 228 DATA 85,69,82,03,01,71,82,13,13,13,6,35,35,35,35,35,773 | 5170 | | |
| 229 DATA 35,35,35,35,35,35,35,35,35,35,35,35,35,35,35,35,560 | 6192 | | |
| 230 DATA 35,35,35,35,35,35,13,6,35,10,10,5,35,13,6,35,378 | 5391 | | |
| 231 DATA 6,67,32,65,32,84,32,03,32,79,32,82,32,04,6,35,783 | 5234 | | |
| 232 DATA 13,6,35,32,32,66,89,32,70,46,77,65,78,70,32,30,797 | 5006 | | |
| 233 DATA 32,60,46,07,69,73,70,69,67,75,32,32,35,13,6,35,017 | 5064 | | |
| 234 DATA 10,10,5,35,13,6,35,35,35,35,35,35,35,35,35,35,429 | 5000 | | |
| 235 DATA 35,35,35,35,35,35,35,35,35,35,35,35,35,35,35,35,560 | 6192 | | |
| 236 DATA 35,13,13,13,32,32,76,69,71,69,78,32,03,73,69,32,790 | 6014 | | |
| 237 DATA 60,73,69,32,82,73,67,72,84,73,71,69,32,00,73,03,1009 | 6068 | | |
| 238 DATA 75,69,84,84,69,32,69,73,70,13,13,7,05,70,68,32,929 | 5960 | | |
| 239 DATA 60,02,85,69,67,75,69,70,32,03,73,69,32,10,32,02,1014 | 6095 | | |
| 240 DATA 69,04,85,02,70,32,146,0,13,13,10,66,73,04,04,69,900 | 6109 | | |
| 241 DATA 32,07,65,69,72,76,69,78,32,03,73,69,32,50,13,13,921 | 6344 | | |
| 242 DATA 9,49,46,32,60,73,82,69,67,04,79,82,09,32,05,70,1004 | 5979 | | |
| 243 DATA 90,69,73,71,69,78,13,9,50,46,32,60,73,82,69,67,959 | 5031 | | |
| 244 DATA 04,79,82,09,32,83,79,82,04,73,69,82,69,70,13,9,1007 | 5640 | | |
| 245 DATA 51,46,32,00,82,79,71,82,65,77,77,32,66,69,69,70,1056 | 6189 | | |
| 246 DATA 60,69,70,13,0,6,42,42,42,32,32,07,32,65,32,862 | 4910 | | |
| 247 DATA 04,32,03,32,79,32,82,32,04,32,32,42,42,42,13,705 | 5797 | | |
| 248 DATA 13,18,6,66,76,75,6,70,65,77,69,9,04,89,00,9,820 | 4915 | | |
| 249 DATA 13,0,129,10,32,70,51,32,146,159,32,32,45,02,32,00,933 | 5071 | | |
| 250 DATA 05,70,70,69,82,32,32,32,32,32,129,10,32,70,55,32,072 | 5767 | | |
| 251 DATA 146,159,32,32,45,62,32,60,73,03,75,69,04,04,69,13,1126 | 5069 | | |
| 252 DATA 129,10,32,70,53,32,146,159,32,32,00,85,70,70,69,82,1159 | 5779 | | |
| 253 DATA 32,45,62,32,32,32,2,129,10,2,95,32,146,159,32,32,802 | 5723 | | |
| 254 DATA 77,69,70,85,69,0,82,69,03,04,76,73,67,72,69,32,1005 | 6348 | | |
| 255 DATA 00,82,79,71,82,65,77,77,69,32,73,77,32,00,85,70,1131 | 6135 | | |
| 256 DATA 70,69,82,13,32,32,32,32,03,67,82,65,04,67,72,32,914 | 5550 | | |
| 257 DATA 50,32,129,10,32,83,32,146,159,32,32,32,45,45,32,32,03,990 | 5765 | | |
| 258 DATA 79,70,83,04,32,02,69,04,05,02,70,0,0,0,131,10,905 | 4772 | | |
| 259 DATA 1,45,45,45,45,45,45,45,45,45,45,45,45,45,45,45,551 | 5551 | | |
| 260 DATA 45,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,45 | 3012 | | |

Basic-Lader

| | | | | | |
|------|---------------------------------------|------|---------------------|-----------|--------------------|
| 7000 | ***** | 007E | NUM1 | .DS 1 | |
| 7000 | !* * | 007F | NUM2 | .DS 1 | |
| 7000 | !* C A T S O R T * | 0000 | NUM3 | .DS 1 | |
| 7000 | !* * | 0001 | HEXNUM1 | .DS 1 | |
| 7000 | !* V O N * | 0002 | HEXNUMH | .DS 1 | |
| 7000 | !* N.MANN & D.WEINECK * | 0003 | I | | |
| 7000 | !* * | 0003 | ***** | | |
| 7000 | !* V E R S I O N 7.0 * | 0003 | I V A R I A B L E N | | |
| 7000 | !* 13.8.1985 * | 02A7 | .BA #02A7 | | |
| 7000 | ***** | 02A7 | MEM | .DS 1 | |
| 7000 | I | 02A8 | FIRSTFIL | .DS 2 | |
| 7000 | I | 02AA | SECTOR | .DS 1 | |
| 7000 | ***** | 02AB | DIRMEML | .DS 1 | |
| 7000 | I Z E R O P A G E - V A R I A B L E N | 02AC | DIRMEMH | .DS 1 | |
| 7000 | I | 02AD | YSAVE | .DS 1 | |
| 0070 | | 02AE | XSAVE | .DS 1 | |
| 0070 | .BA #70 | 02AF | COUNT | .DS 1 | |
| 0070 | TEXT | 02B0 | I | | |
| 0072 | DIRENDL | 02B0 | STATUS | .DE #90 | I STATUSREGISTER |
| 0073 | DIRENDH | 02B0 | BUFFER | .DE #0340 | I ZWISCHENSPEICHER |
| 0074 | USE | 02B0 | OLDDIR | .DE #5000 | I BUFFER-BEREICHE |
| 0076 | OLD | 02B0 | DIRBUF | .DE #6000 | I FUER DIRECTORY |
| 0078 | PREV | 02B0 | I | | |
| 007A | NEXT | 02B0 | BORDER | .DE #D020 | I RAHMENFARBE |
| 007C | BUF | 02B0 | BKORND | .DE #D021 | I HINTERGRUND |

```

0200 COLOR .DE #0206 I SCHREIBFARBE
0200 I
0200 I*****
0200 I BETRIEBSSYSTEMROUTINEN
0200 I
0200 RESET .DE #FCE2
0200 I
0200 ACPTR .DE #FFA5
0200 CHKIN .DE #FFC6
0200 CHKOUT .DE #FFC9
0200 CHRIN .DE #FFCF
0200 CHROUT .DE #FFD2
0200 CIOUT .DE #FFA8
0200 CLOSE .DE #FFC3
0200 CLRCHN .DE #FFC4
0200 GETIN .DE #FFE4
0200 LISTEN .DE #FFB1
0200 OPEN .DE #FFC8
0200 PLOT .DE #FFF8
0200 SECOND .DE #FF93
0200 SETLFS .DE #FFBA
0200 SETNAM .DE #FFBD
0200 TALK .DE #FFB4
0200 TKSA .DE #FF96
0200 UNLSN .DE #FFAE
0200 UNTLK .DE #FFAB
0200 I
0200 I
0801 .JA #0801
0801 I
0801 I
0801 I*****
0801 I SYS2064 AN PROGRAMMANFANG
0801 I
0801 0D 08 0A .BY #0D #08 #0A #08
0804 08
0805 9E 28 32 .BY #9E '(2064)'
0808 30 36 34
080B 29
080C 00 00 00 .BY 0 0 0
080F 00
0810 I
0810 I*****
0810 I START
0810 A2 FF LDX #0FF
0812 9A TXS
0813 A9 06 LDA #6
0815 0D 20 D0 STA BORDER
0816 0D 21 D0 STA BKGRND
0818 A9 03 LDA #3
081D 0D 06 02 STA COLOR
0820 20 74 08 JSR CLRSCR
0823 A0 10 LDY #H,STARTTEXT
0825 A9 08 LDA #L,STARTTEXT
0827 20 50 08 JSR TEXTOUT
082A 20 E4 FF START1 JSR GETIN
082D C9 0D CMP #0D
082F D0 F9 BNE START1
0831 I
0831 I
0831 A0 10 LDY #H,TEXT1
0833 A9 C9 LDA #L,TEXT1
0835 20 50 08 JSR TEXTOUT
0838 20 E4 FF START2 JSR GETIN
083B C9 31 CMP #'1'
083D D0 03 BNE START3
083F 4C 9C 08 JMP DIRECTORY
0842 C9 32 START3 CMP #'2'
0844 D0 03 BNE START4
0846 4C 8F 09 JMP SORT
0849 C9 33 START4 CMP #'3'
084B D0 E3 BNE START2
084D 4C E2 FC JMP RESET
0850 I
0850 I
0850 I
0850 I*****
0850 I TEXTOUT ITEXTANFANG IN (TEXT)
0850 I ITEXTLAENGE MAX 64K
0850 I ITEXTENDE MIT EOT (0)
0850 I ICODE < 13 = ANZAHL SPACES
0850 I ICODE 21 = WEISS (FUER 05)
0850 I
0850 04 71 STY TEXT+1
0852 05 70 STA TEXT
0854 A0 00 LDY #00
0856 01 70 TXT1 LDA (TEXT),Y
0858 F0 19 BEQ TXTEND
085A C9 0D CMP #0D
085C 00 05 BCS TXT2
085E AA TAX
085F CA DEX
0860 20 95 08 JSR SPACES
0863 C9 15 TXT2 CMP #21
0865 00 02 BNE TXT3
0867 A9 05 LDA #5
0869 20 D2 FF TXT3 JSR CHROUT
086C C8 INY
086D 00 E7 BNE TXT1
086F E6 71 INC TEXT+1
0871 D0 E3 BNE TXT1
0873 60 TXTEND RTS
0874 I
0874 I*****
0874 I ILOESCHT BILDSCHIRM UND SETZT
0874 I I FARBRAM AUF SCHREIBFARBE
0874 I
0874 A9 93 CLRSCR LDA #93
0876 20 D2 FF JSR CHROUT
0879 A2 00 LDX #0
087B A9 03 LDA #3
087D 9D 00 D0 CLR61 STA #D000,X
0880 9D 00 D9 STA #D900,X
0883 9D 00 DA STA #DAB0,X
0886 9D 00 DB STA #D800,X
0889 E8 INX
088A D0 F1 BNE CLR61
088C 60 RTS
088D I
088D I*****
088D I IHAEUFIG BENUTZTE AUSGABEN
088D I
088D A9 0D RETURN LDA #0D
088F 2C .BY #2C
0890 I
0890 A9 20 SPACE LDA #20
0892 4C D2 FF JMP CHROUT
0895 I
0895 20 90 08 SPACES JSR SPACE
0898 CA DEX
0899 D0 FA BNE SPACES
089B 60 RTS
089C I
089C I*****
089C I I LIEST DIRECTORY EIN UND GIBT
089C I I ES AUF BILDSCHIRM AUS INCL.
089C I I FEHLERBEHANDLUNG
089C I
089C I DIRECTORY
089C 20 74 08 JSR CLRSCR
089F A9 08 LDA #8
08A1 05 98 STA STATUS
08A3 A9 08 LDA #8
08A5 20 01 FF JSR LISTEN
08A8 A9 6F LDA #6F
08AA 20 93 FF JSR SECOND
08AD A9 49 LDA #'I'
08AF 20 A8 FF JSR CIOUT
08B2 20 AE FF JSR UNLSN
08B5 A5 98 LDA STATUS
08B7 30 78 BMI DIREC6
08B9 A9 08 LDA #8
08BB 20 01 FF JSR LISTEN
08BE A9 F8 LDA #F8
08C0 20 93 FF JSR SECOND
08C3 A9 24 LDA #'S'
08C5 20 A8 FF JSR CIOUT
08C8 20 AE FF JSR UNLSN
08CB A5 98 LDA STATUS
08CD 10 03 SPL DIREC0
08CF 4C 19 09 JMP DIREC5
08D2 I
08D2 A9 08 DIREC8 LDA #8
08D4 20 04 FF JSR TALK
08D7 A9 60 LDA #60
08D9 20 96 FF JSR TKSA
08DC 20 A5 FF JSR ACPTR
08DF 20 A5 FF JSR ACPTR
08E2 4C FC 08 JMP DIREC3
08E5 I
08E5 20 A5 FF DIREC1 JSR ACPTR
08E8 AA TAX
08E9 20 A5 FF JSR ACPTR
08EC 20 CD 0D JSR #BDCD
08EF 20 90 08 JSR SPACE
08F2 20 A5 FF DIREC2 JSR ACPTR
08F5 F0 05 BEQ DIREC3
08F7 20 D2 FF JSR CHROUT
08FA D0 F6 BNE DIREC2
08FC 20 A5 FF DIREC3 JSR ACPTR
08FF 20 A5 FF JSR ACPTR
0902 A5 98 LDA STATUS
0904 D0 13 BNE DIREC5
0906 20 00 08 JSR RETURN
0909 AD 0D 02 LDA #D20D
090C C9 02 CMP #02
090E D0 D5 BNE DIREC1
0910 AD 0D 02 DIREC4 LDA #D20D
0913 C9 01 CMP #01
0915 D0 F9 BNE DIREC4
0917 F0 CC BEQ DIREC1
0919 I
0919 20 AB FF DIREC5 JSR UNTLK
091C A9 08 LDA #8
091E 20 01 FF JSR LISTEN
0921 A9 E8 LDA #E8
0923 20 93 FF JSR SECOND
0926 20 AE FF JSR UNLSN

```

Hier war eine Abo-
Bestellkarte — aber jemand
war schneller als Sie und
hat seine „RUN“ mit rund
12% Preisvorteil bereits
abonniert.
Sie können diesen
Vorteil auch nutzen:
Schreiben Sie einfach an
CW-Publikationen,
Vertrieb RUN,
Postfach 40 04 29,
8000 München 40.
Sichwort: „RUN-Abo“.
Friedrichstraße 31,
D-8000 München 40



Die RUN im Abo, das bringt Action. Klar bin ich dabei!

RUN

11/November 1988
8,50 DM
ISSN 0930-2516

UNABHÄNGIGES
COMMODORE
COMPUTERMAGAZIN



C64

TESTET INTELLIGENZ

C 128 IM VISIER
SOFTWARE, LISTINGS,
TIPS & TRICKS

MITMACHEN - GEWINNEN 20 000 MARK- WETTBEWERB

COCKPIT
MON'S BASIC

NETZENSTATION
41-ALTERNATIVE
TURBO-ACCESS

KURS: BLOCKGRAFIK • VC-20 SICHTET DATEN •
KARATESPIEL: EXPLODING FIST • EPROM-
BANK IM RECHNER • NEUES VON DER
FUNKAUSSTELLUNG UND, UND, UND ...

RUN ist das unabhängige COMMODORE-Computermagazin. Monat für Monat erfahren Sie hier Neues und Interessantes, finden Sie viele Listings, Tips & Tricks aus der Welt des COMMODORE. Bleiben Sie am Ball und halten Sie Ihr COMMODORE-Wissen stets auf dem laufenden. Mit Ihrem persönlichen Abonnement von RUN!

CW  **PUBLIKATIONEN**

Eine Erste Adresse für die Computerwelt.
Friedrichstraße 31, D-8000 München 40

```

0929 20 32 09   DIREC6   JSR FEHLER
092C 20 76 09   JSR TAST
092F 4C 10 00   JMP START
0932
0932 I*****
0932 I LIEST FEHLERKANAL DER FLOPPY
0932 I WENN FEHLER, AUSGEBEN UND
0932 I BILDSCHIRMRAHMEN AUF ROT
0932 I
0932 A9 00   FEHLER   LDA #00
0934 20 04 FF   JSR TALK
0937 A9 6F   LDA #06F
0939 20 96 FF   JSR TKSA
093C 20 A5 FF   JSR ACPTR
093F C9 31   CMP #'1'
0941 00   PHP
0942 90 10   BCC FEHLER2
0944 00   PHA
0945 20 0D 00   JSR RETURN
0948 20 0D 00   JSR RETURN
094B A9 02   LDA #02
094D 0D 20 D0   STA #D020
0950 60   PLA
0951 20 D2 FF   JSR CHROUT
0954 20 A5 FF   JSR ACPTR
0957 C9 0D   CMP #0D
0959 F8 0A   BEQ FEHLER3
095B 20   PLP
095C 00   PHP
095D 90 F5   BCC FEHLER2
095F 20 D2 FF   JSR CHROUT
0962 4C 54 09   JMP FEHLER2
0965 20 AB FF   JSR UNTLK
0968 A9 00   LDA #00
096A 05 90   STA #90
096C 20   PLP
096D 00 01   BCS FEHLER4
096F 60   RTS
0970
0970 I
0970 20 76 09   FEHLER4   JSR TAST
0973 4C 04 09   JMP TASTE1
0976
0976 I*****
0976 I TASTATURABFRAGE
0976 I INDTAUSSTIEG IST LINKSPFEIL
0976 I
0976 20 E4 FF   TAST     JSR GETIN
0979 F0 F0   BEQ TAST
097B 60   TASTRTS  RTS
097C
097C I
097C 20 76 09   TASTE    JSR TAST
097F C9 5F   CMP #'*'
0981 F0 01   BEQ TASTE1
0983 60   RTS
0984 20 CC FF   TASTE1   JSR CLRCHN
0987 A9 02   LDA #2
0989 20 C3 FF   JSR CLOSE
098C 4C 10 00   JMP START
098F
098F I
098F I*****
098F I * HAUPTROUTINE VON CATSORT *
098F I*****
098F I
098F 20 65 0A   SORT     JSR TITEL
0992 20 C0 0F   JSR ERASERAM
0995 20 7E 0A   JSR GETDIR
0998 20 A1 0D   JSR PACK
099B A9 00   SORT1    LDA #L,OLDDIR
099D 0D A0 02   STA FIRSTFIL
09A0 A9 50   LDA #H,OLDDIR
09A2 0D A9 02   STA FIRSTFIL+1
09A5 20 E1 0C   JSR PRDIR
09A8 20 6F 0A   JSR WTEXT
09AB AD A0 02   INMENU   LDA FIRSTFIL
09AE 05 74   STA #USE
09B0 AD A9 02   LDA FIRSTFIL+1
09B3 05 75   STA #USE+1
09B5 10   CLC
09B6 A2 04   LDX #4
09B8 A0 00   LDY #0
09BA 20 F0 FF   JSR PLOT
09BD 20 DF 0D   JSR REVLIN
09C0
09C0 I
09C0 20 7C 09   MENU     JSR TASTE
09C3 F0 FB   BEQ MENU
09C5 C9 91   CMP #145
09C7 D0 03   BNE MENU1
09C9 4C ED 0D   JMP CURUP
09CC C9 11   CMP #17
09CE D0 03   BNE MENU2
09D0 4C 20 0E   JMP CURDOWN
09D3 C9 06   CMP #134
09D5 D0 20   BNE MENU3
09D7 A9 00   LDA #L,OLDDIR
09D9 C3 72   CMP #DIRENDL
09DB A9 50   LDA #H,OLDDIR
09DD 05 73   SBC #DIRENDH
09DF 00 DF   BCS MENU
09E1 20 DF 0D
09E4 A3 74
09E6 C3 72
09E8 A5 75
09EA E5 73
09EC 90 05
09EE A9 91
09F0 20 D2 FF   MENU2.1
09F3 20 31 0D
09F6 20 E1 0C
09F9 20 DF 0D
09FC 4C C0 09
09FF C9 07   MENU3
0A01 D0 0F
0A03 20 DF 0D
0A06 20 6A 0E
0A09 20 E1 0C
0A0C 20 DF 0D
0A0F 4C C0 09
0A12 C9 08   MENU4
0A14 D0 03
0A16 4C C2 0E
0A19 C9 13   MENU5
0A1B D0 06
0A1D 20 DF 0D
0A20 4C 90 09
0A23 C9 4C   MENU6
0A25 D0 21
0A27 20 DF 0D
0A2A A0 02
0A2C 31 74
0A2E 49 40
0A30 91 74
0A32 A5 74
0A34 05 76
0A36 A5 75
0A38 05 77
0A3A 20 75 0C
0A3D A9 91
0A3F 20 D2 FF
0A42 20 DF 0D
0A45 4C C0 09   MENU8
0A48 C9 5F   MENU7
0A4A D0 03
0A4C 4C 10 00
0A4F C9 94   MENU9
0A51 D0 F2
0A53 20 DF 0D
0A56 20 02 0F
0A59 20 6A 0E
0A5C 20 E1 0C
0A5F 20 DF 0D
0A62 4C C0 09
0A65
0A65 I
0A65 I
0A65 20 74 00   TITEL
0A68 A0 11
0A6A A9 26
0A6C 4C 50 00
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C0 09   JSR REVLIN
0AD5 C9 5F   JMP MENU
0AD7 D0 03   CMP #05F
0AD9 05 75   BNE MENU9
0ADB 20 DF 0D   JMP START
0ADC C9 94   CMP #148
0ADE D0 F2   BNE MENU8
0ADF 20 DF 0D   JSR REVLIN
0AE1 20 02 0F   JSR INSFIL
0AE3 20 6A 0E   JSR INSFIL
0AE5 20 E1 0C   JSR PRDIR
0AE7 20 DF 0D   JSR REVLIN
0AE9 4C C0 09   JMP MENU
0A6F
0A6F I*****
0A6F I GIBT ATBEITSMENUE AUS
0A6F I
0A6F I
0A6F A0 00   WTEXT
0A71 A2 17
0A73 10
0A74 20 F0 FF
0A77 A0 11
0A79 A9 53
0A7B 4C 50 00
0A7E
0A7E I
0A7E I
0A7E I*****
0A7E I HOLT DIRECTORY IN DEN BUFFER
0A7E I
0A7E 20 02 00   GETDIR
0A81 20 07 00   GETDIR2
0A84 A9 31
0A86 20 A0 FF
0A89 20 AE FF
0A8C 20 32 09
0A8F A9 00
0A91 05 76
0A93 A9 50
0A95 05 77
0A97 A2 02   GETDIR3
0A99 20 C6 FF
0A9C A0 00
0A9E 20 CF FF   GETDIR4
0AA1 90 03
0AA3 4C 32 09
0AA6 91 76   GETDIR5
0AA8 C0
0AA9 D0 F3
0AAB 20 CC FF
0AAE 31 76
0AB0 F0 12
0AB2 20 07 00
0AB5 A0 01
0A81 20 DF 0D   JSR REVLIN
0A84 A0 0E   LDA #USE
0A86 C3 72   CMP #DIRENDL
0A88 A5 75   LDA #USE+1
0A8A 05 73   SBC #DIRENDH
0A8C 20 02 0E   BCC MENU2.1
0A8E 4C 10 00   LDA #145
0A90 20 D2 FF   JSR CHROUT
0A93 20 31 0D   JSR DELFIL
0A95 20 DF 0D   JSR PRDIR
0A97 C9 07   JSR REVLIN
0A99 D0 0F   JMP MENU
0A9B 05 75   CMP #135
0A9D 05 74   BNE MENU4
0A9F 20 DF 0D   JSR REVLIN
0AA1 20 E1 0C   JSR INSFIL
0AA3 20 DF 0D   JSR PRDIR
0AA5 4C C0 09   JSR REVLIN
0AA7 C9 08   JMP MENU
0AA9 C9 13   CMP #136
0AAB D0 06   BNE MENU5
0AAD 20 DF 0D   JSR REVLIN
0AAE 4C 90 09   JMP SORTEND
0AB0 C9 4C   CMP #19
0AB2 D0 21   BNE MENU6
0AB4 20 DF 0D   JSR REVLIN
0AB6 4C 90 09   JMP SORT1
0AB8 C9 4C   CMP #'L'
0ABA D0 0F   BNE MENU7
0ABC 20 DF 0D   JSR REVLIN
0ABE A0 02   LDY #2
0ABF 31 74   LDA (USE),Y
0AC1 49 40   EOR #00
0AC3 91 74   STA (USE),Y
0AC5 A5 74   LDA #USE
0AC7 05 76   STA #OLD
0AC9 A5 75   LDA #USE+1
0ACB 05 77   STA #OLD+1
0ACE 20 75 0C   JSR ONLINE
0ACF A9 91   LDA #145
0AD1 20 DF 0D   JSR CHROUT
0AD3 4C C
```



```

0C0C          I*****
0C0C          IWAENDELT EINE HEXZAHL IN EINE
0C0C          IDEZIMALZAHL UM UND GIBT SIE
0C0C          I AUF DEM BILDSCHIRM AUS
0C0C          I
0C0C A9 00      HEXDEZ      LDA #0
0C0E 05 7E          STA #NUM1
0C10 05 7F          STA #NUM2
0C12 05 00          STA #NUM3
0C14 A2 10          LDX #16
0C16 F8          NUMLOOP   SED
0C17 06 01          ASL HEXNUML
0C19 26 02          ROL HEXNUMH
0C1B A5 00          LDA #NUM3
0C1D 65 00          ADC #NUM3
0C1F 05 00          STA #NUM3
0C21 A5 7F          LDA #NUM2
0C23 65 7F          ADC #NUM2
0C25 05 7F          STA #NUM2
0C27 A5 7E          LDA #NUM1
0C29 65 7E          ADC #NUM1
0C2B 05 7E          STA #NUM1
0C2D 08          CLD
0C2E CA          DEX
0C2F 00 E5          BNE NUMLOOP
0C31          I
0C31 A5 7E          HEXOUT   LDA #NUM1
0C33 09 30          ORA #030
0C35 9D 40 03      STA BUFFER,X
0C38 A5 7F          LDA #NUM2
0C3A 20 5E 0C      JSR LEFTN
0C3D A5 00          LDA #NUM3
0C3F 20 5E 0C      JSR LEFTN
0C42 A2 00          LDX #00
0C44 3D 40 03      HEXOUT1  LDA BUFFER,X
0C47 C9 30          CMP #30
0C49 D0 03          BNE HEXOUT2
0C4B E8          INX
0C4C E0 05          CPX #5
0C4E 90 F4          BCC HEXOUT1
0C50 20 D2 FF      HEXOUT2  JSR CHROUT
0C53 E8          INX
0C54 E8 05          CPX #05
0C56 80 05          BCS OUTRTS
0C58 3D 40 03      LDA BUFFER,X
0C5B D0 F3          BNE HEXOUT2
0C5D 60          OUTRTS  RTS
0C5E          I
0C5E 48          LEFTN   PHA
0C5F 20 60 0C      JSR LEFT1
0C62 60          PLA
0C63 29 0F          AND #00F
0C65 4C 6E 0C      JMP LEFT2
0C68 29 F0          AND #0F0
0C6A 4A          LSR A
0C6B 4A          LSR A
0C6C 4A          LSR A
0C6D 4A          LSR A
0C6E 09 30          LEFT2   ORA #30
0C70 E8          INX
0C71 9D 40 03      STA BUFFER,X
0C74 60          RTS
0C75          I
0C75          I
0C75          I*****
0C75          I GIBT EINE ZEILE DES DIRECTORY
0C75          I AUF DEM BILDSCHIRM AUS
0C75          I
0C75          I ONELINE
0C75 A9 00          LDA #0
0C77 05 D3          STA #D3
0C79 A2 06          LDX #6
0C7B 20 95 00      JSR SPACES
0C7E A8 1E          LDY #30
0C80 B1 74          LDA (OLD),Y
0C82 05 01          STA HEXNUML
0C84 C8          INY
0C85 B1 74          LDA (OLD),Y
0C87 05 02          STA HEXNUMH
0C89 D0 12          BNE ONEL1
0C8B A5 01          LDA HEXNUML
0C8D C9 64          CMP #100
0C8F 00 0C          BCS ONEL1
0C91 20 90 00      JSR SPACE
0C94 A5 01          LDA HEXNUML
0C96 C9 0A          CMP #10
0C98 00 03          BCS ONEL1
0C9A 20 90 00      JSR SPACE
0C9D 20 0C 0C      ONEL1   JSR HEXDEZ
0CA0 A2 02          LDX #2
0CA2 20 95 00      JSR SPACES
0CA5 A0 03          LDY #5
0CA7 B1 74          ONEL2   LDA (OLD),Y
0CA9 20 FD 00      JSR PROUT
0CAC C0          INY
0CAD C0 15          CPY #21
0CAF 90 F4          BCC ONEL2
0CB1 20 90 00      JSR SPACE
0CB4 A0 02          LDY #2

```

```

0CB6 B1 74          LDA (OLD),Y
0CB8 40          PHA
0CB9 29 0F          AND #00F
0CB8 AA          TAX
0CB8 CA          DEX
0CB8 3D FF 0F      LDA TYTAB1,X
0CC0 20 FD 00      JSR PROUT
0CC3 3D 02 10      LDA TYTAB2,X
0CC6 20 FD 00      JSR PROUT
0CC9 3D 05 10      LDA TYTAB3,X
0CC0 20 FD 00      JSR PROUT
0CCF 60          PLA
0CD0 29 40          AND #40
0CD2 F0 05          BEQ ONEL3
0CD4 A9 3E          LDA #'>'
0CD6 20 FD 00      JSR PROUT
0CD9 A2 02          LDX #2
0CDB 20 95 00      JSR SPACES
0CDE 4C 0D 00      JMP RETURN
0CE1          I
0CE1          I
0CE1          I*****
0CE1          I DRUCKT EINEN BILDSCHIRM AUS
0CE1          I (FIRSTFIL ENTHAELT DIE ADRESSE
0CE1          I DES ERSTEN FILES AUF DEM SCHIRM
0CE1          I
0CE1          I PRDIR   SEC
0CE1 38          JSR PLOT
0CE2 20 F0 FF      STX XSAVE
0CE5 0E AE 02      STY YSAVE
0CE8 0C AD 02      CLC
0CEB 18          LDX #4
0CEC A2 04          LDY #0
0CEE A0 00          JSR PLOT
0CF0 20 F0 FF      LDA FIRSTFIL
0CF3 AD A8 02      STA #OLD
0CF6 05 76          LDA FIRSTFIL+1
0CF8 AD A9 02      STA #OLD+1
0CFB 05 77          LDA #10
0CFD A9 12          PRDIR0  STA COUNT
0CFF 0D AF 02      JSR ONELINE
0D02 20 75 0C      DEC COUNT
0D05 CE AF 02      BEQ PRDIR4
0D08 F0 10          LDA #OLD
0D0A A5 74          CMP #DIRENDL
0D0C C3 72          LDA #OLD+1
0D0E A5 77          SBC #DIRENDH
0D10 E5 73          BCS PRDIR3
0D12 30 0E          LDA #OLD
0D14 A5 74          CLC
0D16 18          ADC #20
0D17 69 20          STA #OLD
0D19 05 76          BCC PRDIR2
0D1B 90 02          INC #OLD+1
0D1D E6 77          JMP PRDIR1
0D1F 4C 02 0D      PRDIR2  LDX #39
0D22 A2 27          PRDIR3  JSR SPACES
0D24 20 95 00      CLC
0D27 10          PRDIR4  LDY XSAVE
0D28 AE AE 02      LDY YSAVE
0D2B AC AD 02      JMP PLOT
0D2E 4C F0 FF
0D31          I
0D31          I*****
0D31          I LUESCHT EINEN FILEEINTRAG
0D31          I (USE IST ADRESSE DES ZU LOESCHENDEN
0D31          I (FILES IM SPEICHER, BUF IST BUFFER
0D31          I (USE STEHT ANSCHLIESSEND AUF DEM
0D31          I (NAECHSTEN EINTRAG
0D31          I
0D31 A0 1F          DELFIL  LDY #1F
0D33 B1 74          DELFIL1 LDA (USE),Y
0D35 91 7C          STA (BUF),Y
0D37 00          DEY
0D38 10 F9          BPL DELFIL1
0D3A A5 7C          LDA #BUF
0D3C 10          CLC
0D3D 69 20          ADC #20
0D3F 05 7C          STA #BUF
0D41 90 02          BCC DELFIL2
0D43 E6 7D          INC #BUF+1
0D45 A5 74          LDA #USE
0D47 05 78          STA #PREV
0D49 A5 75          LDA #USE+1
0D4B 05 79          STA #PREV+1
0D4D A5 78          DELFIL3 LDA #PREV
0D4F C3 72          CMP #DIRENDL
0D51 A5 79          LDA #PREV+1
0D53 E5 73          SBC #DIRENDH
0D55 00 21          BCS DELFIL5
0D57 A5 78          LDA #PREV
0D59 10          CLC
0D5A 69 20          ADC #20
0D5C 05 7A          STA #NEXT
0D5E A5 79          LDA #PREV+1
0D60 69 00          ADC #0
0D62 05 78          STA #NEXT+1
0D64 A0 1F          LDY #1F
0D66 B1 7A          DELFIL4 LDA (NEXT),Y

```

```

0D60 91 78          STA (PREV),Y
0D6A 00            DEY
0D6B 10 F9          BPL DELFIL4
0D6D A5 7A          LDA #NEXT
0D6F 05 78          STA #PREV
0D71 A5 78          LDA #NEXT+1
0D73 05 79          STA PREV+1
0D75 4C 4D 0D      JMP DELFIL3
0D78 A0 1F          DELFIL5 LDY #*1F
0D7A A9 00          LDA #0
0D7C 91 78          DELFIL6 STA (PREV),Y
0D7E 00            DEY
0D7F 10 F9          BPL DELFIL6
0D81 30            SEC
0D82 A5 78          LDA #PREV
0D84 E9 20          SBC #*20
0D86 05 72          STA #DIRENDL
0D88 A5 79          LDA #PREV+1
0D8A E9 00          SBC #0
0D8C 05 73          STA #DIRENDH
0D8E A5 72          LDA #DIRENDL
0D90 C5 74          CMP #USE
0D92 A5 73          LDA #DIRENDH
0D94 E5 75          SBC #USE+1
0D96 00 00          BCS DELFIL7
0D98 A5 72          LDA #DIRENDL
0D9A 05 74          STA #USE
0D9C A5 73          LDA #DIRENDH
0D9E 05 75          STA #USE+1
0DA0 60            DELFIL7 RTS
0DA1                I
0DA1                I*****
0DA1                IENTFERNT GELOESCHTE EINTRAEGE
0DA1                I AUS DEM DIRECTORY UND SCHIEST
0DA1                I ZUSAMMEN
0DA1                I
0DA1 A9 00          PACK    LDA #L,OLDDIR
0DA3 05 74          STA #USE
0DA5 A9 00          LDA #H,OLDDIR
0DA7 05 75          STA #USE+1
0DA9 A9 00          LDA #L,DIRBUF
0DAB 05 7C          STA #BUF
0DAD A9 60          LDA #H,DIRBUF
0DAF 05 7D          STA #BUF+1
0DB1 A0 02          PACK1  LDY #2
0DB3 01 74          LDA (USE),Y
0DB5 29 0F          AND #*0F
0DB7 00 06          BNE PACK2
0DB9 20 31 0D      PACK2  JSR DELFIL
0DBC 4C 01 0D      JMP PACK1
0DBF A5 74          LDA #USE
0DC1 C5 72          CMP #DIRENDL
0DC3 A5 75          LDA #USE+1
0DC5 05 73          SBC #DIRENDH
0DC7 00 0D          BCS PACK4
0DC9 A5 74          LDA #USE
0DCB 69 20          ADC #*20
0DCC 05 74          STA #USE
0DCF 90 02          BCC PACK3
0DD1 E6 75          INC #USE+1
0DD3 4C 01 0D      PACK3  JMP PACK1
0DD6 A9 00          PACK4  LDA #L,DIRBUF
0DD8 05 7C          STA #BUF
0DDA A9 60          LDA #H,DIRBUF
0DDC 05 7D          STA #BUF+1
0DDE 60            RTS
0DDF                I
0DDF                I*****
0DDF                I WANDELT AKTUELLE ZEILE IN
0DDF                I REVERSSCHRIFT ODER ZURUECK
0DDF                I
0DDF A0 04          REVLIN  LDY #4
0DE1 01 01          REVL1  LDA (#D1),Y
0DE3 49 00          EOR #*00
0DE5 91 01          STA (#D1),Y
0DE7 C0            INY
0DE8 C0 22          CPY #34
0DEA 90 F5          BCC REVL1
0DEC 60            RTS
0DED                I
0DED                I*****
0DED                ISETZT ZEILENCURSOR EINE ZEILE
0DED                I HOEHER
0DED                I
0DED A5 75          CURUP   LDA #USE+1
0DEF C9 00          CMP #H,OLDDIR
0DF1 00 09          BNE CURUP1
0DF3 A5 74          LDA #USE
0DF5 C9 00          CMP #L,OLDDIR
0DF7 00 03          BNE CURUP1
0DF9 4C 00 09      CURUP1  JMP MENU
0DFC 30            SEC
0DFD A5 74          LDA #USE
0DFE E9 20          SBC #*20
0E01 05 74          STA #USE
0E03 00 02          BCS CURUP2
0E05 C6 75          DEC #USE+1
0E07 20 0F 0D      CURUP2  JSR REVLIN
0E0A A5 06          LDA #06
0E0C C9 04          DE0C   CMP #4
0E0E F0 00          DE0E   BEQ CURUP3
0E10 A9 91          DE10   LDA #145
0E12 20 02 FF      DE12   JSR CHROUT
0E15 20 0F 0D      DE15   JSR REVLIN
0E18 4C 00 09      DE18   JMP MENU
0E1B A5 74          CURUP3  LDA #USE
0E1D 00 A0 02      DE1D   STA FIRSTFIL
0E20 A5 75          DE20   LDA #USE+1
0E22 0D A9 02      DE22   STA FIRSTFIL+1
0E25 20 E1 0C      DE25   JSR PRDIR
0E28 4C A0 09      DE28   JMP INMENU
0E29                I
0E29                I*****
0E29                ISETZT ZEILENCURSOR EINE ZEILE
0E29                I TIEFER
0E29                I
0E29 A5 74          CURDOWN LDA #USE
0E2D C5 72          DE2D   CMP #DIRENDL
0E2F A5 75          DE2F   LDA #USE+1
0E31 E5 73          DE31   SBC #DIRENDH
0E33 90 03          DE33   BCC CURD1
0E35 4C 00 09      DE35   JMP MENU
0E38 A5 74          CURD1  LDA #USE
0E3A 69 20          DE3A   ADC #*20
0E3C 05 74          DE3C   STA #USE
0E3E 90 02          DE3E   BCC CURD2
0E40 E6 75          DE40   INC #USE+1
0E42 20 0F 0D      CURD2  JSR REVLIN
0E45 A5 06          DE45   LDA #06
0E47 C9 15          DE47   CMP #21
0E49 00 00          DE49   BCS CURD3
0E4B A9 11          DE4B   LDA #17
0E4D 20 02 FF      DE4D   JSR CHROUT
0E50 20 0F 0D      CURD2.1 JSR REVLIN
0E53 4C 00 09      DE53   JMP MENU
0E56 10            CURD3  CLC
0E57 AD A0 02      DE57   LDA FIRSTFIL
0E5A 69 20          DE5A   ADC #*20
0E5C 0D A0 02      DE5C   STA FIRSTFIL
0E5F 90 03          DE5F   BCC CURD4
0E61 EE A9 02      DE61   INC FIRSTFIL+1
0E64 20 E1 0C      CURD4  JSR PRDIR
0E67 4C 00 0E      DE67   JMP CURD2.1
0E6A                I
0E6A                I*****
0E6A                I FUEGT EINEN FILEEINTRAG EIN
0E6A                I USE IST DIE ADRESSE, AN DER EIN FILE
0E6A                I EINGEFUEGT WIRD. DIREND WIRD ERHOEHET
0E6A                I OBERSTER BUF-INHALT WIRD EINGEFUEGT.
0E6A                I USE WIRD NICHT GEANDERT.
0E6A                I
0E6A A9 00          INSFIL  LDA #L,DIRBUF
0E6C C5 7C          DE6C   CMP #BUF
0E6E A9 60          DE6E   LDA #H,DIRBUF
0E70 E5 7D          DE70   SBC #BUF+1
0E72 90 01          DE72   BCC INSF1
0E74 60            DE74   RTS
0E75 A5 72          INSF1  LDA #DIRENDL
0E77 05 78          DE77   STA #PREV
0E79 69 20          DE79   ADC #*20
0E7B 05 72          DE7B   STA #DIRENDL
0E7D 05 7A          DE7D   STA #NEXT
0E7F A5 73          DE7F   LDA #DIRENDH
0E81 05 79          DE81   STA #PREV+1
0E83 69 00          DE83   ADC #0
0E85 05 73          DE85   STA #DIRENDH
0E87 05 70          DE87   STA NEXT+1
0E89 A0 1F          INSF2  LDY #*1F
0E8B 01 70          INSF3  LDA (PREV),Y
0E8D 91 7A          DE8D   STA (NEXT),Y
0E8F 00            DE8F   DEY
0E90 10 F9          DE90   BPL INSF3
0E92 30            DE92   SEC
0E93 A5 78          DE93   LDA #PREV
0E95 05 7A          DE95   STA #NEXT
0E97 E9 20          DE97   SBC #*20
0E99 05 78          DE99   STA #PREV
0E9B A5 79          DE9B   LDA #PREV+1
0E9D 05 73          DE9D   STA #NEXT+1
0E9F E9 00          DE9F   SBC #*00
0EA1 05 79          DEA1   STA #PREV+1
0EA3 A5 78          DEA3   LDA #PREV
0EA5 C5 74          DEA5   CMP #USE
0EA7 A5 79          DEA7   LDA #PREV+1
0EA9 E5 75          DEA9   SBC #USE+1
0EAB 00 DC          DEAB   BCS INSF2
0EAD 30            DEAD   SEC
0EAE A5 7C          DEAE   LDA #BUF
0E90 E9 20          DE90   SBC #*20
0E92 05 7C          DE92   STA #BUF
0E94 00 02          DE94   BCS INSF4
0E96 C6 7D          DE96   DEC #BUF+1
0E98 A0 1F          INSF4  LDY #*1F
0E9A 01 7C          INSF5  LDA (BUF),Y
0E9C 91 74          DE9C   STA (USE),Y
0E9E 00            DE9E   DEY
0E9F 10 F9          DE9F   BPL INSF5

```

```

DEC1 68          RTS
DEC2            I
DEC2            I*****
DEC2            IABSCHLUSS MIT ZURUECKSCHREIBEN
DEC2            IDES GEANDERTEN DIRECTORY
DEC2            I
DEC2 28 08 0E   SORTEND   JSR PRBUF
DEC3 88 03          BCS SORTEND1
DEC7 28 2E 0F          JSR CLRBUF
DECA 28 E4 0A   SORTEND1 JSR PUTDIR
DECd 4C 18 08          JMP START
DED8            I
DED8            I*****
DED8            IGIBT RESTLICHE FILEEINTRAEGE IM
DED8            IBUFFER AUS
DED8            I
DED8 A9 08   PRBUF   LDA #L,DIRBUF
DED2 85 74          STA #OLD
DED4 C5 7C          CMP #BUF
DED6 A9 68   LDA #H,DIRBUF
DED8 85 77          STA #OLD+1
DEDA E5 7D          SBC #BUF+1
DEDC 98 81          BCC PRBUF1
DEDE 68          RTS
DEFD 28 65 0A   PRBUF1 JSR TITEL
DEF2 A8 84          LDY #4
DEF4 A2 17          LDX #23
DEF6 18          CLC
DEF7 28 F8 FF   JSR PLOT
DEEA A8 08          LDY #8
DEEC 89 A7 11   PRBUF2 LDA TSCRATCH,Y
DEF0 F0 26          BEG PRBUF3
DEF1 28 D2 FF   JSR CHROUT
DEF4 C8          INY
DEF5 D8 F5          BNE PRBUF2
DEF7 A5 72   PRBUF3 LDA #DIRENDL
DEF9 8D AB 02          STA DIRMEHL
DEFB A5 73          LDA #DIRENDH
DEFD 8D AC 02          STA DIRMEMH
DEF8 28 7C          SEC
DEF8 A5 7C          LDA #BUF
DEF8 E9 28          SBC #*28
DEF8 85 72          STA #DIRENDL
DEF8 A5 7D          LDA #BUF+1
DEF8 85 73          SBC #8
DEF8 C5 73          STA #DIRENDH
DEF8 A2 84          LDX #4
DEF8 8E AE 02          STX XSAVE
DEF8 A8 08          LDY #8
DEF8 8C AD 02          STY YSAVE
DEF8 18          CLC
DEF8 28 F8 FF   JSR PLOT
DEF8 28 FD 0C   JSR PRDIR8
DEF8 AD AB 02          LDA DIRMEHL
DEF8 85 72          STA #DIRENDL
DEF8 AD AC 02          LDA DIRMEMH
DEF8 85 73          STA #DIRENDH
DEF8 28 DF 8D   JSR REVLINE
DEF8 18          CLC
DEF8 68          RTS
DEF8            I
DEF8            I*****
DEF8            IHAENGT DAS FILE AN DAS DIRECTORY
DEF8            IAN ODER LOESCHT ES
DEF8            I
DEF8 A9 08   CLRBUF   LDA #L,DIRBUF
DEF8 85 74          STA #USE
DEF8 A9 68   LDA #H,DIRBUF
DEF8 85 75          STA #USE+1
DEF8 28 7C 89   CLRBUF1 JSR TASTE
DEF8 C9 53          CMP #'S'
DEF8 F8 3A          BEG SCRATCH
DEF8 C9 8D          CMP #*8D
DEF8 88 F5          BNE CLRBUF1
DEF8 18          CLC
DEF8 A5 72          LDA #DIRENDL
DEF8 69 28          ADC #*28
DEF8 85 72          STA #DIRENDL
DEF8 98 82          BCC APPEND
DEF8 E6 73          INC #DIRENDH
DEF8            I
DEF8            I*****
DEF8            IANHAENGEN EINES FILES
DEF8            I
DEF8 A8 1F   APPEND   LDY #*1F
DEF8 B1 74   APPEND1  LDA (USE),Y
DEF8 91 72          STA (DIRENDL),Y
DEF8 88          DEY
DEF8 18 F9          BPL APPEND1
DEF8 18          CLC
DEF8 A5 74          LDA #USE
DEF8 69 28          ADC #*28
DEF8 85 74          STA #USE
DEF8 98 82          BCC APPEND3
DEF8 E6 75          INC #USE+1
DEF8 C5 7C          CMP #BUF
DEF8 A5 75          LDA #USE+1
DEF8 E3 7D          SBC #BUF+1
DEF8 98 81          BCC APPEND4

```

```

BF68 68          RTS
BF69 28 DF 8D   APPEND4 JSR REVLINE
BF6C A9 11          LDA #17
BF6E 28 D2 FF   JSR CHROUT
BF71 28 DF 8D   JSR REVLINE
BF74 4C 36 8F   JMP CLRBUF1
BF77            I
BF77            I*****
BF77            ILOESCHEN EINES FILES
BF77            I
BF77 A8 05   SCRATCH  LDY #5
BF79 B1 74   SCRATCH1 LDA (USE),Y
BF7B 99 38 83   STA BUFFER-5,Y
BF7E C8          INY
BF7F C8 15   CPY #21
BF81 98 F4          BCC SCRATCH1
BF83 A9 53          LDA #'S'
BF85 8D 3D 83   STA BUFFER-3
BF88 A9 38          LDA #'0'
BF8A 8D 3E 83   STA BUFFER-2
BF8D A9 3A          LDA #'!'
BF8F 8D 3F 83   STA BUFFER-1
BF92 A9 08          LDA #8
BF94 28 B1 FF   JSR LISTEN
BF97 A9 6F          LDA #*6F
BF99 28 93 FF   JSR SECOND
BF9C A8 88          LDY #8
BF9E 39 3D 83   SCRATCH2 LDA BUFFER-3,Y
BFA1 28 AB FF   JSR CIOUT
BFA4 C8          INY
BFA5 C8 13   CPY #19
BFA7 98 F5          BCC SCRATCH2
BFA9 28 AE FF   JSR UNLNL
BFAC 28 32 89   JSR FEHLER
BFAF 4C 55 8F   JMP APPEND2
BF92            I
BF92            I*****
BF92            IFUEGT EINEN STRICH INS DIRECTORY EIN
BF92            I
BF92 A8 1F   INSLINE  LDY #*1F
BF94 89 ED 11   INSL1   LDA INTEXT,Y
BF97 91 7C          STA (BUF),Y
BF99 88          DEY
BF9A 18 F8          BPL INSL1
BF9C 18          CLC
BF9D A5 7C          LDA #BUF
BF9F 69 28          ADC #*28
BFA1 85 7C          STA #BUF
BFA3 98 82          BCC INSL2
BFA5 E6 7D          INC #BUF+1
BFA7 88          INSL2   RTS
BFA8            I
BFA8            I*****
BFA8            ILOESCHT BUFFERBEREICH
BFA8            I
BFA8 A9 88   ERASERAM LDA #L,OLDDIR
BFAA A2 58          LDX #H,OLDDIR
BFAC 85 74          STA USE
BFAC 86 75          STX USE+1
BFAD A9 88          LDA #8
BFAD A8          TAY
BFAD 91 74   ERA1   STA (USE),Y
BFAD C8          INY
BFAD 88 F8          BNE ERA1
BFAD E6 75          INC USE+1
BFAD A6 75          LDX USE+1
BFAD E8 68          CPX #H,DIRBUF
BFAD 98 F3          BCC ERA1
BFAD 68          RTS
BFAE            I
BFAE            I
BFAE 88 F5 28   READM   .BY 'U1 2 8 18 ' *88
BFAE 32 28 38   .BY 'U2 2 8 18 ' *88
BFAE 28 31 38   .BY 'B-P 2 8 ' *88
BFAE 28 00   WRITEM   .BY 'SPU'
BFAE 32 28 38   .BY 'ERS'
BFAE 28 31 38   .BY 'GOR'
BFAE 28 00   STARTTEXT
BFAE 42 2D 58   .BY *8D *8D *8D
BFAE 28 32 28   .BY 6
BFAE 38 88   .BY '*****'
BFAE 53 58 55   TYPTAB1 .BY 'SPU'
BFAE 45 52 53   TYPTAB2 .BY 'ERS'
BFAE 51 47 52   TYPTAB3 .BY 'GOR'
BFAE 88          I
BFAE 88          I
BFAE 8D 8D 8D 8D
BFAE 88          .BY 6
BFAE 23 23 23   .BY '*****'
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23
BFAE 23 23 23

```


Floppy

```

1025 23 23 0D
1028 06
1029 23 0A 0A
102C 05 23 0D
102F 06
1038 23 06 43
1033 20 41 20
1036 54 20 53
1039 20 4F 20
103C 52 20 54
103F 06 23 0D
1042 06
1043 23 20 20
1046 42 59 20
1049 4E 2E 4D
104C 41 4E 4E
104F 20 26 20
1052 44 2E 57
1055 45 49 4E
1058 45 43 4B
105B 20 20 23
105E 0D
105F 06
1060 23 0A 0A
1063 05 23 0D
1066 06
1067 23 23 23
106A 23 23 23
106D 23 23 23
1070 23 23 23
1073 23 23 23
1076 23 23 23
1079 23
107A 23 23 23
107D 23 23 23
1080 23 23 0D
1083 0D 0D
1085 20 20 4C
1088 45 47 45
108B 4E 20 53
108E 49 45 20
1091 44 49 45
1094 20 52 49
1097 43 40 54
109A 49 47 45
109D 20 44 49
10A8 53 48 45
10A3 54 54 45
10A6 20 45 49
10A9 4E 0D 0D
10AC 07 55 4E
10AF 44 20 44
10B2 52 55 45
10B5 43 48 45
10B8 4E 20 53
10BB 49 45 20
10BE 12 20 52
10C1 45 54 55
10C4 52 4E 20
10C7 92 00
10C9 0D 0D
10CB 0A 42 49
10CE 54 54 45
10D1 20 57 41
10D4 45 48 4C
10D7 45 4E 20
10DA 53 49 45
10DD 20 3A
10DF 0D 0D
10E1 09 31 2E
10E4 20 44 49
10E7 52 45 43
10EA 54 4F 52
10ED 59 20 41
10F0 4E 5A 45
10F3 49 47 45
10F6 4E 0D
10F8 09 32 2E
10FB 20 44 49
10FE 52 45 43
1101 54 4F 52
1104 59 20 53
1107 4F 52 54
110A 49 45 52
110D 45 4E 0D
1110 09 33 2E
1113 20 50 52
1116 4F 47 52
1119 41 4D 4D
111C 20 42 45
111F 45 4E 44
1122 45 4E 0D
1125 00
1126
1126 06
1127 2A 2A 2A
112A 2A 20 20

```

```

.BY 6
.BY 'M' 10 10 5 'M' 00D
.BY 6
.BY 'M' & 'C A T S O R T'
.BY 6 'M' 00D
.BY 6
.BY 'M BY N.MANN & D.WEINECK M' 00D
.BY 6
.BY 'M' 10 10 5 'M' 00D
.BY 6
.BY 'XXXXXXXXXXXXXXXXXXXX'
.BY 'XXXXXXXXXX' 00D 00D 00D
.BY ' LEGEN SIE DIE'
.BY ' RICHTIGE DISKETTE EIN' 00D 00D
.BY 7 'UND DRUECKEN SIE '
.BY 12 ' RETURN ' 092 0
TEXT1 .BY 00D 00D
.BY 10 'BITTE WAEHLN SIE ;'
.BY 00D 00D
.BY 9 '1. DIRECTORY ANZEIGEN' 00D
.BY 9 '2. DIRECTORY SORTIEREN' 00D
.BY 9 '3. PROGRAMM BEENDEN' 00D 0

```

```

112D 43 20 41
1130 20 54 20
1133 53 20 4F
1136 20 52 20
1139 54 20 20
113C 2A 2A 2A
113F 2A
1140 0D 0D 12
1143 06 42 4C
1146 48 06 4E
1149 41 4D 45
114C 09 54 59
114F 50
1150 09 0D 00
1153
1153 01 12 20
1156 46 33 20
1159 92 9F 20
115C 20 2D 3E
115F 20 50 55
1162 46 46 45
1165 52 20 20
1168 20
1169 20 20 01
116C 12 20 46
116F 37 20 92
1172 9F
1173 20 20 2D
1176 3E 20 44
1179 49 53 4B
117C 45 54 54
117F 45 0D
1181 01 12 20
1184 46 35 20
1187 92 9F 20
118A 20 50 53
118D 46 40 45
1190 52 20 2D
1193 3E 20 20
1196 20
1197 02 01 12
119A 02 5F 20
119D 92
119E 9F 20 20
11A1 4D 45 4E
11A4 55 45 00
11A7
11A7 52 45 53
11AA 54 4C 49
11AD 43 48 45
11B0 20 50 52
11B3 4F 47 52
11B6 41 4D 4D
11B9 45 20
11BB 49 4D 20
11BE 50 55 46
11C1 46 45 52
11C4 0D 20 20
11C7 20 20
11C9 53 43 52
11CC 41 54 43
11CF 48 20 3A
11D2 20 01 12
11D5 20 53 20
11D8 92 9F
11DA 20 20 2D
11DD 2D 20 2D
11E0 53 4F 4E
11E3 53 54 20
11E6 52 45 54
11E9 55 52 4E
11EC 00
11ED
11ED 00 00 03
11F0 12 01
11F2 2D 2D 2D
11F5 2D 2D 2D
11F8 2D 2D 2D
11FB 2D 2D 2D
11FE 2D 2D 2D
1201 2D
1202 00 00 00
1205 00 00 00
1208 00 00 00
120B 00 00
120D
120D
120D

```

```

.BY 00D 00D #12
.BY 6 'BLK' & 'NAME' 9 'TYP'
.BY 9 00D 0
.TWORK .BY 001 012 ' F3 '
.BY 092 09F ' -> PUFFER '
.BY ' ' 001 012 ' F7 ' 092 09F
.BY ' -> DISKETTE' 00D
.BY 001 012 ' F5 '
.BY 092 09F ' PUFFER -> '
.BY 2 001 012 2 05F 020 092
.BY 09F ' MENUE' 000
.TSCRATCH .BY 'RESTLICHE PROGRAMME '
.BY 'IM PUFFER' 00D ' '
.BY 'SCRATCH : ' 001 012
.BY ' 0 ' 092 09F
.BY ' -- SONST RETURN' 000
.INSTEXT .BY 0 0 003 012 001
.BY '-----'
.BY 0 0 0 0 0 0 0 0 0 0 0 0

```

```

/// ASSEMBLER REPORT ///
ANZAHL DER FEHLER : 0000
PROGRAMMZAehler : 120D
ENDADRESSE SPEICHER : 120D
ENDE DER SYMBOLTAS. : 0A96

```

Direkt-Datei-relativ-richtig

„Beim Erstellen eines Datenbankprogramms störte mich die komplizierte Programmierung von relativen Dateien mit dem Commodore 64,“ schreibt RUN-Autor Thomas Stüwe. „Aus dieser Not heraus“ hat er 'REL.HELP' geschrieben. Das Programm bietet einige neue Befehle, die den Umgang mit dieser Dateiform erheblich vereinfachen. Nicht vereinfachen kann es die theoretische Vorarbeit, die bei dieser Art von Programm unumgänglich ist.

