

monatlich

Compute mit

COMMODORE & SCHNEIDER

2/87

VC-64, VC-20, C-16/116, CPC-464

3,80 DM

33 öS

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

3,80 sFr

Aktuell +++ Aktuell +++ Aktuell +++ Aktuell

VC 20:

**Cave Fighter
Quest for Tires**

C 16/116/plus 4:

**McStoney in the Caves
Cube, Hostage
Adress 16**

C 64:

**Easy Basic
Ski-Jumping
Job Race**

Schneider:

**The Hero, Smash it
16 RSX Befehle
Diskstar**

Der Trend: Low-Cost!



**Leser- + Meckerecke – Report – Kurse
Softwareservice – Tips & Tricks
Großer Kleinanzeigenmarkt**

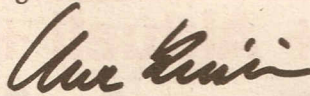
EDITORIAL

Lieber Leser!

Kaum waren die Arbeiten zum Heft 1/87 abgeschlossen, drückten schon die Termine für die Februar-Ausgabe. Im Klartext: Wenn im Handel Heft 1/87 erscheint, ist das nächste Heft verlags-technisch schon fertiggestellt. Aus diesem Grund ist das vorliegende Heft randvoll mit Listings, besonders die C-16 Freunde, aber auch die Schneider-User erwerben mit diesem Heft nicht nur einen umfangreichen Listingteil,

sondern auch qualitativ über-durchschnittliche Software.

In diesem Zusammenhang möchte ich die VC-20-Besitzer aufrufen, sich vermehrt als Autoren zu versuchen, damit wir dieses System auch in Zukunft weiter berücksichtigen können.



Uwe Knierim
Chefredakteur

Report

Bücher	3
Software Review	4

Rubriken

Klartext	5
Leser- & Meckerecke	9
Software-Service	32
Assembler-Kurs	45
Kurs-Interruptprogrammierung	47
Kleinanzeigen	60

IMPRESSUM

"Compute mit"
ISSN 0179-6720
erscheint monatlich im
Tronic-Verlag, Am Stad 35
3440 Eschwege
Tel.: 05651/3 00 11



Redaktion:

Axel Credé (verantwortlich)
Chefredakteur: Uwe Knierim
Redakteure: Frank Brall, Otfried Schmidt, Thomas Brandt, Martina Strack

Gesamtherstellung:

Druckhaus Dierichs Kassel, Frankfurter Str. 168,
3500 Kassel

Anfragen nicht an den Vertrieb oder Druckerei, sondern nur an den Verlag!

Vertrieb

Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
Verlagsunion
6200 Wiesbaden
Friedrich-Bergius-Str. 20
Telefon: 06121/2660

Erscheinungsweise:

Erstverkaufstag von "Compute mit" jeweils Mitte des Monats.

Urheberrecht:

Alle in "Compute mit" veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.

Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.

Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen und Bezeichnungen frei von Schutzrechten sind.

Software Commodore

Quest for Tires (VC 20)	6
Cave Fighter (VC 20)	9

* C 16/116-Top Programm:	*
* Mc Stoney in the Caves	11

Cube (C 16/116)	17
Hostage (C 16/116)	20
Adress 16 (C 16/116/plus 4 m. 64 K)	22
Easy-Basic (C 64)	27
Ski-Jumping (C 64)	31
Job-Race (C 64)	39

Schneider

The Hero	50
Smash it	54
16 RSX Befehle	56
Disc Star 2.1	57

Die Kleinanzeigen-Auftragskarte
finden Sie in der Mitte des Heftes!

Bezugspreis:

Einzelheft 3,80 DM
Abonnement: Inland 42,- DM im Jahr (12 Ausgaben)
Ausland (Europa): 52,- DM
ohne Kassetten!

Programmierabteilung:

Montag - Freitag von 14-16 Uhr
Tel. (05651) 30013

Autoren, Manuskripte:

Der Verlag nimmt Manuskripte zur Veröffentlichung gerne entgegen.

Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,- DM pro abgedruckte Seite einverstanden sind.

Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Alle Einsendungen müssen frei von Rechten Dritter sein. Dies muß mit der Unterschrift des Einsenders bestätigt werden.

Rücksendung erfolgt nur gegen Erstattung der Kosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:

Kopierfähige Kassette oder Diskette mit dem Programm (Computerbezeichnung nicht vergessen!), von Drucker erstelltes Listing (keine Schreibmaschinenlistings!), evtl. Bildschirmfotos oder Hardcopies mit Demonstrationsbeispielen und ausführliche Programmbeschreibung (Erklärung der programmtechnischen Besonderheiten, Spielverlaufbeschreibung). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

Anzeigenpreise:

Bitte Mediaunterlagen anfordern.

Anzeigenverwaltung:

Anzeigenleiter: Hartmut Wendt

Tronic-Verlag GmbH
Am Stad 35, 3440 Eschwege
Telefon: 05651/30011
Telefax: 05651/30014

Commodore 64 Tips und Tools



Mit der Anschaffung des Commodore 64 und der Floppy 1541 steht dem Anwender für wenig Geld ein erstaunlich leistungsfähiges Homecomputersystem zur Verfügung. Um die Möglichkeiten dieses Systems optimal zu nutzen, stellen sich besonders dem Anfänger einige Probleme. Dieses Buch gibt dem unerfahrenen User eine wertvolle Starthilfe u.a. werden hier eine Vielzahl von Tools zur Verfügung gestellt, die das Arbeiten mit dem Computer wesentlich erleichtern.

Die Autoren beschäftigen sich ausführlich mit der Floppy 1541, deren Programmierung sowie die Techniken zur schnellen Datenübertragung. Sie geben außerdem Tips zur Programmgestaltung, stellen Techniken wie Maussimulation oder die Realisierung von Windows vor und beschäftigen sich eingehend mit der hochauflösenden Grafik.

Das Buch enthält eine Fülle von BASIC- und Assembler-Routinen, die zum größten Teil durch Listings dokumentiert sind. Der Leser hat so die Möglichkeit, konkrete Problemlösungen Schritt für Schritt nachzuvollziehen. Alle vorgestellten Programme sind zusammen mit weiteren nützlichen Tools Bestandteil der dem Buch beiliegenden Diskette.

Die Themen:
Manipulation der Diskette

Schnelle Datenübertragung
Kopiertechniken
Floppyprogrammierung
Hardware der Floppy
Fenster-Techniken
Maussimulation
Rasterzeileninterrupt
Eigene Interruptroutinen
Hires-Grafik

Die Tools:

Disk ID-Manager
Hires-Scanner
Character-Scanner
Disk-Monitor
Backup-Copy
Fast-Format
Error-Scanner

Commodore 64 - Tips und Tools, Sybex-Verlag in Düsseldorf, Autoren: Günter Krämer und Wolfgang Sipinski, ISBN 3-88745-641-6, Preis: 58,- DM

Computer sind nicht nur zum Spielen da

Heimcomputer-Bastelkiste

Messen, Steuern und Regeln mit Homecomputern

Homecomputer können mehr leisten als man annimmt. Sie sind zum Beispiel in der Lage, Kleingeräte automatisch ein- und abzuschalten, Meß- und Steuerfunktionen zu übernehmen und komplizierte Abläufe zu überwachen.



Das Buch „HEIMCOMPUTER-BASTELKISTE“ von Gerhard A. Karl, erschienen im Falken-Verlag, verrät anhand von zahlreichen Beispielen, Programmen und Schaltplänen, wie solche Aufgaben gelöst werden. Voraussetzung ist lediglich etwas Bastelge-

schick und Verständnis für elektronische Zusammenhänge. Funkamateure, Modelleisenbahner und Computerfreunde kommen dabei genauso auf ihre Kosten wie Technik-Fans.

Nach einer eingehenden Einführung in die Maschinensprache der Prozessoren 6502 und Z80 werden die wichtigsten Schnittstellenbausteine vorgestellt und Fachbegriffe wie „Transistoren“, „Thyristoren“, „TRIACs“, „Relais“, „LEDs“ u.v.a.m. erläutert. Der Hauptteil des Buches beschäftigt sich mit Anwendungen, wie zum Beispiel das Messen von Widerständen oder Flüssigkeiten, Filter- und Verstärkersteuerungen, Programm- und Schaltungstests, Heizungssteuerung, Anschluß an das Telefon, Schaltungen für Modelleisenbahner, Sprachausgabe, Logikprüfer und EPROM-Programmiergerät. Durch Schaltplan, Aufbauplan auf Lochrasterplatine und Verdrahtungsliste als Check-list sind die Projekte auch für den

weniger elektronisch Vorbelasteten sicher nachzubauen. Die Programme für die vorgeschlagenen Anwendungen sind in der populärsten Computersprache „BASIC“ geschrieben und für die geläufigsten Hobby-Computer ausgelegt: Commodore 64, Sinclair-, MSX-, Apple 2-, MC-, Atari und Tandy-Rechner. Die Listings werden jedoch in ihrem Aufbau geräteunabhängig erklärt.

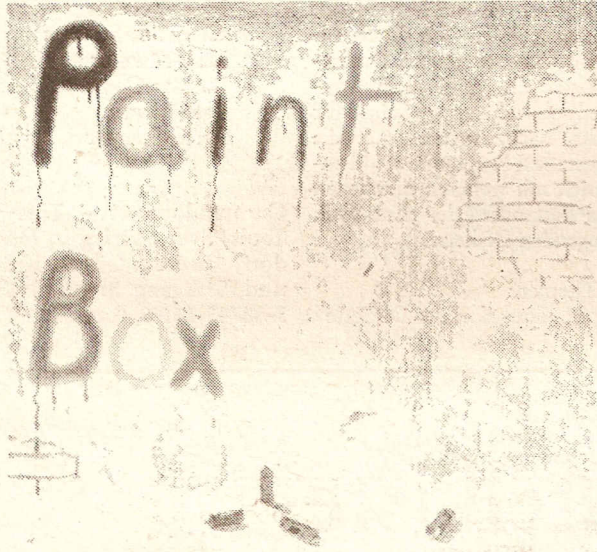
Autor: Gerhard A. Karl, Falken-Verlag Postfach 1120, Schöne Aussicht 21, 6272 Niedernhausen/Ts. - Preis: 39,- DM.



Preissenkungen bei VIZA

Der Preis von VIZASTAR (Version XL-8), dem Kalkulationsprogramm für den C 64, wurde von bisher 398,-DM auf 298,- DM gesenkt. Damit ist das integrierte Datenbank- und Kalkulationsprogramm mit drei-dimensionaler Farbgrafik interessanter als vergleichbare Einzellösungen. Die kleinere Version XL-4 kostet jetzt 198,- DM. Weiterhin bietet der deutsche Distributor den Austausch von XL-4 in XL-8 zu einem Preis von 120,- DM an. Neu auf dem Markt ist nun auch VIZASTAR 128 für den Commodore 128. Da dieses Programm bisher nur in der englischen Version verfügbar ist, bietet DTM den registrierten Anwendern die kostenlose Nachsendung der deutschen Version an, sobald lieferbar. Fortsetzung nächste Seite.

Bildschirm-graffities



Programm: Paintbox, System: C 64, Preis: (Kass.) ca. 9,-DM, Hersteller: Golden Games, Vertrieb: Micro-Händler

Ein Malprogramm, noch ein Malprogramm, und noch eines für den C 64. Natürlich muß man sich da fragen, ob es sich überhaupt lohnt, ein neues Zeichenprogramm auf den Markt zu bringen. Besonders für den C 64 haben bestimmte Programme einen Standard gesetzt, der schwerlich zu überbieten ist. Hier ist es insbesondere der Koala Painter, der in fast jeder C-64-Malerwerkstatt vertreten sein dürfte. Dies hat letztlich dazu geführt, daß sich andere Hersteller von Software und sogar von Hardware (z.B. bei Videodigitizern) sich an diesem Format orientiert haben, obwohl der Koala Painter in einem ungünstigen Format arbeitet, da er das Commodoreformat ignoriert und die Reihenfolge von Videoram und Farbram vertauscht und so die Einbindung erstellter Bilder in eigene Programme unnötig erschwert.

Weitere Argumente für ein neues Malprogramm könnten sein: Die

schwere Beschaffbarkeit von Koalainter oder anderen alteingesessenen Programmen, das ungünstige Preis-Leistungsverhältnis.

Mit all diesen Dingen räumt die Paint Box auf. Dieses explizite Low-Cost-Produkt bietet weitgehend die gleichen Features wie die teureren Brüder, jedoch weist die Zoom-Funktion besondere Qualitäten auf.

Um Paint Box beurteilen zu können, muß man wissen, daß es sich hierbei nicht um ein nagelneues Programm handelt, sondern um eine Schubladenentwicklung. Paint Box hat daher gar nicht erst den Anspruch, etwas umwerfend Neues zu bieten. Vor allem für Neueinsteiger bietet Paintbox daher ausreichend Qualitäten zu einem Preis, der kaum der Rede Wert ist.

Positiv: Alle wichtigen Features vorhanden, einfache Bedienung, sehr gutes Preis-Leistungs-Verhältnis.

Negativ: dürftige Anleitung

Höllisches Unternehmen

Programm: Dantes Inferno, System: C-64, Preis: ..., Hersteller: Beyond Software.

Die neue Beyond-Veröffentlichung ist nichts für schwache Nerven, denn in Dantes Inferno werden Sie durch die Hölle gehen (und das im wahrsten Sinne des Wortes).

Grundlage für dieses Spiel war Dante Alighieris klassisches mittelalterliche Abenteuer "Die Hölle", das die Reise des "Pilgers" beschreibt, der alle Regionen der Hölle durchwandern muß, um das Fegefeuer auf der anderen Seite zu erreichen. Die Reise findet ihren Höhepunkt in einer Konfrontation mit Luzifer höchstpersönlich.

So geht es dann auch in dem Programm ziemlich "rund". Sie steuern den Pilger und müssen mit ihm bis zur Region IX der Hölle innerhalb von 7 Tagen vordringen. Wenn Sie unterwegs versagen, ereilt Sie das gleiche Schicksal, das die armen Sünder in der jeweiligen Region der Hölle zu ertragen haben, in der Sie gerade waren.

Sie fangen im Wald an, nach Gegenständen zu suchen. Es liegen dort ein Geldbeutel und ein Seil. Man sollte sich übrigens schon hier mit der Steuerung der "Pilgerfigur" vertraut machen. Das Spiel kann über Tastatur oder Joystick gesteuert werden, wobei mit dem Joystick (ohne Feuerknopf) die vier Bewegungsrichtungen möglich sind. Hält man den Feuerknopf gedrückt, kann mit "Joystick nach links" der Gegenstand in der linken Hand benutzt werden, mit "Joystick nach rechts" der Gegenstand in der rechten, und mit "Joystick nach vorn" kann ein Gegenstand aufgenommen oder (!) abgelegt werden.

Nachdem Sie also die obengenannten Gegenstände aufgenommen haben, gehen Sie durch die Höhle im Wald in die Vorhölle. Der "Pilger" steht dann vor einem Fluß, hinter sich den Höhlenausgang. Jetzt kommt ein Fährmann vorbei. Bieten Sie dem den Geldbeutel an, sonst hat Ihre Reise schon in der Vorhölle ihr Ende

gefunden (Sie geraten in einen Hornissenschwarm). Der Fährmann nimmt Sie dann mit und läßt Sie vor dem Eingang zur Region 1 wieder aus dem Boot heraus.

Jetzt fängt es langsam an. spannend zu werden, denn jetzt muß mit äußerster Vorsicht zu Werke gegangen werden. Erfahrungsgemäß hält man sich in der Region 1 zunächst links und geht vorwärts, bis am linken Bildschirmrand ein grünes Teufelchen auftaucht. Den versucht man anzulocken und läuft vor ihm weg. Nach ein oder zwei kleinen Wegstrecken in eine Richtung hat man ihn dann abgehängt. Auf der rechten Seite der Region eins befindet sich eine Kreatur mit einer "ellenlangen" Zunge, der man besser aus dem Weg gehen sollte, da man sonst unweigerlich im Sumpf landet (auch eine Art, sich ganz schnell durch mehrere Regionen zu bewegen). Auch sonst scheint dieser Weg nicht ganz das Wahre zu sein, obwohl dort ein Fisch herumliegt, den man aufnehmen kann. Hier wird man allerdings meist vom grünen Teufelchen eingeholt und in ein Skelett verwandelt...

An DANTES INFERNO wird sicherlich jeder, der sich auf die "Höllenfahrt" einläßt, einiges zu knobeln haben. Die Grafik ist gut, der Sound ist jedoch eher als mäßig zu bezeichnen (nervt!). Die Spielmotivation würde ich gerade wegen der hohen Schwierigkeit und der "ungewöhnlichen" Sujet als recht hoch einstufen. Das Spiel wird sicherlich seine Freunde finden.

Martina Strack

Fortsetzung von Seite 3.

Ebenfalls neu bei DTM sind verschiedene Soft- und Hardwareprodukte für den AMIGA (es geht also aufwärts mit der kleinen "Freundin"!).

Eine 2 Megabyte-Speichererweiterung (Preis: leider knappe 2000,- DM) wird ebenso angeboten wie ein Database-Manager, eine AMIGA-Lisp-Version, Lattice C und MCC Pascal.

Info: DTM, Bornhofenweg 5, 6200 Wiesbaden, Tel. (06121) 407989

Quest for Tires

Au weia! Da hat doch die selbst programmierte Zeitmaschine einen ganz bösen Defekt gehabt. Gezischt, gekracht und gefunkt hat es in unserem VC 20. Nun stehe ich wie ein Wolperdinger (Südbayrisches Fabeltier) mit Astronautenanzug im Neandertal und fahre Einrad! Das finde ich schon gar nicht mehr komisch. Erst wird man aus seiner Zirkusnummer gerissen, in Weltraumkleidung gesteckt und muß in der Steinzeit auch noch diesen widerlichen Urvögeln ausweichen. Die Dinger sind wirklich gefährlich. Ducken und Springen muß ich – ich will zurück in die Zeit der Kernspaltung zu meinem geliebten VC 20, um ein neues Spiel zu programmieren.

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

1 REM*****
2 REM** (C) 1986 BY **
3 REM** THOMAS **
4 REM** ROEDERN **
5 REM*****
9 POKE56,28:POKE52,28:CLR:PRINT"(CLEAR BLACK RVS
ON SPACES)BITTE(SPACE)WARTEN(SPACES RVSOFF)"CHR$(
8)
10 DATA224,23,111,157,31,29,30,15
11 DATA0,0,128,192,240,240,192,0
12 DATA7,13,29,27,59,55,35,55
13 DATA,128,128,128,128,128,0
14 DATA27,27,17,13,30,30,30,12
15 DATA14,15,27,59,59,59,59,59
16 DATA59,49,59,31,31,31,27,27
17 DATA10,223,255,255,255,255,255,255
18 DATA64,235,255,255,255,255,255,255
19 DATA130,235,255,255,255,255,255,255
20 DATA240,240,120,120,60,28,6,0
21 DATA0,6,28,60,120,120,240,240
22 DATA255,252,248,240,240,224,224,224
23 DATA224,224,224,240,240,248,252,255
24 DATA255,63,31,15,15,7,7,7
25 DATA7,7,15,15,31,63,255
26 DATA60,28,56,24,24,8,8,0
27 DATA255,126,63,126,254,126,62,60
28 DATA0,30,30,76,255,28,0,0
29 DATA0,0,0,92,255,24,60,60
30 DATA247,227,227,195,195,129,129,0
31 DATA255,251,251,241,241,225,193,0
32 DATA255,255,255,239,227,195,193,0
33 DATA255,255,255,255,255,255,255,255
34 DATA0,0,0,16,0,0,0,0
35 DATA0,0,0,16,16,0,0,0
36 DATA0,44,62,60,60,60,126,255
37 DATA8,28,28,60,60,126,126,255
38 DATA255,255,255,255,255,255,211,130
39 DATA255,255,255,255,255,255,187,34
40 DATA60,66,153,161,161,153,66,60
41 DATA112,248,216,248,252,221,58,240
42 DATA0,0,0,0,0,0,0,0
43 DATA0,96,240,248,248,252,254,255
44 DATA0,0,0,0,0,0,0,0
45 DATA0,1,1,3,15,31,63,255
46 DATA96,144,8,4,4,2,2,2

```

Was – immer noch mini?

Satte 64 K für C 16/116

Hardwareservice des Tronic-Verlags

Unser 1. Angebot:

64 K-Speichererweiterung für C 16 + C 116

Bausatz – bestehend aus:

allen Bauteilen und ausführlicher Anleitung,
umschaltbar 64 K/16 K – geliefert in stabiler Hartbox.

Bitte Computertyp angeben!

Bestell-Nr. HWS-1-786 45, – DM + NN (bei Vorkasse + 2, – DM Versand)

Achtung:

- Bitte nur schriftlich bestellen!
- Bei Vorkasse sparen Sie unnötige Kosten!
- Auslandsversendungen nur gegen Vorkasse!
- Bitte nur Bestellungen separat vom Softwareservice!
- Bearbeitungsdauer bis zu 3 Wochen!

Tronic-Verlag · Hardwareservice · Pf 870 · 3440 Eschwege

```

47 DATA2,2,1,1,1,1,2,2
48 DATA0,0,1,2,2,4,4,4
49 DATA4,4,8,8,8,8,4,4
50 DATA4,4,4,2,2,1,0,0
51 DATA2,2,98,148,12,12,146,97
52 DATA33,33,33,33,33,33,82,140
53 DATA28,34,65,127,64,64,160,31
54 DATA12,18,33,65,65,65,146,12
55 DATA0,32,16,16,16,124,16,16
56 DATA16,16,16,16,16,16,168,70
57 DATA6,9,9,9,10,12,24,40
58 DATA72,8,43,28,8,8,8,8
59 DATA56,68,131,130,130,130,68,56
60 DATA0,174,113,33,32,32,32,32
61 DATA127,130,130,2,2,2,2,2
62 DATA240,0,0,0,0,0,0,0
63 DATA2,2,2,2,2,18,13
64 DATA16,0,16,16,16,16,40,199
65 DATA0,46,113,33,32,32,96,160
66 DATA28,34,65,255,64,64,32,31
67 DATA12,18,33,65,65,65,146,12
68 DATA0,0,1,3,3,1,0,0
69 DATA224,240,176,176,176,16,176
70 DATA0,1,3,3,3,1,1,1
71 DATA224,96,96,96,96,96,224
72 DATA63,64,170,170,186,170,64,63
73 DATA248,4,154,146,154,210,1,254
95 0=36877:POKE0+1,15:POKE0+2,29
100 FORX1=7168TO7455:READA:POKEX1,A:S=S+A:NEXT
101 IFS<>33558THENPRINT"(CLEAR DOWN2)?ERROR(SPAC
E)IN(SPACE)DATAS(SPACE)10-45":END
102 FORX1=7456TO7679:READA:POKEX1,A:S=S+A:NEXTX1
103 IFS<>44955THENPRINT"(CLEAR DOWN2)?ERROR(SPAC
E)IN(SPACE)DATAS(SPACE)46-73":END
105 PRINT"(CLEAR)":FORX1=1TO2000:NEXT
110 A$="THOMAS(SPACE)ROEDERN":GOSUB310
112 A$="(SPACE3)PRESENTS":GOSUB310:A$=""
114 PRINT"(CLEAR RED)":POKE0-8,255:GOSUB300:PRIN
T"(DOMNS BLUE SPACE)^(RVSON SPACE)BY(SPACE)THOMA
S(SPACE)ROEDERN":
116 FORX1=1TO2500:NEXT
117 POKE0-8,240:PRINTCHR$(14)
118 PRINT"(CLEAR DOWN WHITE RIGHT SH)IR(SPACE)SC
HREIBEN(SPACE)DAS(SPACES SJ)AHR(SPACE)2995(SPACE
)V.(SPACE SC)HR."
119 FORX1=38422TO38466:POKEX1,0:POKE0,160:FORX2=
1TO40:NEXT:POKE0,0:NEXT:FORX1=1TO2000

```



```

120 NEXT:PRINT"(DOHN RIGHT SE)S(SPACE)IST(SPACE
SI)HNEN(SPACE)GELUNGEN(SPACE)EINE(SPACE SZ)EITMA
SCHINE(SPACE)ZU"
122 PRINT"(RIGHT)BAUEN,(SPACE)MIT(SPACE)DER(SPAC
E SS)IE(SPACE4)JETZT(SPACE)IN(SPACE)DIE(SPACE SV
JER-(SPACE5)GANGENHEIT(SPACE)GEREIST"
124 PRINT"(RIGHT)SIND.(SD)ORT(SPACE)GIND(SPACE S
S)IE(SPACE)ZU(SPACE)EINEM(SPACE)JUNGEN(SPACE SM)
ANN(SPACE5)GEKOMMEN,(SPACE)DER(SPACE)GERADE"
125 PRINT"(RIGHT)DAS(SPACE SR SA SD SPACE)ERFUND
EN(SPACE)HAT."
127 PRINT"(RIGHT SE)R(SPACE)HAT(SPACE)ABER(SPACE
)EIN(SPACE7)GROSSES(SPACE SP)ROBLEM,(SPACE)BEI(S
PACE2)DEM(SPACE SS)IE(SPACE)IHM(SPACE)HELFFEN
129 PRINT"(RIGHT)SOLLEN,(SPACE)DENN(SPACE)ES(SPA
CE7)HAENGT(SPACE)NICHT(SPACE)NUR(SPACE)BEIN(SPA
CE SL)EBEN(SPACE)DAVON(SPACE)AB..."
130 PRINT"(DOHN SPACE RVSON SC ST SR SL RVSOFF S
PACE)DRUECKEN(BLACK)";
135 FORX1=38489T038905:POKEX1,0:POKED,160:POKED,
0:NEXT
136 WAIT653,4
138 PRINT"(CLEAR DOWN4 SI)ST(SPACE SI)HNEN(SPACE
)DER(SPACE)WEITERE(SPACE SS)PIELABLAUF(SPACE)BEK
ANNT(SPACE)?"
139 GETA$:IFA$="J"THEN200
140 IFA$<"N"THEN139
142 PRINT"(CLEAR RVSON SPACE3 SQ)UEST(SPACE)FOR(
SPACE ST)IRES(SPACE4 RVSOFF)";
144 PRINT"(DOHN SUBNSER(SPACE)JUNGER(SPACE SM)AN
N(SPACE5)HEISST(SPACE SF)REDDI(SPACE)UND(SPACE)D
AS(SPACE SP)ROBLEM(SPACE)IST(SPACE)FOLGENDES:"
146 PRINT"(SF)REDDIS(SPACE SQ)ELIEBTE(SPACE)IST(
SPACE2)VERSCHUNDEN(SPACE)UND(SPACE)NUN(SPACE2)V
ERSUCHT(SPACE)ER(SPACE)SIE(SPACE)ZU"
148 PRINT"FINDEIN.(SPACE SE)R(SPACE)HOERT(SPACE)I
MMERWIEDER(SPACE)NUR(SPACE)IHRE(SPACE7 SH)ILFERU
FE(SPACE)UND(SPACE)MACHT"
150 PRINT"NICHT(SPACE)DAHER(SPACE)MIT(SPACE)SEINE
R(SPACE SE)RFINDUNG(SPACE)AUF(SPACE2)DEN(SPACE S
W)EGSIE(SPACE2)ZU(SPACE)SUCHEN."
152 PRINT"(RED SD)ABEI(SPACE)MUESSEN(SPACE SS SI
SE SPACE)IHM(SPACE)HELFFEN,(SPACE)DENN(SPACE)ER(
SPACE)KANN(SPACE2)NOCH(SPACE)NICHT(SPACE SE)INRA
D(SPACE5)FAHREN!!"
154 PRINT"(DOHN2 BLACK SPACE RVSON SC ST SR SL R
VSOFF SPACE)DRUECKEN(BLACK)";WAIT653,4
156 PRINT"(CLEAR RVSON SPACE3 SQ)UEST(SPACE)FOR(
SPACE ST)IRES(SPACE4 RVSOFF)";
158 PRINT"(DOHN SD)AMIT(SPACE)ER(SPACE)SEINE(SPA
CE SG)E-(SPACE4)LIEBTE(SPACE)FINDET,(SPACE)MUESSE
EN(SS)IE(SPACE)IHN(SPACE)UEBER(SPACE SH)ULDEN,"
160 PRINT"(SB)AUMSTUEMPFE(SPACE)UND(SPACE6 SS)TE
INE(SPACE)HUEPFEN(SPACE)LASSEN.";
162 PRINT"(SV)OR(SPACE SF)LEDERMAEUSEN(SPACE)UND
(SPACE)VOR(SPACE ST)ROPFSTEINEN(SPACE)MUSS(SPACE
)ER(SPACE)SICH(SPACE)DUECKEN."
163 PRINT"(DOHN SS)TEUERUNG(SPACE SJ)OYSTICK":PR
INT"(DOHN SJ)OY.(SPACE)OBEN(SPACE2):(SPACE SS)PR
UNG"
164 PRINT"(SJ)OY.(SPACE)UNTEN(SPACE):(SPACE SD)U
CKEN":PRINT"(DOWN SF)EUERKNOFF(SPACE):(SPACE SP)
AUSE"
165 PRINT"(NACH(SPACE SP)AUSE:(ST)ASTE)"
166 PRINT"(DOHN2 SPACE RVSON SC ST SR SL RVSOFF
SPACE)DRUECKEN(BLACK)";WAIT653,4
168 PRINT"(CLEAR RVSON SPACE3 SQ)UEST(SPACE)FOR(
SPACE ST)IRES(SPACE4 RVSOFF)";
170 PRINT"(DOHN SD)A(SPACE SF)REDDIS(SPACE SG)EL
IEBTE(SPACE3)SCHON(SPACE)LANGE(SPACE)NICHTS(SPAC
E4)MEHR(SPACE)GEGESSEN(SPACE)HAT,"
172 PRINT"IST(SPACE)IHRE(SPACE SL)EBENSENERGIEKU
RZ(SPACE)VOR(SPACE)DEM(SPACE SE)NDE.(SPACE4 SD)E
SHALB(SPACE)MUSS(SPACE SF)REDDI"
174 PRINT"WAEHREND(SPACE)SEINER(SPACE SS)UCHE(SP
ACE)KLEINE,(SPACE2)ABER(SPACE)BEHR(SPACE4)STAERK
ENDE(SPACE SF)LIEGEN"
176 PRINT"FANGEN,(SPACE)DAMIT(SPACE)IHRE(SPACE4
SL)EBENSKRAFT(SPACE)WIEDER(SPACE4)STEIGT(SPACE)U
ND(SPACE)SIE(SPACE)NICHT"
177 PRINT"AN(SPACE SS)CHWAECHE(SPACE)STIRBT!"
178 PRINT"(DOHN2 SPACE RVSON SC ST SR SL RVSOFF
SPACE)DRUECKEN(BLACK)";WAIT653,4
180 PRINT"(CLEAR RVSON SPACE3 SQ)UEST(SPACE)FOR(
SPACE ST)IRES(SPACE4 RVSOFF)";
182 PRINT"(DOHN BLACK RIGHT SP SU SN SK ST SE):"

```