Ein Datenbankprogramm ohne Direktzugriff ist undenkbar. Doch mit relativen Dateien umzugehen, ist sehr schwer. Mit 'REL.HELP' geht's leicht

soll. Arbeitet man mit einer sequentiellen Datei, muß man entweder die gesamte Datei in den Rechner laden oder Datensätze überlesen. Das heißt, wenn man auf den letzten Datensatz zugreifen will, muß man vorher die Datensätze 1-499 überlesen.

Der Nachteil dieser Dateiform liegt auf der Hand: Um Dateien vollständig einzulesen, reicht der Speicherplatz meist nicht aus. Muß man viele Datensätze überlesen, ergibt sich hieraus eine extrem lange Zugriffszeit.

Anders bei einer relativen Datei. Hier kann sofort auf jeden beliebigen Datensatz zugegriffen werden. Man kann also sofort den Datensatz Nr. 500 einlesen, ohne sich um die restlichen Sätze zu kümmern. Es wird kaum Speicherplatz verbraucht, da keine unnötigen Datensätze eingelesen werden müssen. Außerdem kann man auf einer Diskette 167 132 Bytes mit

Prinzip der relativen

Datenspeicherung

Die relative Datei ist eine sehr komfortable Methode zur Datenspeicherung. Sie wird im Commodore Handbuch nicht erklärt, da

der C 64 diesen Dateityp nicht unterstützt.

Um das Prinzip der relativen Datenspeicherung zu erklären, nehmen wir an, eine Datei bestehe aus 500 Datensätzen, wobei jeder Satz eine Länge von 50 Bytes haben

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|
| C170 | 20 | FD | AE | 20 | 8A | AD | 20 | F7 | B8 | C220 | AE | 20 | 8A | AD | 20 | F7 | B7 | 20 | 2B |
| C178 | B7 | A2 | 0F | 20 | C9 | FF | A9 | 50 | 27 | C228 | B1 | C1 | 20 | 79 | C1 | A2 | 01 | 20 | 98 |
| C180 | 20 | D2 | FF | A9 | 02 | 20 | D2 | FF | 59 | C230 | C9 | FF | A9 | FF | 20 | D2 | FF | 20 | A2 |
| C188 | A5 | 14 | 20 | D2 | FF | A5 | 15 | 20 | 2E | C238 | CC | FF | A9 | 01 | 20 | C3 | FF | A2 | 2F |
| C190 | D2 | FF | A9 | 01 | 20 | D2 | FF | 20 | A8 | C240 | 0F | 20 | C6 | FF | 20 | E4 | FF | C9 | 29 |
| C198 | CC | FF | 60 | 20 | FD | AE | 20 | 57 | 25 | C248 | 8D | D0 | F9 | 20 | CC | FF | A9 | 0F | 7D |
| C1A0 | E2 | 8D | C0 | 02 | 86 | 57 | 84 | 58 | 6D | C250 | 20 | C3 | FF | 60 | 20 | FD | AE | 20 | AE |
| C1A8 | 20 | FD | AE | 20 | 9E | B7 | 8E | C1 | 65 | C258 | 8A | AD | 20 | F7 | B7 | 20 | FD | AE | EC |
| C1B0 | 02 | A2 | 00 | A0 | 00 | B1 | 57 | 99 | CF | C260 | 20 | 57 | E2 | C9 | 00 | D0 | 03 | 4C | CA |
| C1B8 | C2 | 02 | C8 | CC | C0 | 02 | D0 | F5 | 53 | C268 | 48 | B2 | 8D | C0 | 02 | 86 | 57 | 84 | B3 |
| C1C0 | A9 | 2C | 99 | C2 | 02 | C8 | C8 | 99 | BD | C270 | 58 | 20 | 79 | C1 | A2 | 01 | 20 | C9 | E8 |
| C1C8 | C2 | 02 | 88 | A9 | 4C | 99 | C2 | 02 | 11 | C278 | FF | A0 | 00 | B1 | 57 | 20 | D2 | FF | 8E |
| C1D0 | C8 | C8 | AD | C1 | 02 | 99 | C2 | 02 | 15 | C280 | C8 | CC | C0 | 02 | D0 | F5 | A9 | 0D | 30 |
| C1D8 | AD | C0 | 02 | 69 | 03 | 8D | C0 | 02 | BB | C288 | 20 | D2 | FF | 20 | CC | FF | 60 | 20 | 3A |
| C1E0 | A9 | 0F | A2 | 08 | A0 | 0F | 20 | BA | 2A | C290 | 70 | C1 | AD | C1 | 02 | E9 | 01 | 48 | 6F |
| C1E8 | FF | A9 | 00 | 20 | BD | FF | 20 | C0 | 54 | C298 | A2 | 01 | 20 | C6 | FF | 20 | FD | AE | 28 |
| C1F0 | FF | A9 | 01 | A2 | 08 | A0 | 02 | 20 | 00 | C2A0 | 20 | 8B | B0 | 85 | 49 | 84 | 4A | 20 | B0 |
| C1F8 | BA | FF | AD | C0 | 02 | A2 | C2 | A0 | 2B | C2A8 | A3 | B6 | 68 | 20 | 75 | B4 | A0 | 02 | 94 |
| C200 | 02 | 20 | BD | FF | 20 | C0 | FF | 60 | 9F | C2B0 | B9 | 61 | 00 | 91 | 49 | 88 | 10 | F8 | A1 |
| C208 | 20 | FD | AE | 20 | 57 | E2 | 8D | C0 | F4 | C2B8 | C8 | 20 | 12 | E1 | 91 | 62 | C8 | C4 | 0C |
| C210 | 02 | 86 | 57 | 84 | 58 | 20 | FD | AE | D6 | C2C0 | 61 | D0 | F6 | 20 | CC | FF | 60 | 0D | 87 |
| C218 | 20 | 9E | B7 | 8E | C1 | 02 | 20 | FD | 8D | | | | | | | | | | |

Listing 1: Hexdump von REL.HELP

```

90 FOR I=49520 TO 49910                1837
100 READ A:S=S+A:POKE I,A:NEXT        1620
110 IF S<>51629 THEN PRINT "FEHLER IN DATA- 3769
    ZEILEN!":END
120 :
130 REM PROGRAMM DIREKT ABSPEICHERN
140 :
150 PRINT "[DOWN]LEGEN SIE EINE DISKETTE EIN!" 2726
160 POKE 198,0:WAIT 198,1            1588
170 :
180 SYS 49864:REM ABSPEICHERN        1058
190 :
200 PRINT "[DOWN]DAS PROGRAMM KANN JETZT MIT:" 2711
210 PRINT "[DOWN]LOAD";CHR$(34);"REL.HELP 3985
    1.0";CHR$(34);",8,1"
220 PRINT "[DOWN]GELADEN WERDEN!"      1838
230 :
299 :
300 DATA 32,253,174,32,138,173,32,247,183,162, 6676
    15,32,201,255,169,80,32,210
301 DATA 255,169,2,32,210,255,165,20,32,210, 7214
    255,165,21,32,210,255,169,1,32
302 DATA 210,255,32,204,255,96,32,253,174,32, 7344
    87,226,141,192,2,134,87,132,88
303 DATA 32,253,174,32,158,183,142,193,2,162, 6974
    0,160,0,177,87,153,194,2,200
304 DATA 204,192,2,208,245,169,44,153,194,2, 6754
    200,200,153,194,2,136,169,76,153
305 DATA 194,2,200,200,173,193,2,153,194,2, 6954
    173,192,2,105,3,141,192,2,169,15
306 DATA 162,8,160,15,32,186,255,169,0,32,189, 7953
    255,32,192,255,169,1,162,8,160
307 DATA 2,32,186,255,173,192,2,162,194,160,2, 6795
    32,189,255,32,192,255,96,32
308 DATA 253,174,32,87,226,141,192,2,134,87, 7683
    132,88,32,253,174,32,158,183,142
309 DATA 193,2,32,253,174,32,138,173,32,247, 7209
    183,32,177,193,32,121,193,162
310 DATA 1,32,201,255,169,255,32,210,255,32, 7166
    204,255,169,1,32,195,255,162,15
311 DATA 32,198,255,32,228,255,201,13,208,249, 6689
    32,204,255,169,15,32,195,255
312 DATA 96,32,253,174,32,138,173,32,247,183, 7176
    32,253,174,32,87,226,201,0,208
313 DATA 3,76,72,178,141,192,2,134,87,132,88, 7980
    32,121,193,162,1,32,201,255,160
314 DATA 0,177,87,32,210,255,200,204,192,2, 6909
    208,245,169,13,32,210,255,32,204
315 DATA 255,96,32,112,193,173,193,2,233,1,72, 7043
    162,1,32,198,255,32,253,174
316 DATA 32,139,176,133,73,132,74,32,163,182, 6667
    104,32,117,180,160,2,185,97,0
317 DATA 145,73,136,16,248,200,32,18,225,145, 6567
    98,200,196,97,208,246,32,204
318 DATA 255,96,234,169,112,133,251,169,193, 7152
    133,252,162,8,32,186,255,169,14
319 DATA 162,233,160,194,32,189,255,169,251, 6525
    162,199,160,194,32,216,255,96
320 DATA 234,64,58,82,69,76,46,72,69,76,80,32, 5415
    49,46,48

```

Listing 2: Basic-Lader von REL.HELP

relativen Daten belegen. Ein Datensatz von 50 Bytes kann also über 3300mal abgespeichert werden!

Wie arbeitet man mit relativen Dateien?

1. Datei anlegen (auf der Diskette wird Platz für die Daten reserviert)
2. Datei öffnen
3. Datensatzzeiger positionieren (Angabe, welchen Datensatz man bearbeiten will)
4. Datensatz schreiben oder lesen
5. Datei schließen (wenn die Arbeit mit der Datei beendet ist)

Auf den ersten Blick sieht das sehr einfach aus. Da der C64 aber über keine speziellen Datei-befehle verfügt, muß man sich mit chr\$()-Anweisungen abquälen. Bei der Datei von 500 Datensätzen wären folgende Eingaben nötig, um die Datei anzulegen, zu öffnen, den Datensatz 100 zu lesen und die Datei wieder zu schließen:

1. Datei anlegen:
open15,8,15:open1,8,2,"test-datei,1,"+chr\$(51)
print#15,"P"+chr\$(2)+chr\$(244)+chr\$(1)+chr\$(1)
print#1,chr\$(255):close1:close15
2. Datei öffnen, Datensatzzeiger positionieren, Datensatz lesen:
open15,8,15:open1,8,2,"test-datei,1,"+chr\$(51)
print#15,"P"+chr\$(2)+chr\$(100)+chr\$(0)+chr\$(1)
input#1,a\$
3. Datei schließen:
close1:close15

Erschwerend kommt noch hinzu, daß mit dem Input-Befehl nur maximal 88 Zeichen eingelesen werden können. Das heißt, daß längere Datensätze über eine Get-Schleife gelesen werden müssen. Außerdem darf in dem Datensatz kein Komma oder Doppelpunkt enthalten sein.

„REL.HELP“ ist ein Programm, das mit fünf neuen Befehlen die Verwaltung einer relativen Datei wesentlich vereinfacht. Sie werden mit SYS und nachfolgenden Parametern aufgerufen:

1. Anlegen einer relativen Datei:
SYS 49672,"Name",L,AN
L = Länge des Datensatzes
AN = Anzahl der Datensätze, die die Datei enthalten soll
2. Öffnen der relativen Datei:

SYS 49563,"Name",L
L = Länge des Datensatzes

3. Datensatz schreiben:

SYS 49748,N,AS

N = Datensatz-Nummer

AS = Datensatz

4. Datensatz lesen:

SYS 49807,N,AS

N = Datensatz-Nummer

AS = Datensatz in AS

5. Schließen der Datei

SYS 49722

Beim Lesen des Datensatzes können maximal 255 beliebige Bytes eingelesen werden, auch Kommas und Doppelpunkte.

Wichtig: Zusätzlich wird nach jedem Datensatz ein Return-Zeichen (CHR\$(13)) geschrieben. Bei Datensätzen von 50 Bytes Länge muß die Datei also mit L = 51 angelegt werden.

Die Befehlsfolge (oben ohne neue Befehle) verkürzt sich wie folgt:

```

100 RA=49672:RO=49563:RW=49748:RR=49807:RC=49722 4894
!10 PRINT "[CLR,RVS]DATEI ANLEGEN[DOWN]" 1973
120 INPUT "DATEINAME";N$:IF N$="" THEN 120 2482
130 INPUT "DATENSATZLAENGE (MIT CHR$(13))";L:IF 5928
L<1 OR L>255 THEN 130
140 INPUT "ANZAHL DER DATENSATZE";AN:IF AN<1 OR 4944
AN>167132/L THEN 140
150 SYS RA,N$,L,AN:REM ANLEGEN 1273
160 :
170 PRINT "[RVS,DOWN]DATEN EINGEBEN[DOWN]" 1841
180 SYS RO,N$,L:REM OPEN 947
190 FOR I=1 TO AN 875
200 PRINT "SATZ" I;:INPUT S$:IF LEN (S$)<>L-1 4405
THEN PRINT "FALSCH LAENGE!":GOTO 200
210 SYS RW,I,S$:REM WRITE 1081
220 NEXT 130
230 SYS RC:REM CLOSE 544
240 :
250 PRINT "[RVS,DOWN]DATEN LESEN[DOWN]" 1491
260 SYS RO,N$,L:REM OPEN 947
270 FOR I=1 TO AN 875
280 SYS RR,I,S$:REM READ 1061
290 PRINT "SATZ" I "[2SPACES]" S$ 1180
300 NEXT 130
310 SYS RC:REM CLOSE 544

```

Listing 3: Demoprogramm zu REL.HELP

```

10 :SYS34304
20 :
30 :
40 #=#C170
50 :
70 : REL.HELP 1.0
80 :
85 :----- SATZZEIGER POSITIONIEREN -----
86 :
99 L100
100 JSR $A9FD :CHKCOM PRUEFT AUF KOMMA
105 JSR $AD9A :FRMNUM AUSDRUCK HOLEN
110 JSR $B7F7 :GETADR 16-BIT WERT HOLEN
                $14=LO-BYTE $15=HI-BYTE

114 L115
115 LDX #$0F :15
120 JSR $FFC9 :AUSGABE AUF #15
125 LDA #$50 :ASC("P")
130 JSR $FFD2 :PRINT
135 LDA #$02 :CHR$(02)
140 JSR $FFD2 :PRINT
145 LDA $14 :CHR$(LB)
150 JSR $FFD2 :PRINT
155 LDA $15 :CHR$(HB)
160 JSR $FFD2 :PRINT
165 LDA #$01 :CHR$(1)
170 JSR $FFD2 :PRINT
175 JSR $FFCC :CLRCH AUSGABE WIEDER NORMAL
180 RTS :RUECKSPRUNG BASIC
181 :
182 :----- DATEI OEFFNEN -----
183 :
185 JSR $A9FD :CHKCOM PRUEFT AUF KOMMA
190 JSR $E257 :STRING HOLEN
195 STA $02C0 :LAENGE NACH 1704
200 STX $57
205 STY $58 :ADRESSE SPEICHERN
210 JSR $A9FD :CHKCOM
215 JSR $B79E :GETBYT HOLT 8-BIT WERT
220 STX $02C1 :SATZLAENGE NACH 1705
224 L225
225 LDX #$00
230 LDY #$00
234 L235
235 LDA ($57),Y :DATEINAME NACH 1706... SCHIEBEN
240 STA $02C2,Y
245 INY
250 CPY $02C0
255 BNE L235
260 LDA #$2C :L,CHR$(L) ZUFUEGEN
265 STA $02C2,Y
270 INY

275 INY
280 STA $02C2,Y
285 DEY
290 LDA #$4C
295 STA $02C2,Y
300 INY
305 INY
310 LDA $02C1
315 STA $02C2,Y
320 LDA $02C0
325 ADC #$03 :LAENGE DES NAMENS KORRIDIEREN +3
330 STA $02C0
335 LDA #$0F :OPEN15,0,15
340 LDX #$08
345 LDY #$0F
350 JSR $FFBA :SETFLS
355 LDA #$00
360 JSR $FFBD :SETNAM
365 JSR $FFC0 :OPEN
370 LDA #$01 :OPEN1,0,2
375 LDX #$08
380 LDY #$02
385 JSR $FFBA :SETFLS
390 LDA $02C0
395 LDX #$C2
400 LDY #$02
405 JSR $FFBD :SETNAM
410 JSR $FFC0 :OPEN
415 RTS :RUECKSPRUNG BASIC

416 :
417 :----- DATEI ANLEGEN -----
418 :
420 JSR $A9FD :CHKCOM PRUEFT AUF KOMMA
425 JSR $E257 :STRING HOLEN
430 STA $02C0 :LAENGE NACH 1704
435 STX $57
440 STY $58 :ADRESSE SPEICHERN

445 JSR $A9FD :CHKCOM PRUEFT AUF KOMMA
450 JSR $B79E :GETBYT HOLT 8-BIT WERT
455 STX $02C1 :DATENSATZLAENGE NACH 1705
460 JSR $A9FD :CHKCOM PRUEFT AUF KOMMA
465 JSR $AD9A :FRMNUM AUSDRUCK HOLEN
470 JSR $B7F7 :GETADR 16-BIT WERT
475 JSR L225 :OEFFNEN DER DATEI
480 JSR L115 :AUF LETZTEN DATENSATZ POSITIONIEREN
485 LDX #$01 :11
490 JSR $FFC9 :CHKOUT AUSGABE AUF #1
495 LDA #$FF
500 JSR $FFD2 :PRINT#1,CHR$(255)
505 JSR $FFCC :CLRCH
510 LDA #$01

```

```

515 JSR $FFC3 :CLOSE 1
520 LDX #0F
525 JSR $FFC6 :CHKIN EINGABE AUF #15
529 L530 :FEHLERKANAL AUSLESEN
:FEHLER 50 UNTERDRUECKEN

530 JSR $FFE4 :GET
535 CMP #0D :CHR$(13)PRINT
540 BNE L530
545 JSR $FFCC :CLRCH
550 LDA #0F
555 JSR $FFC3 :CLOSE 15
560 RTS :RUECKSPRUNG BASIC
561 :
562 :----- DATENSATZ SCHREIBEN -----
563 :
565 JSR $AEFD :CHKCOM PRUEFT AUF KOMMA
570 JSR $AD8A :FRMNUM AUSDRUCK
575 JSR $B7F7 :OETADR 16-BIT
580 JSR $AEFD :CHKCOM
585 JSR $E257 :STRING HOLEN
590 CMP #00 :LAENGE=0PRINT
595 BNE L605
600 JMP $B248 :ILLIGAL QUANTITY ERROR
604 L605
605 STA $02C0 :LAENGE NACH 1704
610 STX #57
615 STY #58 :ADRESSE SPEICHERN
620 JSR L115 :ZEIGER POSITIONIEREN
625 LDX #01 :11
630 JSR $FFC9 :CHKOUT AUSGABE AUF #1
635 LDY #00
639 L640
640 LDA (#57),Y
645 JSR $FFD2 :STRIND AUF #1 AUSGEBEN
650 INY
655 CPY $02C0
660 BNE L640
665 LDA #0D :CHR$(13)

670 JSR $FFD2 :AUSGEBEN
675 JSR $FFCC :CLRCH
680 RTS :RUECKSPRUNG BASIC
681 :

682 :----- DATENSATZ LESEN -----
683 :
685 JSR L100 :WERTE HOLEN + ZEIGER POSITIONIEREN
690 LDA $02C1
695 SBC #01 :DATENSATZLAENGE-1 OHNE CHR$(13)
700 PHA :STACK SPEICHERN
705 LDX #01 :11
710 JSR $FFC6 :CHKIN EINGABE AUF #1
715 JSR $AEFD :CHKCOM PRUEFT AUF KOMMA
720 JSR $B88B :STRING VORBEREITEN
725 STA #49

730 STY #4A
735 JSR $B6A3 :FRESTR
740 PLA
745 JSR $B475 :STRRES STRING RESERVIEREN
750 LDY #02
754 L755
755 LDA $0061,Y
760 STA (#49),Y
765 DEY
770 BPL L755

775 INY
779 L780
780 JSR $E112 :BASIN
785 STA (#62),Y
790 INY
795 CPY #61
800 BNE L780
805 JSR $FFCC :CLRCH
810 RTS :RUECKSPRUNG BASIC
811 :
815 :-----
820 .END

```

Assemblerlisting von REL.HELP

1. Anlegen: SYS49672, "test-datei", 51,500
2. Öffnen: SYS49563, "test-datei", 51
3. Satz 100 lesen: SYS49807, 100, a\$
4. Schließen: SYS49722

Bei relativen Dateien ist zu beachten, daß die Datensatzlänge immer gleich sein muß. Hat man die Datei für Datensätze mit einer Länge von 51 Bytes (50 Bytes + Chr\$(13)) angelegt, so muß ein Datensatz mit Leerzeichen aufgefüllt werden, falls er kürzer als

50 Bytes ist.

Listing 3 enthält ein kurzes Demoprogramm, das die Funktion der einzelnen Befehle zeigt.

Das Programm „REL.HELP“ hat einen Umfang von 343 Bytes und belegt den freien Speicherbereich von \$C170 — \$C2C6 (49620-49862). Zur Zwischenspeicherung werden die Speicherstellen 704-720 benutzt. File #1 und #15 sind bei der Arbeit mit „REL.HELP“ geöffnet und dürfen nicht anderweitig

verwendet werden.

Das Programm (Listing 1) muß mit dem Checker eingegeben werden. Als Startadresse ist C170 anzugeben. Statt des Checker-Listings kann auch der Basic-Lader in Listing 2 eingegeben werden. Er speichert das fertige Programm auf Diskette ab.

„REL.HELP“ wird mit Load „8,1“ geladen. Danach ist NEW einzugeben.

(Thomas Stüwe)

Ihre Spende = 100% Hilfe!

»Brot für die Welt« leistet wirksame Hilfe zur Selbsthilfe. Seit 25 Jahren. Spenden für »Brot für die Welt« werden in voller Höhe für Entwicklungsprojekte und Soforthilfemaßnahmen in Entwicklungsländern verwandt. Die anfallenden Kosten für Verwaltung oder Werbung werden nicht aus Spendenmitteln, sondern durch Zinsen

bzw. aus Kirchensteuern finanziert. Bitte helfen Sie mit Ihrer Spende, damit die Arbeit fortgesetzt werden kann. Auch kleine Summen können eine große Hilfe bedeuten. Aktion »Brot für die Welt«
Staffenbergstraße 76
7000 Stuttgart 1
Konto: 500 500-500
beim Postscheckamt Köln
(BLZ 370 100 50)



Zusatztastatur für professionelles Arbeiten

Anbausatz im Commodore-Design passend zu C 16, C 20 und C 64.

Technische Daten:

- +/- Tasten
- große Return-Taste
- bewährte CHERRY-Tasten (16 Mio. Betätigt.)
- problemloser Anschluß

DM 149,- inkl. Mwst., Versandpausch. 6,-
Zahlbar per Scheck o. Nachnahme
Bei Bestellung Farbe und Rechnerotyp angeben.

apd

Uhlandstr. 4 · 7209 Denkingen · Tel. 07424/84204

Kartei auf Kassette

Die Datasette ist das Stiefkind der Dateiverwaltung. Die einschlägigen Programme arbeiten fast alle nur mit der Floppy. Der Grund: Kassetten erlauben keinen direkten Zugriff. Um an Daten am Ende des Bandes heranzukommen, muß der vordere Teil erst überlesen werden. Raffinessen wie relative Dateien sind daher auf die Floppy beschränkt.

RUN-Programmierer Matthias Richter hat sich von diesen Problemen nicht entmutigen lassen. Sein Dateiprogramm verwaltet bis zu 200 Datensätze. Anzahl und Inhalt der Felder sind frei wählbar. Zu Beginn wird gefragt, ob eine bereits geschriebene Datei benutzt werden soll, oder ob man eine neue Datei eröffnen will. Beim Anlegen einer neuen Datei wird erst nach der Anzahl der Felder und nach ihren Namen gefragt. Bis zu zehn Felder sind erlaubt. Diese Angaben werden in einer kurzen Hilfsdatei namens „VORSPANN“ abgelegt.

Daten eingeben und ansehen

Soll eine bereits vorhandene Datei benutzt werden, wird die Hilfsdatei erst eingelesen.

Nun erscheint das Hauptmenü:

- 1 Daten schreiben
- 2 Daten anfügen
- 3 Daten lesen
- 4 Weitere Daten lesen
- 5 Daten ansehen
- 6 Daten ändern
- 7 Daten löschen
- 8 Exit

Daten schreiben:

Hier werden die Datensätze eingegeben. Die Namen aller Felder werden angezeigt. Darunter ist jeweils der gewünschte Text einzugeben. Danach wird die Datei unter dem Namen „CBM 64 DATEI“ abgespeichert und ins Untermenü

Floppydateien gibt's viele. Kassettenbenutzer werden schlechter bedient. Ein Basicprogramm macht die Datasette zur Datenbank.

„Daten ansehen“ gesprungen (siehe unten).

Daten anfügen:

Zunächst wird geprüft, ob schon eine Datei im Speicher steht. Wenn nicht, wird sie erst eingelesen. Dann können weitere Datensätze wie unter Menüpunkt 1 eingegeben werden. Zum Schluß wird die Datei wieder abgespeichert und ins Hauptmenü zurückgekehrt.

Daten lesen:

Die Datei wird in den Speicher gelesen. Danach wird ins Untermenü „Daten ansehen“ gesprungen.

Weitere Daten lesen:

An die Daten im Speicher wird eine weitere Datei angehängt. Dadurch besteht die Möglichkeit, eine Datei in mehreren Teilen anzulegen. Sie können bei Bedarf aneinandergelinkt werden, sind aber auch alleine ladbar. Man braucht also nur die gerade benötigte Teildatei einzulesen und spart sich die langen Ladezeiten für die restlichen Daten.

Daten ansehen:

Es erscheint ein Untermenü mit den folgenden Punkten:

- 1 Hauptmenü
- 2 Daten ansehen:
- 3 Gesamte Daten
- 4 Ein Datensatz
- 5 Verschiedene Datenfelder
- 6 ... (Name von Feld 1)
- 7 ... (Name von Feld 2)

.....
Mit Taste 1 wird ins Hauptmenü zurückgekehrt. Mit den übrigen

Tasten kann die Datei im Speicher nach verschiedenen Gesichtspunkten betrachtet werden. Zur Auswahl stehen: Ansehen der Gesamtdatei (2), eines einzelnen Datensatzes (3), eines einzelnen Feldes von allen Datensätzen (5-.) und einer beliebigen Kombination von Feldern von allen Datensätzen. Nach der Ausgabe wird immer ins Menü „Daten ansehen“ zurückgesprungen.

Daten ändern:

Das Untermenü bietet folgende Möglichkeiten:

- 1 Hauptmenü
- 2 Abspeichern
- 3 Kompletter Datensatz
- 4 ... (Name von Feld 1)
- 5 ... (Name von Feld 2)

.....
Taste 1 führt wieder ins Hauptmenü zurück.

| | | |
|-------|---------|--|
| 2 | - 7 | Abfrage, ob schon ein Vorspann existiert |
| 15 | | Dimensionieren |
| 20 | - 100 | Hauptmenü |
| 150 | | Ende |
| 200 | - 310 | Menü Daten ansehen |
| 350 | - 447 | Menü Daten ändern |
| 450 | - 460 | Auswahl des zu ändernden Datensatzes |
| 480 | - 500 | Ändern von Einzeldaten |
| 700 | - 740 | Ändern eines gesamten Datensatzes |
| 850 | - 890 | Daten abspeichern (Text) |
| 1000 | - 1040 | Unterprogramm zur Eingabe eines Datensatzes |
| 1100 | - 1150 | Unterprogramm zum Daten abspeichern |
| 1200 | - 1280 | Unterprogramm zum Daten lesen |
| 1500 | - 1700 | Daten anfügen |
| 1500 | - 1540 | Einlesen von Daten, wenn noch keine im Speicher sind |
| 1550 | - 1650 | Eingabe neuer Daten |
| 1660 | - 1700 | Abspeichern |
| 1800 | - 1820 | Daten einlesen (Text) |
| 1850 | - 1970 | Daten schreiben + abspeichern |
| 2000 | - 2030 | Einen Datensatz ausgeben |
| 2350 | - 2390 | Ausgabe der gesamten Daten |
| 2500 | - 2610 | Einlesen des Vorspanns |
| 2650 | - 2790 | Eingabe des Vorspanns und Abspeichern |
| 2800 | - 2810 | Einlesen einer weiteren Datei |
| 3000 | - 3140 | Ansehen verschiedener Datenfelder |
| 3200 | - 3250 | Ansehen eines bestimmten Datenfeldes |
| 3300 | - 3660 | Daten löschen |
| 3300 | - 3350 | Menü Daten löschen |
| 3400 | - 3540 | Eingabe der Nummern der zu löschenden Sätze |
| 3550 | - 3580 | Sortieren der Nummern der zu löschenden Sätze |
| 3590 | - 3650 | Löschen |
| 10000 | - 10030 | Unterprogramm für Dateiende |

Nach Drücken von 3 wird ein ganzer Datensatz geändert.

Taste 4 und folgende ermöglichen die Änderung einzelner Felder innerhalb eines bestimmten Satzes.

Unter Punkt 2 wird die geänderte Datei neu abgespeichert.

Alle Menüpunkte führen wieder ins Änderungs Menü zurück.

Daten löschen:

Folgende Unterpunkte stehen zur

Auswahl:

- 1 Hauptmenü
- 2 Daten löschen
- 3 Abspeichern

Hier können einzelne Datensätze aus der Datei im Speicher gelöscht werden (2) und die geänderte Datei wieder abgespeichert werden (3). Mit Taste 1 geht's wieder zurück ins Hauptmenü.

Exit:

Das Programm wird beendet.

Programmbeschreibung

Wichtige Variablen:

a\$ (Feldnummer/Satznummer):

Dateninhalt

b\$ (Feldnummer): Feldname

i: Anzahl der Datensätze

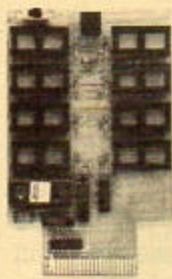
zz: Anzahl der Felder

(Matthias Richter)

| | | | |
|--|---|---|---|
| <pre> 1 poke 53280,11:poke 53281,11:print "[clr, cbm 7]" chr# (14) 2 print "Wollen Sie bereits geschriebene Daten 3 print "[down]benutzen (b), oder wollen Sie eine neue 4 print "[down]Datei eroeffnen (n)?" 5 get s\$:if s\$="n" then 2650 6 if s\$="b" then 2500 7 goto 5 15 dim a\$(zz,200) 19 rem *** hauptmenue *** 20 print "[clr,12spaces]Menue: 30 print "[2down]- Daten schreiben 40 print "[down]2- Daten anfragen 50 print "[down]3- Daten lesen 60 print "[down]4- Weitere Daten lesen 65 print "[down]5- Daten ansehen 66 print "[down]6- Daten aendern 70 print "[down]7- Daten loeschen 80 print "[down]8- Exit 90 get a\$:a=val (a\$):if a<1 or a>8 then 90 100 on a goto 1850,1500,1800,2800,200,350, 3300,150 150 poke 53281,246:poke 53280,254:print "[pur, clr]"end 199 rem *** daten ansehen *** 200 print "[clr,12spaces]Menue: 210 print "[down]1- Hauptmenue 220 print "[down,3spaces]Daten ansehen: 230 print "[down]2- Gesamte Daten 240 print "[down]3- Ein Datensatz 250 print "[down]4- Verschiedene Datenfelder 260 for y=1 to zz 270 print y+4"- "b\$(y) 280 next 300 input a\$:a=val (a\$):if a<1 or a>zz+4 then 300 305 if a>4 then 3200 310 on a goto 20,2350,2000,3000 349 rem *** daten aendern *** 350 print "[clr,12spaces]Menue: 360 print "[down]1- Hauptmenue 370 print "[down]2- Abspeichern 380 print "[down]3- Kompletter Datensatz 390 for t=1 to zz:print t+3;b\$(t):next 440 input a\$:a=val (a\$):if a<1 or a>zz+3 then 440 445 if a=1 then 20 447 if a=2 then 850 450 input "[clr]Welche Satznummer?"a\$ 460 e=val (e\$):if e<1 or e>1 then 450 470 if a=3 then 700 479 rem *** einzeldaten *** 480 print "[clr]Bisher:[down]"a\$(a-3,e) 490 input "[down]Neu?"a\$(a-3,e) 500 goto 350 699 rem *** Kompletter datensatz *** 700 print "[clr]Bisher: 710 for t=1 to zz:print a\$(t,e):next 720 print "[down]Neu: 730 gosub 1800 </pre> | <pre> 3272 3479 3645 2470 1789 1528 243 1057 1314 1995 2019 1961 2872 2023 1828 2202 1094 2875 3460 3054 1314 1814 1856 2055 1995 2938 945 1532 130 3504 1864 1868 1314 1814 1392 2979 2306 3704 812 846 2251 2260 1012 1997 1743 580 953 1950 803 563 </pre> | <pre> 740 goto 350 849 rem *** daten abspeichern *** 850 print "[clr,rvs]Zurueckspulen!![off]" 860 print "[16spaces]Taste 870 get a\$:if a\$="" then 870 880 gosub 1100 890 goto 350 999 rem *** eingabe eines datensatzes *** 1000 for t=1 to zz 1010 print "[down]"b\$(t)": 1020 input a\$(t,e) 1030 next 1040 return 1099 rem *** daten abspeichern *** 1100 open 1,1,1,"cbm 64 date1" 1110 for y=1 to i:for yy=1 to zz 1120 print#i,a\$(yy,y) 1130 next yy,y 1140 close 1 1150 return 1199 rem *** daten lesen *** 1200 open 1,1,0,"cbm 64 date1" 1210 i=i+1 1220 for s=1 to zz 1230 input#i,a\$(s,i) 1240 next 1250 if st and 64 then 1270 1260 goto 1210 1270 close 1 1280 return 1500 if i>0 then 1550 1509 rem *** noch keine daten im speicher *** 1510 print "[clr]Die Daten werden erst eingelesen 1520 print "[19down,3spaces,rvs,5spaces]Taste [4spaces,off]" 1530 poke 198,0:wait 198,1:i=0:print "[clr]" 1540 gosub 1200 1549 rem *** daten anfragen *** 1550 print "[clr]Eingabe der neuen Daten:"ie=i 1560 e=e+1:print "[2down]Satznummer:"e 1570 gosub 1000 1580 print "O.K. (j/n)?" 1590 get r\$:if r\$="j" then 1620 1600 if r\$="n" then 1570 1610 goto 1590 1620 print "[down]Noch mehr Daten (j/n)?" 1630 get r\$:if r\$="j" then 1560 1640 if r\$="n" then 1660 1650 goto 1630 1659 rem *** daten abspeichern *** 1660 print "[clr]Ggf. zurueckspulen!! 1670 print "[19down,3spaces,rvs,5spaces]Taste [4spaces,off]" 1680 poke 198,0:wait 198,1:print "[clr]" 1690 i=e:gosub 1100 1700 goto 20 1799 rem *** daten lesen *** 1800 print "[clr]Spulen Sie bis zum Anfang der Datei!"i=0: 1805 print "[2down,10spaces]Taste":poke 198,0: </pre> | <pre> 580 2320 1096 1540 567 580 935 884 903 130 142 1720 2000 1151 655 258 142 1688 315 933 876 130 1481 575 258 142 1051 3225 2280 2238 571 3425 1796 563 1351 1794 1682 851 2696 1763 1680 807 2579 2280 1993 806 429 4116 2697 </pre> |
|--|---|---|---|

| | | | |
|--|--|---|--|
| <pre> wait 198,1 1810 gosub 1200 1820 goto 200 1849 rem *** daten schreiben *** 1850 print "[clr]":i=1:e=1 1860 print "[down]Satznummer:"i 1870 gosub 1000 1880 print "o.k. (j/n)? 1890 get a\$:if a*="j" then 1920 1900 if a*="n" then 1870 1910 goto 1890 1920 print "Weitere Daten (j/n)? 1930 get a\$:if a*="j" then i=i+1:e=i:goto 1860 1940 if a*="n" then 1960 1950 goto 1930 1954 rem *** daten abspeichern *** 1955 print "[2down]Spulen Sie das Band nun so, dass ich 1956 print "[down]die Datei abspeichern kann. 1957 print "[2down,1spaces]Taste:"poke 198,0: wait 198,1 1960 gosub 1100 1970 goto 200 1999 rem *** daten auf bildschirm: ein datensatz z *** 2000 input "[clr]Welche Satznummer"/q\$ 2010 q=val (q\$):if q<1 or q>1 then 2000 2020 for t=1 to zz:print b\$(t)":i:print a\$(t, q):next 2030 goto 10000 2349 rem *** daten ansehen: gesamte daten *** 2350 print "[clr]":for u=1 to i:print "[down] Satznummer:"u 2355 for y=1 to zz 2360 print a\$(y,u) 2370 next y 2380 poke 198,0:wait 198,1:next 2390 goto 10000 2499 rem *** einlesen des vorspannes *** 2500 print "[clr]Spulen Sie nun die Kassette bis zum 2510 print "[down]'Vorspann' und druecken Sie dann bitte 2520 print "[down]eine Taste 2530 poke 198,0:wait 198,1 2540 open 1,1,0,"vorspann" 2550 input#1,zz\$ 2555 zz=val (zz\$) 2560 z=z+1:input#1,b\$(z) 2560 if z=zz then close 1:goto 15 2610 goto 2560 2649 rem *** eingabe des vorspannes *** 2650 print "[clr]Anzahl der Datenfelder (max. 10) 2660 input "[down,17spaces]"/zz 2670 if zz<1 or zz>10 then 2650 2680 for z=1 to zz 2690 print "Name des Datenfeldes N.:"z 2700 input "[down,17spaces]"/b\$(z) 2710 next 2715 print "[clr]":for t=1 to zz:print b\$(t): next 2716 print "[2down,4spaces]O.K.[2spaces](j/n)? 2717 get r\$:if r*="j" then 2720 2718 if r*="n" then 2650 2719 goto 2717 2720 print "[clr,down]Spulen Sie das Band nun so, dass ich 2730 print "[down]den Vorspann abspeichern kann. dann 2740 print "[down]druecken Sie bitte eine Taste. 2745 zz\$=str\$(zz) 2750 get b\$:if b*="" then 2750 2759 rem *** abspeichern des vorspannes *** 2760 open 1,1,1,"vorspann" 2770 print#1,zz\$ 2780 for z=1 to zz:print#1,b\$(z):next 2785 close 1 2790 goto 15 2799 rem *** einlesen einer weiteren datei *** 2800 print "[clr]Spulen Sie bis zum Anfang der </pre> | <pre> 571 550 1379 1536 563 1341 1567 1654 663 2195 2906 1652 613 3458 2922 2931 567 558 2254 2086 2520 565 3046 945 1007 308 1819 565 3777 3906 1442 1588 1534 913 932 1243 1774 629 2817 1112 1679 947 2571 1463 130 2071 1583 1842 1677 709 3468 3593 2895 880 1698 1566 933 1852 258 447 5330 </pre> | <pre> anzufue-[down,2spaces]genden Datei" 2805 poke 198,0:wait 198,1 2810 gosub 1200:goto 200 2999 rem *** ansehen verschiedener datenfelder *** 3000 print "[clr]":for t=1 to zz:print t, b\$(t)":i:next 3010 input "Wieviele Datenfelder"/u 3020 if u<1 or u>zz then 3000 3030 if u=zz then 2350 3040 print "[down]Geben Sie nun nacheinander die Zahlen 3050 print "[down]der Datenfelder ein, die Sie sich 3060 print "[down]ansehen wollen. 3070 for o=1 to u:input w(o):next 3080 for y=1 to i 3090 for y2=1 to o-1 3100 print b\$(w(y2)):"[2spaces]"a\$(w(y2),y) 3110 next y2 3120 poke 198,0:wait 198,1 3130 next y1 3140 goto 10000 3199 rem *** ansehen eines bestimmten datenfeld es *** 3200 print "[clr]"b\$(a-4) 3210 for t=1 to i 3220 print a\$(a-4,t) 3230 poke 198,0:wait 198,1 3240 next 3250 goto 10000 3299 rem *** daten loeschen *** 3300 print "[clr,1spaces]Menue: 3310 print "1 - Hauptmenue 3320 print "2 - Daten loeschen 3330 print "3 - Abspeichern 3340 get a\$:a=val (a\$):if a=0 or a>3 then 3340 3350 on a goto 20,3400,850 3400 print "[down]Geben Sie nun nacheinander die Satz- 3410 print "[down]nummern der zu loeschenden Daten ein. 3420 input "[down]Satznummer"/kz 3430 if kz<1 or kz>i then print "[3up]":goto 3420 3435 goto 3500 3440 print "[down]Noch mehr? (j/n)" 3450 get a\$:if a*="j" then print "[7up]":goto 3420 3460 if a*="n" then 3550 3470 print "[up,rvs]Noch mehr? (j/n)[2up]":goto 3440 3499 rem *** Kontrolle der eingegebenen zahlen *** 3500 u=u+1:l(w)=kz:if u=1 then 3540 3510 for t=1 to u-1:if l(w)=l(t) then r=1 3520 next 3530 if r=1 then r=0:w=u-1:print "[3up]":goto 3420 3540 print "[down]"tab(3#w)l(w):goto 3440 3550 if u=1 then l(1)=kz:w=1:goto 3590 3554 rem *** sortieren nach groesse *** 3555 for tt=1 to u-1 3560 for t=1 to u-1 3570 if l(t)<l(t+1) then rr=l(t):l(t)=l(t+1): l(t+1)=rr 3580 next t,tt 3589 rem *** loeschen *** 3590 for o=1 to w:print ,l(o) 3600 kz=int (l(o)):for t=kz to i-1 3610 for tt=1 to zz 3620 a\$(tt,t)=a\$(tt,t+1) 3630 next tt,t 3640 i=i-1 3650 next o 3660 w=0:goto 20 9999 rem *** anzeige fuer dateiende *** 10000 print "Ende der Datei!" 10010 print "Druecken Sie nun 'u' zum weitermachen 10020 get a\$:if a*=""u" then 10020 10030 goto 200 </pre> | <pre> 1588 929 2227 2385 1497 1572 3013 2795 2021 1750 901 1296 2345 508 1588 504 565 1033 751 838 1588 130 565 1316 1806 2150 1384 2972 1426 2815 3567 1674 2406 583 1861 3171 1642 2866 2405 2521 130 3203 2035 2794 1189 1330 4232 705 1904 2472 889 2039 545 323 280 823 1471 4285 1791 558 </pre> |
|--|--|---|--|

SUPEREPROMKARTE 256K



Mehr Speicherkapazität als eine Diskette! Eine echte Alternative zum zweiten Diskettenlaufwerk. Sie können auf der 256-K-Epromkarte eine Directory anlegen und die Programme aus dieser aufrufen und direkt starten. Ein in der Steuerung eingebauter Modulgenerator macht alle Programme für die Karte lauffertig, sodaß der Anwender keine Spezialkenntnisse benötigt.

**PREIS incl. Steuereprom
129,— DM**

Technische Daten:

8 Steckplätze für 2764, 27128, 27128, 27256. Mitgeliefertes Steuereprom auf separatem Sockel. Betrieb am Expansionsport. Keine Eingriffe in den Rechner nötig!

Superepromkarte 8.3 95,— DM

Eine Epromkarte für 8x8K-Eproms. Jedes Eprom kann über die Funktionstasten angewählt werden. Viele Sonderfunktionen für Betriebssysteme, Basicinterpreter usw. Jetzt auch mit der Directoryfunktion und Modulgenerator wie die 256 K-Karte!

Technische Daten:

8 Steckplätze für 2732, 2764. Einer mit Steuereprom belegt. Steuerfunktionen sind in Basic- oder Maschinenprogramme einzubauen. Betrieb am Expansionsport ohne Eingriff in den Rechner.

DELA-Epromkarten: Epromkarte 2.1 19,50 DM

Für 2 Eproms 2716-2764 (oder 1x27128). Voll bestückt mit Sockel, Ein-/Ausschalter, Umschalter, Reset-Taste.

Universalkarte 4.1 29,— DM

Für 2x2764 oder 2x27128. Betriebsart als Epromkarte für 4 Modulprogramme à 8 K oder für vier Betriebssysteme. Mit dieser Karte können Sie Ihre neuen Betriebssysteme am Modulport ohne Eingriff in den Rechner betreiben.

Modulkarte Dela.MO 8,90 DM

Diese Karte wird als Leerplatine für ein Eprom (2732-27128) verkauft. Es sind je nach Bestückung 5 Betriebsarten möglich:

1. 8/16 K Epromkarte mit elektronischer Abschaltung.
2. 8/16 K Epromkarte mit elektr. Ein- und Ausschaltung
3. Externes Betriebssystem (Expansionsport)

Leergehäuse für DELA.MO **6,— DM**

5fach Betriebssystemschaltkarte 5/1. Für vier zusätzliche Betriebssysteme auf zwei 27128. Absturzfrei!

ohne Schalter **35,— DM**
mit Schalter **45,— DM**

2fach Betriebssystemschaltkarte 2/1. Für ein zusätzliches Betriebssystem auf 2764. Absturzfrei!

ohne Schalter **38,— DM**
Einbau der Karte in den Sockel des Kernaroms.

Profi-Betriebssystem 30,— DM

7x schneller laden, DOS auf F-Tasten, viele Sonderfunktionen mit CTRL-Taste. Auf 2764-Eprom für Universalkarte oder Umschaltkarten

Eprommer I

DELA-EPROMMER

Unser bewährter Eprommer für 2716-27128 (27256 nach Umbau). Alle Funktionen sind softwaregesteuert. Keine Einstellarbeiten! Betrieb am USER-Port ohne sep. Netzteil.

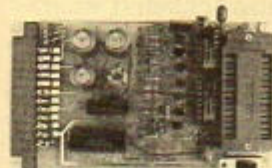
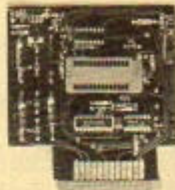
Funktionen:

Leertest, Auslesen, Monitor, Einzelbyteprogrammierung, Wiederholungsmethode zur Programmierung mehrerer gleicher Eproms. Schnelle Programmierung (8 K in 28 s) 50 ms/Byte ebenfalls möglich. Verify während der Programmierung.

Nur als Bausatz lieferbar:

mit Textoolsockel 199,— DM
ohne Textoolsockel 79,— DM

incl. Software auf Diskette. (Kass. 10,— DM Aufpreis)



DELA-Eprommer II 149,—

Unser neues Superding!!

Brennt 2716-27256 und alle pinkompatiblen C-MOS-Eproms. Programmierspannungen: 12 V, 21 V, 25 V (softwaregesteuert)

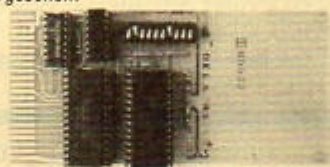
Zusätzlich zu den Funktionen von Eprommer 1 sind in der Software eingebaut: Eprom-Speichervergleich

Autostartgenerator für Basic- und Maschinenprogramme

Der Eprommer II ist nur als Fertigerät mit Textoolsockel lieferbar. Software auf Diskette.

16 K-RAM Modul 79,— DM

Eine Soft-Epromkarte für den Modulentwickler. Das RAM kann normal beschrieben werden. Die Funktion der Karte entspricht der einer Epromkarte für 8/16 K oder Betriebssystem. Ein Anschluß für eine Batterie oder Accu ist vorgesehen.



DELA-Modulgenerator 59,— DM

Der Modulgenerator macht aus Ihren Programmen (Basic oder Maschine) brennfertige Modulprogramme. Sie können bis 4 Programme auf ein Eprom brennen und diese über die F-Tasten auswählen. Für 8 K oder 16 K Eproms. Der Modulgenerator wird als Steckmodul mit Gehäuse geliefert.

Adapter/Stecker

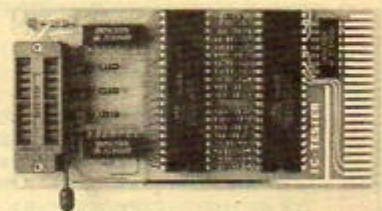
| | |
|--|---------|
| 3fach Userportadapter mit Reset-Taster. | 38,— DM |
| Winkeladapterplatine | 9,90 DM |
| Experimentierplatte für Modulport/Userport | 18,— DM |
| Steckplatzerweiterung | |
| komplett: | 99,— DM |
| Bausatz: | 69,— DM |
| Userportstecker | 5,— DM |
| Gehäuse hierfür | 4,— DM |
| Modulportstecker | 9,90 DM |

Module:

| | |
|----------------------------|---------|
| alle mit Gehäuse! | |
| Turbo-Tape | 39,— DM |
| Turbo-Disk | 39,— DM |
| Turbo-Modul 85 | 39,— DM |
| (T-Tape + T-Disk) | |
| S/4-Modul | 65,— DM |
| (T-Tape, T-Disk, DOS, OLD) | |

IC-Tester 134,— DM

Der IC-Tester testet alle TTL-IC bis 28-Pins auf korrekte Funktionen. Ein Erkennen von unbekanntem (abgeschliffenen) IC's ist möglich, da der Tester die Typen selbständig erkennt. Über die Dauertestfunktion können auch IC-Fehler festgestellt werden, die nur zeitweise auftreten. Eine sinnvolle Ergänzung Ihres Elektroniklabors. Der Tester ist mit einem 28poligen Textoolsockel bestückt. Software auf Diskette. Betrieb am Modulport.