```

184 PRINT"(DOHN BLACK)PRO(SPACE)10(SPACE SM)ETER
":PRINT"(RED)+1(SPACE3 SP SK ST)." <78>
185 PRINT"(DOHN BLACK)FUER(SPACE)JEDES(SPACE SD)
UCKEN(SPACE)BEI(SPACE)EINEM(SPACE ST)ROPFSTEIN":
PRINT"(RED)+30(SPACE2 SP SK ST)." <169>
186 PRINT"(DOHN BLACK)FUER(SPACE)JEDES(SPACE SD)
UCKEN(SPACE)BEI(SPACE)EINER(SPACE SF)LEDERMAUS":
PRINT"(RED)+50(SPACE2 SP SK ST)." <193>
187 PRINT"(DOHN BLACK)PRO(SPACE)VERPASST(SPACE
SF)LIEGE":PRINT"(RED)-5(SPACE2 SP SK ST)." <206>
188 PRINT"(DOHN BLACK)PRO(SPACE SS)TURZ":PRINT"(
RED)-90(SPACE SP SK ST)." <170>
189 PRINT"(DOHN SPACE RVSON SC ST SR SL RVSOFF S
PACE)DRUECKEN(BLACK)";WAIT653,4 <146>
190 POKED-8,240:POKED-8,255 <36>
192 PRINT"(CLEAR DOWN3 RIGHT GREEN)!#{BLUE RVSON
SPACE}=(SPACE)MULDE":PRINT"(DOHN RIGHT BLACK)Z(
BLUE RVSON SPACE2)=(SPACE)BAUMSTUPF":PRINT"(DOHN
RIGHT BLACK){(BLUE RVSON SPACE2)=(SPACE)STEIN" <207>
194 PRINT"(DOHN RIGHT BLACK)S(BLUE RVSON SPACE2)
=(SPACE)FLEDERMAUS":PRINT"(DOHN RIGHT BLACK)Y(CL
UE RVSON SPACE2)=(SPACE)FLIEGE":PRINT"(DOHN RIGH
T BLACK)Q(DOHN LEFT)P(UP BLUE RVSON SPACE2)=(SPA
CE)TROPFSTEIN" <207>
199 PRINT"(DOHN4 SPACE RVSON RED)TASTE(SPACE)DRU
ECKEN":POKE198,0:WAIT198,1 <9>
200 POKED-8,255:PRINT"(CLEAR RVSON RED)ALSO(SPAC
E)VIEL(SPACE)SPASS(SPACE)BEI" <36>
204 GOSUB300:PRINT"(DOHN4 BLUE SPACE)^(RVSON SPA
CE)BY(SPACE)THOMAS(SPACE)ROEDERN":FORX1=1T03000:
NEXT <231>
205 POKED-8,240:PRINT"(CLEAR RED)BITTE(SPACE)WAR
TEN!":POKE198,3:POKE631,131:END <251>
300 RESTORE:FORX1=0T0287:READA:NEXT <151>
301 FORX1=7456T07679:POKEX1,0:NEXT <125>
302 PRINT"(HOME)"TAB(91)"%*(DOHN LEFT2)'%(DOHN L
EFT2)(!)*+.,(UP LEFT)-" <148>
303 PRINT"(DOHN3 RIGHT2)/(DOHN LEFT)012":PRINT"(
DOHN2 RIGHT12)34(DOHN LEFT2)56789" <47>
304 FORX1=7456T07679:READA:POKEX1,A:FORX2=1T035:
NEXTX2,X1:RETURN <220>
310 PRINT"(CLEAR DOWN10 RIGHT4 WHITE)"A$ <96>
311 FORX1=38624T038624+LEN(A$):POKEX1,0:POKED,20
0:FORX2=1T050:NEXT <222>
312 POKED,0:NEXT:FORX1=1T01500:NEXT:RETURN <167>
ENDE DES LISTINGS

```

Teil 2

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

0 POKES6,28:POKE52,28:CLR:GOTO90 <107>
1 POKEX211,X:POKE214,Y:SY588640:RETURN <222>
4 IFPEEK(8126+Z1)>32THEN300 <85>
6 RETURN <148>
8 IFPEEK(8105)=26ANDU>3THEN300 <186>
9 IFPEEK(8105+Z1)=26THEN300 <126>
10 IFPEEK(8126+Z1)>32THEN300 <91>
11 RETURN <153>
13 IFPEEK(8105+Z1)=27THEN300 <144>
14 IFPEEK(8126+Z1)=23THEN300 <108>
15 Y=14:X=1:GOSUB1:PRINTCHR*(20):POKE8009,28+INT
(RND(1)*2):POKE38729,0 <228>
17 I=I+1:IFI<10THENRETURN <246>
18 X=21-J:Y=15:GOSUB1:PRINT"(DOHN LEFT)P(SPACE
UP LEFT SPACE)":IFJ=0THENPOKE8009,23 <129>
19 J=J+1:IFJ=22THENX=0:Y=15:GOSUB1:PRINT"(SPACE2
DOWN LEFT2 SPACE2)":J=0:I=0:SC=SC+30:GOSUB33:RE
TURN <35>
22 IFJ>14ANDJ<18THENIFZ2=0THEN300 <27>
23 RETURN <165>
24 Q=0:Y=11:X=1:GOSUB1:PRINTCHR*(20):POKE7943,20
+INT(RND(1)*4):SC=SC+1:GOSUB33 <147>
25 POKEX38663,6:RETURN <80>
26 G=0:F=F+1:PRINT"(HOME)"TAB(129-F)F$:RETURN <57>
28 Y=16:X=20:GOSUB1:PRINT"(RED)?" <83>
29 IFH=55THENY=16:X=20:GOSUB1:PRINT"(SPACE2)":H=
0:E=E-35:GOSUB34:POKE36875,0:RETURN <191>
30 POKES,250-(H/2*3):RETURN <88>
31 E=E+50:GOSUB35:POKEB031-M,32:X=6:Y=15:GOSUB1:
PRINTF2#+C$:Z3=1:M=0:RETURN <99>
32 IFSC<0THENSC=0 <63>
33 PRINT"(HOME RIGHTS RED RVSON)"SC(LEFT SPACE)
":RETURN <253>

```


Leser- und Meckerecke

Checksummer C 16

Sehr geehrte Herren, (...wir haben auch Damen in unserem Verlag! Anm. d. R.)
 Ich war sehr erfreut, als in Ausgabe 6/86 der Checksummer für den C 16 abgedruckt wurde. Doch schon nach dem Abtippen ergab sich ein Problem. Die Checksummen wurden zwar angezeigt, aber auch nach dem Abschalten des Checksummers mit SYS 62158:SYS 33047 konnte ich das eingegebene Programm nicht absaven. Was sich nach SAVE-Vorgang auf der Kassette befand, war der Checksummer und nicht das eingetippte Programm. Vielleichtwissen Sie, wie man diesen Fehler vermeidet? Dann noch eine zweite Frage: Gibt es eine Pascal-

Version für den C-16?

Kotthaus, Braunschweig
Checksummer C 16
 Sehr geehrte Damen und Herren!
 Der oben genannte Checksummer arbeitet nur bei Neueingabe eines Programmes einwandfrei. Wenn ich ein Programm einschließlich Checksummer von der Datensette lade, um weitere Programmzeilen einzugeben, läßt sich der Checksummer nicht mehr starten.
 ...
 Ich wäre Ihnen dankbar, wenn Sie mir mitteilen könnten, wie man hier Abhilfe schaffen kann.
Ellerweg, Halle
 Die Redaktion:
 Dieser beiden Leseranfragen, möchte hier stellvertretend für eine Vielzahl

von Anfragen beantworten. Im Fall 1 wurde der Checksummer von Herrn Müller falsch bedient. Sie müssen sich klarmachen, daß das vorliegende Listing des Checksummers ein Basicprogramm darstellt, welches nach RUN im Speicher ein Maschinenspracheprogramm ablegt und aktiviert. Wollen Sie nun ein neues Listing eintippen, müssen Sie natürlich das ursprüngliche Basicprogramm des Checksummers aus dem Basicspeicher mit NEW entfernen. Diese Bemerkung hat leider bei älteren Ausgaben gefehlt, aber auch mit korrekter Angabe haben sich bei einigen Mißverständnisse eingestellt, so daß der Checksummer nach dem Eintippen mit NEW gelöscht wurde, bevor er

überhaupt abgesaved wurde (also alles für die Katz).

Zum zweiten Fall müssen wir leider sagen, daß es mit der vorliegenden Version nicht möglich ist, bereits fertige Programmteile einzuladen oder sogar einer erneuten Checksummenprüfung zu unterwerfen. Abhilfe kann hier nur eine MERGE-Routine schaffen, mit der Sie separat abgespeicherte Programmteile zu einem einheitlichen Programm zusammenfügen können. Unsere Programmabteilung arbeitet momentan an einer überarbeiteten Version, die diesen Mangel behebt und auch bei Überlangen Zeilen problemlos arbeitet. (Eine Pascal-Version für den C-16 ist uns nicht bekannt)

Uwe Knierim

Cave Fighter

Steuern Sie Ihr Raumschiff an allen Gefahren der Höhle (Laserstrahlen; Säuretröpfchen; feindliches Raumschiff) vorbei zu den 4 Bonussymbolen hin. Hat man dies überstanden, so dringt man in den Meteoritentunnel ein, wo man für eine bestimmte Zeit fliegenden Meteoriten ausweichen

muß. Ist auch dies geschafft, so beginnt das Ganze wieder von vorne. Während des Spiels ist außerdem darauf zu achten, daß man nicht gegen die Wände stößt oder die Zeit abläuft. Diese verringert sich von Level zu Level.

Gesteuert wird mit dem Joystick.

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

10 POKE36879,8:F=7168 <3>
11 PRINT "{CLEAR WHITE}" <95>
12 PRINT "{RVSON DOWN RIGHT6}ELCHSOFT{DOWN2 LEFT <174>
5}PRESENTS:{RVSOFF}" <14>
16 A$="===== <45>
100 FOR Y=1 TO 40 <17>
101 FOR I=0 TO 7:READ A:POKE P+I,A:NEXT I <180>
102 P=P+8:NEXT I <4>
110 POKE36869,255 <21>
150 PRINT "{CLEAR YELLOW DOWN RVSON}"A$;"C{SPACE <212>
}A{SPACE}V{SPACE}E{SPACE2}F{SPACE}I{SPACE}G{SPACE <189>
E}H{SPACE}T{SPACE}E{SPACE}R";A$;"{DOWN2 RVSOFF}" <9>
152 PRINT "{RIGHT7}DDDI{SPACE4}" <178>
154 PRINT "{RIGHT7 SPACE}B{SPACE3}V{SPACE2}" <230>
156 PRINT "{RIGHT7 SPACE2}DV{SPACE}DV{SPACE}" <142>
158 PRINT "{RIGHT7 SPACE2}DDDDDI" <222>
160 PRINT "{RIGHT7 SPACE2}DC{SPACE}" <21>
161 PRINT "{RIGHT7 SPACE}W{SPACE3}C{SPACE2}" <51>
162 PRINT "{RIGHT7}DDDI{SPACE4}" <93>
164 PRINT "{RVSON DOWN3}"A$;"{SPACE3}WRITTEN{SPACE <58>
E}BY{SPACE}M.ALT{SPACE3}";A$;"{RVSOFF}" <42>
165 POKE198,0:WAIT 198,1 <113>
170 PRINT "{CLEAR CYAN RVSON DOWN7 RIGHT4 SPACE}A <2>
NLEITUNG{SPACE}?{DOWN2 LEFT4} (Y/N)" <77>
171 GET A$ <125>
172 IFA$="Y" THEN PRINT "{CLEAR WHITE}":GOTO 180 <122>
173 IFA$="N" THEN 700 <219>
174 GOTO 171 <185>
180 PRINT "{RVSON}SIE{SPACE}MUESSEN{SPACE}IHR{SPACE <177>
CE}FLUG-{SPACE DOWN}OBJEKT{SPACE}ZU{SPACE}ALLEN{ <30>
SPACE}4{SPACE}BO-{SPACE DOWN}NUSSYMBOLEN{SPACE}S <1>
TEUERN{SPACE}!" <35>
181 PRINT "{RVSON DOWN}DABEI{SPACE}DUERFEN{SPACE} <29>
SIE{SPACE DOWN}NICHT{SPACE}GEGEN{SPACE}DIE{SPACE <91>
E}MAUER,{DOWN}DIE{SPACE}LASERSTRAHLEN,{SPACE}DEN <207>
" <42>
182 PRINT "{RVSON}SAEURETROPFEN{SPACE}ODER{SPACE} <59>
DAS{DOWN}FEINDLICHE{SPACE}FLUGOBJEKT{SPACE DOWN} <186>
STOESSEN." <6>
183 PRINT "{RVSON DOWN}AUSSERDEM{SPACE}IST{SPACE} <127>
DIE{SPACE}ZEIT{DOWN}ZU{SPACE}BEACHTEN{SPACE}!" <211>
185 POKE198,0:WAIT 198,1 <77>
186 PRINT "{RVSON CLEAR}HAT{SPACE}MAN{SPACE}DIESE <125>
{SPACE}STUFE{SPACE3 DOWN}SIEGREICH{SPACE}BEENDET <122>
,{SPACE}SO{SPACE DOWN}DRINGT{SPACE}MAN{SPACE}IN{ <219>
SPACE}DEN{SPACE}ME-" <185>
187 PRINT "{RVSON DOWN}TEORIDENTUNNEL{SPACE}EIN,{ <177>
SPACE}WO{DOWN}MAN{SPACE}METEORITEN,{SPACE}DIE{SP <125>
ACE3 DOWN}AUF{SPACE}EINEN{SPACE}ZUKOMMEN{SPACE}" <148>
188 PRINT "{RVSON DOWN}AUSWEICHEN{SPACE}MUSS.{SPA <59>
CE6 DOWN}HAT{SPACE}MAN{SPACE}AUCH{SPACE}DIES{SPA <74>
CE}ER-{SPACE DOWN}FOLGREICH{SPACE}ABGESCHLOS-" <76>
189 PRINT "{RVSON DOWN}SSEN,{SPACE}SO{SPACE}ERHAE <9>
LT{SPACE}MAN{SPACE2 DOWN}200{SPACE}BONUSPUNKTE." <237>
190 POKE198,0:WAIT 198,1 <11>
191 PRINT "{CLEAR RVSON}NACH{SPACE}AUFZAEHLUNG{SP <200>
ACE}DER{SPACE2 DOWN}PUNKTE{SPACE}WIRD{SPACE}DAS{ <01>
SPACE}ZEIT-{SPACE DOWN}LIMIT{SPACE}ETWAS{SPACE}V <4>
ERKLEI-" <6>
192 PRINT "{RVSON DOWN}NERT,{SPACE}UND{SPACE}DAS{ <127>
SPACE}GANZE{SPACE3 DOWN}BEGINNT{SPACE}WIEDER{SPA <211>
CE}VON{SPACE4 DOWN}VORNE{SPACE}!" <77>
193 PRINT "{RVSON DOWN2}STEUERUNG:JOYSTICK" <125>
195 POKE198,0:WAIT 198,1 <122>
700 PRINT "{CLEAR YELLOW DOWN3 RIGHT2 RVSON}FART{ <219>
SPACE}2{SPACE}IS{SPACE}LOADING{SPACE BLACK}":POK <185>
E631,131:POKE198,1 <177>
899 END <30>
900 DATA240,68,54,63,54,68,240,0 <1>
901 DATA15,34,108,252,108,34,15,0 <35>
902 DATA255,127,63,31,15,7,3,1 <29>
903 DATA255,254,252,248,240,224,192,128 <91>
904 DATA255,255,255,255,255,255,255,255 <207>
905 DATA28,28,28,28,28,28,28,28 <42>
906 DATA127,14,28,119,119,28,14,127 <59>
907 DATA16,146,84,56,56,84,146,16 <186>
908 DATA126,195,165,153,153,165,195,126 <6>
909 DATA128,192,224,240,240,224,192,128 <127>
910 DATA28,29,127,255,255,255,255,255 <211>
911 DATA255,255,255,255,255,127,29,28 <77>
912 DATA9,95,255,255,255,255,255,255 <125>
913 DATA255,255,255,255,255,255,9,9 <122>
914 DATA0,0,213,253,255,255,255,255 <219>
915 DATA255,255,255,255,253,213,0,0 <185>
916 DATA240,240,224,240,248,248,252,255 <177>
    
```


programme

```

917 DATA255,252,248,248,240,224,240,240
918 DATA248,240,192,240,248,224,224,240
919 DATA31,15,31,63,63,31,63,31
920 DATA255,63,31,31,15,7,15,15
921 DATA15,15,7,15,31,31,63,255
922 DATA128,192,224,240,248,252,254,255
923 DATA1,3,7,15,31,63,127,255
924 DATA0,0,3,7,31,63,63,127
925 DATA127,63,63,127,63,15,3,0
926 DATA0,192,240,252,254,252,252,254
927 DATA254,252,252,248,224,192,0,0
928 DATA63,31,15,31,63,31,15,31
929 DATA248,240,248,252,248,240,248,252
930 DATA126,60,56,60,124,60,28,124
931 DATA0,24,60,60,24,0,0,0
932 DATA0,0,0,0,0,0,0,0
933 DATA0,0,0,1,146,68,16,56
934 DATA0,3,7,15,31,63,63,31
935 DATA248,252,252,248,240,224,192,0
936 DATA31,63,63,31,15,7,3,0
937 DATA0,192,224,240,248,252,252,248
938 DATA0,3,7,15,31,63,63,31
939 DATA56,126,127,255,127,127,63,30
950 *****
951 * *
952 * CAVE FIGHTER *
953 * *
954 *****
955 * *
956 *(C) BY ELCHSOFT*
957 * *
958 *****
959 * *
960 * WRITTEN BY *
961 * *
962 * M.ALT *
963 * *
964 *****
ENDE DES LISTINGS

```

```

<111> 233 IFPEEK(A)<>32THEN790 <163>
<161> 235 RETURN <121>
<78> 250 IFS1=AORS2=AORS3=ATHEN790 <203>
<77> 251 IFS1=8114THENF=32:D=-D <86>
<191> 252 IFS1=8158THENF=5:D=-D <2>
<196> 255 RETURN <141>
<105> 280 FORT=1TO100:POKEV-1,G:SC=SC+1:PRINT"HOME RV
<122> SON)PUNKTE:";SC=G+1:NEXT:G=120:POKEV-1,0:CC=CC
<36> +1 <22>
<2> 281 IFCC=4THENS=SC+ZT:GOSUB950:OP=1:CC=0 <189>
285 RETURN <172>
290 IFPEEK(J+21)<>32ANDJ+21<>ATHENH=-23 <185>
292 IFPEEK(J-23)<>32ANDJ-23<>ATHENH=21 <96>
293 IFJ=ATHEN790 <251>
294 IFJ=7728ORJ=7750ORJ=7772THENPOKEJ,32:J=7744 <192>
295 RETURN <182>
300 IFK=ATHEN790 <4>
301 IFPEEK(K+22)<>32THENL=33 <103>
302 IFPEEK(K)<>32THENK=7841:L=31 <137>
305 RETURN <192>
310 ONVV6SUB312,313,314,315,315,315,315,315 <48>
311 RETURN <198>
312 ZT=270:RETURN <82>
313 ZT=240:RETURN <68>
314 ZT=210:RETURN <54>
315 ZT=180:RETURN <86>
350 U=INT(3*RND(1)+1):P=1 <122>
351 IFU=1THENW1=7879 <203>
352 IFU=3THENW1=7963 <139>
353 IFU=2THENR=A+11 <137>
355 RETURN <242>
400 POKER,32:R=R-1:P=P+1:IFR=ATHEN790 <151>
401 POKER,39:RETURN <120>
420 POKEW1,32:W1=W1+01:P=P+1:GOSUB 450:POKEW1,39
:RETURN <218>
440 POKEW1,32:W1=W1+02:P=P+1:GOSUB 450:POKEW1,39
:RETURN <253>
450 IFW1=ATHEN790 <127>
451 IFPEEK(W1+44)=4THEND1=-21:O2=-23 <148>
452 IFPEEK(W1-44)=4THEND1=23:O2=21 <232>
455 RETURN <86>
790 POKEV-1,0:POKEA,7:POKEV,200:FORT=15TOOSTEP-.
1:POKEV+1,7:NEXT:POKEV,0 <67>
800 PRINT"HOME RVSON DOWN6 RIGHT7)GAME(SPACE)OV
ER" <166>
805 POKE198,0:WAIT198,1:RUN <248>
900 PRINT"CLEAR WHITE)DDDDDDDDDDDDDDDDDDDDDDDD"; <207>
901 PRINT"DDMMKMKOMKMKOMKMMKMMK"; <181>
902 PRINT"DD(SPACE20)"; <6>
903 PRINT"O(SPACE20)H"; <42>
904 PRINT"R(SPACE2)%(SPACE3)%(SPACE5)"; <205>
905 PRINT"J(SPACE)H(SPACE)XLLNNJDDNNJDDLNN"; <86>
906 PRINT"R(SPACE3)YKMMDDDDDDDDMMDD"; <86>
907 PRINT"R(SPACE7)SOKKODM(SPACE4)TD"; <74>
908 PRINT"P(SPACE20)T"; <76>
909 PRINT"DP(SPACE7)NLLJJ(SPACE5)S"; <47>
910 PRINT"DDLNNL(SPACE)S+DDDDDDDL(SPACE3)S"; <51>
911 PRINT"DDDDDDJDDDDDDDD(SPACE)H(SPACE)\"; <71>
912 PRINT"DDKMMDDDDMMDDMMDD(SPACE3)S"; <144>
913 PRINT"DD(SPACE5)^(SPACE3)^(SPACE3)^(SPACE5)\
"; <221>
914 PRINT"O(SPACE4)^(SPACE)^(SPACE)^(SPACE)^(SPA
CE)^(SPACE)^(SPACE5)S"; <33>
915 PRINT"J(SPACE4)^(SPACE)^(SPACE)^(SPACE)^(SPA
CE)^(SPACE)^(SPACE5)U"; <118>
916 PRINT"R(SPACE4)^(SPACE3)^(SPACE3)^(SPACE6)UD
"; <85>
918 PRINT"R(SPACE)H(SPACE)XNNNDLNLNLJLDD"; <106>
919 PRINT"J(SPACE3)YOMKMKOMKMMKMMOM"; <103>
920 PRINT"R(SPACE21)"; <126>
921 PRINT"P(SPACE21)"; <121>
922 PRINT"DP(SPACE20)"; <22>
923 PRINT"DDLNNLNJNNLNJLNJLNN"; <234>
925 POKE185,12 <18>
940 RETURN <61>
950 PRINT"CLEAR";X=22:Y=0:B$="DDDDDDDDDDDDDDDD
DDDDDD"; <110>
951 A$(0)="OKMKMKMKMKMKMKMMKMMK":A$(1)="JLNNJL
LJLNNLNJLNNLNJL"; <104>
952 S$="HOME WHITE DOWN7)"; <243>
953 PRINT"HOME";FORT=1TO7:PRINTB$;NEXT:PRINT"
DOWN7";FORT=1TO7:PRINTB$;NEXT:PRINT"
HOME RVSON)PUNKTE:";SC <192>
955 A=7909:ZZ=0 <198>
956 RETURN <77>
ENDE DES LISTINGS

```

Teil 2

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

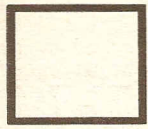
5 ZT=300 <249>
10 V=36877:POKEV+2,8:POKEV-8,255:POKEV+1,15:POKE
37139,0:A=8140:D=22:F=5:G=120:ZZ=1:VV=-1 <240>
11 S1=8158:S2=8110:S3=8150:H=21:J=7744:K=7841:L=
31:O1=23:O2=-01 <167>
15 GOSUB900 <94>
17 VV=VV+1:GOSUB310 <160>
110 POKEA,1:POKES1,5:POKES2,5:POKES3,5:POKEJ,6:P
OKEK,L <233>
111 PRINT"HOME RVSON)PUNKTE:";SC <7>
180 IFOP=ATHEN190 <62>
181 PRINTS*RIGHT$(A$(0),X)LEFT$(A$(0),Y); <58>
182 PRINTS*+*(DOWN7)"RIGHT$(A$(1),X)LEFT$(A$(1),
Y); <68>
183 X=X-1:Y=Y+1:IFX=0THENX=22 <219>
184 IFY=22THENY=0 <102>
185 O=O+1:IFO=200THENA=8140:OP=0:GOSUB280:GOSUB2
80:O=0:CC=0:GOTO15 <158>
186 IFF=0THENGOSUB350 <9>
187 ONUGOSUB420,400,440 <131>
188 IFF=19THENPOKER,32:POKEW1,32:F=0 <28>
189 GOTO199 <55>
190 POKES1,F:POKES2,F:POKES3,F:S1=S1-D:S2=S2+D:S
3=S3-D:GOSUB250 <210>
191 POKES3,5:POKES2,5:POKES1,5 <53>
192 POKEJ,32:J=J+H:GOSUB290:POKEJ,6 <69>
194 POKEK,32:K=K+22:GOSUB300:POKEK,L <21>
195 ZT=ZT-1:PRINT"HOME RVSON RIGHT16)ZT"(LEFT
SPACE)";IFZT=0THEN790 <81>
199 J9=PEEK(37137):POKE37154,127:J8=PEEK(37152):
POKE37154,255 <236>
200 IFDP=1THEN203 <49>
201 IF(J9AND16)=0THENZ=-1:ZZ=1:GOSUB220 <57>
202 IF(J8AND128)=0THENZ=1:ZZ=0:GOSUB220 <42>
203 IF(J9AND8)=0THENZ=-22:GOSUB220 <32>
204 IF(J9AND4)=0THENZ=-22:GOSUB220 <119>
210 GOTO180 <37>
220 POKEV-1,177:POKEA,32:A=A+Z:GOSUB230:POKEA,ZZ
:POKEV-1,0 <210>
225 RETURN <111>
230 IFPEEK(A)=8THENGOSUB280:RETURN <113>

```

```

<249>
<240>
<167>
<94>
<160>
<233>
<7>
<62>
<58>
<68>
<219>
<102>
<158>
<9>
<131>
<28>
<55>
<210>
<53>
<69>
<21>
<81>
<236>
<49>
<57>
<42>
<32>
<119>
<37>
<210>
<111>
<113>

```

Mc Stoney

Mc Stoney in the Caves

Pssst: Kennen Sie "Boulder Dash"? Und haben Sie eine tolle Version für den C16 bisher vermisst? Hier ist sie!

Superschnell, saubere Grafik, eben alles, was man sich von einem Profi-Programm wünschen kann, hat dieses Programm.

Worum es geht? Ist doch glasklar!

Sie sollen in den 20 Höhlen eine bestimmte Anzahl von Diamanten sammeln und dann die jeweilige Höhle durch den Herz-Ausgang verlassen.

Auf Ihrem Spielfeld (also in der Höhle) befinden sich folgende Sachen und Typen:

Blinkies:

Läßt man einen Stein auf sie fallen, so verwandeln sie sich in mehrere Diamanten. Eine Berührung allerdings mit ihnen ist tödlich.

Glibbermasse:

Sie verwandelt sich in Steine, wenn eine bestimmte Anzahl von Feldern eingenommen worden ist.

Diamanten:

müssen eingesammelt werden und bringen die Punkte.

Steine:

sollte man sich nicht auf den Kopf fallen lassen.

Mc Stoney:

Er ist der Held unseres Spieles und muß die schwierigen Aufgaben meistern.

Zum Programm:

Es besteht aus zwei Teilen:

1. Teil: Basic-Programm, das nur die Erläuterungen enthält und für das Spiel unbedeutend ist.
2. Teil: M-Code: belegt den Speicher von HEX 1001 bis HEX 3C1E.

Beude Teile sind einfach mit RUN zu starten.

Eingabehinweise:

Teil 1 eingeben und unter dem Namen "Mc Stoney" abspeichern.

Teil 2 im Monitor-Modus eingeben und mit S*GAME",8,1001,3C1E abspeichern(bei der Verwendung einer Datensette ist statt der 8 eine 1 einzugeben!)

Das Spiel wird gesteuert mit einem Joystick in Port 1 oder mit den Tasten F, X, (Punkt), (Doppelpunkt) und SHIFT.