Epromlöschgerät 115,— DM

Komplett mit Gehäuse. Nur 3 Min. Löszeit

Löschset 43,50 DM

Selbstbausatz mit UV-Röhre, Vorschaltgerät, Fassungen usw. für 228 V-Betrieb. Ohne Gehäuse.

Eproms:

| | | | |
|-------|---------|--------------|---------|
| 2764 | 9,— DM | 8 K-RAM | 14,— DM |
| 27128 | 10,— DM | (6264 15 ns) | |
| 27256 | 30,— DM | 8255 | 7,— DM |

Zubehör:

| | |
|--------------------------------|---------|
| Nashua-Disketten SSDD 10 Stck. | 39,— DM |
| Noname-Disketten DSDD 10 Stck. | 30,— DM |

Diskettenhüllen: Klarsichttaschen für Inhaltsverzeichnis. Selbstklebend 50 Stck 8,90 DM

Vertrieb in der Schweiz:

MFS Hard- & Software, 0 37/36 20 60
Postfach, 3185 Schmitzen

Vertrieb in Holland:

Gima Printservice, Tele: 0 77/70 937
PB. 186-5900 AD — Venlo

DELA-Elektronik GmbH

5000 Köln 1

Aquinostr. 12, Tel.: 02 21/72 53 36

Ladenöffnungszeiten: Mo.—Fr. 10 — 18 Uhr.

Nachnahmeversand!

NN-Spesen 7,50
Vorkasse-Betrag + 3,— DM Porto
ab 200,— DM spesenfrei
Ausland: nur Vorkasse
Ab 10 Stck./Artikel gibt es auf
diesen Artikel 10 % Rabatt!!

Info kostenlos!!

Siebzehn und Vier

Beim Glücksspiel 17 und 4 treten zwei Spieler gegeneinander an. Beide decken abwechselnd eine Karte auf. Ziel ist es, möglichst viele Punkte zu machen. Der Haken dabei: Wer mehr als 21 Punkte kassiert, hat verloren. Da heißt es aufpassen und überlegen: Reichen die Punkte schon, um den Gegner auf's Kreuz zu legen? Oder muß noch eine weitere Karte her, mit dem Risiko, die tödlichen 21 zu überschreiten?

32 Karten sind im Spiel. Die Zahlenkarten Sieben bis Zehn geben Punkte entsprechend ihrem Zahlenwert. Für die Buben gibt's zwei, die Damen drei, die Könige vier und die Asse elf Punkte. Jeder Spieler bekommt maximal sechs Karten. Wer genug Punkte hat, kann natürlich vorher mit dem

Kleines Spielchen gefällig? Zum Zocken zu zweit sind keine Karten nötig. Der C 64 teilt aus und zählt die Punkte.

Aufdecken aufhören.

Das Programm erklärt nach einer kurzen Titelgrafik die Regeln und fragt nach den Namen der Spieler. Dann werden die Karten durchgemischt und das Spielfeld aufgebaut. Vor jedem Spieler liegen sechs Karten mit dem Rücken nach oben. Jeder wählt mit den Tasten 1 bis 6 eine Karte aus. Wer

keine mehr braucht, drückt die Taste @. Die Karten werden nun herumgedreht und für jeden Spieler die bisher erreichten Punkte angezeigt. Wenn ein Spieler das Limit von 21 Punkten überschritten hat, oder wenn beide keine Karten mehr wollen, ist das Spiel zu Ende. Nach der Siegerehrung wird eine neue Runde gestartet oder das Programm verlassen. Programmbeschreibung:

0 — 95 Einleitung, Spielregeln, teilweise Definition der Variablen
100 — 985 Definition der Karten, Mischen und Austeilen der Karten
990 — 2740 Aufbau des Spielfeldes, Abfrage der Karten, Punktewertung
2800 — 4003 Auswertung des Spiels

(Michael Balke)

| | | | |
|---|------|---|------|
| 8 POKE 808,2:POKE 657,128 | 1837 | [OFF]/[RVS] [OFF]//[CBM G]" | |
| 1 PRINT "[CLR,WHT]" | 373 | 27 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2988 |
| 9 FOR Y=1 TO 5 | 886 | 2SPACES,RVS] [OFF] [CBM G] [RVS, | |
| 10 PRINT "[HOME,6DOWN,8SPACES,25CBM P]" | 5781 | SHIFT £,OFF,SHIFT £,2SPACES,RVS, | |
| 11 PRINT "[7SPACES]//[24SPACES]//[CBM G]" | 1472 | 3SPACES,OFF]/ [RVS,4SPACES,OFF]/" | |
| 12 PRINT "[7SPACES,RVS]**<C> MICHAEL | 2936 | 28 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2724 |
| BALKE !!!**[OFF] [CBM G]" | | 2SPACES,RVS] [OFF] [CBM G,RVS, | |
| 13 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | SHIFT £,OFF,SHIFT £,4SPACES,RVS] | |
| 21SPACES,RVS] [OFF] [CBM G]" | | [OFF]//[4SPACES,RVS] [OFF] [CBM G]" | |
| 14 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | 29 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2186 |
| 21SPACES,RVS] [OFF] [CBM G]" | | 2SPACES,RVS] [OFF] [SHIFT £, | |
| 15 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | 11SPACES,RVS] [OFF]/" | |
| 21SPACES,RVS] [OFF] [CBM G]" | | 30 NEXT | 130 |
| 16 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | 32 PRINT "[CLR]" | 316 |
| 21SPACES,RVS] [OFF] [CBM G]" | | 33 PRINT "[14SPACES,RVS]SPIELREGELN:" | 2092 |
| 17 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | 34 PRINT "[15SPACES,CBM T,CBM Y,CBM U, | 2315 |
| 21SPACES,RVS] [OFF] [CBM G]" | | RVS,CBM I,CBM O,CBM P,CBM O,CBM I, | |
| 18 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | OFF,CBM U,CBM Y]" | |
| 21SPACES,RVS] [OFF] [CBM G]" | | 36 PRINT "SPIELER: 2" | 1237 |
| 19 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2144 | 37 PRINT "[DOWN] JEDER SPIELER HAT 6 | 2981 |
| 21SPACES,RVS] [OFF] [CBM G]" | | KARTEN !" | |
| 20 PRINT "[7SPACES,RVS] [OFF] L | 5329 | 38 PRINT "[DOWN] WECHSELWEISE WERDEN | 4616 |
| [21CBM P,RVS] [OFF] [CBM G]" | | DIE KARTEN EINZELN" | |
| 21 PRINT "[7SPACES,RVS] [OFF] / | 2202 | 39 PRINT "[DOWN] AUFGEDECKT.WER DIE | 4362 |
| [22SPACES,RVS] [OFF] [CBM G]" | | MEISTEN PUNKTE HAT," | |
| 22 PRINT "[7SPACES,RVS]**PRODUCED BY | 3112 | 40 PRINT "[DOWN] HAT GEWONNEN | 3104 |
| M.B.-SOFT**[OFF]/" | | (VORAUSGESETZT,DASS" | |
| 23 PRINT "[HOME,9DOWN,7SPACES,RVS] | 3120 | 41 PRINT "[DOWN] KEINER DER BEIDEN | 3525 |
| [OFF] [CBM G,3SPACES,CBM P,2SPACES, | | SPIELER MEHR ALS 21" | |
| 4CBM P,6SPACES,CBM P]" | | 42 PRINT "[DOWN] PUNKTE HAT !)." | 1772 |
| 24 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 2949 | 43 PRINT "[DOWN] WENN SIE KEINE KARTE | 3430 |
| 2SPACES]//[CBM G]/[3SPACES]//[CBM G] | | MEHR AUFDECKEN" | |
| [CBM P,2SPACES]//[CBM G,CBM P]" | | 44 PRINT "[DOWN] WOLLEN,DANN GEBEN SIE | 3617 |
| 25 PRINT "[7SPACES,RVS] [OFF] [CBM G] | 3077 | BITTE '0' EIN !" | |
| [RVS,SHIFT £] [OFF] [CBM G,RVS, | | 46 PRINT "[DOWN] [RVS]TASTE DRUECKEN !" | 1900 |
| 4SPACES,OFF] / /L [RVS] [OFF] /L" | | 47 GET Z\$:IF Z#="" THEN 47 | 1587 |
| 26 PRINT "[7SPACES,RVS] [OFF] [CBM G, | 4022 | 60 DIM A\$(32):DIM E(32) | 1354 |
| RVS,SHIFT £,OFF,SHIFT £,RVS] [OFF] | | 70 PRINT "[CLR,WHT]" | 373 |
| [CBM G,2SPACES,RVS,SHIFT £,OFF, | | | |
| SHIFT £] / [RVS] [OFF]//[CBM G,RVS] | | | |

| | | | |
|---|------|---|------|
| 71 S=0:R=0:P(1)=0:P(2)=0:P(3)=0:P(4)=0: P(5)=0:P(6)=0:Q(1)=0:Q(2)=0:Q(3)=0: Q(4)=0 | 7159 | B(4)=A(6) THEN 730 | |
| 72 Q(5)=0:Q(6)=0:K*(1)="A":K*(2)="B" | 3145 | 840 IF B(5)=A(1) OR B(5)=A(2) OR B(5)= A(3) OR B(5)=A(4) OR B(5)=A(5) OR B(5)=A(6) THEN 740 | 6019 |
| 80 POKE 53281,0 | 810 | 850 IF B(6)=A(1) OR B(6)=A(2) OR B(6)= A(3) OR B(6)=A(4) OR B(6)=A(5) OR B(6)=A(6) THEN 750 | 6038 |
| 90 INPUT "[DOWN] SPIELER 1";S*(1) | 2211 | 855 FOR C=1 TO 32 | 918 |
| 95 INPUT "[DOWN] SPIELER 2";S*(2) | 1732 | 860 D=D+1 | 290 |
| 100 A*(1)="*7*" | 1283 | 870 IF A(1)=D THEN B*(1)=A*(D) | 2131 |
| 110 A*(2)="*8*" | 1292 | 880 IF A(2)=D THEN B*(2)=A*(D) | 2147 |
| 120 A*(3)="*9*" | 1301 | 890 IF A(3)=D THEN B*(3)=A*(D) | 2163 |
| 130 A*(4)="*Z*" | 1342 | 900 IF A(4)=D THEN B*(4)=A*(D) | 2179 |
| 140 A*(5)="*B*" | 1326 | 910 IF A(5)=D THEN B*(5)=A*(D) | 2195 |
| 150 A*(6)="*D*" | 1336 | 920 IF A(6)=D THEN B*(6)=A*(D) | 2211 |
| 140 A*(5)="*B*" | 1326 | 930 IF B(1)=D THEN C*(1)=A*(D) | 2134 |
| 150 A*(6)="*D*" | 1336 | 940 IF B(2)=D THEN C*(2)=A*(D) | 2150 |
| 160 A*(7)="*K*" | 1351 | 950 IF B(3)=D THEN C*(3)=A*(D) | 2166 |
| 170 A*(8)="*A*" | 1349 | 960 IF B(4)=D THEN C*(4)=A*(D) | 2182 |
| 200 A*(9)="*7*" | 1457 | 970 IF B(5)=D THEN C*(5)=A*(D) | 2198 |
| 210 A*(10)="*8*" | 1082 | 980 IF B(6)=D THEN C*(6)=A*(D) | 2214 |
| 220 A*(11)="*9*" | 1180 | 985 NEXT | 130 |
| 230 A*(12)="*Z*" | 1182 | 990 PRINT "[CLR,WHT,9SPACES]PUNKTE: [3SPACES,SHIFT -] PUNKTE:" | 2308 |
| 240 A*(13)="*B*" | 1150 | 1000 PRINT "[6SPACES] _____ [SHIFT +] _____" | 3847 |
| 250 A*(14)="*D*" | 1170 | 1010 PRINT "[6SPACES] _____ [2SPACES] [2SPACES] _____ " | 3303 |
| 260 A*(15)="*K*" | 1200 | 1020 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 270 A*(16)="*A*" | 1196 | 1030 PRINT "[6SPACES] [GRN,CBM +,WHT] [GRN,CBM +,WHT] [SHIFT -,GRN,CBM +, WHT]2[GRN,CBM +,WHT] [2SPACES] [2SPACES] [GRN,CBM +,WHT] [GRN, CBM +,WHT,SHIFT -] [GRN,CBM +,WHT]2 [GRN,CBM +,WHT]]" | 4450 |
| 300 A*(17)="*7*" | 1102 | 1040 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 310 A*(18)="*8*" | 1120 | 1050 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 320 A*(19)="*9*" | 1138 | 1060 PRINT "[6SPACES] _____ [2SPACES] [2SPACES] _____]" | 3342 |
| 330 A*(20)="*Z*" | 1068 | 1070 PRINT "[6SPACES] _____ [2SPACES] [2SPACES] _____ " | 3303 |
| 340 A*(21)="*B*" | 1036 | 1080 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 350 A*(22)="*D*" | 1056 | 1090 PRINT "[6SPACES] [GRN,CBM +,WHT] 3 [GRN,CBM +,WHT] [SHIFT -,GRN,CBM +, WHT]4[GRN,CBM +,WHT] [2SPACES] [2SPACES] [GRN,CBM +,WHT]3[GRN, CBM +,WHT,SHIFT -] [GRN,CBM +,WHT]4 [GRN,CBM +,WHT]]" | 4491 |
| 360 A*(23)="*K*" | 1086 | 1100 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 370 A*(24)="*A*" | 1082 | 1110 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 400 A*(25)="*7*" | 1193 | 1120 PRINT "[6SPACES] _____ [2SPACES] [2SPACES] _____ " | 3342 |
| 410 A*(26)="*8*" | 1211 | 1130 PRINT "[6SPACES] _____ [2SPACES] [2SPACES] _____ " | 3303 |
| 420 A*(27)="*9*" | 1229 | 1140 PRINT "[6SPACES] [GRN,3CBM +,WHT] [SHIFT -,GRN,3CBM +,WHT] [2SPACES] [2SPACES] [GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]]" | 4263 |
| 430 A*(28)="*Z*" | 1311 | 1150 PRINT "[6SPACES] [GRN,CBM +,WHT] 5 [GRN,CBM +,WHT] [SHIFT -,GRN,CBM +, WHT]6[GRN,CBM +,WHT] [2SPACES] | 4779 |
| 440 A*(29)="*B*" | 1279 | | |
| 450 A*(30)="*D*" | 1147 | | |
| 460 A*(31)="*K*" | 1177 | | |
| 470 A*(32)="*A*" | 1173 | | |
| 500 A(1)=INT (RND (1)*32)+1 | 1602 | | |
| 510 A(2)=INT (RND (1)*32)+1 | 1606 | | |
| 520 A(3)=INT (RND (1)*32)+1 | 1610 | | |
| 530 A(4)=INT (RND (1)*32)+1 | 1614 | | |
| 540 A(5)=INT (RND (1)*32)+1 | 1618 | | |
| 550 A(6)=INT (RND (1)*32)+1 | 1622 | | |
| 560 IF A(2)=A(1) THEN 510 | 1389 | | |
| 570 IF A(3)=A(2) OR A(3)=A(1) THEN 520 | 2511 | | |
| 580 IF A(4)=A(3) OR A(4)=A(2) OR A(4)= A(1) THEN 530 | 2993 | | |
| 590 IF A(5)=A(4) OR A(5)=A(3) OR A(5)= A(2) OR A(5)=A(1) THEN 540 | 4170 | | |
| 600 IF A(6)=A(5) OR A(6)=A(4) OR A(6)= A(3) OR A(6)=A(2) OR A(6)=A(1) THEN 550 | 5082 | | |
| 700 B(1)=INT (RND (1)*32)+1 | 1603 | | |
| 710 B(2)=INT (RND (1)*32)+1 | 1607 | | |
| 720 B(3)=INT (RND (1)*32)+1 | 1611 | | |
| 730 B(4)=INT (RND (1)*32)+1 | 1615 | | |
| 740 B(5)=INT (RND (1)*32)+1 | 1619 | | |
| 750 B(6)=INT (RND (1)*32)+1 | 1623 | | |
| 760 IF B(2)=B(1) THEN 710 | 1471 | | |
| 770 IF B(3)=B(2) OR B(3)=B(1) THEN 720 | 2650 | | |
| 780 IF B(4)=B(3) OR B(4)=B(2) OR B(4)= B(1) THEN 730 | 3185 | | |
| 790 IF B(5)=B(4) OR B(5)=B(3) OR B(5)= B(2) OR B(5)=B(1) THEN 740 | 4174 | | |
| 800 IF B(1)=A(1) OR B(1)=A(2) OR B(1)= A(3) OR B(1)=A(4) OR B(1)=A(5) OR B(1)=A(6) THEN 700 | 5688 | | |
| 810 IF B(2)=A(1) OR B(2)=A(2) OR B(2)= A(3) OR B(2)=A(4) OR B(2)=A(5) OR B(2)=A(6) THEN 710 | 5707 | | |
| 820 IF B(3)=A(1) OR B(3)=A(2) OR B(3)= A(3) OR B(3)=A(4) OR B(3)=A(5) OR B(3)=A(6) THEN 720 | 5981 | | |
| 830 IF B(4)=A(1) OR B(4)=A(2) OR B(4)= A(3) OR B(4)=A(4) OR B(4)=A(5) OR | 6000 | | |

Anwendung

| | | | |
|---|------|--------------------------------|------|
| [2SPACES][GRN,CBM +,WHT]5[GRN, CBM +,WHT,SHIFT -][GRN,CBM +,WHT]6 [GRN,CBM +,WHT]11" | | 1543 IF G*(2)="8" THEN 1551 | 1495 |
| 1160 PRINT "[6SPACES] [GRN,3CBM +,WHT]" | 4263 | 1544 IF G*(2)="9" THEN 1552 | 1560 |
| [SHIFT -,GRN,3CBM +,WHT][2SPACES] [2SPACES][GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]11" | | 1545 IF G*(2)="Z" THEN 1553 | 1657 |
| 1170 PRINT "[6SPACES] [GRN,3CBM +,WHT]" | 4263 | 1546 IF G*(2)="B" THEN 1554 | 1442 |
| [SHIFT -,GRN,3CBM +,WHT][2SPACES] [2SPACES][GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]11" | | 1547 IF G*(2)="D" THEN 1555 | 1508 |
| 1180 PRINT "[6SPACES] [GRN,3CBM +,WHT]" | 4263 | 1548 IF G*(2)="K" THEN 1556 | 1579 |
| [SHIFT -,GRN,3CBM +,WHT][2SPACES] [2SPACES][GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]11" | | 1549 IF G*(2)="A" THEN 1557 | 1633 |
| 1190 PRINT "[6SPACES] [GRN,3CBM +,WHT]" | 4263 | 1550 P(2)=7:GOTO 1340 | 1532 |
| [SHIFT -,GRN,3CBM +,WHT][2SPACES] [2SPACES][GRN,3CBM +,WHT,SHIFT -] [GRN,3CBM +,WHT]11" | | 1551 P(2)=8:GOTO 1340 | 1309 |
| 1200 PRINT "[6SPACES] [2SPACES]" | 3342 | 1552 P(2)=9:GOTO 1340 | 1341 |
| [2SPACES] [2SPACES]" | | 1553 P(2)=10:GOTO 1340 | 1158 |
| 1210 PRINT "[6SPACES,CBM Q]" | 3972 | 1554 P(2)=2:GOTO 1340 | 1372 |
| [SHIFT +][CBM W]" | | 1555 P(2)=3:GOTO 1340 | 1404 |
| 1220 PRINT "[6SPACES][12SPACES]" | 1515 | 1556 P(2)=4:GOTO 1340 | 1436 |
| [12SPACES]" | | 1557 P(2)=11:GOTO 1340 | 1222 |
| 1210 PRINT "[6SPACES][CBM E]" | 3775 | 1600 PRINT "[HOME,WHT,8DOWN]" | 881 |
| [CBM E]" | | 1610 PRINT SPC(8);B*(3) | 940 |
| 1220 PRINT "[HOME,21DOWN,6SPACES]";S*(1) | 2622 | 1620 PRINT SPC(8);"[3SPACES]" | 898 |
| 1230 PRINT "[HOME,21DOWN]";SPC(19);"1"; | 3569 | 1630 PRINT SPC(8);"[3SPACES]" | 898 |
| S*(2) | | 1640 PRINT SPC(8);B*(3) | 940 |
| 1240 PRINT "[HOME,21DOWN]";SPC(32);"1" | 3120 | 1641 G*(3)=MID*(B*(3),2,1) | 1909 |
| [10SPACES]" | | 1642 IF G*(3)="7" THEN 1650 | 1462 |
| 1245 IF K*(1)="e" THEN 1256 | 1512 | 1643 IF G*(3)="8" THEN 1651 | 1527 |
| 1250 INPUT "[HOME,23DOWN,3SPACES]WELCHE | 3403 | 1644 IF G*(3)="9" THEN 1652 | 1592 |
| KARTE";L\$ | | 1645 IF G*(3)="Z" THEN 1653 | 1689 |
| 1251 K*(1)=MID*(L\$,1,1) | 1641 | 1646 IF G*(3)="B" THEN 1654 | 1474 |
| 1255 IF K*(1)<"1" OR K*(1)>"6" AND K*(1)< | 3722 | 1647 IF G*(3)="D" THEN 1655 | 1540 |
| >"e" THEN 1250 | | 1648 IF G*(3)="K" THEN 1656 | 1611 |
| 1256 IF K*(1)="e" AND K*(2)="e" THEN 2700 | 2247 | 1649 IF G*(3)="A" THEN 1657 | 1665 |
| 1257 IF K*(2)="e" THEN 1260 | 1240 | 1650 P(3)=7:GOTO 1340 | 1536 |
| 1260 PRINT "[HOME,23DOWN]";SPC(21);:INPUT | 4134 | 1651 P(3)=8:GOTO 1340 | 1313 |
| "WELCHE KARTE";M\$ | | 1652 P(3)=9:GOTO 1340 | 1345 |
| 1265 K*(2)=MID*(M\$,1,1) | 1650 | 1653 P(3)=10:GOTO 1340 | 1162 |
| 1270 IF K*(2)<"1" OR K*(2)>"6" AND K*(2)< | 3807 | 1654 P(3)=2:GOTO 1340 | 1376 |
| >"e" THEN 1260 | | 1655 P(3)=3:GOTO 1340 | 1408 |
| 1275 IF K*(1)="e" THEN 1340 | 1367 | 1656 P(3)=4:GOTO 1340 | 1440 |
| 1280 IF K*(1)="1" THEN 1400 | 1240 | 1657 P(3)=11:GOTO 1340 | 1226 |
| 1290 IF K*(1)="2" THEN 1500 | 1257 | 1700 PRINT "[HOME,WHT,8DOWN]" | 881 |
| 1300 IF K*(1)="3" THEN 1600 | 1274 | 1710 PRINT SPC(13);B*(4) | 1298 |
| 1310 IF K*(1)="4" THEN 1700 | 1291 | 1720 PRINT SPC(13);"[3SPACES]" | 1212 |
| 1320 IF K*(1)="5" THEN 1800 | 1308 | 1730 PRINT SPC(13);"[3SPACES]" | 1212 |
| 1330 IF K*(1)="6" THEN 1900 | 1325 | 1740 PRINT SPC(13);B*(4) | 1298 |
| 1340 IF K*(2)="1" THEN 2000 | 1200 | 1741 G*(4)=MID*(B*(4),2,1) | 1925 |
| 1345 IF K*(2)="e" THEN 2600 | 1311 | 1742 IF G*(4)="7" THEN 1750 | 1494 |
| 1350 IF K*(2)="2" THEN 2100 | 1217 | 1743 IF G*(4)="8" THEN 1751 | 1559 |
| 1360 IF K*(2)="3" THEN 2200 | 1234 | 1744 IF G*(4)="9" THEN 1752 | 1624 |
| 1370 IF K*(2)="4" THEN 2300 | 1251 | 1745 IF G*(4)="Z" THEN 1753 | 1721 |
| 1380 IF K*(2)="5" THEN 2400 | 1268 | 1746 IF G*(4)="B" THEN 1754 | 1506 |
| 1390 IF K*(2)="6" THEN 2500 | 1285 | 1747 IF G*(4)="D" THEN 1755 | 1572 |
| 1400 PRINT "[HOME,WHT,20DOWN]" | 524 | 1748 IF G*(4)="K" THEN 1756 | 1643 |
| 1410 PRINT SPC(8);B*(1) | 938 | 1749 IF G*(4)="A" THEN 1757 | 1697 |
| 1420 PRINT SPC(8);"[3SPACES]" | 898 | 1750 P(4)=7:GOTO 1340 | 1540 |
| 1430 PRINT SPC(8);"[3SPACES]" | 898 | 1751 P(4)=8:GOTO 1340 | 1317 |
| 1440 PRINT SPC(8);B*(1) | 938 | 1752 P(4)=9:GOTO 1340 | 1349 |
| 1441 G*(1)=MID*(B*(1),2,1) | 1077 | 1753 P(4)=10:GOTO 1340 | 1166 |
| 1442 IF G*(1)="7" THEN 1450 | 1398 | 1754 P(4)=2:GOTO 1340 | 1380 |
| 1443 IF G*(1)="8" THEN 1451 | 1463 | 1755 P(4)=3:GOTO 1340 | 1412 |
| 1444 IF G*(1)="9" THEN 1452 | 1528 | 1756 P(4)=4:GOTO 1340 | 1444 |
| 1445 IF G*(1)="Z" THEN 1453 | 1625 | 1757 P(4)=11:GOTO 1340 | 1230 |
| 1446 IF G*(1)="B" THEN 1454 | 1418 | 1800 PRINT "[HOME,WHT,14DOWN]" | 1289 |
| 1447 IF G*(1)="D" THEN 1455 | 1476 | 1810 PRINT SPC(8);B*(5) | 942 |
| 1448 IF G*(1)="K" THEN 1456 | 1547 | 1820 PRINT SPC(8);"[3SPACES]" | 898 |
| 1449 IF G*(1)="A" THEN 1457 | 1601 | 1830 PRINT SPC(8);"[3SPACES]" | 898 |
| 1450 P(1)=7:GOTO 1340 | 1528 | 1840 PRINT SPC(8);B*(5) | 942 |
| 1451 P(1)=8:GOTO 1340 | 1305 | 1841 G*(5)=MID*(B*(5),2,1) | 1941 |
| 1452 P(1)=9:GOTO 1340 | 1337 | 1842 IF G*(5)="7" THEN 1850 | 1526 |
| 1453 P(1)=10:GOTO 1340 | 1154 | 1843 IF G*(5)="8" THEN 1851 | 1591 |
| 1454 P(1)=2:GOTO 1340 | 1368 | 1844 IF G*(5)="9" THEN 1852 | 1656 |
| 1455 P(1)=3:GOTO 1340 | 1400 | 1845 IF G*(5)="Z" THEN 1853 | 1753 |
| 1456 P(1)=4:GOTO 1340 | 1432 | 1846 IF G*(5)="B" THEN 1854 | 1538 |
| 1457 P(1)=11:GOTO 1340 | 1218 | 1847 IF G*(5)="D" THEN 1855 | 1604 |
| 1500 PRINT "[HOME,WHT,20DOWN]" | 524 | 1848 IF G*(5)="K" THEN 1856 | 1675 |
| 1510 PRINT SPC(13);B*(2) | 1294 | 1849 IF G*(5)="A" THEN 1857 | 1729 |
| 1520 PRINT SPC(13);"[3SPACES]" | 1212 | 1850 P(5)=7:GOTO 1340 | 1544 |
| 1530 PRINT SPC(13);"[3SPACES]" | 1212 | 1851 P(5)=8:GOTO 1340 | 1321 |
| 1540 PRINT SPC(13);B*(2) | 1294 | 1852 P(5)=9:GOTO 1340 | 1353 |
| 1541 G*(2)=MID*(B*(2),2,1) | 1093 | 1853 P(5)=10:GOTO 1340 | 1170 |
| 1542 IF G*(2)="7" THEN 1550 | 1430 | 1854 P(5)=2:GOTO 1340 | 1384 |
| | | 1855 P(5)=3:GOTO 1340 | 1416 |

| | | | |
|--------------------------------|------|--------------------------------------|------|
| 1856 P(5)=4:GOTO 1340 | 1448 | 2247 IF H*(3)="D" THEN 2255 | 1486 |
| 1857 P(5)=11:GOTO 1340 | 1234 | 2248 IF H*(3)="K" THEN 2256 | 1557 |
| 1900 PRINT "[HOME,WHT,14DOWN]" | 1289 | 2249 IF H*(3)="A" THEN 2257 | 1611 |
| 1910 PRINT SPC(13);B*(6) | 1302 | 2250 Q(3)=7:GOTO 2600 | 1528 |
| 1920 PRINT SPC(13);"[3SPACES]" | 1212 | 2251 Q(3)=8:GOTO 2600 | 1305 |
| 1930 PRINT SPC(13);"[3SPACES]" | 1212 | 2252 Q(3)=9:GOTO 2600 | 1337 |
| 1940 PRINT SPC(13);B*(6) | 1302 | 2253 Q(3)=10:GOTO 2600 | 1145 |
| 1941 G*(6)=MID*(B*(6),2,1) | 1957 | 2254 Q(3)=2:GOTO 2600 | 1368 |
| 1942 IF G*(6)="7" THEN 1950 | 1558 | 2255 Q(3)=3:GOTO 2600 | 1400 |
| 1943 IF G*(6)="8" THEN 1951 | 1623 | 2256 Q(3)=4:GOTO 2600 | 1432 |
| 1944 IF G*(6)="9" THEN 1952 | 1688 | 2257 Q(3)=11:GOTO 2600 | 1209 |
| 1945 IF G*(6)="2" THEN 1953 | 1785 | 2300 PRINT "[HOME,WHT,8DOWN]" | 881 |
| 1946 IF G*(6)="B" THEN 1954 | 1570 | 2310 PRINT SPC(28);C*(4) | 1406 |
| 1947 IF G*(6)="D" THEN 1955 | 1636 | 2320 PRINT SPC(28);"[3SPACES]" | 1256 |
| 1948 IF G*(6)="K" THEN 1956 | 1707 | 2330 PRINT SPC(28);"[3SPACES]" | 1256 |
| 1949 IF G*(6)="A" THEN 1957 | 1761 | 2340 PRINT SPC(28);C*(4) | 1406 |
| 1950 P(6)=7:GOTO 1340 | 1548 | 2341 H*(4)=MID*(C*(4),2,1) | 1927 |
| 1951 P(6)=8:GOTO 1340 | 1325 | 2342 IF H*(4)="7" THEN 2350 | 1448 |
| 1952 P(6)=9:GOTO 1340 | 1357 | 2343 IF H*(4)="8" THEN 2351 | 1505 |
| 1953 P(6)=10:GOTO 1340 | 1174 | 2344 IF H*(4)="9" THEN 2352 | 1578 |
| 1954 P(6)=2:GOTO 1340 | 1388 | 2345 IF H*(4)="2" THEN 2353 | 1667 |
| 1955 P(6)=3:GOTO 1340 | 1420 | 2346 IF H*(4)="B" THEN 2354 | 1452 |
| 1956 P(6)=4:GOTO 1340 | 1452 | 2347 IF H*(4)="D" THEN 2355 | 1518 |
| 1957 P(6)=11:GOTO 1340 | 1238 | 2348 IF H*(4)="K" THEN 2356 | 1589 |
| 2000 PRINT "[HOME,WHT,2DOWN]" | 524 | 2349 IF H*(4)="A" THEN 2357 | 1643 |
| 2010 PRINT SPC(23);C*(1) | 1360 | 2350 Q(4)=7:GOTO 2600 | 1532 |
| 2020 PRINT SPC(23);"[3SPACES]" | 1216 | 2351 Q(4)=8:GOTO 2600 | 1389 |
| 2030 PRINT SPC(23);"[3SPACES]" | 1216 | 2352 Q(4)=9:GOTO 2600 | 1341 |
| 2040 PRINT SPC(23);C*(1) | 1360 | 2353 Q(4)=10:GOTO 2600 | 1149 |
| 2041 H*(1)=MID*(C*(1),2,1) | 1879 | 2354 Q(4)=2:GOTO 2600 | 1372 |
| 2042 IF H*(1)="7" THEN 2050 | 1344 | 2355 Q(4)=3:GOTO 2600 | 1404 |
| 2043 IF H*(1)="8" THEN 2051 | 1409 | 2356 Q(4)=4:GOTO 2600 | 1436 |
| 2044 IF H*(1)="9" THEN 2052 | 1474 | 2357 Q(4)=11:GOTO 2600 | 1213 |
| 2045 IF H*(1)="2" THEN 2053 | 1571 | 2400 PRINT "[HOME,WHT,14DOWN]" | 1289 |
| 2046 IF H*(1)="B" THEN 2054 | 1356 | 2410 PRINT SPC(23);C*(5) | 1368 |
| 2047 IF H*(1)="D" THEN 2055 | 1422 | 2420 PRINT SPC(23);"[3SPACES]" | 1216 |
| 2048 IF H*(1)="K" THEN 2056 | 1493 | 2430 PRINT SPC(23);"[3SPACES]" | 1216 |
| 2049 IF H*(1)="A" THEN 2057 | 1547 | 2440 PRINT SPC(23);C*(5) | 1368 |
| 2050 Q(1)=7:GOTO 2600 | 1520 | 2441 H*(5)=MID*(C*(5),2,1) | 1943 |
| 2051 Q(1)=8:GOTO 2600 | 1297 | 2442 IF H*(5)="7" THEN 2450 | 1472 |
| 2052 Q(1)=9:GOTO 2600 | 1329 | 2443 IF H*(5)="8" THEN 2451 | 1537 |
| 2053 Q(1)=10:GOTO 2600 | 1137 | 2444 IF H*(5)="9" THEN 2452 | 1682 |
| 2054 Q(1)=2:GOTO 2600 | 1368 | 2445 IF H*(5)="2" THEN 2453 | 1699 |
| 2055 Q(1)=3:GOTO 2600 | 1392 | 2446 IF H*(5)="B" THEN 2454 | 1484 |
| 2056 Q(1)=4:GOTO 2600 | 1424 | 2447 IF H*(5)="D" THEN 2455 | 1550 |
| 2057 Q(1)=11:GOTO 2600 | 1201 | 2448 IF H*(5)="K" THEN 2456 | 1621 |
| 2100 PRINT "[HOME,WHT,2DOWN]" | 524 | 2449 IF H*(5)="A" THEN 2457 | 1675 |
| 2110 PRINT SPC(28);C*(2) | 1402 | 2450 Q(5)=7:GOTO 2600 | 1536 |
| 2120 PRINT SPC(28);"[3SPACES]" | 1256 | 2451 Q(5)=8:GOTO 2600 | 1313 |
| 2130 PRINT SPC(28);"[3SPACES]" | 1256 | 2452 Q(5)=9:GOTO 2600 | 1345 |
| 2140 PRINT SPC(28);C*(2) | 1402 | 2453 Q(5)=10:GOTO 2600 | 1153 |
| 2141 H*(2)=MID*(C*(2),2,1) | 1895 | 2454 Q(5)=2:GOTO 2600 | 1376 |
| 2142 IF H*(2)="7" THEN 2150 | 1376 | 2455 Q(5)=3:GOTO 2600 | 1408 |
| 2143 IF H*(2)="8" THEN 2151 | 1441 | 2456 Q(5)=4:GOTO 2600 | 1440 |
| 2144 IF H*(2)="9" THEN 2152 | 1506 | 2457 Q(5)=11:GOTO 2600 | 1217 |
| 2145 IF H*(2)="2" THEN 2153 | 1603 | 2500 PRINT "[HOME,WHT,14DOWN]" | 1289 |
| 2146 IF H*(2)="B" THEN 2154 | 1388 | 2510 PRINT SPC(28);C*(6) | 1410 |
| 2147 IF H*(2)="D" THEN 2155 | 1454 | 2520 PRINT SPC(28);"[3SPACES]" | 1256 |
| 2148 IF H*(2)="K" THEN 2156 | 1525 | 2530 PRINT SPC(28);"[3SPACES]" | 1256 |
| 2149 IF H*(2)="A" THEN 2157 | 1579 | 2540 PRINT SPC(28);C*(6) | 1410 |
| 2150 Q(2)=7:GOTO 2600 | 1524 | 2541 H*(6)=MID*(C*(6),2,1) | 1959 |
| 2151 Q(2)=8:GOTO 2600 | 1301 | 2542 IF H*(6)="7" THEN 2550 | 1504 |
| 2152 Q(2)=9:GOTO 2600 | 1333 | 2543 IF H*(6)="8" THEN 2551 | 1569 |
| 2153 Q(2)=10:GOTO 2600 | 1141 | 2544 IF H*(6)="9" THEN 2552 | 1634 |
| 2154 Q(2)=2:GOTO 2600 | 1364 | 2545 IF H*(6)="2" THEN 2553 | 1731 |
| 2155 Q(2)=3:GOTO 2600 | 1396 | 2546 IF H*(6)="B" THEN 2554 | 1516 |
| 2156 Q(2)=4:GOTO 2600 | 1428 | 2547 IF H*(6)="D" THEN 2555 | 1582 |
| 2157 Q(2)=11:GOTO 2600 | 1205 | 2548 IF H*(6)="K" THEN 2556 | 1653 |
| 2200 PRINT "[HOME,WHT,8DOWN]" | 881 | 2549 IF H*(6)="A" THEN 2557 | 1707 |
| 2210 PRINT SPC(23);C*(3) | 1364 | 2550 Q(6)=7:GOTO 2600 | 1548 |
| 2220 PRINT SPC(23);"[3SPACES]" | 1216 | 2551 Q(6)=8:GOTO 2600 | 1317 |
| 2230 PRINT SPC(23);"[3SPACES]" | 1216 | 2552 Q(6)=9:GOTO 2600 | 1349 |
| 2240 PRINT SPC(23);C*(3) | 1364 | 2553 Q(6)=10:GOTO 2600 | 1157 |
| 2241 H*(3)=MID*(C*(3),2,1) | 1911 | 2554 Q(6)=2:GOTO 2600 | 1380 |
| 2242 IF H*(3)="7" THEN 2250 | 1408 | 2555 Q(6)=3:GOTO 2600 | 1412 |
| 2243 IF H*(3)="8" THEN 2251 | 1473 | 2556 Q(6)=4:GOTO 2600 | 1444 |
| 2244 IF H*(3)="9" THEN 2252 | 1538 | 2557 Q(6)=11:GOTO 2600 | 1221 |
| 2245 IF H*(3)="2" THEN 2253 | 1635 | 2600 R=P(1)+P(2)+P(3)+P(4)+P(5)+P(6) | 3089 |
| 2246 IF H*(3)="B" THEN 2254 | 1420 | 2610 S=Q(1)+Q(2)+Q(3)+Q(4)+Q(5)+Q(6) | 3312 |

```

2611 PRINT "[HOME,WHIT,SPACES]PUNKTE:";R; 3000
      * PUNKTE:";S
2612 FOR X=1 TO 2000:NEXT 998
2615 IF R>21 AND S=21 THEN 3200 1727
2616 IF R>21 AND S>21 THEN 3300 1866
2620 IF R=21 THEN 2800 1225
2630 IF S=21 THEN 2900 1355
2640 IF R>21 THEN 3000 1201
2650 IF S>21 THEN 3100 1411
2660 GOTO 1220 503
2700 IF R=S THEN 3200 1095
2710 IF R>S AND R<21 THEN 2800 1770
2720 IF R>S AND R>21 THEN 3000 1592
2730 IF S>R AND S<21 THEN 2900 1804
2740 IF R>21 AND S>21 THEN 3300 1866
2800 PRINT "[CLR]" 316
2810 PRINT "[BDOWN]" 867
2820 PRINT "[RVS]";S*(1);"[OFF] HAT MIT" 4928
      ;R;"PUNKTEN GEWONNEN !"
2830 PRINT "[DOWN] [RVS]";S*(2);"[OFF]" 4550
      HAT 'NUR';S;"PUNKTE GEHOLT ."
2840 GOTO 4000 565
2900 PRINT "[CLR]" 316
2910 PRINT "[BDOWN]" 867
2920 PRINT "[RVS]";S*(2);"[OFF] HAT MIT" 4931
      ;S;"PUNKTEN GEWONNEN !"
2930 PRINT "[DOWN] [RVS]";S*(1);"[OFF]" 4538
      HAT 'NUR';R;"PUNKTE GEHOLT ."
2940 GOTO 4000 565
3000 PRINT "[CLR]" 316
3010 PRINT "[BDOWN]" 867
3020 PRINT "[RVS]";S*(2);"[OFF] HAT MIT" 4931
      ;S;"PUNKTEN GEWONNEN !"
3030 PRINT "[DOWN] [RVS]";S*(1);"[OFF]" 3760
      HAT";R;"PUNKTE GEHOLT ."
3040 GOTO 4000 565
3100 PRINT "[CLR]" 316
3110 PRINT "[BDOWN]" 867
3120 PRINT "[RVS]";S*(1);"[OFF] HAT MIT" 4928
      ;R;"PUNKTEN GEWONNEN !"
3130 PRINT "[DOWN] [RVS]";S*(2);"[OFF]" 3796
      HAT";S;"PUNKTE GEHOLT ."
3140 GOTO 4000 565
3200 PRINT "[CLR]" 316
3210 PRINT "[BDOWN]" 867
3220 PRINT " UNENTSCHEIDEN !!! 2007
3230 PRINT "[DOWN] JEDER HAT";S;"PUNKTE 3160
      GEHOLT !"
3240 GOTO 4000 565
3300 PRINT "[CLR]" 316
3310 PRINT "[BDOWN]" 867
3320 PRINT " VERLOREN !!! 1275
3330 PRINT "[DOWN] JEDER HAT MIT";R;"/"; 4475
      S;"PUNKTEN VERLOREN!"
3340 GOTO 4000 565
4000 PRINT "[BDOWN] [RVS]NOCH EINMAL? 2326
      (J/N)"
4001 GET Z$;IF Z$<"J" AND Z$<"N" THEN 2693
      4001
4002 IF Z$="J" THEN RUN :END 1322
4003 SYS 64730 717
    
```

ANWENDER BLEIBEN ÜBERRASCHEN

PARALLEL DRUCKER INTERFACE
 Jeder Drucker auf Ihren Commodore
 Epson, Brother, Star, Seikosha, Dai usw.
 • Problemloser Anschluss
 • Kompatibel zu allen bestehenden Software
 • Bietet wieder Expansion - nach Überport
 • Benötigt keinen Datenspeicher
 • Braucht nicht geladen zu werden

OPTION: eingebauter 192K Druckerpuffer 49,50

VC-20
C-64

80 ZEICHEN/GRAFIK KARTE
 mit verbessertem Textverarbeiter
 • Hartharacter und stabiles 80 Zeichenbild
 • Erweiteter Zeilenabstand
 • Digitale Linr
 • Mit Simons Basic viele Möglichkeiten

• Grafische Textmöglichkeiten
 • Ausführliche deutsche Anleitung

279,-

- 32KB RAMKARTE, ersatzbar, in Gehäuse 189; 149;
- 64KB RAMKARTE, idem, incl. superschnelle Ramfile Software 379; 119;
- 40/80 ZEICHENKARTE, macht VC-20 zum perf. Computer 249; 139;
- EPROM KARTEN, 2K 4K + 2K Adressengabel einstellbar -54; 18;
- STECKPLATZ ERWEITERUNG, 2 Plätze -69; 31;
- STECKPLATZ ERWEITERUNG, 5 Plätze, vollgepuffert, schwarz 139; 85;
- STECKPLATZ ERWEITERUNG, 2 Plätze 59;
- STECKPLATZ ERWEITERUNG, 5 Plätze, vollgepuffert, schwarz 174;
- EPROM KARTEN, 2K 4K + 8K Adressengabel einstellbar 54;
- COM-IN 64, wandelt CBM-64 in einen Kommunikations Schnittstelle 596; 498;
- EPROMPROGRAMMIERER, ihr eigenes Programm auf Eprom 139,50
- EPROM LÖSCHER, löscht 4 Eproms zugleich 120;
- ZENITH MONITOR, entspiegelter Bildschirm: grün/berstein 295;
- RELAISBOX 79,50
- INFRA-ROT SYS. 149,50
- POWERBOX 79,50
- SERVO SYS. 39,50

ROOS
ELEKTRONIK
 KLEINER MARKT 7 • 4100 KLEVE • TELEFON 02021 / 28026

Alle Preise einschließlich MwSt.
 • Versand per Nachnahme oder Vorkasse.
 • Von all unseren Produkten haben wir technische Prospekte, die wir Ihnen gerne kostenlos zusenden.
 • Alle Geräte können freigelegt und erweitert werden. Umsonst! (erfordert für 10 Tage zurückgeschickt, bezahlen Sie nur für das neue Verschickte)

Händler Aufträge erwünscht!

Der schnelle TURBO-ASS:

ca. 500 (fünfhundert!) Zeilen pro Sekunde. ein 40 Druckerseiten langer Quelltext wird in ca. 6 Sekunden vollständig assembliert.

Der komfortable TURBO-ASS:

Ihren Quelltext erstellen Sie bequem mit dem eingebauten Full-Screen-Editor. Im Quelltext auf- und abwärtsrollen, Sonderfunktionen wie „Suchen“ und „Ersetzen“, Blöcke verschieben/kopieren oder Teile aus anderen Programmen hinzuladen (MERGE) sind für den Editor des TURBO-ASS keine Fremdwörter, sondern eingebaute Kommandos.

Der speichereffektive TURBO-ASS:

Durch ein spezielles Codierverfahren nützt der TURBO-ASS den Speicher besonders effektiv. Der Quelltext eines Programms, das assembliert 8 K Maschinencode und mehr umfaßt, kann noch im Speicher gehalten und als Ganzes bearbeitet werden.

Der freundliche TURBO-ASS:

Auf Syntax-Felder macht Sie der TURBO-ASS bereits bei der Eingabe aufmerksam, „Vertipper“ fallen nicht erst beim Assemblieren auf, sondern können sofort korrigiert werden.

Der kompatible TURBO-ASS:

Außer seinen besonders schnellen und Platz sparenden „Sprint-Files“ kann der TURBO-ASS auch sequentielle Dateien lesen und schreiben. Dadurch können Quelltexte des ASSI/ASM/FSE und viele andere Assembler in den TURBO-ASS übernommen werden — bereits begonnene Projekte können Sie bequem mit dem TURBO-ASS fortsetzen. Und wenn Sie gar keinen Quelltext, sondern nur das fertige Maschinenprogramm haben: Eine auf dem TURBO-ASS zugeschnittene Version des Omikron-REASS wird mitgeliefert.

Der Lieferumfang des TURBO-ASS:

Diskette mit TURBO-ASS, Omikron-REASS (Spezialversion) sowie diverse Hilfs- und Beispielprogramme (z. B. Quelltext für Maschinensprache-Monitor); ausführliches Handbuch spiralgebunden; RESET-Taster für RESET ohne Datenverlust.
 Lieferbar ist der TURBO-ASS ab 17. Oktober 85, pünktlich zur „Commodore '85“ in Frankfurt. Unsere Messestand können Sie gar nicht verfehlen — er ist der zweite nach dem Eingang rechts, Nr. 200. Neben dem TURBO-ASS führen wir Ihnen dort auch gerne unsere anderen Programmierhilfen vor:
 BASIC 64, die vielseitige und schnelle Basic-Erweiterung, mehrfacher „Testslieger“ im 64er, CHIP, RUN und anderen Fachzeitschriften.
 Und den Omikron-REASS, den intelligenten Re-Assembler, der selber denkt.

Sollten Sie nicht zur Messe kommen, können beraten wir Sie natürlich auch gerne telefonisch oder fordern Sie unsere Gratisprospekte an.

OMIKRON Software
 Erlichstr. 15
 7534 Birkenfeld 2
 Tel. 0 70 82/53 86

Unsere Produkte erhalten Sie in dem Horten-Computer-Center, in guten Fachgeschäften oder direkt von uns.

Händleranfragen erwünscht!

Einmaleins gleich Vierundsechzig

Rechnen lernen ist für Grundschüler üble Plackerei. Susanne Jungnickl ist Volksschullehrerin und kennt die Nöte ihrer Zöglinge. Mit vier unterhaltsamen Lernprogrammen will sie den Kindern den Einstieg ins Reich der Zahlen erleichtern: „Da ich in unserem Haushalt, mein Mann besitzt einen C64, ich einen VC20, bald feststellen konnte, welche Faszination Computer auf Kinder ausüben, beschloß ich, diese Begeisterung für Lern- und Übungszwecke zu nutzen.“

Jedes Programm behandelt eine der vier Grundrechenarten. Dabei sind die lehrplanmäßigen Entwicklungsschritte berücksichtigt worden. Je nach Alter beziehungsweise erreichten Wissensstand können verschiedene Schwierigkeitsgrade gewählt werden. Zur Kontrolle läuft ab dem Start des Programms die Zeit mit. Am Ende der Sitzung wird die erbrachte Leistung bewertet.

Addieren

Hier werden einfache bis schwierige Additionen geübt (Listing 1). Nach kurzer Einführung können Grundschüler mit diesem Programm selbständig arbeiten.

Nach kurzer Einleitung wird die Startzeit eingegeben. Danach erscheint das Menü:

- F1 Einer + Einer
- F2 Reine Zehner + Reine Zehner
- F3 Reine Zehner + Einer
- F4 Gemischte Zehner + Einer
- F5 Gemischte Zehner + Reine Zehner
- F6 Gemischte Zehner + Gemischte Zehner
- F7 Programmende

Nach Auswahl des Aufgabentyps wird die Anzahl der gewünschten Rechnungen festgelegt. Danach erscheint die erste Addition. Korrekt eingetippte Ergebnisse werden mit „RICHTIG!“ bestätigt. Bei falschem Ergebnis erscheint „LEIDER FALSCH!“ mit der Aufforderung, noch einmal zu rechnen. Irrt

Mathe ohne Mühe: Vier Lernprogramme bringen Abc-Schützen das Rechnen bei.

man sich ein zweites Mal, wird das richtige Ergebnis gezeigt.

Nach der gewünschten Anzahl von Aufgaben wird ins Menü zurückgekehrt. Drücken von F7 beendet das Programm. Vorher erscheint die Auswertung: wie viele Rechnungen insgesamt durchgeführt wurden, wie viele davon richtig waren, wie lange man dafür gebraucht hat und eine grundsätzliche Bewertung der Leistung. Interessierte Eltern können sich so über den Lernerfolg informieren. Programmbeschreibung:

- 10 — 22 Einleitung
- 25 — 26 Variablen für Auswertung festlegen
- 27 — 28 Zeit festlegen
- 30 — 38 Menü
- 39 — 49 Funktionstasten belegen, Schleifenanzahl festlegen
- 50 — 77 Zufallszahlen festlegen
- 100 — 322 Hauptprogramm: Zeigen der Rechnung, Überprüfen der Eingabe, entsprechende Verzweigung,
- 330 — 380 Auswertung: Zeitangabe, Anzahl der durchgeführten Rechnungen, Anzahl der richtig gelösten Rechnungen, Beurteilung der Leistung

Subtrahieren

Das Programm behandelt Subtraktionsaufgaben unterschiedlicher Schwierigkeit (Listing 2). Die Aufgabentypen und die Menütechnik entsprechen denen des Additionsprogramms.

- Programmbeschreibung:
- 10 — 22 Einleitung
 - 25 — 26 Variablen für Auswertung festlegen
 - 27 — 28 Zeit festlegen
 - 30 — 38 Menü

- 39 — 49 Funktionstasten belegen, Schleifenanzahl festlegen
- 50 — 79 Zufallszahlen festlegen
- 100 — 322 Hauptprogramm: Zeigen der Rechnung, Überprüfen der Eingabe, entsprechende Verzweigung, Zwischenbewertung
- 330 — 380 Auswertung: Zeitangabe, Anzahl der durchgeführten Rechnungen, Anzahl der richtig gelösten Rechnungen, Beurteilung der Leistung

Multiplizieren

Bei den Multiplikationsaufgaben (Listing 3) wird das kleine Einmaleins geübt. Ist eine Aufgabe zweimal falsch gelöst worden, wird das richtige Ergebnis gezeigt. Um den Lerneffekt zu erhöhen, soll nun die Rechnung laut gesprochen und das Ergebnis eingetippt werden. Eintippen von „0“ als Ergebnis führt zur Auswertung und zum Programmende.