Am unteren Rand zeigt die Statuszeileolgendes an:

- SC = Score
- HI = High-Score
- LI = Anzahl der verbleibenden Leben
- TI = Zeit
- DI = Diamanten (zu sammeln und bereits gesammelt)
- CA = Die Nummer der Höhle

Man erhält für jeden eingesammelten Diamanten 25 Punkte und außerdem noch die Restzeit des Levels.

in the Caves

```
>1001 0d 10 c2 07 9e 20 31 35
>1009 33 36 30 00 00 00 38 0c
>1011 01 0c 0d 0d 04 0d 0d 0d
>1019 0d 0c 0d 0c 0d 0d 0d 0d
>1021 0d 0d 0d 0d 0d 0d 0d 0d
>1029 0d 0d 0c 0d 0d 0d 0d 01
>1031 0c 0d 0d 0c 0c 0c 0d 0c
>1039 0d 0d 0d 0d 0d 0d 0d 0d
>1041 0d 0d 0d 0d 0d 0d 0c 0d
>1049 0d 0c 0d 0d 0d 0c 0d 0d
>1051 0d 0d 0d 0c 0d 0d 00 0d
>1059 0c 0d 0d 0d 0d 0d 0d 0d
>1061 0d 0d 0d 0d 0d 0d 0d 0d
>1069 0d 0d 0d 0d 0d 0d 0d 0d
>1071 0d 0d 0d 0c 0d 0d 0d 0c
>1079 0d 0d 0d 0c 0d 0d 0d 0c
>1081 0c 0d 0d 0c 0d 0d 0d 0d
>1089 0d 0d 0d 0d 0c 0d 0d 0d
>1091 01 0d 0d 0d 0d 0c 0d 0d
>1099 0d 0d 0d 0d 0d 0c 0d 0d
>10a1 0d 0d 0c 0d 01 0d 0c 0c
>10a9 0c 0d 0d 0d 0d 0d 0d 01
>10b1 0c 0d 0c 0d 0d 0d 0d 0d
>10b9 0d 0d 0d 0d 0c 0c 0d 0d
>10c1 0d 0d 0d 06 0f 0d 01 01
>10c9 0d 04 0e 01 0d 0d 0d 0d
>10d1 0d 0d 0d 0d 0d 0d 0f 0e
>10d9 0e 0e 0e 0e 0e 0e 0e 0e
>10e1 0e 0e 0e 0e 0e 0e 0f 0e
>10e9 0f 0e 0f 0e 0f 0e 0f 0e
>10f1 0f 0e 0f 0e 0f 0e 0f 0e
>10f9 0f 0e 0f 0e 0f 0e 0f 0e
>1101 0f 0e 0f 0e 0f 0e 0f 0e
>1109 0f 0e 0f 0e 0f 0e 0f 0e
>1111 0f 0e 0f 0e 0f 0e 0f 0e
>1119 0f 0e 0f 0e 0f 0e 0f 0d
>1121 0f 0e 0f 0e 0f 0e 0f 0e
>1129 0f 0e 0f 0e 0f 0e 0f 0e
>1131 0f 0d 0f 0e 0f 0e 0f 0e
>1139 0f 0e 0f 0e 0f 0e 0f 0e
>1141 0f 0d 0f 0d 0f 0e 0f 0e
```

```
>1149 0f 0e 0f 0e 0f 0e 0f 0e
>1151 0f 0e 0f 0d 02 0d 02 0d
>1159 02 0d 02 0d 02 0d 02 0d
>1161 02 0d 02 0d 02 0d 0f 0e
>1169 01 0e 01 0e 01 0e 01 0e
>1171 01 0e 01 0e 01 0e 01 0f
>1179 0b 0d 0f 0d 0d 0d 04 0c
>1181 0c 0c 0c 0c 0c 0d 0d 0d
>1189 0d 0d 0d 0d 0d 0d 0d 0c
>1191 0c 0c 0c 0c 0c 0c 0c 0c
>1199 0c 0d 0d 0d 0d 0d 0d 0d
>11a1 0c 0c 0d 0d 0d 0d 0d 0d
>11a9 0d 0d 0c 0c 0d 0d 0d 0d
>11b1 0d 0c 0c 0d 0d 01 01 01
>11b9 01 01 01 0d 0d 0c 0c 0d
>11c1 0d 0d 0d 0c 0c 0d 0d 01
>11c9 01 01 01 01 01 0d 0d 0c
>11d1 0c 0d 0d 0d 0d 0c 0c 0d
>11d9 0d 01 01 01 01 01 01 0d
>11e1 0d 0c 0c 0d 0d 0d 0d 0c
>11e9 0c 0d 0d 0d 0d 0d 0d 0d
>11f1 0d 0d 0d 0c 0c 0d 0d 0d
>11f9 0d 0d 0c 0c 0d 0d 0d 0d
>1201 0d 0d 0d 0d 0c 0c 0d 0d
>1209 0d 0d 0d 0d 0d 0c 0c 0d
>1211 0f 0c 0c 0c 0c 0c 0c 0d
>1219 0d 0d 0d 0d 0d 0d 0d 0d
>1221 0d 0c 0c 0c 0c 0c 0c 0d
>1229 0d 0d 0d 0d 0d 12 10 10
>1231 10 10 0e 05 0e 10 10 10
>1239 10 10 0e 10 10 10 10 10
>1241 0f 0f 0e 0f 0f 0f 0f 0f
>1249 0f 0e 0f 0f 0e 0f 0e 0f
>1251 0f 0e 0f 0f 0e 0f 0e 01
>1259 0e 0f 0e 0f 01 0f 0e 0e
>1261 01 0f 0f 0e 0f 0f 0e 0e
>1269 0f 0f 0e 0f 0e 0f 0f 0e
>1271 0f 0e 0f 0e 0f 0f 0e 01
>1279 0e 0f 0e 0f 0e 0f 0f 0f
>1281 0e 0f 0f 0e 0e 0f 0f 0e
>1289 0f 01 01 0f 0e 0f 0f 0f
```

```
>1291 0f 0e 0f 0f 0e 0e 0e 0e
>1299 0f 0e 0f 0f 0e 0e 0e 0e
>12a1 0e 0e 01 0f 0f 0e 0f 0e
>12a9 0e 0e 0f 0f 0f 0f 0e
>12b1 0f 0e 0e 0f 01 0e 0f 0f
>12b9 0e 0e 0f 0e 0f 0f 0f 0e
>12c1 0e 01 0f 0e 01 01 0f 0f
>12c9 0f 0f 0f 0f 0e 0f 0f 01
>12d1 0f 0f 0f 0f 0f 0f 01 01
>12d9 0e 0f 0e 0f 0e 0f 0f 0f
>12e1 0e 0f 0e 0c 0c 0c 0c 0c
>12e9 05 0c 0c 0c 0c 0c 0c 0c
>12f1 0c 0c 0c 0c 0c 0d 0d 0d
>12f9 0d 0d 0d 0d 0d 0d 0d 0d
>1301 0d 0d 0d 0d 0d 0d 0d 07
>1309 0f 0f 0f 0f 0f 0f 0f 0f
>1311 0f 0f 0f 0f 0f 0f 0f 0f
>1319 0f 0f 0d 0d 0d 0d 0d 0d
>1321 0d 0d 0d 0d 0d 0d 0d 0d
>1329 0d 0d 0f 0f 0d 07 0f 0f
>1331 0f 0f 0f 0f 0f 0f 0f 0f
>1339 0f 0f 0f 0d 0d 0f 0d 0f
>1341 0d 0d 0d 0d 0d 0d 0d 0d
>1349 0d 0d 0d 0d 0f 0d 0f 0f
>1351 0d 0f 0d 07 0f 0f 0f 0f
>1359 0f 0f 0f 0f 0f 0f 0f 0d
>1361 0f 0f 0d 0f 0d 0f 0d 0d
>1369 0d 0d 0d 0d 0d 0d 0d 0d
>1371 0d 0d 0d 0f 0f 0f 0f 0f
>1379 0f 0f 0f 0f 0f 0f 0f 0f
>1381 0f 0f 0f 0f 0f 0f 0c 0c
>1389 0c 0c 0c 0c 0c 0c 0c 0c
>1391 0c 0c 0c 0c 0c 0c 0f 12
>1399 0f 0f 0f 0f 0f 0f 05 0e 0e
>13a1 0c 0c 0c 0c 0c 0c 0c 0c
>13a9 0c 0c 0f 0e 0e 0e 0e 0e
>13b1 0e 0c 01 0c 0c 01 0c
>13b9 0c 0c 0c 0e 0f 0e 0c 01
>13c1 0c 0c 0c 0c 0c 0c 0c 0c
```



>13c9 0c 0c 01 0c 0e 0f 0f 0e
>13d1 0c 0c 0c 0c 0c 0c 01 0c
>13d9 0c 0c 0c 0c 0c 0e 0f 0f
>13e1 0f 0e 0e 0c 0c 0c 0c 01
>13e9 0c 0c 0c 0c 0c 0c 0e 0f
>13f1 0f 0f 0f 0d 0f 0e 0e 0e
>13f9 0d 0d 0d 0d 0d 0e 0e 0e
>1401 0f 0f 0f 0f 0f 0e 0f 0d
>1409 0d 0d 0d 0d 0d 0d 0d 0d
>1411 0d 0d 0f 0f 0f 0f 0f 0e
>1419 0f 0d 0d 0d 0d 0d 0f 0d
>1421 0d 0d 0d 0d 0f 0f 0f 0f
>1429 0f 0e 0f 0d 0d 0d 0d 0d
>1431 0d 0d 0d 0d 0d 0d 0f 0f
>1439 0f 0f 0f 02 0e 0f 0d 0d 0d
>1441 0d 0d 0d 0d 0d 0d 0d 0d
>1449 0f 0f 0f 0f 05 0f 0f 0f
>1451 0f 0f 05 0e 0c 0c 0c 0c
>1459 0c 0c 0c 0c 0c 0c 0c 0c
>1461 0e 0e 0e 0e 0e 0e 0c 0c
>1469 01 0c 0c 01 0c 0c 0c 0c
>1471 0e 0f 0e 0c 01 0c 0c 0c
>1479 0c 0c 0c 0c 0c 0c 0c 01
>1481 0c 0e 0f 0f 0e 0c 0c 0c
>1489 0c 0c 0c 01 0c 0c 0c 0c
>1491 0c 0c 0e 0f 0f 0f 0e 0e
>1499 0c 0c 0c 0c 01 0c 0c 0c
>14a1 0c 0c 0c 0e 0f 0f 0f 0f
>14a9 0d 0f 0e 0e 0e 0d 0d 0d
>14b1 0d 0d 0d 0e 0e 0f 0f 0f
>14b9 0f 0f 0e 0f 0d 0d 0d 0d
>14c1 0d 0d 0d 0d 0d 0d 0d 0f
>14c7 0f 0f 0f 0f 0e 0f 0d 0d
>14d1 0d 0d 0d 0d 0d 0d 0d 0d
>14d9 0d 0f 0f 0f 0f 0e 0f
>14e1 0d 0d 0d 0d 0d 0d 0d 0d
>14e9 0d 0d 0d 0f 0f 0f 02
>14f1 0e 0f 0d 07 0f 07 0f 07
>14f9 0f 07 0f 07 0f 0f 0f 0f
>1501 0f 14 09 0f 0f 0f 0d 05
>1509 0d 0d 0d 0d 0d 0d 0c 0d
>1511 09 0f 0f 0f 0f 0f 0f 0f
>1519 0d 0c 0d 0d 0d 0d 0d 0d
>1521 0c 0c 0c 0f 0f 0f 0e 0e
>1529 0e 0e 0c 0c 0d 0c 0c 0d
>1531 0c 0d 0d 0d 0e 0e 0e 0e
>1539 09 0f 0f 0f 0d 0d 0d 0f
>1541 0c 0d 0d 0c 0d 0d 09 0f
>1549 0f 0f 0f 0f 0f 0f 0c 0d
>1551 0c 0c 0d 0d 0d 0d 0d 0d
>1559 0f 0f 0f 0f 0e 0e 0e 0e
>1561 0d 0c 0c 0d 0d 0d 0d 0c
>1569 0d 0d 0e 0e 0e 0e 09 0f
>1571 0f 0f 0d 0d 0d 0c 0c 0c
>1579 0c 0c 0d 0c 09 0f 0f 0f
>1581 0f 0f 0f 0f 0d 0c 0d 0f
>1589 0d 0c 0c 0d 0c 0d 0f 0f
>1591 0f 0f 0e 0e 0e 0e 0c 0d
>1599 0d 0d 0d 0d 0c 0d 0d 0d
>15a1 0e 0e 0e 0e 0d 0d 0c 0d
>15a9 0d 0c 0d 0d 0d 0c 0c 0c
>15b1 0d 0d 0d 0d 0d 12 0c
>15b9 01 0c 0d 0d 04 0d 0d 0d
>15c1 0d 0d 0d 0d 0d 0d 0d 0d
>15c9 0d 0d 0d 0d 0d 0d 0d 0d
>15d1 0d 0d 0c 0d 0d 0d 0d 01
>15d9 0c 0d 0d 0c 0c 0c 0d 0d
>15e1 0d 0d 0d 0d 0d 0d 0d 0d
>15e9 0d 0d 0d 0d 0d 0d 0c 0d
>15f1 0d 0c 0d 0d 0c 0c 0d 0c
>15f9 0d 0d 0c 0c 0d 0d 0d 0d
>1601 0c 0d 0d 0d 0d 0d 0d 0d
>1609 0d 0d 0d 0d 0d 0d 0d 0d
>1611 0d 0d 0d 0d 0d 0d 0d 0d
>1619 0d 0d 0c 0d 0d 0d 0d 0c
>1621 0d 0d 0c 0c 0d 0d 0d 0c
>1629 10 10 10 10 10 10 10 10
>1631 0d 0d 0d 0d 0c 0d 0d 0d
>1639 01 0d 07 0d 01 0d 07 0d
>1641 01 0d 07 0d 0d 0c 0d 0d
>1649 0d 0d 0c 0d 01 0d 0c 0c
>1651 0c 0d 0d 0d 0d 0d 0d 01
>1659 0c 0d 0c 0d 0d 0d 0d 0d
>1661 0d 0d 0d 0d 0c 0c 0c 0d
>1669 0d 0d 0d 0d 0d 0d 0d 0d

>1671 0d 05 0d 0d 0d 0d 0d 0d
>1679 0d 0d 0d 0d 0d 0c 0d
>1681 0c 0d 0c 0d 0c 0d 0c 0d
>1689 0c 0d 0c 0d 0c 0d 0c 0d
>1691 0d 0d 0d 0d 0d 0d 0d 0d
>1699 0d 0d 0d 0d 0d 0d 0d 0d
>16a1 0d 0d 0d 0c 0d 0c 0d 0c
>16a9 0d 0c 0d 0c 0d 0c 0d 0c
>16b1 0d 0c 0d 0d 0c 0d 0d 0d
>16b9 0d 0d 0d 0d 0d 0d 0d 0d
>16c1 0d 0d 0d 0d 0d 0d 0f 0f
>16c9 0f 0f 0f 0f 0f 0f 0f 0f
>16d1 0f 0f 0f 0f 0f 0f 0f 0f
>16d9 0f 0f 0f 0f 0f 0f 0f 0f
>16e1 0f 0f 0f 0f 0f 0f 0f 0f
>16e9 0f 0f 0f 0f 0f 0f 0f 0f
>16f1 0f 0f 0f 0f 0f 0f 0f 0f
>16f9 0f 0f 0f 0f 0f 0f 0f 0f
>1701 0f 0f 0f 0f 0f 0f 0f 0f
>1709 0f 0f 0f 0f 0f 0f 02 0f
>1711 07 0f 02 0f 07 0f 02 0f
>1719 07 0f 02 0f 07 0f 02 0f
>1721 0a 09 0f 0f 0f 0f 0f 0f
>1729 0f 0f 0f 0f 0f 0f 0f 0f
>1731 0f 0f 0f 0f 0f 0f 0f 0f
>1739 0f 0f 0f 0f 0f 0f 0f 0f
>1741 0f 0f 0f 0f 0f 0f 0f 0f
>1749 0f 0f 0f 0f 0f 0c 0c 0f
>1751 0f 0f 0f 0f 0f 0f 0f 0f
>1759 0f 0f 0f 0f 0f 0f 0f 0d
>1761 0d 0f 0f 0f 0f 0f 0f 0f
>1769 0f 0f 0f 0f 0f 0f 0f 0f
>1771 0f 0f 0f 0f 0f 0f 0f 0f
>1779 0f 0f 0f 0f 0f 0f 0f 0f
>1781 0f 0f 0f 0f 0f 0f 0f 0f
>1789 0f 0f 0f 0f 0f 0f 0f 0f
>1791 0f 0f 0f 0f 0f 0f 0f 0f
>1799 0f 0f 0f 0f 0f 0f 0f 0f
>17a1 0f 0f 0f 0f 0f 0f 0f 0f
>17a9 0f 0f 0f 0f 0f 0f 0f 0f
>17b1 0f 0f 0f 0f 0f 0f 0f 0f
>17b9 0f 0f 0f 0f 0f 0f 0f 0f
>17c1 0f 0f 0f 0f 0f 0f 0f 0f
>17c9 0f 0f 0f 0f 0f 0f 0f 0f
>17d1 0f 0f 0f 0f 0f 06 0d 0d
>17d9 0d 0d 0d 0d 0d 0d 0d 0d
>17e1 0d 0d 0d 0d 0d 0d 0d 0d
>17e9 0d 0c 0c 0c 0d 0e 0e 0e
>17f1 0e 0e 0e 0e 0e 0d 0c 0c
>17f9 0c 0d 0d 0d 0d 0d 0e 0e
>1801 0d 0d 0d 0e 0d 0d 0e 0e
>1809 0d 0d 0d 0d 0d 0d 0d 0d
>1811 0e 0d 0d 0d 0d 0e 0d 0d
>1819 0e 0e 0d 0d 0d 0d 0d 0d
>1821 0d 0d 0e 0d 01 01 0d 0e
>1829 0d 0d 0e 0e 0d 0d 0d 0d
>1831 0d 0d 0d 0d 0e 0d 0d 0d
>1839 0d 0e 0e 0e 0e 0d 0d 0d
>1841 0d 07 0d 0d 0d 0d 0e 0d
>1849 0d 0d 0d 0e 0e 0d 0d 0d
>1851 0d 0d 0d 0c 0d 0d 0d 0d
>1859 0e 0e 0d 0d 0d 0e 0d 0e
>1861 0d 0d 0d 0d 0d 0d 0d 0d
>1869 0d 0d 0d 0e 0e 0e 0e 0e
>1871 0d 0d 0d 0e 0e 0d 0d 0d
>1879 0d 0d 0d 0d 0d 0d 0d 0d
>1881 0d 0d 0d 0d 0d 0d 0d 0d
>1889 0d 0d 0b 09 0e 01 0f 0e
>1891 0f 0f 0e 01 0f 0e 01 0f
>1899 0e 01 0f 0e 01 0e 0e 0e
>18a1 0d 0e 0e 0d 0e 0e 0d 0e
>18a9 0e 0d 0e 0e 0d 0e 0d 0f
>18b1 0f 0f 0f 0f 0f 0f 0f 0f
>18b9 0f 0f 0f 0f 0f 0f 0f 0f
>18c1 0f 0e 0f 0e 0e 0e 0e 0f
>18c9 0e 0e 0e 0e 0f 0e 0e 0e
>18d1 0e 0f 0e 0f 0f 0f 0f 01
>18d9 0e 0f 0f 0f 01 0e 0f 0f
>18e1 0f 01 0e 0f 0f 01 0e 0f
>18e9 0e 0e 0e 0e 0f 0e 0e 0e
>18f1 0e 0f 0e 0e 0e 0e 0f 0e
>18f9 0e 0f 0e 0f 0f 0e 0f 0e
>1901 0f 0f 0e 0f 0e 0f 0f 0e
>1909 0f 0f 0f 0f 0e 0e 0f 0f
>1911 0f 0e 0e 0f 0f 0f 0e 0e

>1919 0f 0f 0f 0e 0e 0f 0f 0f
>1921 0f 0e 0f 0f 0f 0f 0e 0f
>1929 0f 0f 0f 0e 0f 01 0f 0f
>1931 0e 0f 0e 0e 0f 0e 0f 0e
>1939 0e 0f 0e 0f 0e 0e 0f 0a
>1941 0d 0d 0d 0d 0d 0d 0d 0d
>1949 0d 0d 0d 0d 0d 0d 0d 0d
>1951 0d 0d 0d 0c 0c 0c 0c 0c
>1959 0c 0c 0c 0c 0c 0c 0c 0c
>1961 0c 0c 0c 0c 0c 0c 0d 0d
>1969 0d 0d 0d 0d 0d 0d 0d 0d
>1971 0d 0d 0d 0d 0d 0d 0d 0c
>1979 0c 0c 0c 0c 0c 0c 0c 0c
>1981 0c 0c 0c 0c 0c 0c 0c 0c
>1989 0d 0d 0d 0d 0d 0d 0d 0d
>1991 0d 0d 0d 0d 0d 0d 0d 0d
>1999 0d 0d 07 07 0f 0f 0f 0f
>19a1 0f 0f 0f 0f 0f 0f 0f 0f
>19a9 0f 0f 0f 0f 0f 0f 0f 0f
>19b1 0f 0f 0f 0f 0f 0f 0f 0f
>19b9 0f 0f 0f 0f 0f 0f 0f 0f
>19c1 0f 0f 0f 10 10 10 10 10
>19c9 10 10 10 10 10 10 10 0f
>19d1 0f 0f 0f 0f 0f 0f 0f 0f
>19d9 0f 0f 0f 0f 0f 0f 0f 0f
>19e1 0f 0f 02 0f 0f 0f 0f 02
>19e9 0f 0f 0f 02 0f 0f 0f 0f
>19f1 0f 02 0f 0f 0a 0c 01 0c
>19f9 0e 0d 04 0d 0e 0d 0d 0d
>1a01 0d 0c 0d 0e 0d 0d 0d 0d
>1a09 0d 0d 0d 0d 0d 0d 0d 0d
>1a11 0c 0d 0d 0d 0d 01 0c 0d
>1a19 0d 0c 0c 0c 0e 0c 0d 0d
>1a21 0e 0d 0d 0e 0d 0d 0d 0e
>1a29 0d 0d 0d 0e 0e 0e 0e 0e
>1a31 0e 0e 0e 0d 0d 0e 0e 0e
>1a39 0e 0e 0e 0e 0e 01 0c 0d
>1a41 0e 0d 0d 0d 0e 0c 0d 0e
>1a49 0d 0d 0d 0e 01 0d 0d 0d
>1a51 0d 0d 0d 0d 0d 0d 0d 0d
>1a59 0d 0d 0d 0d 0d 0c 0d 0d
>1a61 0d 0d 0d 0d 0e 0c 0c 0d
>1a69 0e 0d 0d 0e 0d 0d 0d 0e
>1a71 0d 0d 0c 0e 0e 0e 0e 0e
>1a79 0e 0e 0e 0d 0d 0e 0e 0e
>1a81 0e 0e 0e 0e 0e 0d 0d 0c
>1a89 0e 0d 0d 0d 0e 0d 0d 0e
>1a91 0d 0d 0d 0e 0d 01 0c 01
>1a99 0d 0d 0d 0d 0c 0d 0d 0d
>1aa1 0d 0d 0c 0f 0f 0f 0d 0d
>1aa9 0d 06 09 0f 0f 0f 0f 0f
>1ab1 0f 0f 0f 0f 0f 0f 0f 0f
>1ab9 0f 0f 0f 0f 0f 0c 0e 0e
>1ac1 0e 0e 0e 0e 0e 0e 0e 0e
>1ac9 0e 0e 0e 0e 0c 0f 0f 0d
>1ad1 0e 0f 0f 0f 0f 0f 0f 0f
>1ad9 0f 0f 0f 0f 0f 0f 0d 0f
>1ae1 0f 0c 0e 0f 0e 0f 0f 0f
>1ae9 0f 0f 0f 0f 0f 0f 0f 0f
>1af1 0c 0f 0f 0d 0e 0f 0e 0f
>1af9 0f 0f 0f 0f 0f 0f 0f 0e
>1b01 0f 0e 0d 0f 0f 0f 0e 0f
>1b09 0e 0f 0f 0f 0f 0f 0f 0f
>1b11 0f 0e 0f 0e 0c 0f 0f 0d
>1b19 0e 0f 0e 02 02 0f 02 02
>1b21 0f 02 02 0e 0f 0e 0d 0f
>1b29 0f 0d 0e 0f 0e 01 01 0e
>1b31 01 01 0e 01 01 0e 0f 0e
>1b39 0d 0f 0f 0f 0f 0f 0e 0e
>1b41 0e 0e 0e 0e 0e 0e 0e 0e
>1b49 0f 0f 0f 0f 0f 0f 0e 0f
>1b51 0f 0f 0f 0f 0f 0f 0f 0f
>1b59 0f 0f 0f 0f 0f 0f 06 0c
>1b61 0c 0d 0d 0d 0d 0d 0d 0d
>1b69 0c 0c 0c 0e 0c 0d 0d 0c
>1b71 0d 0d 0c 0c 0d 0d 0c 0e
>1b79 0c 0d 0d 0c 0c 0e 0d 0c
>1b81 0d 0d 0d 0d 0c 0d 0d 0d
>1b89 0c 0e 0c 0c 0d 0d 0e 0e
>1b91 0c 0d 0d 0e 0d 0c 0c 0c
>1b99 0d 0d 0d 0e 0c 0d 0d 0d
>1ba1 0d 0e 01 0d 0d 0e 0d 0c
>1ba9 0d 0d 0d 0c 0d 0e 0c 0c
>1bb1 0c 0d 0d 0e 0d 0d 0d 0e
>1bb9 0d 01 0d 0c 0d 0d 0d 0e


```

>1bc1 0c 0d 0c 0d 0d 0e 0d 0c
>1bc9 0d 0e 0d 0c 0c 0d 0d 0d
>1bd1 0d 0e 0c 0c 0c 0d 0d 0e
>1bd9 0d 0c 0c 0e 0d 0c 0c 0d
>1be1 0d 0d 0d 0e 0d 01 0d 0d
>1be9 0d 0d 0d 0d 0d 0e 0d 0e
>1bf1 0e 0e 0c 0e 0e 0e 0e 0e
>1bf9 0e 0e 0e 0e 0e 0e 0e 0e
>1c01 0d 01 0d 0d 0d 0d 0d 0d
>1c09 0d 0d 0d 0d 0d 0d 0d 0d
>1c11 0d 0d 0d 04 0c 02 0e 0f
>1c19 0e 0d 0e 0f 0e 02 0c 0e
>1c21 0f 0e 0f 0e 02 0c 0d 01
>1c29 0e 0f 0e 0d 0d 0f 0e 01
>1c31 0d 0e 0f 0d 0f 0e 01 0d
>1c39 0f 0e 0e 0f 0e 0f 0e 0f
>1c41 0e 0e 0f 0d 0f 0e 0f 0e
>1c49 0e 0f 0f 0f 0d 0f 0e 0f
>1c51 0e 0f 0d 0d 0f 0e 0f 0e
>1c59 0f 0e 0e 0f 0f 0e 0e 0f
>1c61 0e 0f 0e 0f 0e 0e 0f 0e
>1c69 0f 0e 0f 0d 0d 0f 0f 0e
>1c71 0e 0f 0d 0f 0d 0f 0e 0e
>1c79 0f 0e 0f 0e 0f 0e 0e 0c
>1c81 0f 0e 0e 0f 0e 0f 0e 0f
>1c89 0e 0e 0f 0e 0f 0e 0f 0d
>1c91 0d 0d 0f 0e 0e 0f 0e 0f
>1c99 0e 0f 0e 0e 0f 0e 0f 0e
>1ca1 0f 0e 0e 0f 0f 0e 0e 0f
>1ca9 0e 0f 0e 0f 0e 0f 0e 0e
>1cb1 0f 0e 0f 0e 0e 0f 0f 0e
>1cb9 0e 02 0e 02 0e 02 0e 0e
>1cc1 0f 0e 02 0e 02 0e 0e 0f
>1cc9 03 09 0f 0f 0f 0f 0f 0f
>1cd1 0f 0f 0f 0f 0f 0f 0f 0f
>1cd9 0f 0f 0f 0f 0d 0d 0f 0d
>1ce1 0d 0d 0d 0f 0f 0d 0d 0d
>1ce9 0d 0f 0d 0d 0f 0f 0f 0f
>1cf1 0f 0d 0f 0f 0f 0f 0f 0f
>1cf9 0f 0f 0d 0f 0f 0f 0f 0f
>1d01 0d 0d 0f 0d 0f 0d 0d 0d
>1d09 0d 0d 0d 0f 0d 0f 0d 0d
>1d11 0f 0f 0f 0f 0f 0f 0f 0f
>1d19 0c 0c 0c 0c 0c 0f 0f 0f
>1d21 0f 0f 0f 0f 0f 0f 0f 0f
>1d29 0f 0d 0c 0c 0c 0c 0d 0f
>1d31 0f 0f 0f 0f 0f 0f 0d 0d
>1d39 0f 0d 0f 0d 0d 0d 0d 0d
>1d41 0d 0f 0d 0f 0d 0d 0f 0f
>1d49 0f 0f 0f 0d 0f 0f 0f 0f
>1d51 0f 0f 0f 0f 0d 0f 0f 0f
>1d59 0f 0f 0d 0d 0f 0d 0d 0d
>1d61 0d 0f 0f 0d 0d 0d 0d 0f
>1d69 0d 0d 0f 08 0f 0f 0f 0f
>1d71 0f 0f 0f 0f 0f 0f 0f 0f
>1d79 0f 0f 0f 0f 0f 0c 01 01
>1d81 01 01 01 01 01 01 01 01
>1d89 01 01 01 01 01 01 01 01
>1d91 0d 0c 0c 0c 0c 0d 0c 0d
>1d99 0d 0d 0d 0c 0d 0c 0c 0c
>1da1 0d 0d 0d 0c 0d 0d 0d 0d
>1da9 0c 0c 0d 0d 0d 0c 0d 0c
>1db1 0d 0c 0c 0d 0d 0c 0c 0c
>1db9 0d 0d 0c 0d 0c 0d 0d 0c
>1dc1 0d 0c 0d 0d 0c 0d 0d 0c
>1dc9 0d 0d 0d 0d 0c 0d 0d 0c
>1dd1 0d 0c 0d 0c 0d 0c 0c 0d
>1dd9 0d 0c 0c 0c 0c 0d 0c 0d
>1de1 0d 0d 0c 0c 0c 0c 0c 0c
>1de9 0d 0d 0d 0d 0d 0d 0d 0d
>1df1 0d 0d 0d 0d 0d 0d 0d 0d
>1df9 0d 0d 0d 0d 09 0f 0f 0f
>1e01 0f 0f 0f 0f 0f 0f 0f 0f
>1e09 0f 0f 0f 0f 0f 09 0f 0f
>1e11 0f 0f 0f 0f 10 10 10 10
>1e19 10 10 0f 0f 0f 0f 0f 0f
>1e21 02 0f 0f 0f 0f 0f 0f 02
>1e29 0f 0f 02 0f 0f 0f 0f 0f
>1e31 0f 0d 0f a9 00 8d 38 03
>1e39 0d 36 03 18 6d 3e 03 90
>1e41 03 ee 37 03 8d 36 03 ea
>1e49 ea ea ad 3e 03 0a 8d 0f
>1e51 ff a9 01 8d 10 ff ad 11
>1e59 ff a9 2f 8d 11 ff ac 3e
>1e61 03 c0 00 f0 1a 20 6b 28
>1e69 20 80 28 20 42 31 98 a0

```

```

>1e71 10 a2 00 ca d0 fd 88 d0
>1e79 f8 a8 ce 3e 03 d0 cb 78
>1e81 a9 b3 8d 1a 03 a9 2e 8d
>1e89 15 03 58 4c a9 22 ce 48
>1e91 03 d0 08 ad 11 ff 29 df
>1e99 8d 11 ff 60 ce 49 03 d0
>1ea1 fa ee 4a 03 ee 4a 03 ae
>1ea9 4a 03 bd 00 20 8d 0f ff
>1eb1 bd 00 21 8d 0e ff bd 01
>1eb9 20 8d 10 ff ad 12 ff 29
>1ec1 fc 1d 01 21 8d 12 ff a9
>1ec9 12 8d 49 03 a9 0a 8d 48
>1ed1 03 ad 11 ff 09 3f 8d 11
>1ed9 ff 60 20 8f 1e 20 9d 1e
>1ee1 4c 0e ce ad 47 03 c9 00
>1ee9 d0 01 60 ce 48 03 d0 fa
>1ef1 ee 4a 03 ee 4a 03 ae 4a
>1ef9 03 bd 00 21 8d 0e ff ad
>1f01 12 ff 29 fc 1d 01 21 8d
>1f09 12 ff a9 14 8d 48 03 ad
>1f11 11 ff 09 1f 8d 11 ff 60
>1f19 20 e4 1e ad 48 03 c9 01
>1f21 d0 08 ad 11 ff 29 ef 8d
>1f29 11 ff 4c 0e ce a9 ff 85
>1f31 d0 a9 3c 85 d1 ac 33 03
>1f39 c0 00 f0 11 18 a5 d0 69
>1f41 12 85 d0 a5 d1 69 00 85
>1f49 d1 88 4c 39 1f ad 43 05
>1f51 c9 01 d0 10 a5 c6 c9 17
>1f59 d0 03 4c 35 34 c9 0c d0
>1f61 03 4c f6 33 a5 c6 c9 0c
>1f69 d0 03 4c 00 33 c9 17 0d
>1f71 03 4c 48 33 c9 2c d0 03
>1f79 4c 90 33 c9 2d f0 01 60
>1f81 38 a5 d0 e9 12 85 d0 a5
>1f89 d1 e9 00 85 d1 4c c3 33
>1f91 20 72 34 20 2e 1f a5 c6
>1f99 c9 29 f0 01 60 a2 00 bd
>1fa1 d9 1f 9d 11 0c a9 77 9d
>1fa9 11 03 e8 e0 06 d0 f0 a9
>1fb1 02 a2 00 a0 00 88 d0 fd
>1fb9 ca d0 f8 38 e9 01 d0 f1
>1fc1 a5 c6 c9 40 f0 fa a2 00
>1fc9 a9 78 9d 11 0c a9 51 9d
>1fd1 11 08 e8 e0 06 d0 f1 60
>1fd9 18 09 1d 1b 0d 0c 0d 0d
>1fe1 0d 0d 0d 0d 0d 0d 0d 0d
>1fe9 0d 0d 0d 0d 0d 0d 0d 0d
>1ff1 0d 0d 36 01 ff 0d 0d 0d
>1fff 0d 0d 0d 0d 0d 0d 0d a9
>2001 00 a9 00 a9 00 a9 00 59
>2009 01 59 01 59 01 59 01 7f
>2011 01 7f 01 7f 01 7f 01 a9
>2019 00 a9 00 a9 00 a9 00 a9
>2021 00 a9 00 a9 00 a9 00 59
>2029 01 59 01 59 01 59 01 7f
>2031 01 7f 01 7f 01 7f 01 a9
>2039 00 a9 00 a9 00 a9 00 04
>2041 02 04 02 04 02 04 02 7f
>2049 01 7f 01 7f 01 7f 01 59
>2051 01 c5 01 c5 01 c5 01 c5
>2059 01 59 01 59 01 59 01 04
>2061 02 04 02 04 02 04 02 7f
>2069 01 7f 01 7f 01 7f 01 c5
>2071 01 c5 01 c5 01 c5 01 59
>2079 01 59 01 59 01 59 01 06
>2081 01 06 01 06 01 06 01 06
>2089 01 06 01 06 01 06 01 06
>2091 01 06 01 06 01 06 01 06
>2099 01 06 01 06 01 06 01 59
>20a1 01 59 01 59 01 59 01 7f
>20a9 01 7f 01 7f 01 7f 01 a9
>20b1 00 a9 00 a9 00 a9 00 54
>20b9 02 54 02 54 02 54 02 ad
>20c1 02 ad 02 ad 02 ad 02 c0
>20c9 02 c0 02 c0 02 c0 02 54
>20d1 02 54 02 54 02 54 02 02
>20d9 03 02 03 02 03 02 03 0c
>20e1 02 c0 02 c0 02 c0 02 e3
>20e9 02 e3 02 e3 02 e3 02 ad
>20f1 02 ad 02 ad 02 ad 02 54
>20f9 02 54 02 54 02 54 02 2a
>2101 03 2a 03 2a 03 2a 03 56
>2109 03 56 03 56 03 56 03 60
>2111 03 60 03 60 03 60 03 2a
>2119 03 2a 03 2a 03 2a 03 2a

```

```

>2121 03 2a 03 71 03 2a 03 56
>2129 03 56 03 56 03 56 03 60
>2131 03 60 03 60 03 60 03 2a
>2139 03 2a 03 2a 03 2a 03 81
>2141 03 81 03 71 03 81 03 60
>2149 03 60 03 60 03 60 03 71
>2151 03 71 03 60 03 71 03 56
>2159 03 56 03 56 03 56 03 81
>2161 03 81 03 71 03 81 03 60
>2169 03 60 03 60 03 60 03 71
>2171 03 71 03 60 03 71 03 56
>2179 03 56 03 56 03 56 03 42
>2181 03 71 03 71 03 71 03 42
>2189 03 71 03 71 03 71 03 42
>2191 03 71 03 71 03 71 03 42
>2199 03 71 03 71 03 71 03 56
>21a1 03 56 03 56 03 56 03 60
>21a9 03 60 03 60 03 60 03 2a
>21b1 03 2a 03 2a 03 2a 03 2a
>21b9 03 2a 03 2a 03 2a 03 56
>21c1 03 56 03 56 03 56 03 60
>21c9 03 60 03 60 03 60 03 2a
>21d1 03 2a 03 2a 03 2a 03 81
>21d9 03 81 03 71 03 81 03 60
>21e1 03 60 03 60 03 60 03 71
>21e9 03 71 03 60 03 71 03 56
>21f1 03 56 03 56 03 56 03 42
>21f9 03 71 03 71 03 71 03 a9
>2201 00 8d 33 03 8d 35 03 8d
>2209 39 03 8d 3a 03 8d 3b 03
>2211 85 de 85 df a9 39 8d de
>2219 0f 8d 0f 0f a9 63 8d 3e
>2221 03 20 00 30 20 5a 2c 20
>2229 e4 30 a9 96 8d 32 03 a9
>2231 00 85 d4 a9 3d 85 d4 a9
>2239 11 8d 05 3d 20 57 30 a9
>2241 03 8d 0f ff 8d 10 ff a9
>2249 00 8d 11 ff a9 02 8d 38
>2251 03 20 a8 26 20 96 26 20
>2259 b6 26 20 96 26 20 c4 26
>2261 20 96 26 a9 05 8d 05 3d
>2269 a9 30 8d fd 28 8d fe 28
>2271 a9 01 85 dd 4c f8 26 20
>2279 93 2f 20 00 30 20 ed 30
>2281 20 7e 31 20 88 2c 20 b3
>2289 2a a2 00 a9 30 9d f0 28
>2291 9d fa 28 e8 e0 05 d0 f5
>2299 a9 48 0d fc 28 a9 00 8d
>22a1 43 03 8d 36 03 8d 37 03
>22a9 ad 43 03 18 69 09 8d e5
>22b1 0f ee 43 03 ad 43 03 c9
>22b9 15 f0 03 4c 00 22 4c f7
>22c1 2d ad 35 03 c9 80 f0 03
>22c9 4c f8 26 78 a9 0e 8d 14
>22d1 03 a9 ce 8d 15 03 5d a9
>22d9 00 8d 1d ff 4c 34 1e a9
>22e1 00 8d 3f 03 8d 40 03 a9
>22e9 30 a2 00 9d f5 28 e8 e0
>22f1 05 d0 f8 4c f7 22 a9 fe
>22f9 8d 4a 03 4c 78 22 ff 03
>2301 0f 3e 3a f5 d5 c0 c0 c0
>2309 f0 bc ac 5f 57 03 03 ea
>2311 ea d5 d5 f0 3c 0f 03 ab
>2319 ab 57 57 0f 3c f0 c0 00
>2321 1e 1c 1c 16 06 03 07 00
>2329 00 00 1e 0e be f2 c0 03
>2331 4f 7d 70 78 00 00 00 e0
>2339 c0 60 69 38 38 78 00 0a
>2341 2a 2a 2f 2f 2a 2a 2a 0a
>2349 a8 a8 aa aa aa aa aa 3a
>2351 0e 3a 2a 0a 0c 0f 3c aa
>2359 a8 a8 a8 a8 c0 c0 00 2a
>2361 aa aa be be aa aa aa a8
>2369 aa be 82 be aa aa aa aa
>2371 ab ab aa 2a 0c 3c 00 aa
>2379 ea ea aa a8 30 30 3c 0a
>2381 2a 2a aa aa aa aa aa a0
>2389 a8 a8 f8 f8 a8 a8 a8 aa
>2391 aa 2a 00 03 03 00 c0 c0
>2399 b0 ac a8 a0 30 f0 3c ff
>23a1 ff c0 c0 ca ca c9 c9 ff
>23a9 ff 03 03 a3 a3 63 63 c9
>23b1 c9 ca c0 c0 c0 ff ff c9
>23b9 63 a3 a3 03 03 ff ff f7
>23c1 df df 7f ff ff fd f7 f7
>23c9 fd fd ff ff ff 7f df f7

```



```

>23d1 fd ff ff ff 7f df f7 df
>23d9 7f ff ff ff ff df f7 df 03
>23e1 0f 3d 35 f0 c0 ea ea c0
>23e9 f0 7c 5c 0f 03 ab ab d5
>23f1 d5 c0 c0 fa 3e 0f 03 57
>23f9 57 03 03 af bc f0 c0 01
>2401 03 07 01 01 21 63 ff 80
>2409 c0 e0 00 00 84 c6 ff ff
>2411 63 21 01 01 07 03 01 ff
>2419 c6 84 80 00 e0 c0 00 0a
>2421 2a 2f 20 2f 2a 2a 2a 0a
>2429 a8 a8 aa aa aa aa aa 3a
>2431 0e 3a 2a 0a 0c 3c 03 aa
>2439 aa a8 a8 a0 c0 c0 00 2a
>2441 aa be 82 be aa aa aa a8
>2449 aa be 82 be aa aa aa aa
>2451 ab ab aa 2a 0c 0c 3c aa
>2459 ea ea aa a8 30 3c 00 0a
>2461 2a 2a aa aa aa aa aa 0a
>2469 a8 f8 03 f8 a8 a8 a8 a0
>2471 aa 2a 2a 00 03 03 03 ac
>2479 b0 ac a0 a0 3c c0 ff
>2481 ff d5 d5 d0 d0 d2 d2 ff
>2489 ff 57 57 07 07 87 87 d2
>2491 d2 d0 d0 d5 d5 ff ff 87
>2499 87 07 07 57 57 ff ff 07
>24a1 1f 1f 7f ff ff fd f4 f4
>24a9 fd fd ff ff ff 7f 1f f4
>24b1 fd ff ff ff 7f 1f 07 1f
>24b9 7f ff ff ff fd f4 d0 03
>24c1 0f 3c 30 fa ea d5 d5 c0
>24c9 f0 3c 0c af ab 57 57 c0
>24d1 c0 ea ea f5 3d 0f 03 03
>24d9 03 ab ab 5f 7c f0 c0 00
>24e1 00 00 78 70 7d 4f 03 00
>24e9 78 39 38 68 60 c0 e0 07
>24f1 03 06 16 1c 1c 1e 00 c0
>24f9 f2 be 0e 1e 00 00 00 0a
>2501 2a 2f 20 2f 2a 2a 0a 0a
>2509 a8 a0 aa aa aa aa aa 3a
>2511 0e 3a 2a 0a 0c 0c 3f aa
>2519 aa a8 a8 a0 c0 c0 00 2a
>2521 aa be 82 be aa aa aa a8
>2529 aa be 82 be aa aa aa aa
>2531 ab ab aa 2a 0c 0c 3c aa
>2539 ea ea aa a8 30 30 3c 0a
>2541 2a 2a aa aa aa aa aa 0a
>2549 a8 f8 03 f8 a8 a8 a8 a0
>2551 aa 2a 2a 00 03 03 03 ac
>2559 b0 ac a8 a0 30 3c fc ff
>2561 ff ea ea e5 e5 e4 e4 ff
>2569 ff ab ab 5b 5b 1b 1b e4
>2571 e4 e5 e5 ea ea ff ff 1b
>2579 1b 5b 5b ab ab ff ff 07
>2581 1f 1f 7f ff ff fd f4 f4
>2589 fd fd ff ff ff 7f 1f f4
>2591 fd ff ff ff 7f 1f 07 1f
>2599 7f ff ff ff fd f4 d0 00
>25a1 00 00 00 00 00 01 01 00
>25a9 00 00 00 00 00 00 00 03
>25b1 01 00 00 00 00 00 00 c0
>25b9 00 00 00 00 00 00 00 00
>25c1 00 00 00 03 04 04 04 00
>25c9 00 00 00 c0 60 30 20 04
>25d1 02 04 07 00 00 00 00 20
>25d9 30 10 a0 80 00 00 00 10
>25e1 1f 10 10 10 20 40 40 70
>25e9 50 10 00 0c 07 02 04 20
>25f1 10 10 10 10 17 18 00 00
>25f9 10 10 10 10 f0 00 00 e6
>2601 dc a5 dc c9 03 d0 04 a9
>2609 00 05 dc c9 00 d0 03 4c
>2611 8b 26 c9 01 d0 03 4c 80
>2619 26 a9 00 20 78 26 ea a0
>2621 00 b1 d0 99 00 3a 99 20
>2629 3a c0 c0 20 d0 f3 b1 d0
>2631 99 20 3a 99 40 3a c0 c0
>2639 40 d0 f3 b1 d0 99 40 3a
>2641 c8 c0 60 d0 f6 b1 d0 99
>2649 40 3a c8 c0 80 d0 f6 b1
>2651 d0 99 40 3a c8 c0 a0 d0
>2659 f6 b1 d0 99 40 3a 99 60
>2661 3a 99 80 3a 99 a0 3a c0
>2669 c0 c0 d0 ea b1 d0 99 40
>2671 37 c0 c0 e0 d0 f6 60 85
>2679 d0 a9 23 85 d1 60 00 a9

```

```

>2681 e0 85 d0 a9 23 85 d1 4c
>2689 20 26 a9 c0 85 d0 a9 24
>2691 85 d1 4c 20 26 a9 03 a2
>2699 00 a0 00 88 d0 fd ca d0
>26a1 f8 38 e9 01 d0 f1 60 a2
>26a9 00 bd a0 25 9d 20 38 e8
>26b1 e0 20 d0 f5 60 a2 00 bd
>26b9 c0 25 9d 20 38 e8 e0 20
>26c1 d0 f5 60 a2 00 bd e0 25
>26c9 9d 20 38 e8 e0 20 d0 f5
>26d1 60 20 c4 26 20 96 26 20
>26d9 b6 26 20 96 26 20 a8 26
>26e1 20 96 26 a2 00 bd 00 3d
>26e9 c9 11 d0 05 a9 0f 9d 00
>26f1 3d e8 e0 b5 d0 ef 60 20
>26f9 91 1f 20 ef 34 20 a9 28
>2701 ad 35 03 c9 ff f0 2a 20
>2709 c0 31 20 42 37 20 a0 35
>2711 20 00 2d 20 7e 2d 20 73
>2719 2e 20 00 35 20 09 2f 20
>2721 3d 2f 20 51 2f 20 28 31
>2729 20 00 2c 20 db 2a 4c c2
>2731 22 c9 ff d0 00 ad 33 03
>2739 0a 0a 0a 0a 0a 95 d0 a9
>2741 3d 85 d1 a9 00 ac 33 03
>2749 c0 00 f0 07 18 69 0e 88
>2751 4c 49 27 85 d2 18 a5 d0
>2759 e5 d2 18 6d 32 03 85 d0
>2761 a0 00 20 a1 27 ad 33 03
>2769 c9 00 f0 11 a5 d0 38 e9
>2771 12 85 d0 20 a1 27 a5 d0
>2779 18 69 12 85 d0 ad 33 03
>2781 c9 09 f0 11 a5 d0 18 69
>2789 12 85 d0 20 a1 27 a5 d0
>2791 38 e9 12 85 d0 a9 00 85
>2799 d4 a9 3d 85 d5 4c c6 27
>27a1 a0 00 a9 11 91 d0 ad 32
>27a9 03 c9 01 f0 08 c6 d0 a9
>27b1 11 91 d0 e6 d0 ad 32 03
>27b9 c9 12 f0 08 e6 d0 a9 11
>27c1 91 d0 c6 d0 60 20 57 30
>27c9 a9 00 8d 0f ff 8d 10 ff
>27d1 8d 11 ff a9 02 8d 38 03
>27d9 20 d2 26 20 57 30 20 96
>27e1 26 ce d8 0f ad d8 0f c9
>27e9 30 f0 03 4c 00 22 4c ff
>27f1 22 ff ff ff 00 ff ff ff
>27f9 ff ff ff ff 00 ff ff c0
>2801 00 f0 0d c0 12 f0 09 c0
>2809 24 f0 05 88 20 26 28 c8
>2811 20 26 28 c0 11 f0 0d c0
>2819 23 f0 09 c0 35 f0 05 c8
>2821 20 26 28 88 60 b1 d0 c9
>2829 04 f0 0a c9 05 f0 06 c9
>2831 06 f0 02 d0 05 a9 ff 8d
>2839 35 03 a9 00 91 d0 a9 c0
>2841 8d 0f ff a9 00 8d 10 ff
>2849 a9 02 8d 38 03 8d ad 36
>2851 03 18 69 32 90 03 ee 37
>2859 03 8d 36 03 ea ea ea a0
>2861 32 20 6b 28 88 d0 fa 4c
>2869 80 28 a2 04 fe f0 28 bd
>2871 f0 28 c9 3a d0 08 a9 30
>2879 9d f0 28 ca 10 ee 60 a2
>2881 18 a0 04 18 20 f0 ff a2
>2889 00 bd f0 28 20 d2 ff e8
>2891 e0 05 d0 f5 60 ee ff 28
>2899 ad fb 28 c9 3a d0 08 a9
>28a1 30 8d fb 28 ee fa 28 60
>28a9 20 c0 2d 20 aa 31 20 00
>28b1 29 4c da 28 ad 3b 03 cd
>28b9 3c 03 d0 1c a5 dd c9 01
>28c1 00 16 a9 71 8d 19 ff a9
>28c9 cd 8d 0f ff a9 00 85 dd
>28d1 8d 10 ff a9 02 8d 38 03
>28d9 60 a9 00 8d 19 ff 4c 00
>28e1 26 4c aa 31 4c aa 31 30
>28e9 30 30 36 30 30 33 30 30
>28f1 30 38 38 38 30 30 38 38
>28f9 38 31 31 48 30 30 38 ad
>2901 00 ff c9 0a 90 06 30 e9
>2909 0a 4c 03 29 a8 a9 ee c0
>2911 00 f0 07 18 69 12 88 4c
>2919 10 29 85 d0 a9 3d 85 d1
>2921 e6 de a5 de c9 12 d0 04
>2929 a9 00 85 de 85 d2 18 69

```

```

>2931 12 85 d3 18 69 12 85 d4
>2939 a4 d3 b1 d0 c9 0d 10 05
>2941 c9 0f f0 01 60 c0 12 f0
>2949 0a 88 b1 d0 c9 10 d0 03
>2951 4c 7d 29 a4 d3 c0 23 f0
>2959 07 c8 b1 d0 c9 10 f0 f0
>2961 a5 d0 c9 ee f0 08 a4 d2
>2969 b1 d0 c9 10 f0 e2 a5 d0
>2971 c9 90 f0 d0 a4 d4 b1 d0
>2979 c9 10 d0 c8 a9 10 a4 d3
>2981 91 d0 e6 df a5 df c9 32
>2989 f0 01 60 a2 00 bd 00 3d
>2991 c9 10 d0 05 a9 0f 9d 00
>2999 3d e8 e0 b5 d0 eb a9 c0
>29a1 8d 0f ff a9 00 8d 10 ff
>29a9 a9 02 8d 3c 58 03 60 c9 10
>29b1 d0 08 4c 50 2d 4c 64 d2
>29b9 c9 00 f0 0d c9 01 f0 09
>29c1 c9 0b f0 05 c9 00 f0 01
>29c9 60 a4 d3 20 00 28 a5 d0
>29d1 c9 00 f0 05 a4 d2 20 00
>29d9 28 a5 d0 c9 a2 f0 e9 a5
>29e1 d3 18 69 12 a8 4c 00 28
>29e9 ff ff ff ff ff ff ff a9
>29f1 2a 8d 39 0e 8d 4e 0e a9
>29f9 71 8d 39 0a 8d 4e 0a a2
>2a01 00 bd 78 2b 9d 3a 0e a9
>2a09 71 9d 3a 0a e8 e0 14 d0
>2a11 f0 a2 00 bd 8c 2b 9d 8a
>2a19 0e a9 71 9d 3a 0a e8 e0
>2a21 14 d0 f0 20 a8 2a a5 c6
>2a29 c9 0d f0 09 c9 24 f0 3e
>2a31 c9 3c d0 ef 60 ad 46 03
>2a39 49 ff 8d 46 03 c9 ff f0
>2a41 09 c9 00 f0 17 a9 ff 8d
>2a49 46 03 a9 2a 8d 39 0e a9
>2a51 71 8d 39 0a a9 23 8d 89
>2a59 0e 4c 24 2a a9 23 8d 39
>2a61 0e a9 2a 8d 89 0e a9 71
>2a69 8d 89 0a 4c 24 2a ad 47
>2a71 03 49 ff 8d 47 03 c9 ff
>2a79 f0 09 c9 00 f0 17 a9 ff
>2a81 8d 47 03 a9 2a 8d 4e 0e
>2a89 a9 71 8d 4e 0a a9 23 8d
>2a91 9e 0e 4c 24 2a a9 23 8d
>2a99 4e 0e a9 2a 8d 9e 0e a9
>2aa1 71 8d 9e 0a 4c 24 2a a2
>2aa9 80 a0 00 88 d0 fd ca d0
>2ab1 f8 60 a9 02 a2 00 a0 00
>2ab9 88 d0 fd ca d0 fa e9 01
>2ac1 d0 f6 78 a9 b3 8d 14 03
>2ac9 a9 2e 8d 15 03 58 a9 00
>2ad1 8d 36 03 8d 37 03 8d 43
>2ad9 03 60 ad 46 03 c9 ff f0
>2ae1 0a a2 40 a0 00 88 d0 fd
>2ae9 ca d0 f8 60 a9 00 b1 d4
>2af1 99 00 3d c8 c0 b4 d0 f6
>2af9 60 20 5a 2c 20 e0 30 4c
>2b01 00 27 ad 35 03 c9 ff d0
>2b09 0a ce d0 0f a9 30 cd d0
>2b11 0f d0 c6 60 8c 2b 9d 8a
>2b19 0e a9 71 9d 8a 0a e8 e0
>2b21 14 d0 f0 ad 44 03 c9 00
>2b29 f0 09 c9 ff f0 0c a9 00
>2b31 8d 44 03 a9 3a 85 d0 4c
>2b39 3f 2b a9 8a 85 d0 a9 0a
>2b41 85 d1 a0 00 a9 32 91 d0
>2b49 c8 ce 44 03 d0 23 a9 04
>2b51 8d 44 03 ad 45 03 38 e9
>2b59 10 b0 02 a9 71 8d 45 03
>2b61 a2 00 0d f9 08 38 e9 10
>2b69 b0 02 a9 71 e8 e0 17 d0
>2b71 f1 4c db 1e 00 d0 20 0e
>2b79 09 1b 1c 23 23 23 23 23
>2b81 23 23 23 23 23 23 23 23
>2b89 23 17 16 1b 14 17 1f 23
>2b91 23 23 23 23 23 23 23 23
>2b99 23 23 23 17 0e 0e 0d
>2ba1 19 1d 0d 1b 3c 1b 17 0e
>2ba9 1c 23 18 1a 0d 1b 0d 16
>2bb1 1c 1b 1c 1b 15 0b 23 1b
>2bb9 1c 17 16 0d 21 23 11 16
>2bc1 23 1c 10 0d 23 0b 09 1e
>2bc9 0d 1b 23 0a 21 0b 09 1a
>2bd1 1b 1c 0d 16 23 1a 0d 11
>2bd9 16 13 0d 1b 18 0d 0d 0c

```



```

>2be1 3a 23 23 23 23 23 23
>2be9 23 23 23 15 1d 1b 11 0b
>2bf1 3a 1b 18 09 0b 0d 23 1c
>2bf9 17 23 18 14 09 21 0f ce
>2c01 41 03 f0 0d ad 42 03 c9
>2c09 01 d0 03 4c 55 31 4c 94
>2c11 31 a9 10 8d 41 03 ee 42
>2c19 03 ad 42 03 29 01 8d 42
>2c21 03 c9 01 d0 10 a2 00 bd
>2c29 46 2c 9d cb 0f e8 e0 0a
>2c31 d0 f5 4c 55 31 a2 00 bd
>2c39 50 2c 9d cb 0f e8 e0 0a
>2c41 d0 f5 4c 94 31 10 11 3a
>2c49 30 30 30 30 30 23 23 0c
>2c51 11 3a 30 30 30 30 23
>2c59 23 a9 5b 85 d0 a9 0f 85
>2c61 d1 ae 43 03 e0 00 f0 0f
>2c69 a5 d0 18 69 b5 90 02 e6
>2c71 d1 85 d0 ca 4c 65 2c a5
>2c79 d0 85 d4 a5 d1 85 d5 a0
>2c81 b4 b1 d0 8d 3c 03 60 a2
>2c89 00 bd a0 2b 9d 82 0c a9
>2c91 63 9d 82 08 e8 e0 13 d0
>2c99 f0 a2 00 bd b5 2b 9f f9
>2ca1 0c a3 71 9d f9 03 e8 e0
>2ca9 17 d0 f0 a2 00 bd cc 2b
>2cb1 9d 53 0d a9 56 9d 53 09
>2cb9 e8 e0 02 d0 f0 a2 00 bd
>2cc1 ce 2b 9d 9d 0d a9 52 9d
>2cc9 9d 09 e8 e0 0e d0 f0 a2
>2cd1 00 bd cd 2b 9d e9 0d a9
>2cd9 61 9d e9 09 e8 e0 16 d0
>2ce1 f0 a2 00 bd f2 2b 9d 2e
>2ce9 0f a9 75 9d 2e 0b e8 e0
>2cf1 8d d0 f0 a9 ff 8d 46 03
>2cf9 8d 47 03 4c f0 29 00 a2
>2d01 00 bd a2 3d c9 0b f0 06
>2d09 c9 00 f0 0a d0 0d a9 0c
>2d11 9d a2 3d 4c 1c 2d a9 01
>2d19 9d a2 3d e8 e0 12 d0 e1
>2d21 a9 90 85 d0 a9 3d 85 d1
>2d29 a9 00 85 d2 a9 12 85 d3
>2d31 a4 d3 b1 d0 c9 0d f0 17
>2d39 c9 0e f0 13 c9 01 f0 0f
>2d41 c9 0c f0 0b c9 02 f0 07
>2d49 c9 03 f0 03 4c af 29 a4
>2d51 d2 b1 d0 c9 00 f0 c9
>2d59 0b f0 03 4c 64 2d 18 69
>2d61 01 91 d0 e6 d2 e6 d3 a5
>2d69 d2 c9 12 d0 c3 a5 d0 38
>2d71 e9 12 b0 02 c6 d1 85 d0
>2d79 c9 ee d0 ac 60 a9 90 85
>2d81 d0 a9 3d 85 d1 a9 00 85
>2d89 d2 a9 12 85 d3 a4 d3 b1
>2d91 d0 c9 07 90 14 c9 0b b0
>2d99 10 a9 d2 b1 d0 20 b9 29
>2da1 ea ea ea ea ea ea ea ea
>2da9 ea e6 d2 e6 d3 a5 d2 c9
>2db1 12 d0 da a5 d0 38 e9 12
>2db9 85 d0 c9 ee d0 c7 60 a9
>2dc1 ed 85 d0 a9 3c 85 d1 ae
>2dc9 33 03 e0 00 f0 0f a5 d0
>2dd1 18 69 12 90 02 e6 d1 85
>2dd9 d0 ca 4c cb 2d ad 32 03
>2de1 85 d2 18 69 12 85 d3 18
>2de9 69 12 85 d4 a5 d0 c9 ed
>2df1 f0 25 a4 d2 b1 d0 c9 00
>2df9 f0 1a c9 02 f0 16 c9 09
>2e01 f0 12 c9 0b f0 0e c9 07
>2e09 f0 0a c9 08 f0 06 c9 0a
>2e11 f0 02 d0 03 4c 6d 2e a4
>2e19 d3 c8 13 f0 13 8d b1 d0
>2e21 c9 07 f0 f0 c9 08 f0 ec
>2e29 c9 09 f0 e8 c9 0a f0 e4
>2e31 a4 d3 c8 24 f0 13 c8 b1
>2e39 d0 c9 07 f0 d7 c9 03 f0
>2e41 d3 c9 09 f0 cf c9 0a f0
>2e49 cb a5 d0 c9 a2 f0 b0 a4
>2e51 d4 b1 d0 c9 03 f0 bd c9
>2e59 07 f0 b9 c9 08 f0 b5 c9
>2e61 09 f0 b1 c9 0a f0 ad c9
>2e69 02 f0 a9 60 a9 ff 8d 35
>2e71 03 60 ad 34 03 c9 00 f0
>2e79 05 c9 01 f0 01 60 a9 ff
>2e81 8d 34 03 20 4f 28 ee 3b
>2e89 03 ee fe 28 ad fe 28 c9

```

```

>2e91 3a d0 08 a9 30 8d fe 28
>2e99 ee fd 28 ea ea 20 b5
>2ea1 28 ea a9 60 8d 0f ff a9
>2ea9 02 8d 10 ff a9 01 8d 38
>2eb1 03 60 ad 38 03 c9 01 d0
>2eb9 20 ce 0f ff ce 0f ff ce
>2ec1 0f ff ad 11 ff 09 2f 8d
>2ec9 11 ff ad 0f ff c9 10 b0
>2ed1 05 a9 00 8d 38 03 4c 19
>2ed9 1f c9 02 d0 20 ee 0f ff
>2ee1 ee 0f ff ee 0f ff ad 11
>2ee9 ff 09 4f 8d 11 ff ad 0f
>2ef1 ff c9 d6 d0 05 a9 00 8d
>2ef9 38 03 4c 19 1f ad 11 ff
>2f01 29 9f 8d 11 ff 4c 19 1f
>2f09 ad 39 03 cd 32 03 d0 09
>2f11 ad 3a 03 cd 33 03 d0 01
>2f19 60 ad 38 03 cf a9 01 f0 f8
>2f21 a9 d0 8d 0f ff a9 01 8d
>2f29 10 ff a9 02 8d 38 03 ad
>2f31 32 03 8d 39 03 ad 33 03
>2f39 8d 3a 03 60 ea a5 c6 c9
>2f41 34 d0 05 a9 ff 8d 35 03
>2f49 60 20 f0 ff ee 3b 03 60
>2f51 ad 3b 03 cd 3c 03 90 f7
>2f59 a9 24 8d 46 0f a9 25 8d
>2f61 47 0f a9 26 8d 6e 0f a9
>2f69 27 8d 6f 0f a5 a5 29 f7
>2f71 09 51 8d 46 0b 8d 47 0b
>2f79 8d 6e 0b 8d 6f 0b ad 32
>2f81 03 c9 12 d0 0c ad 33 03
>2f89 c9 09 d0 05 a9 80 8d 35
>2f91 03 60 ad 07 ff 09 10 8d
>2f99 07 ff a9 02 8d 12 ff a9
>2fa1 38 8d 13 ff a9 00 8d 15
>2fa9 ff 8d 19 ff a9 52 8d 16
>2fb1 ff a9 35 8d 17 ff a9 77
>2fb9 8d 3b 05 a2 00 a9 23 9d
>2fc1 00 0c 9d 00 d0 9d 00 0e
>2fc9 ca d0 f4 a2 00 9d 00 8d
>2fd1 e8 e0 98 d0 f8 a9 00 8f
>2fd9 38 03 78 a9 4a 8d 14 03
>2fe1 a9 2b 8d 15 03 58 60 ea
>2fe9 ea ea ea ea ea ea ea ea
>2ff1 ea ea ea ea ea ea ea ea
>2ff9 ea ea ea ea ea ea ea a9
>3001 00 85 d0 85 d2 85 d4 85
>3009 d5 a9 0c 85 d1 a9 00 85
>3011 d3 a5 d4 c9 d2 90 11 c9
>3019 26 b0 0d a5 d5 c9 02 90
>3021 07 c9 16 b0 03 4c 33 30
>3029 a4 d4 a9 78 91 d0 a9 51
>3031 91 d2 e6 d4 a5 d4 c9 28
>3039 d0 d7 a9 00 85 d4 e6 d5
>3041 a5 d0 18 69 28 90 04 e6
>3049 d1 e6 d3 85 d0 85 d2 a5
>3051 d5 c9 18 d0 bc 60 a9 52
>3059 85 d0 85 d2 a9 00 85 d6
>3061 85 d7 a9 0c 85 d1 a9 08
>3069 85 d3 a4 d6 b1 d4 aa bd
>3071 d1 30 85 d9 b1 d4 0a 0a
>3079 69 40 85 d8 a4 d7 a5 d8
>3081 91 d0 a5 d9 91 d2 e6 d8
>3089 c8 a5 d8 91 d0 a5 d9 91
>3091 d2 e6 d8 90 69 27 a8 a5
>3099 d8 91 d0 a5 d9 91 d2 e6
>30a1 d8 c8 a5 d8 91 d0 a5 d9
>30a9 91 d2 e6 d6 e6 d7 e6 d7
>30b1 a5 d7 c9 24 d0 13 a5 d0
>30b9 18 69 50 90 04 e6 d1 e6
>30c1 d3 85 d0 85 d2 a9 00 85
>30c9 d7 a5 d6 c9 b4 d0 9b 60
>30d1 7f 7f 61 61 79 79 79 6f
>30d9 6f 6f 6f 47 47 01 51 00
>30e1 4d 71 2a 4c ed 2a 3c 88
>30e9 d0 f8 60 60 a2 00 bd 00
>30f1 31 9d c0 0f a9 77 9d c0
>30f9 0b e8 e0 28 d0 90 60 23
>3101 1b 0b 3a 30 30 30 30
>3109 23 23 10 11 3a 30 30 30
>3111 30 30 23 23 14 11 3a 33
>3119 23 23 1c 11 3a 39 39 23
>3121 23 0b 09 3a 09 23 23 ad
>3129 3e 03 c9 00 d0 06 a9 ff
>3131 8d 35 03 60 ce 3d 03 d0
>3139 fa a9 09 8d 3d 03 ce 3e