- Programmbeschreibung:
- 10 — 50 Einleitung
 - 60 — 68 Zeit festlegen
 - 70 — 75 Variablen für Auswertung festlegen
 - 80 — 90 Zufallszahlen festlegen
 - 100 Schleife für zweiten Versuch bei Rechenfehler
 - 110 Anzeige der Rechnung, Warten auf Eingabe
 - 120 — 170 Überprüfen der Eingabe mit entsprechender Verzweigung
 - 200 — 270 Wenn wieder falsch, richtige Rechnung mit Aufforderung, Rechnung zu wiederholen
 - 300 — 330 Auswertung

Dividieren

In Form von „In“-Sätzen werden Divisionsaufgaben geübt (Listing 4). Das Menü bietet zur Auswahl:

- F1 Umkehrung des kleinen Einmaleins (zum Beispiel „3 in 24“)
- F2 Wie bei F1, aber Aufgaben mit Rest („3 in 25“)
- F3 „In-Sätze“, deren Ergebnis über zehn liegt („3 in 48“)
- F5 Wie bei F3, aber Aufgaben mit Rest („3 in 50“)

Anwendung

F7 Programmende
Es muß jeweils das Ergebnis und der verbleibende Rest angegeben werden. Wie beim Additionsprogramm wird nach zweimaliger Falschlösung das richtige Ergebnis gezeigt und zum Schluß eine Gesamtauswertung durchgeführt.
Programmbeschreibung:

10 — 22 Einleitung
25 — 26 Variablen für Auswertung festlegen
27 — 28 Zeit festlegen
30 — 38 Menü
39 — 49 Funktionstasten belegen, Schleifenanzahl festlegen
50 — 68 Zufallszahlen festlegen
100 — 322 Hauptprogramm: Zei-

gen der Rechnung, Überprüfen der Eingabe, entsprechende Verzweigung
330 — 380 Auswertung: Zeitangabe, Anzahl der durchgeführten Rechnungen, Anzahl der richtig gelösten Rechnungen, Beurteilung der Leistung.

(Susanne Jungnikl)

| | |
|--|------|
| 1 REM *** ADDIEREN *** | |
| 2 REM *UEBUNGSPROGRAMM FUER* | |
| 3 REM * GRUNDSCHUELER * | |
| 4 REM * VC-20, C-64, C-16 * | |
| 5 REM * SUSANNE JUNGNIKL * | |
| 6 REM * REINERSDORF 99 * | |
| 7 REM *** A-7522 *** | |
| 8 REM ** 0043/03324/30507 ** | |
| 10 PRINT CHR# (147) | 964 |
| 15 PRINT :PRINT "[2SPACES]KANNST DU GUT [9SPACES]ADDIEREN?" | 3070 |
| 20 PRINT :PRINT "[2SPACES]BEWEISE ES!" | 1555 |
| 22 FOR T=1 TO 2000:NEXT | 990 |
| 25 LET S=0 | 633 |
| 26 LET A=0 | 597 |
| 27 PRINT CHR# (147):PRINT :PRINT "GIB ALS STARTZEIT" | 2801 |
| 28 PRINT :PRINT "[2SPACES]000000 [2SPACES]EIN:":INPUT T1# | 2011 |
| 30 PRINT CHR# (147):PRINT " WAHLE AUS:" | 2340 |
| 31 PRINT :PRINT "F1-----EINER+EINER" | 2753 |
| 32 PRINT :PRINT "F2--REINE 10ER+R.10ER" | 2481 |
| 33 PRINT :PRINT "F3--REINE 10ER+EINER" | 2607 |
| 34 PRINT :PRINT "F4----GEM.10ER+EINER" | 2696 |
| 35 PRINT :PRINT "F5-GEM.10ER+REIN.10ER" | 2863 |
| 36 PRINT :PRINT "F6--GEM.10ER+GEM.10ER" | 2333 |
| 38 PRINT :PRINT "F7-----PROGRAMMENDE" | 2942 |
| 39 GET A#:IF A#="" THEN 39 | 1526 |
| 40 IF ASC (A#)<133 OR ASC (A#)>139 THEN 39 | 2651 |
| 41 IF ASC (A#)=136 THEN 330 | 1404 |
| 42 LET R=0 | 631 |
| 45 PRINT :PRINT :PRINT "WIEVIELE RECHNUNGEN[4SPACES]WILLST DU MACHEN?":INPUT W | 5486 |
| 46 FOR J=1 TO W | 924 |
| 47 A=A+1 | 275 |
| 48 IF J=W+1 THEN GOTO 300 | 1651 |
| 49 ON ASC (A#)-132 GOTO 50,55,60,65,67,70,75 | 3407 |
| 50 X=INT (RND (1)*9+1) | 1623 |
| 52 Y=INT (RND (1)*9+1):GOTO 100 | 2089 |
| 55 X=INT (RND (1)*9+1)*10 | 1872 |
| 57 Y=INT (RND (1)*9+1):GOTO 100 | 2089 |
| 60 X=INT (RND (1)*88+11) | 1608 |
| 62 Y=INT (RND (1)*9+1)*10:GOTO 100 | 2278 |
| 67 X=INT (RND (1)*9+1)*10 | 1872 |
| 69 Y=INT (RND (1)*9+1)*10:GOTO 100 | 2278 |
| 70 X=INT (RND (1)*88+11) | 1576 |
| 72 Y=INT (RND (1)*9+1):GOTO 100 | 2089 |
| 75 X=INT (RND (1)*88+11) | 1608 |
| 77 Y=INT (RND (1)*88+11):GOTO 100 | 2194 |
| 100 FOR I=1 TO 2 | 758 |
| 110 PRINT CHR# (147):PRINT :PRINT :PRINT "[3SPACES]X**Y**":INPUT P | 3053 |
| 120 IF P=X+Y THEN R=R+1 | 1831 |
| 125 IF P=X+Y THEN S=S+1 | 1961 |
| 130 IF P=X+Y THEN PRINT :PRINT :PRINT "[6SPACES]RICHTIG!":GOTO 270 | 3553 |
| 150 IF I=2 THEN GOTO 200 | 1015 |
| 160 IF P<>X+Y THEN PRINT :PRINT :PRINT "LEIDER FALSCH!" | 3132 |
| 170 PRINT :PRINT "RECHNE NOCH EINMAL!" | 2301 |
| 180 FOR T=1 TO 1500:NEXT | 1023 |
| 190 NEXT I | 276 |
| 200 PRINT :PRINT :PRINT "LEIDER WIEDER FALSCH!!!" | 2884 |
| 220 PRINT :PRINT "RICHTIG WAERE GEWESEN:" | 2306 |
| 221 PRINT | 153 |
| 222 PRINT :PRINT :PRINT X**Y**=X+Y | 1997 |
| 230 FOR T=1 TO 4000:NEXT :GOTO 280 | 1523 |
| 270 FOR T=1 TO 1000:NEXT | 958 |
| 280 NEXT J | 278 |
| 300 PRINT CHR# (147):PRINT :PRINT "DU HAST*W*RECHNUNGEN[2SPACES]GERECHNET. | 4531 |
| 305 PRINT :PRINT "DAVON WAREN **R** [3SPACES]RICHTIG!" | 2852 |
| 310 IF R=W THEN PRINT :PRINT :PRINT :PRINT "[7SPACES]BRAVO!!!" | 2595 |
| 322 FOR T=1 TO 8000:NEXT :GOTO 30 | 1402 |
| 330 PRINT CHR# (147) | 964 |
| 332 PRINT :PRINT :PRINT "BENDEITGTE ZEIT:"T1# | 2859 |
| 340 PRINT "[3SPACES](**STD**MIN**SEK) | 1779 |
| 350 PRINT :PRINT :PRINT "INSGESAM HAST DU" | 2229 |
| 352 PRINT :PRINT "[4SPACES]" | 1717 |
| A*RECHNUNGEN" | |
| 355 PRINT :PRINT "GERECHNET." | 1694 |
| 360 PRINT :PRINT "DAVON WAREN **S** [4SPACES]RICHTIG!" | 3127 |
| 365 IF S=A THEN PRINT :PRINT "DU BIST RECHENMEISTER!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!" | 4757 |
| 368 IF S>=A-A/3 AND S<A THEN PRINT :PRINT "DU HAST GUT GERECHNET!" | 4211 |
| 370 IF S<A-A/3 THEN PRINT :PRINT "DU SOLLTEST FLEISSIG UEBEN!!" | 4562 |
| 380 END | 128 |

Listing 1: Addieren

| | |
|---|------|
| 1 REM *** SUBTRAHIEREN *** | |
| 2 REM *UEBUNGSPROGRAMM FUER* | |
| 3 REM * GRUNDSCHUELER * | |
| 4 REM * VC-20, C-64, C-16 * | |
| 5 REM * SUSANNE JUNGNIKL * | |
| 6 REM * REINERSDORF 99 * | |
| 7 REM *** A-7522 *** | |
| 8 REM ** 0043/03324/30507 ** | |
| 10 PRINT CHR# (147) | 964 |
| 15 PRINT :PRINT "[2SPACES]KANNST DU GUT [9SPACES]SUBTRAHIEREN?" | 3191 |
| 20 PRINT :PRINT "[2SPACES]BEWEISE ES!" | 1555 |
| 22 FOR T=1 TO 2000:NEXT | 990 |
| 25 LET S=0 | 633 |
| 26 LET A=0 | 597 |
| 27 PRINT CHR# (147):PRINT :PRINT "GIB ALS STARTZEIT" | 2801 |
| 28 PRINT :PRINT "[2SPACES]000000 [2SPACES]EIN:":INPUT T1# | 2011 |
| 30 PRINT CHR# (147):PRINT " WAHLE AUS:" | 2340 |
| 31 PRINT :PRINT "F1-----EINER-EINER" | 2789 |
| 32 PRINT :PRINT "F2--REINE 10ER-R.10ER" | 2489 |

Anwendung

| | | | |
|---|------|--|------|
| 33 PRINT :PRINT "F3---REINE 10ER-EINER" | 2623 | 125 IF P=X-Y THEN S=S+1 | 1977 |
| 34 PRINT :PRINT "F4-----GEM.10ER-EINER" | 2712 | 130 IF P=X-Y THEN PRINT :PRINT :PRINT * | 3569 |
| 35 PRINT :PRINT "F5-GEM.10ER-REIN.10ER" | 2664 | [6SPACES]RICHTIG!":GOTO 270 | |
| 36 PRINT :PRINT "F6--GEM.10ER-GEM.10ER" | 2335 | 150 IF I=2 THEN GOTO 200 | 1015 |
| 38 PRINT :PRINT "F7-----PROGRAMMENDE" | 2942 | 160 IF P<>X-Y THEN PRINT :PRINT :PRINT | 3164 |
| 39 GET A\$:IF A\$="" THEN 39 | 1526 | "LEIDER FALSCH!" | |
| 40 IF ASC (A\$)<133 OR ASC (A\$)>139 THEN | 2651 | 170 PRINT :PRINT "RECHNE NOCH EINMAL!" | 2301 |
| 39 | | 180 FOR T=1 TO 1500:NEXT | 1023 |
| 41 IF ASC (A\$)=136 THEN 330 | 1404 | 190 NEXT I | 276 |
| 42 LET R=0 | 631 | 200 PRINT :PRINT :PRINT "LEIDER WIEDER | 2894 |
| 45 PRINT :PRINT :PRINT "WIEVIELE | 5486 | FALSCH!!!" | |
| RECHNUNGEN[4SPACES]WILLST DU | | 220 PRINT :PRINT "RICHTIG WARE GEWESEN:" | 2306 |
| MACHEN?":INPUT W | | 221 PRINT | 153 |
| 46 FOR J=1 TO W | 924 | 222 PRINT :PRINT :PRINT X*"-Y*="X-Y | 2062 |
| 47 A=A+1 | 275 | 230 FOR T=1 TO 4000:NEXT :GOTO 200 | 1523 |
| 48 IF J=W+1 THEN GOTO 300 | 1651 | 270 FOR T=1 TO 1000:NEXT | 959 |
| 49 ON ASC (A\$)-132 GOTO 50,55,60,65,66, | 3406 | 280 NEXT J | 278 |
| 70,75 | | 300 PRINT CHR# (147):PRINT :PRINT "DU | 4531 |
| 50 X=INT (RND (1)*9+1) | 1623 | HAST"W"RECHNUNGEN[2SPACES]GERECHNET. | |
| 52 Y=INT (RND (1)*9+1) | 1624 | 305 PRINT :PRINT "DAVON WAREN **R**" | 2852 |
| 53 IF Y>X THEN GOTO 50 | 981 | [3SPACES]RICHTIG!" | |
| 54 GOTO 100 | 556 | 310 IF R=W THEN PRINT :PRINT :PRINT : | 2595 |
| 55 X=INT (RND (1)*9+1)*10 | 1072 | PRINT "[7SPACES]BRAVO!!!" | |
| 57 Y=INT (RND (1)*9+1):GOTO 100 | 2089 | 322 FOR T=1 TO 8000:NEXT :GOTO 30 | 1402 |
| 60 X=INT (RND (1)*88+11) | 1608 | 330 PRINT CHR# (147) | 964 |
| 62 Y=INT (RND (1)*9+1)*10 | 1873 | 332 PRINT :PRINT :PRINT "BENDETIGTE ZEIT:" | 2059 |
| 63 IF Y>X THEN GOTO 60 | 1045 | "T!\$ | |
| 64 GOTO 100 | 556 | 340 PRINT "[3SPACES](**STD**MIN**SEK) | 1779 |
| 66 X=INT (RND (1)*9+1)*10 | 1072 | 350 PRINT :PRINT :PRINT "INSGESAMT HAST | 2229 |
| 67 Y=INT (RND (1)*9+1)*10 | 1073 | DU" | |
| 68 IF Y>X THEN GOTO 66 | 1048 | 352 PRINT :PRINT "[4SPACES]"A"RECHNUNGEN" | 1717 |
| 69 GOTO 100 | 556 | 355 PRINT :PRINT "GERECHNET." | 1694 |
| 70 X=INT (RND (1)*88+11) | 1576 | 360 PRINT :PRINT "DAVON WAREN **S**" | 3127 |
| 72 Y=INT (RND (1)*9+1):GOTO 100 | 2089 | [4SPACES]RICHTIG! | |
| 75 X=INT (RND (1)*88+11) | 1608 | 365 IF S=A THEN PRINT :PRINT "DU BIST | 4757 |
| 77 Y=INT (RND (1)*88+11) | 1609 | RECHENMEISTER!!!!!!!!!!!!!!!!!!!!!!!!!!!! | |
| 78 IF Y>X THEN GOTO 75 | 1239 | !" | |
| 79 GOTO 100 | 556 | 368 IF S>=A-A/3 AND S<A THEN PRINT :PRINT | 4211 |
| 100 FOR I=1 TO 2 | 750 | "DU HAST GUT GERECHNET!" | |
| 110 PRINT CHR# (147):PRINT :PRINT :PRINT | 3085 | 370 IF S<A-A/3 THEN PRINT :PRINT "DU | 4562 |
| "[3SPACES]"X*"-Y*="":INPUT P | | SOLLTEST FLEISSIG UEBEN!!!" | |
| 120 IF P=X-Y THEN R=R+1 | 1847 | 380 END | 128 |

Listing 2: Subtrahieren

| | | | |
|--|------|--|------|
| 1 REM *** I X 1 *** | | 150 IF I=2 THEN GOTO 200 | 1015 |
| 2 REM *UEBUNGSPROGRAMM FUER* | | 160 IF P<>X*Y THEN PRINT :PRINT :PRINT | 3196 |
| 3 REM * GRUNDSCHUELER * | | "LEIDER FALSCH!" | |
| 4 REM * VC-20, C-64, C-16 * | | 170 PRINT :PRINT "RECHNE NOCH EINMAL!" | 2301 |
| 5 REM * SUSANNE JUNGNIKL * | | 180 FOR T=1 TO 1500:NEXT | 1023 |
| 6 REM * REINERSDORF 99 * | | 190 NEXT I | 276 |
| 7 REM *** A-7522 *** | | 200 PRINT :PRINT :PRINT "!!!!WIEDER | 2467 |
| 8 REM ** 0043/03324/30507 ** | | FALSCH!!!" | |
| 10 PRINT CHR# (147) | 964 | 220 PRINT :PRINT "ES HEISST "X*X*Y*="X*Y | 2600 |
| 15 PRINT :PRINT "[2SPACES]KANNST DU DAS | 2051 | 230 PRINT :PRINT :PRINT "SAGE DIE | 5065 |
| IXI?" | | RECHNUNG UND TIPPE DAS ERGEBNIS | |
| 20 PRINT :PRINT "[2SPACES]BEWEISE ES!" | 1555 | [4SPACES]EIN!":INPUT R | |
| 30 PRINT :PRINT :PRINT " | 2725 | 240 IF R=X*Y THEN GOTO 270 | 1725 |
| [2SPACES]WENN DU AUFHOEREN" | | 250 IF R<>X*Y THEN GOTO 200 | 1787 |
| 40 PRINT :PRINT "[2SPACES]WILLST, TIPPE | 2415 | 270 FOR T=1 TO 1000:NEXT | 950 |
| 0 EIN!:" | | 275 A=A+1 | 275 |
| 50 FOR T=1 TO 4000:NEXT | 1054 | 280 GOTO 90 | 441 |
| 60 PRINT CHR# (147) | 964 | 300 PRINT CHR# (147):PRINT :PRINT "DU | 4434 |
| 67 PRINT :PRINT "GIB ALS STARTZEIT " | 1800 | HAST"A"RECHNUNGEN[2SPACES] | |
| 68 PRINT :PRINT "[2SPACES]000000 | 2011 | GERECHNET. | |
| [2SPACES]EIN!":INPUT T!\$ | | 305 PRINT :PRINT "DAVON WAREN **S**" | 2860 |
| 70 LET S=0 | 633 | [3SPACES]RICHTIG!" | |
| 75 LET A=0 | 597 | 306 IF S=A THEN PRINT :PRINT :PRINT : | 3987 |
| 80 X=INT (RND (1)*9+2) | 1631 | PRINT "DU BIST RECHENKOENIG!" | |
| 90 Y=INT (RND (1)*9+2) | 1632 | 307 IF S>=A-A/3 AND S<A THEN PRINT : | 4641 |
| 100 FOR I=1 TO 2 | 750 | PRINT :PRINT :PRINT "DU HAST GUT | |
| 110 PRINT CHR# (147):PRINT :PRINT :PRINT | 3008 | GERECHNET!" | |
| "[3SPACES]"X*X*Y*="":INPUT P | | 308 IF S<A-A/3 THEN PRINT :PRINT :PRINT | 4289 |
| 120 IF P=X*Y THEN S=S+1 | 1993 | :PRINT "DU MUSST MEHR UEBEN!" | |
| 130 IF P=X*Y THEN PRINT :PRINT :PRINT * | 3505 | 310 PRINT :PRINT :PRINT "BENDETIGTE | 2059 |
| [6SPACES]RICHTIG!":GOTO 270 | | ZEIT:"T!\$ | |
| 140 IF P=0 THEN GOTO 300 | 1077 | 320 PRINT "[3SPACES](**STD**MIN**SEK) | 1779 |
| | | 330 END | 128 |

Listing 3: Multiplizieren

| | | | |
|---|------|---|-------|
| 1 REM *** IN-SAETZCHEN *** | | 100 FOR I=1 TO 2 | 750 |
| 2 REM #UEBUNGSPROGRAMM FUER# | | 110 PRINT CHR# (147):PRINT :PRINT :PRINT | 4049 |
| 3 REM * GRUNDSCHUELER * | | "[3SPACES]X*IN*Y*X+Z*="":INPUT Q | |
| 4 REM * VC-20, C-64, C-16 * | | 112 PRINT SPC(10):PRINT "R="":INPUT R | 1935 |
| 5 REM * SUSANNE JUNGNIKL * | | 120 IF Q=Y AND R=2 THEN K=K+1 | 2063 |
| 6 REM * REINERSDORF 99 * | | 125 IF Q=Y AND R=2 THEN S=S+1 | 2143, |
| 7 REM *** A-7522 *** | | 130 IF Q=Y AND R=2 THEN PRINT :PRINT : | 3921 |
| 8 REM ** 0043/03324/30507 ** | | PRINT "[6SPACES]RICHTIG!":GOTO 270 | |
| 10 PRINT CHR# (147) | 964 | 150 IF I=2 THEN GOTO 200 | 1015 |
| 15 PRINT :PRINT "[2SPACES]KANNST DU DIE | 3366 | 160 IF Q<>Y AND R<>2 THEN PRINT :PRINT : | 3920 |
| [9SPACES]IN-SAETZCHEN?" | | PRINT "LEIDER FALSCH!" | |
| 20 PRINT :PRINT "[2SPACES]BEWEISE ES!" | 1555 | 162 IF Q=Y AND R<>2 THEN PRINT :PRINT : | 3510 |
| 22 FOR T=1 TO 2000:NEXT | 990 | PRINT :PRINT "REST FALSCH!" | |
| 25 LET S=0 | 633 | 164 IF Q<>Y AND R=2 THEN PRINT :PRINT : | 4030 |
| 26 LET A=0 | 597 | PRINT :PRINT "RECHNUNG FALSCH!" | |
| 27 PRINT CHR# (147):PRINT :PRINT "GIB | 2801 | 170 PRINT :PRINT "RECHNE NOCH EINMAL!" | 2301 |
| ALS STARTZEIT" | | 180 FOR T=1 TO 1500:NEXT | 1023 |
| | | 190 NEXT I | 276 |
| 28 PRINT :PRINT "[2SPACES]000000 | 2011 | 200 PRINT :PRINT :PRINT "LEIDER WIEDER | 2804 |
| [2SPACES]EIN:"INPUT T1# | | FALSCH!!!" | |
| 30 PRINT CHR# (147):PRINT " WAEHLE AUS: | 2340 | 220 PRINT :PRINT "RICHTIG WAERE GEWESEN:" | 2306 |
| " | | 221 PRINT | 153 |
| 31 PRINT :PRINT "F1-----2.B.: 3 IN 24 " | 2576 | 222 PRINT :PRINT :PRINT X*IN*Y*X+Z*="Y" | 2607 |
| 32 PRINT :PRINT "F2-----2.B.: 3 IN 25 " | 2736 | REST*Z | |
| 33 PRINT :PRINT "F3-----2.B.: 3 IN 48 " | 2515 | 230 FOR T=1 TO 4000:NEXT :GOTO 200 | 1523 |
| 34 PRINT :PRINT "F5-----2.B.: 3 IN 50 " | 2639 | 270 FOR T=1 TO 1000:NEXT | 950 |
| 38 PRINT :PRINT "F7-----PROGRAMMENDE" | 2685 | 280 NEXT J | 278 |
| | | 300 PRINT CHR# (147):PRINT :PRINT "DU | 4531 |
| 39 GET A#:IF A#="" THEN 39 | 1526 | HAST"W"RECHNUNGEN[2SPACES] | |
| 40 IF ASC (A#)<133 OR ASC (A#)>139 THEN | 2651 | GERECHNET. | |
| 39 | | 305 PRINT :PRINT "DAVON WAREN ***K*** | 2796 |
| 41 IF ASC (A#)=136 THEN 330 | 1404 | [3SPACES]RICHTIG!" | |
| 42 LET K=0 | 617 | 310 IF K=W THEN PRINT :PRINT :PRINT : | 2501 |
| 45 PRINT :PRINT :PRINT "WIEVIELE | 5486 | PRINT "[7SPACES]BRAVO!!!" | |
| RECHNUNGEN[4SPACES]WILLST DU | | 322 FOR T=1 TO 8000:NEXT :GOTO 30 | 1402 |
| MACHEN?":INPUT W | | 330 PRINT CHR# (147) | 964 |
| 46 FOR J=1 TO W | 924 | 332 PRINT :PRINT :PRINT "BENoETIGTE | 2059 |
| 47 A=A+1 | 275 | ZEIT:"T1# | |
| 48 IF J=W+1 THEN GOTO 300 | 1651 | 340 PRINT "[3SPACES](**STD**MIN**SEK) | 1779 |
| 49 ON ASC (A#)-132 GOTO 50,55,60,330,65 | 2458 | 350 PRINT :PRINT :PRINT "INSGESAM HAST | 2229 |
| 50 X=INT (RND (1)*9+2) | 1631 | DU" | |
| 51 Y=INT (RND (1)*9+2) | 1632 | 352 PRINT :PRINT "[4SPACES]" | 1717 |
| 52 Z=0:GOTO 100 | 818 | A"RECHNUNGEN" | |
| 55 X=INT (RND (1)*9+2) | 1631 | 355 PRINT :PRINT "GERECHNET." | 1694 |
| 56 Y=INT (RND (1)*20+2) | 1538 | 360 PRINT :PRINT "DAVON WAREN ***S*** | 3127 |
| 58 Z=0:GOTO 100 | 818 | [4SPACES]RICHTIG!" | |
| 60 X=INT (RND (1)*9+2) | 1631 | 365 IF S=A THEN PRINT :PRINT "DU BIST | 4757 |
| 61 Z=INT (RND (1)*9+2) | 1633 | RECHENMEISTER!!!!!!!!!!!!!!!!!!!!!! | |
| 62 IF Z=>X THEN GOTO 60 | 1212 | !!" | |
| 63 Y=INT (RND (1)*20+2):GOTO 100 | 2213 | 368 IF S>=A-A/3 AND S<A THEN PRINT : | 4211 |
| 65 X=INT (RND (1)*9+2) | 1631 | PRINT "DU HAST GUT GERECHNET!" | |
| 66 Z=INT (RND (1)*9+2) | 1633 | 370 IF S<A-A/3 THEN PRINT :PRINT "DU | 4562 |
| 67 IF Z=>X THEN GOTO 65 | 1217 | SOLLTEST FLEISSIG UEBEN!!" | |
| 68 Y=INT (RND (1)*9+2):GOTO 100 | 2097 | 380 END | 120 |

Listing 4: Dividieren

Reparaturen

Reparatur C64 DM 150,—
(Festpreis inkl. Bauteile)

Reparatur Diskettenlaufwerke DM 120,—
(Kosten für Motor bzw. Kopf werden
separat in Rechnung gestellt)

Rollmaus für den C64 DM 195,—

Büro für Software- und Hardware-Entwicklung,
Hackstückstr. 11, 4320 Hattingen,
0 23 24/5 22 40

W. DEDERICHS

Brandheiße Knüllerpreise

| | | | |
|--|-----------------|---------------------------------------|--------|
| Commodore 64, VC 1541 | a. A. | Star-Drucker SG-10 mit Star- oder | |
| SX-64 (Commodore Executive) | a. A. | Wiesemann-Interface | 1129,— |
| Commodore 128, Floppy 1571, Monitor 1902 | | dto. mit Gerlitz-Interf. 8423 | 1179,— |
| (Bestellservice) | je 949,— | Star-Drucker SG-15 mit Star- oder | |
| Commodore PC 10 256 K | 4499,— | Wiesemann-Interface | 1449,— |
| Drucker MPS 801, 389,— | MPS 802 699,— | dto. mit Gerlitz-Interf. 8423 | 1489,— |
| MPS 803 519,— | DPS 1120 1049,— | Star-Drucker SG-10 C (anschlussfertig | |
| Farbplotter 1520 | 299,— | an C 64) | 899,— |
| Epson-Drucker RX 80 + Gerlitz- | | Farbmonitor Commodore 1702 | 729,— |
| Grafikinterface 8422 | 1129,— | Akustikkopier Dataphon S 21 d + | |
| dto. + RX 80 FT + | 1279,— | Kabel + Terminalprogramm | 359,— |
| dto. + FX 85 | 1549,— | Grafiktablett Commodore Grafik | |
| dto. + FX 105 | 2029,— | Commander | 189,— |
| dto. + JX 80 | 2069,— | Disketten | |
| dto. + LX 80 | 1129,— | Scotch 3M SS00 10 St. | 54,— |
| Epson-Drucker LX 90 mit Interface + | | 100 St. | 495,— |
| Kabel für C 64 | 949,— | Scotch 3M DS00 10 St. | 75,— |
| Traktoraufsatz LX 80/90 | 75,— | 100 St. | 695,— |

Alle Preise inkl. MwSt. zuz. Versandkostenpauschale (Warenwert bis DM 1000,—/darüber): Vorauskasse (DM 8,—/20,—), Nachnahme (DM 11.20/23.20), Ausland (DM 18,—/30,—). Lieferung nur gegen Vorauskasse oder per NN; Ausland nur Vorauskasse. Gesamtpreisliste gegen Freiumschlag.

CSV RIEGERT

Schloßhotstr. 5, 7324 Rechberghausen, Tel. (0 71 61) 5 28 89

Dreiecks- verhältnis

Trigonometrische Rechnungen sind ein Alptraum jedes „Nicht-Mathematikers“. Wer nicht daran scheitern, sondern den Stier bei den Hörnern packen will, kann seine Ergebnisse auch vom Computer nachrechnen lassen und so die Fortschritte ständig kontrollieren. 'Trigonometrie' von Michael Wegmer bietet diese Möglichkeit. Das Programm berechnet nach Eingabe der Winkel und Seitenlängen Dreiecke aller Art.

Nach dem Start wird zuerst nach der Anzahl der bekannten Winkel und Seiten gefragt (Zeile 100 bis 630). Mögliche Eingaben sind: zwei Winkel und eine Seite, ein Winkel und zwei Seiten und kein Winkel und drei Seiten. Macht man mehr Angaben als zur Be-

*Trigonometrie ist
Prüfstein vieler
Schülerkarrieren. Mit
einem Programm kann
man die eigenen
Rechenversuche
überprüfen.*

rechnung nötig, beispielsweise zwei Winkel und zwei Seiten, weist das Programm darauf hin. Danach werden die Größen der Winkel und Seiten eingegeben. Als letzte Angabe benötigt das

Programm die Namen der Punkte, an denen die Winkel liegen, und die Namen der Seiten. Zugelassen sind hier nur die Buchstaben a, b und c. Wie in der Geometrie üblich, liegt die Seite a dem Punkt A gegenüber. Entsprechendes gilt für die anderen Seiten.

Nun werden die Berechnungen durchgeführt (Zeile 640 bis 1360). Hierbei kommen hauptsächlich der Sinus- und der Cosinussatz zum Einsatz. Ausgegeben werden schließlich:

- alle Seitenlängen und Winkel
- Fläche
- Umkreisradius
- Inkreisradius
- Höhe

Der Ausgabeteil liegt in den Zeilen 1370 bis 1540. (M. Wegmer)

```

00100 def fn r(x)=int (x*10000+.5)/10000      2482
00110 poke 53272,23:poke 53280,5:poke 53281,  3237
      5:print "[cbm 6]"
00120 def fn b(x)=(-atn (x/( sqrt (1-x*x))+x=  7065
      1)))*(x<>1)+-(x<0+2)*-(x<>1)-[pi]/2*(
      x=1)
00130 def fn a(x)=(-atn ( sqrt (1-x*x)/(x+(x=  6549
      0)))*(x<>0)-[pi]*(x<0)-[pi]/2*(x=0)
00140 print "[clr,cbm a,38shift *,cbm s]"    3355
00150 print "[shift -,38cbm +,shift -]"      5749
00160 print "[shift -,cbm +,rvs,cbm a,      3372
      34shift *,cbm s,off,cbm +,shift -]"
00170 print "[shift -,cbm +,rvs,shift -,    3753
      4spaces]T[shift-space]R[shift-space]]
      [shift-space]O[shift-space]O
      [shift-space]N[shift-space]D
      [shift-space]M[shift-space]E
      [shift-space]T[shift-space]R
      [shift-space]]I[shift-space]E[5spaces,
      shift -,off,cbm +,shift -]"
00180 print "[shift -,cbm +,rvs,cbm z,      3546
      34shift *,cbm x,off,cbm +,shift -]"
00190 print "[shift -,38cbm +,shift -]"      5749
00200 print "[cbm z,38shift *,cbm x]"        3080
00210 print "[2spaces]Geschrieben von      4298
      Michael Wegmer 1983"
00220 print "[2down]Dieses Programm        4295
      berechnet Ihnen Winkel,"
00230 print "Seitenlaengen, Flaechе, Hoehe,  3331
      In- und"
00240 print "Umkreisradius von beliebigen    6113
      Dreiecken."idim a(3),b(3),a*(3),b*(3)
00250 print "[2down]Bitte geben Sie die    3430
      Anzahl der Ihnen"
00260 print "bekannten Winkel ein ! "      2664
00270 get a$:if a*("<0" or a*("<3" then 270  2313
00280 print a$                               427
00290 print "[down]Bitte geben Sie die    3959
      Anzahl der Ihnen"
00300 print "bekannten Seiten ein ! "      2543
00310 get b$:if b*("<1" or b*("<3" then 310  2277
00320 print b$:ix=val (a$):y=val (b$)      2170
00330 if x+y<3 then print "[clr,rvs]Zu wenig 3813
      Eingaben !!!":goto 250
00340 if x+y=3 then 380                      1370
00350 y=3-x:if y=0 then y=1:x=2            2268
00360 print "[down]Es reichen auch"x"Winkel 4908
      und"y"Seite":;if y>1 then print "n"
00370 print :for i=1 to 5000:next          1210
00380 print "[clr]":;if x=0 then 420        1470
00390 if x=2 then print "Bitte geben Sie die 6258
      zwei Winkel in Grad ein !:[down]"goto
      410
00400 print "Bitte geben Sie den einen      4638
      Winkel in Gradein !:[down]"
00410 for i=1 to x:print "Winkel"i:;input  2574
      a(i):next
00420 if y=1 then print "[2down]Bitte geben 5261
      Sie die eine Seite in cm[4spaces]ein
      !:[down]"goto 450
00430 if y=2 then print "[2down]Bitte geben 5315
      Sie die zwei Seiten in cm[3spaces]ein
      !:[down]"goto 450
00440 print "[2down]Bitte geben Sie die drei 3984
      Seiten in cm[3spaces]ein !:[down]"
00450 for i=1 to y:print "Seite "i:;input  2436
      b(i):next
00460 for i=1 to y:if b(i)<=0 then 510       2037
00470 next :if x=0 then 530                 1069
00480 for j=1 to x:if a(j)<=0 or a(j)>=180  2499
      then 510
00490 next                                  130
00500 goto 530                              576
00510 print "[clr,rvs]Falsche Eingabe!" :if 2691
      x=0 then 420
00520 goto 380                              596
00530 if x=0 then 590                       1036
00540 print "[2down]Bitte geben Sie den    3815

```

| | | | |
|--|------|--|------|
| Namen des Punktes" | | | |
| 00550 print "ein, an dem der jeweilige Winkel liegt:[down]" | 4032 | 01080 if s=2 then b1=fn b(b/w)*180/[pi]:c1=180-a1-b1:c=w*sin (c1*[pi]/180):goto 1390 | 7027 |
| 00560 for i=1 to x:print "Winkel"i"? " | 2320 | 01090 c1=fn b(c/w)*180/[pi]:b1=180-a1-c1:b=w*sin (b1*[pi]/180):goto 1390 | 5645 |
| 00570 get a\$:if a\$("&a" or a\$("&c") then 570 | 2387 | 01100 if s=2 then b1=fn b(b/w)*180/[pi]:a1=180-c1-b1:a=w*sin (a1*[pi]/180):goto 1390 | 6943 |
| 00580 print a\$:a\$(i)=a\$:next | 1385 | 01110 a1=fn b(a/w)*180/[pi]:b1=180-a1-c1:b=w*sin (b1*[pi]/180):goto 1390 | 5515 |
| 00590 print "[2down]Bitte geben Sie den Namen der jeweiligen:" | 3885 | 01120 gosub 1300:a2=(b*b+c*c-a*a)/(2*b*c):if abs (a2)>=1 then 1370 | 4248 |
| 00600 print "Seite ein:[down]" | 1291 | 01130 a1=fn a(a2)*180/[pi] | 1498 |
| 00610 for i=1 to y:print "Seite "i"? " | 2166 | 01140 b2=(a*a+c*c-b*b)/(2*a*c):if abs (b2)>=1 then 1370 | 4209 |
| 00620 get b\$:if b\$("&a" or b\$("&c") then 620 | 2365 | 01150 b1=fn a(b2)*180/[pi] | 1563 |
| 00630 print b\$:b\$(i)=b\$:next | 1407 | 01160 c2=(a*a+b*b-c*c)/(2*a*b):if abs (c2)>=1 then 1370 | 4237 |
| 00640 on y goto 650,740,1120 | 1355 | 01170 c1=fn a(c2)*180/[pi]:goto 1390 | 2300 |
| 00650 gosub 1180:u=180-a(1)-a(2):if u<=0 then 1370 | 3802 | 01180 if b\$(1)="a" then a=b(1):r=1:return | 1935 |
| 00660 gosub 1210:gosub 1240 | 1247 | 01190 if b\$(1)="b" then b=b(1):r=2:return | 1960 |
| 00670 if (p=1 and q=2) or (p=2 and q=1) then c1=u:goto 700 | 2984 | 01200 c=b(1):r=3:return | 1109 |
| 00680 if (p=2 and q=3) or (p=3 and q=2) then a1=u:goto 700 | 3085 | 01210 if a\$(1)="a" then a1=a(1):p=1:return | 2016 |
| 00690 if (p=1 and q=3) or (p=3 and q=1) then b1=w | 2689 | 01220 if a\$(1)="b" then b1=a(1):p=2:return | 2057 |
| 00700 on r goto 710,720,730 | 1287 | 01230 c1=a(1):p=3:return | 1474 |
| 00710 v=a/sin (a1*[pi]/180):b=v*sin (b1*[pi]/180):c=v*sin (c1*[pi]/180):goto 1390 | 6273 | 01240 if a\$(2)="a" then a1=a(2):q=1:return | 2041 |
| 00720 v=b/sin (b1*[pi]/180):a=v*sin (a1*[pi]/180):c=v*sin (c1*[pi]/180):goto 1390 | 6276 | 01250 if a\$(2)="b" then b1=a(2):q=2:return | 2082 |
| 00730 w=c/sin (c1*[pi]/180):a=v*sin (a1*[pi]/180):b=v*sin (b1*[pi]/180):goto 1390 | 6279 | 01260 c1=a(2):q=3:return | 1507 |
| 00740 if a\$(1)=b\$(1) or a\$(1)=b\$(2) then 840 | 2530 | 01270 if b\$(2)="a" then a=b(2):s=1:return | 1828 |
| 00750 gosub 1210:gosub 1180:gosub 1270 | 2060 | 01280 if b\$(2)="b" then b=b(2):s=2:return | 1853 |
| 00760 w=b(1)*b(1)+b(2)*b(2)-2*b(1)*b(2)*cos (a(1)*[pi]/180):if w<=0 then 1370 | 5907 | 01290 c=b(2):s=3:return | 1253 |
| 00770 on p goto 780,790,800 | 1362 | 01300 for i=1 to 3:if b\$(i)="a" then a=b(i):goto 1320 | 3034 |
| 00780 a=sqr (w):goto 810 | 1303 | 01310 next | 130 |
| 00790 b=sqr (w):goto 820 | 1306 | 01320 for i=1 to 3:if b\$(i)="b" then b=b(i):goto 1340 | 2943 |
| 00800 c=sqr (w):goto 830 | 1309 | 01330 next | 130 |
| 00810 w=a/sin (a1*[pi]/180):b1=fn b(b/w)*180/[pi]:c1=180-a1-b1:goto 1390 | 5409 | 01340 for i=1 to 3:if b\$(i)="c" then c=b(i):goto 1360 | 3107 |
| 00820 w=b/sin (b1*[pi]/180):a1=fn b(a/w)*180/[pi]:c1=180-a1-b1:goto 1390 | 5412 | 01350 next | 130 |
| 00830 w=c/sin (c1*[pi]/180):a1=fn b(a/w)*180/[pi]:b1=180-a1-c1:goto 1390 | 5486 | 01360 return | 142 |
| 00840 gosub 1210:gosub 1180:gosub 1270 | 2060 | 01370 print "[clr,2down]Mit diesen Werten kann kein Dreieck be-" | 4483 |
| 00850 if r=p then 990 | 898 | 01380 print "rechnet werden !!!":goto 1510 | 2219 |
| 00860 on p goto 870,890,910 | 1251 | 01390 print "[clr,down]Seiten:" tab(15)"[rvs]a=[off]" fn r(a)"cm" | 2854 |
| 00870 w=a/sin (a1*[pi]/180):if w>0 then 950 | 2678 | 01400 print tab(15)"[down,rvs]b=[off]" fn r(b)"cm" | 1628 |
| 00880 goto 1370 | 627 | 01410 print tab(15)"[down,rvs]c=[off]" fn r(c)"cm" | 1757 |
| 00890 w=b/sin (b1*[pi]/180):if w>0 then 970 | 2874 | 01420 print "[down]Winkel:" tab(15)"[rvs]alpha=[off]" fn r(a)"Grad" | 3316 |
| 00900 goto 1370 | 627 | 01430 print tab(15)"[down,rvs]beta =[off]" fn r(b)"Grad" | 2697 |
| 00910 w=c/sin (c1*[pi]/180):if w>0 then 930 | 2941 | 01440 print tab(15)"[down,rvs]gamma=[off]" fn r(c)"Grad" | 2788 |
| 00920 goto 1370 | 627 | 01450 print "[down]Flaeche:" tab(15)"[rvs]A=[off]" fn r(.5*a*b*sin (c1*[pi]/180))"cm^2" | 4922 |
| 00930 if r=1 then a1=fn b(a/w)*180/[pi]:b1=180-a1-c1:b=w*sin (b1*[pi]/180):goto 1390 | 6904 | 01460 print "[down]Umkreisradius:" tab(15)"[rvs]r=[off]" fn r(a/sin (a1*[pi]/180)/2)"cm" | 5343 |
| 00940 b1=fn b(b/w)*180/[pi]:a1=180-b1-c1:a=w*sin (a1*[pi]/180):goto 1390 | 5499 | 01470 s=(a+b+c)/2:print "[down] Inkreisradius:" tab(15)"[rvs]r=[off]" | 4784 |
| 00950 if r=2 then b1=fn b(b/w)*180/[pi]:c1=180-a1-b1:c=w*sin (c1*[pi]/180):goto 1390 | 7025 | 01480 print fn r(tan (a1*[pi]/360)*(s-a))"cm" | 2657 |
| 00960 c1=fn b(c/w)*180/[pi]:b1=180-a1-c1:b=w*sin (b1*[pi]/180):goto 1390 | 5645 | 01490 print "[down]Hoehe:" tab(15)"[rvs]hc=[off]" | 2356 |
| 00970 if r=1 then a1=fn b(a/w)*180/[pi]:c1=180-a1-b1:c=w*sin (c1*[pi]/180):goto 1390 | 6977 | 01500 print fn r(sqr (b*b-(b*cos (a1*[pi]/180))*(b*cos (a1*[pi]/180))))"cm" | 4622 |
| 00980 c1=fn b(c/w)*180/[pi]:a1=180-b1-c1:a=w*sin (a1*[pi]/180):goto 1390 | 5564 | 01510 print "[2down]Wollen Sie noch weitere Dreiecke be-[4spaces]rechnen lassen ? " | 6058 |
| 00990 on p goto 1000,1020,1040 | 1315 | 01520 get a\$:if a\$("&j") and a\$("&n") then 1520 | 2542 |
| 01000 w=a/sin (a1*[pi]/180):if w>0 then 1060 | 2664 | 01530 if a\$="j" then run | 1154 |
| 01010 goto 1370 | 627 | 01540 print "[clr]":poke 53272,21:end | 1687 |
| 01020 w=b/sin (b1*[pi]/180):if w>0 then 1100 | 2792 | | |
| 01030 goto 1370 | 627 | | |
| 01040 w=c/sin (c1*[pi]/180):if w>0 then 1060 | 2799 | | |
| 01050 goto 1370 | 627 | | |
| 01060 if s=1 then a1=fn b(a/w)*180/[pi]:b1=180-a1-c1:b=w*sin (b1*[pi]/180):goto 1390 | 6906 | | |
| 01070 b1=fn b(b/w)*180/[pi]:a1=180-b1-c1:a=w*sin (a1*[pi]/180):goto 1390 | 5499 | | |

Input mit Komfort

*Der INPUT-Befehl des
C64 hat Schwächen.*

Jürgen und Roland

*Hank wollen sich damit
nicht herumärgern und
haben kurzerhand eine
bessere Eingabe-
Routine geschrieben.*

Zur Dateneingabe ist im Basic der Befehl INPUT vorgesehen. Doch der läßt unerwünschte Cursorbewegungen zu, die das schönste Menü zerstören. Bei der Eingabe von Zahlen werden dummerweise auch Buchstaben angenommen. Nach dem Drücken von Return steigt das Programm dann mit TYPE MISMATCH ERROR aus.

Die neue Eingabe-Routine beseitigt diese Mängel und bietet einige Extras. Mit SYS 49152, X, Y, BEMERKTEXT, LÄNGE, VARIABLENTYP, VARIABLE werden Strings oder Zahlen an einer frei wählbaren Bildschirmstelle übernommen. X und Y stehen für die Koordinaten, an die der Bemerktext geschrieben wird. Er kann auch als Stringvariable angegeben werden oder ganz entfallen. Dahinter steht in eckigen Klammern das Eingabefeld. Seine Größe wird durch LÄNGE bestimmt. Es darf nicht über den rechten Zeilenrand hinausragen. Bei VARIA-

BLENTYP=0 wird eine Stringvariable übernommen, bei Typ=1 eine Realzahl. Die Eingabe wird bei der angegebenen Variable abgelegt. Innerhalb des Eingabefeldes bewegt man sich mit [RIGHT] und [LEFT]. [DELETE] und [INSERT] funktionieren wie gewöhnlich. Mit [HOME] wird das Zeichen unter dem Cursor ge-

löscht. Die [CLR]-Funktion ist auf die Eingabezone beschränkt. Böswillige Zeitgenossen können den Bildschirmaufbau nicht verhunzen, denn Cursorbewegungen nach oben und unten werden blockiert.

Bei der Stringeingabe ist jedes sinnvolle Zeichen zugelassen. Wird eine Zahl erwartet, so werden nur die Zahlenzeichen, das Minuszeichen, der Dezimalpunkt und das „E“ für die Exponential-schreibweise akzeptiert.

Zur Vorgabe einer Antwort muß vor dem SYS der erwartete Text in die spätere Eingabezone gedruckt werden. Nun nur noch die Return-Taste gedrückt, und schon ist die Antwort im Kasten.

Mit SYS 50208 wird der neue Basic-Befehl ←I definiert. Das Eingabe-Programm wird jetzt mit ←I X, Y, BEMERKTEXT, LÄNGE, VARIABLENTYP, VARIABLE aufgerufen. SYS 50219 schaltet diese Betriebsart wieder aus.

```

0010 ; eingaberoutine
0020 ; start mit
0030 ; sys 49152,x,y,bemerk#,
                                laenge,typ,variable
0040 .os
0050 .ba %c000
0060 endlaenge .de %33c
0070 position .de %33d
0080 altzeich .de %33e
0090 aktlaenge .de %33f
0100 neuzeich .de %340
0110 cursorx .de %341
0120 cursory .de %342
0130 cursorx1 .de %343
0140 cursory1 .de %344
0150 vartyp .de %345
0160 puffer .de %346
0170 chkcom .de %aefd
0180 getbyt .de %b79e
0190 bsout .de %ffd2
0200 cursor .de %fff0
0210 frmavl .de %ad9e
0220 sid .de %4272
0230 ; cursor positionieren
C000- 20 FD AE 0240 .de %33c jsr chkcom
C003- 20 9E B7 0250 .de %b79e jsr getbyt
C006- 8A 0260 .de %ffd2 txa
C007- 4B 0270 .de %33d pha
C008- 20 FD AE 0280 .de %aefd jsr chkcom
C00B- 20 9E B7 0290 .de %b79e jsr getbyt
C00E- 6B 0300 .de %33f pla
C00F- AB 0310 .de %340 tay
C010- 20 1B C4 0320 .de %341 jsr curset
C013- 20 FD AE 0330 .de %342 jsr chkcom

```

```

0340 ; bemerktext uebernehmen
C016- 20 9E AD 0350 .de %342 jsr frmavl
C019- A5 0D 0360 .de %343 lda 13
C01B- F0 12 0370 .de %344 beq bu2
C01D- 20 A6 B6 0380 .de %b6a6 jsr %b6a6
C020- AA 0390 .de %345 tax
C021- A0 00 0400 .de %346 ldy #0
C023- EB 0410 .de %347 inx
C024- CA 0420 bu1 .de %348 dex
C025- F0 0B 0430 .de %349 beq bu2
C027- B1 22 0440 .de %34a lda ($22),y
C029- 20 D2 FF 0450 .de %34b jsr bsout
C02C- CB 0460 .de %34c iny
C02D- D0 F5 0470 .de %34d bne bu1
C02F- A9 5B 0480 bu2 .de %34e lda #'ä'
C031- 20 D2 FF 0490 .de %34f jsr bsout
C034- 20 1C C4 0500 .de %350 jsr curget
C037- BE 42 03 0510 .de %351 stx cursory
C03A- BC 41 03 0520 .de %352 sty cursorx
0530 ; laenge
C03D- A5 0D 0540 .de %353 lda 13
C03F- D0 11 0550 .de %354 bne bu5
C041- 20 8D AD 0560 .de %355 jsr %ad8d
C044- 20 8B B1 0570 .de %356 jsr %b1b8
C047- A6 64 0580 .de %357 ldx %64
C049- D0 04 0590 .de %358 bne bu7
C04B- A6 65 0600 .de %359 ldx %65
C04D- D0 0D 0610 .de %35a bne bu6
C04F- 4C 4B B2 0620 bu7 .de %35b jmp %b248
C052- 20 FD AE 0630 bu5 .de %35c jsr chkcom
C055- 20 9E B7 0640 .de %35d jsr getbyt
C05B- E0 00 0650 .de %35e cpx #0
C05A- F0 F3 0660 .de %35f beq bu7
C05C- BE 3C 03 0670 bu6 .de %360 stx endlaenge

```


| | | | | | |
|-------|------------|------|---------------------------------------|------------|--|
| | 3720 i35.1 | | | | |
| C319- | AC 3C 03 | 3730 | ldy | endlaenge | |
| C31C- | C8 | 3740 | iny | | |
| C31D- | CC 3F 03 | 3750 | cpy | aktlaenge | |
| C320- | DO 0E | 3760 | bne | i36 | |
| C322- | AC 3F 03 | 3770 | ldy | aktlaenge | |
| C325- | CC 3D 03 | 3780 | cpy | position | |
| C328- | DO 06 | 3790 | bne | i36 | |
| C32A- | 20 E4 C3 | 3800 | jsr | beep | |
| C32D- | 4C FF C0 | 3810 | jmp | i4 | |
| C330- | AD 3D 03 | 3820 | lda | position | |
| C333- | CD 3F 03 | 3830 | cmp | aktlaenge | |
| C336- | B0 12 | 3840 | bcs | i37 | |
| C338- | AC 3D 03 | 3850 | ldy | position | |
| C33B- | AD 40 03 | 3860 | lda | neuzeich | |
| C33E- | 99 45 03 | 3870 | sta | puffer-1,y | |
| C341- | EE 3D 03 | 3880 | inc | position | |
| C344- | 20 E4 C3 | 3890 | jsr | beep | |
| C347- | 4C 59 C3 | 3900 | jmp | i38 | |
| C34A- | AC 3D 03 | 3910 | ldy | position | |
| C34D- | AD 40 03 | 3920 | lda | neuzeich | |
| C350- | 99 45 03 | 3930 | sta | puffer-1,y | |
| C353- | EE 3F 03 | 3940 | inc | aktlaenge | |
| C356- | EE 3D 03 | 3950 | inc | position | |
| C359- | AD 40 03 | 3960 | lda | neuzeich | |
| C35C- | 20 D2 FF | 3970 | jsr | bsout | |
| C35F- | AD 3D 03 | 3980 | lda | position | |
| C362- | CD 3C 03 | 3990 | cmp | endlaenge | |
| C365- | F0 07 | 4000 | beq | i38.1 | |
| C367- | 90 05 | 4010 | bcc | i38.1 | |
| C369- | CE 3D 03 | 4020 | dec | position | |
| C36C- | C6 D3 | 4030 | dec | 211 | |
| C36E- | 4C FF C0 | 4040 | jmp | i4 | |
| C371- | AC 3D 03 | 4050 | ldy | position | |
| C374- | B9 45 03 | 4060 | lda | puffer-1,y | |
| C377- | DO 02 | 4070 | bne | i41 | |
| C379- | A9 20 | 4080 | lda | #32 | |
| C37B- | 20 D2 FF | 4090 | jsr | bsout | |
| C37E- | A9 0D | 4100 | lda | #13 | |
| C380- | 20 D2 FF | 4110 | jsr | bsout | |
| C383- | CE 3F 03 | 4120 | dec | aktlaenge | |
| C386- | 20 FD AE | 4130 | jsr | chkcom | |
| C389- | AD 45 03 | 4140 | lda | vartyp | |
| C38C- | DO 2D | 4150 | bne | i45 | |
| | | 4160 | ; string-variable | | |
| C38E- | 20 BB B0 | 4170 | jsr | %b0Bb | |
| C391- | B5 49 | 4180 | sta | \$49 | |
| C393- | B4 4A | 4190 | sty | \$4a | |
| C395- | 20 A3 B6 | 4200 | jsr | %b6a3 | |
| C398- | AD 3F 03 | 4210 | lda | aktlaenge | |
| C39B- | F0 18 | 4220 | beq | i44 | |
| C39D- | 20 75 B4 | 4230 | jsr | %b475 | |
| C3A0- | A0 02 | 4240 | ldy | #2 | |
| C3A2- | B9 61 00 | 4250 | lda | %61,y | |
| C3A5- | 91 49 | 4260 | sta | (\$49),y | |
| C3A7- | BB | 4270 | dey | | |
| C3AB- | 10 FB | 4280 | bpl | i42 | |
| C3AA- | CB | 4290 | iny | | |
| C3AB- | B9 46 03 | 4300 | lda | puffer,y | |
| C3AE- | 91 62 | 4310 | sta | (\$62),y | |
| C3B0- | CB | 4320 | iny | | |
| C3B1- | C4 61 | 4330 | cpy | %61 | |
| C3B3- | DO F6 | 4340 | bne | i43 | |
| C3B5- | A9 00 | 4350 | lda | #0 | |
| C3B7- | 8D 8A 02 | 4360 | sta | 650 | |
| C3BA- | 60 | 4370 | rts | | |
| | | 4380 | ; real-variable | | |
| C3BB- | A9 46 | 4390 | lda | #1,puffer | |
| C3BD- | B5 22 | 4400 | sta | \$22 | |
| C3BF- | A9 03 | 4410 | lda | %h,puffer | |
| C3C1- | B5 23 | 4420 | sta | \$23 | |
| C3C3- | AD 3F 03 | 4430 | lda | aktlaenge | |
| C3C6- | AB | 4440 | tay | | |
| C3C7- | A2 00 | 4450 | ldx | #0 | |
| C3C9- | B6 0D | 4460 | stx | 13 | |
| C3CB- | 20 B5 B7 | 4470 | jsr | %b7b5 | |
| C3CE- | 20 BB B0 | 4480 | jsr | %b0Bb | |
| C3D1- | B5 49 | 4490 | sta | \$49 | |
| C3D3- | B4 4A | 4500 | sty | \$4a | |
| C3D5- | A5 0D | 4510 | lda | 13 | |
| C3D7- | DO 08 | 4520 | bne | i46 | |
| C3D9- | A5 49 | 4530 | lda | \$49 | |
| C3DB- | 20 DO BB | 4540 | jsr | %bbd0 | |
| C3DE- | 4C B5 C3 | 4550 | jmp | i44 | |
| | | | ;type mismatch error | | |
| C3E1- | 4C 99 AD | 4560 | jmp | %ad99 | |
| | | 4570 | beep | | |
| C3E4- | A0 32 | 4580 | ldy | #50 | |
| C3E6- | A2 00 | 4590 | ldx | #0 | |
| C3E8- | CA | 4600 | dex | | |
| C3E9- | DO FD | 4610 | bne | bel | |
| C3EB- | B8 | 4620 | dey | | |
| C3EC- | DO FA | 4630 | bne | bel | |
| C3EE- | A9 05 | 4640 | lda | #5 | |
| C3F0- | BD 18 D4 | 4650 | sta | sid+24 | |
| C3F3- | A9 15 | 4660 | lda | #21 | |
| C3F5- | BD 05 D4 | 4670 | sta | sid+5 | |
| C3F8- | A9 A0 | 4680 | lda | #160 | |
| C3FA- | BD 06 D4 | 4690 | sta | sid+6 | |
| C3FD- | A9 46 | 4700 | lda | #70 | |
| C3FF- | BD 01 D4 | 4710 | sta | sid+1 | |
| C402- | A9 11 | 4720 | lda | #17 | |
| C404- | BD 04 D4 | 4730 | sta | sid+4 | |
| C407- | A0 1E | 4740 | ldy | #30 | |
| C409- | A2 00 | 4750 | ldx | #0 | |
| C40B- | CA | 4760 | dex | | |
| C40C- | DO FD | 4770 | bne | be2 | |
| C40E- | B8 | 4780 | dey | | |
| C40F- | DO FA | 4790 | bne | be2 | |
| C411- | BC 04 D4 | 4800 | sty | sid+4 | |
| C414- | BC 18 D4 | 4810 | sty | sid+24 | |
| C417- | 60 | 4820 | rts | | |
| | | 4830 | ; cursor setzen | | |
| | | 4840 | curset | | |
| C418- | 18 | 4850 | clc | | |
| C419- | 4C F0 FF | 4860 | jmp | cursor | |
| | | 4870 | ; cursor holen | | |
| | | 4880 | curget | | |
| C41C- | 38 | 4890 | sec | | |
| C41D- | 4C F0 FF | 4900 | jmp | cursor | |
| | | 4910 | : als basic-befehl mit ' i ' aufrufen | | |
| | | 4920 | newinit | | |
| C420- | A9 38 | 4930 | lda | #1,neurout | |
| C422- | BD 08 03 | 4940 | sta | \$308 | |
| C425- | A9 C4 | 4950 | lda | %h,neurout | |
| C427- | BD 09 03 | 4960 | sta | \$309 | |
| C42A- | 60 | 4970 | rts | | |
| | | 4980 | ; nicht mehr als befehl benutzen | | |
| | | 4990 | newreset | | |
| C42B- | A9 E4 | 5000 | lda | #e4 | |
| C42D- | BD 08 03 | 5010 | sta | \$308 | |
| C430- | A9 A7 | 5020 | lda | #a7 | |
| C432- | BD 09 03 | 5030 | sta | \$309 | |
| C435- | 60 | 5040 | rts | | |
| C436- | 00 | 5050 | hv1 | .by 0 | |
| C437- | 00 | 5060 | hv2 | .by 0 | |
| | | 5070 | ; ueberpruefung auf neuen befehl | | |
| | | 5080 | neurout | | |
| C43B- | 20 73 00 | 5090 | jsr | \$73 | |
| C43B- | F0 04 | 5100 | beq | nr1 | |
| C43D- | C9 5F | 5110 | cmp | #' ' | |
| C43F- | F0 03 | 5120 | beq | nr2 | |
| C441- | 4C E7 A7 | 5130 | jmp | %a7e7 | |
| C444- | 20 73 00 | 5140 | nr2 | jsr \$73 | |
| C447- | C9 49 | 5150 | cmp | #' I' | |
| C449- | F0 03 | 5160 | beq | nr3 | |
| C44B- | 4C 08 AF | 5170 | jmp | %af08 | |
| C44E- | 20 73 00 | 5180 | nr3 | jsr \$73 | |
| C451- | 20 03 C0 | 5190 | jsr | %c003 | |
| C454- | 4C AE A7 | 5200 | jmp | %a7ae | |
| | | 5210 | .en | | |

Impressum

Chefredakteur: Manfred S. Schmidt (mss)
Redaktion: Dr. Horst Höflin (hh), Barbara Mittl (bam), Wolfgang Schnabel (wosch)
Redaktionelle Mitarbeiter: Christoph Grunwald (cgr), Peter Kierings (pk), Siegfried Schwarze (sis), Andreas Vichr (avi), Peter Vogel (pv)
Redaktionsassistentin: Siggie Pesch (sp)
Gestaltung: Darinka Bratuscha (verantwortlich), Karin Wirth
Anschrift der Redaktion: RUN, Postfach 400 429, Friedrichstraße 31, 8000 München 40, Telefon: 0 89/3 81 72-0, Telex: 5 215 350 comw d, Telekopierer: 0 89/3 81 72-1 09
Auslandsredaktionen:
Österreich: Erich K. Surböck, c/o ADV, Trattnerhof 2, A-1010 Wien, Tel.: 00 43/222/52 32 71
Schweiz: Günter Schilling, Karl-Jaspers-Allee 4, CH-4052 Basel, Tel.: 00 41/61/42 47 16
USA: CW-COMMUNICATIONS INC., 375 Cochituate Road, Box 880, USA-Frammingham, Mass. 01701, Tel.: 001/617/879 07 00, Tx.: 00230/951 183 computwrd fmh
© Copyright CW-Publikationen Verlags GmbH
Anzeigenpreise: Für Produktanzeigen fordern Sie bitte unsere Mediaunterlagen an. MARKTFÜHRER: Der mm einseitig DM 5,-, Chiffregebühr DM 10,-, Fließsatzanzeigen nach Zeilen DM 7,-, gewerblich zzgl. MwSt., privat DM 5,- inkl. MwSt. (Z. Zt. ist die Anzeigenpreisliste Nr. 1 vom 1. 4. 1984 gültig)
Erscheinungsweise: monatlich
Abonnement-Bestellungen: Direkt beim Verlag (Anschrift s. u.) oder Buchhandel
Vertrieb Handelsauflage: MZV Moderner Zeitschriften Vertrieb GmbH, Breslauer Straße 5, 8067 Eching, Tel.: 089/3 19 10 67, Telex: 522 656.
Bezugspreise: RUN erscheint jeweils Mitte des Vormonats. EV-Preis DM/sfr 5,50. Im Inland beträgt der Jahresbezugspreis DM 58,- inkl. Vertriebskosten und gesetzl. MwSt. für 12 Ausgaben. Auslandsendpreis: DM 66,-; für die Schweiz Sfr 66,-. Luftpostversand auf Anfrage. Der Abonnent kann seine Bestellung innerhalb einer Woche nach Erhalt des ersten Exemplars mit einer schriftlichen Mitteilung an den Verlag widerrufen. Das Abonnement verlängert sich automatisch um ein Jahr, wenn es nicht acht Wochen vor Ablauf schriftlich gekündigt wird. Im Falle höherer Gewalt hat der Abonnent keinen Anspruch auf Lieferung oder auf Rückerstattung der Abonnementgebühr.
Telefon-Durchwahl: 089/3 81 72 ()
Vertriebsleitung: Brigitte Schleibinger (-153/-155)
Leser-Service (Vertrieb): Gerlinde Abdullah (-218)
Anzeigenleitung: Sylvia Stier (-118); (verantwortlich für Anzeigen, Anschrift siehe unter Anzeigen)
Anzeigenverkauf: Barbara Schönberger (130), Marianne Gad (201)
Anzeigendisposition: Ursel Sauter (-126)
Anschrift für Anzeigen und Vertrieb: RUN, Postfach 400 429, Friedrichstraße 31, 8000 München

40, Telefon: 0 89/3 81 72-0, Telex: 5 215 350 comw d, Telekopierer 089/3 81 72-1 09
Objektdisposition: Rainer Oberländer (Leitung), Stefan Liba
Druck und Beilagen: Carl Gerber Grafische Betriebe, Muthmannstraße 4, 8000 München 45, Tel.: 089/3 23 93-233 (Anschrift für Beilagen)
Zahlungsmöglichkeiten: Bayerische Vereinsbank, BLZ 700 202 70, Konto-Nr. 116 000, Pschk. München 97 40-800
Für Abonnenten: Bayerische Vereinsbank, BLZ 700 202 70, Konto-Nr. 111 888, Pschk. München 233 900 808, Schweizerische Volksbank Winterthur, Kto.-Nr. KK 10.251 730-0
Erfüllungsort, Gerichtsstand: München
Verlag: CW-Publikationen Verlagsgesellschaft mbH, Friedrichstraße 31, 8000 München 40, Telefon: 089/3 81 72-0, Telex: 5 215 350 comw d, Telekopierer: 089/3 81 72-109
Produktionsleitung: Heinz Zimmermann
Redaktionsdirektor: Dieter Eckbauer
Geschäftsführer: Eckhard Utpadel, Walter Boyd, Patrick McGovert
ISSN-Nr. 0176-1927
Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e. V. (IVW), Bad Godesberg.



RUN ist ein Mitglied der CW-Communications/Inc.-Gruppe, der Welt größter Verleger für computerbezogene Informationen. Die Gruppe veröffentlicht 55 Computer-Publikationen in mehr als 20 Ländern. Neun Millionen Menschen lesen eine oder mehrere Publikationen dieser Gruppe pro Monat.
 Mitglieder dieser Verlagsgruppe sind:
 Argentinien: Computerworld/Argentina; Asien: The Asian Computerworld; Australien: Computerworld Australia, Australian PC World, Macworld; Brasilien: DataNews, MicroMundo, PC Mundo; China: China Computerworld; Dänemark: Computerworld/Danmark, PC World, Run (Commodore); Finnland: Mikro; Frankreich: Le Monde Informatique, Golden (Apple), OPC (IBM), Distributive; Deutschland: Computerwoche, microcomputerwelt, PC Welt, Software-Markt, Computer Business, RUN, apple's, CW-Edition/Seminare; Italien: Computerworld Italia, PC Magazine; Japan: Computerworld Japan; Mexiko: Computerworld/Mexico, CompuMundo; Niederlande: Computerworld Benelux, PC World Benelux; Norwegen: Computerworld Norge, PC Mikrodata; Saudi Arabien: Saudi Computerworld; Spanien: Computerworld Espana, PC World, Commodore World; Schweden: ComputerSweden, Mikrodatorn, Svenska PC; Großbritannien: Computer Management, Computer News, PC Business World, Computer Business Europe; Venezuela: Computerworld Venezuela; USA: Computerworld, Hot CoCo, inCider, Infoworld, MacWorld, Micro Marketworld, PC World, Run, 73 Magazine, 80 Micro, Focus Publications, On Communications.

RUN board-

Gebrauchsanweisung

1. Anzeigentext in die vorgedruckten Zeilen der Karte schreiben — maximal 6 Zeilen mit Anschrift oder Telefon-Nummer (pro Buchstabe, Satzzeichen, Wortzwischenraum = 1 Kästchen).
2. Rubrik, Bereich und Zahlungsweise ankreuzen, Adresse auf der Rückseite der Karte angeben.
3. Datum und Unterschrift nicht vergessen
4. Karte zusammen mit 5-DM-Schein oder V-Scheck über DM 5,- in Umschlag stecken.
5. Umschlag zukleben, ausreichend frankieren und abschicken an: CW-Publikationen — RUNboard — Postfach 40 04 29 8000 München 40
6. Die Anzeige erscheint im nächstmöglichen RUNboard.

Ich habe die Gelegenheit genutzt!

MEMO

Ich haben am _____ eine Gelegenheits-Anzeige im RUN-board bestellt. Sechs Zeilen zum Taschengeld-Tarif von nur DM 5,- inkl. ges. Mehrwertsteuer. Die DM 5,- habe ich der Bestellung beigelegt als

- V-Scheck Geldschein

RUN-o-thek Disketten-Service

MEMO

Ich habe am _____ bei der RUN-o-thek _____ Diskette(n) mit den Programmen aus RUN-

Ausgabe _____ bestellt. Preis pro Diskette: DM 21,80 plus DM 3,- Versandkosten = DM 24,80

- Bezahlt mit V-Scheck-Nr. _____
 Konto-Nummer _____
 BLZ/Geldinstitut _____
 Ich bezahle per Nachnahme.

Inserentenverzeichnis

| | | | |
|---------------------|-------|-------------------------|-------|
| Abacus Software | 125 | Mükra Datentechnik | 79 |
| apd | 99 | Okidata | 3. US |
| Boston Computer | 5 | Omikron-Software | 135 |
| Bühler Elektronik | 99 | Rat & Tat | 57 |
| Christiani | 149 | Peter & Partner | 66 |
| Dederichs | 139 | Riegert | 139 |
| Dela Elektronik | 129 | Roos Elektronik | 135 |
| Enterprise Computer | 4. US | Roßmüller Datentechnik | 94 |
| HDS Prüftechnik | 99 | Rushware | 2. US |
| M. A. G. Software | 99 | Stockem Computertechnik | 66 |
| Maxell Europe | 29 | Wedewer | 99 |
| Möllenbeck | 99 | Zaporowski | 51 |

Basic- Erweiterung

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|
| C000 | 20 | FD | AE | 20 | 9E | B7 | 8A | 4B | DA | C220 | BB | D0 | FA | 4C | 7F | C0 | AD | 40 | 92 |
| C008 | 20 | FD | AE | 20 | 9E | B7 | 68 | AB | 58 | C228 | 03 | C9 | 13 | D0 | 3E | AD | 3D | 03 | 4A |
| C010 | 20 | 18 | C4 | 20 | FD | AE | 20 | 9E | 7C | C230 | CD | 3F | 03 | 90 | 06 | 20 | E4 | C3 | 08 |
| C018 | AD | A5 | 0D | F0 | 12 | 20 | A6 | B6 | 08 | C238 | 4C | FF | C0 | CE | 3F | 03 | 20 | 1C | 80 |
| C020 | AA | A0 | 00 | E8 | CA | F0 | 08 | B1 | 54 | C240 | C4 | BE | 44 | 03 | 8C | 43 | 03 | AC | 17 |
| C028 | 22 | 20 | D2 | FF | C8 | D0 | F5 | A9 | 19 | C248 | 3D | 03 | 88 | C8 | B9 | 46 | 03 | 99 | D7 |
| C030 | 5B | 20 | D2 | FF | 20 | 1C | C4 | 8E | 48 | C250 | 45 | 03 | 20 | D2 | FF | CC | 3F | 03 | FF |
| C038 | 42 | 03 | 8C | 41 | 03 | A5 | 0D | D0 | 26 | C258 | D0 | F1 | A9 | 20 | 20 | D2 | FF | AE | 07 |
| C040 | 11 | 20 | 8D | AD | 20 | B8 | B1 | A6 | 45 | C260 | 44 | 03 | AC | 43 | 03 | 20 | 18 | C4 | E1 |
| C048 | 64 | D0 | 04 | A6 | 65 | D0 | 0D | 4C | 4A | C268 | 4C | FF | C0 | AD | 40 | 03 | C9 | 9D | C9 |
| C050 | 48 | B2 | 20 | FD | AE | 20 | 9E | B7 | 2F | C270 | D0 | 28 | AC | 3D | 03 | 88 | D0 | 06 | 5B |
| C058 | E0 | 00 | F0 | F3 | 8E | 3C | 03 | AD | 84 | C278 | 20 | E4 | C3 | 4C | FF | C0 | AC | 3D | 61 |
| C060 | 41 | 03 | 18 | 6D | 3C | 03 | C9 | 27 | C4 | C280 | 03 | CC | 3F | 03 | D0 | 04 | A9 | 20 | C8 |
| C068 | B0 | E5 | AB | AE | 42 | 03 | 20 | 18 | 0C | C288 | D0 | 03 | AD | 3E | 03 | 20 | D2 | FF | F1 |
| C070 | C4 | A9 | 5D | 20 | D2 | FF | 20 | FD | 4E | C290 | C6 | D3 | C6 | D3 | CE | 3D | 03 | 4C | 2E |
| C078 | AE | 20 | 9E | B7 | 8E | 45 | 03 | A9 | 18 | C298 | FF | C0 | AD | 40 | 03 | C9 | 1D | D0 | 33 |
| C080 | 00 | A0 | 32 | 99 | 45 | 03 | 88 | D0 | F4 | C2A0 | 2C | AC | 3C | 03 | CC | 3D | 03 | F0 | 5B |
| C088 | FA | 84 | C6 | C8 | 8C | 3F | 03 | 8C | 38 | C2AB | 02 | B0 | 06 | 20 | E4 | C3 | 4C | FF | A7 |
| C090 | 3D | 03 | A9 | 80 | 8D | 8A | 02 | AE | 8A | C2B0 | C0 | AC | 3D | 03 | CC | 3F | 03 | F0 | EE |
| C098 | 42 | 03 | AC | 41 | 03 | 20 | 18 | C4 | E1 | C2B8 | 0C | AD | 3E | 03 | 20 | D2 | FF | EE | FA |
| C0A0 | A9 | 03 | 85 | 99 | A0 | 00 | 20 | 3E | 13 | C2C0 | 3D | 03 | 4C | FF | C0 | A9 | 20 | 8D | 32 |
| C0AB | F1 | 99 | 46 | 03 | C8 | CC | 3C | 03 | D0 | C2C8 | 40 | 03 | 4C | F4 | C2 | AD | 40 | 03 | E8 |
| C0B0 | D0 | F4 | AE | 42 | 03 | AC | 41 | 03 | DE | C2D0 | C9 | 20 | B0 | 06 | 20 | E4 | C3 | 4C | 05 |
| C0B8 | 20 | 18 | C4 | A2 | 00 | AC | 3C | 03 | 88 | C2D8 | FF | C0 | C9 | 60 | 90 | 16 | C9 | 7B | 6F |
| C0C0 | B9 | 45 | 03 | C9 | 20 | D0 | 07 | 8A | 7C | C2E0 | 90 | F2 | C9 | 7F | 90 | 0E | C9 | C1 | 71 |
| C0C8 | 99 | 45 | 03 | 88 | D0 | F2 | C8 | 8C | E9 | C2E8 | 90 | EA | C9 | DB | 90 | 06 | 20 | E4 | 5A |
| C0D0 | 3F | 03 | AD | 45 | 03 | F0 | 28 | AE | 32 | C2F0 | C3 | 4C | FF | C0 | AC | 45 | 03 | F0 | 2F |
| C0D8 | 3C | 03 | BD | 45 | 03 | F0 | 1D | C9 | 19 | C2F8 | 20 | C9 | 2D | F0 | 1C | C9 | 2E | F0 | 27 |
| C0E0 | 20 | F0 | 19 | C9 | 2D | F0 | 15 | C9 | 0C | C300 | 18 | C9 | 45 | F0 | 14 | C9 | 20 | F0 | 21 |
| C0E8 | 2E | F0 | 11 | C9 | 45 | F0 | 0D | C9 | 22 | C308 | 10 | C9 | 30 | 90 | 09 | C9 | 39 | 90 | D2 |
| C0F0 | 30 | 90 | 06 | C9 | 39 | F0 | 05 | 90 | 9D | C310 | 08 | F0 | 06 | 20 | E4 | C3 | 4C | FF | 6D |
| C0FB | 03 | 4C | 99 | AD | CA | D0 | DB | A9 | CF | C318 | C0 | AC | 3C | 03 | C8 | CC | 3F | 03 | 85 |
| C100 | 03 | 85 | 99 | 20 | 3E | F1 | A2 | 00 | E7 | C320 | D0 | 0E | AC | 3F | 03 | CC | 3D | 03 | A0 |
| C108 | 86 | 99 | C6 | D3 | C9 | 20 | D0 | 02 | 56 | C328 | D0 | 06 | 20 | E4 | C3 | 4C | FF | C0 | BB |
| C110 | A9 | A4 | C9 | A4 | F0 | 0B | 20 | 1C | 12 | C330 | AD | 3D | 03 | CD | 3F | 03 | B0 | 12 | 80 |
| C118 | C4 | B1 | D1 | 09 | 80 | 91 | D1 | D0 | CA | C338 | AC | 3D | 03 | AD | 40 | 03 | 99 | 45 | 56 |
| C120 | 05 | 20 | D2 | FF | C6 | D3 | A5 | C6 | 8A | C340 | 03 | EE | 3D | 03 | 20 | E4 | C3 | 4C | 03 |
| C128 | F0 | FC | C6 | C6 | AD | 77 | 02 | C9 | 63 | C348 | 59 | C3 | AC | 3D | 03 | AD | 40 | 03 | 99 |
| C130 | 22 | D0 | 06 | 20 | E4 | C3 | 4C | FF | A7 | C350 | 99 | 45 | 03 | EE | 3F | 03 | EE | 3D | 56 |
| C138 | C0 | 8D | 40 | 03 | C9 | 0D | D0 | 03 | F8 | C358 | 03 | AD | 40 | 03 | 20 | D2 | FF | AD | 34 |
| C140 | 4C | 71 | C3 | AC | 3D | 03 | B9 | 45 | A0 | C360 | 3D | 03 | CD | 3C | 03 | F0 | 07 | 90 | 56 |
| C148 | 03 | 8D | 3E | 03 | AD | 40 | 03 | C9 | 58 | C368 | 05 | CE | 3D | 03 | C6 | D3 | 4C | FF | B2 |
| C150 | 94 | D0 | 5B | AC | 3C | 03 | C8 | CC | AB | C370 | C0 | AC | 3D | 03 | B9 | 45 | 03 | D0 | F5 |
| C158 | 3F | 03 | D0 | 06 | 20 | E4 | C3 | 4C | 88 | C378 | 02 | A9 | 20 | 20 | D2 | FF | A9 | 0D | C9 |
| C160 | FF | C0 | 20 | 1C | C4 | 8E | 44 | 03 | BA | C380 | 20 | D2 | FF | CE | 3F | 03 | 20 | FD | DF |
| C168 | 8C | 43 | 03 | AC | 3F | 03 | CC | 3D | 6B | C388 | AE | AD | 45 | 03 | D0 | 2D | 20 | 8B | 7A |
| C170 | 03 | F0 | 0C | 88 | B9 | 45 | 03 | 99 | 76 | C390 | B0 | 85 | 49 | 84 | 4A | 20 | A3 | B6 | 07 |
| C178 | 46 | 03 | CC | 3D | 03 | D0 | F4 | A9 | 50 | C398 | AD | 3F | 03 | F0 | 18 | 20 | 75 | B4 | 3B |
| C180 | 20 | 99 | 45 | 03 | AE | 42 | 03 | AC | 8C | C3A0 | A0 | 02 | B9 | 61 | 00 | 91 | 49 | 88 | 26 |
| C188 | 41 | 03 | 20 | 18 | C4 | A0 | 00 | B9 | B1 | C3AB | 10 | F8 | C8 | B9 | 46 | 03 | 91 | 62 | 9A |
| C190 | 46 | 03 | D0 | 02 | A9 | 20 | 20 | D2 | E7 | C3B0 | C8 | C4 | 61 | D0 | F6 | A9 | 00 | 8D | 56 |
| C198 | FF | C8 | CC | 3C | 03 | D0 | F0 | EE | FC | C3B8 | 8A | 02 | 60 | A9 | 46 | 85 | 22 | A9 | 79 |
| C1A0 | 3F | 03 | AE | 44 | 03 | AC | 43 | 03 | 3D | C3C0 | 03 | 85 | 23 | AD | 3F | 03 | AB | A2 | 37 |
| C1A8 | 20 | 18 | C4 | 4C | FF | C0 | AD | 40 | 2B | C3C8 | 00 | 86 | 0D | 20 | B5 | B7 | 20 | 8B | FB |
| C1B0 | 03 | C9 | 14 | D0 | 53 | AC | 3D | 03 | 61 | C3D0 | B0 | 85 | 49 | B4 | 4A | A5 | 0D | D0 | D3 |
| C1B8 | 88 | D0 | 06 | 20 | E4 | C3 | 4C | FF | 0C | C3D8 | 08 | A5 | 49 | 20 | D0 | BB | 4C | B5 | 39 |
| C1C0 | C0 | 20 | 1C | C4 | 88 | 8E | 44 | 03 | 33 | C3E0 | C3 | 4C | 99 | AD | A0 | 32 | A2 | 00 | 72 |
| C1C8 | 8C | 43 | 03 | AC | 3D | 03 | 88 | B9 | A9 | C3E8 | CA | D0 | FD | 88 | D0 | FA | A9 | 05 | E9 |
| C1D0 | 46 | 03 | 99 | 45 | 03 | C8 | CC | 3F | 5F | C3F0 | 8D | 18 | D4 | A9 | 15 | 8D | 05 | D4 | 59 |
| C1D8 | 03 | D0 | F4 | AE | 42 | 03 | AC | 41 | 23 | C3F8 | A9 | A0 | 8D | 06 | D4 | A9 | 46 | 8D | 73 |
| C1E0 | 03 | 20 | 18 | C4 | A0 | 00 | B9 | 46 | 4A | C400 | 01 | D4 | A9 | 11 | 8D | 04 | D4 | A0 | 81 |
| C1E8 | 03 | D0 | 02 | A9 | 20 | 20 | D2 | FF | 60 | C408 | 1E | A2 | 00 | CA | D0 | FD | 88 | D0 | 3E |
| C1F0 | C8 | CC | 3C | 03 | D0 | F0 | AE | 44 | 7F | C410 | FA | 8C | 04 | D4 | 8C | 18 | D4 | 60 | 86 |
| C1FB | 03 | AC | 43 | 03 | 20 | 18 | C4 | CE | 95 | C418 | 18 | 4C | F0 | FF | 38 | 4C | F0 | FF | 9A |
| C200 | 3F | 03 | CE | 3D | 03 | 4C | FF | C0 | C2 | C420 | A9 | 38 | 8D | 08 | 03 | A9 | C4 | 8D | 86 |
| C208 | AD | 40 | 03 | C9 | 93 | D0 | 17 | AE | D4 | C428 | 09 | 03 | 60 | A9 | E4 | 8D | 08 | 03 | 19 |
| C210 | 42 | 03 | AC | 41 | 03 | 20 | 18 | C4 | E1 | C430 | A9 | A7 | 8D | 09 | 03 | 60 | 00 | 00 | 29 |
| C218 | AC | 3C | 03 | A9 | 20 | 20 | D2 | FF | 9D | C438 | 20 | 73 | 00 | F0 | 04 | C9 | 5F | F0 | 69 |
| | | | | | | | | | | C440 | 03 | 4C | E7 | A7 | 20 | 73 | 00 | C9 | DC |
| | | | | | | | | | | C448 | 49 | F0 | 03 | 4C | 08 | AF | 20 | 73 | 18 |
| | | | | | | | | | | C450 | 00 | 20 | 03 | C0 | 4C | AE | A7 | FF | 6A |

Basic- Erweiterung

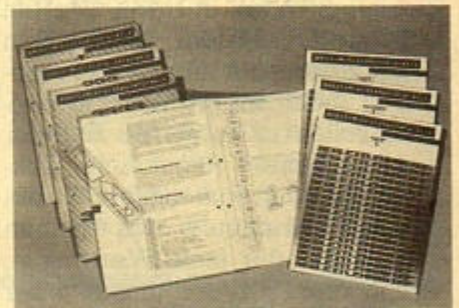
| | | |
|-----|---|------|
| 103 | DATA 32,253,174,32,158,183,138,72,32,253, 174,32,158,183,104,168,32,24,196 | 6929 |
| 104 | DATA 32,253,174,32,158,173,165,13,240,18, 32,166,182,170,160,0,232,202,240,8 | 7196 |
| 105 | DATA 177,34,32,210,255,200,208,245,169,91, 32,210,255,32,28,196,142,66,3,140 | 6699 |
| 106 | DATA 65,3,165,13,208,17,32,141,173,32,184, 177,166,100,208,4,166,101,208,13 | 6967 |
| 107 | DATA 76,72,178,32,253,174,32,158,183,224,0, 240,243,142,60,3,173,65,3,24,109 | 7605 |
| 108 | DATA 60,3,201,39,176,229,168,174,66,3,32, 24,196,169,93,32,210,255,32,253 | 7893 |
| 109 | DATA 174,32,158,183,142,69,3,169,0,160,50, 153,69,3,136,208,250,132,198,200 | 7117 |
| 110 | DATA 140,63,3,140,61,3,169,128,141,138,2, 174,66,3,172,65,3,32,24,196,169,3 | 8113 |
| 111 | DATA 133,153,160,0,32,62,241,153,70,3,200, 204,60,3,208,244,174,66,3,172,65 | 7913 |
| 112 | DATA 3,32,24,196,162,0,172,60,3,185,69,3, 201,32,208,7,138,153,69,3,136,208 | 7015 |
| 113 | DATA 242,200,140,63,3,173,69,3,240,40,174, 60,3,189,69,3,240,29,201,32,240 | 6949 |
| 114 | DATA 25,201,45,240,21,201,46,240,17,201,69, 240,13,201,48,144,6,201,57,240,5 | 6492 |
| 115 | DATA 144,3,76,153,173,202,208,219,169,3, 133,153,32,62,241,162,0,134,153,198 | 6919 |
| 116 | DATA 211,201,32,208,2,169,164,201,164,240, 11,32,28,196,177,209,9,128,145 | 6960 |
| 117 | DATA 209,208,5,32,210,255,198,211,165,198, 240,252,198,198,173,119,2,201,34 | 8003 |
| 118 | DATA 208,6,32,228,195,76,255,192,141,64,3, 201,13,208,3,76,113,195,172,61,3 | 8428 |
| 119 | DATA 185,69,3,141,62,3,173,64,3,201,148, 208,91,172,60,3,200,204,63,3,208,6 | 6713 |
| 120 | DATA 32,228,195,76,255,192,32,28,196,142, 68,3,140,67,3,172,63,3,204,61,3 | 6743 |
| 121 | DATA 240,12,136,185,69,3,153,70,3,204,61,3, 208,244,169,32,153,69,3,174,66,3 | 7829 |
| 122 | DATA 172,65,3,32,24,196,160,0,185,70,3,208, 2,169,32,32,210,255,200,204,60,3 | 7002 |
| 123 | DATA 208,240,238,63,3,174,68,3,172,67,3,32, 24,196,76,255,192,173,64,3,201 | 7538 |
| 124 | DATA 20,208,83,172,61,3,136,208,6,32,228, 195,76,255,192,32,28,196,136,142 | 6862 |
| 125 | DATA 68,3,140,67,3,172,61,3,136,185,70,3, 153,69,3,200,204,63,3,208,244,174 | 7274 |
| 126 | DATA 66,3,172,65,3,32,24,196,160,0,185,70, 3,208,2,169,32,32,210,255,200,204 | 7437 |
| 127 | DATA 60,3,208,240,174,68,3,172,67,3,32,24, 196,206,63,3,206,61,3,76,255,192 | 7821 |
| 128 | DATA 173,64,3,201,147,208,23,174,66,3,172, 65,3,32,24,196,172,60,3,169,32,32 | 8317 |
| 129 | DATA 210,255,136,208,250,76,127,192,173,64, 3,201,19,208,62,173,61,3,205,63 | 7641 |
| 130 | DATA 3,144,6,32,228,195,76,255,192,206,63, 3,32,28,196,142,68,3,140,67,3,172 | 7041 |
| 131 | DATA 61,3,136,200,185,70,3,153,69,3,32,210, 255,204,63,3,208,241,169,32,32 | 6603 |
| 132 | DATA 210,255,174,68,3,172,67,3,32,24,196, | 8277 |

| | |
|---|------|
| 76,255,192,173,64,3,201,157,208,40 | |
| 133 DATA 172,61,3,136,208,6,32,228,195,76,255,192,172,61,3,204,63,3,208,4,169 | 7294 |
| 134 DATA 32,208,3,173,62,3,32,210,255,198,211,198,211,206,61,3,76,255,192,173 | 7513 |
| 135 DATA 64,3,201,29,208,44,172,60,3,204,61,3,240,2,176,6,32,228,195,76,255,192 | 7731 |
| 136 DATA 172,61,3,204,63,3,240,12,173,62,3,32,210,255,238,61,3,76,255,192,169 | 8143 |
| 137 DATA 32,141,64,3,76,244,194,173,64,3,201,32,176,6,32,228,195,76,255,192,201 | 7473 |
| 138 DATA 96,144,22,201,123,144,242,201,127,144,14,201,193,144,234,201,219,144,6 | 7235 |
| 139 DATA 32,228,195,76,255,192,172,69,3,240,32,201,45,240,28,201,46,240,24,201 | 6960 |
| 140 DATA 69,240,20,201,32,240,16,201,48,144,9,201,57,144,8,240,6,32,228,195,76 | 6678 |
| 141 DATA 255,192,172,60,3,200,204,63,3,208,14,172,63,3,204,61,3,208,6,32,228 | 7580 |
| 142 DATA 195,76,255,192,173,61,3,205,63,3,176,18,172,61,3,173,64,3,153,69,3,238 | 7785 |
| 143 DATA 61,3,32,228,195,76,89,195,172,61,3,173,64,3,153,69,3,238,63,3,238,61,3 | 7978 |
| 144 DATA 173,64,3,32,210,255,173,61,3,205,60,3,240,7,144,5,206,61,3,198,211,76 | 7383 |
| 145 DATA 255,192,172,61,3,185,69,3,208,2,169,32,32,210,255,169,13,32,210,255 | 7198 |
| 146 DATA 206,63,3,32,253,174,173,69,3,208,45,32,139,176,133,73,132,74,32,163 | 8289 |
| 147 DATA 182,173,63,3,240,24,32,117,180,160,2,185,97,0,145,73,136,16,248,200 | 7322 |
| 148 DATA 185,70,3,145,98,200,196,97,208,246,169,0,141,138,2,96,169,70,133,34 | 7257 |
| 149 DATA 169,3,133,35,173,63,3,168,162,0,134,13,32,181,183,32,139,176,133,73 | 7828 |
| 150 DATA 132,74,165,13,208,8,165,73,32,208,187,76,181,195,76,153,173,160,50,162 | 7951 |
| 151 DATA 0,202,208,253,136,208,250,169,5,141,24,212,169,21,141,5,212,169,160 | 6557 |
| 152 DATA 141,6,212,169,70,141,1,212,169,17,141,4,212,160,30,162,0,202,208,253 | 6944 |
| 153 DATA 136,208,250,140,4,212,140,24,212,96,24,76,240,255,56,76,240,255,169,56 | 7789 |
| 154 DATA 141,8,3,169,196,141,9,3,96,169,228,141,8,3,169,167,141,9,3,96,0,0,32 | 7046 |
| 155 DATA 115,0,240,4,201,95,240,3,76,231,167,32,115,0,201,73,240,3,76,8,175,32 | 7501 |
| 156 DATA 115,0,32,3,192,76,174,167,255 | 3239 |
| 180 A=49152:E=50263 | 1877 |
| 190 S=0 | 376 |
| 200 FOR I=A TO E | 729 |
| 210 READ W | 309 |
| 220 POKE I,W | 659 |
| 230 S=S+W | 463 |
| 240 NEXT I | 276 |
| 250 IF S<>128964 THEN PRINT "FEHLER IN DATAS!!!":END | 3055 |
| 260 END | 128 |

(Jürgen und Roland Hank)

**Praxis-
orientiertes
Programmieren
BASIC +
Mikrocomputer-
praxis**

Eigene Programme in BASIC erstellen, fremde Programme verstehen und umschreiben, gekaufte Standardprogramme beurteilen und einsetzen, Aufbau und Funktionsweise eines Mikrocomputers kennen, die wichtigsten Anwendungen beherrschen und über ein solides theoretisch-praktisches EDV-Grundwissen verfügen. Das alles vermittelt Ihnen der Lehrgang BASIC + Mikrocomputerpraxis.



Hauptfächer: Programmieren in BASIC – Programmierertechniken – Hardware.
Anwendungsfächer: Programmierstudien – Kaufm.-kommerzielle Anwendungen – Technisch-wissenschaftl. Anwendungen – Computer-Graphik – Musik und Spiele.
Nebenfächer: Betriebssysteme – Programmiersprachen – Daten und ihre Darstellung – Evaluation von Mikrocomputern – Mathematische Grundlagen. Die Funktion und Arbeitsweise von **Druckern** wird ausführlich besprochen!
Der Lehrgang umfaßt 14 Lehrbriefe zu je DM 128,- inklusive Begleitlehrbriefe zum C 64.

**Christiani
Fortbildung**

Technisches Lehrinstitut
Postfach 3503167 · 7750 Konstanz
in Österreich: Ferntechnikum 6901 Bregenz

Coupon auf Postkarte aufkleben oder im Umschlag einsenden.
Sie erhalten sofort **kostenlos** ausführliches Informationsmaterial über den Lehrgang BASIC + Mikrocomputerpraxis.

Name, Vorname _____
Straße, Nr. _____
PLZ, Ort _____ 03167

Lokale Variablen für den C 64

Es gibt Programmiersprachen, die bekannteste ist PASCAL, die in Prozeduren und Funktionen lokale Variablen zur Verfügung stellen, die nur in der gerade ausgeführten Prozedur oder den wiederum von ihnen aufgerufenen Prozeduren gültig und nach ihrer Ausführung nicht mehr verfügbar sind. So können die gleichen Variablennamen in diesen Prozeduren und im Hauptprogramm verwendet werden, ohne daß die Variablen des Hauptprogramms beeinflusst oder in ihren Werten verändert werden. Das allein reicht aber nicht aus, denn Prozeduren (oder in BASIC: Subroutinen) sollen ja universell und für verschiedene Argumente anwendbar sein. Aus diesem Grund müssen Parameter, das können Variablen oder feste Werte sein, mit in die Prozedur hinein- oder aus ihr herausgenommen werden. So kann in der Prozedur mit variablen Werten gerechnet werden, wobei jedoch die verwendeten Variablen nur lokale Bedeutung haben. Genauso können die Ergebnisse der Subroutine wieder

Viele Programmiersprachen haben in Subroutinen eigene Variablen. Damit es sie auch in Basic gibt, hat Dirk Sagwitz das Maschinenprogramm „PROC“ geschrieben.

in das aufrufende Programm übernommen werden. Man nennt sie Argument- beziehungsweise Argument-Ergebnis-Parameter, je nachdem, ob sie nur als Argumente zur Berechnung dienen oder auch Ergebnisse liefern sollen. Es gibt jedoch keine reinen Ergebnisparameter. Mit dem Programm 'PROC' kann das auch in Basic simuliert werden. Es ist vollständig in Maschinensprache geschrieben und

kann direkt oder über einen Basic-Loader eingeladen werden. Nach der Initialisierung stehen zwei neue Befehle (!PROC und !EPROC) zur Verfügung. Mit dem Befehl !PROC wird auf lokale Variablen 'umgeschaltet' und mit dem Befehl !EPROC wieder zurück. Das bedeutet, daß alle Variablen, die vor Ausführung des !PROC-Befehls vorhanden waren, wieder ihre vorherigen Werte besitzen und Variablen, die erst nach dem !PROC-Befehl benutzt wurden, unbekannt sind und keinen definierten Wert besitzen (Beispiel 1). Es ist aber auch möglich, daß Argumente mit in die Subroutine hinein- oder Ergebnisse aus ihr herausgenommen werden. Der Aufruf erfolgt dann zum Beispiel mit !PROC(A,B%,C1\$).

Aufbau von Rekursion

Das bedeutet, daß die Variablen A,B% und C1\$ ihre Werte behalten und auch nach dem !PROC-Befehl bekannt sind, während alle anderen Variablen undefiniert

```

0 DATA 162,0,142,141,193,162,16,160,192, 6776
  142,0,3,140,9,3,96,32,115,0,8,201
1 DATA 33,240,4,40,76,231,167,160,0,40, 7221
  32,115,0,217,131,193,240,11,217,135
2 DATA 193,240,3,76,0,175,76,222,192,32, 6620
  115,0,200,217,131,193,200,241,192
3 DATA 3,200,243,32,115,0,201,40,240,6, 6040
  32,113,192,76,174,167,165,122,72
4 DATA 165,123,72,32,195,192,104,133, 7154
  123,104,133,122,72,165,123,72,32,113
5 DATA 192,32,195,192,104,133,123,104, 6571
  133,122,32,4,193,76,174,167,174,141
6 DATA 193,224,254,144,3,76,105,193,165, 7672
  45,157,142,193,165,47,157,142,194
7 DATA 165,49,157,142,195,232,165,46, 6794
  157,142,193,165,48,157,142,194,165
8 DATA 50,157,142,195,165,49,133,45,133, 6540
  47,165,50,133,46,133,40,232,142
9 DATA 141,193,96,32,231,176,133,73,132, 7293
  74,36,13,48,7,36,14,16,6,76,196
10 DATA 169,76,44,170,76,203,187,32,115, 7113
  0,32,40,175,32,171,192,169,44,160
11 DATA 0,209,122,240,239,169,41,203,122, 6857
  240,3,76,8,175,96,32,115,0,200,217
12 DATA 135,193,200,243,192,4,200,243,32, 6979
  115,0,201,40,240,6,32,61,193,76
13 DATA 174,167,32,31,193,32,4,193,32,31, 6900
  193,76,242,192,32,115,0,32,31,193
14 DATA 32,40,175,32,31,193,32,171,192, 7187
  160,0,177,122,201,44,240,233,32,115
15 DATA 0,96,174,141,193,202,165,46,160, 7525
  189,142,193,133,46,152,157,142,193
16 DATA 202,165,45,160,189,142,193,133, 6911
  45,152,157,142,193,36,174,141,193
17 DATA 200,3,76,105,193,202,189,142,193, 7252
  133,46,189,142,194,133,48,189,142
18 DATA 195,133,50,202,189,142,193,133, 6979
  45,189,142,194,133,47,189,142,195
19 DATA 133,43,142,141,193,96,162,0,189, 6400
  121,193,32,210,255,232,224,10,200
20 DATA 245,76,93,164,63,76,79,87,32,83, 7165
  84,65,67,75,80,92,73,67,69,80,82
21 DATA 79,67,232,0 1307
22 FOR I=49152 TO 49549:READ A:S=S+A:POKE 3413
  I,A:NEXT
23 IF S<>48542 THEN PRINT 2895
  *PRUEFSUMMENFEHLER !*:END
24 SYS 49152:PRINT "PROC AKTIVIERT!":NEW 2820

```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|--|
| C000 | A2 | 00 | 8E | 8D | C1 | A2 | 10 | A0 | 32 | C0C0 | 4C | D0 | BB | 20 | 73 | 00 | 20 | 28 | 82 | |
| C008 | C0 | 8E | 08 | 03 | 8C | 09 | 03 | 60 | 5D | C0C8 | AF | 20 | AB | C0 | A9 | 2C | A0 | 00 | 96 | |
| C010 | 20 | 73 | 00 | 08 | C9 | 21 | F0 | 04 | 38 | C0D0 | D1 | 7A | F0 | EF | A9 | 29 | D1 | 7A | 2E | |
| C018 | 28 | 4C | E7 | A7 | A0 | 00 | 28 | 20 | C4 | C0D8 | F0 | 03 | 4C | 08 | AF | 60 | 20 | 73 | 2C | |
| C020 | 73 | 00 | D9 | 83 | C1 | F0 | 0B | D9 | CC | C0E0 | 00 | C8 | D9 | 87 | C1 | D0 | F3 | C0 | AE | |
| C028 | 87 | C1 | F0 | 03 | 4C | 08 | AF | 4C | 59 | C0E8 | 04 | D0 | F3 | 20 | 73 | 00 | C9 | 28 | 1A | |
| C030 | DE | C0 | 20 | 73 | 00 | C8 | D9 | 83 | 5A | C0F0 | F0 | 06 | 20 | 3D | C1 | 4C | AE | A7 | 48 | |
| C038 | C1 | D0 | F1 | C0 | 03 | D0 | F3 | 20 | 28 | C0F8 | 20 | 1F | C1 | 20 | 04 | C1 | 20 | 1F | E6 | |
| C040 | 73 | 00 | C9 | 28 | F0 | 06 | 20 | 71 | AB | C100 | C1 | 4C | F2 | C0 | 20 | 73 | 00 | 20 | 34 | |
| C048 | C0 | 4C | AE | A7 | A5 | 7A | 48 | A5 | 49 | C108 | 1F | C1 | 20 | 28 | AF | 20 | 1F | C1 | 43 | |
| C050 | 7B | 48 | 20 | C3 | C0 | 68 | 85 | 7B | F2 | C110 | 20 | AB | C0 | A0 | 00 | B1 | 7A | C9 | 96 | |
| C058 | 68 | 85 | 7A | 48 | A5 | 7B | 48 | 20 | 67 | C118 | 2C | F0 | E9 | 20 | 73 | 00 | 60 | AE | 2A | |
| C060 | 71 | C0 | 20 | C3 | C0 | 68 | 85 | 7B | 71 | C120 | 8D | C1 | CA | A5 | 2E | A8 | BD | 8E | A6 | |
| C068 | 68 | 85 | 7A | 20 | 04 | C1 | 4C | AE | 1F | C128 | C1 | 85 | 2E | 98 | 9D | 8E | C1 | CA | D8 | |
| C070 | A7 | AE | 8D | C1 | E0 | FE | 90 | 03 | 34 | C130 | A5 | 2D | A8 | BD | 8E | C1 | 85 | 2D | 88 | |
| C078 | 4C | 69 | C1 | A5 | 2D | 9D | 8E | C1 | 5C | C138 | 98 | 9D | 8E | C1 | 60 | AE | 8D | C1 | 46 | |
| C080 | A5 | 2F | 9D | 8E | C2 | A5 | 31 | 9D | 36 | C140 | D0 | 03 | 4C | 69 | C1 | CA | BD | 8E | D5 | |
| C088 | 8E | C3 | E8 | A5 | 2E | 9D | 8E | C1 | 6C | C148 | C1 | 85 | 2E | BD | 8E | C2 | 85 | 30 | CF | |
| C090 | A5 | 30 | 9D | 8E | C2 | A5 | 32 | 9D | 36 | C150 | BD | 8E | C3 | 85 | 32 | CA | BD | 8E | 04 | |
| C098 | 8E | C3 | A5 | 31 | 85 | 2D | 85 | 2F | EC | C158 | C1 | 85 | 2D | BD | 8E | C2 | 85 | 2F | CF | |
| C0A0 | A5 | 32 | 85 | 2E | 85 | 30 | E8 | 8E | 77 | C160 | BD | 8E | C3 | 85 | 31 | 8E | 8D | C1 | DC | |
| C0A8 | 8D | C1 | 60 | 20 | E7 | B0 | 85 | 49 | 7F | C168 | 60 | A2 | 00 | BD | 79 | C1 | 20 | D2 | 09 | |
| C0B0 | 84 | 4A | 24 | 0D | 30 | 07 | 24 | 0E | 90 | C170 | FF | E8 | E0 | 0A | D0 | F5 | 4C | 62 | B1 | |
| C0B8 | 10 | 06 | 4C | C4 | A9 | 4C | 2C | AA | 71 | C178 | A4 | 3F | 4C | 4F | 57 | 20 | 53 | 54 | 98 | |
| | | | | | | | | | | C180 | 41 | 43 | 4B | 50 | 52 | 4F | 43 | 45 | FA | |
| | | | | | | | | | | C188 | 50 | 52 | 4F | 43 | E8 | 00 | 0D | 0D | F2 | |

| ***** | | | | | | | | | | ***** | | | | | | | | | | | | |
|---------------------------------------|---|-----------|--|--|--|--|--|--|--|-------------------------------|---|-------|--|--|--|--|--|--|--|------------------------------|------|----|
| ;# | | | | | | | | | | PLP | | | | | | | | | | | | |
| ;# Prozeduren mit lokalen Variablen * | | | | | | | | | | JMP nploop | | | | | | | | | | | | |
| ;# und Parameteruebergabe fuer C-64 * | | | | | | | | | | LDY #800 ; '!' ueberlesen | | | | | | | | | | | | |
| ;# Vers. 2.5 (c) 30.08.1984 * | | | | | | | | | | PLP | | | | | | | | | | | | |
| ***** | | | | | | | | | | JSR chrget ;naechstes Zeichen | | | | | | | | | | | | |
| ; | | | | | | | | | | CMP itext,y ;='P' | | | | | | | | | | | | |
| iflg | = | #0E | | | | | | | | iflg | = | #0D | | | | | | | | BEQ t2 ;ja => | C01E | 28 |
| tflg | = | #0D | | | | | | | | chrget | = | #0B73 | | | | | | | | CMP t3 ;ja => | C027 | 09 |
| varstpl | = | #2D | | | | | | | | nploop | = | #A7E7 | | | | | | | | JSR incst ;Keine Parameter | C028 | 09 |
| arrstpl | = | #2F | | | | | | | | iploop | = | #A7Ae | | | | | | | | JMP iploop ;fertig | C02F | 4C |
| arrndpl | = | #31 | | | | | | | | uzint | = | #A9C4 | | | | | | | | LDI baspl ;merke Pointer | C032 | 20 |
| vname1 | = | #49 | | | | | | | | wzrea | = | #8B08 | | | | | | | | PHA | C035 | 08 |
| baspl | = | #7A | | | | | | | | wzstr | = | #AA2C | | | | | | | | JSR init ;initialisiere | C036 | 09 |
| varstph | = | varstpl+1 | | | | | | | | synterr | = | #AF08 | | | | | | | | PLA ;Variablen | C038 | 08 |
| arrstph | = | arrstpl+1 | | | | | | | | varhol | = | #AF28 | | | | | | | | STA basph | C039 | 08 |
| arrndph | = | arrndpl+1 | | | | | | | | vsuch | = | #8BE7 | | | | | | | | PLA | C03B | 08 |
| vnameh | = | vname1+1 | | | | | | | | basout | = | #FFD2 | | | | | | | | STA baspl | C03D | 08 |
| basph | = | baspl+1 | | | | | | | | | | | | | | | | | | PHA | C03E | 48 |
| chrget | = | #0B73 | | | | | | | | | | | | | | | | | | LDI basph | C043 | F0 |
| iloopl | = | #0308 | | | | | | | | | | | | | | | | | | PHA | C044 | F0 |
| ilooph | = | iloopl+1 | | | | | | | | | | | | | | | | | | LDI baspl | C046 | 20 |
| erexit | = | #A4E2 | | | | | | | | | | | | | | | | | | PHA | C049 | 4C |
| nploop | = | #A7E7 | | | | | | | | | | | | | | | | | | LDI basph | C04C | A5 |
| iploop | = | #A7Ae | | | | | | | | | | | | | | | | | | PHA | C04E | 48 |
| uzint | = | #A9C4 | | | | | | | | | | | | | | | | | | LDI basph | C04F | A5 |
| wzrea | = | #8B08 | | | | | | | | | | | | | | | | | | PHA | C051 | 48 |
| wzstr | = | #AA2C | | | | | | | | | | | | | | | | | | JSR incst ;initialisiere | C052 | 20 |
| synterr | = | #AF08 | | | | | | | | | | | | | | | | | | PLA ;Variablen | C053 | 08 |
| varhol | = | #AF28 | | | | | | | | | | | | | | | | | | STA basph | C055 | 68 |
| vsuch | = | #8BE7 | | | | | | | | | | | | | | | | | | PLA | C058 | 68 |
| basout | = | #FFD2 | | | | | | | | | | | | | | | | | | STA baspl | C059 | 85 |
| | | | | | | | | | | | | | | | | | | | | PHA | C05B | 48 |
| | | | | | | | | | | | | | | | | | | | | LDI basph | C05C | A5 |
| | | | | | | | | | | | | | | | | | | | | PHA | C05E | 48 |
| | | | | | | | | | | | | | | | | | | | | JSR incst ;initialisiere | C05F | 20 |
| | | | | | | | | | | | | | | | | | | | | PLA ;Variablen auf | C062 | 20 |
| | | | | | | | | | | | | | | | | | | | | PLA ;lokaler Ebene | C065 | 68 |
| | | | | | | | | | | | | | | | | | | | | STA basph | C066 | 85 |
| | | | | | | | | | | | | | | | | | | | | PLA | C068 | 68 |
| | | | | | | | | | | | | | | | | | | | | STA baspl | C069 | 85 |
| | | | | | | | | | | | | | | | | | | | | JSR copy ;kopiere Variablen | C06B | 20 |
| | | | | | | | | | | | | | | | | | | | | JMP iploop ;auf lokale Ebene | C06E | 4C |
| | | | | | | | | | | | | | | | | | | | | LDX stackp ;reicht Stack ? | C071 | AE |
| | | | | | | | | | | | | | | | | | | | | CPX #8FE | C074 | E0 |
| | | | | | | | | | | | | | | | | | | | | BCC ioK ;ja => | C076 | 90 |