```

```

>3141 03 ce df 0f ad df 0f c9
>3149 2f d0 e8 a9 39 8d 0f 0f
>3151 ce de 0f 60 ad 37 03 cd
>3159 40 03 d0 08 ad 36 03 cd
>3161 3f 03 f0 19 90 17 a2 05
>3169 bd ef 28 9d f4 28 ca d0
>3171 f7 ad 37 03 8d 40 03 ad
>3179 36 03 8d 3f 03 a2 18 a0
>3181 0e 18 20 f0 ff a2 00 bd
>3189 f5 28 20 d2 ff e8 e0 05
>3191 d0 f5 60 a2 18 a0 0e 18
>3199 20 f0 ff a2 00 bd fa 28
>31a1 20 d2 ff e8 e0 05 d0 f5
>31a9 60 a9 30 8d fa 28 8d fb
>31b1 28 ac 3c 03 20 96 28 88
>31b9 d0 fa 60 ea ea ea ea a9
>31c1 8f 85 d0 a9 3d 85 d1 a9
>31c9 01 85 d2 a9 13 85 d3 a4
>31d1 d3 b1 d0 c9 0f d0 19 a4
>31d9 d2 b1 d0 c9 0b f0 07 c9
>31e1 0c f0 03 4c f4 31 a9 0f
>31e9 91 d0 a4 d3 a9 0b 91 d0
>31f1 4c d3 32 c9 00 f0 07 c9
>31f9 01 f0 03 4c a0 32 a9 0f
>3201 91 d0 a4 d3 a9 00 91 d0
>3209 4c d3 32 c9 0f d0 f9 a4
>3211 d2 c0 01 f0 5d 88 b1 d0
>3219 c9 0b f0 07 c9 0f 03
>3221 4c 46 32 a4 d3 88 b1 d0
>3229 c9 07 90 0b c9 0c f0 07
>3231 c9 0e f0 03 4c 73 32 a4
>3239 d2 a9 0b 91 d0 88 a9 0f
>3241 91 d0 4c d3 32 c9 00 f0
>3249 07 c9 01 f0 03 4c 73 32
>3251 a4 d3 88 b1 d0 c9 07 90
>3259 0b c9 0c f0 07 c9 0e f0
>3261 03 4c 73 32 a4 d2 a9 00
>3269 91 d0 88 a9 0f 91 d0 4c
>3271 d3 32 a4 d2 c0 12 f0 5a
>3279 c8 b1 d0 c9 0b f0 07 c9
>3281 0c f0 03 4c a9 32 a4 d3
>3289 c8 b1 d0 c9 07 90 0b c9
>3291 0c f0 07 c9 0e f0 03 4c
>3299 d3 32 a4 d2 a9 0b 91 d0
>32a1 c8 a9 0f 91 d0 4c d3 32
>32a9 c9 00 f0 07 c9 01 f0 03
>32b1 4c d3 32 a4 d3 c8 b1 d0
>32b9 c9 07 90 0b c9 0c f0 07
>32c1 c9 0e f0 03 4c d3 32 a4
>32c9 d2 a9 00 91 d0 c8 a9 0f
>32d1 91 d0 e6 d2 e6 d3 a5 d2
>32d9 c9 13 f0 03 4c d0 31 a9
>32e1 01 85 d2 a9 13 85 d3 a5
>32e9 d0 38 e9 12 b0 02 c6 d1
>32f1 85 d0 c9 ed f0 03 4c d0
>32f9 31 60 ff ff ff ff 00 ad
>3301 32 03 c9 01 d0 01 60 ac
>3309 32 03 88 b1 d0 c9 0f f0
>3311 1f c9 0d f0 1b c9 01 f0
>3319 17 c9 00 f0 13 c9 0c d0
>3321 e5 c0 01 f0 e1 88 b1 d0
>3329 c9 0f d0 da a9 0c 91 d0
>3331 ac 32 03 ac a9 0f 91 d0 ce
>3339 32 03 ac 32 03 b1 d0 8d
>3341 34 03 a9 04 91 d0 60 ad
>3349 32 03 c9 12 d0 01 60 ac
>3351 32 03 c8 b1 d0 c9 0f f0
>3359 1f c9 0d f0 1b c9 01 f0
>3361 17 c9 00 f0 13 c9 0c d0
>3369 e5 c0 12 f0 e1 c8 b1 d0
>3371 c9 0f d0 da a9 0c 91 d0
>3379 ac 32 03 a9 0f 91 d0 ee
>3381 32 03 ac 32 03 b1 d0 8d
>3389 34 03 a9 06 91 d0 60 ad
>3391 33 03 c9 09 d0 01 60 ad
>3399 32 03 69 12 a3 b1 d0 c9
>33a1 0f 0f 0d c9 0d f0 09 c9
>33a9 01 f0 05 c9 00 f0 01 60
>33b1 8d 34 03 a9 05 91 d0 ac
>33b9 32 03 a9 91 d0 ee 33
>33c1 03 60 ad 33 03 c9 00 d0
>33c9 01 60 ac 32 03 b1 d0 c9
>33d1 0f 0f 0d c9 0d f0 09 c9
>33d9 01 f0 05 c9 00 f0 01 60
>33e1 8d 34 03 a9 05 91 d0 ad
>33e9 32 03 69 11 a8 a9 0f 91

```


programme

```
>33f1 d0 ce 33 03 60 ad 32 03
>33f9 c9 01 d0 01 60 ac 32 03
>3401 83 b1 d0 c9 00 f0 25 c9
>3409 01 f0 21 c9 0d f0 1d c9
>3411 0b f0 04 c9 0c d0 e5 c0
>3419 01 f0 e1 83 b1 d0 c9 0f
>3421 d0 da a9 0c 91 d0 a9 0f
>3429 c8 91 d0 60 8d 34 03 a9
>3431 0f 91 d0 60 ad 32 03 c9
>3439 12 d0 01 60 20 f6 34 ea
>3441 c9 00 f0 25 c9 01 f0 21
>3449 c9 0d f0 1e c9 0b f0 04
>3451 c9 0c d0 e7 c0 12 f0 e3
>3459 c8 b1 d0 c9 0f d0 dc a9
>3461 0c 91 d0 a9 0f 83 91 d0
>3469 60 8d 34 03 a9 0f 91 d0
>3471 60 a9 ff 85 d0 a9 3c 85
>3479 d1 ac 33 03 c0 00 f0 0f
>3481 a5 d0 18 69 12 90 02 e6
>3489 d1 85 d0 88 4c 7d 34 a2
>3491 00 bd fb bf aa 78 8e 08
>3499 ff ad 08 ff 8e 08 ff cd
>34a1 08 ff d0 f2 58 49 ff a3
>34a9 29 0f aa bd f0 bf c0 0f
>34b1 90 02 09 60 c9 01 d0 0e
>34b9 a5 d0 38 e9 12 b0 02 c6
>34c1 d1 05 d0 4c c3 33 c9 03
>34c9 d0 03 4c 48 c3 c9 05 d0
>34d1 03 4c 90 33 c9 07 d0 03
>34d9 4c 00 33 c9 83 d0 03 4c
>34e1 35 34 c9 87 d0 03 4c f6
>34e9 33 60 a9 00 85 d4 a9 3d
>34f1 85 d5 4c 57 30 ac 32 03
>34f9 c8 b1 d0 60 00 02 02 a9
>3501 a1 85 d0 a9 3d 85 d1 a9
>3509 01 85 d2 a9 13 85 d3 a4
>3511 d2 b1 d0 c9 02 d0 1b a4
>3519 d3 b1 d0 c9 0f f0 09 a4
>3521 d2 a9 03 91 d0 4c 33 35
>3529 a9 02 91 d0 a4 d2 a9 0f
>3531 91 d0 e6 d2 e6 d3 a5 d2
>3539 c9 13 d0 d3 a9 01 85 d2
>3541 a9 13 85 d3 a5 d0 38 e9
>3549 12 b0 02 c6 d1 85 d0 a5
>3551 d0 c9 ed d0 ba a4 d3 b1
>3559 d0 c9 03 d0 1b a4 d2 b1
>3561 d0 c9 0f f0 09 a4 d3 a9
>3569 02 91 d0 4c 79 35 a9 03
>3571 91 d0 a4 d3 a9 0f 91 d0
>3579 e6 d2 e6 d3 a5 d2 c9 13
>3581 d0 d3 a9 01 85 d2 a9 13
>3589 85 d3 a5 d0 18 69 12 90
>3591 02 e6 d1 85 d0 a5 d0 c9
>3599 a1 d0 ba 60 00 00 00 a9
>35a1 00 85 d0 a9 3d 85 d1 a9
>35a9 11 85 d2 a4 d2 b1 d0 c9
>35b1 08 d0 14 c0 11 f0 10 c8
>35b9 b1 d0 c9 0f d0 09 a9 08
>35c1 91 d0 88 a9 0f 91 d0 c6
>35c9 d2 a5 d2 c9 ff d0 dc a5
>35d1 d0 18 69 12 85 d0 c9 b4
>35d9 d0 cd 4c e0 35 02 02 a9
>35e1 00 85 d0 a9 3d 85 d1 a9
>35e9 00 85 d2 a4 d2 b1 d0 c9
>35f1 0a d0 14 c0 00 f0 10 88
>35f9 b1 d0 c9 0f d0 09 a9 0a
>3601 91 d0 c8 a9 0f 91 d0 e6
>3609 d2 a5 d2 c9 12 d0 dc a5
>3611 d0 18 69 12 85 d0 c9 b4
>3619 d0 cd 4c 20 36 00 00 a9
>3621 00 85 d0 a9 3d 85 d1 a9
>3629 00 85 d2 a9 12 85 d3 a4
>3631 d3 b1 d0 c9 07 d0 12 a4
>3639 d2 b1 d0 c9 0f d0 0a a9
>3641 07 91 d0 a4 d3 a9 0f 91
>3649 d0 e6 d2 e6 d3 a5 d2 c9
>3651 12 d0 dc a5 d0 18 69 12
>3659 85 d0 c9 a2 d0 c9 ea a9
>3661 90 85 d0 a9 3d 85 d1 a9
>3669 00 85 d2 a9 12 85 d3 a4
>3671 d2 b1 d0 c9 09 d0 12 a4
>3679 d3 b1 d0 c9 0f d0 0a a9
>3681 09 91 d0 a4 d2 a9 0f 91
>3689 d0 e6 d2 e6 d3 a5 d2 c9
>3691 12 d0 dc a5 d0 38 e9 12
>3699 85 d0 c9 ee d0 c9 60 a5
>36a1 d0 c9 ee f0 10 a4 d2 b1
```

```
>36a9 d0 c9 0f d0 08 a4 d3 a9
>36b1 07 91 d0 38 60 18 60 a4
>36b9 d3 c0 23 f0 f8 c8 b1 d0
>36c1 c9 0f d0 f1 83 a9 08 91
>36c9 d0 38 60 35 d0 c9 90 f0
>36d1 e4 a4 d4 b1 d0 c9 0f d0
>36d9 dc a4 d3 a9 09 91 d0 38
>36e1 60 a4 d3 c0 12 f0 ce 88
>36e9 b1 d0 c9 0f d0 c7 c8 a9
>36f1 0a 91 d0 38 60 20 08 36
>36f9 b0 0d 20 a0 36 b0 08 20
>3701 e2 36 b0 03 20 cc 36 60
>3709 20 cc 36 b0 0d 20 b8 36
>3711 b0 08 20 a0 36 b0 03 20
>3719 e2 36 60 20 e2 36 b0 0d
>3721 20 cc 36 b0 08 20 b8 36
>3729 b0 03 20 a0 36 60 20 a0
>3731 36 b0 0d 20 e2 36 b0 08
>3739 20 cc 36 b0 03 20 b8 36
>3741 60 a9 ee 85 d0 a9 3c 85
>3749 d1 a9 00 85 d2 a9 12 85
>3751 d3 a9 24 85 d4 a4 d3 b1
>3759 d0 c9 07 d0 06 20 f6 36
>3761 4c 7f 37 c9 08 06 20
>3769 09 37 4c 7f 37 c9 09 d0
>3771 06 20 1c 37 4c 7f 37 c9
>3779 0a d0 03 20 2f 37 e6 d2
>3781 e6 d3 e6 d4 a5 d2 c9 12
>3789 d0 cb a5 d0 18 69 12 90
>3791 02 e6 d1 85 d0 c9 a2 d0
>3799 b0 60 a2 00 bd a2 3d c9
>37a1 0b f0 06 c9 00 f0 0a d0
>37a9 0d a9 0c 9d a2 3d 4c b7
>37b1 37 a9 01 9d a2 3d e8 e0
>37b9 12 d0 e1 a9 90 85 d0 a9
>37c1 3d 85 d1 a9 00 85 d2 a9
>37c9 12 85 d3 a4 d3 b1 d0 c9
>37d1 0d f0 17 c9 0e f0 13 c9
>37d9 01 f0 0f c9 0c f0 0b c9
>37e1 02 f0 07 c9 03 f0 03 4c
>37e9 fe 37 a4 d2 b1 d0 c9 00
>37f1 f0 07 c9 0b f0 03 4c fe
>37f9 37 49 01 91 d0 ff 00 07
>3801 1f 1f 7f ff ff fd f4 f4
>3809 fd fd ff ff ff 7f 1f f4
>3811 fd ff ff ff 7f 1f 07 1f
>3819 7f ff ff ff fd f4 d0 00
>3821 00 00 00 00 00 01 01 00
>3829 00 00 00 00 00 00 00 03
>3831 01 00 00 00 00 00 c0
>3839 00 00 00 00 00 00 3c
>3841 66 7e 00 7e 66 c0 00 38
>3849 7c c6 c6 fe c6 c6 00 fc
>3851 c6 c6 fc c6 c6 fc 00 fc
>3859 c6 c6 c0 c0 c6 fc 00 fc
>3861 c6 c6 c6 c6 c6 fc 00 fe
>3869 c0 c0 f8 c0 c0 fe 00 fe
>3871 c0 c0 f8 c0 c0 c0 fc
>3879 c6 c0 de c6 c6 fc 00 c6
>3881 c6 c6 fe c6 c6 c0 3c
>3889 18 18 18 18 18 3c 00 1e
>3891 0c 0c 0c 6c 6c 38 00 c6
>3899 cc d8 f0 d8 c0 c6 00 c0
>38a1 c0 c0 c0 c0 c0 fe 00 c6
>38a9 c6 ee fe c6 c6 c6 00 c6
>38b1 c6 e6 fe de ce c6 00 7c
>38b9 c6 c6 c6 c6 c6 7c 00 fc
>38c1 c6 c6 c6 fc c0 c0 00 7c
>38c9 c6 c6 c6 c6 ce 7e 00 fc
>38d1 c6 c6 c6 fc d8 ce 00 7c
>38d9 c6 c0 7c 06 c6 7c 00 fc
>38e1 30 30 30 30 30 30 c6
>38e9 c6 c6 c6 c6 c6 7c 00 c6
>38f1 c6 c6 c6 c6 c6 38 00 c6
>38f9 c6 c6 c6 c6 fe 6c 00 c6
>3901 c6 6c 38 6c c6 c6 00 cc
>3909 cc cc 78 30 30 30 fe
>3911 06 c0 18 30 60 fe 00 00
>3919 00 00 00 00 00 00 ff
>3921 f0 e7 cf df dd dc dc ff
>3929 0f e7 f3 fb bb 3b 3b de
>3931 df df df df df df 7b
>3939 fb fb fb fb fb fb 0c
>3941 18 30 30 30 18 00 30
>3949 18 0c 0c 0c 18 30 00
>3951 00 c6 cc d8 f0 60 00 00
>3959 18 18 7e 18 18 00 00 00
```

```
>3961 00 00 00 00 18 18 30 00
>3969 00 00 7e 00 00 00 00 00
>3971 00 00 00 00 18 18 00 00
>3979 03 06 0c 18 30 60 00 fe
>3981 c6 c6 c6 c6 c6 fe 00 06
>3989 06 06 06 06 06 06 00 fe
>3991 06 06 fe c0 c0 fe 00 fe
>3999 06 06 fe 06 06 fe 00 c6
>39a1 c6 c6 fe 06 06 06 00 fe
>39a9 c0 c0 fe 06 06 fe 00 fe
>39b1 c0 c0 fe c6 c6 fe 00 fe
>39b9 c6 c6 06 06 06 06 00 fe
>39c1 c6 c6 fe c6 c6 fe 00 fe
>39c9 c6 c6 fe 06 06 fe 00 00
>39d1 00 18 00 00 18 00 00 00
>39d9 00 18 00 00 18 18 30 00
>39e1 00 00 7e 7e 00 00 00 00
>39e9 00 7e 00 7e 00 00 00 70
>39f1 18 0c 06 c0 18 70 00 3c
>39f9 66 06 c0 18 00 18 00 03
>3a01 0f 3d 35 f0 c0 ea ea c0
>3a09 f0 7c 5c 0f 03 ab ab d5
>3a11 d5 c0 c0 fa 3e 0f 03 57
>3a19 57 03 03 af bc f0 c0 03
>3a21 0f 3d 35 f0 c0 ea ea c0
>3a29 f0 7c 5c 0f 03 ab ab d5
>3a31 d5 c0 c0 fa 3e 0f 03 57
>3a39 57 03 03 af bc f0 c0 01
>3a41 03 07 01 01 21 63 ff 80
>3a49 c0 e0 80 80 84 c6 ff ff
>3a51 63 21 01 01 07 03 01 ff
>3a59 c6 84 80 80 e0 c0 80 01
>3a61 03 07 01 01 21 63 ff 80
>3a69 c0 e0 80 80 84 c6 ff ff
>3a71 63 21 01 01 07 03 01 ff
>3a79 c6 84 80 80 e0 c0 80 0a
>3a81 2a 2f 20 2f 2a 2a 2a a0
>3a89 a8 aa aa aa aa aa aa 3a
>3a91 0e 3a 2a 0a 0c 03 aa
>3a99 aa a8 a8 a0 c0 c0 c0 2a
>3aa1 aa be 82 be aa aa aa a8
>3aa9 aa be 82 be aa aa aa aa
>3ab1 ab ab aa 2a 0c 0c 3c aa
>3ab9 ea ea aa 80 30 3c 00 0a
>3ac1 2a 2a aa aa aa aa aa a0
>3ac9 a8 f8 08 f8 a8 a8 a8 aa
>3ad1 aa 2a 2a 00 03 03 03 ac
>3ad9 b0 ac a8 a0 30 3c 00 ff
>3ae1 ff d5 d5 d0 d0 d2 d2 ff
>3ae9 ff 57 57 07 07 87 87 d2
>3af1 d2 d0 d0 d5 d5 ff ff 87
>3af9 87 07 07 57 57 ff ff ff
>3b01 ff d5 d5 d0 d0 d2 d2 ff
>3b09 ff 57 57 07 07 87 87 d2
>3b11 d2 d0 d0 d5 d5 ff ff 87
>3b19 87 07 07 57 57 ff ff ff
>3b21 ff d5 d5 d0 d0 d2 d2 ff
>3b29 ff 57 57 07 07 87 87 d2
>3b31 d2 d0 d0 d5 d5 ff ff 87
>3b39 87 07 07 57 57 ff ff ff
>3b41 ff d5 d5 d0 d0 d2 d2 ff
>3b49 ff 57 57 07 07 87 87 d2
>3b51 d2 d0 d0 d5 d5 ff ff 87
>3b59 87 07 07 57 57 ff ff 05
>3b61 1b 7f f3 df ff cf 4f f0
>3b69 60 fe ff a5 79 7a de e3
>3b71 ff ff f3 ff ff 3c 0f ff
>3b79 3f cf ff 3f fe fe f8 05
>3b81 1b 7f f3 df ff cf 4f f0
>3b89 60 fe ff a5 79 7a de e3
>3b91 ff ff f3 ff ff 3c 0f ff
>3b99 3f cf ff 3f fe fe f8 ff
>3ba1 ff f5 fc 3c 7a ff fe 7b
>3ba9 ff fa e9 ff ff ff e9 7b
>3bb1 ff fa e9 ff ff ff e9 ff
>3bb9 ff f5 fc 3c 7a ff fe cf
>3bc1 cf cf 00 f3 f3 f3 00 cf
>3bc9 cf cf 00 f3 f3 f3 00 cf
>3bd1 cf cf 00 f3 f3 f3 00 cf
>3bd9 cf cf 00 f3 f3 f3 00 00
>3be1 00 00 ff ff 00 00 00 00
>3be9 00 00 ff ff 00 00 00 00
>3bf1 00 00 ff ff 00 00 00 00
>3bf9 31 39 38 36 04 ff 00 a2
>3c01 00 a9 fe 9d 80 3c 9d 80
>3c09 3d 9d 80 3e ca d0 f2 8d
>3c11 07 10 a2 30 9d 31 03 ca
```


C U B E



In diesem Spiel müssen Sie alle grünen Würfel rot einfärben. Das ist nur zu schaffen, indem Sie die 4 umhergeisternden Gespenster nicht berühren und auf alle Würfel mit Ihrem einzigen Männchen springen. Sind Sie mit einem Gespenst zusammengeraten, so ist das Spiel beendet. Das Umfärben der Würfel ist von Stufe zu Stufe

anders: Bei Stufen mit einer ungeraden Zahl bleibt der besprungene Würfel rot. Bei Stufen mit geraden Zahlen färbt sich der Würfel bei jedem Draufspringen wieder um (also mal rot und mal grün). Sind alle Würfel rot eingefärbt, so geht es weiter zur nächsten Stufe. Zwischen jeder Stufe können Sie Ihren Punktestand aufbessern, indem Sie innerhalb von 10 Sekunden die rot einfärbbaren Würfel herausfinden. Jeder

gefundene Würfel gibt Punkte. Haben Sie keinen gefunden, geht es trotzdem zur nächsten Stufe. Wenn Sie alle 20 Stufen geschafft haben, so haben Sie das Spiel gewonnen.

Dieses Programm hat eine sehr gute Multicolor-Grafik, zahlreiche verschiedene Spielfelder und eine schnelle Spielgeschwindigkeit.

Eingabehinweise:

Das Programm besteht aus 3 Tei-

len, die nacheinander abgetippt und abgespeichert werden müssen. Anschließend müssen Sie Teil 1 laden und starten. Die anderen beiden Teile lädt das Programm selbst nach (Funktioniert nur mit Datasette)!!!!!!

Aber jetzt wünsche ich Ihnen viel Erfolg beim Abtippen und viel Spaß beim Spielen.

Wichtig: Datasette erst dann stoppen, wenn der letzte Teil eingeladen wurde!!!

C U B E

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```

10 SCNLDR:FOKE55,255:POKE56,55:CLR          <147>
20 REM*****                               <213>
30 REM***   C U B E   (1)   ***             <229>
40 REM***   BY   GUENTER DREES   ***       <190>
50 REM***   CANISIUSSTR.18   ***         <80>
60 REM***   ***                               <57>
70 REM***   4440 RHEINE   ***             <40>
80 REM*****                               <17>
90 PRINT" <DOHNT RIGHTS>DATAS<SPACE>WERDEN<SPACE>
EINGELESEN"                                <204>
100 DATA3C,66,6E,6E,60,62,3C,00, 636     <241>
110 DATA1B,3C,66,7E,66,66,66,00, 618     <104>
120 DATA7C,66,66,7C,66,66,7C,00, 780     <100>
130 DATA3C,66,60,60,60,66,3C,00, 612     <248>
140 DATA7B,6C,66,66,66,6C,7B,00, 762     <3>
150 DATA7E,60,60,7B,60,60,7E,00, 756     <101>
160 DATA7E,60,60,7B,60,60,60,00, 726     <81>
170 DATA3C,66,60,6E,66,66,3C,00, 632     <172>
180 DATA66,66,66,7E,66,66,66,00, 738     <195>
190 DATA3C,1B,1B,1B,1B,1B,3C,00, 240     <34>
200 DATA1E,0C,0C,0C,0C,6C,3B,00, 242     <66>
210 DATA66,6C,7B,70,7B,6C,66,00, 772     <39>
220 DATA60,60,60,60,60,60,60,00, 702     <8>
230 DATA63,77,7F,6B,63,63,63,00, 749     <245>
240 DATA66,76,7E,7E,6E,66,66,00, 786     <195>
250 DATA15,15,15,15,15,16,02,02, 131     <4>
260 DATA95,95,95,95,95,96,AA,AA, 1235    <115>
270 DATA3C,66,66,66,66,66,3C,0E,00, 542     <57>
280 DATA95,95,95,95,95,9A,9A, 1228    <232>
290 DATA54,54,54,54,06,06,02,02, 352     <237>
300 DATA56,56,56,56,56,56,AA,AA, 1016    <7>
310 DATA66,66,66,66,66,66,3C,00, 672     <6>
320 DATA56,56,56,55,AA,AA,AA,AA, 1007    <39>
330 DATA15,15,15,15,12,12,02,02, 124     <25>
340 DATA95,95,95,95,9A,9A,AA,AA, 1244    <60>
350 DATA66,66,66,3C,1B,1B,1B,00, 438     <73>
360 DATA95,95,95,95,99,99,AA,AA, 1236    <158>
370 DATA54,54,54,54,56,16,02,02, 448     <59>
380 DATA56,56,56,56,56,56,AA,AA, 920     <243>
390 DATA3C,0C,0C,0C,0C,0C,3C,00, 180     <154>
400 DATA56,56,56,55,55,9A,AA,AA, 911     <19>
410 DATA00,10,30,7F,7F,30,10,00, 382     <1>
420 DATA00,00,00,00,00,00,00,00, 0       <248>
430 DATA03,0F,3C,F0,F0,3C,0F,03, 636     <252>
440 DATACO,F0,3C,0F,0F,3C,F3,C3, 1020    <68>
450 DATA03,0F,3C,F0,F0,3C,CF,C3, 1020    <110>
460 DATACO,F0,3C,0F,0F,3C,F0,C0, 1014    <41>
470 DATAC3,CF,3C,F0,F0,3C,0F,03, 1020    <214>
480 DATAE3,F3,3C,0F,0F,3C,F0,C0, 1020    <0>
490 DATA06,0C,1B,00,00,00,00,00, 42       <192>
500 DATA0C,1B,30,30,30,1B,0C,00, 216     <144>
510 DATA30,1B,0C,0C,0C,1B,30,00, 180     <88>
520 DATA00,66,3C,FF,3C,66,00,00, 579     <0>
530 DATA00,1B,1B,7E,1B,1B,00,00, 222     <78>
540 DATA00,00,00,00,00,1B,1B,30, 96       <175>
550 DATA00,00,00,7E,00,00,00,00, 126     <12>
560 DATA00,00,00,00,00,1B,1B,00, 48       <50>
570 DATA00,03,06,0C,1B,30,60,00, 189     <42>
580 DATA3C,66,6E,76,66,66,3C,00, 654     <202>
590 DATA1B,1B,3B,1B,1B,1B,7E,00, 302     <255>
600 DATA3C,66,06,0C,30,60,7E,00, 450     <92>
610 DATA3C,66,06,1C,06,66,3C,00, 364     <4>
620 DATA06,0E,1E,66,7F,06,06,00, 291     <201>
630 DATA7E,60,7C,06,06,66,3C,00, 520     <96>
640 DATA3C,66,60,7C,66,66,3C,00, 646     <2>
650 DATA7E,66,0C,1B,1B,1B,1B,00, 336     <254>
660 DATA3C,66,66,3C,66,66,3C,00, 588     <170>
670 DATA3C,66,66,3E,06,66,3C,00, 494     <9>
680 DATA00,00,1B,00,00,1B,00,00, 48       <62>
690 DATA00,00,1B,00,00,1B,1B,00, 120     <153>
700 DATA00,00,00,00,02,02,02,02, 8       <125>
710 DATAAA,AA,AA,AA,AA,AA,AA,AA, 1360    <190>
720 DATA0A,0A,0A,0A,2A,2A,2A,2A, 208     <73>
730 DATAAA,AA,AA,A9,A9,A9,A9,AS, 1351    <163>
740 DATAAS,AS,AS,9A,9A,9A,9A,50, 1167    <163>
750 DATA50,50,50,40,40,40,40,00, 496     <217>
760 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <145>
770 DATA00,00,00,00,00,00,00,00, 0       <88>
780 DATA00,00,00,00,00,00,00,00, 0       <98>
790 DATA00,00,00,00,00,00,00,00, 0       <108>
800 DATA00,00,00,00,00,00,00,00, 0       <118>
810 DATA00,00,00,00,00,00,00,00, 0       <128>
820 DATA00,00,00,00,00,00,00,00, 0       <138>
830 DATA00,00,00,00,00,00,00,00, 0       <148>
840 DATA00,00,00,00,00,00,00,00, 0       <158>
850 DATA00,00,00,00,00,00,00,00, 0       <168>
860 DATA00,00,00,00,00,00,00,00, 0       <178>
870 DATA00,00,00,00,00,00,00,00, 0       <188>
880 DATA00,00,00,00,00,00,00,00, 0       <198>
890 DATA00,00,00,00,00,00,00,00, 0       <208>
900 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <30>
910 DATA01,05,05,15,19,19,19,1D, 136     <236>
920 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <50>
930 DATA40,50,50,54,64,64,64,724     <95>
940 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <70>
950 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <80>
960 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <90>
970 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <100>
980 DATAFF,00,FF,FF,FF,FF,FF,FF, 1785    <32>
990 DATA01,05,05,15,1D,19,19,19, 136     <217>
1000 DATAFF,FF,FF,FF,FF,FF,FF,FF, 2040    <130>
1010 DATA40,50,50,54,74,64,64,724     <166>
1020 DATA00,00,01,05,15,15,55,55, 218     <146>
1030 DATA00,00,40,50,54,54,55,55, 482     <83>
1040 DATA55,55,95,95,AD,AE,AE,AE, 1163    <0>
1050 DATA55,55,56,56,7A,BA,BA,BA, 1022    <178>
1060 DATAAE,EE,BE,AA,AA,AA,AA,AA, 1452    <187>
1070 DATABA,BB,BE,AA,AA,AA,AA,AA, 1413    <71>
1080 DATA00,00,01,05,15,11,5D,51, 218     <98>
1090 DATA00,00,40,50,54,44,75,45, 482     <143>
1100 DATA55,5D,97,95,AD,AE,AE,AE, 1173    <196>
1110 DATA55,75,D6,56,7A,BA,BA,BA, 1182    <51>
1120 DATAAE,AE,BE,EA,AA,AA,AA,AA, 1452    <15>
1130 DATABA,BA,BE,AB,AA,AA,AA,AA, 1413    <137>
1140 DATA00,00,01,05,15,15,15,90     <218>
1150 DATA00,00,40,50,54,50,5C,50, 480     <249>
1160 DATA95,95,95,AA,AB,AB,AB,AB, 1276    <215>
1170 DATAD6,7E,56,5A,6A,6A,6A,6A, 940     <181>

```