| | | | | | | |
|--------|---------------|---------------------|---------------|---------------|---------------------|---------------------|
| 10K | JMP error | !LOW STACK Error | C078 4C 69 C1 | JSR varhol | !Pointer | C10A 20 28 AF |
| | LDA varstpl | !momentane Pointer | C078 A5 20 | JSR verst | !und Kopiere die | C180 20 1F C1 |
| | STA vptab,x | !auf Stack | C070 90 8E C1 | JSR cpl | !Variablen | C118 20 AB C0 |
| | LDA arrstpl | !merke aktuelle | C088 A5 2F | LDY #800 | | C113 A0 80 |
| | STA asptab,x | !Pointer der | C082 90 8E C2 | LDA (baspl),y | | C115 81 7A |
| | LDA arrdedpl | !Variabliantabelle | C085 A5 31 | CMP #' | !Ende ? | C117 C9 2C |
| | STA aeptab,x | | C087 9D 8E C3 | BEG copy | !nein => | C119 F0 E9 |
| | INX | | C08A E8 | JSR chrget | !'')' ueberlesen | C11B 28 73 80 |
| | LDA varstph | | C088 A5 2E | RTS | !Ende | C11E 60 |
| | STA vptab,x | | C090 9D 8E C1 | LDX stackp | !vertausche | C11F AE 8D C1 |
| | LDA arrstph | | C098 A5 38 | DEX | !Pointer zum Suchen | C122 CA |
| | STA asptab,x | | C092 9D 8E C2 | LDA varstph | !und Kopieren | C123 A5 2E |
| | LDA arrdedph | | C095 A5 32 | TAY | | C125 A8 |
| | STA aeptab,x | | C097 9D 8E C3 | LDA vptab,x | | C126 8D 8E C1 |
| | LDA arrdedpl | !Pointer neu | C09A A5 31 | STA varstph | | C129 85 2E |
| | STA varstpl | !setzen | C09C 95 2D | TYA | | C129 98 |
| | STA arrstpl | | C09E 85 2F | STA vptab,x | | C12C 9D 8E C1 |
| | LDA arrdedph | | C0A8 A5 32 | DEX | | C12F CA |
| | STA varstph | | C0A2 85 2E | LDA varstpl | | C138 A5 2D |
| | STA arrstph | | C0A4 85 38 | TAY | | C132 A8 |
| | INX | | C0A6 E8 | LDA vptab,x | | C133 8D 8E C1 |
| | STX stackp | | C0A7 8E 8D C1 | STA varstpl | | C136 85 2D |
| | RTS | !Ende | C0AA 68 | TYA | | C138 98 |
| cpl | JSR vsuch | !suche Variable | C0A8 28 E7 80 | STA vptab,x | | C139 9D 8E C1 |
| | STA vname1 | !merke Namen | C0AE 85 49 | RTS | | C13C 68 |
| | STY vnameh | | C0B8 84 4A | decst | LDX stackp | !Stackpointer >8 ? |
| | BIT tfig | !Numerisch ? | C0B2 24 8D | | BNE dl | !Ja => |
| | BMI ins | | C0B4 38 87 | JMP error | !LOW STACK ERROR | C142 4C 69 C1 |
| | BIT ifig | !String ? | C0B6 24 8E | dl | DEX | C145 CA |
| | BPL inr | | C0B8 18 86 | LDA vptab,x | !setze Pointer | C146 8D 8E C1 |
| | JMP wzint | !Integerzuweisung | C0BA 4C C4 A9 | STA varstph | !zurueck | C149 85 2E |
| ins | JMP wzstr | !Stringzuweisung | C0B0 4C 2C AA | LDA asptab,x | | C148 8D 8E C2 |
| inr | JMP wzrea | !Realzuweisung | C0C8 4C D8 BB | STA arrstph | | C14E 85 38 |
| init | JSR chrget | !ueberlese Zeichen | C0C3 28 73 80 | LDA aeptab,x | | C158 8D 8E C3 |
| | JSR varhol | !hole Variable | C0C6 28 28 AF | STA arrdedph | | C153 85 32 |
| | JSR cpl | !initialisiere | C0C9 28 AB C8 | DEX | | C155 CA |
| | LDA #' | | C0CC A9 2C | LDA vptab,x | | C156 8D 8E C1 |
| | LDY #800 | !weitere | C0CE A8 88 | STA varstpl | | C159 85 2D |
| | CMP (baspl),y | !Parameter ? | C0D8 D1 7A | LDA asptab,x | | C158 8D 8E C2 |
| | BEG init | !Ja => | C0D2 F8 EF | STA arrstpl | | C15E 85 2F |
| | LDA #' | | C0D4 A9 29 | LDA aeptab,x | | C160 8D 8E C3 |
| | CMP (baspl),y | | C0D6 D1 7A | STA arrdedpl | | C163 85 31 |
| | BEG ie | !Ende ? | C0D8 F8 83 | STX stackp | | C165 8E 8D C1 |
| gerror | JMP synterr | !SYNTAX ERROR | C0DA 4C 88 AF | RTS | !Ende | C168 60 |
| ie | RTS | | C0DD 68 | error | LDX #800 | C169 A2 80 |
| gtext | JSR chrget | !Test auf 'EPROC' | C0DE 28 73 80 | e1 | LDA tttab,x | !Fehlermeldung |
| | INX | | C0E1 C8 | JSR basout | !ausgeben | C188 8D 79 C1 |
| | CMP gtext,y | !naechstes Zeichen | C0E2 D9 87 C1 | INX | | C18E 28 D2 FF |
| | BNE gerror | !testen | C0E5 D8 F3 | CPX #80A | | C171 E8 |
| | CPY #804 | | C0E7 C8 84 | BNE e1 | | C172 E8 8A |
| | BNE gtext | !Ende ? | C0E9 D8 F3 | JMP arexit | !Sprung in Fehler- | C174 08 F5 |
| | JSR chrget | | C0EB 28 73 80 | | !routine | C178 4C 62 A4 |
| | CMP #' | !Parameter ? | C0EE C9 28 | tttab | .text 'LOW STACK' | C179 3F 4C 4F 57 28 |
| | BEG g1 | !Ja => | C0F0 F8 86 | ltext | .text 'PROC' | C17E 53 54 41 43 48 |
| g2 | JSR decst | !vermindere Stack | C0F2 28 3D C1 | gtext | .text 'EPROC' | C183 58 52 4F 43 |
| | JMP iploop | !Ende | C0F5 4C AE A7 | stackp | = #1 | C187 45 58 52 4F 43 |
| g1 | JSR verst | | C0F8 28 1F C1 | vptab | = #2 | |
| | JSR copy | !kopiere Variablen | C0FB 28 84 C1 | asptab | = #258 | |
| | JSR verst | | C0FE 28 1F C1 | aeptab | = #514 | |
| | JMP g2 | !vermindere Stack | C101 4C F2 C8 | | | |
| copy | JSR chrget | !Zeichen ueberlesen | C104 28 73 80 | | | |
| | JSR verst | !vertausche | C107 28 1F C1 | | | |
| | | | | | !Ende des Programms | |

sind. Diese Variablen können nun ganz normal benutzt werden. Trifft der Rechner nun auf den !EPROC-Befehl, entscheidet sich, welche Werte die Variablen im weiteren Verlauf besitzen. Sind hinter dem Befehl keine Parameter angegeben, so besitzen sie wieder die Werte, die sie vor Ausführung des PROC-Befehls hatten. Andernfalls werden die Werte der Variablen übernommen, die als Parameter hinter dem EPROC-Befehl angegeben sind, und alle anderen erhalten ihren alten Wert zurück (Beispiel 2). Es ist auch möglich, Rekursionen aufzubauen. Rekursive Algorithmen rufen sich selber wieder auf oder sind durch sich

selbst definiert, zum Beispiel die natürlichen Zahlen oder die Fakultät einer natürlichen Zahl (Beispiel 3). Im Standard-BASIC ist das nicht möglich, da die Variablen von allen 'Ebenen' eines rekursiven Aufrufes verfügbar sind und sich somit gegenseitig beeinflussen. Will man diese rekursiven Aufrufe in Basic simulieren, so muß man die Lokalität der Variablen durch Arrays, in denen die momentanen Werte abgelegt werden, simulieren. Mit den neuen Befehlen !PROC und !EPROC ist es möglich, bis zu 127 Ebenen zu benutzen, bevor ein !EPROC-Befehl folgen muß. Das wird wohl keine Einschränkung bedeuten,

denn ein GOSUB-Befehl, bei dem der !PROC-Befehl am sinnvollsten eingesetzt werden kann, erlaubt einen maximalen Rekursionsaufruf von 26 offenen Routinen. Hat man dennoch die Grenze von 127 einmal erreicht oder wird versucht, den !EPROC-Befehl öfter als den !PROC-Befehl anzuwenden, so meldet sich der Computer mit einem '?LOW STACK ERROR'. Der Programmierer hat dafür zu sorgen, daß die durchlaufenen !PROC- und !EPROC-Befehle gleich groß sind wie bei GOSUB-RETURN, damit nach Beendigung des Programms der Stack ausgeglichen ist. Sonst muß mit SYS49152 der Stack zurückgesetzt

werden. Es ist jedoch zu beachten, daß keine Arrays, Array-Elemente oder Funktionen kopiert werden können. Alle anderen Variablen können aber als Parameter angegeben werden.

Funktionsweise des Programms

Um die beiden Befehle !PROC und !EPROC in das Basic des C64 zu integrieren, wird zunächst der Pointer, der auf die Interpreter-schleife zeigt, verändert. In einer eigenen Routine wird dann überprüft, ob das laufende Zeichen '=' ist. Ist das der Fall, so wird in die eigentliche Decodieroutine gesprungen. Andernfalls wird die normale Interpreterschleife weiter ausgeführt. In der eigenen Decodieroutine wird dann auf !PROC oder !EPROC getestet und in die entsprechende Routine verzweigt. Wird keines der beiden Wörter gefunden, so wird in die 'SYNTAX ERROR'-Routine gesprungen. In der PROC- beziehungsweise EPROC-Routine wird als nächstes überprüft, ob Parameter angegeben sind.

Ist das der Fall, so werden die Variablen auf der globalen und lokalen Ebene initialisiert und kopiert. Das ist dadurch möglich, daß der Basic-Interpreter erst an der Stelle mit der Suche beginnt, auf die die Pointer in der Zero-Page zeigen, und dann aufhört, wenn die Variable gefunden worden ist. Setzt man den Pointer hoch, so werden die 'globalen Variablen' nicht mehr durchsucht und können so auch nicht verändert werden. Will man Werte von Variablen mit in eine andere 'Ebene' übernehmen, so muß man diese suchen, den Wert holen, die Ebene umschalten und dann auf dieser Ebene den Variablen den Wert zuweisen. Beim Hochschalten merkt man sich die alten Werte der Pointer in einer Tabelle und setzt den Anfang der Variablen auf das Ende der Arrays. Beim Herunterschalten werden diese dann wieder zurückgeholt.

Um das Programm möglichst kurz-

zuhalten, wurden einige Routinen des Basic-Interpreters und des Betriebssystems verwendet, ebenso einige spezielle Speicherstellen, die in der Zero-Page liegen. Diese sind im einzelnen:

\$000D —Typflag für numerisch (= \$00) und String (= \$FF). Das Flag wird gesetzt, wenn ein Ausdruck ausgewertet oder eine Variable gesucht wird.

\$000E —Typflag für Integer (= \$80) und Real (= \$00). Siehe oben

\$002D —(1,h) Zeiger auf Start der Variablen

\$002F —(1,h) Zeiger auf Start der Array-Tabelle

\$0031 —(1,h) Zeiger auf Ende der Array-Tabelle

\$0045 —(1,h) Variablenname

\$0073 —CHRGET-Routine erhöht Zeiger in den Basic-Text und holt das nächste Zeichen in den Akkumulator.

\$007A —(1,h) Zeiger in den Basic-Text, zeigt auf den laufenden Befehl beziehungsweise das laufende Zeichen.

\$0308 —(1,h) Vektor für die Interpreterschleife (\$A7E4).

\$A462 —Sperrt 'CONT' und gibt 'ERROR' aus, danach Sprung in den Direkt-Modus.

\$A7E7 —'Normale' Interpreterschleife. Holt nächsten Befehl und führt diesen aus.

\$A7AE —Beginn der Interpreterschleife. Testet auf STOP, merkt Zeiger für CONT, erkennt Programmende.

\$A9C4 —Wertzweisung Integer. Weist einer Integervariablen den Wert des Fließkomma-Akku #1 zu.

\$AA2C —Wertzweisung String. Weist einer Stringvariablen den aktuellen Stringdeskriptor zu.

\$AF08 —Gibt 'SYNTAX ERROR' aus. Danach Sprung in den Direkt-Modus.

\$AF28 —Holt Wert einer Variablen (Zeiger = \$7A/7B)

\$B08B —Sucht Adresse einer Variablen

\$BBD0 —Wertzweisung an Realvariable.

\$FFD2 —Gibt Zeichen aus Akku auf dem Bildschirm aus.

Das Programm kann mit Hilfe des Basic-Loaders oder mit Hilfe des

Checkers eingegeben werden. Eine Anpassung an den VC-20 ist kein Problem, denn die verwendeten Adressen und Routinen existieren auch bei diesem Computer, es muß nur zu den Adressen, die im Bereich von \$A000 bis \$BFFF liegen, ein Wert von \$2000 addiert werden. \$A462 beim C 64 ergibt also \$C462 beim VC-20. Da das Programm aber in gerade diesem Bereich liegt und beim VC-20 dort normalerweise kein RAM-Bereich ist, muß das Programm in einen anderen Bereich verschoben werden.

Beispiel 1:

10 A = 1:B = 2

20 !PROC

30 REM IRGENDWELCHE
BERECHNUNGEN

40 !EPROC

50 END

In Zeile 30 sind alle Variablen unbekannt! In Zeile 50 haben A und B wieder die Werte 1 und 2, egal welcher Wert ihnen in Zeile 30 zugewiesen wurde!

Beispiel 2:

10 A = 1:B% = 4:C\$ = "PROC-TEST"

20 !PROC(A,C\$)

30 REM

40 A = 5:C\$ = "TEST-ENDE"

50 !EPROC(C\$)

60 END

In Zeile 30 haben die Variablen A und C\$ ihre vorherigen Werte, also 1 und „PROC-TEST“, während B% und alle anderen Variablen keinen beziehungsweise den Wert 0 haben. In Zeile 60 hat A wieder den Wert 1, B% den Wert 4 und C\$ hat den Inhalt "TEST-ENDE".

Beispiel 3:

Berechnung der Fakultät durch Rekursion nach der Formel

$N! = N * (N - 1)!$

$1! = 1$

10 INPUT "ZAHL:";N

20 GOSUB 100

30 F = F * N

40 PRINT "FAKULTÄT VON "N"
IST "F

50 END

100 !PROC(F,N)

110 N = N - 1:IF N > 1 THEN GOSUB

100:F = F * N:GOTO 130

120 F = 1

130 !EPROC(F)

140 RETURN

(Dirk Sagwitz)

Computer unter Kontrolle

Bei der Fehlersuche ist es praktisch, wenn ein Programm nicht in seinem normalen Tempo abläuft. 'Trace' verlangsamt den Programmablauf um das 70fache (Mit Poke 2,15). 'Tracen' ist aber nur sinnvoll, wenn man weiß, welche Zeile gerade abgearbeitet wird. Deshalb wird im linken oberen Eck die Zeilennummer eingeblendet. Will man noch mehr bremsen, kann man zusätzlich ein paar Print-Befehle verwenden.

Programmfehler, die keine Errormeldung provozieren, sind nur zu finden, wenn man den Computer Schritt für Schritt kontrolliert.

dem Interrupt in eine eigene Routine gesprungen. Dort wird getestet, ob der C64 gerade im Direkt-Modus läuft. Wenn ja, wird sofort in die normale IRQ-Routine gesprungen. Andernfalls wird die aktuelle Zeilennummer ausgegeben. Vorher wird noch eine Verzögerungsschleife durchlaufen, sofern die Verzögerung auf einen Wert größer als Null eingestellt ist.

Die Routine wird mit SYS 49152 auch wieder ausgeschaltet.

(Andreas Prott)

70mal langsamer

Die Verzögerung wird mit „POKE 2,X“ eingestellt. „X“ darf Werte zwischen 0 (keine Verzögerung = Anfangseinstellung) und 15 annehmen (70fache Verlangsamung). Dies kann auch innerhalb des Programms geschehen; es können also auch einzelne Programmteile „getraced“ werden. Nach Drücken von RUN/STOP + RESTORE muß 'Trace' wieder eingeschaltet werden.

Das Verbiegen des Interrupt-Vektors liegt diesem Programm zugrunde.

Jede sechzigstel Sekunde unterbricht ein Timer den Programmablauf, damit das Betriebssystem verschiedene Routinearbeiten durchführen kann.

Beim Start mit SYS 49152 wird 'Trace' in diesen System-Interrupt eingebettet. Nun wird bei je-

```

7 REM *****
8 REM * AKTIVIEREN MIT SYS 12*4096 *
9 REM * 12*4096=49152=#C000 *
10 REM * VERZOEGERUNG: POKE2,X *
11 REM * 0<X<15 ! *
12 REM *****
13 :
30 FOR I=12*4096 TO 12*4096+135                2193
31 READ A:S=S+A                                1130
32 POKE I,A                                     483
33 NEXT I                                       276
34 IF S<>17493 THEN PRINT "ERROR IN DATAS!"    2729
35 :
40 DATA 120,173,21,3,201,234,208,12,169,192    4316
50 DATA 141,21,3,169,36,141,20,3,208,10        3576
60 DATA 169,49,141,20,3,169,234,141,21,3      3678
70 DATA 88,169,0,133,2,96,165,157,48,83      3549
80 DATA 165,2,240,13,201,16,176,9,168,162    3591
90 DATA 62,232,208,253,136,208,248,165,58,133 3692
100 DATA 252,165,57,133,251,160,0,162,8,169   3727
110 DATA 48,133,253,165,252,221,127,192,144,28 3518
120 DATA 209,7,165,251,221,126,192,144,19,56  4066
130 DATA 165,251,253,126,192,133,251,165,252,253 5134
140 DATA 127,192,133,252,230,253,208,221,165,253 4891
150 DATA 153,0,4,169,1,153,0,216,200,202     3378
160 DATA 202,16,202,76,49,234,1,0,10,0      3213
170 DATA 100,0,232,3,16,39                   1826
    
```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|----|--------|
| C000 | 78 | AD | 15 | 03 | C9 | EA | D0 | 0C | 80 |
| C008 | A9 | C0 | 8D | 15 | 03 | A9 | 24 | 8D | 53 |
| C010 | 14 | 03 | D0 | 0A | A3 | 31 | 8D | 14 | C7 |
| C018 | 03 | A9 | EA | 8D | 15 | 03 | 58 | A9 | 78 |
| C020 | 00 | 85 | 02 | 60 | A5 | 9D | 30 | 53 | 03 |
| C028 | A5 | 02 | F0 | 0D | C9 | 10 | B0 | 09 | E4 |
| C030 | A8 | A2 | 3E | E8 | D0 | FD | 88 | D0 | E8 |
| C038 | F8 | A5 | 3A | 85 | FC | A5 | 39 | 85 | 13 |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| C040 | FB | A0 | 00 | A2 | 08 | A9 | 30 | 85 | C4 |
| C048 | FD | A5 | FC | DD | 7F | C0 | 90 | 1C | AA |
| C050 | D0 | 07 | A5 | FB | DD | 7E | C0 | 90 | 01 |
| C058 | 13 | 38 | A5 | FB | FD | 7E | C0 | 85 | 3F |
| C060 | FB | A5 | FC | FD | 7F | C0 | 85 | FC | 9D |
| C068 | E6 | FD | D0 | DD | A5 | FD | 99 | 00 | 1D |
| C070 | 04 | A9 | 01 | 99 | 00 | D8 | C8 | CA | EB |
| C078 | CA | 10 | CA | 4C | 31 | EA | 01 | 00 | 80 |
| C080 | 0A | 00 | 64 | 00 | E8 | 03 | 10 | 27 | 3B |

END OF MAE PASS!

```

1000 ;*****
1010 ;*      TRACE-MS 6502      *
1020 ;*****
1030 ;
1040 ;*****
1050 ;*
1060 ;*      ANDREAS PROTT      *
1070 ;*      THERESIENSTR. 100 *
1080 ;*      8000 MUENCHEN 2   *
1090 ;*****
1100 ;
1110 ;EIN- BZW. AUSSCHALTEN
1120 ;DURCH 'SYS 12*4096'
1130 ;VERZOEGERUNG VON 1-15
1140 ;WAEHLBAR (POKE2,VERZOE.)
1150 ;
1160 VERZOEGER .DE $02
1170 ZEILENNR .DE $FB
1180 ZIFFER .DE $FD
1190 IRQ .DE $0314
1200 ;
1210 .BA $C000
1220 .OS
1230 ;
1240 ;**** EIN- UND AUSSCHALTEN ****
1250 ;
C000- 7B      1260 SEI ;INTERRUPT VERHINDERN
C001- AD 15 03 1270 LDA IRQ+1
C004- C9 EA      1280 CMP #$EA
C006- D0 0C      1290 BNE AUS ;AUSSCHALTEN =>
C00B- A9 C0      1300 LDA #H,NEU ;EINSCHALTEN!
C00A- BD 15 03   1310 STA IRQ+1
C00D- A9 24      1320 LDA #L,NEU
C00F- BD 14 03   1330 STA IRQ
C012- D0 0A      1340 BNE FERTIG
C014- A9 31      1350 AUS LDA #$31 ;AUSSCHALTEN!
C016- BD 14 03   1360 STA IRQ
C019- A9 EA      1370 LDA #$EA
C01B- BD 15 03   1380 STA IRQ+1
C01E- 5B      1390 FERTIG CLI ;INTERRUPT WIEDER ZULASSEN
C01F- A9 00      1400 LDA #$00
C021- 85 02      1410 STA *VERZOEGER ;VERZOEGERUNG AUF NULL
C023- 60      1420 RTS
1430 ;
1440 ;**** NEUER IRQ-ANFANG ****
1450 ;
C024- A5 9D      1460 NEU LDA *$9D
C026- 30 53      1470 BMI SPRUNG ;DIREKTMODUS =>
C028- A5 02      1480 LDA *VERZOEGER ;*** VERZOEGERUNG ***
C02A- F0 0D      1490 BEQ KEINEVERZ ;KEINE VERZOEGERUNG =>
C02C- C9 10      1500 CMP #$10
C02E- B0 09      1510 BCS KEINEVERZ ;>15 => KEINE VERZOEGERUNG
C030- AB      1520 TAY
C031- A2 3E      1530 LOOP1 LDX #$3E ;VERZOEGERUNGSSCHLEIFEN

```

```

C033- E8          1540 LOOP2 INX
C034- D0 FD      1550 BNE LOOP2
C036- 88          1560 DEY
C037- D0 FB      1570 BNE LOOP1
                1580 ;*** AUSGABE DER ZEILENNUMMER ***
C039- A5 3A      1590 KEINEVERZ LDA *$3A ;AKTUELLE ZEILENNUMMER
C03B- 85 FC      1600 STA *ZEILENNR+1 ;MERKEN
C03D- A5 39      1610 LDA *$39
C03F- 85 FB      1620 STA *ZEILENNR
C041- A0 00      1630 LDY #$00 ;OFFSET BILDSCHIRM
C043- A2 08      1640 LDX #$08 ;OFFSET VERGLEICHSTABELLE
C045- A9 30      1650 NEXTZIFFER LDA #$30 ;AKTUELLE ZIFFER
C047- 85 FD      1660 STA *ZIFFER ;ANFANGSWERT = 0
C049- A5 FC      1670 VERGLEICH LDA *ZEILENNR+1
C04B- DD 7F CO   1680 CMP LISTE+1,X
C04E- 90 1C      1690 BCC AUSGABE ;HBYTE ZEILE < LISTE =>
C050- D0 07      1700 BNE ABZIEHEN ;ZEILE < LISTE =>
C052- A5 FB      1710 LDA *ZEILENNR ;HBYTES SIND GLEICH
C054- DD 7E CO   1720 CMP LISTE,X ;LBYTES?
C057- 90 13      1730 BCC AUSGABE ;ZEILE<LISTE => AUSGABE
C059- 38         1740 ABZIEHEN SEC ;ZEILENNUMMER:=
C05A- A5 FB      1750 LDA *ZEILENNR ;:=ZEILENNR-VERGLEICHSWERT
C05C- FD 7E CO   1760 SBC LISTE,X
C05F- 85 FB      1770 STA *ZEILENNR
C061- A5 FC      1780 LDA *ZEILENNR+1
C063- FD 7F CO   1790 SBC LISTE+1,X
C066- 85 FC      1800 STA *ZEILENNR+1
C068- E6 FD      1810 INC *ZIFFER ;ZIFFER ERHOEHEN
C06A- D0 DD      1820 BNE VERGLEICH ;NOCH EINMAL VERGLEICHEN
C06C- A5 FD      1830 AUSGABE LDA *ZIFFER ;* AUSGABE EINER ZIFFER *
C06E- 99 00 04   1840 STA $0400,Y
C071- A9 01      1850 LDA #$01
C073- 99 00 DB   1860 STA $DB00,Y ;IN WEISSER FARBE!
C076- CB         1870 INY ;BS-OFFSET ERHOEHEN
C077- CA         1880 DEX ;VERGLEICHSTABELLEN-OFFSET
C078- CA         1890 DEX ;ERNIEDRIGEN
C079- 10 CA      1900 BPL NEXTZIFFER ;NOCH NICHT FERTIG =>
C07B- 4C 31 EA   1910 SPRUNG JMP $EA31 ;SPRUNG ZUR EIGENTLICHEN
                1920 ;                               IRQ-ROUTINE
                1930 ;
C07E- 01 00      1940 LISTE .BY $01 $00 ; 1 VERGLEICHSTABELLE
C080- 0A 00      1950 .BY $0A $00 ; 10
C082- 64 00      1960 .BY $64 $00 ; 100
C084- E8 03      1970 .BY $E8 $03 ; 1000
C086- 10 27      1980 .BY $10 $27 ; 10000
                1990 ;
                2000 ;*** ENDE ***
                2010 .EN

```

END OF MAE PASS!

--- LABEL FILE: ---

| | | |
|------------------|------------------|-----------------|
| ABZIEHEN =C059 | AUS =C014 | AUSGABE =C06C |
| FERTIG =C01E | IRQ =0314 | KEINEVERZ =C039 |
| LISTE =C07E | LOOP1 =C031 | LOOP2 =C033 |
| NEU =C024 | NEXTZIFFER =C045 | SPRUNG =C07B |
| VERGLEICH =C049 | VERZOEGL =0002 | ZEILENNR =00FB |
| ZIFFER =00FD | | |
| //0000,C08B,C08B | | |

Input ohne type mismatch

Gibt man innerhalb eines Programms zum Beispiel 2/7 ein, führt das zur Fehlermeldung „redo from start“. Das gleiche passiert, wenn man ein Bogenmaß mit π verwendet. Der Basic-Interpreter liest alle nichtnumerischen Zeichen (mit Ausnahme des Punktes) als Teil eines Strings und akzeptiert sie nicht. Das Komma nimmt er an, schneidet aber mit der Meldung „extra ignored“ die Nachkommastellen ab.

Brüche und Bogenmaß akzeptiert der C 64 nur als Dezimalzahl. Wer mathematische Schreibweisen benutzen will, dem hilft diese praktische Subroutine.

Eingabe mit

Strichrechnung

Um das zu vermeiden, wird statt des numerischen Inputs ein String abgefragt. Dieser wird mit Hilfe der Stringfunktionen zerlegt und zu den gewünschten Zahlenvariablen wieder zusammengesetzt. Wer außer

den hier vorgesehenen Konstanten π und $e = 2.718281828$... anderen Tasten irgendwelche Konstanten zuordnen will, kann dies auch tun, indem er zwischen 30630 und 30660 Zeilen nach dem Muster der Zeilen 30640 und 30650 einfügt.

Die Syntax der Stringvariablen: In einem Eingabeterm kann multipli-

ziert und dividiert, nicht aber addiert, subtrahiert und potenziert werden, Klammern sind verboten, der Malpunkt wird, wie bei mathematischen Termen üblich, nicht gesetzt.

Pi und e als Extras

Das heißt, das einzig zulässige Rechenzeichen ist „/“ für die Division. In der vorliegenden Version sind zwei Buchstaben als Variable festgelegt: π steht für die Kreiszahl und e steht für die Eulersche Zahl. Damit keine Verwechslungen mit dem „e“ in der Exponentialdarstellung einer reellen Zahl aufkommen, müssen π und e im Zähler wie im Nenner stets an letzter Stelle stehen.

Beispiele:

$\frac{2}{3} \pi$ wird eingegeben als: 2 π /3
„RETURN“ (nicht etwa 2/3 π „RETURN“)

$\frac{1}{7} \cdot \frac{5\pi^2}{e^5}$:5 $\pi\pi$ /7eeeee „RETURN“
 $\frac{e \cdot 10^{-5}}{3\pi}$:1e-5e/3 π „RETURN“

Das Programm ist als Subroutine gedacht. Deshalb müssen die Variablen von Haupt- und Unterprogrammen aufeinander abgestimmt werden. Folgende Variablen finden Verwendung:

V;V%;V\$;VF;VL%;VN;VN%;VP\$;VP%;VZ;VZ%

Gleichnamige Variablen im Hauptprogramm müssen geändert werden. Das Hauptprogramm übergibt ans Unterprogramm die String-Variablen „VAR\$“; das Unterprogramm gibt die numerische Variable „VAR“ zurück. Im Hauptprogramm muß deshalb der numerische Input A wie folgt geändert werden:

Aus... INPUT A:.. wird
... INPUT VAR\$:GOSUB 30500:
A=VAR:

(Horst-Arnim Kosog)

```

30500 VL%=LEN (VAR$):IF RIGHT$ (VAR$,1)<>"/" 3255
      GOTO 30520
30510 INPUT "[2SPACES]FEHLER! NEUE EINGABE"; 3695
      VAR$:GOTO 30500
30520 FOR V=1 TO VL% 1001
30530 IF MID$ (VAR$,V,1)="/" THEN VN%=VL%-V: 4502
      VZ%=V-1:GOTO 30550
30540 NEXT V:VZ%=VL%:VN%=0 2404
30550 VN=1:IF VN%>0 THEN V$=RIGHT$ (VAR$, 6153
      VN%):V%=VN%:GOSUB 30600:VN=V:IF V$="F"
      GOTO 30510
30560 IF VN=0 GOTO 63999 1262
30570 V$=LEFT$ (VAR$,VZ%):V%=VZ%:GOSUB 4801
      30600:VZ=V:IF V$="F" GOTO 30510
30580 VAR=VZ/VN:RETURN 1016
30600 VF=1:FOR V=V% TO 1 STEP -1 1804
30610 VP$=MID$ (V$,V,1):VP%=ASC (VP$) 2316
30620 IF (VP%>47) AND (VP%<58) GOTO 30680 2343
30630 IF VP%=45 THEN V=VF*(-1):RETURN 2016
30640 IF VP%=255 THEN VF=VF*[PI]:GOTO 30670 2624
30650 IF VP%=63 THEN VF=VF*2.718281828:GOTO 3488
      30670
30660 V$="F":RETURN 758
30670 NEXT V:V=VF:RETURN 1273
30680 V$=LEFT$ (V$,V):V=VF*VAL (V$):RETURN 2626
    
```

Zeilennummern im Zehnerschritt

Die ersten Programmierschritte macht man am besten mit großen Programmschritten. Das ist eine alte Weisheit. Spätestens nach dem ersten Probelauf werden sich die Lücken mit neuen Zeilen füllen. Wer das Durcheinander wieder auf die Reihe bringen will, kann das mit „Renumber“ tun. Renumber ordnet alle Programmzeilen in Zehnerschritten. Damit nach der Umnummerierung das Programm noch funktioniert, müssen auch alle Sprünge geändert werden: GOTO, GOSUB, RUN und THEN. Auch ON X GOTO ... und ON X GOSUB ... werden mitverändert. Nach dem Start wird das Maschi-

Richtigen Abstand halten ist nicht nur guter Fahrstil ... auch guter Programmierstil.

nenprogramm in den Speicherbereich geschrieben, der vor Reset und Basic-Interpreter geschützt ist. Am Ende löscht sich der Basiclader selbst. Jetzt kann man die Programme, die man „renumbern“ will, wie gewohnt laden. Mit SYS (12*4096) wird das Maschinenprogramm gestartet. Nach

drei Sekunden ist die ganze Arbeit getan. Jetzt das „neue“ Programm, am besten unter anderem Namen, abspeichern.

Renumber in drei

Sekunden

Solange man den Computer nicht ausschaltet, ist das Programm vorhanden und kann mit SYS immer wieder aktiviert werden. Einem Renumber der gesamten Programmbibliothek steht nichts mehr im Wege.

(A. Boldt/J. Ackermann)

| | | | | | |
|-------|--|------|-----------|--|--|
| 00010 | print "(clr)";chr\$(14)+chr\$(8);"Bitte warten!" | 3173 | | | |
| 00020 | s=0;for i=0 to 742: read a:s=s+a:poke 12*4096+i,a:next | 4384 | | | |
| 00030 | if s<>96871 then print "(clr,2down) data-fehler!":end | 2834 | | | |
| 00040 | print "(clr)" chr\$(9)+chr\$(142); | 1449 | | | |
| 00050 | poke 198,0:new | 1093 | | | |
| 00060 | data 160,0,152,133,87,169,203,133,88,165,43,133,251,165,44,133,252,177 | 6886 | | | |
| 00070 | data 251,133,253,230,251,208,2,230,252,177,251,133,254,208,7,165,253,208 | 7229 | | | |
| 00080 | data 3,76,119,192,230,251,208,2,230,252,177,251,145,87,200,177,251,145 | 6707 | | | |
| 00090 | data 87,24,165,87,105,2,133,87,144,2,230,88,56,165,87,133,80,133,82,165 | 6707 | | | |
| 00100 | data 88,233,203,133,81,133,83,6,80,38,81,6,80,38,81,24,165,80,101,82,133 | 6823 | | | |
| 00110 | data 80,165,81,101,83,145,251,160,0,165,80,145,251,165,253,133,251,165 | 6357 | | | |
| 00120 | data 254,133,252,76,17,192,24,165,43,105,4,133,251,165,44,105,0,133,252 | 7583 | | | |
| 00130 | data 160,0,177,251,201,137,240,70,201,141,240,66,201,167,240,39,201,138 | 6763 | | | |
| 00140 | data 240,35,230,251,208,2,230,252,160,0,177,251,208,226,160, | 6658 | | | |
| | | | 2,177,251 | | |
| 00150 | data 208,3,76,153,194,24,165,251,105,5,133,251,144,2,230,252,76,132,192 | 7229 | | | |
| 00160 | data 200,177,251,201,32,240,249,201,48,48,7,201,58,176,3,76,214,192,160 | 7114 | | | |
| 00170 | data 0,76,152,192,169,0,133,2,200,177,251,201,0,240,10,201,58,240,6,201 | 6746 | | | |
| 00180 | data 44,208,241,132,2,136,132,79,200,169,1,133,80,133,82,169,0,133,81 | 6726 | | | |
| 00190 | data 133,83,133,90,133,91,136,240,97,177,251,201,32,240,247,133,89,169 | 7327 | | | |
| 00200 | data 0,133,87,133,88,162,4,102,89,144,13,24,165,87,101,80,133,87,165,88 | 7456 | | | |
| 00210 | data 101,81,133,88,6,80,38,81,202,208,232,165,82,133,80,165,83,133,81 | 6165 | | | |
| 00220 | data 24,165,87,101,90,133,90,165,88,101,91,133,91,6,80,38,81,6,80,38,81 | 6857 | | | |
| 00230 | data 24,165,80,101,82,133,80,165,81,101,83,133,81,6,80,38,81,165,80,133 | 6631 | | | |
| 00240 | data 82,165,81,133,83,76,251,192,162,1,160,1,169,203,133,88,169,0,133 | 6802 | | | |
| 00250 | data 87,177,87,213,90,240,37,144,21,165,2,240,14,24,101,251,133,251,144 | 7117 | | | |
| 00260 | data 2,230,252,160,0,76,210,192,76,152,192,24,165,87,105,2, | 6232 | | | |
| 00270 | data 2,230,88,76,107,193,136,202,240,209,56,165,87,233,254,133,80,133 | 7071 | | | |
| 00280 | data 82,165,88,233,202,133,81,133,83,6,80,38,81,6,80,38,81,24,165,80,101 | 6542 | | | |
| 00290 | data 82,133,92,165,81,101,83,133,93,162,255,160,16,169,0,133,94,133,90 | 7005 | | | |
| 00300 | data 133,91,6,92,38,93,38,94,6,90,38,91,165,94,201,10,144,11,56,233,10 | 6862 | | | |
| 00310 | data 133,94,230,90,208,2,230,91,136,208,226,24,105,48,232,157,0,202,165 | 6499 | | | |
| 00320 | data 90,208,7,165,91,208,3,76,7,194,165,90,133,92,165,91,133,93,76,194 | 6890 | | | |
| 00330 | data 193,134,78,232,228,79,240,92,48,81,230,45,208,2,230,46,165,2,240 | 6202 | | | |
| 00340 | data 2,230,2,164,45,169,0,133,90,169,1,133,92,165,46,133,91,133,93,197 | 6866 | | | |
| 00350 | data 252,240,42,177,90,145,92,136,192,255,208,247,198,93,198,91,165,252 | 7485 | | | |
| 00360 | data 197,91,144,237,208,37,165,251,73,255,168,24,165,251,133,90,105,1 | 7035 | | | |
| 00370 | data 133,92,144,2,230,93,76,48,194,56,165,45,229,251,176,232,164,79,169 | 8323 | | | |
| 00380 | data 32,145,251,136,208,251,160,1,166,78,189,0,202,145, | 6027 | | | |

00390 251,200,202,16
 data 247,165,2,208, 6573
 16,230,78,24,165,
 251,101,78,133,251,
 144,2,230,252,76
 00400 data 152,192,24,101, 6023
 251,133,251,144,2,
 230,252,160,0,76,
 210,192,165,43

00410 data 133,251,133, 7348
 253,165,44,133,252,
 133,254,160,0,24,
 165,251,105,3,133
 00420 data 251,144,2,230, 6709
 252,230,251,208,2,
 230,252,177,251,208,
 246,230,251,208
 00430 data 2,230,252,165, 6298

251,145,253,200,165,
 252,145,253,177,251,
 208,1,96,165
 00440 data 251,133,253, 7062
 165,252,133,254,136,
 76,167,194,0,0,0,0,
 0,0,0,0,0,0,0

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|--------|----|------|----|----|----|----|----|----|----|----|----|----|
| C000 | A0 | 00 | 98 | 85 | 57 | A9 | CB | 85 | A6 | C170 | 25 | 90 | 15 | A5 | 02 | F0 | 0E | 18 | 0F | |
| C008 | 58 | A5 | 2B | 85 | FB | A5 | 2C | 85 | 57 | C178 | 65 | FB | 85 | FB | 90 | 02 | E6 | FC | 6C | |
| C010 | FC | B1 | FB | 85 | FD | E6 | FB | D0 | 02 | C180 | A0 | 00 | 4C | D2 | C0 | 4C | 98 | C0 | 65 | |
| C018 | 02 | E6 | FC | B1 | FB | 85 | FE | D0 | 0B | C188 | 18 | A5 | 57 | 69 | 02 | 85 | 57 | 90 | A6 | |
| C020 | 07 | A5 | FD | D0 | 03 | 4C | 77 | C0 | FE | C190 | 02 | E6 | 58 | 4C | 6B | C1 | 88 | CA | 91 | |
| C028 | E6 | FB | D0 | 02 | E6 | FC | B1 | FB | 58 | C198 | F0 | D1 | 38 | A5 | 57 | E9 | FE | 85 | 99 | |
| C030 | 91 | 57 | C8 | B1 | FB | 91 | 57 | 18 | F9 | C1A0 | 50 | 85 | 52 | A5 | 58 | E9 | CA | 85 | 2D | |
| C038 | A5 | 57 | 69 | 02 | 85 | 57 | 90 | 02 | 70 | C1A8 | 51 | 85 | 53 | 06 | 50 | 26 | 51 | 06 | 8E | |
| C040 | E6 | 58 | 38 | A5 | 57 | 85 | 50 | 85 | BE | C1B0 | 50 | 26 | 51 | 18 | A5 | 50 | 65 | 52 | CA | |
| C048 | 52 | A5 | 58 | E9 | CB | 85 | 51 | 85 | 2F | C1B8 | 85 | 5C | A5 | 51 | 65 | 53 | 85 | 5D | 86 | |
| C050 | 53 | 06 | 50 | 26 | 51 | 06 | 50 | 26 | EB | C1C0 | A2 | FF | A0 | 10 | A9 | 00 | 85 | 5E | 02 | |
| C058 | 51 | 18 | A5 | 50 | 65 | 52 | 85 | 50 | D5 | C1C8 | 85 | 5A | 85 | 5B | 06 | 5C | 26 | 5D | C8 | |
| C060 | A5 | 51 | 65 | 53 | 91 | FB | A0 | 00 | 9C | C1D0 | 26 | 5E | 06 | 5A | 26 | 5B | A5 | 5E | 87 | |
| C068 | A5 | 50 | 91 | FB | A5 | FD | 85 | FB | 1D | C1D8 | C9 | 0A | 90 | 0B | 38 | E9 | 0A | 85 | 18 | |
| C070 | A5 | FE | 85 | FC | 4C | 11 | C0 | 18 | 13 | C1E0 | 5E | E6 | 5A | D0 | 02 | E6 | 5B | 88 | F3 | |
| C078 | A5 | 2B | 69 | 04 | 85 | FB | A5 | 2C | E1 | C1E8 | D0 | E2 | 18 | 69 | 30 | E8 | 9D | 00 | 83 | |
| C080 | 69 | 00 | 85 | FC | A0 | 00 | B1 | FB | 47 | C1F0 | CA | A5 | 5A | D0 | 07 | A5 | 5B | D0 | 9D | |
| C088 | C9 | 89 | F0 | 46 | C9 | 8D | F0 | 42 | D2 | C1F8 | 03 | 4C | 07 | C2 | A5 | 5A | 85 | 5C | 71 | |
| C090 | C9 | A7 | F0 | 27 | C9 | 8A | F0 | 23 | F5 | C200 | A5 | 5B | 85 | 5D | 4C | C2 | C1 | 86 | 37 | |
| C098 | E6 | FB | D0 | 02 | E6 | FC | A0 | 00 | 42 | C208 | 4E | E8 | E4 | 4F | F0 | 5C | 30 | 51 | 6E | |
| C0A0 | B1 | FB | D0 | E2 | A0 | 02 | B1 | FB | F8 | C210 | E6 | 2D | D0 | 02 | E6 | 2E | A5 | 02 | E0 | |
| C0A8 | D0 | 03 | 4C | 99 | C2 | 18 | A5 | FB | D3 | C218 | F0 | 02 | E6 | 02 | A4 | 2D | A9 | 00 | F0 | |
| C0B0 | 69 | 05 | 85 | FB | 90 | 02 | E6 | FC | 65 | C220 | 85 | 5A | A9 | 01 | 85 | 5C | A5 | 2E | 72 | |
| C0B8 | 4C | 84 | C0 | C8 | B1 | FB | C9 | 20 | 1F | C228 | 85 | 5B | 85 | 5D | C5 | FC | F0 | 2A | E0 | |
| C0C0 | F0 | F9 | C9 | 30 | 30 | 07 | C9 | 3A | 47 | C230 | B1 | 5A | 91 | 5C | 88 | C0 | FF | D0 | 82 | |
| C0C8 | B0 | 03 | 4C | D6 | C0 | A0 | 00 | 4C | F7 | C238 | F7 | C6 | 5D | C6 | 5B | A5 | FC | C5 | 85 | |
| C0D0 | 98 | C0 | A9 | 00 | 85 | 02 | C8 | B1 | 1A | C240 | 5B | 90 | ED | D0 | 25 | A5 | FB | 49 | 1A | |
| C0D8 | FB | C9 | 00 | F0 | 0A | C9 | 3A | F0 | CF | C248 | FF | A8 | 18 | A5 | FB | 85 | 5A | 69 | 31 | |
| C0E0 | 06 | C9 | 2C | D0 | F1 | 84 | 02 | 88 | 81 | C250 | 01 | 85 | 5C | 90 | 02 | E6 | 5D | 4C | 77 | |
| C0E8 | 84 | 4F | C8 | A9 | 01 | 85 | 50 | 85 | 9B | C258 | 30 | C2 | 38 | A5 | 2D | E5 | FB | B0 | 95 | |
| C0F0 | 52 | A9 | 00 | 85 | 51 | 85 | 53 | 85 | BF | C260 | E8 | A4 | 4F | A9 | 20 | 91 | FB | 88 | EC | |
| C0F8 | 5A | 85 | 5B | 88 | F0 | 61 | B1 | FB | ED | C268 | D0 | FB | A0 | 01 | A6 | 4E | BD | 00 | 88 | |
| C100 | C9 | 20 | F0 | F7 | 85 | 59 | A9 | 00 | 76 | C270 | CA | 91 | FB | C8 | CA | 10 | F7 | A5 | 76 | |
| C108 | 85 | 57 | 85 | 58 | A2 | 04 | 66 | 59 | 05 | C278 | 02 | D0 | 10 | E6 | 4E | 18 | A5 | FB | 3D | |
| C110 | 90 | 0D | 18 | A5 | 57 | 65 | 50 | 85 | B3 | C280 | 65 | 4E | 85 | FB | 90 | 02 | E6 | FC | 18 | |
| C118 | 57 | A5 | 58 | 65 | 51 | 85 | 58 | 06 | C4 | C288 | 4C | 98 | C0 | 18 | 65 | FB | 85 | FB | 50 | |
| C120 | 50 | 26 | 51 | CA | D0 | E8 | A5 | 52 | EC | C290 | 90 | 02 | E6 | FC | A0 | 00 | 4C | D2 | 91 | |
| C128 | 85 | 50 | A5 | 53 | 85 | 51 | 18 | A5 | 2E | C298 | C0 | A5 | 2B | 85 | FB | 85 | FD | A5 | 8E | |
| C130 | 57 | 65 | 5A | 85 | 5A | A5 | 58 | 65 | 70 | C2A0 | 2C | 85 | FC | 85 | FE | A0 | 00 | 18 | 64 | |
| C138 | 5B | 85 | 5B | 06 | 50 | 26 | 51 | 06 | A0 | C2A8 | A5 | FB | 69 | 03 | 85 | FB | 90 | 02 | 28 | |
| C140 | 50 | 26 | 51 | 18 | A5 | 50 | 65 | 52 | CA | C2B0 | E6 | FC | E6 | FB | D0 | 02 | E6 | FC | 8C | |
| C148 | 85 | 50 | A5 | 51 | 65 | 53 | 85 | 51 | CE | C2B8 | B1 | FB | D0 | F6 | E6 | FB | D0 | 02 | 49 | |
| C150 | 06 | 50 | 26 | 51 | A5 | 50 | 85 | 52 | 13 | C2C0 | E6 | FC | A5 | FB | 91 | FD | C8 | A5 | 4C | |
| C158 | A5 | 51 | 85 | 53 | 4C | FB | C0 | A2 | F5 | C2C8 | FC | 91 | FD | B1 | FB | D0 | 01 | 60 | 83 | |
| C160 | 01 | A0 | 01 | A9 | CB | 85 | 58 | A9 | AF | C2D0 | A5 | FB | 85 | FD | A5 | FC | 85 | FE | 63 | |
| C168 | 00 | 85 | 57 | B1 | 57 | D5 | 5A | F0 | 0F | C2D8 | 88 | 4C | A7 | C2 | 00 | 00 | 00 | 00 | 21 | |
| | | | | | | | | | | C2E0 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | F3 |

Fehler ohne Programmabsturz

Es gibt immer noch Software zu kaufen, die im Falle eines nicht-eingeschalteten Peripheriegeräts mit der Fehlermeldung 'Device not present' aussteigt. Wer sich die kleine Routine anschaut, die das verhindern kann, ist sich sofort über die „Professionalität“ solcher Programme im klaren.

Poke verhindert

Errormeldung

Mit einem einzigen Poke kann man den Computer dazu bringen, den Fehler zu ignorieren. Dabei wird der Warmstartvektor zum nächsten RTS ('Return to Subrou-

Wer professionell programmiert, fängt bei I/O-Befehlen die Fehlermeldung 'decice not present' ab. Dazu wird der C64 kurzzeitig auf dumm geschaltet.

ne') verbogen. Im Falle von Programmfehlern fährt der C64 jetzt fort, als wäre nichts gewesen. Nun wird versuchsweise ein Kanal zum betreffenden Gerät geöffnet. Dem Drucker wird ein Print #-Be-

fehl, der Floppy ein Initialize-Kommando hinübergeschickt. Danach werden Fehlermeldungen wieder zugelassen. Nun wird geprüft, ob das angesprochene Gerät reagiert hat. Durch Abfragen der Statusvariablen ST erfährt man, ob am anderen Ende der Leitung jemand ist. Das Bit 7 der Statusvariablen wird nämlich gesetzt, wenn am IEC-Bus das erwartete Echo ausbleibt. Hat alles seine Richtigkeit, geht der normale Programmablauf weiter. Andernfalls kann der aufgetretene Fehler behandelt werden.

Den Drucker und die Floppy getrennt abzufragen, hat seinen guten Grund: Was nützt alles Bewahren vor dem Absturz, wenn man hinterher nicht weiß, welches Gerät der Sündenbock war?!

```

100 REM *** DRUCKERTEST ***
110 :
115 REM KANAL EROEFFNEN
120 OPEN 4,4 600
125 :
130 REM FEHLERMELDUNG VERHINDERN
135 POKE 768,185 1070
145 :
150 REM SCHREIBVERSUCH DURCHFUEHREN
155 PRINT#4:CLOSE 4 560
160 :
165 REM FEHLERMELDUNG WIEDERERLAUBEN
170 POKE 768,139 1262
180 :
185 REM STATUS ABFRAGEN
190 IF ST<>-128 THEN 205 1335
195 PRINT "DRUCKER NICHT EINGESCHALTET!" 2665
200 GOTO 215 602
205 PRINT "DRUCKER EINGESCHALTET!" 2646
210 :
215 REM HIER GEHT'S WEITER I

```

```

M PROGRAMMABLAUF
220 :
225 REM *** FLOPPYTEST ***
230 :
235 REM FEHLERMELDUNG VERHINDERN
240 POKE 768,185 1070
250 :
255 REM KANAL EROEFFNEN
260 OPEN 15,8,15,"I":CLOSE 1639
265 :
270 REM FEHLERMELDUNG WIEDERERLAUBEN
275 POKE 768,139 1262
285 :
290 REM STATUS ABFRAGEN
295 IF ST<>-128 THEN 310 1267
300 PRINT "FLOPPY NICHT EINGESCHALTET!" 2629
305 GOTO 320 568
310 PRINT "FLOPPY EINGESCHALTET!" 2492
315 :
320 REM HIER GEHT'S WEITER I
M PROGRAMMABLAUF

```

(Andreas Prott)

Speed up

Manchmal kommt es in einem Programm darauf an, das letzte bißchen Speed aus dem Rechner herauszuholen. Eine Möglichkeit ist, die Interrupt Service Routine (ISR) zu verbieten. Damit wird der Rechner aber nur minimal schneller. Leider wird damit die Tastatur nicht mehr abgefragt und TI\$ nicht mehr gestellt. Viel wirkungsvoller ist es, den Videochip zu manipulieren. Der VIC hält den Prozessor jedesmal auf, wenn er Daten auf den Bildschirm zaubern muß. Besonders deutlich wird das, wenn viele Sprites aktiv sind. Das ist der Grund, warum bei Kassettenoperationen der Bildschirm abgeschaltet wird. Dabei hat der Prozessor nämlich einige zeitkritische Aufgaben zu erledigen, bei denen er nicht gestört werden darf. Was das Betriebssystem kann, kann

Zwei Pokes bringen den C64 auf Turbo-Speed. Bildschirm und Tastatur bleiben auf der Strecke.

der Anwender schon lange. Sobald zeitkritische Programmteile auftreten, wird einfach der VIC abgeschaltet. Das läßt sich durch

Tabelle der Ausführungszeiten

| Betriebsart | Ausführungszeit |
|--------------------|-----------------|
| 8 Sprites aktiv | 1095 sec |
| normal | 1037 sec |
| ohne ISR | 1027 sec |
| ohne VIC | 990 sec |
| ohne VIC, ohne ISR | 978 |

Löschen von Bit 4 in Adresse 53265 erreichen. Natürlich ist dann auf dem Bildschirm nichts mehr zu sehen, aber bei Berechnungen und ähnlichem stört das nicht weiter. Sollte das Programm immer noch zu langsam sein, schaltet man halt auch noch die Interrupt-Routine ab. Und zwar verbietet man den Interrupt direkt an der Quelle: der IRQ-CIA. Beide Tricks zusammen machen den 64er um mehr als zehn Prozent flotter. Hier nun die nötigen POKES:
 VIC abschalten: POKE 53265, PEEK (53265) AND 239
 VIC einschalten: POKE 53265, PEEK (53265) OR 16
 ISR verbieten: POKE 56334, PEEK (56334) AND 254
 ISR erlauben: POKE 56334, PEEK (56334) OR 1
 (sis)

Taube Taste

Mit einer Maschinenroutine kann man Tasten blockieren. Das hilft gegen Fehlbedienung.

Ein Vorzug des 64er ist die Möglichkeit, den Basic-Interpreter und das Kernal vom ROM in ein adressengleiches RAM zu verlegen. Dadurch können Manipulationen am Interpreter oder Kernal vorgenommen werden.

ROM ins RAM mit REM

Erfolgt die Umspeicherung mittels PEEK und POKE, so dauert das 72 Sekunden. Ein entsprechendes Maschinenprogramm benötigt hierfür nur weniger als 0.3 Sekunden!

Das Programm bedient sich der indirekten Adressierung mit Nachindizierung. Die Adressen \$FB/\$FC und \$FD/\$FE sind die Hilfszeiger der indirekten Adressierung.

Nutzanwendung ist die Blockie-

Da hilft kein Hämmern:

Eine verrückte REM-Zeile legt jede Taste lahm.

rung einer beliebigen Taste, wirksam sowohl im Programm- als auch im Direktmodus. Das Basic-Programm erzeugt eine REM-Zeile (Zeilen-Nr. 0), welche das Maschinenprogramm enthält. Die REM-Zeilen 10, 12 und 14 des Basic-Programmes dürfen nicht weg-

```

10 REM *****
12 REM * REM-ROUTINE GENERIEREN *
14 REM *****
16 AD=2048
18 FOR I=0 TO 75:READ X:POKE AD+I,X:NEXT
20 POKE 45,76:POKE 46,8
22 PRINT "[CLR]";LIST
24 END
26 DATA 0,74,8,0,0,143,32
28 DATA 160,1,136,132,251,132,253
30 DATA 169,160,133,252,169,224,133
32 DATA 254,177,251,145,251,177,253
34 DATA 145,253,200,200,245,230,252
36 DATA 230,254,165,252,201,192,208
38 DATA 235,234,169,76,141,180,229
40 DATA 169,60,141,181,229,169,8
42 DATA 141,182,229,96,172,119,2
44 DATA 192,147,208,3,160,1,136
46 DATA 76,183,229,0,0,0
    
```

| | | | | | |
|------|-------|----------|-------------|---|-----------------------------|
| 2055 | #0807 | a0 01 | ldy # 1 | ┌ | |
| 2057 | #0809 | 88 | dey | | Indezähler = 0 setzen |
| 2058 | #080a | 84 fb | sty #fb | | LB von #a000 merken |
| 2060 | #080c | 84 fd | sty #fd | | LB von #e000 merken |
| 2062 | #080e | a9 a0 | lda # 160 | | HB von #a000 |
| 2064 | #0810 | 85 fc | sta #fc | | merken |
| 2066 | #0812 | a9 e0 | lda # 224 | | HB von #e000 |
| 2068 | #0814 | 85 fe | sta #fe | | merken |
| 2070 | #0816 | b1 fb | lda (#fb),y | | [#a000+y] aus ROM in Akku |
| 2072 | #0818 | 91 fb | sta (#fb),y | | [Akku] nach #a000+y (RAM) |
| 2074 | #081a | b1 fd | lda (#fd),y | | [#e000+y] aus ROM in Akku |
| 2076 | #081c | 91 fd | sta (#fd),y | | [Akku] nach #e000+y (RAM) |
| 2078 | #081e | c8 | iny | | Indezähler erhöhen |
| 2079 | #081f | d0 f5 | bne #0816 | | [y] ≠ 0, dann nächste Zelle |
| 2081 | #0821 | e6 fc | inc #fc | | HB erhöhen, wenn Seitenende |
| 2083 | #0823 | e6 fe | inc #fe | | HB erhöhen, wenn Seitenende |
| 2085 | #0825 | a5 fc | lda #fc | | |
| 2087 | #0827 | c9 c0 | cmp # 192 | | Ist HB = 192 (#c0) ? |
| 2089 | #0829 | d0 eb | bne #0816 | | Nein, dann nächste Zelle |
| 2091 | #082b | ea | nop | | |
| 2092 | #082c | a9 4c | lda # 76 | | Befehl jmp #083c als |
| 2094 | #082e | 8d b4 e5 | sta #e5b4 | | Umleitung bei #e5b4 in die |
| 2097 | #0831 | a9 3c | lda # 60 | | Tastaturpuffer-Routine |
| 2099 | #0833 | 8d b5 e5 | sta #e5b5 | | des Kernals einbauen |
| 2102 | #0836 | a9 08 | lda # 8 | | |
| 2104 | #0838 | 8d b6 e5 | sta #e5b6 | | |
| 2107 | #083b | 60 | rts | | zurück zum Basic |
| 2108 | #083c | ac 77 02 | ldy #0277 | | 1. Byte aus Tastaturpuffer |
| 2111 | #083f | c0 93 | cpy # 147 | | Vergl. mit Code für CLR |
| 2113 | #0841 | d0 03 | bne #0846 | | Code ≠ 147, dann -> #0846 |
| 2115 | #0843 | a0 01 | ldy # 1 | | Code = 147, dann |
| 2117 | #0845 | 88 | dey | | [y] = 0 setzen |
| 2118 | #0846 | 4c b7 e5 | jmp #e5b7 | | zur Tastaturpuffer-Routine |

gelassen werden und müssen jeweils mindestens 23 Zeichen enthalten!

Diese REM-Zeile enthält eine wirre Folge von Zeichen und Basic-

Befehlen (Codes über 127 sind für Basic-Befehle reserviert) und erstreckt sich über 5 Bildschirmzeilen. Sie darf von Basic aus nicht verändert werden. Selbst ein in

dieser Zeile durchgeführtes „return“ zerstört das Maschinenprogramm! An die erzeugte REM-Routine kann ein beliebiges Basic-Programm angefügt werden.

Das Maschinenprogramm in der REM-Zeile wird mit SYS 2055 gestartet. Dabei wird das ROM ins RAM kopiert und die Tastaturabfrage-Routine (\$E5B4) des Kernals auf das Maschinenprogramm „umgeleitet“.

Blockade für CLR

Die CLR-Taste ist jetzt noch nicht blockiert, da der Prozessor nach wie vor auf das ROM zugreift. Durch POKE 1,53 wird der Prozessor auf das RAM umgeschaltet und die Umleitung auf die Maschinenroutine wirksam. Mit POKE 1,55 greift der Prozessor wieder auf das ROM zu und die Blockierung ist wieder aufgehoben.

In der Speicherzelle 2112 steht der Code der zu blockierenden Taste. Durch Verändern des Inhaltes von Adresse 2112 kann jede andere Taste blockiert werden.

(H. Büche)

Flexibler Schreibschutz

Jeder Anfänger lernt erst mal, welche „Hardware“-Methode zur Verhinderung von unerwünschtem Schreiben auf die Diskette benutzt wird.

Vielleicht wissen aber noch nicht alle C64-Freunde, wie man durch Software erreichen kann, daß auf die Diskette nicht mehr geschrieben wird. In der BAM ist die Belegung aller Blöcke aufgezeichnet.

BAM voll Nullen

Für jeden belegten Block steht in der BAM an der entsprechenden Stelle eine 0. Wenn der Block noch frei ist, steht dort eine 1. Weiterhin steht in der BAM für jeden

Will man eine Diskette vor unüberlegten Schreibzugriffen bewahren, kann man mit einer kleinen Routine einen flexiblen Schreibschutz anbringen.

einzelnen Track auch die Anzahl der noch freien Blöcke. Nun braucht man nichts anderes zu tun, als in die BAM einzugreifen und die Werte so zu verändern,

daß man zum Schluß eine anscheinend völlig belegte Diskette erhält.

Dazu liest man zunächst Track 18, Sektor 0 von der Diskette/Programmzeilen 80–175/. Jetzt ist der Block, in dem sich die BAM befindet, bereits im Speicher des C64. In den Zeilen 180–186 erfolgt das Verändern der Werte, das heißt, man setzt in der BAM alle Werte gleich 0. Zum Schluß müssen die neuen Werte wieder auf die Diskette geschrieben werden /Programmzeilen 200–230/.

Zeile 182 sorgt dafür, daß die Stellen in der BAM, die für Track 18 zuständig sind, unberührt bleiben. Diese Programmzeile kann aber auch weggelassen werden. Die

| | | | |
|---|------|--|------|
| 35 DIM Q\$(255) | 931 | 170 NEXT | 130 |
| 40 PRINT "[CLR,YEL, 11DOWN]":POKE 53280, 6:POKE 53281,6 | 2607 | 175 CLOSE 5 | 266 |
| 45 PRINT "[6SPACES,RVS] BELEGEN ALLER BLOECKE [OFF]" | 2810 | 180 FOR I=4 TO 143 | 916 |
| 50 FOR I=0 TO 5000:NEXT | 1056 | 182 IF I>71 AND I<76 THEN | 2018 |
| 60 PRINT "[CLR,11DOWN]"; | 1289 | 186 | |
| 70 PRINT "[CLR,12SPACES] GESTARTET." | 1470 | 184 Q\$(1)=CHR\$(0) | 1010 |
| 80 OPEN 15,8,15:OPEN 5, 8,5,"#":PRINT#15,"U1: 5,"0,18,0 | 4027 | 186 NEXT | 130 |
| 140 FOR I=0 TO 255 | 981 | 200 OPEN 2,8,2,"#" | 1023 |
| 150 A\$="":GET #5,A\$:IF A\$="" THEN A\$=CHR\$(0) | 2557 | 210 FOR I=2 TO 255 | 997 |
| 160 Q\$(1)=A\$ | 717 | 212 PRINT#15,"B-P:"2,I | 1351 |
| | | 214 PRINT#2,Q\$(1); | 326 |
| | | 216 NEXT | 130 |
| | | 220 PRINT#15,"U2:2,"0,18, 0 | 1651 |
| | | 230 CLOSE 2 | 260 |
| | | 240 PRINT "[CLR,CBM 61": POKE 53280,4:POKE 53281,4 | 2881 |
| | | 250 PRINT#15,"I":CLOSE 15 | 1269 |

REM-, PRINT- und POKE-Befehle werden nicht unbedingt gebraucht, es können also die Zeilen 10–30, 40–70 sowie 240 auch entfallen.

Wenn man dieses Programm laufen läßt, erhält man eine laut Di-

rectory völlig belegte Diskette. Nach LOAD "\$", 8 und LIST bekommt man nach den enthaltenen Filenamen die Meldung „0 BLOCKS FREE“. Auf diese Diskette kann man nicht mehr schreiben, das Lesen wird jedoch nicht

gestört. Auch können die Disk-Befehle, wie etwa SCRATCH oder RENAME auch weiterhin durchgeführt werden. Durch VALIDATE kann der ursprüngliche Zustand wieder hergestellt werden. (Péter Pusztay/Budapest)

Blocksave und Blockload

Zwei kurze Basic-Routinen erlauben es, beliebige Speicherbereiche, wie zum Beispiel ein Programm, auf Kassette oder Diskette abzuspeichern und zu laden.

Die Routinen sind als Unterprogramme ab Zeile 60 000 ausgeführt und können so leicht an bestehende Basic-Programme angehängt werden. Ab Zeile 60 000 befindet sich die SAVE-Routine, ab Zeile 61 000 die LOAD-Routine.

Kassette oder Disk

Zunächst wird abgefragt, ob mit Kassette (K) oder Diskette (D) gearbeitet werden soll. Dann folgt die Eingabe des Filenamens. Wird hier direkt mit RETURN geantwortet, so wird der Block ohne

*Maschinenprogramme
muß man aus einem be-
stimmten Bereich save
und in denselben wie-
der laden. Da hilft kein
'save' und kein 'load',
sondern nur ein
Programm.*

Dateiname gespeichert, beziehungsweise das nächste File geladen. Das ist nur mit der Datasette möglich, da die Floppy einen File-

namen verlangt. Jetzt müssen nur noch die Start- und Endadressen eingegeben werden. Dabei ist zu beachten, daß die Endadresse immer die Adresse des ersten Bytes nach dem zu speichernden Block ist. Bei der LOAD-Routine wird keine Endadresse abgefragt, sie ergibt sich aus der Länge des zu ladenden Files.

Nachladen ohne

Programmverlust

Anzumerken wäre noch, daß sich mit BLOCKSAVE gespeicherte Blöcke auch mit LOAD "filename", Gerätenummer,1 wieder laden lassen; allerdings ist hier zu be-

```

60000 REM *****
60010 REM * BLOCKSAVE-MODUL *
60020 REM *****
60030 :
60040 PRINT "KASSETTE ODER DISK(2SPACES)(K/D)" 4667
      ;:INPUT DV#:DV=1:IF DV#="D" THEN DV=0
60050 PRINT "FILENAME :":INPUT FI#:IF FI#="" 4607
      THEN POKE 103,0:GOTO 60090
60060 FA=500 708
60070 FOR I=1 TO LEN (FI#):POKE FA+I-1,ASC ( 3750
      MID# (FI#,I,1)):NEXT
60080 POKE 103,LEN (FI#):POKE 100,INT (FA/ 4573
      256):POKE 107,FA-PEEK (100)*256
60090 POKE 105,1:POKE 106,DV 1963
60100 INPUT "STARTADRESSE":AN:AH=INT (AN/256): 4929
      AL=AN-AH*256
60110 INPUT "ENDADRESSE":EN:EH=INT (EN/256): 3493
      EL=EN-EH*256
60120 POKE 103,AL:POKE 104,AH:POKE 174,EL:POKE 3089
      175,EH
60130 SYS 62957 900
60140 :
60150 :
61000 REM *****
61010 REM * BLOCKLOAD-MODUL *
61020 REM *****
61030 :
61040 PRINT "KASSETTE ODER DISK(2SPACES)(K/D)" 4667
      ;:INPUT DV#:DV=1:IF DV#="D" THEN DV=0
61050 PRINT "FILENAME :":INPUT FI#:IF FI#="" 4639
      THEN POKE 103,0:GOTO 61090
61060 FA=500 708
61070 FOR I=1 TO LEN (FI#):POKE FA+I-1,ASC ( 3750
      MID# (FI#,I,1)):NEXT
61080 POKE 103,LEN (FI#):POKE 100,INT (FA/ 4573
      256):POKE 107,FA-PEEK (100)*256
61090 POKE 105,1:POKE 106,DV 1963
61100 INPUT "STARTADRESSE":AN:AH=INT (AN/256): 4929
      AL=AN-AH*256
61110 POKE 701,AL:POKE 702,AH:POKE 700,0 2629
61120 SYS 62622 748
  
```

achten, daß der Variablenstart-Pointer des Basic-Interpreters mit der Adresse des letzten geladenen Bytes belegt wird. Vorhandene Basic-Programme werden unbrauchbar gemacht. Dies wird durch BLOCK-LOAD verhindert.

Bei Aufruf aus einem Basic-Programm erscheinen keine Meldungen des Betriebssystems, wie „saving“, beziehungsweise „found“. Beim Saven und Laden von Maschinenfiles sind drei Dinge zu beachten:

Verbotene Zonen für

Maschinensprache

1. Die Routine ist nicht verwendbar, wenn das Maschinenprogramm am Basicstart liegt. Es sei denn, man verlegt den Basicstart.
2. Ein Programm, das im Kassettenpuffer liegt, kann nicht auf Kasette abgespeichert werden.
3. Programme, die in den Bildschirmspeicher hineinreichen, werden von eventuellen Bildschirmhalten überschrieben.

(T. Kretschmer)

Speicherbelegung auf einen Blick

Will der C-64-Anwender wissen, wieviel Speicherplatz noch zur Verfügung steht, so benutzt er den Befehl. FRE (0). Ist der freie Speicherbereich größer als 32767, erhält man einen negativen Wert. Man muß diesen Wert von 65536 abziehen, um den freien Speicherplatz zu ermitteln. Das ist eine unangenehme Eigenart des Basic-Interpreters. Das Programm „Speicherbelegung“ informiert nicht nur darüber, wieviel Bytes noch frei sind, sondern gibt an, wieviel Platz ein Programm oder die Variablen belegen.

In der Basic-Erweiterung „Exbasic-Level II“ gibt es den Befehl MEM. Dieses Programm wirkt wie der MEM Befehl. Hat man fehler-

Information über Bewegungen im Speicher helfen beim Optimieren von Programmen. Eine Subroutine in Maschinensprache holt die aktuelle Speicherbelegung auf den Bildschirm.

frei eingetippt und vorsichtshalber abgespeichert, so tippt man SYS 49223 ein und drückt die Return-Taste. Auf dem Bildschirm erscheint dann:

```

SPEICHER: XXX BYTES
PROGRAMM: XXX BYTES
VARIABLEN: XXX BYTES
ARRAYS: XXX BYTES
STRINGS: XXX BYTES
FREI: XXX BYTES
  
```

Anstelle des Zeichens X stehen Zahlen, die sagen wieviel Bytes die einzelnen Bereiche belegen. Bild 1 zeigt wie der Basic-Speicher aufgeteilt ist. Die Grenzen der einzelnen Bereiche stehen in zwei aufeinanderfolgenden Speicherstellen als 16-Bit-Zahl. Will man

Vier Tasten – ein Dutzend Funktionen

Nichts ist lästiger beim Arbeiten am Computer als das ewige Eintippen von immer gleichen Texten und Befehlen. Es gibt einen einfachen Weg, diese Arbeit zu erleichtern. Mit diesem Maschinenprogramm können die Funktionstasten des Commodore 64 mit zwölf Texten zu je 20 Zeichen belegt werden.

Nachdem mit dem Basic-Lader der Maschinencode an die entsprechenden Speicherstellen gebracht worden ist, kann das Programm mit SYS 50000 aktiviert werden. Nun kann man mit '@A, "TEXT"' eine Taste belegen. A kann eine Zahl oder Variable mit dem Wert 1 bis 12 sein. Der Wert 1,3,5 und 7 entspricht den ungeschifteten Funktionstasten. 2,4,6 und 8 erreicht man mit der SHIFT-Taste, und 9 bis 12 entsprechen den Funktionstasten in Verbindung mit

Die Belegung der Funktionstasten ist ein professionelles Mittel, Programme attraktiver zu machen. Eine Routine zaubert zwölf Texte auf die vier Tasten.

der Commodore-Taste. Bei einem Wert größer 12 kommt die Fehlermeldung 'ILLEGAL QUANTITY'. Diese Fehlermeldung sieht man auch, wenn der Text, mit dem die Taste belegt werden soll, mehr als 20 Zeichen hat. Der Text kann ein

String oder eine Stringvariable sein. Soll dieser Text mit 'RETURN' abgeschlossen sein, so ist als letztes Zeichen die 'Pfeil nach links'-Taste einzugeben. Die Belegung kann auch innerhalb eines Programms erfolgen. So können die Funktionstasten gleich in dem Basic-Lader aktiviert werden, wie in Zeile 147. Will man wissen, wie die Tasten belegt sind, so gibt man ein '@0'. Nun wird die Belegung der Funktionstasten aufgelistet. Wird die RUN/STOP- und RESTORE-Taste betätigt, so funktioniert das Programm nicht mehr. Mit 'SYS50020' kann die alte Belegung wieder hergestellt werden. Eine Aktivierung mit 'SYS 50000' löscht sämtliche Tastenbelegungen. Eine einzelne Taste kann man löschen, wenn man einen Leerstring eingibt. Zum Beispiel: '@1, ""'.

| | | | | | |
|--|------|--|------|---|------|
| 100 DATA 162,13,160,0, 169,0,153,160,196 | 3255 | 240,3,76,49,234 | | 133 DATA 247,32,205,189, 169,61,32,210 | 2924 |
| 101 DATA 202,240,9,152, 24,105,20,163 | 2923 | 117 DATA 165,203,201,4, 208,5,160,1,76 | 3384 | 134 DATA 255,169,34,32, 210,255,166,248 | 3184 |
| 102 DATA 76,84,195,120, 169,217,141,20 | 3037 | 118 DATA 9,196,201,5, 208,5,160,3,76,9 | 3080 | 135 DATA 189,168,196, 240,6,32,210,255 | 3432 |
| 103 DATA 3,169,195,141, 21,3,169,123,141 | 2898 | 119 DATA 196,201,6,208, 5,160,5,76,9,196 | 3389 | 136 DATA 232,208,245, 169,34,32,210,255 | 3257 |
| 104 DATA 8,3,169,195, 141,9,3,88,96,32 | 3290 | 120 DATA 201,3,208,5, 160,7,76,9,196,76 | 3430 | 137 DATA 169,13,32,210, 255,230,247,165 | 2950 |
| 105 DATA 115,0,201,64, 240,6,32,121,0 | 2881 | 121 DATA 49,234,173,141, 2,201,1,208,4 | 2694 | 138 DATA 247,201,13,208, 3,76,174,167 | 3114 |
| 106 DATA 76,231,167,32, 115,0,32,158,183 | 3171 | 122 DATA 200,76,33,196, 201,2,208,9,152 | 3381 | 139 DATA 165,248,24,105, 20,133,248,76 | 2764 |
| 107 DATA 224,0,208,3,76, 86,196,224,13 | 3280 | 123 DATA 56,233,1,74,24, 105,9,168,132 | 3095 | 140 DATA 107,196 | 943 |
| 108 DATA 176,61,169,0, 202,240,6,24,105 | 3332 | 124 DATA 247,32,135,234, 165,203,201,64 | 3119 | 141 S=0:FOR I=50000 TO 50343:READ D | 2350 |
| 109 DATA 20,76,155,195, 168,132,251,32 | 3095 | 125 DATA 208,247,166, 247,169,0,202,240 | 3352 | 142 POKE I,D:S=S+D:NEXT | 1675 |
| 110 DATA 253,174,32,158, 173,32,163,182 | 3137 | 126 DATA 6,24,105,20,76, 48,196,168,185 | 3678 | 143 PRINT "DIE DATAZEILEN SIND "; | 2057 |
| 111 DATA 201,21,176,34, 133,252,134,247 | 3305 | 127 DATA 168,196,240,20, 201,95,240,7 | 3051 | 144 IF S<>43735 THEN PRINT "FEHLERHAFT": STOP | 2577 |
| 112 DATA 132,248,166, 251,160,0,165,252 | 3395 | 128 DATA 32,210,255,200, 76,58,196,169 | 2843 | 145 PRINT "IN ORDNUNG ": SYS 50000 | 1916 |
| 113 DATA 240,12,177,247, 157,168,196,232 | 3312 | 129 DATA 1,133,198,169, 13,141,119,2,76 | 3272 | 146 : | |
| 114 DATA 200,198,252,76, 190,195,169,0 | 3072 | 130 DATA 48,234,32,121, 0,240,3,76,8,175 | 0030 | 147 @1,"LIST+":@3,"RUN+":@5,"DATA ":@7,"LOAD" | 3727 |
| 115 DATA 157,168,196,76, 174,167,76,72 | 3628 | 131 DATA 169,0,133,248, 162,1,134,247 | 3186 | 148 @0:REM FUNKTIONSTAST ENBELEGUNG ANZEIGEN | 392 |
| 116 DATA 178,165,204, 3255 | 3384 | 132 DATA 169,13,32,210, 255,169,0,166 | 2817 | | |

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM | | | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|--------|------|----|----|----|----|----|----|----|----|----|--|
| C350 | A2 | 0D | A0 | 00 | A9 | 00 | 99 | A8 | CD | C3F8 | A0 | 05 | 4C | 09 | C4 | C9 | 03 | D0 | 0C | |
| C358 | C4 | CA | F0 | 08 | 98 | 18 | 69 | 14 | B6 | C400 | 05 | A0 | 07 | 4C | 09 | C4 | 4C | 31 | 82 | |
| C360 | A8 | 4C | 54 | C3 | 78 | A9 | D9 | 8D | 08 | C408 | EA | AD | 8D | 02 | C9 | 01 | D0 | 04 | 5A | |
| C368 | 14 | 03 | A9 | C3 | 8D | 15 | 03 | A9 | C9 | C410 | C8 | 4C | 21 | C4 | C9 | 02 | D0 | 09 | 66 | |
| C370 | 7B | 8D | 08 | 03 | A9 | C3 | 8D | 09 | 5D | C418 | 98 | 38 | E9 | 01 | 4A | 18 | 69 | 09 | D9 | |
| C378 | 03 | 58 | 60 | 20 | 73 | 00 | C9 | 40 | E6 | C420 | A8 | 84 | F7 | 20 | 87 | EA | A5 | CB | 71 | |
| C380 | F0 | 06 | 20 | 79 | 00 | 4C | E7 | A7 | 84 | C428 | C9 | 40 | D0 | F7 | A6 | F7 | A9 | 00 | BA | |
| C388 | 20 | 73 | 00 | 20 | 9E | B7 | E0 | 00 | 54 | C430 | CA | F0 | 06 | 18 | 69 | 14 | 4C | 30 | 39 | |
| C390 | D0 | 03 | 4C | 56 | C4 | E0 | 0D | B0 | 04 | C438 | C4 | A8 | B9 | A8 | C4 | F0 | 14 | C9 | 4C | |
| C398 | 3D | A9 | 00 | CA | F0 | 06 | 18 | 69 | 64 | C440 | 5F | F0 | 07 | 20 | D2 | FF | C8 | 4C | A6 | |
| C3A0 | 14 | 4C | 9B | C3 | A8 | 84 | FB | 20 | 9F | C448 | 3A | C4 | A9 | 01 | 85 | C6 | A9 | 0D | 79 | |
| C3A8 | FD | AE | 20 | 9E | AD | 20 | A3 | B6 | 4A | C450 | 8D | 77 | 02 | 4C | 31 | EA | 20 | 79 | BB | |
| C3B0 | C9 | 15 | B0 | 22 | 85 | FC | 86 | F7 | 59 | C458 | 00 | F0 | 03 | 4C | 08 | AF | A9 | 00 | CB | |
| C3B8 | 84 | F8 | A6 | FB | A0 | 00 | A5 | FC | 80 | C460 | 85 | F8 | A2 | 01 | 86 | F7 | A9 | 0D | 59 | |
| C3C0 | F0 | 0C | B1 | F7 | 9D | A8 | C4 | E8 | 6E | C468 | 20 | D2 | FF | A9 | 00 | A6 | F7 | 20 | D6 | |
| C3C8 | C8 | C6 | FC | 4C | BE | C3 | A9 | 00 | 55 | C470 | CD | BD | A9 | 3D | 20 | D2 | FF | A9 | 20 | |
| C3D0 | 9D | A8 | C4 | 4C | AE | A7 | 4C | 48 | 78 | C478 | 22 | 20 | D2 | FF | A6 | F8 | BD | A8 | 98 | |
| C3D8 | B2 | A5 | CC | F0 | 03 | 4C | 31 | EA | E8 | C480 | C4 | F0 | 06 | 20 | D2 | FF | E8 | D0 | A5 | |
| C3E0 | A5 | CB | C9 | 04 | D0 | 05 | A0 | 01 | 07 | C488 | F5 | A9 | 22 | 20 | D2 | FF | A9 | 0D | BD | |
| C3E8 | 4C | 09 | C4 | C9 | 05 | D0 | 05 | A0 | D9 | C490 | 20 | D2 | FF | E6 | F7 | A5 | F7 | C9 | E7 | |
| C3F0 | 03 | 4C | 09 | C4 | C9 | 06 | D0 | 05 | 8A | C498 | 0D | D0 | 03 | 4C | AE | A7 | A5 | F8 | A9 | |
| | | | | | | | | | | C4A0 | 18 | 69 | 14 | 85 | F8 | 4C | 6B | C4 | 91 | |

(Herbert Kunz)

Interpreter überlisten

Ein Befehl, den man in fast jeder Basicerweiterung findet, ist DELETE zum Löschen ganzer Programmabschnitte. Doch der C64 kann's auch so! Man fügt als erste Programmzeile ein:

```
1 PRINT "[CLR,3DOWN]"F:
PRINT "F="F+1":L="L":
IF L=L THEN [HOME]":POKE
198,2:POKE 631,13:POKE
632,13:END
```

Diese Zeile muß genauso eingetippt werden; die Symbole in Klammern bedeuten die entsprechenden Cursorsteuerzeichen.

Dann tippt man im Direktmodus:
F=(Nummer der ersten Zeile);
L=(Nummer der letzten Zeile);
GOTO 1

Der Rechner löscht nun alle Zeilen von F bis L. F=100:L=150:GOTO1 löscht beispielsweise alle Zeilen von 100 bis einschließlich 150.

Die Schwachstelle des C64 sei sein Basic, sagen die einen. Die anderen sehen genau darin seine Stärke. Tips, wie man von Fall zu Fall den Basic-Interpreter austrickst, können Fans und Kritiker gleichermaßen gebrauchen.

Auch die Funktion CENTRE, bekannt aus SIMON'S BASIC, zur zentrierten Ausgabe von Text läßt

sich leicht simulieren: Statt CENTRE A\$ muß man in Standard Basic schreiben:
PRINT TAB((40-LEN(A\$))/2);A\$.

Sinnvoller

Variablenname

Um Programme selbstdokumentierend zu schreiben, ist es oft nützlich, Variablen einen sinnvollen Namen zu geben. Versucht man aber so etwas wie '10 TOP=100:BOTTOM=12', so bekommt man als Resultat den bekannten ?SYNTAX ERROR vom Interpreter, da das Schlüsselwort 'TO' in den Variablen vorkommt. Doch das läßt sich beheben, wenn man die Variablenamen wie folgt eingibt:

```
10 T(shift J)OP=100:BOTT(shift J)OM=12
```

Das geschiftete J bleibt beim Listen unsichtbar, verhindert aber die Fehlermeldung. Trotzdem beachtet der Interpreter nur die ersten beiden Zeichen des Variablennamens.

Wer hat nicht schon einmal Zeilen mit mehr als 80 Zeichen gebraucht? Eingegeben sind sie schnell mit Abkürzungen; zum Beispiel „?“ anstatt PRINT.

Die Basic Zeile darf

nicht länger als 80

Zeichen sein?!

Aber wehe, wenn ein Fehler in so einer überlangen Zeile auftaucht! Sie kann nicht mehr editiert werden. Alle Zeichen nach dem achtzigsten werden vom Interpreter ignoriert. Diese Malheur läßt sich einfach ausbügeln: Sobald man die Zeile zum ersten Mal eingetippt hat, fährt man mit dem Cursor auf die Zeilennummer und ändert sie in eine Nummer, die nicht

im Programm vorkommt. Diese Zeile sollte außerdem am Ende des Programms stehen und niemals angesprungen werden. Dann fügt man als erstes Zeichen ein Anführungszeichen in die Zeile ein. Der Interpreter behandelt jetzt diese Zeile als Text, das heißt, beim Listen werden Abkürzungen nicht „ausgeschrieben“. Will man nun die Originalzeile ändern, fährt man mit dem Cursor in die Textzeile, ändert die Zeilennummer und löscht das Anführungszeichen. Danach kann die Zeile wie gewohnt editiert werden.

Mittlerweile gibt es mindestens drei Betriebssystemversionen für den C 64. Die älteren haben einen interessanten Fehler im Editor. Gibt man in der untersten Bildschirmzeile mehr als 80 Zeichen ein und versucht dann, die letzten Zeichen mit DEL zu löschen, hängt sich der Rechner auf. Die Reaktionen sind dabei recht unterschiedlich. Das Problem läßt sich vermeiden, wenn man mit folgenden Cursorfarben arbeitet: black,

white, purple, green, orange, brown, grey2, light green. Wie gesagt, das Problem tritt nur bei Rechnern mit älteren Betriebssystemversionen auf.

Reset mit

Was aber, wenn sich der Rechner schon „verabschiedet“ hat, und man sein gerade eingetipptes Programm nicht verlieren will? Auch dazu einige Tips:

—(shift 3) drücken. Wenn die Nachricht 'PRESS PLAY ON TAPE' erscheint, PLAY an der Datensette drücken und den Rechner mit RUN/STOP zurückholen:

—Wer keine Datensette hat, kann einen Joystick in Port 1 solange bearbeiten, bis ebenfalls die 'PRESS PLAY..' -Nachricht erscheint. —Letzter Ausweg ist ein RESET-Taster. Nach dem Reset läßt man ein UNNEW-Programm und erweckt das eingetippte Programm wieder aus seinem Dornröschenschlaf.

(sis)

Tipphilfe für Datas

Wenn man die Ausgabe für eine Hardwareerweiterung scheut, aber doch nicht auf den Komfort einer Zehnertastatur (die Zahlen sind wie auf einem Taschenrechner angeordnet) verzichten will, gibt es die Möglichkeit der softwareseitigen Tastaturbelegung. Die Commodoretastatur wird mit Zahlen statt mit Buchstaben belegt (siehe Abbildung). Als Besonder-

Datazeilen sind fehlerträchtig. Wer die Quote reduzieren will, kann das mit einer Software-Zehnertastatur.

heit hat Autor Ralf Kellerbauer auf die Taste der „6“ den Befehl DATA gelegt.

Hex und Data

Programmbeschreibung: Das Programm ist ein Maschinenprogramm im Bereich des Kassettenpuffers. Mit der Tastenkombina-

| | | | | | |
|-----------------------|------|-----------------------|------|-----------------------|------|
| 111 NR=112 | 670 | 118 DATA 174,141,002, | 3691 | 200,003,240,005,202 | |
| 112 DATA 120,173,143, | 3654 | 224,005,208,021,236 | | 125 DATA 016,248,048, | 3532 |
| 002,174,144,002,236 | | 119 DATA 242,003,240, | 3785 | 040,189,216,003,224 | |
| 113 DATA 238,003,240, | 3642 | 085,173,241,003,073 | | 126 DATA 001,144,015, | 3336 |
| 014,141,239,003,142 | | 120 DATA 128,141,241, | 3904 | 208,031,236,141,002 | |
| 114 DATA 240,003,173, | 3925 | 003,173,032,208,073 | | 127 DATA 140,141,002, | 3696 |
| 237,003,174,238,003 | | 121 DATA 007,141,032, | 3666 | 240,002,169,064,076 | |
| 115 DATA 208,006,173, | 3788 | 208,142,242,003,173 | | 128 DATA 192,003,204, | 3596 |
| 239,003,174,240,003 | | 122 DATA 241,003,016, | 3836 | 243,003,208,016,185 | |
| 116 DATA 141,143,002, | 3329 | 061,165,203,162,015 | | 129 DATA 14454 | 815 |
| 142,144,002,088,096 | | 123 DATA 9773 | 701 | 130 DATA 232,003,032, | 3562 |
| 117 DATA 4889 | 799 | 124 DATA 160,000,221, | 3179 | 053,235,200,192,005 | |

| | | | | | |
|--|------|--|------|--|-------------|
| 131 DATA 208,245,169, 064,140,243,003,133 | 3956 | 138 DATA 065,100,003, 072,235,000,000,000 | 3245 | 147 :READ Q 148 IF NOT S=Q THEN PRINT "FEHLER IN ZEILE"NR"- "NR+5:END | 142 3293 |
| 132 DATA 203,108,239, 003,019,055,048,036 | 3702 | 139 DATA 000,000,072, 235,000,000,008,000 | 3369 | 149 :NR=NR+6 | 935 |
| 133 DATA 034,037,042, 030,033,038,035,045 | 3881 | 140 DATA 000,000,000, 000,000,000,000,000 | 2875 | 150 NEXT | 130 |
| 134 DATA 050,041,046, 040,064,001,055,035 | 3655 | 141 DATA 19267 | 931 | 151 PRINT :PRINT "ALLES OK !" | 1198 |
| 135 DATA 17948 | 724 | 142 FOR A=020 TO 1027 STEP 40 | 1407 | 152 PRINT :PRINT "MIT [2SPACES]SYS 828 [2SPACES]STARTEN" | 2066 |
| 136 DATA 056,059,008, 011,016,019,014,010 | 3744 | 143 :FOR B=0 TO 39 | 730 | | |
| 137 DATA 028,020,018, 021,032,068,065,084 | 3628 | 144 :READ Q:S=S+Q | 868 | | |
| | | 145 :POKE A+B,Q | 448 | | |
| | | 146 :NEXT | 63 | | |

00 01 02 03 04 05 06 07 CHKSUM

```

033C 78 AD 8F 02 AE 90 02 EC 8C
0344 EE 03 F0 0E 8D EF 03 8E DF
034C F0 03 AD ED 03 AE EE 03 ED
0354 D0 06 AD EF 03 AE F0 03 CA
035C 8D 8F 02 8E 90 02 58 60 F8
0364 AE 8D 02 E0 05 D0 15 EC A3
036C F2 03 F0 55 AD F1 03 49 FF
0374 90 8D F1 03 AD 20 D0 49 F3
037C 07 8D 20 D0 8E F2 03 AD BE
0384 F1 03 10 3D A5 CB A2 0F 2D
038C A0 00 DD C8 03 F0 05 CA 04
0394 10 F8 30 28 B0 D8 03 E0 29
    
```

```

039C 01 90 0F D0 1F EC 8D 02 70
03A4 8C 8D 02 F0 02 A9 40 4C 60
03AC C0 03 CC F3 03 D0 10 B9 21
03B4 E8 03 20 35 EB C8 C0 05 AD
03BC D0 F5 A9 40 8C F3 03 85 5C
03C4 CB 6C EF 03 13 37 30 24 32
03CC 22 25 2A 1E 21 26 23 2D FA
03D4 32 29 2E 28 40 01 37 23 62
03DC 38 38 08 0B 10 13 0E 0A FB
03E4 1C 14 12 15 20 44 41 54 CE
03EC 41 64 03 48 EB 00 00 00 83
03F4 00 00 48 EB 00 00 08 00 66
03FC 00 00 00 00 00 00 00 00
    
```

```

1100: 033c      * = $033c ; dez 828
        ;
        ; variable/konstanten
1140: 033c      level  = $028d ; speicher fuer 'ctrl'-'shift'-'cbm'
1150: 033c      taste  = $cb   ; tastaturmatrix wert
1160: 033c      rand   = $d020 ; farbe des bildschirmrandes
1170: 033c      vektor = $028f ; tastaturvektor
        ;
        ; austausch des tastaturvektors
1260: 033c 78      start  sei
1290: 033d ad 8f 02      lda  vektor ; alter wert low
1300: 0340 ae 90 02      ldx  vektor+1 ; ' ' high
1330: 0343 ec ae 03      cpx  new+1
1340: 0346 f0 0e      beq  off ; routine war eingeschaltet !
1350: 0348 8d ef 03 on  sta  old ; uebernehmen low
1360: 034b 8e f0 03      stx  old+1 ; ' ' high
1370: 034e ad ed 03      lda  new ; neuer wert low
1380: 0351 ae ee 03      ldx  new+1 ; ' ' high
1390: 0354 d0 06      bne  ready ; 'jmp ready' 0 nur in zero page
1410: 0356 ad ef 03 off  lda  old ; alter wert low
1420: 0359 ae f0 03      ldx  old+1 ; neuer wert high
1440: 035c 8d 8f 02 ready sta  vektor ; tastaturvektor low
1450: 035f 8e 90 02      stx  vektor+1 ; ' ' high
1460: 0362 58      cli
1470: 0363 60      rts
        ;
        ; hauptprogramm
1600: 0364 ae 8d 02 mainprg ldx  level ; " 'ctrl' ?"
1610: 0367 e0 05      cpx  #$05 ; ' '
1620: 0369 d0 15      bne  weiter ; keine umschaltung aktiv-standby
        ;
        ; umschaltung zwischen 'aktiv' und 'standby'
1660: 036b ec f2 03      cpx  repeat ; "taste beim letzten irq schon
        gedrueckt ?"
    
```

tion „SHIFT-CTRL“ wird auf eine Tastatur mit 10er Tastenblock umgeschaltet, in einer Belegung, wie sie einige Rechner (zum Beispiel Epson) bereits haben. Ist die 10er Tastaturbelegung eingeschaltet, so wird die Randfarbe als Einschaltkontrolle verändert. Zusätzlich sind die Buchstaben A–F direkt neben dem Zahlenblock, sowie ein Generator für das Wort „DATA“ auf der Taste „6“ vorhanden. Das Programm ist als Basic-Lader und als Assemblerlisting vorhanden. Es wird mit SYS 828 gestartet und mit SYS 828 oder RUNSTOP-RESTORE gestoppt.

Schneller trotz

schräger Lage

Praxis: Etwas gewöhnungsbedürftig ist die schräge Lage des Tastaturblockes, was jedoch nach kurzer Zeit keine Schwierigkeiten mehr macht. Tippfehler werden

```

1670: 036e f0 55      beq end      ; ja d.h ignorieren
1680: 0370 ad f1 03    lda flag    ; ein-aus flag
1690: 0373 49 80      eor #$80   ; flag drehen
1700: 0375 8d f1 03    sta flag    ; und speichern
1710: 0378 ad 20 d0    lda rand    ; randfarbe laden
1720: 037b 49 07      eor #$07   ; und drehen
1730: 037d 8d 20 d0    sta rand    ; (dient als einschaltkontrolle)
; "austausch des matrixwertes noetig ?"
1760: 0380 8e f2 03    weiter stx repeat ; '0'=kein shift-ctrl , '5'=shift-ctrl
1770: 0383 ad f1 03    lda flag    ; prueft auf 'ein'
1780: 0386 10 3d      bpl end     ; falls standby
;
1800: 0388 a5 cb      lda taste   ; matrixwert
1805: 038a a2 0f      ldx #15    ; 16 werte
1810: 038c a0 00      ldy #0     ; s.u.
1820: 038e dd c8 03    loop  cmp tab1,x ; "ist die gedruckte taste dabei ?"
1830: 0391 f0 05      beq cont   ; ja
1840: 0393 ca      dex       ; naechste taste
1850: 0394 10 f8      bpl loop   ; naechster tabellenwert
1860: 0396 30 28      bmi secure ; 'jmp secure' (normale
; tastaturroutine)
; austausch
1880: 0398 bd d8 03    cont  lda tab2,x ; austauschwert
1890: 039b e0 01      cpx #01   ; "" / ""
1900: 039d 90 0f      bcc six   ; "" 6 ""
1910: 039f d0 1f      bne secure ; alle ausser 0,1
;
1950: 03a1 ec 8d 02    return cpx level ; x=1 " 'shift' ?"
1955: 03a4 8c 8d 02    sty level ; 'cbm' oder 'shift' loeschen
1960: 03a7 f0 02      beq ret   ; 'shift' /
1970: 03a9 a9 40      lda #64   ; nur /
1990: 03ab 4c c0 03    ret      jmp secure ; tastewert sichern
;
2010: 03ae cc f3 03    six    cpy datamem ; (y-reg = 0)
2016: 03b1 d0 10      bne tas
2020: 03b3 b9 e8 03    write lda text,y ; " data"
2030: 03b6 20 35 eb    jsr $eb35 ; auf tastaturpuffer legen
2040: 03b9 c8      lry
2050: 03ba c0 05      cpy #5    ; " 5 zeichen ?"
2060: 03bc d0 f5      bne write ; weitere zeichen
2070: 03be a9 40      lda #64   ; = keine taste gedrueckt
;
2110: 03c0 8c f3 03    secure sty datamem ; zaehler fuer data
2115: 03c3 85 cb      tas     sta taste ; austauschwert speichern
2120: 03c5 6c ef 03    end     jmp (old) ; normal in der decoding weiter
;
; tabelle der tastencodes
;
2160: 03c8 13 37 30    tab1   .byte19,55,48 ; 6,/ ,pfund
2170: 03cb 24 22 25    .byte36,34,37,42,30,33,38,35 ; n,jkluo
2175: 03cd 2d 32 29    .byte45,50,41,46,40 ; "" ;p'affe'+""
2180: 03d8 40 01 37    tab2   .byte64,01,55 ; nop,return,/
2190: 03db 23 38 3b    .byte35,56,59, 8,11,16,19,14 ; 0123456e
2195: 03e3 0a 1c 14    .byte10,28,20,18,21 ; abcdf
2300: 03e8 20 44 41    text   .asc " data"
;
; arbeitsspeicher der routine
;
3510: 03ed 64 03    new    .wordmainprg ; hauptprogramm
3520: 03ef 48 eb    old    .word$eb48 ; preset basic v2 tastaturvektor
3530: 03f1 00      flag   .byte$00
3540: 03f2 00      repeat .byte$00
3550: 03f3 00      datamem .byte$00
;

```

eine Taste gedrückt ist, verlangsamen sich die Programme nicht. Bei der Entwicklung des Programmes war es deshalb notwendig, bei der Erzeugung des Wortes „DATA“ recht aufwendig vorzugehen, um einen Repeat zu unterdrücken.

Zusammenarbeit mit anderen Programmen

Dazu dient der Zähler DATAMEM in Adresse \$03F3. Er wird normalerweise bei jedem Durchlauf der Routine auf Null gesetzt. Wenn aber die Taste „6“ gedrückt wird, so wird in DATAMEM nach Ausgabe des Textes „DATA“ der

| | | | | | | | |
|---|---|---|---|---|-------|---|---|
| 6 | 7 | 8 | 9 | 0 | + | - | £ |
| Y | U | I | O | P | @ | * | |
| H | J | K | L | : | ; | = | |
| N | M | , | . | / | SHIFT | | |

Originalbelegung der C64-Tastatur

| | | | | | | | |
|---|---|---|---|--------|---|---|---|
| d | 7 | 8 | 9 | E | F | - | / |
| Y | 4 | 5 | 6 | C | D | | |
| H | 1 | 2 | 3 | A | B | - | |
| N | 0 | , | . | RETURN | | | |

10er-Tastatur Belegung der C64-Tastatur

seltener und die Eingabegeschwindigkeit ist wesentlich höher als mit der Originaltastaturbelegung. Anpassung an VC20: (vgl. Assembler-Listing) Dazu müßte es ausreichen, in der Routine „JSR \$EB35“ (Daten auf Tastaturpuffer legen) die Adresse zu ändern und gegebenenfalls die Matrixcode-Werte zu modifizieren. Eine Anpassung der Zellen \$03EF-\$03F0 ist nicht unbedingt nötig. Funktionsprinzip: Der Matrixwert der Tastatur wird mit einer Tabel-

le verglichen, in der die Matrixwerte der zu modifizierenden Tasten stehen. Ist die Taste zu modifizieren, wird der Matrixwert ausgetauscht. Um beispielsweise der Taste „M“ den Wert 0 zuzuordnen, wird der Wert 36 (Taste M) gegen 35 (Taste 0) ausgetauscht. Bei der „DATA“ Funktion werden die Buchstaben des Wortes „DATA“ als Ascii-Zeichen auf den Tastaturpuffer gelegt. Besonderheiten: Da die Routine nur dann bearbeitet wird, wenn

Wert Fünf untergebracht. Wird beim nächsten Interrupt wieder die Taste „6“ registriert, so erkennt das Programm an diesem Wert, daß der Text schon ausgegeben wurde.

Nach RUN/STOP-RESTORE muß das Programm wieder mit SYS 828 aktiviert werden.

Die Zusammenarbeit mit Basic V2, Simon's Basic, Datamat, Textomat, Ass 64/Mon 64 ist möglich.

(Ralf Kellerbauer)

Komfort-Listing

Wenn man ein langes Listing in Etappen sehen will, bietet der Commodore den Befehl LIST A-B an. Dabei ist A die erste und B die letzte Zeilennummer, die gelistet werden sollen. Nach jedem Einblick muß man jedoch die Abschnitte neu eingeben; eine sehr mühsame Angelegenheit. Viel komfortabler und professioneller löst das die LIST-Utility: Der Befehl wird wie gewohnt eingegeben; erst nach Drücken von f1 beginnt der Listvorgang. Bei Loslassen der Taste bleibt das Listing stehen. Sollen Korrekturen vorgenommen werden, stoppt man mit der STOP-Taste. Um an der Stelle fortzufahren, an der man vorher aufgehört hat, muß man wieder LIST und RETURN

Die List-Routine des Commodore-Betriebssystems ist unbefriedigend: Ohne Halt läuft ein Listing über den Bildschirm. Wer es stoppen will, braucht die List-Utility.

eingeben: danach kurz f7 und wieder f1. Für die Interessierten der Programmablauf: Das Programm

kopiert zuerst das BASIC-ROM in den RAM-Bereich (eine auch sonst ganz nützliche schnelle Routine) und ändert eine Sprungadresse der normalen LIST-Routine.

Falls ein RUN/STOP + RESTORE gegeben wird, muß deshalb danach wieder der RAM-Bereich mit POKE 1,54 aktiviert werden. Zum Listingdrucken sollte man die Utility abschalten (z. B. durch RUN/STOP + RESTORE).

Das Programm selbst liegt im Kassettenpuffer, der bei Diskettenbetrieb ungenutzt ist. Datasette-Besitzer müßten das Programm an eine andere Stelle schieben, wobei ein JMP-Befehl und zwei Hilfsregister geändert werden müssen.

(Horst-Peter Wiesnet)

| KOPIER-ROUTINE | | | | LIST-UTILITY HAUPTPROGRAMM | | | |
|----------------|----------|----------------------|-------------------------------------|----------------------------|----------|------|---|
| C000 | A9 00 | LDA #00 | ;Adresse low | 033C | A5 C5 | LOOP | LDA \$C5 ;gedruckte Taste in Akku |
| C002 | 85 5C | STA \$5C | ;in Zeropage | 033E | 85 02 | | STA \$02 ;in 02 zwischenspeichern |
| C004 | A9 A0 | LDA #A0 | ;Adresse high | 0340 | 20 2C A8 | | JSR \$ABC2 ;STOP_Taste gedrueckt? |
| C006 | 85 5D | STA \$5D | ;in Zeropage | 0343 | A5 02 | | LDA \$02 |
| C008 | A2 20 | LDX #20 | ;Anzahl der zu kopierenden Blöcke | 0345 | C9 04 | | CMP #04 ;f1_Taste gedrueckt? |
| C00A | A0 00 | LDY #00 | | 0347 | F0 11 | | BEQ ENDE ;wenn ja weiter |
| C00C | B1 5C | NAECHST LDA (\$5C),Y | ;256 Bytes=1Block kopieren | 0349 | C9 03 | | CMP #03 ;f7_Taste gedrueckt? |
| C00E | C6 01 | DEC \$01 | ;auf RAM schalten | 034B | D0 EF | | BNE LOOP ;wenn nein, zurueck in Warteschleife |
| C010 | 91 5C | STA (\$5C),Y | | 034D | AD FB 03 | | LDA \$03FB ;wenn ja Adresse der zuletz gelisteten |
| C012 | E6 01 | INC \$01 | ;auf ROM schalten | 0350 | 85 60 | | STA \$60 ;zuletz gelisteten |
| C014 | 88 | DEY | | 0352 | AD FA 03 | | LDA \$03FA ;Zeile zurueckholen |
| C015 | D0 F5 | BNE NAECHST | | 0355 | 85 5F | | STA \$5F |
| C017 | E6 5D | NBLOCK INC \$5D | ;nächster Block | 0357 | 4C 3C 03 | | JMP LOOP |
| C019 | CA | DEX | | 035A | A5 60 | ENDE | LDA \$60 ;zuletz gelistete Zeile |
| C01A | 30 02 | BMI FERTIG | | 035C | 8D FB 03 | | STA \$03FB ;zwischenspeichern |
| C01C | D0 EE | BNE NAECHST | | 035F | A5 5F | | LDA \$5F |
| C01E | C6 01 | DEC \$01 | | 0361 | 8D FA 03 | | STA \$03FA |
| C020 | A9 3C | FERTIG LDA #3C | ;Sprungadresse in ;der LIST-Routine | 0364 | 60 | | RTS |
| C022 | 8D D2 A6 | STA \$A6D2 | ;ändern | | | | |
| C025 | A9 03 | LDA #03 | | | | | |
| C027 | 8D D3 A6 | STA \$A6D3 | | | | | |
| C02A | 60 | RTS | | | | | |

| | | | | | |
|-----|--|------|-----|--|------|
| 10 | SUM=0 | 458 | 140 | IF SUM<5580 THEN PRINT "FEHLER IN DATAS 2.TEIL!":END | 3439 |
| 20 | FOR I=828 TO 868 | 947 | 150 | SYS 49152 | 780 |
| 30 | READ X | 311 | 160 | DATA 165,197,133,2,32,44,169,165,2 | 3705 |
| 40 | SUM=SUM+X | 934 | 170 | DATA 201,4,240,17,201,3,208,239,173 | 3242 |
| 50 | POKE I,X | 667 | 180 | DATA 251,3,133,96,173,250,3,133,95 | 3667 |
| 60 | NEXT I | 276 | 190 | DATA 76,60,3,165,96,141,251,3,165 | 3086 |
| 70 | IF SUM<4876 THEN PRINT "FEHLER IN DATAS 1.TEIL!":END | 3228 | 200 | DATA 95,141,250,3,96 | 1539 |
| 80 | SUM=0 | 459 | 210 | REM ENDE 1.,ANFANG 2.TEIL DER DATAS | |
| 90 | FOR I=49152 TO 49194 | 1839 | 220 | DATA 169,0,133,92,169,160,133,93 | 3213 |
| 100 | READ X | 311 | 230 | DATA 162,32,160,0,177,92,199,1,145 | 3403 |
| 110 | SUM=SUM+X | 934 | 240 | DATA 92,230,1,136,208,245,230,93,202 | 3427 |
| 120 | POKE I,X | 667 | 250 | DATA 48,2,208,238,199,1,169,60,141 | 3203 |
| 130 | NEXT I | 276 | 260 | DATA 210,166,169,3,141,211,166,96 | 3303 |

Antiratter-Routine für Simon's Basic

In kommerzieller Software werden als Kopierschutz einige Sektoren auf der Diskette nicht formatiert. Wird nun der Lesekopf durch das Programm veranlaßt, einen nicht formatierten Sektor zu lesen, findet er ihn nicht und macht fünf Leseversuche, bevor das Programm mit der entsprechenden Fehlermeldung abgebrochen wird.