```

430 M*(3)="(D.BLUE C1 S1 DOWN LEFT2 CN CQ DOWN L
EFT2 CD C2)":G(3)=DEC("34C") <146>
440 M*(4)="(D.BLUE CS CP DOWN LEFT2 CA CE DOWN L
EFT2 CR CH)":G(4)=DEC("351") <165>
450 POKE65298,192:POKE65299,56 <7>
460 L$="{BRN SPACE2 DOWN LEFT2}="{DOWN LEFT2}="" <25>
470 E$(1)="!"+CHR$(34) <168>
480 E$(2)="*"$ <69>
490 E$(3)="%"$ <99>
500 E$(4)="!&":IFP<>OTHENDIMP(4,8) <29>
510 LS$="{BRN SPACE2}" <95>
520 LE=1 <186>
530 M1=5:M2=21:M=2:FORL=832TO870:POKEL,32:NEXT <122>
540 PRINT "{BRN}":SCNCLR <212>
550 T=0:ONINT((LE-1)/2)+160SUB1010,1020,1030,104
0,1050,1060,1070,1080,1090,1100 <139>
560 IFLE>20THEN2480 <55>
570 FORL=1TO4:FORI=1TO8 <97>
580 READP(L,I):NEXTI,L <118>
590 REM **** BILD **** <6>
600 FORL=1TO4:FORI=1TO8 <127>
610 IFP(L,I)=1THENGOSUB1130 <41>
620 NEXT:NEXT:READPS <154>
630 FORL=1TO4:REM GEISTERPOSITIONEN <228>
640 READA:POKE6(L),A:NEXT <240>
650 GOSUB1110:IFAX>0AND127THENGOSUB1140:ELSEGOSU
B2030 <122>
660 IFT=PTHEN1170 <86>
670 X=USR(.) <253>
680 CHAR1,M1,M2,M$(M):IFPEEK(3072+40*M2+M1-1)>80
THENB90 <185>
690 A=JOY(1):ONAAND127GOTO710,720,730,740,750,76
0,770,780 <154>
700 GOTO650 <18>
710 M=1:GOTO790 <250>
720 M=1:GOTO790 <4>
730 M=3:GOTO790 <20>
740 M=2:GOTO790 <27>
750 M=2:GOTO790 <37>
760 M=2:GOTO790 <47>
770 M=4:GOTO790 <64>
780 M=1:GOTO790 <65>
790 ONMGOTO800,820,850,870 <85>
800 IFPEEK(3072+40*(M2-4)+M1)=61THEN920:ELSEIFPE
EK(3072+40*(M2-4)+M1)=32THEN650 <119>
810 GOTO890 <145>
820 IFM2>6>23THEN650 <70>
830 IFPEEK(3072+40*(M2+6)+M1)=61THEN920:ELSEIFPE
EK(3072+40*(M2+6)+M1)=32THEN650 <213>
840 GOTO890 <175>
850 IFPEEK(3072+40*M2+M1+44)=61THEN920:ELSEIFPEE
K(3072+40*M2+M1+44)=32THEN650 <124>
860 GOTO890 <195>
870 IFPEEK(3072+40*M2+M1+36)=61THEN920:ELSEIFPEE
K(3072+40*M2+M1+36)=32THEN650 <213>
880 GOTO890 <215>
890 FORL=8TO0STEP-.7:VOLL:COLOR1,2,5 <214>
900 CHAR1,M1,M2-1,E$(INT(RND(O)*4+1)):SOUND1,880
+10*L,5 <187>
910 SOUND1,345,5:NEXT:SCNCLR:GOTO2080 <45>
920 CHAR1,M1,M2,L$ <77>
930 ONMGOTO940,950,960,970 <69>
940 M2=M2-5:GOTO980 <156>
950 M2=M2+5:GOTO980 <160>
960 M1=M1+4:GOTO980 <156>
970 M1=M1-4:GOTO980 <172>
980 SOUND1,345,2:SOUND1,880,5:GOTO650 <15>
990 T=T-1:CHAR,M1-1,M2+3,"{L.GRN SB3}" <167>
1000 RETURN <121>
1010 RESTORE1780:RETURN <150>
1020 RESTORE1830:RETURN <143>
1030 RESTORE1880:RETURN <174>
1040 RESTORE1930:RETURN <167>
1050 RESTORE1980:RETURN <197>
1060 RESTORE2230:RETURN <168>
1070 RESTORE2280:RETURN <198>
1080 RESTORE2330:RETURN <191>
1090 RESTORE2380:RETURN <221>
1100 RESTORE2430:RETURN <214>
1110 IFPEEK(837)=81THENPOKE837,89:POKE843,81:ELS
EPOKE837,81:POKE843,89 <113>
1120 RETURN <242>
1130 CHAR1,I*4,2+L*5,"{L.GRN}" +W$:RETURN <17>
1140 IFPEEK(2048+40*(M2+3)+M1)=50THENIFINT(LE/2)
=LE/2THEN990 <67>
1150 IFPEEK(2048+40*(M2+3)+M1)=95THENCHAR1,M1-1,
M2+3,"{RED SB3}":T=T+1:SC=SC+10:SOUND3,1020,5 <14>
1160 RETURN <26>
1170 RESTORE1180:FORL=1TO11:READA,B:SOUND1,1022,
2:SOUND1,A,B:NEXT <186>
1180 DATAB10,10,810,5,810,5,810,5,834,5,810,5,834,5 <215>
1190 DATAB54,5,881,10,881,5,881,5,881,15 <100>
1200 PRINT "{BRN}":SCNCLR <106>
1210 M1=5:M2=21:M=2 <193>
1220 FORI=1TO8:FORL=1TO4:GOSUB1130:P(L,I)=0:NEXT
:NEXT:PS=0:T=0 <0>
1230 FORL=1TO8:FORI=1TO4 <219>
1240 D=INT(RND(1)*4+1) <60>
1250 IFD=4THENP(I,L)=1:PS=PS+1 <114>
1260 NEXT:NEXT:TI$="000000" <156>
1270 CHAR1,M1,M2,M$(M):GOSUB2030 <46>
1280 IFVAL(TI$)>10THEN1570 <99>
1290 PRINT "{HOME DOWN4 RIGHT17}":10-VAL(TI$)"{LE
FT SPACE}" <38>
1300 A=JOY(1):ONAGOTO1320,1330,1340,1350,1360,13
70,1380,1390 <71>
1310 GOTO1270 <112>
1320 M=1:GOTO1400 <200>
1330 M=1:GOTO1400 <210>
1340 M=3:GOTO1400 <226>
1350 M=2:GOTO1400 <233>
1360 M=2:GOTO1400 <243>
1370 M=2:GOTO1400 <253>
1380 M=4:GOTO1400 <13>
1390 M=1:GOTO1400 <14>
1400 ONMGOTO1410,1430,1450,1470 <247>
1410 IFPEEK(3072+40*(M2-4)+M1)=61THEN1490 <104>
1420 GOTO1270 <222>
1430 IFPEEK(3072+40*(M2+6)+M1)=61THEN1490 <141>
1440 GOTO1270 <242>
1450 IFPEEK(3072+40*M2+44+M1)=61THEN1490 <236>
1460 GOTO1270 <6>
1470 IFPEEK(3072+40*M2+36+M1)=61THEN1490 <17>
1480 GOTO1270 <26>
1490 CHAR1,M1,M2,L$:ONMGOTO1500,1510,1520,1530 <185>
1500 M2=M2-5:GOTO1540 <253>
1510 M2=M2+5:GOTO1540 <1>
1520 M1=M1+4:GOTO1540 <253>
1530 M1=M1-4:GOTO1540 <13>
1540 SOUND1,345,2:SOUND1,880,5:J=INT(M2/5):K=INT
(M1/4) <59>
1550 IFP(J,K)=1THENT=T+1:SOUND3,1020,5:P(J,K)=0:
CHAR1,M1-1,M2+3,"{RED SB3}":SC=SC+10 <226>
1560 IFT<PTHEN1270 <233>
1570 REM <183>
1580 SCNCLR <26>
1590 PRINT "{YELLOW SB3 SPACE3 SB3 SPACE SB9 SPAC
E SB3 SPACE3 SB3 SPACE SB2}" <105>
1600 PRINT "{SB3 SPACE2 SB4 SPACE SB3 SPACE8 SB3
SPACE SB3 SPACES SB3}" <139>
1610 PRINT "{SB3 SPACE SB5 SPACE SB6 SPACE6 SB5 S
PACE6 SB3}" <51>
1620 PRINT "{SB9 SPACE SB6 SPACES5 SB3 SPACE SB3 S
PACE5 SB3}" <31>
1630 PRINT "{SB5 SPACE SB3 SPACE SB3 SPACE7 SB3 S
PACE3 SB3 SPACE4 SB3}" <211>
1640 PRINT "{SB4 SPACE2 SB3 SPACE SB9 SPACE SB3 S
PACE3 SB3 SPACE4 SB3}" <173>
1650 PRINT:PRINT <53>
1660 PRINT "{BRN SPACE3}="{SPACE4}=""=""="{SPACE}="
="{SPACE2}="{SPACE}=""=""="{SPACE}="" <214>
1670 PRINT "{SPACE3}="{SPACE4}=""=""="{SPACE4}="{SPAC
E2}="{SPACE}=""=""="{SPACE4}=""="" <241>
1680 PRINT "{SPACE3}="{SPACE4}=""=""="{SPACE4}="{SPAC
E2}="{SPACE}=""=""="{SPACE4}=""="" <251>
1690 PRINT "{SPACE3}="{SPACE4}=""=""="{SPACE2}="{SP
ACE2}="{SPACE}=""=""="{SPACE2}=""="" <171>
1700 PRINT "{SPACE3}="{SPACE4}=""=""="{SPACE2}="{SP
ACE2}="{SPACE}=""=""="{SPACE2}=""="" <181>
1710 PRINT "{SPACE3}="{SPACE4}=""=""="{SPACE4}="{SPAC
E2}="{SPACE}=""=""="{SPACE4}=""="" <25>
1720 PRINT "{SPACE3}=""=""="{SPACE}=""=""="{SPACES}="{SPA
CE2}="{SPACE2}=""=""="{SPACE4}=""=""="" <0>
1730 PRINT "{SPACE3}=""=""="{SPACE}=""=""="{SPACE2}=""
=""="{SPACE2}=""=""=""="{SPACE}=""=""="" <115>
1740 FORL=1TO10:POKE65286,11:SOUND1,880,5:POKE65
286,27:SOUND1,345,5:NEXT <240>
1750 LE=LE+1 <164>
1760 SCNCLR:X=USR(.) :X=USR(.) :X=USR(.) <195>
1770 GOTO530 <60>
1780 DATA1,1,0,1,0,0,0,1 <239>
1790 DATA1,0,1,1,1,1,1,1 <31>

```



```

1800 DATA1,1,1,0,1,0,1,1
1810 DATA1,0,1,0,1,1,1,1
1820 DATA23,9,3,10,15
1830 DATA1,1,1,0,1,1,1,0
1840 DATA1,0,1,1,1,0,1,1
1850 DATA1,1,1,0,1,1,1,0
1860 DATA1,0,1,1,1,0,1,1
1870 DATA24,11,19,9,2
1880 DATA1,0,1,0,1,1,1,0
1890 DATA1,1,1,0,0,0,1,1
1900 DATA0,1,0,1,0,1,1,1
1910 DATA1,1,1,1,1,1,0,0
1920 DATA21,36,4,23,0
1930 DATA1,0,1,1,1,0,1,0
1940 DATA1,0,1,0,1,0,1,0
1950 DATA1,0,1,0,1,1,1,0
1960 DATA1,1,1,1,0,0,1,1
1970 DATA20,25,9,19,25
1980 DATA1,0,1,0,1,0,1,0
1990 DATA1,0,1,0,1,0,1,0
2000 DATA1,0,1,0,1,0,1,0
2010 DATA1,1,1,1,1,1,1,1
2020 DATA20,37,19,35,39
2030 PRINT"HOME RIGHT14 YELLOW RIGHT"SC
2040 PRINT"HOME BLUE"HI:"WHITE";:IFHI<SCTHENHI
=SC
2050 PRINTHI
2060 PRINT"HOME RIGHT29 BLUE"LE:"WHITE"LE
2070 RETURN
2080 REM * SPIELLENDE *
2090 FORL=0T040:PRINT:NEXT
2100 PRINT"SPACE9"=====SPACE=====SPACE3=====SPACE
=====
2110 PRINT"SPACE9"=====SPACES=====SPACE3=====SPACE
=====SPACE2=====
2120 PRINT"SPACE9"=====SPACES=====SPACE=====SPAC
E=====SPACE3=====
2130 PRINT"SPACE9"=====SPACE3=====SPACE=====SP
ACE=====SPACE3=====
2140 PRINT"SPACE9"=====SPACES=====SPACE2=====SPAC
E=====SPACE3=====
2150 PRINT"SPACE9"=====SPACES=====SPACE3=====SPACE
=====SPACE2=====
2160 PRINT"SPACE9"=====SPACE=====SPACE3=====SP
ACE=====
2170 PRINT"DOWN3"=====
=====;
2180 PRINT"=====
=====;
<26> 2190 PRINT"=====
<44> =====;
<183> 2200 IFHI<SCTHENHI=SC
<52> 2210 PRINT"UP2";TAB(20-(LEN(STR$(SC))/2)), "EY
<70> LLOW LEFT"SC:PRINT:PRINT:PRINT:PRINT"RIGHT17"HI
<72> I:"HI
<90> 2220 FORL=0T02500:NEXT:SC=0:GOTO130
<37> 2230 DATA0,1,0,1,0,1,0,1
<98> 2240 DATA0,1,0,1,0,1,1,1
<106> 2250 DATA0,1,1,1,1,1,0,1
<128> 2260 DATA1,1,0,0,0,0,0,1
<126> 2270 DATA18,16,4,13,13
<249> 2280 DATA0,0,0,0,0,1,1,1
<144> 2290 DATA1,1,1,0,1,0,0,1
<146> 2300 DATA1,0,0,0,1,0,0,1
<168> 2310 DATA1,1,1,1,1,1,1,1
<184> 2320 DATA19,37,17,17,4
<125> 2330 DATA0,0,0,0,1,1,1,1
<186> 2340 DATA1,1,1,1,0,0,0,1
<196> 2350 DATA1,0,0,0,0,1,0,1
<206> 2360 DATA1,1,1,1,1,1,1,1
<0> 2370 DATA20,37,30,30,15
<218> 2380 DATA1,1,1,0,1,1,1,1
<46> 2390 DATA0,1,1,0,1,0,0,1
<64> 2400 DATA0,1,1,0,0,0,0,1
<14> 2410 DATA1,1,1,1,1,1,1,1
<27> 2420 DATA22,37,37,2,2
<172> 2430 DATA1,1,1,1,1,1,1,1
<224> 2440 DATA0,1,0,0,0,0,0,1
<232> 2450 DATA0,1,0,0,0,0,0,1
2460 DATA1,1,1,1,1,1,1,1
2470 DATA20,37,4,4,37
2480 FORL=0T030:PRINT:PRINT:POKE65298,196:POKE652
97,208
2490 COLOR0,1:COLOR4,1:COLOR1,4,4
2500 CHAR1,7,9,"CSPACE0SPACENSPACEGSPACE
RSPACEASPACEUSPACEUSPACEASPACE
TSPACEISPACE0SPACENSPACE"
2510 CHAR1,14,15,"YELLOW"DU"SPACE"HAST"SPACE"AL
LE"
2520 CHAR1,14,17,"SPACE2"LEVEL"S"
2530 CHAR1,14,19,"SPACE"GESCHAFFT"
2540 PRINT"HOME DOWN5 RIGHT13"SCORE"SPACE":RVS
ON FLASHON";SC
2550 FORL=1T05
2560 FORLL=500T0700STEP20
2570 SOUND2,LL+49*L,2:SOUND1,LL+50*L,2:NEXT:NEXT
2580 POKE239,0:GETKEY#:GOTO130
ENDE DES LISTINGS

```



Hostage

Bei Hostage geht es darum, Geiseln aus einem Hochhaus zu retten, die von Gangstern dorthin verschleppt wurden. Der Spieler steuert einen Hubschrauber über einem Hochhaus hin und her und läßt über einem Fenster, aus dem eine Geisel herauschaut, eine Leiter herunter. Die Leiter wird heruntergelassen und wieder heraufgezogen. Wenn die Geisel hochgezogen wurde, erhält der Spieler Punkte. War es jedoch eine Bombe, explodiert der Hubschrauber. Drei Hubschrauber stehen zur Verfügung. Nach jeweils 26 Gei-

seln, die gerettet wurden, geht es zum nächsten Level über. Dann wechseln sich Geiseln, Bomben und Fenster schneller ab.

Gesteuert wird mit den Cursorstasten für links und rechts und mit der Space-Taste um die Leiter runterzulassen.

Das Spiel wird ganz normal mit LOAD "HOSTAGE", 8 oder DLOAD "HOSTAGE" geladen und mit RUN gestartet.

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```

0 POKE56,47:CLR:GOTO500
10 GETT$:IFT$=""THEN50
20 IFT$="RIGHT"THENIFF<29THENCHAR1,P,6,"SPACE
23":P=P+1:CHAR1,P,6,"D.BLUE S* SA":GOTO50
30 IFT$="LEFT"THENIFF>9THENCHAR1,P,6,"SPACE2"
:P=P-1:CHAR1,P,6,"D.BLUE S* SA":GOTO50
40 IFT$="SPACE"THENGOTO300
50 GOSUB100:GOTO10
100 Z=Z-1:IFZ<LVTHENRETURN
110 Z=6:R=FNW(5):S=FNW(11):0=FNW(7)
120 CHAR1,9+S*2,10+R*2,6*(0):SOUND1,400+0*20,1
199 RETURN
300 I=0:PS=3312+P
310 I=I+1:T=PEEK(PS+I*40):M(I)=T:SOUND2,800-I*30
,3:GOSUB100
320 IFT=32THENPOKEPS+I*40,68:GOTO310
330 IFT=69THENPOKEPS+I*40,196:GOTO310
340 IFT=160THENGOTO400
345 IFT=70THENPOKEPS+I*40,68:GOTO310
350 POKEPS+I*40,32:I=I-1

```



```

360 I=I-1:POKEPS+I*40,T:POKEPS+40*I+40,M(I+1)
365 SOUND2,800-I*30,3:FORJ=1TO10:NEXT
370 IFI>1THEN360
380 POKEPS+40,32
385 IFT=66THEN700:ELSEGDT0800
399 GDT010
400 I=I-1:POKEPS+I*40,M(I):SOUND2,800-I*30,3
410 IFI>1THEN400
499 GDT010
500 GDSUB1000:REM NEUE ZEICHEN
510 GDSUB2000:REM INITIALISIERUNG
520 GDSUB3000:REM TITEL/ANLEITUNG
530 GDSUB4000:REM BILDAUFBAU
550 GDT010
700 PU=PU+LV:GS=GS+1
710 PRINT" {HOME BLACK DOWN}";TAB(13);PU;TAB(21);
" {WHITE}";GS
730 SOUND1,330,7:FORI=1TO110:NEXT:SOUND1,330,7:S
OUND1,500,15
735 IFGS=26THENP=20:LV=LV+1:IFLV>10THENLV=10
736 IFGS=26THENGS=0:FORI=1TO1000:NEXT:GDSUB4000:
Z=0
740 GOTO10
800 HB=HB-1:PRINT" {HOME DOWN D.BLUE}";TAB(30);HB
810 FORI=1TO8:CHAR1,P,6," {YELLOW S* SA}":SOUND3,
400,7
820 FORJ=1TO20:NEXT:CHAR1,P,6," {BLACK S* SA}":SD
UND3,400,7
830 FORJ=1TO20:NEXT:VOLB-I:NEXT:CHAR1,P,6," {SPAC
E2}"
835 IFHB=0THENB50
840 FORI=1TO500:NEXT:VOL3:P=20:CHAR1,20,6," {D.BL
UE S* SA}":GOTO10
850 SCNCLR:CHAR1,15,9," {BLACK FLASHON}GAME {SPACE
}OVER {FLASHOFF}"
852 IFRK<PUTHENCHAR1,11,12," {WHITE}NEUER {SPACE}R
EKORD: "+STR$(PU):RK=PU
855 VOL3:FORI=1TO6:SOUND1,800-I*100,7:SOUND2,700
-I*100,7:NEXT
856 FORI=1TO2000:NEXT
860 P=20:VOL3:Z=0:PU=0:LV=1:GS=0:HB=3:GDSUB3500:
GDSUB4000:GDT010
990 END
1000 REM *** NEUE ZEICHEN ***
1010 FORI=832TO849:READX:POKEI,X:NEXT:SYS832
1020 DATA 162,0,189,0,208,157,0,48,189,0,209,157
,0,49,202,208,241,96
1030 FORI=1280TO12855:READX:POKEI,X:NEXT
1040 DATA 255,4,63,88,156,127,145,127
1050 DATA 225,3,255,254,192,128,0,128
1060 DATA 56,186,146,124,56,56,68,130
1070 DATA 48,72,24,60,126,126,60,24
1080 DATA 36,60,36,60,36,60,36,60
1090 DATA 0,251,251,251,0,191,191,191
1100 DATA 0,0,0,0,0,68,85,255
1110 POKE65298,PEEK(65298)AND251
1120 POKE65299,PEEK(65299)AND30R48
1130 DATA169,1,262,1,345,2,169,2,383,2,262,1,516
,1,453,3,345,2,169,1,262,1,345,2
1140 DATA169,1,262,1,345,2,169,1,200,1,262,3,383
,1,453,1,516,2,383,2,596,2,516,1
1150 DATA383,1,453,3,345,1,345,1,383,1,410,1,453
,3,345,1,345,1,383,1,410,1,453,3
1160 DATA262,2
1799 RETURN
2000 REM *** INITIALISIERUNG ***
2010 PU=0:HB=3:DIMM(15),N(36),L(36)
2020 LV=1:GS=0:Z=10
2030 VOL3:FORI=0TO36:READN(I),L(I):NEXTI
2040 DEF FN W(X)=INT(RND(1)*X)
2050 G$(0)=" {SPACE}":G$(1)=" {WHITE SB}":G$(2)=" {
BLACK SC}":G$(3)=" {BLACK SC}":G$(4)=" {BLACK SC}"
:G$(5)=" {SPACE}":G$(6)=" {WHITE SB}"
2999 RETURN
3000 SCNCLR:PRINT" {HOME DOWN RED}";:COLOR0,1:COL
OR4,1
3005 PRINT" {SPACE4 SD SPACE2 SD SPACE2 SE2 SPACE
2 SE3 SPACE SE3 SPACE2 SE2 SPACE3 SC2 SPACE2 SE3
}"
3010 PRINT" {SPACE4 SD SPACE2 SD SPACE SE SPACE2
SE SPACE SE SPACE4 SD SPACE2 SE SPACE2 SE SPACE
SC SPACE4 SE}"
3020 PRINT" {SPACE4 SD SE2 SD SPACE SE SPACE2 SE
SPACE SE3 SPACE2 SD SPACE2 SE SPACE2 SE SPACE SC
SPACE SC SPACE2 SE2}"
3030 PRINT" {SPACE4 SD SPACE2 SD SPACE SE SPACE2

```

```

<73> SE SPACE3 SE SPACE2 SD SPACE2 SE4 SPACE SC SPACE
<15> 2 SC SPACE SE"
<214> 3040 PRINT" {SPACE4 SD SPACE2 SD SPACE2 SE2 SPACE
<7> 2 SE3 SPACE2 SD SPACE2 SE SPACE2 SE SPACE2 SC2 S
<107> PACE2 SE3}"
<111> 3050 PRINT" {DOWN2 WHITE SPACE3}RETTE {SPACE}DIE {S
<206> PACE}GEISELN {SPACE}AUS {SPACE}DEM {SPACE}HOCHHAUS!
<218> "
<111> 3060 PRINT" {DOWN3 L.GRN SPACE5}1986 {SPACE}A {SPAC
<196> E}SWORDMASTER {SPACE}PRODUCTION"
<54> 3070 PRINT" {DOWN SPACE3}...AND {SPACE}THE {SPACE}A
<220> DVENTURE {SPACE}CONTINUES..."
<215> 3075 PRINT" {DOWN SPACE7}PROGRAMM {SPACE}VON {SPACE
<163> }ACHIM {SPACE}LEIDIG"
<91> 3080 PRINT" {DOWN2 PINK SPACE9} {DRUECKE {SPACE}EIN
<35> E {SPACE}TASTE}"
3090 B=-1
<137> 3100 B=B+1:SOUND1,N(B),L(B)*10:SOUND2,N(B)+5,L(B
)*10:IFB=36THENB=-1
<138> 3105 SOUND1,0,1:SOUND2,0,1
<44> 3110 GETT$:IFT$=""THEN3100
<14> 3500 COLOR0,1:COLOR4,1
<89> 3505 SCNCLR:PRINT" {WHITE SPACE9 SC3 SPACE}H {SPAC
<141> E}D {SPACE}S {SPACE}T {SPACE}A {SPACE}G {SPACE}E {SPAC
E SC3}"
3510 PRINT" {DOWN PINK SPACE3}GANGSTER {SPACE}HABE
<179> N {SPACE}SICH {SPACE}MIT {SPACE}26 {SPACE}GEISELN"
3520 PRINT" {DOWN SPACE}IN {SPACE}EINEM {SPACE}HOCH
<52> HAUS {SPACE}VERSCHANZT. {SPACE}DU {SPACE}HAST"
3530 PRINT" {DOWN}DIE {SPACE}AUFGABE {SPACE}MIT {SPA
<169> CE}DEINEM {SPACE}HUBSCHRAUBER {SPACE} {D.BLUE S* S
A PINK}"
<12> 3540 PRINT" {DOWN SPACE}DIE {SPACE}GEISELN {SPACE} (
{WHITE SB PINK}) {SPACE}AUS {SPACE}DEM {SPACE}HOCHH
<31> AUS {SPACE}ZU"
3550 PRINT" {DOWN}RETTEN. {SPACE}DAZU {SPACE}MUSST {
<220> SPACE}DU {SPACE}EINE {SPACE}STRICKLEITER"
3560 PRINT" {DOWN SPACE}HERUNTERLASSEN. {SPACE}GIB
<238> {SPACE}JEDOCH {SPACE}ACHT, {SPACE}DASS"
3570 PRINT" {DOWN SPACE}DIE {SPACE}GANGSTER {SPACE}
<39> KEINE {SPACE}BOMBEN {SPACE} {L.BLU SC PINK}) {SPACE
JAN {SPACE}DIE"
<225> 3580 PRINT" {DOWN SPACE2}LEITER {SPACE}HAENGEN, {SP
ACE}SONST {SPACE}FLIEGT {SPACE}DEIN"
<98> 3590 PRINT" {DOWN SPACE7}HUBSCHRAUBER {SPACE}IN {SP
ACE}DIE {SPACE}LUFT."
<60> 3600 PRINT" {DOWN} (STEUERUNG {SPACE}MIT {SPACE}CURS
ORTASTEN {SPACE}UND {SPACE}SPACE)"
<18> 3640 PRINT" {DOWN CYAN SPACE11} (TASTE {SPACE}DRUEC
KEN)"
<230> 3650 B=-1
<187> 3652 B=B+1:SOUND1,N(B),L(B)*10:SOUND2,N(B)+5,L(B
)*10:IFB=36THENB=-1
<180> 3654 SOUND1,0,1:SOUND2,0,1
<83> 3656 GETT$:IFT$=""THEN3652
<215> 3670 COLOR0,14,6:COLOR4,7,2
<224> 3999 RETURN
<60> 4000 REM *** BILDAUFBAU ***
<116> 4005 SCNCLR:P=20:GDSUB4500
<22> 4010 FORI=1TO9STEP2
<95> 4020 CHAR1,B,8+I," {RED SE23}"
<82> 4030 CHAR1,B,9+I," {SE SPACE SE SPACE SE SPACE SE
SPACE SE SPACE SE SPACE SE SPACE SE SPACE SE SP
ACE SE SPACE SE SPACE SE}"
<95> 4040 NEXTI
<235> 4050 CHAR1,8,19," {SE23}"
<187> 4060 CHAR1,0,19," {GREEN SF8}"
<254> 4070 CHAR1,31,19," {SF10}"
<47> 4080 FORI=1TO4
<88> 4090 CHAR1,0,19+I," {RVSON SPACE40}"
<145> 4100 NEXT
<150> 4110 CHAR1,8,8," {RVSOFF BRN SF23}"
<8> 4120 PRINT" {HOME DOWN RIGHT6 BLACK}PUNKTE: ";PU;
<187> 4130 PRINTTAB(19);" {WHITE SB}";:GS;
<183> 4140 PRINTTAB(27);" {D.BLUE S* SA}";:HB
<167> 4150 CHAR1,P,6," {D.BLUE S* SA}"
<125> 4499 RETURN
<50> 4500 PRINT" {DOWN9}";TAB(10)" {BLACK}ACHTUNG! {SPAC
E}EINSATZ {SPACE}IN"
<146> 4510 PRINT" {DOWN}";TAB(16)" {WHITE}LEVEL: ";LV
<112> 4512 PRINT" {DOWN}";TAB(9)" {BLACK}BISHERIGER {SPAC
E}REKORD: ";:PRINTUSING"####";RK
<232> 4515 SOUND1,700,10:SOUND1,600,10:SOUND1,700,10
<100> 4520 FORI=1TO2000:NEXT:SCNCLR
<224> 4999 RETURN
<24> ENDE DES LISTINGS

```


plus 4
C16/116 + 64K

Adress 16

Im Grunde genommen ist das Programm selbsterklärend. Nachdem man das Programm gestartet und die "F1"-Taste gedrückt hat, befindet man sich im Hauptmenü. Erklärung der einzelnen Punkte:

1. Abrufen Kunden von Diskette:

Bevor man diesen Punkt aufruft, muß man eine Diskette mit den Adressdaten einlegen. Dann wird die Datei mit dem Namen "Adressdatei" gesucht und geladen. Möchte man eine alte Datei erweitern, so lädt man diese zuerst ein und speichert sie hinterher wieder neu ab.