Im Programm wird nun die Fehlermeldung abgefragt. Ist sie vorhanden, dann wird fortgesetzt – wenn nicht, wird zu einer Selbstmordroutine verzweigt. Das ist bei Kopien der Fall, die mit einem normalen Kopierprogramm kopiert wurden.

Diese Art des Kopierschutzes ist sehr umstritten, weil bei der häufigen Verwendung solcher Programme die Mechanik der Floppy leidet.

Wen beim Laden von Simon's Basic das Rattern der Floppy stört, der kann mit einer Routine abhelfen.

Simon's Basic ist ein sehr oft angewendetes Programm und damit ein Floppykiller. Abhilfe zu schaffen, ist dringend nötig. Den Weg dazu zeigt das Studium der Memory-Map und des Rom-Listings im Floppy-Buch von Data Becker. In der Memory-Map steht in Adresse 106 die Zahl der Leseversuche. Ändert man diese Speicherstelle, so kann das Kopfrattern unterbunden werden. Für

das Formatieren einer Diskette ist diese Methode nicht wirksam. Hier rattert es nach wie vor, wodurch der Kopf weiterhin allmählich dejustiert wird.

```
10 OPEN 1,8,15
20 PRINT #1,"M-W";CHR$(106)
   CHR$(0)CHR$(1)CHR$(133)
30 CLOSE 1
```

Die Antiratter-Routine wurde für Simon's Basic entwickelt und damit ausprobiert. Theoretisch müßte sie aber auch bei anderer Software funktionieren, die nach dem gleichen Prinzip kopiergeschützt ist. In letzter Zeit kommen jedoch immer mehr Programme auf den Markt, die in das DOS (Disk Operating System) eingreifen. Hier funktioniert die Antiratter-Routine nicht.

(Rudolf Wolf)

Shorties im Sechserpack

Im ersten Programmabschnitt von **MIPs 1** wird der Vektor für Funktionen 'verstellt'. Der Hauptteil verändert die PEEK-Anweisung. Falls man beispielsweise mit einem Grafikprogramm arbeitet, kann man mit dem neuen PEEK-Befehl sehr leicht ein Hardcopy-Programm schreiben. Die Zeilen \$02CC bis \$02D1 prüfen jetzt auf den Befehl 'PEEK'; genau gesagt, auf den Token für PEEK. Wenn man eine Basic-Programmzeile eingibt, vergleicht der Computer die Zeile mit einer Liste von Basic-Anweisungen. Wird dabei eine bekannte 'Wortkombination' (wie PEEK) gefunden, wandelt er diese vier Bytes in eine Abkürzung (hier \$C2) um. Tritt kein PEEK-Befehl auf, dann wird – wie gehabt – der Basic-Zeiger (\$7A/7B) herab-

„MIPse“ sagt Heino Velder zu seinen Mini-Programmen. Die sechs Knirpse passen in jedes Programm und leisten dort gute Dienste.

gezählt, und es folgt die Fortsetzung in das ROM (JMP(\$02FA)). Der Zeiger ins Basic weist auf den PEEK-Befehl. JSR \$0073 erhöht den Zeiger um eins und JSR \$ABA3 holt die PEEK-Adresse aus dem Basic-Programm (Klammer-Ausdruck).

Das Ergebnis steht im FAC#1 (Floatingpoint-Akku Nr. 1). Die Arbeitsweise dieses 'Zwischenspeichers' zu erläutern, würde einige Seiten umfassen. Es genügt im Grunde zu wissen, daß sich der Computer hier Zahlen 'merkt'. JSR \$B7F7 wandelt FAC#1 nun in eine (handliche) Integerzahl um. Das Ergebnis erhält man im Y-Register (Low-Byte) und im Akku (High-Byte), sowie in den Adressen \$14/15. Der Lade-Befehl in Zeile \$02E6 holt den Wert der adressierten Speicherzelle. Das (PEEK-)Ergebnis wird ins Y-Register kopiert, und die Routine \$B3A2 überträgt den Y-Wert in den FAC#1. Dieses Unterprogramm führt außerdem ein 'RTS' aus. Alles weitere erfolgt in den ROMs. Lautet der Befehl bei-

MIPs 1) Beispiel zu Funktionen (PEEK unter's ROM).

```

..
.. 02A8 A9 CC LDA #0CC ;
.. 02AA A2 02 LDX #02 ; Wie korrigierte FRE-Funktion.
.. 02AC EC FB 02 CPX #030B ;
.. 02AC F0 1A BEQ #02CB ; Der Vektor zur Auswertung von
.. ;
.. 02B1 AC 0A 03 LDY #030A ; Funktionen wird umgeladen
.. 02B4 BC FA 02 STY #02FA ;
.. 02B7 AC 0B 03 LDY #030B ; und anschließend 'verbogen'.
.. 02BA BC FB 02 STY #02FB ;
.. 02BD D0 06 BNE #02C5 ;
.. ;
.. 02BF AD FA 02 LDA #02FA ; #02AB = Programm einschalten.
.. 02C2 AE FB 02 LDX #02FB ; #02BF = " ausschalten.
.. #02C5 BD 0A 03 STA #030A ;
.. 02CB BE 0B 03 STX #030B ;
.. #02CB 60 RTS ;
..
.. ----- Hauptprogramm -----
..
.. 02CC 20 73 00 JSR #0073 > nächstes Zeichen aus dem BASIC.
.. 02CF C9 B8 CMP #0C2 > Vergleicht mit Token 'PEEK'.
.. 02D1 D0 1C BNE #02EF > nein, dann zur ROM-Routine.
..
.. 02D3 20 73 00 JSR #0073 > Zeiger auf folgendes Zeichen.
.. 02D6 20 AB AD JSR #ADAB > Holt Adresse zu PEEK(XXXX).
.. 02D9 20 F7 B7 JSR #B7F7 > Zahl nach Integer umwandeln.
.. ; Ergebnis in $14/15 (low/high).
..
.. 02DC 7B SEI > Interrupt aus (siehe Text)
.. 02DD A5 01 LDA #01 ;
.. 02DF AA TAX ; ROM's abschalten. Wieder Ein-
.. 02E0 29 F8 AND #FB ; schalten durch (Original-)
.. 02E2 B5 01 STA #01 ; Wert im X-Register.
..
.. 02E4 A0 00 LDY #00 ; PEEK-Wert laden und in's
.. 02E6 B1 14 LDA ($14),Y ; Y-Register übertragen.
.. 02EB AB TAY ;
.. 02E9 B6 01 STX #01 > ROM's wieder einschalten.
.. 02EB 58 CLI > Interrupt wieder zulassen.
.. 02EC 4C A2 B3 JMP #B3A2 > Wert im Y-Register zum
.. ; Floatingpoint-ACCUM 1 (Ende).
..
.. #02EF A5 7A LDA #7A ;
.. 02F1 D0 02 BNE #02F5 ; Kein PEEK-Befehl gefunden.
.. 02F3 C6 7B DEC #7B ; Zeiger in's BASIC minus 1
.. #02F5 C6 7A DEC #7A ; und
..
.. 02F7 6C FA 02 JMP (#02FA) > weiter zur ROM-Routine.
..
.. 02FA ** ** < VEKTOR > > Fortsetzungs-Vektor.

```

spielsweise 'PRINT PEEK(X)', dann druckt der Computer nun den Wert im FAC#1. Die Übergabe eines numerischen Wertes aus der Arithmetik-Routine geschieht immer über den FAC#1. Zusätzlich schalten die Zeilen \$02DD bis 02E2 die ROMs ab. Diese sind ausgeblendet, falls die ersten beiden Bits der Steueradresse \$01 gelöscht sind. Da das Betriebssystem (und die Interrupt-Routine) ausgeschaltet ist, muß der Interrupt – mit SEI – blockiert werden.

Restore mit Zeilennummer

Der Computer unterscheidet zwischen Basic-Funktionen, bei denen (Strings ausgeklammert) 'gerechnet' wird, und solchen Anweisungen, die 'direkt' etwas ausführen; beispielsweise PRINT, GOTO usw.

Die letzteren Befehle holt der Mikroprozessor über den Vektor \$0308/9; also ein weiterer 'Angriffspunkt' für Maschinen-Programme.

MIPs2 erweitert den RESTORE-Befehl, so daß der Zeiger für den READ-Befehl 'gezielt' auf eine bestimmte Zeilennummer gerichtet werden kann. Diese Basic-Erweiterung eignet sich gut zum Erläutern des Grundprinzips.

SYS 680 überträgt den Basic-Vektor an die Adresse \$02FC/FD. Der Vektor wird nun auf die Adresse \$02C7 gestellt, und RESTORE XX ist eingeschaltet. Falls man jetzt erneut SYS 680 eingibt, kopiert das Programm den neuen Basic-Vektor (\$02C7) nach \$02FC/FD. Anschließend läuft der Computer vom Anfang des Hauptprogramms zu dem JMP (\$02FC). Dieser Sprung führt wieder an den Programmstart usw. ...

```

1 REM *****
2 REM * PEEK UNTER'S ROM (C64) *
3 REM *****
4 ;
5 REM EINSCHALTEN: SYS 680 / AUSSCHALTEN: SYS 703
6 ;
10 FOR I=680 TO 766:READ D:POKE I,D 2153
20 S=S+D:Z=Z+1:IF Z<10 THEN NEXT I:Z=-1 2934
30 READ D:IF D<>S THEN PRINT "FEHLER IN";L+1000:STOP 2921
40 L=L+1:IF Z>0 THEN Z=0:S=0:NEXT 2199
50 ;
60 ;
1000 DATA 169,204,162,002,236,011,003,240,026,172,1225 4976
1001 DATA 010,003,140,250,002,172,011,003,140,251,982 5037
1002 DATA 002,208,006,173,250,002,174,251,002,141,1209 5197
1003 DATA 010,003,142,011,003,096,032,115,000,201,613 4971
1004 DATA 194,208,029,032,115,000,032,138,173,032,952 5039
1005 DATA 247,183,120,165,001,170,041,248,133,001,1309 5226
1006 DATA 160,000,177,020,168,134,001,088,076,162,986 4939
1007 DATA 179,165,122,208,002,198,123,198,122,108,1425 4953
1008 DATA 250,002,000,000,000,000,000,252 3030

```

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|--------|
| 02A8 | A9 | CC | A2 | 02 | EC | 0B | 03 | F0 70 |
| 02B0 | 1A | AC | 0A | 03 | BC | FA | 02 | AC 5E |
| 02B8 | 0B | 03 | 0C | FB | 02 | D0 | 06 | AD 26 |
| 02C0 | FA | 02 | AE | FB | 02 | 8D | 0A | 03 27 |
| 02C8 | 9E | 0B | 03 | 60 | 20 | 73 | 00 | C9 0B |
| 02D0 | C2 | D0 | 1C | 20 | 73 | 00 | 20 | 8A F7 |
| 02D8 | AD | 20 | F7 | B7 | 78 | A5 | 01 | AA F7 |
| 02E0 | 29 | F8 | 05 | 01 | A0 | 00 | B1 | 14 F1 |
| 02E8 | A8 | 96 | 01 | 58 | 4C | A2 | B3 | A5 83 |
| 02F0 | 7A | D0 | 02 | C6 | 7B | C6 | 7A | 6C AA |
| 02F8 | FA | 02 | 00 | 00 | 00 | 00 | 00 | 0D EB |

MIPs 2) Beispiel erweiterter BASIC-Befehl (RESTORE XX).

```

..
..
.. 02AB A9 CC LDA #*CC
.. 02AA A2 02 LDX #*02 | Wie neuer 'FETCH-Befehl'.
..
.. 02AC AC 08 03 LDY #0308 | Achtung: Der Programmteil
.. 02AF BC FC 02 STY #02FC | prüft nicht, ob das Hauptpro-
.. 02B2 AC 09 03 LDY #0309 | gramm schon eingeschaltet ist.
.. 02B5 BC FD 02 STY #02FD | Zweimaliges Einschalten führt
.. 02B8 D0 06 BNE #02C0 | zum Programmabsturz..!
..
.. 02BA AD FC 02 LDA #02FC | Original-Vektor laden.
.. 02BD AE FD 02 LDX #02FD | (Befehlsweiterung aus)
.. 02C0 8D 08 03 STA #0308
.. 02C3 8E 09 03 STX #0309 | Vektor abspeichern und
.. 02C6 60 RTS | zurück in's BASIC.
..
----- Hauptprogramm -----
..
.. 02C7 20 73 00 JSR #0073 | Nächstes Zeichen aus BASIC.
.. 02CA C9 8C CMP #*8C | 'RESTORE-Befehl' ?
.. 02CC D0 24 BNE #02E2 | nein, dann weiter in's ROM.
..
.. 02CE 20 1D AB JSR #A81D | RESTORE-Routine ausführen.
.. 02D1 20 73 00 JSR #0073 | nächstes Zeichen holen.
.. 02D4 B0 19 BCS #02EE | Zahl, sonst Schluss.
..
.. 02D6 20 BA AD JSR #AD8A | Holt Zahl < 255 (RESTORE XXX).
.. 02D9 20 F7 B7 JSR #B7F7 | Nach Integer umwandeln.
.. 02DC 84 3F STY #3F | Ergebnis in Y/ACCU (low/high)
.. 02DE 85 40 STA #40 | nach READ-Zeiger (Zeilen-Nr.).
.. 02E0 20 13 A6 JSR #A613 | Sucht nach dieser Zeilen-Nr.
..
.. 02E3 A6 60 LDX #60 | Adresse der Zeile (oder der
.. 02E5 A2 5F LDY #5F | folgenden) in $5F/60.
.. 02E7 D0 01 BNE #02EA |
.. 02E9 CA DEX | Adresse minus 1 in 'genauen'
.. 02EA 8A DEY |
.. 02EB 84 41 STY #41 | READ-Zeiger (siehe Text).
.. 02ED 86 42 STX #42 |
.. 02EE 4C AE A7 JMP #A7AE | zum nächsten Befehl (Ende).
..
.. 02F2 A0 FF LDY #*FF |
.. 02F4 C6 7B DEC #7B | Zeiger in's BASIC minus eins.
.. 02F6 20 FB AB JSR #ABFB |
.. 02F9 6C FC 02 JMP (#02FC) | Fortsetzung über Vektor.
..
.. 02FC ** ** < VEKTOR > | Hier 'alter' Vektor.

```

JSR \$A81D führt den 'normalen' RESTORE durch. Dann holt CHRGET das (auf RESTORE) folgende Zeichen. Ist anschließend das Carry-Flag gelöscht, so wurde eine Zahl (Buchstaben 1-9) gefunden. Anderenfalls wird das Programm mit JMP \$A7AE verlassen. Mit dieser ROM-Routine geht's regulär weiter im Basic.

Ist das Carry-Flag nach CHRGET jedoch gelöscht, dann liegt ein erweiterter RESTORE-Befehl vor (z. B. RESTORE 1000). Die folgenden beiden Unterprogramme wurden schon beim PEEK-Befehl besprochen. JSR \$AD8A holt eine Zahl und \$B7F7 wandelt diese in einen Integer-Wert um. Diese 'Zeilennummer' wird in dem DATA-Zeiger \$3F/40 abgespeichert. Die Adresse benutzt der Computer nur bei Fehlermeldung, falls beispielsweise Zahlen gelesen werden (READ D), die DATA-Zeilen jedoch Strings enthalten (DATA "A", "B", ...). Außerdem steht die Zeilennummer in dem Zeiger \$14/15, und die folgende Routine \$A613 durchsucht das Basic-Programm nach dieser Nummer. Bei der Rückkehr befindet sich die Startadresse plus 1 dieser Basic-Zeile in dem Zeiger \$5F/60. Wird die Nummer nicht gefunden, dann zeigen \$5F/60 auf die folgenden Programmzeilen.

Die Adresse wird zuletzt um eins herabgezählt und in dem eigentlichen (genauen) READ-Zeiger abgespeichert. Mit JMP \$A7AE geht's dann zum nächsten Befehl. Dieses Programm erlaubt hinter RESTORE nur eine Zahlenangabe, jedoch keine Variable (CLC-Test nach CHRGET). Falls CHRGET auf das Ende einer Programmzeile trifft (100 RESTORE) oder einen Doppelpunkt findet (RESTORE: PRINT ...) ist das ZERO-FLAG gesetzt. Wird die Zeile \$02D4 in 'F0 19 BEQ \$02EF' geändert, akzeptiert

```

1 REM *****
2 REM * RESTORE ZEILEN-NR. (C64) *
3 REM *****
4 :
5 REM EINSCHALTEN: SYS 680 / AUSSCHALTEN: SYS 698
6 :
10 FOR I=680 TO 766:READ D:POKE I,D 2153
20 S=S+D:I=I+1:IF I<10 THEN NEXT I:I=I-1 2934
30 READ D:IF D<>S THEN PRINT "FEHLER IN";L+1000:STOP 2921
40 L=L+1:IF I>0 THEN I=0:I=I+1 2199
50 :
60 :
1000 DATA 169,199,162,002,172,000,003,140,252,002,1109 4705
1001 DATA 172,009,003,140,253,002,208,006,173,252,1218 4721
1002 DATA 002,174,253,002,141,008,003,142,009,003,737 4696
1003 DATA 096,032,115,000,201,140,208,036,032,029,889 5072
1004 DATA 168,032,115,000,176,025,032,138,173,032,891 5261
1005 DATA 247,183,132,063,133,064,032,019,166,166,1205 5284
1006 DATA 096,164,095,208,001,202,136,132,065,134,1233 5159
1007 DATA 066,076,174,167,160,255,198,123,032,251,1502 5673
1008 DATA 168,108,252,002,000,000,000,530 3394

```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|----|--------|
| 02A8 | A9 | C7 | A2 | 02 | AC | 08 | 03 | 8C | 9C |
| 02B0 | FC | 02 | AC | 09 | 03 | 8C | FD | 02 | 0D |
| 02B8 | D0 | 06 | AD | FC | 02 | AE | FD | 02 | CA |
| 02C0 | 8D | 08 | 03 | 8E | 03 | 03 | 60 | 20 | 40 |
| 02C8 | 73 | 00 | C3 | 8C | D0 | 24 | 20 | 1D | 5E |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| 02D0 | A8 | 20 | 73 | 00 | B0 | 19 | 20 | 8A | 27 |
| 02D8 | AD | 20 | F7 | B7 | 84 | 3F | 05 | 40 | 56 |
| 02E0 | 20 | 13 | A6 | A6 | 60 | A4 | 5F | D0 | 58 |
| 02E8 | 01 | CA | 88 | 84 | 41 | 86 | 42 | 4C | ED |
| 02F0 | AE | A7 | A0 | FF | C6 | 7B | 20 | FB | 18 |
| 02F8 | A8 | 6C | FC | 02 | 00 | 00 | 00 | 0D | 28 |

MIPs 3) Beispiel neuer BASIC-Befehl (FETCH-Befehl).

```

..
..----- Befehl ein/ausschalten -----
..
.. 02A8 A9 CC LDA #5CC > Neuer BASIC-Vektor in ACCU/X
.. 02AA A2 02 LDX #02 > (low/high) nach #02CC.
.. 02AC EC FE 02 CPX #0309 > BASIC-Vektor schon verstellt ?
.. 02AC F0 1A BEQ #02CB > dann Programmteil überspringen.
..
.. 02B1 AC 08 03 LDY #0308 > Original-Vektor nach #02FD/E
.. 02B4 BC FD 02 STY #02FD > umladen. Fortsetzung mit
.. 02B7 AC 09 03 LDY #0309 > 'JMP (#02FD)'.
.. 02BA BC FE 02 STY #02FE >
.. 02BD D0 06 BNE #02C5 > Neuen Vektor abspeichern.
..
.. 02BF AD FD 02 LDA #02FD > 'Alten' Vektor laden um den
.. 02C2 AE FE 02 LDX #02FE > Befehl abzuschalten.
.. 02C5 BD 08 03 STA #0308 > Vektor ändern.
.. 02C8 BE 09 03 STX #0309 >
.. 02CB 60 RTS > Zurück zum BASIC (Ende).
..
..----- Hauptprogramm -----
..
.. 02CC 20 73 00 JSR #0073 > Zeiger in's BASIC erhöhen.
.. 02CF A0 FF LDY #FF > Durch 'INY' beginnt die folgen-
.. > de Routine mit 0 (LDA #02F5,Y).
..
.. 02D1 C8 INY >
.. 02D2 B9 F5 02 LDA #02F5,Y > Vergleicht BASIC-Programmteil
.. 02D5 F0 0E BEQ #02E5 > mit dem Text 'FETCH'
.. 02D7 D1 7A CMP #7A,Y > Holt Anzahl gedrückte Tasten.
.. 02D9 F0 F6 BEQ #02D1 > Vergleich stimmt, sonst
..
.. 02DB C6 7B DEC #7B > Zeiger in's BASIC minus 1
.. 02DD A0 FF LDY #FF > Genau: (Zeiger - 256) + 255
.. 02DF 20 FB A8 JSR #A8FB >
.. 02E2 6C FD 02 JMP #02FD > Zur ROM-Befehls-Routine.
..
.. 02E5 20 FB A8 JSR #A8FB > Zeiger in's BASIC plus Y-Reg.
.. 02EB 20 A6 B3 JSR #B3A6 > Prüft Direkt-Mode. Gegebenen-
.. > falls 'ILLEGAL DIRECT ERROR'.
.. 02EB A5 C6 LDA #C6 > Holt Anzahl gedrückte Tasten.
.. 02ED F0 FC BEQ #02EB > Null, dann erneut prüfen.
..
.. 02EF 20 92 AB JSR #A892 > ROM-Routine 'normaler GET'.
..
.. 02F2 4C AE A7 JMP A7AE > Zum nächsten Befehl (Ende).
..
.. 02F5 46 45 54 43 4B > Vergleichstext zum Befehl.
.. F..E..T..C..H
..
.. 02FA 00 00 00 > Null als End-Kennung.
..
.. 02FD ** ** < VEKTOR > > Hier Vektor zur Fortsetzung.

```

tiert RESTORE auch Variablen. Im Basic-Lader muß hierzu ein Wert der Zeile 1004 geändert werden (176 → 240).

(1004 DATA ***, ***, ***,
,000,240,025, ... :

Bei **MIPs 3** wird's interessant, daß es einen wirklich neuen Basic-Befehl gibt. Der FETCH-Befehl ersetzt den folgenden Ausdruck:
'100 GET AS : IF AS = "" THEN 100' durch '100 FETCH AS'.

Das Hauptprogramm beginnt wieder mit CHRGET. Übrigens 'überliest' diese Routine Leerzeichen (Spaces), so daß die Basic-Zeiger anschließend immer auf ein Zeichen, Token oder das Ende der Programmzeile weisen.

Der Abschnitt \$02D1 bis 02D9 vergleicht das Basic-Programm mit der Tabelle ab \$02F5. Der folgende Programmteil (\$02DB-02DF) setzt den Basic-Zeiger herab. Genau gesagt werden durch DEC \$7B zuerst 256 Bytes abgezogen. Die Routine \$A8FB addiert dann den Wert des Y-Registers (255) hinzu. Eine umständliche, aber notwendige 'Byte-Knauerei'. Falls das Programm die Buchstaben 'F-E-T-C-H' gefunden hat, wird im 6. Durchgang der Wert Null (\$02FA) geladen und die Schleife verzweigt zur Adresse \$02E5.

Der Basic-Zeiger wird mit JSR \$A8FB um sechs Bytes erhöht und zeigt nun auf die String-Variable hinter dem Befehl.

Die Adresse \$C6 gibt an, wieviele Tasten gedrückt wurden, und hier (\$02EB-\$02ED) wartet das Programm, bis sich ein Wert ungleich Null ergibt. Anschließend folgt die 'normale' GET-Routine und mit JMP \$A7AE geht's dann weiter zum nächsten Basic-Befehl.

Da der FETCH-Befehl nicht in einen Token umgewandelt wird, ist dies sicherlich keine sehr elegante Methode. In jedem Fall kann

```

1 REM *****
2 REM *          NEUER BASIC-BEFEHL (FETCH)      (64) *
3 REM *****
4 :
5 REM EINSCHALTEN: SYS 680 / AUSSCHALTEN: SYS 703
6 :
10 FOR I=680 TO 766:READ D:POKE I,D          2153
20 S=S+D:Z=Z+1:IF Z<10 THEN NEXT I:Z=-1    2934
30 READ D:IF D<>S THEN PRINT "FEHLER IN"/L+1000:STOP 2921
40 L=L+1:IF Z>0 THEN Z=0:S=0:NEXT          2199
50 :
80 :
1000 DATA 169,204,162,002,236,009,003,240,026,172,1223 4884
1001 DATA 009,003,140,253,002,172,009,003,140,254,984 5176
1002 DATA 002,200,006,173,253,002,174,254,002,141,1215 5039
1003 DATA 008,003,142,009,003,096,032,115,000,160,568 4998
1004 DATA 255,200,185,245,002,240,014,209,122,240,1712 4907
1005 DATA 246,198,129,160,255,032,251,160,108,253,1794 5345
1006 DATA 002,032,251,168,032,166,179,165,198,240,1433 5284
1007 DATA 252,032,146,171,076,174,167,070,069,084,1241 5424
1008 DATA 067,072,000,000,000,000,000,139 3505

```

```

00 01 02 03 04 05 06 07 CHKSUM
02A8 A9 CC A2 02 EC 09 03 F0 72
02B0 1A AC 08 03 0C FD 02 AC 59
02B8 03 03 0C FE 02 D0 06 AD 21
02C0 FD 02 AE FE 02 8D 08 03 25
02C8 0E 09 03 60 20 73 00 A0 36

```

```

02D0 FF C8 B9 F5 02 F0 0E D1 4B
02D8 7A F0 F6 C6 7B A0 FF 20 74
02E0 FB A8 6C FD 02 20 FB A8 F7
02E8 20 A6 B3 A5 C6 F0 FC 20 3A
02F0 92 AB 4C AE A7 46 45 54 D7
02F8 43 48 00 00 00 00 00 0D EF

```

MIPs 4 Beispiel zum OUT-Vektor (neue CHR\$(Funktion).

```

..
..
.. 02A8 A9 CC LDA #0CC ; Neue OUT-Verzweigung über die
.. 02AA A2 02 LDX #002 ; Adresse $02CC.
.. 02AC EC FE 02 CPX $0327 > Vektor schon verstellt ?
.. 02AD F0 1A BED $02CB > dann fertig.
..
.. 02B1 AC 26 03 LDY $0326 ;
.. 02B4 BC FD 02 STY $02FD ; Original-Vektor
.. 02B7 AC 27 03 LDY $0327 ; umladen.
.. 02BA BC FE 02 STY $02FE ;
.. 02BD D0 06 BNE $02C5 > neuen Vektor abspeichern.
..
.. 02BF AD FD 02 LDA $02FD ; Programm abschalten.
.. 02C2 AE FE 02 LDX $02FE ;
.. 02C5 BD 26 03 STA $0326 ; Vektor 'verstellen'.
.. 02CB BE 27 03 STX $0327 ;
.. 02CD 60 RTS > Programmteil Ende.
..
.. -----
.. Hauptprogramm -----
..
.. 02CC C9 80 CMP #000 > Zeichen gleich CHR$(128) ?
.. 02CE D0 2A BNE $02E6 > nein, dann Prog. überspringen.
..
.. 02D0 48 PHA > Zeichen 'sichern'.
.. 02D1 A5 9A LDA $9A > Aktives Ausgabe-Gerät abfragen.
.. 02D3 C9 03 CMP #003 > Gleich Bildschirm ?
.. 02D5 D0 22 BNE $02E2 > sonst Befehl übergehen.
..
.. 02D7 98 TYA ;
.. 02D8 48 PHA ; Y- und X-Register in den
.. 02D9 8A TXA ; Stack 'retten'.
.. 02DA 48 PHA ;
.. 02DB A0 00 LDY #000 ;
.. 02DD 84 AE STY $AE ; Indirekte Adresse ($AE/AF) auf
.. 02DF A9 DB LDA #00B ; Anfang Farb-RAM stellen.
.. 02E1 B5 AF STA $AF ;
..
.. 02E3 A2 04 LDX #004 > 4 Durchgänge zu 256 Bytes.
.. 02E5 AD 86 02 LDA $0286 > Aktuelle 'Schrift-Farbe' laden.
.. 02E9 91 AE STA ($AE),Y ;
.. 02EA C8 INY ; Farb-RAM setzen.
.. 02EB D0 FB BNE $02EB ;
.. 02ED E6 AF INC $AF > Zeiger (high) erhöhen.
.. 02EF CA DEX > Schon vier Durchgänge = 1 kByte
.. 02F0 D0 F6 BNE $02EB > nein, dann erneut.
..
.. 02F2 68 PLA ;
.. 02F3 AA TAX ; X- und Y-Register
.. 02F4 68 PLA ; zurückholen.
.. 02F5 AB TAY ;
.. 02F6 68 PLA > CHR$(Code zurückholen.
.. 02F7 18 CLC > Hauptprogramm Ende.
.. 02F8 60 RTS ;
..
.. 02F9 68 PLA > CHR$(Code zurückholen.
.. 02FA 6C FD 02 JMP ($02FD) > OUT-Vektor fortführen.
..
.. 02FB 88 88 < VEKTOR > > Adresse 'OUT-Fortsetzung'.

```

das Programm jedoch als Ausgangsbasis für Experimente dienen.

Sonderfunktion durch CHR\$(

MIPs 4 gibt dem CHR\$(128) eine Sonderfunktion. Zeichen werden immer im Akku übergeben. Das Hauptprogramm überprüft stets, ob das Zeichen CHR\$(128) gedruckt werden soll. Falls man beispielsweise 'PRINT "1234"' eingibt, bekommt die Routine nacheinander die Zahlen 1 bis 4 (im Akku) 'angeboten'. Anschließend geht's direkt mit JMP (\$02FD) weiter, da nur das Zeichen CHR\$(128) eine Sonderfunktion erhält. Wird das Zeichen erkannt, dann prüft das Programm den Zeiger \$9A. Der Wert Drei bedeutet, daß die Ausgabe zum Bildschirm geht. Andernfalls (z. B. vier beim Drucker) wird ein Peripheriegerät angesprochen, und der Programmteil \$02D7 bis \$02F8 wird übersprungen.

Durch 'PRINT CHR\$(28); CHR\$(128)' werden beispielsweise alle Zeichen auf dem Bildschirm rot. Da — neben CHR\$(128) — noch weitere Codes unbenutzt sind (siehe Handbuch), kann man beispielsweise mit CHR\$(21) (und einem entsprechenden Programm) den Bildschirm nach links rollen. Nur eines muß man unbedingt beachten. Die Routine darf keinesfalls das X- und Y-Register verändern. Es ist am einfachsten, diese Werte — wie hier gezeigt — im Stack zu sichern.

MIPs 5 ändert den Vektor für Fehlermeldungen (\$0300-\$0301). Doch zuerst ein Anwendungsbeispiel:

```

10 SYS 680,100
20 :
30 :Halt
40 :

```

```

1 REM *****
2 REM *          NEUER CHR$(128)          (C-64) *
3 REM *****
4 :
5 REM EINSCHALTEN: SYS 680 / AUSSCHALTEN: SYS 703
6 :
10 FOR I=680 TO 766:READ D:POKE I,D           2153
20 S=S+D:Z=Z+1:IF Z<10 THEN NEXT I:Z=-1     2934
30 READ D:IF D<S THEN PRINT "FEHLER IN"/L+1000:STOP 2921
40 L=L+1:IF Z>0 THEN Z=0:S=0:NEXT I         2199
50 :
60 :
1000 DATA 189,204,162,002,236,039,003,240,026,172,1253 5100
1001 DATA 038,003,140,253,002,172,038,003,140,254,1044 5219
1002 DATA 002,208,006,173,253,002,174,254,002,141,1215 5039
1003 DATA 038,003,142,039,003,096,201,128,208,042,900 4953
1004 DATA 072,165,154,201,003,208,034,152,072,138,1199 5345
1005 DATA 072,160,000,132,174,169,216,133,175,162,1393 5596
1006 DATA 004,173,134,002,145,174,200,208,251,230,1521 5168
1007 DATA 175,202,208,246,104,170,104,168,104,024,1505 5031
1008 DATA 096,104,108,253,002,000,000,563 3751

```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|----|--------|
| 02A8 | A9 | CC | A2 | 02 | EC | 27 | 03 | F0 | 54 |
| 02B0 | 1A | AC | 26 | 03 | 8C | FD | 02 | AC | 77 |
| 02B8 | 27 | 03 | 8C | FE | 02 | D0 | 06 | AD | 3F |
| 02C0 | FD | 02 | AE | FE | 02 | 8D | 26 | 03 | 43 |
| 02C8 | 8E | 27 | 03 | 60 | C9 | 80 | D0 | 2A | F8 |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| 02D0 | 48 | A5 | 9A | C9 | 03 | D0 | 22 | 98 | 33 |
| 02D8 | 48 | 8A | 48 | A0 | 00 | 84 | AE | A9 | E8 |
| 02E0 | D8 | 85 | AF | A2 | 04 | AD | 86 | 02 | 38 |
| 02E8 | 91 | AE | C8 | D0 | FB | E6 | AF | CA | 05 |
| 02F0 | D0 | F6 | 68 | AA | 68 | A8 | 68 | 18 | A9 |
| 02F8 | 60 | 68 | 6C | FD | 02 | 00 | 00 | 00 | 5D |

MIPS 5) Beispiel zum ERROR-Vektor (ON ERROR GOTO-Ersatz).

```

..
.. 02A0 20 9B B7 JSR $B79B  ; Holt Zeilen-Nr. (SYS 680,XXXX)
.. 02AB B6 02 STX $02      ; und speichert diese nach $02.
..
.. 02AD A9 D1 LDA #$01    ; Neuer ERROR-Vektor in ACCU/X
.. 02AF A2 02 LDX #$02    ; (low/high) = $02D1.
.. 02B1 EC 01 03 CPX $0301 ; Vektor schon verstellt ?
.. 02B4 F0 1A BEQ $02D0    ; dann nach RTS.
..
.. 02B6 AC 00 03 LDY $0300 ;
.. 02B9 BC FB 02 STY $02FB ; Original-Vektor nach $02FB/C
.. 02BC AC 01 03 LDY $0301 ; umladen.
.. 02BF BC FC 02 STY $02FC ;
.. 02C2 D0 06 BNE $02C6    ; Neuen Vektor abspeichern.
..
.. 02C4 AD FB 02 LDA $02FB ; Original-Vektor zurueckladen.
.. 02C7 AE FC 02 LDX $02FC ;
.. 02CA 8D 00 03 STA $0300 ; Vektor abspeichern.
.. 02CD BE 01 03 STX $0301 ;
.. 02D0 60 RTS             ; Programmteil Ende.
..
----- Hauptprogramm -----
..
.. 02D1 BA TXA             ; X = Fehler-Nr. (siehe Text)
.. 02D2 30 24 BMI $02FB    ; groeuer $7F, dann kein Fehler
..
.. 02D4 B6 9E STX $9E      ; Fehler-Nr. in $9E (15B) merken.
.. 02D6 A4 3A LDY $3A      ; Zeilen-Nr. (high) fuer CONT.
.. 02D8 C8 INY             ; gleich $FF (255), dann
.. 02D9 F0 1D BEQ $02FB    ; direkte Eingabe (Fehler).
..
.. 02DB A5 02 LDA $02      ; ACCU = Zeilen-Nr. 'ON ERROR'.
.. 02DD 8B DEY            ;
.. 02DE D0 04 BNE $02E4    ; Fehler in Zeile fuer 'ON ERROR',
.. 02E0 C5 39 CMP $39      ; dann Programmabbruch.
.. 02E2 F0 14 BEQ $02FB    ;
..
.. 02E4 B5 14 STA $14      ; Zeilen-Nr. in Zeiger (low).
.. 02E6 20 7A A6 JSR $A67A ; CLR-Befehl zum Teil ausfuehren.
.. 02E9 B5 15 STA $15      ; Zeilen-Nr. (high) (ACCU = $00).
..
.. 02EB 20 13 A6 JSR $A613 ; 'ON ERROR'-Zeile suchen.
.. 02EE 90 06 BCC $02F6    ; Nicht gefunden, falls CLR CARRY
..
.. 02F0 20 A8 A3 JSR $A3A8 ; ROM-Routine 'GOTO-Befehl'.
.. 02F3 4C A7 AE JMP $AEA7 ; Weiter im BASIC (Ende).
..
.. 02F6 A6 9E LDX $9E      ; Fehler-Nr. zurueckladen.
.. 02FB 6C FA 02 JMP ($02FB) ; Fehlerbehandlung im ROM.
..
.. 02FB ** ** < VEKTOR > ; Fehler-Vektor (siehe $02CA-D0)

```

```

100 PRINT "FEHLER-NR.:";PEEK(15B)
110 END

```

Die Anweisung SYS 680 macht das Programm 'scharf', und die folgende Zahl gibt an, mit welcher Zeile es — bei einem Fehler — weitergehen soll. Nach RUN findet der Computer in der Zeile 30 einen prächtigen 'SYNTAX ERROR' und es erscheint die Meldung:

FEHLER-Nr.:11

Nach SYS 680,100 holt das Programm durch JSR \$B79B die Zeilennummer (im Beispiel 100) und merkt sich den Wert in der Adresse \$02. Das Hauptprogramm prüft zuerst, ob die Fehlernummer — im X-Register — größer als 127 ist. Der Fehlervektor wird auch bei einem korrekten Programmstopp (z. B. durch END) durchlaufen. In diesem Fall enthält das X-Register den Wert \$80, und das Programm läuft weiter ins ROM. Dies geschieht ebenso, falls ein direkter Eingabefehler vorliegt (z. B. PRINT A = 1). Anderenfalls würde das Basic-Programm automatisch bei jedem Eingabe-Fehler starten, was auf die Dauer lästig ist. Solange die Adresse \$3A den Wert 255 enthält, befindet sich der Computer im Direkt-Modus; es läuft also kein Programm. Außerdem wird geprüft, ob die fehlerhafte Zeile genau jene ist, zu der das Programm verzweigen soll. Also: ERROR in 100, verzweige nach 100, ERROR in 100, usw. . . . Die Routine \$A67A initialisiert den Computer teilweise. So werden beispielsweise unbeendete FOR-NEXT-Schleifen gelöscht. Dieser Programmteil führt jedoch (mit Absicht) kein 'RESTORE' durch. Die Adresse \$14/15 enthält nun die Nummer der Programm-Zeile, zu der die Verzweigung führen soll. Mit JSR \$A613 wird anschließend geprüft, ob diese Zeile auch im Programm auftritt, anderenfalls

```

1 REM *****
2 REM *          ON ERROR GOTO ... (C-64) *
3 REM *****
4 :
5 REM EINSCHALTEN: SYS 680 / AUSSCHALTEN: SYS 708
6 :
10 FOR I=680 TO 766:READ D:POKE I,D           2153
20 S=S+D:I=I+1:IF Z<10 THEN NEXT I:Z=-1     2934
30 READ D:IF D<>S THEN PRINT "FEHLER IN";L+1000:STOP 2921
40 L=L+1:IF Z>0 THEN Z=0:S=0:NEXT I         2193
50 :
60 :
1000 DATA 032,155,183,134,002,169,209,162,002,236,1284 5535
1001 DATA 001,003,240,026,172,000,003,140,251,002,838 4549
1002 DATA 172,001,003,140,252,002,206,006,173,251,1208 4828
1003 DATA 002,174,252,002,141,000,003,142,001,003,720 4560
1004 DATA 096,138,049,036,134,158,164,058,200,240,1272 5064
1005 DATA 029,165,002,136,208,004,197,057,240,020,1058 5328
1006 DATA 133,020,032,122,166,133,021,032,019,166,844 5328
1007 DATA 144,006,032,163,168,076,174,167,166,158,1254 5336
1008 DATA 108,251,002,000,000,000,000,361 3567

```

| | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | CHKSUM |
|------|----|----|----|----|----|----|----|----|--------|
| 02A8 | 20 | 9B | B7 | 86 | 02 | A9 | D1 | A2 | 3F |
| 02B0 | 02 | EC | 01 | 03 | F0 | 1A | AC | 00 | 96 |
| 02B8 | 03 | 8C | FB | 02 | AC | 01 | 03 | 8C | 91 |
| 02C0 | FC | 02 | D0 | 06 | AD | FB | 02 | AE | C9 |
| 02C8 | FC | 02 | 0D | 00 | 03 | 0E | 01 | 03 | FA |

| | | | | | | | | | |
|------|----|----|----|----|----|----|----|----|----|
| 02D0 | 60 | 8A | 30 | 24 | 86 | 9E | A4 | 3A | 34 |
| 02D8 | C8 | F0 | 1D | A5 | 02 | 88 | D0 | 04 | 97 |
| 02E0 | C5 | 39 | F0 | 14 | 85 | 14 | 20 | 7A | 7E |
| 02E8 | A6 | 85 | 15 | 20 | 13 | A6 | 90 | 06 | 0D |
| 02F0 | 20 | A3 | A8 | 4C | AE | A7 | A6 | 9E | E8 |
| 02F8 | 6C | FB | 02 | 00 | 00 | 00 | 00 | 0D | 67 |

MIPs 6) Kassetten-Analyse (C-64)

```

..
..
.. 02A8 A9 00 LDA #00      > Länge des Filenamens = 0.
.. 02AA B5 B7 STA #B7      >
.. 02AC 20 D0 F7 JSR $F7D0  > Setzt Puffer-Adresse.
.. 02AF 20 17 F8 JSR $F817  > Wartet auf Play-Taste.
.. 02B2 B0 3F BCS $02F3    > Verzweigung bei STOP.
.. 02B4 20 AF F5 JSR $F5AF  > Schreibt 'SEARCHING'.
.. 02B7 20 2C F7 JSR $F72C  > Sucht Header (Pg.-Start).
.. 02BA A9 0D LDA #0D      >
.. 02BC 20 D2 FF JSR $FFD2  > RETURN auf dem Bildschirm.
..
.. 02BF A0 01 LDY #01      >
.. 02C1 B1 B2 LDA ($B2),Y  > Läd Startadresse der
.. 02C3 85 C1 STA #C1      > Daten (low, high)
.. 02C5 AA TAX              >
.. 02C6 C8 INY              > in's X-Reg., Accu
.. 02C7 B1 B2 LDA ($B2),Y  > und #C1, C2.
.. 02C9 85 C2 STA #C2      >
.. 02CB 20 CD BD JSR $BDCD  > Druckt Zahl in X, Accu.
.. 02CE A9 2D LDA #2D      >
.. 02D0 20 D2 FF JSR $FFD2  > Schreibt '-'.
.. 02D3 A0 03 LDY #03      >
.. 02D5 B1 B2 LDA ($B2),Y  > Läd Endadresse in
.. 02D7 AA TAX              >
.. 02D8 C8 INY              > X-Reg. und Accu.
.. 02D9 B1 B2 LDA ($B2),Y  >
.. 02DB 20 CD BD JSR $BDCD  > Druckt Endadresse.
.. 02DE A9 3D LDA #3D      >
.. 02E0 20 D2 FF JSR $FFD2  > Schreibt '='.
.. 02E3 38 SEC              >
.. 02E4 A0 03 LDY #03      > Errechnet Programm-
.. 02E6 B1 B2 LDA ($B2),Y  >
.. 02E8 E5 C1 SBC #C1      > Länge.
.. 02EA AA TAX              >
.. 02EB C8 INY              > #C1,C2 = Startadr.
.. 02EC B1 B2 LDA ($B2),Y  >
.. 02EE E5 C2 SBC #C2      >
.. 02F0 20 CD BD JSR $BDCD  > Druckt Länge.
.. 02F3 60 RTS              >

```

würde sich der Computer 'aufhängen'. Ein gesetztes Carry-Flag signalisiert, daß die Zeile gefunden wurde. JSR \$A3A8 ist eine Unter-routine des GOTO-Befehls, und setzt den Zeiger ins Basic (\$7A/7B) entsprechend der Zeilennummer in \$14/15. In der Zeile \$02F3 geht's dann weiter mit dieser Programmzeile.

Wie in dem Testprogramm gezeigt, kann man durch PEEK(158) die Fehlernummer (Fehler-Art) abfragen. Zuletzt sollte man beachten, daß als 'Ziel-Adresse' nur eine Zeilennummer kleiner 255 angegeben werden kann, anderenfalls gibt's ILLEGAL QUANTITY ERROR.

MIPs 6 ist mit dem Namen „Kassettenanalyse“ überschrieben. Nach einem Durchlauf des dazugehörigen Basic-Ladeprogramms kann die Maschinenroutine mit SYS 640 aufgerufen werden. Nachdem die Play-Taste des Kassettenrecorders gedrückt wurde, sucht die Routine nach dem ersten Programm, das auf der Kassette abgespeichert ist. Anschließend werden die Start- und End-Adresse sowie die Länge des Programms auf dem Bildschirm ausgegeben.

Mit diesem Analyse-Programm kann man schnell ermitteln, in welchem Speicherbereich ein Programm liegt und wie lang es ist. Der Kernteil des Programms ist die Zeile \$02B7. Jedes auf Kassette abgespeicherte Programm beginnt mit einem Start-Teil (engl. HEADER), der den Programmnamen sowie die Start- und Endadresse umfaßt. Dieser Header wird von der Routine \$F72C geladen und gespeichert. Der Zeropa-ge-Zeiger \$B2/\$B3 zeigt anschließend auf diese Daten, die in dem Programmteil \$02BF-\$02F3 ausgewertet werden.

(Heino Velder)

```

1 REM *****
2 REM *           KASSETTEN-ANALYSE      (C-64) *
3 REM *****
4 :
5 REM AUFRUF: SYS 680
6 :
10 FOR I=680 TO 759:READ D:POKE I,D           2158
40 S=S+D:Z=Z+1:IF Z<10 THEN NEXT :Z=-1      2934
50 :
60 READ D:IF D<>S THEN PRINT "PRUEFFEHLER !":STOP 2730
70 IF Z>0 THEN Z=0:S=0:NEXT                 1837
80 :
90 :
1000 DATA 169,000,133,183,032,200,247,032,023,240,1275 4947
1001 DATA 176,063,032,175,245,032,044,247,169,013,1196 5717
1002 DATA 032,210,255,160,001,177,178,133,193,170,1509 5691
1003 DATA 200,177,178,133,194,032,205,189,169,045,1522 5615
1004 DATA 032,210,255,160,003,177,178,170,200,177,1562 5578
1005 DATA 178,032,205,189,169,061,032,210,255,056,1307 5366
1006 DATA 160,003,177,178,229,193,170,200,177,170,1665 5380
1007 DATA 229,194,032,205,189,096,000,000,000,000,945 4688

```

| 00 01 02 03 04 05 06 07 CHKSUM | 02C8 B2 05 C2 20 CD BD A9 2D 5A |
|---------------------------------|---------------------------------|
| 02A8 A9 00 B5 B7 20 D0 F7 20 9E | 02D0 20 D2 FF A0 03 B1 B2 AA 08 |
| 02B0 17 F8 B0 3F 20 AF F5 20 D7 | 02D8 C8 B1 B2 20 CD BD A9 3D 24 |
| 02B9 2C F7 A9 0D 20 D2 FF A0 7E | 02E0 20 D2 FF 38 A0 03 B1 B2 B0 |
| 02C0 01 B1 B2 85 C1 AA C8 B1 AB | 02E8 E5 C1 AA C8 B1 B2 E5 C2 28 |
| | 02F0 20 CD BD 60 00 00 00 00 B1 |

Wir stellen aus:
SYSTEMS 85 SIS
München 28. Oktober - 1. November
Halle 15 (O.G.)
Stand B 4/C5

WARUM DER COMMODORE EINEN OKI-DRUCKER BRAUCHT

OKI IST O.K.

Bei aller OKI-typischen Bescheidenheit gibt es doch immer wieder eine Nachricht, die es Wert ist, an die große Glocke gehängt zu werden:

Zwei der sagenhaft standfesten OKI-DRUCKER sind ab sofort und ohne Einschränkungen auch mit den Commodores kompatibel.

„Eine Nachricht, die besser ist als 72 Stunden Landurlaub in Bangkok“, meint dazu der Commodore. „Ay, ay, Sir“, sagt OKI und weiß, wovon er spricht.

So denkt der Commodore zunächst einmal an den MICROLINE 182 von OKIDATA. Ein idealer Partner, wenn es darum geht, maximales Druckvergnügen mit minimalem Aufwand zu erzielen. „Hi-Tech zum Low-Preis“, sagt der alte Seebär und meint damit, daß der MICROLINE 182 nicht nur kleiner, leiser und leistungsfähiger als die meisten seiner Kollegen ist, sondern daß er all' seine Vorzüge auch zu einem vorzüglichen Preis anbieten kann.

Noch glänzender werden die Augen des Commodore – allerdings, wenn die Sprache auf den OKIMATE 20 kommt.

Ein Drucker mit schier unbegrenzten Möglichkeiten – der ideale Partner für alle, die es gar nicht bunt genug treiben können.

So ist der OKIMATE 20 überaus diskret, was seine Lautstärke angeht. Überaus attraktiv, wenn man



seine schlanke Erscheinung ins Auge fasst. Und überaus verlockend, wenn man bedenkt, daß er vielfarbig im Druck, aber dabei nur 3-stellig im Preis ist.

Bleibe nur noch die Frage des geeigneten Zielhafens für alle interessierten Commodores zu klären.

Wir empfehlen auch hier den Computer-Fachhandel, wo man Sie nicht nur mit einem „Welcome on board“ begrüßen wird, sondern auch mit allem, was Sie sonst noch über die anderen Drucker von OKIDATA wissen wollen.

Und wenn nicht – dann haben wir noch einen kleinen Coupon für Sie – als „Rettungsring“ sozusagen.

OKIDATA

GMBH

OKIDATA GmbH · Abt. 11C
Emanuel-Leutze-Straße 8 · 4000 Düsseldorf 11
Telefon 0211-59794-01 · Telex 8587218
Telefax 0211-593345

Coupon

Schicken Sie mir uns mehr Informationen über

- | | |
|---|--|
| <input type="checkbox"/> MICROLINE 82A/83A | <input type="checkbox"/> OKIMATE 20 |
| <input type="checkbox"/> MICROLINE 84 | <input type="checkbox"/> MICROLINE 182 |
| <input type="checkbox"/> MICROLINE 92/93 | <input type="checkbox"/> MICROLINE 192 |
| <input type="checkbox"/> PACEMARK 2350/2410 | <input type="checkbox"/> MICROLINE 193 |

Name _____

Straße _____

PLZ _____ Wohnort _____



Für Freaks das englische Original!



Jetzt komplett im Angebot:

Enterprise 64 K

Der original englische Computer incl. EXOS 2.0 – auf Wunsch mit deutschem Bedienerhandbuch. Nachrüstbar auf 128 K. Upgrade – Kits (für deutsches Betriebssystem und damit deutsche Textbearbeitung) bereits verfügbar.

Monochrom Monitor

Monitorkabel

Matrixdrucker
Epson – kompatibel, grafikfähig, Einzelblatteinzug, Centronics Schnittstelle.

Druckerkabel

BASIC

integrierter Wordprocessor



Alles komplett zum Freakpreis nur DM

1.998,-

incl. MwSt. – unverbindliche Preisempfehlung. Sofort Händlerverzeichnis anfordern!

ENTERPRISE[®] COMPUTERS

ENTERPRISE GMBH
COMPUTERSASSE 3
COMNENSTRASSEN 2
SOONENWÄNCHEN
D-8000 MÜNCHEN 5170550
TELEFON (089) 5170550

Ich möchte Mitglied im ENTERPRISING-Club werden

Ihr Händlerverzeichnis Produktinformationen ENTERPRISING – Clubzeitung

Schicken Sie mir schnellstens über ENTERPRISE. Ich brauche Informationen

Name _____

Anschrift _____