2. Speichern Kunden auf Diskette:

Bei diesem Punkt wird man sicherheitshalber noch einmal gefragt, ob wirklich abgespeichert werden soll. Andernfalls gelangt man mit "F1" ins Hauptmenü. Achtung: Eine evtl. vorhandene Datei mit dem Namen "Adressdatei" wird gelöscht.

3. Kundenliste Schirm/Drucker

Bei diesem Punkt gelangt man

in ein Untermenü.

NBildschirm - Namensliste: Hierbei wird auf dem Bildschirm der Name sowie deren Kundennummer ausgegeben (gesamter Speicher).

ABildschirm - Adressliste: Hierbei werden je 3 komplette Adressen mit Kundennummer auf dem Bildschirm ausgegeben. Mit der "+"-Taste kann man 3 Adressen weiterschalten, mit "-" 3 Adressen zurück.

DDrucker - Adressliste: Ähnlich wie Punkt A, aber alle Adressen werden ausgedruckt.

EDrucker - Etiketten: Hierbei werden alle Adressen als Etiketten (ohne Telefonnummer) gedruckt, welche man direkt auf Briefumschläge kleben kann.

FDrucker - Namensliste: wie Punkt N, nur auf Drucker.

4. Suche nach Kundenname:

Nach Eingabe des Namens wird danach gesucht. Es reicht auch, nur den Vornamen anzugeben.

5. Suche nach Kundennummer:

Nach Eingabe der Kundennummer wird der Kunde gesucht.

6. Änderungen:

Nach Eingabe des Namens ist es möglich, alle gemachten Angaben zu ändern.

7. Neuer Kundenname.

Nach Aufruf dieses Punktes werden 10 neue Kundennummern vorgegeben, welche man verwenden kann. Hat man eine gewählt, so gibt man nacheinander den Kundenname, die Straße, die Haus-Nr., die PLZ, den Wohnort, sowie die Telefonnummer an. Anschließend kann man neue Kunden eingeben oder aber ins Menü zurückkehren.

8. Lösche alle Kunden

Hierbei werden alle Kunden im Speicher gelöscht. Eine Sicherheitsabfrage ist vorhanden.

9. Lösche einen Kunden

Wie Punkt 8, aber für einen Kunden.

A. Auswahl nach PLZ oder Name

Bei diesem Punkt kann man

speziell nach allen Namen suchen, die z. B. mit A anfangen oder mit BL anfangen. Ebenso verfährt man bei den Postleitzahlen. Man kann z. B. den ganzen Bereich, der mit 6 anfängt durchschauen oder aber noch feiner unterteilen z. B. 6110.

B. Directory

Inhalt der im Laufwerk befindlichen Diskette.

F7. Ende des Programmes

Mit "Goto 35" kommt man ohne Datenverlust wieder ins Hauptmenü.

Das Programm hat eine Länge von 12313 Bytes. Somit stehen bei der 64-K-Version noch 48356 Bytes für Daten frei. Von vornherein ist das Programm auf 200 Adressen festgelegt. Durch geringfügige Änderungen ist es aber jederzeit erweiterbar. Sollte jemand vor Start des Programmes den Graphic-Modus aufrufen, so ändert der C16 seine Speicherverwaltung und es stehen dann nur noch 36068 Bytes zur Verfügung. Das Programm ist für Kassetten- und Diskettenbetrieb ausgelegt. Für die Datensätze sind folgende Zeilen wie folgt zu ändern:

1008 OPEN 7, 1, 7, "ADRESS-DATEI, SEQ, READ"

Zeile 2001 löschen

2008 OPEN 7, 1, 7, "ADRESS-DATEI, SEQ, WRITE"

Damit wäre eigentlich alles beschrieben und wir wünschen viel Spaß mit "Adress 16".

C-16 LISTING + CHECKSUMMEN (OC V1.0)

```

1 COLOR 0,2,7:COLOR 4,2,5:COLOR 1,2,2 <92>
2 KEY 1,CHR$(133):KEY 7,CHR$(140) <73>
3 PRINT "{CLEAR DOWN3 SPACE7 RVSON SPACE24 RVSOFF}" <98>
4 PRINT "{SPACE7 RVSON SPACE4 SA SPACE SD SPACE SR SPACE SE SPACE SS SPACE SS SPACE2}1{SPACE}6{SPACE4 RVSOFF}" <37>
5 PRINT "{SPACE7 RVSON SPACE24 RVSOFF}" <20>
6 PRINT "{SPACE7 RVSON SPACE4}{(SC)}DPYRIGHT{SPACE}1986{SPACE4 RVSOFF}" <112>
7 PRINT "{SPACE7 RVSON SPACE24 RVSOFF}" <22>
8 PRINT "{SPACE7 RVSON SPACE4 SH}ANSPETER{SPACE S2}DELLE{SPACE4 RVSOFF}" <0>
9 PRINT "{SPACE7 RVSON SPACE24 RVSOFF}" <24>
10 PRINT:PRINT:PRINT:PRINT:PRINT <215>
11 PRINT "{SPACE7 RVSON SPACE24 RVSOFF}" <26>
12 PRINT "{SPACE7 RVSON SPACE4 SF}1{SPACE ST}AST E{SPACE}DRUECKEN{SPACE3 RVSOFF}" <148>
13 PRINT "{SPACE7 RVSON SPACE24 RVSOFF}",CHR$(14) <208>
14 GET W$:IF W$="" THEN 14 <161>
15 IF W$=CHR$(133) THEN 20 <152>
16 GOTO 14 <151>
20 EN=0 <203>
30 DIM A$(200),NA$(200),AN(200),AD$(200),TE$(200),AD$(200),WI$(200),DA$(200) <238>
34 REM <177>
35 REM *** HAUPTMENUE ***** <191>
36 REM <179>

```

```

40 PRINT CHR$(14) <133>
47 PRINT "{CLEAR}:" <116>
48 PRINT "{DOWN RVSON SPACE}1{SPACE RVSOFF SPACE }-{SPACE SA}BRUFEN{SPACE SK}UNDEN{SPACE}VON{SPACE S2}ISKETTE" <73>
49 PRINT <202>
50 PRINT "{RVSON SPACE}2{SPACE RVSOFF SPACE}-{SPACE SS}PEICHERN{SPACE SK}UNDEN{SPACE}AUF{SPACE S2}ISKETTE" <54>
55 PRINT <208>
60 PRINT "{RVSON SPACE}3{SPACE RVSOFF SPACE}-{SPACE SK}UNDENLISTE{SPACE SS}CHIRM/{SD}RUCKER" <19>
65 PRINT <218>
70 PRINT "{RVSON SPACE}4{SPACE RVSOFF SPACE}-{SPACE SS}UCHE{SPACE}NACH{SPACE SK}UNDENNAME" <163>
75 PRINT <228>
80 PRINT "{RVSON SPACE}5{SPACE RVSOFF SPACE}-{SPACE SS}UCHE{SPACE}NACH{SPACE SK}UNDENNUMMER" <100>
85 PRINT <238>
90 PRINT "{RVSON SPACE}6{SPACE RVSOFF SPACE}-{SPACE SA}ENDERUNGEN" <10>
95 PRINT <248>
100 PRINT "{RVSON SPACE}7{SPACE RVSOFF SPACE}-{SPACE SN}EUER{SPACE SK}UNDENNAME" <187>
105 PRINT <2>
110 PRINT "{RVSON SPACE}8{SPACE RVSOFF SPACE}-{SPACE}LOESCHE{SPACE}ALLE{SPACE SK}UNDEN" <137>
115 PRINT <12>
120 PRINT "{RVSON SPACE}9{SPACE RVSOFF SPACE}-{SPACE}LOESCHE{SPACE}1{SPACE SK}UNDEN" <185>

```



```

122 PRINT <19>
125 PRINT "{RVSON SPACE SA SPACE RVSOFF SPACE}-(<81>
SPACE SA)USWAHL(SPACE)NACH(SPACE SP SL SZ SPACE)
ODER(SPACE SN)AME" <23>
126 PRINT <170>
127 PRINT "{RVSON SPACE SB SPACE RVSOFF SPACE}-(<25>
SPACE SD)IRECTORY"
128 PRINT <97>
130 PRINT "{RVSON SPACE SE}7{RVSOFF SPACE}-(<206>
SPACE SE)NDE(SPACE)DES(SPACE SP)ROGRAMMS"
141 GET W$:IF W$="" THEN 141 <218>
142 IF W$="1" THEN 1000 <234>
143 IF W$="2" THEN 2000 <4>
144 IF W$="3" THEN 3100 <19>
145 IF W$="4" THEN 5000 <35>
146 IF W$="5" THEN 6000 <51>
147 IF W$="6" THEN 7000 <67>
148 IF W$="7" THEN 8000 <83>
149 IF W$="8" THEN 9000 <130>
150 IF W$="9" THEN 10000 <199>
151 IF W$="A" THEN 12000 <216>
152 IF W$="B" THEN 13000 <128>
153 IF W$=CHR$(140) THEN 11000 <235>
160 GOTO 141 <67>
180 REM <141>
181 PRINT:PRINT
183 PRINT "{RVSON SM}ENUE(SPACE6)--(SPACE SE)1{R<154>
VSOFF}" <87>
184 GET W$:IF W$="" THEN 184 <144>
186 IF W$=CHR$(133) THEN 35 <31>
188 GOTO 184 <120>
998 REM <159>
999 REM *** LADEN ***** <122>
1000 REM <50>
1002 PRINT "{CLEAR}"; <49>
1006 PRINT "{RVSON SA}DRESSENLISTE(SPACE)WIRD(SP<252>
ACE)GESUCHT" <228>
1007 PRINT:PRINT "{RVSON SB}ITTE(SPACE)WARTEN!" <4>
1008 OPEN 7,8,7,"ADRESSDATEI,SEQ,READ" <244>
1009 INPUT#7,EN <154>
1010 FOR X=1 TO EN <77>
1015 INPUT#7,AN(X) <19>
1018 IF AN(X)=0 THEN 1030 <240>
1020 INPUT#7,NA$(X) <66>
1022 INPUT#7,AD$(X) <42>
1023 INPUT#7,TE$(X) <101>
1024 INPUT#7,AD$(X) <242>
1025 INPUT#7,WI$(X) <60>
1026 INPUT#7,DA$(X) <34>
1030 NEXT X <172>
1040 CLOSE 7 <100>
1050 GOTO 35 <233>
1998 REM <102>
1999 REM *** SPEICHERN *****
2000 REM
2001 PRINT "{CLEAR DOWN2 RVSON SPACE SS}PEICHERN(<8>
SPACE S)--(SPACE2 SR)ETURN(SPACE2)":PRINT "{RVSON<210>
SPACE S}ENUE(SPACE2)--(SPACE2 SF SPACE)1(SPACE<177>
S)"
2002 GET W$:IF W$="" THEN 2002
2003 IF W$<>CHR$(13) THEN 35
2004 OPEN 15,8,15:PRINT#15,"S:ADRESSDATEI":CLOSE<170>
15
2005 PRINT "{CLEAR RVSON SN}AME(SPACE)DER(SPACE<154>
SD)ATEI(SPACE)=(SPACE SA)DRESSDATEI(SPACE RVSOFF<205>
S)"
2006 PRINT "{DOWN RVSON SA}DRESSENLISTE(SPACE)WI<116>
RD(SPACE)GESPEICHERT"
2007 PRINT:PRINT "{RVSON SB}ITTE(SPACE)WARTEN" <42>
2008 OPEN 7,8,7,"ADRESSDATEI,SEQ,WRITE" <4>
2009 PRINT#7,EN <224>
2010 FOR X=1 TO EN <154>
2015 PRINT#7,AN(X) <67>
2018 IF AN(X)=0 THEN 2030 <19>
2020 PRINT#7,NA$(X) <240>
2022 PRINT#7,AD$(X) <66>
2023 PRINT#7,TE$(X) <41>
2024 PRINT#7,AD$(X) <100>
2025 PRINT#7,WI$(X) <241>
2026 PRINT#7,DA$(X) <39>
2030 NEXT X <13>
2040 CLOSE 7 <152>
2050 GOTO 35 <82>
3000 REM <93>
3001 PRINT
3002 PRINT "{RVSON SA}DRESSENLISTE(SPACE)DRUCKEN
(SPACE2)--(SPACE SJ SPACE RVSOFF)" <136>
3003 PRINT "{RVSON SM}ENUE(SPACE18)--(SPACE SF)1<37>
(RVSOFF)" <234>
3004 GET W$:IF W$="" THEN 3004 <197>
3005 IF W$="J" THEN 4000 <159>
3006 IF W$=CHR$(133) THEN 35 <13>
3007 GOTO 3004 <18>
3010 PRINT "{CLEAR}"; <35>
3015 Z=1 <216>
3022 FOR X=1 TO EN <101>
3023 IF AN(X)=0 THEN 3033 <25>
3024 IF AN(X)<=9 THEN 3027 <51>
3025 IF AN(X)>=10 AND AN(X)<=99 THEN 3028 <187>
3026 IF AN(X)>=100 THEN 3029
3027 PRINT "{SPACE2}";:PRINT AN(X);:PRINT "{SPAC<63>
E3}";:PRINT NA$(X):GOTO 3030
3028 PRINT "{SPACE}";:PRINT AN(X);:PRINT "{SPACE<179>
3}";:PRINT NA$(X):GOTO 3030
3029 PRINT AN(X);:PRINT "{SPACE3}";:PRINT NA$(X) <149>
3030 IF Z/20=INT(Z/20) THEN 3050 <169>
3032 Z=Z+1 <76>
3033 NEXT X <22>
3035 PRINT <127>
3040 GOTO 3000 <26>
3050 REM <132>
3052 PRINT <144>
3053 PRINT "{RVSON}WEITERE(SPACE)20(SPACE SN)AME<30>
N(SPACE2)--(SPACE2 SR)ETURN(SPACE4 RVSOFF)"
3054 PRINT "{RVSON SM}ENUE(SPACE13)--(SPACE2 SF<155>
SPACE)1(SPACE2Z RVSOFF)"
3060 Z=1 <80>
3061 GET W$:IF W$="" THEN 3061 <74>
3062 IF W$=CHR$(133) THEN 35 <215>
3063 IF W$=CHR$(13) THEN 3065 <61>
3064 GOTO 3061 <79>
3065 PRINT "{CLEAR}"; <73>
3066 GOTO 3033 <79>
3098 REM <181>
3099 REM *** KUNDENLISTE ANZEIGE ***** <42>
3100 REM <183>
3110 PRINT "{CLEAR}"; <119>
3120 PRINT "{RVSON SPACE SN SPACE RVSOFF SPACE S<102>
B}ILDSCHIRM(SPACE)-(SPACE SN)AMENSLISTE"
3122 PRINT <215>
3124 PRINT "{RVSON SPACE SA SPACE RVSOFF SPACE S<242>
B}ILDSCHIRM(SPACE)-(SPACE SA)DRESSENLISTE"
3126 PRINT:PRINT:PRINT <255>
3127 PRINT "{RVSON SPACE SD SPACE RVSOFF SPACE S<95>
B}RUCKER(SPACE4)-(SPACE SA)DRESSENLISTE
3128 PRINT <221>
3130 PRINT "{RVSON SPACE SE SPACE RVSOFF SPACE S<96>
B}RUCKER(SPACE4)-(SPACE SE)TIKETTEN"
3131 PRINT "{DOHN RVSON SPACE SF SPACE RVSOFF SP<137>
ACE SD}RUCKER(SPACE4)-(SPACE SN)AMENSLISTE"
3132 PRINT:PRINT <32>
3134 PRINT "{RVSON SPACE SF}1(SPACE RVSOFF SPACE<39>
SN)ENUE"
3140 GET W$:IF W$="" THEN 3140 <124>
3142 IF W$="N" THEN 3010 <105>
3143 IF W$="F" THEN 3600 <107>
3144 IF W$="D" THEN 4000 <45>
3145 IF W$="E" THEN 3500 <93>
3146 IF W$="A" THEN 3200 <40>
3147 IF W$=CHR$(133) THEN 35 <45>
3148 GOTO 3140 <154>
3200 REM <27>
3201 X=1 <220>
3205 PRINT "{CLEAR}"; <214>
3210 FOR X=X TO X+2 <183>
3211 IF X>200 THEN 3260 <212>
3212 IF X<=9 THEN PRINT "{SPACE2}";X;:GOTO 3215 <234>
3213 IF X<=99 THEN PRINT "{SPACE}";X;:GOTO 3215 <131>
3214 IF X>=100 THEN PRINT X; <167>
3215 IF NA$(X)="" OR NA$(X)="{SPACE}" THEN 3350 <72>
3220 PRINT SPC(1) NA$(X) <182>
3230 PRINT SPC(6) AD$(X); "{SPACE}"; TE$(X) <68>
3240 PRINT SPC(6) AD$(X); "{SPACE}"; WI$(X) <4>
3250 PRINT SPC(6) "{ST}EL.: {SPACE}" DA$(X) <0>
3252 PRINT:PRINT <152>
3260 NEXT X <250>
3262 PRINT "{HOME DOHN20}";:PRINT "{RVSON SPACE}<8>
+(SPACE7)=(SPACE2)3(SPACE2 SA)DRESSEN(SPACE2)WEI<8>
TER(SPACE)"
3263 PRINT "{RVSON SPACE)-(SPACE7)=(SPACE2)3(SPA<3>
CE2 SA)DRESSEN(SPACE2)ZURUECK"
3264 PRINT "{RVSON SPACE SF}1(SPACE6)=(SPACE2 SN

```


programme

```

)ENUE (SPACE15)"
3270 GET W$:IF W$="" THEN 3270
3272 IF W$="+" THEN 3205
3274 IF W$="-" THEN 3300
3276 IF W$=CHR$(133) THEN 35
3278 GOTO 3270
3300 REM
3302 X=X-6
3303 IF X<=1 THEN X=1
3304 GOTO 3205
3350 REM
3352 PRINT SPC(1) "{SK}UNDENNUMMER{SPACE2}";X
3353 PRINT SPC(6) "NICHT"
3354 PRINT SPC(6) "BELEGT{SPACE}!"
3355 PRINT
3356 GOTO 3252
3500 REM
3505 OPEN 1,4,7
3540 XX=1
3550 FOR X=1 TO EN STEP 2
3552 PRINT#1,NA$(X);
3553 PRINT#1,CHR$(141)TAB(40);NA$(X+1)
3555 PRINT#1
3560 PRINT#1,AD$(X);"{SPACE}";TE$(X);
3561 PRINT#1,CHR$(141)TAB(40);AD$(X+1);"{SPACE}";
TE$(X+1)
3563 PRINT#1
3570 PRINT#1,AD$(X);"{SPACE2}";WI$(X);
3571 PRINT#1,CHR$(141)TAB(40);AD$(X+1);"{SPACE2}";
WI$(X+1)
3575 PRINT#1:PRINT#1:PRINT#1
3576 IF X/8=INT(X/8) THEN PRINT#1,CHR$(12);
3578 NEXT X
3594 CLOSE 1
3595 GOTO 47
3600 OPEN 1,4,7
3620 FOR X=1 TO EN
3630 IF AN(X)=0 THEN 3710
3640 IF AN(X)<=9 THEN 3680
3650 IF AN(X)>=10 AND AN(X)<=99 THEN 3690
3660 IF AN(X)>100 THEN 3700
3680 PRINT#1,"{SPACE2}";AN(X);"{SPACE3}";NA$(X);
GOTO 3710
3690 PRINT#1,"{SPACE}";AN(X);"{SPACE3}";NA$(X);G
OTO 3710
3700 PRINT#1,AN(X);"{SPACE3}";NA$(X)
3710 NEXT X
3720 GOTO 3100
4000 REM
4001 PRINT "{CLEAR}":PRINT:PRINT
4002 PRINT "{SM}ELCHE{SPACE SK}UNDEN-NUMMERN{SPA
CE}DRUCKEN{SPACE}?" :PRINT:PRINT
4003 PRINT "{RVSON SPACE SA SPACE}={SPACE}ALLE{S
PACE SA}DRESSEN{SPACE2}"
4004 PRINT "{RVSON SPACE SY SPACE}={SPACE SN}UMM
ER{SPACE2}Y{SPACE2}-{SPACE2}Z{SPACE7}":PRINT "{RV
SON SPACE SF}1{SPACE}={SPACE SN}ENUE{SPACE16 RVS
OFF}"
4005 GET W$:IF W$=CHR$(133) THEN 35
4006 IF W$="A" THEN 4009
4007 IF W$="Y" THEN 4070
4008 GOTO 4005
4009 PRINT "{DOHN RVSON SD}RUCKER{SPACE}IN{SPACE
SB}ETRIEB{SPACE}-{SPACE}BITTE{SPACE}WARTEN!{RVS
OFF}"
4010 OPEN 1,4,7
4017 FOR X=1 TO EN
4018 IF AN(X)=0 THEN 4040
4020 PRINT#1,"{SPACE SK}D-NR.: {SPACE}AN(X)
4021 PRINT#1,"{SPACE}";NA$(X):PRINT#1
4022 PRINT#1,"{SPACE}";AD$(X);
4023 PRINT#1,TE$(X)
4024 PRINT#1,"{SPACE}";AD$(X);:PRINT#1,"{SPACE2}";
";
4025 PRINT#1,WI$(X)
4026 PRINT#1,"{SPACE ST}EL: {SPACE}";DA$(X)
4028 PRINT#1:PRINT#1:PRINT#1
4040 NEXT X
4050 CLOSE 1
4060 GOTO 35
4070 PRINT:PRINT
4072 INPUT "Y: {SPACE2}";Y
4074 INPUT "Z: {SPACE2}";Z
4075 IF Z<Y THEN 4070
4076 GOTO 4100
4100 REM
<24> 4110 OPEN 1,4,7
<53> 4117 FOR X=Y TO Z
<94> 4118 IF AN(X)=0 THEN 4140
<58> 4120 PRINT#1,"{SPACE SK}D-NR.: {SPACE}AN(X)
<174> 4121 PRINT#1,"{SPACE}";NA$(X):PRINT#1
<43> 4122 PRINT#1,"{SPACE}";AD$(X);
<127> 4123 PRINT#1,TE$(X)
<112> 4124 PRINT#1,"{SPACE}";AD$(X);:PRINT#1,"{SPACE2}";
";
<159> ";
<66> 4125 PRINT#1,WI$(X)
<178> 4126 PRINT#1,"{SPACE}";DA$(X)
<3> 4128 PRINT#1:PRINT#1:PRINT#1
<44> 4140 NEXT X
<163> 4150 CLOSE 1
<193> 4160 GOTO 35
<124> 4998 REM
<72> 4999 REM *** SUCHEN NAME *****
<57> 5000 REM
<195> 5005 PRINT "{CLEAR}";
<244> 5010 PRINT "{RVSON SN}AME{SPACE}DES{SPACE SK}UND
EN"
<87>
<53> 5012 PRINT
<234> 5015 INPUT N$
<39> 5020 FOR X=1 TO EN
5025 IF LEFT$(NA$(X),LEN(N$))=N$ THEN 5060
5030 NEXT X
5035 PRINT
5040 PRINT "{RVSON SK}UNDE{SPACE}NICHT{SPACE}GEF
UNDEN"
5045 PRINT
5050 PRINT "{RVSON SN}AME{SPACE}NEU{SPACE}EINGEB
EN{SPACE2}!!!{RVSOFF}"
5052 FOR T=1 TO 1500:NEXT T
5053 GOTO 35
5060 PRINT "{CLEAR}";
5070 PRINT "{SK}UNDENNUMMER";AN(X):PRINT:PRINT:P
RINT:PRINT NA$(X)
5075 PRINT
5080 PRINT AD$(X);"{SPACE}";TE$(X)
5085 PRINT
5090 PRINT AD$(X);"{SPACE}";WI$(X)
5095 PRINT:PRINT
5100 PRINT "{ST}ELEFON: ";"{SPACE}";DA$(X)
5105 PRINT:PRINT
5111 PRINT "{RVSON SA}DRESSE{SPACE}DRUCKEN{SPACE
2}--{SPACE2 SJ SPACE2 RVSOFF}"
5112 PRINT "{RVSON SN}ENUE{SPACE12}--{SPACE2 SF}
1{SPACE RVSOFF}"
5115 GET W$:IF W$="" THEN 5115
5116 IF W$="J" THEN 5300
5117 IF W$=CHR$(133) THEN 35
5118 GOTO 5115
5130 GOTO 180
5300 REM
5310 OPEN 1,4,7
5320 PRINT#1,NA$(X)
5322 PRINT#1
5324 PRINT#1,AD$(X);"{SPACE2}";
5325 PRINT#1,TE$(X)
5326 PRINT#1
5328 PRINT#1,AD$(X);"{SPACE3}";WI$(X)
5331 PRINT#1
5332 PRINT#1
5333 CLOSE 1
5340 GOTO 35
5997 REM
5998 REM *** SUCHEN NUMMER *****
5999 REM
6000 PRINT "{CLEAR}";:PRINT "{RVSON SN}IE{SPACE}
IST{SPACE}DIE{SPACE SK}UNDENNUMMER"
6001 PRINT
6002 INPUT X
6010 IF AN(X)=0 THEN 6200
6060 PRINT "{CLEAR}";
6070 PRINT "{SK}UNDEN-{SN}R. ";AN(X):PRINT:PRINT:
PRINT:PRINT NA$(X)
6075 PRINT
6080 PRINT AD$(X);"{SPACE}";TE$(X)
6085 PRINT
6090 PRINT AD$(X);"{SPACE}";WI$(X)
6095 PRINT:PRINT
6100 PRINT "{ST}ELEFON: {SPACE}";DA$(X)
6105 PRINT:PRINT
6130 GOTO 5105
6200 REM
6205 PRINT:PRINT:PRINT
<153>
<32>
<171>
<134>
<106>
<52>
<115>
<183>
<149>
<251>
<57>
<110>
<72>
<222>
<40>
<69>
<42>
<229>
<107>
<64>
<55>
<174>
<156>
<235>
<87>
<34>
<97>
<168>
<153>
<94>
<28>
<82>
<127>
<61>
<137>
<199>
<210>
<236>
<220>
<2>
<171>
<115>
<51>
<230>
<100>
<113>
<87>
<77>
<248>
<216>
<12>
<41>
<220>
<215>
<225>
<226>
<235>
<126>
<19>
<165>
<21>
<93>
<33>
<190>
<5>
<8>
<27>
<107>
<41>
<117>
<179>
<190>
<119>
<200>
<88>
<223>
<18>

```


6210 PRINT "{RVSON SK}UNDENUMMER{SPACE}NICHT{SPACE}BELEGT{SPACE}!!{RVSOFF}"	<165>	INT: INPUT AD\$(X)	<250>
6215 PRINT:PRINT	<55>	7260 GOTO 7080	<207>
6220 GOTO 180	<183>	7270 PRINT "{CLEAR}NEUER{SPACE SH}OHNDORT":PRINT:	
6997 REM	<255>	INPUT WI\$(X)	<113>
6998 REM *** AENDERN *****	<36>	7280 GOTO 7080	<227>
6999 REM	<1>	7290 PRINT "{CLEAR}NEUE{SPACE ST}ELEFONNUMMER":P	
7000 PRINT "{CLEAR RVSON SH}ELCHER{SPACE SK}UNDE	<121>	RINT: INPUT DA\$(X)	<244>
NNAME":PRINT:INPUT N\$	<124>	7295 GOTO 7080	<242>
7010 FOR X=1 TO EN	<153>	7400 REM	<147>
7020 IF LEFT\$(NA\$(X),LEN(N\$))=N\$ THEN 7060	<195>	7402 IF AN(U)=AN(X) THEN 35	<132>
7030 NEXT X	<47>	7410 AN(X)=0	<12>
7035 PRINT	<46>	7420 NA\$(U)=NA\$(X):NA\$(X)=""	<209>
7040 PRINT "{RVSON SK}UNDE{SPACE}NICHT{SPACE}GEF	<53>	7430 AD\$(U)=AD\$(X):AD\$(X)=""	<245>
UNDEN":PRINT	<113>	7440 TE\$(U)=TE\$(X):TE\$(X)=""	<226>
7050 PRINT "{RVSON SH}AME{SPACE}NEU{SPACE}EINGEB	<35>	7450 AD\$(U)=AD\$(X):AD\$(X)=""	<50>
EN{SPACE}!!{RVSOFF}	<8>	7460 WI\$(U)=WI\$(X):WI\$(X)=""	<170>
7052 FOR T=1 TO 1500:NEXT T	<55>	7470 DA\$(U)=DA\$(X):DA\$(X)=""	<20>
7055 GOTO 35	<94>	7480 GOTO 35	<227>
7060 AN(U)=AN(X)	<70>	7500 PRINT "{DOHN2 RVSON SPACE SD}IESE{SPACE SN}	
7062 GOTO 7080	<97>	UMMER{SPACE}IST{SPACE}SCHON{SPACE}VORHANDEN{SPAC	
7080 PRINT "{CLEAR}";:PRINT "{SK}UNDENUMMER";AN	<31>	E)!!{SPACE RVSOFF}"	<67>
(U)	<107>	7510 FOR I=1 TO 1500:NEXT I	<163>
7082 PRINT	<169>	7998 REM	<236>
7084 PRINT NA\$(X)	<117>	7999 REM *** NEUER KUNDE *****	<193>
7085 PRINT	<206>	8000 REM	<238>
7090 PRINT AD\$(X);"{SPACE}";TE\$(X)	<132>	8001 PRINT "{CLEAR}";	<165>
7095 PRINT	<33>	8004 Y=1	<179>
7100 PRINT AD\$(X);"{SPACE}";WI\$(X)	<135>	8005 FOR X=1 TO 200	<159>
7105 PRINT	<182>	8008 IF NA\$(X)="" THEN 8200	<181>
7110 PRINT "{ST}ELEFON: ";"{SPACE}";DA\$(X)	<85>	8011 IF Y=11 THEN 8014	<196>
7120 PRINT	<147>	8013 NEXT X	<158>
7122 PRINT "{RVSON SH}AS{SPACE}SOLL{SPACE}GEBEAND	<23>	8014 PRINT "{RVSON SV}ERWENDEN{SPACE}SIE{SPACE}F	
ERT{SPACE}WERDEN{SPACE}?{RVSOFF}"	<3>	OLGENDE{SPACE SK}UNDEN-NUMMERN{SPACE}: {RVSOFF}":	
7123 PRINT	<106>	PRINT	<94>
7125 PRINT "{RVSON SPACE SK SPACE RVSOFF SPACE S	<53>	8015 FOR X=1 TO Y-1	<248>
K}UNDENUMMER"	<41>	8017 PRINT PP(X)	<231>
7126 PRINT "{RVSON SPACE SN SPACE RVSOFF SPACE S	<156>	8019 NEXT X	<164>
N}AME"	<61>	8020 PRINT:INPUT "{RVSON SK}UNDEN-NUMMER{SPACE4	
7127 PRINT "{RVSON SPACE SS SPACE RVSOFF SPACE S	<115>	RVSOFF}";NF	<233>
S}TRASSE"	<168>	8021 FOR X=1 TO Y-1:IF NF=PP(X) THEN 8023	<2>
7128 PRINT "{RVSON SPACE SH SPACE RVSOFF SPACE S	<129>	8022 NEXT X:X=Y-1:GOTO 7500	<199>
H}AUSNUMMER"	<122>	8023 AN(NF)=NF:IF NF<=0 THEN 8000	<13>
7129 PRINT "{RVSON SPACE SP SPACE RVSOFF SPACE S	<79>	8024 IF NF>PP(10) THEN 8000	<59>
P}OSTLEITZAHL"	<150>	8025 REM	<7>
7131 PRINT "{RVSON SPACE SH SPACE RVSOFF SPACE S	<215>	8028 PRINT "{CLEAR}";:PRINT "{RVSON SK}UNDEN-NUM	
H}OHNDORT"	<220>	MER{SPACE4 RVSOFF}";NF	<25>
7133 PRINT "{RVSON SPACE ST SPACE RVSOFF SPACE S	<123>	8030 PRINT	<22>
I}ELEFONNUMMER"	<97>	8032 INPUT "{RVSON SK}UNDEN-NAME{SPACE5 RVSOFF S	
7135 PRINT "{RVSON SPACE SF}1{RVSOFF SPACE SH}EN	<71>	PACE}";NA\$(NF)	<110>
UE"	<109>	8033 IF NA\$(NF)="" THEN 8400	<40>
7144 PRINT	<35>	8034 PRINT	<26>
7145 PRINT "{RVSON SK}ENNBUCHSTABE{SPACE2}EINGEB	<88>	8036 INPUT "{RVSON SS}TRASSE{SPACE9 RVSOFF SPACE	
EN{SPACE}!!{RVSOFF}	<61>	}";AD\$(NF)	<91>
7150 GET W\$:IF W\$="" THEN 7150	<115>	8037 IF AD\$(NF)="" THEN AD\$(NF)="X"	<177>
7152 IF W\$="N" THEN 7190	<168>	8038 PRINT	<30>
7153 IF W\$="K" THEN 7170	<129>	8040 INPUT "{RVSON SH}AUS-NR. {SPACE8 RVSOFF SPAC	
7154 IF W\$="S" THEN 7210	<122>	E}";TE\$(NF)	<4>
7155 IF W\$="H" THEN 7230	<79>	8042 IF TE\$(NF)="" THEN TE\$(NF)="X"	<228>
7156 IF W\$="P" THEN 7250	<150>	8045 PRINT	<37>
7157 IF W\$="W" THEN 7270	<215>	8050 INPUT "{RVSON SP}OSTLEITZAHL{SPACE4 RVSOFF	
7158 IF W\$="T" THEN 7290	<220>	SPACE}";AD\$(NF)	<185>
7159 IF W\$=CHR\$(133) THEN 7400	<123>	8052 IF AD\$(NF)="" THEN AD\$(NF)="X"	<123>
7160 GOTO 7150	<97>	8055 PRINT	<47>
7170 PRINT "{CLEAR SH}EUE{SPACE SK}UNDENUMMER{S	<71>	8060 INPUT "{RVSON SH}OHNDORT{SPACE9 RVSOFF SPACE	
PACE2}?" :PRINT	<109>	}";WI\$(NF)	<131>
7171 PRINT "{RVSON}NUR{SPACE}FOLGENDE{SPACE SK}U	<35>	8062 IF WI\$(NF)="" THEN WI\$(NF)="X"	<105>
NDENUMMERN{SPACE}WAEHLEN{SPACE}: {RVSOFF}":PRINT	<133>	8065 PRINT	<57>
7172 FOR Z=1 TO EN	<88>	8070 INPUT "{RVSON ST}ELEFON-NR. {SPACE5 RVSOFF S	
7173 IF AN(Z)=0 THEN PRINT Z	<61>	PACE}";DA\$(NF)	<254>
7174 NEXT Z	<85>	8072 IF DA\$(NF)="" THEN DA\$(NF)="X"	<206>
7175 PRINT:PRINT "{RVSON SE}INGABE{SPACE}DER{SPA	<28>	8075 PRINT:PRINT:PRINT	<103>
CE SK}UNDENUMMER{SPACE}: {RVSOFF}":PRINT	<127>	8078 AN(NF)=NF	<82>
7178 INPUT U	<116>	8080 PRINT "{RVSON}WEITERE{SPACE2}NEUE{SPACE2 SK	
7179 AN(U)=U	<147>	}UNDEN{SPACE2}--{SPACE SJ SPACE RVSOFF}"	<203>
7180 GOTO 7080	<147>	8081 PRINT "{RVSON SH}ENUE{SPACE18}--{SPACE SF}1	
7190 PRINT "{CLEAR}";:PRINT "{RVSON}DER{SPACE}NE	<147>	(RVSOFF)"	<15>
UE{SPACE SH}AME":PRINT:INPUT NA\$(X)	<167>	8082 GET W\$:IF W\$="" THEN 8082	<98>
7200 GOTO 7080	<190>	8083 IF W\$="J" THEN 8000	<211>
7210 PRINT "{CLEAR}DIE{SPACE}NEUE{SPACE SS}TRASS	<187>	8084 IF W\$=CHR\$(133) THEN 8300	<27>
E":PRINT:INPUT AD\$(X)		8085 GOTO 8082	<23>
7220 GOTO 7080		8200 REM	<183>
7230 PRINT "{CLEAR}DIE{SPACE}NEUE{SPACE SH}AUSNU		8210 PP(Y)=X	<95>
MMER":PRINT:INPUT TE\$(X)		8220 Y=Y+1	<161>
7240 GOTO 7080		8230 GOTO 8011	<136>
7250 PRINT "{CLEAR}NEUE{SPACE SP}OSTLEITZAHL":PR		8300 REM	<27>
		8310 EN=PP(Y-1)	<103>

programme

```

8320 GOTO 35
8400 REM
8410 PRINT "{CLEAR}"
8420 PRINT "{RVSON SQ}HNE{SPACE SN}AMEN{SPACE}KE
INE{SPACE SR}EGISTRIERUNG{SPACE}!!!{RVSOFF}"
8425 AN(NF)=0
8426 NF=0
8430 PRINT
8440 GOTO 8004
8999 REM
9000 REM *** LOESCHUNG ALLE *****
9001 REM
9003 PRINT "{CLEAR}";PRINT "{RVSON}ALLE{SPACE SK
}UNDEN{SPACE}LOESCHEN{SPACE2}--{SPACE SJ SPACE R
VSOFF}"
9005 PRINT "{RVSON SH}ENUE{SPACE17}--{SPACE SF}1
{SPACE RVSOFF}"
9006 GET W$:IF W$="" THEN 9006
9007 IF W$="J" THEN 9020
9008 IF W$=CHR$(133) THEN 35
9009 GOTO 9006
9020 CLR
9030 RUN 20
9999 REM
10000 REM *** LOESCHUNG 1 KUNDE *****
10001 REM
10002 PRINT "{CLEAR}";PRINT "{RVSON SH}ELCHER{S
PACE SK}UNDE{SPACE}SOLL{SPACE}GELOESCHT{SPACE}WE
RDEN{SPACE2}?"
10005 PRINT: INPUT N$
10010 FOR X=1 TO EN
10020 IF LEFT$(NA$(X),LEN(N$))=N$ THEN 10060
10030 NEXT X
10040 PRINT:PRINT "{RVSON SK}UNDE{SPACE}NICHT{SP
ACE}GEFUNDEN"
10050 PRINT
10051 PRINT "{RVSON SH}AME{SPACE}NEU{SPACE}EINGE
BEN{SPACE}!!!{RVSOFF}"
10052 FOR T=1 TO 1500:NEXT T
10053 GOTO 35
10055 PRINT
10060 PRINT "{CLEAR}"
10070 PRINT "{SK}UNDEN-NUMMER";AN(X):PRINT:PRINT
NA$(X)
10075 PRINT
10080 PRINT AD$(X);"{SPACE}";TE$(X)
10090 PRINT
10100 PRINT AD$(X);"{SPACE}";WI$(X)
10105 PRINT
10110 PRINT "{ST}ELEFON-NR."; "{SPACE}";DA$(X)
10112 PRINT:PRINT
10114 PRINT "{RVSON}SOLL{SPACE SK}UNDE{SPACE}GEL
OESCHT{SPACE}WERDEN{SPACE}--{SPACE SJ SPACE RVSO
FF}"
10115 PRINT "{RVSON SH}ENUE{SPACE23}--{SPACE SF}
1{RVSOFF}"
10116 GET W$:IF W$="" THEN 10116
10117 IF W$="J" THEN 10120
10118 IF W$=CHR$(133) THEN 35
10119 GOTO 10116
10120 NA$(X)=" "
10130 AN(X)=0
10140 AD$(X)=" "
10150 TE$(X)=" "
10160 AO$(X)=" "
10170 WI$(X)=" "
10180 DA$(X)=" "
10200 GOTO 35
10999 REM
11000 REM *** ENDE *****
11001 REM
11002 PRINT CHR$(142)
11010 PRINT "{CLEAR DOWN8 RIGHT2 SPACE7 RVSON SP
ACE3}MIT{SPACE}GOTO{SPACE}35{SPACE}KOMMEN{SPACE4
RVSOFF}"
11011 PRINT "{HOME DOWN10 RIGHT2 SPACE7 RVSON SP
ACE2}SIE{SPACE}OHNE{SPACE}DATENVERLUST{SPACE2 RV
SOFF}"
11012 PRINT "{HOME DOWN12 RIGHT2 SPACE7 RVSON SP
ACE}IN'S{SPACE}HAUPTMENUE{SPACE}ZURUECK{SPACE RV
SOFF}"
11020 PRINT "{HOME DOWN16}"
11026 PRINT "GOTO{SPACE}35"
11030 PRINT "{UP4}";END
11999 REM
12000 REM *** AUSWAHL PLZ/NAME *****

```

```

<46> 12001 REM <158>
<127> 12002 PRINT "{CLEAR DOWN3 RVSON SP RVSOFF}OSTLEI <158>
<24> TZAHL";PRINT <222>
12003 PRINT "{RVSON SN RVSOFF}AME" <49>
12008 GET W$:IF W$="" THEN 12008 <127>
12009 IF W$="P" THEN 12015 <238>
12010 IF W$="N" THEN 12200 <172>
12011 GOTO 12008 <154>
12015 PRINT:PRINT <245>
12016 PRINT "{RVSON SS}ORTIERUNG{SPACE}NACH{SPAC
E SP}OSTLEITZAHL:{RVSOFF}" <83>
12017 PRINT:PZ$="" <98>
12020 INPUT "{RVSON SH}ELCHE{SPACE SP}OSTLEITZAH
L{SPACE6 RVSOFF SPACE}";PZ$ <162>
12022 PRINT <189>
12025 L=LEN(PZ$) <176>
12040 PRINT "{CLEAR}";PRINT "{RVSON SP}OSTLEITZ
AHLEN:{RVSOFF SPACE3}";PZ$;".":PRINT:PRINT <162>
12041 FOR X=1 TO EN <55>
12042 IF LEFT$(AD$(X),L)<>PZ$ THEN 12046 <104>
12043 IF X<=9 THEN 12045 <67>
12044 PRINT AN(X);"{SPACE4}";NA$(X):GOTO 12046 <74>
12045 PRINT "{SPACE}";AN(X);"{SPACE4}";NA$(X) <141>
12046 NEXT X <111>
12047 PRINT:PRINT "{RVSON SL}ISTE{SPACE}DRUCKEN{
SPACE3}--{SPACE SJ SPACE RVSOFF}" <140>
12048 PRINT "{RVSON SH}ENUE{SPACE11}--{SPACE SF}
1{RVSOFF}" <123>
12050 GET W$:IF W$="" THEN 12050 <117>
12051 IF W$="J" THEN 12055 <37>
12052 IF W$=CHR$(133) THEN 35 <25>
12053 GOTO 12050 <174>
12055 OPEN 1,4,7 <193>
12058 FOR X=1 TO EN <72>
12060 IF LEFT$(AD$(X),L)<>PZ$ THEN 12100 <173>
12080 PRINT#1,NA$(X);CHR$(141)TAB(30)"{SK}D-NR.:
{SPACE}";AN(X):PRINT#1 <79>
12082 PRINT#1,AD$(X);"{SPACE2}";TE$(X):PRINT#1 <91>
12084 PRINT#1,AD$(X);"{SPACE2}";WI$(X):PRINT#1:P
RINT#1 <77>
12100 NEXT X <165>
12110 CLOSE 1 <127>
12120 GOTO 35 <21>
12200 REM <102>
12210 PRINT:PRINT <185>
12215 PRINT "{RVSON SS}ORTIERUNG{SPACE}NACH{SPAC
E SN}AMEN:{RVSOFF}" <194>
12216 PRINT <128>
12220 INPUT "{RVSON SH}ELCHE{SPACE SA}NFANGSBUCH
STABEN{SPACE2 RVSOFF SPACE}";AB$ <247>
12222 PRINT <134>
12230 L=LEN(AB$) <163>
12243 PRINT "{CLEAR}";PRINT "{RVSON SH}AMEN{SPA
CE}MIT{SPACE SA}NFANGSBUCHSTABEN{RVSOFF SPACE3}"
;AB$:PRINT:PRINT <14>
12244 FOR X=1 TO EN <2>
12245 IF LEFT$(NA$(X),L)<>AB$ THEN 12249 <47>
12246 IF X<=9 THEN 12248 <65>
12247 PRINT AN(X);"{SPACE4}";NA$(X):GOTO 12249 <152>
12248 PRINT "{SPACE}";AN(X);"{SPACE4}";NA$(X) <88>
12249 NEXT X:PRINT:PRINT <4>
12250 PRINT "{RVSON SL}ISTE{SPACE}DRUCKEN{SPACE5
}--{SPACE SJ SPACE RVSOFF}" <182>
12251 PRINT "{RVSON SH}ENUE{SPACE13}--{SPACE SF}
1{RVSOFF}" <176>
12253 GET W$:IF W$="" THEN 12253 <140>
12254 IF W$="J" THEN 12258 <45>
12255 IF W$=CHR$(133) THEN 35 <228>
12256 GOTO 12253 <147>
12258 OPEN 1,4,7 <140>
12260 FOR X=1 TO EN <18>
12261 IF LEFT$(NA$(X),L)<>AB$ THEN 12300 <45>
12270 PRINT#1,NA$(X);CHR$(141)TAB(30)"{SK}D-NR.:
{SPACE}";AN(X):PRINT#1 <13>
12272 PRINT#1,AD$(X);"{SPACE2}";TE$(X):PRINT#1 <25>
12274 PRINT#1,AD$(X);"{SPACE2}";WI$(X):PRINT#1:P
RINT#1 <11>
12300 NEXT X <110>
12310 CLOSE 1 <72>
12350 GOTO 35 <252>
13000 PRINT "{CLEAR}"; <63>
13010 DIRECTORY <242>
13020 PRINT "{DOWN2 SPACES ST}ASTE{SPACE6}" <173>
13030 GET W$:IF W$="" THEN 13030 <59>
13040 GOTO 35 <176>
ENDE DES LISTINGS

```


»Easy Basic System« – eine neue Idee!!!

Easy-Basic ist keine Befehlsweiterung im üblichen Sinne, bei der die Programmierung von Sound und Grafik unterstützt wird. Vielmehr ist dieses Programm zu dem Zweck geschrieben worden, das Einprogrammieren von Basic-Programmen wesentlich zu vereinfachen. Die Idee, für ein Programm und deren Umsetzung bleibt freilich immer noch Ihre Aufgabe, jedoch nimmt Ihnen EASY BASIC einen unnützen Teil dieser Arbeit ab. Durch viele komfortable und nützliche Befehle wird die Zeit, die Sie benötigen um ein Programm zu schreiben um ein Wesentliches verkürzt. Viele Befehle sind auch im normalen Basic gar nicht möglich. EASY BASIC ist vollkommen absturzsicher.

Sollte sich das SYSTEM trotzdem einmal "aufhängen" oder stürzt IHR Programm ab, an dem Sie Stunden um Stunden geschrieben haben, so ist auch das klein Grund zum Verzweifeln, denn für diesen Fall hat EASY BASIC die RESTORE-Taste belegt. Ein Druck auf diese Taste genügt und schon wird der Bildschirm gelöscht und der Cursor meldet sich mit einem freundlichem READY. Bei ganz harten Fällen, also wenn auch die RESTORE-Taste versagt, hilft immer noch ein leichter Touch auf den RESET BUTTON (falls vorhanden), wobei sich der selbe Effekt einstellt wie bei RESTORE. Und noch etwas ist zu bemerken, EASY BASIC ist zu 100% in Maschinencode geschrieben. Das hat den Vorteil, daß der volle Basic-Speicherplatz erhalten bleibt und das EASY BASIC vor dem Überschreiben von Basic geschützt ist.

EASY BASIC in Stichworten:

- 25 neue Befehle mit insgesamt
- 32 Funktionen
- 100% Maschinencode
- kein Speicherplatzverlust
- keine Geschwindigkeitseinbußen
- einfache Bedienung
- komfortable Systemmeldungen
- Vollkommen absturzsicher, dadurch keine Gefahr, Programme zu verlieren
- liegt im geschützten Bereich (\$C000 - \$D000)
- alle normalen Befehle sind weiterhin nutzbar

- arbeitet sowohl mit Disk als auch mit Tape
- größtenteils in Deutsch gehalten, vor allem die SYSTEM-Meldungen

EASY BASIC Befehlsliste:

Jeder Befehl von EASY BASIC wird mit einem Pfeil nach links (Taste links oben) eingeleitet. Direkt dahinter wird der jeweilige Buchstabe und eventuell nötige Parameter eingegeben.

(= Taste Pfeil nach links setzen)

(←) 1 = ändern der Hintergrundfarbe

(←) 2 = ändern der Vordergrundfarbe

(←) A (Schrittweite) = Automatische Zeilennummerierung. Folgt auf das "A" keine Zahl für den Zeilenabstand, so wird dieser mit Zehn vorausgesetzt. Dieser Befehl numeriert ein Basicprogramm, das heißt, es wird am Anfang jeder Zeile automatisch eine Zeilennummer ausgegeben.

(←) B (E/A) = Anzeige des noch freien Speicherplatzes (nach jedem RETURN in der obersten Zeile).

Dieser Befehl benötigt auf jeden Fall ein "E" oder ein "A" als Parameter (A für Ausschalten, E für Einschalten).

(←) D = Ausgabe des DIRECTORYS der gerade im Laufwerk befindlichen Diskette. Natürlich ohne Programmverlust. (←) F (E/A) = Fehlerblockade ein- bzw. ausschalten. Dieser Befehl ist äußerst nützlich beim Ausräumen von Programmen, denn hierdurch wird der Computer dazu veranlaßt, bei einem Fehler im Programm automatisch den nächsten Befehl zu bearbeiten. Dadurch wird ein normalerweise nicht lauffähiges Basicprogramm doch noch lauffähig.

(←) G (E/A) = Grafik-Modus ein- bzw. ausschalten. Dieser Befehl ist dazu gedacht, das man schnell mal auf den Grafik-Modus umschalten kann, ohne durch lange POKE-Befehle Zeit zu verlieren. Er dient hauptsächlich als Kontrollfunktion, um den Inhalt des Grafik-Speichers zu überprüfen. Jedoch arbeitet er auch innerhalb eines Programmes, um beispielsweise die Titelgrafik eines Spieles einzuschalten. (←) H = Help. Dieser Befehl gibt dem Benutzer eine Auflistung sämtlicher

EASY BASIC Befehle, in übersichtlich geordneter Weise auf den Bildschirm aus.

EASY BASIC Befehlsliste Teil II:

(←) I = INITIAL DISK. Die gerade im Laufwerk befindliche Diskette wird initialisiert, so daß im Normalfall keine Disk Errors auftreten können (HEADER und BAM der Disk werden neu geladen).

(←) K = SPACEKILLER. Dieser Befehl ist sehr nützlich, wenn es darum geht, möglichst wenig Speicherplatz zu verbrauchen, denn er "küllt" alle in einem Programm überflüssigen Spaces (Lehrzeichen). Auf diese Weise können in vielen Programmen noch etliche BYTES eingespart werden, wenngleich dies auch nicht gerade zur Übersichtlichkeit des Programmes beiträgt.

(←) M = MEMORY DUMP. Mit Hilfe des MEMORY DUMP können Sie sich den Inhalt des gesamten Speichers, ausgegeben in ASCII Zeichen, ansehen. Nach Eingabe des Befehles wird der Bildschirm gelöscht und es erscheint in der obersten Zeile die Meldung "MEMORY DUMP: \$ 000". Das bedeutet, daß der MEMORY DUMP ab der Hexadezimal Adresse \$ 000 gestartet wird. \$ 000 deshalb, weil ja noch keine weitere Anweisung erfolgt ist. Sie haben jetzt die Möglichkeit den MEMORY DUMP entweder vorwärts oder rückwärts zu starten. Dies geschieht mit "F1" (rückwärts) und "F7" (vorwärts). Beim Druck auf eine dieser Tasten werden jeweils die nächsten 256 BYTES im ASCII-CODE ausgegeben. Während des ganzen MEMORY DUMPES haben Sie jederzeit die Möglichkeit diese Routine zu verlassen. Dazu genügt ein Druck auf die "Space-Taste" (das Titelbild erscheint). Sind die ersten 4096 BYTES ausgegeben erscheint wieder die Meldung "MEMORY DUMP: \$xxxx", jedoch steht jetzt anstelle der vier "x" die neue MEMORY DUMP Startadresse.

(←) O = OLD. OLD stellt ein mit neu gelöscht Programm wieder lauffähig und vor allem wieder abspeicherbar zur Ver-

fügung, so daß Sie quasi nie ein Programm verlieren können.

EASY BASIC Befehlsliste Teil III:

(←) Q = QUIT SYSTEM. Zu diesem Befehl ist eigentlich nicht viel zu sagen, nur soviel, daß diese Funktion die einzige Möglichkeit ist (abgesehen vom Ausschalten des Rechners) EASY BASIC zu verlassen. Ein eventuell vorhandenes Programm geht dabei verloren. Deshalb wird bei "QUIT SYSTEM" noch einmal nachgefragt, ob Sie sich auch ganz sicher sind. Die Frage des Computers ist mit einem "J" oder einem "N" gefolgt von RETURN zu beantworten.

(←) R (5/7) = RENAME KEYS. Mit RENAME KEYS können zwei der vier Funktionstasten neu belegt werden. Bei (←) R 5 kann "F5" neu belegt werden, bei (←) R 7 entsprechend "F7". Die neue Belegung darf maximal 5 Zeichen lang sein, was in abgekürzter Form auch für den längsten Befehl reicht. Die neue Belegung wird dann als Befehl im Direktmodus, bei einem Druck auf "F5" oder "F7" ausgeführt. Diese Tasten lassen sich sowohl mit normalen als auch mit EASY BASIC Befehlen belegen.

(←) S = DISK STATUS. Tritt beim Umgang mit der Floppy ein Error auf und Sie wissen nicht woran es liegen könnte oder Sie wollen einfach nur wissen, welches DOS Ihr Laufwerk besitzt, so genügt die Eingabe dieses Befehls und Sie erhalten die gewünschten Informationen in schön geordneter Weise auf dem Bildschirm serviert. Wollen Sie die Informationen über das Floppy-DOS abrufen, so müssen Sie dieses jedoch vorher kurz aus- und wieder einschalten.

(←) T (E/A) = Tastaturpieps ein bzw. ausschalten. Wie die großen Personal-Computer es meist serienmäßig eingebaut haben, so bietet auch EASY BASIC diese Routine, die bei jedem Druck auf eine der Tasten einen Piepton hervorruft, so daß Sie auch "blind" programmieren können. Die Routine kann wahlweise ("E") ein bzw. ("A") ausgeschaltet werden.

5,240,16,76,8,175,169,29,141	<59>	,32,240,5,145,169,32,61,197,32	<49>
32020 DATA24,208,169,56,141,17,208,76,228,167,16		32063 DATA68,197,76,243,196,160,0,132,7,162,5,17	<28>
9,21,141,24,208,169,27,141,17	<240>	7,167,145,169,240,16,132,7,32	
32021 DATA208,76,228,167,32,115,0,201,69,240,7,2	<84>	32064 DATA61,197,32,68,197,202,208,239,132,2,76,	<93>
01,65,240,37,76,8,175,169,213		243,196,230,7,165,7,201,3,240	
32022 DATA160,193,141,0,3,140,1,3,76,247,193,138	<155>	32065 DATA17,76,36,197,230,169,208,2,230,170,96,	<81>
,48,3,76,222,193,76,116,164,32		230,167,208,2,230,168,96,32,61	
32023 DATA6,169,136,32,115,0,208,250,76,225,167,	<208>	32066 DATA197,166,169,164,170,134,45,132,46,32,8	<2>
169,139,141,0,3,169,227,141,1		9,166,32,51,165,76,134,227,234	
32024 DATA3,76,37,194,162,0,189,7,194,32,210,255	<223>	32067 DATA162,0,165,0,189,114,197,32,210,255,232	<130>
,232,201,0,208,245,76,228,167		,201,0,208,245,76,139,197,13	
32025 DATA158,70,69,72,76,69,82,45,66,76,79,67,7	<60>	32068 DATA158,65,85,84,79,78,85,77,66,69,82,32,6	<112>
5,65,68,69,32,65,75,84,73,86		5,75,84,73,86,73,69,82,84,32	
32026 DATA73,69,82,84,32,33,13,0,162,0,189,53,19	<69>	32069 DATA33,0,32,115,0,208,4,162,10,208,3,32,15	<99>
4,32,210,255,232,201,0,208,245		8,183,142,197,197,169,167,141	
32027 DATA76,228,167,158,70,69,72,76,69,82,45,66	<105>	32070 DATA4,3,169,197,141,5,3,76,174,167,104,72,	<83>
,76,79,67,75,65,68,69,32,65,85		201,161,208,57,136,177,122,208	
32028 DATA83,71,69,83,67,72,65,76,84,69,84,32,33	<192>	32071 DATA13,169,124,141,4,3,169,165,141,5,3,76,	<123>
,13,0,32,0,192,76,228,167,169		116,164,165,20,166,21,24,105	
32029 DATA1,162,208,160,255,32,189,255,169,1,162	<140>	32072 DATA11,144,1,232,133,99,134,98,162,144,56,	<126>
,8,160,0,32,186,255,32,192,255		32,73,188,32,223,189,32,135,180	
32030 DATA176,2,169,0,133,255,169,1,32,195,255,1	<191>	32073 DATA32,130,183,132,198,177,34,153,119,2,13	<151>
73,255,0,201,5,240,3,96,162,0		6,16,248,76,124,165,162,0,32	
32031 DATA189,149,194,32,210,255,232,201,0,208,2	<240>	32074 DATA115,0,157,28,198,232,224,11,208,245,32	<211>
45,76,228,167,158,65,67,72,84		,68,229,173,33,208,141,134,2	
32032 DATA85,78,71,44,70,76,79,80,80,89,32,78,73	<74>	32075 DATA162,0,165,0,189,17,198,32,210,255,232,	<3>
,67,72,84,32,66,69,82,69,73,84		201,0,208,245,76,40,198,147,19	
32033 DATA32,33,13,0,32,179,238,162,0,189,227,19	<150>	32076 DATA83,89,83,53,48,55,55,56,44,32,32,32,32	<208>
4,32,210,255,232,201,0,208,245		,32,32,32,32,32,32,32,0,162,0	
32034 DATA169,8,133,186,32,180,255,169,111,133,1	<241>	32077 DATA165,0,189,17,198,201,171,240,3,232,208	<255>
85,32,150,255,32,165,255,32,210		,246,169,45,157,17,198,162,0	
32035 DATA255,201,13,208,246,32,171,255,76,228,1	<134>	32078 DATA165,0,189,17,198,32,210,255,232,201,0,	<72>
67,158,68,73,83,75,32,83,84,65		208,245,169,13,141,120,2,169	
32036 DATA84,85,83,32,58,32,0,32,93,194,162,0,18	<192>	32079 DATA19,141,119,2,169,2,141,198,0,234,32,25	<200>
9,41,195,32,210,255,232,201,0		3,174,32,121,0,144,6,240,4,201	
32037 DATA208,245,169,1,162,8,160,15,32,186,255,	<108>	32080 DATA171,208,23,32,107,169,32,19,166,165,95	<149>
169,0,32,189,255,32,192,255,162		,133,25,165,96,133,26,32,121	
32038 DATA1,32,201,255,169,73,32,210,255,32,204,	<140>	32081 DATA0,240,4,201,171,240,5,162,11,76,58,164	<114>
,255,169,1,32,195,255,76,228,167		,32,115,0,32,107,169,208,243	
32039 DATA158,68,73,83,75,32,87,73,82,68,32,73,7	<77>	32082 DATA165,20,5,21,208,8,169,255,133,20,133,2	<166>
8,73,84,73,65,76,73,83,73,69		1,208,6,230,20,208,2,230,21,32	
32040 DATA82,84,32,33,13,0,162,0,189,135,195,32,	<213>	32083 DATA19,166,165,95,133,36,165,96,133,37,56,	<196>
210,255,232,201,0,208,245,160		165,36,229,25,165,37,229,26,144	
32041 DATA3,200,177,43,208,251,200,200,152,160,0	<228>	32084 DATA201,165,45,229,36,133,95,165,46,229,37	<252>
,145,43,165,44,200,145,43,160		,133,96,24,165,25,101,95,133	
32042 DATA0,132,59,162,0,200,208,2,270,60,177,59	<167>	32085 DATA45,165,26,101,96,133,46,160,0,177,36,1	<242>
,208,245,232,224,3,208,242,200		45,25,230,25,208,2,230,26,230	
32043 DATA208,2,230,60,132,32,164,60,132,46,76,2	<162>	32086 DATA36,208,2,230,37,56,165,95,233,1,133,95	<241>
28,167,158,66,65,83,73,67,32		,165,96,233,0,133,96,16,225,32	
32044 DATA80,82,79,71,82,65,77,77,32,73,83,84,32	<70>	32087 DATA89,166,32,51,165,32,68,229,162,0,165,0	<62>
,82,69,71,69,78,69,82,73,69,82		,189,11,199,32,210,255,232,201	
32045 DATA84,32,33,13,0,1,8,160,0,162,0,169,0,23	<253>	32088 DATA0,208,245,76,145,227,19,13,13,158,66,6	<214>
4,234,234,165,0,189,197,195,32		5,83,73,67,32,90,69,73,76,69	
32046 DATA210,255,232,224,25,208,245,76,222,195,	<65>	32089 DATA78,32,71,69,76,79,69,83,67,72,84,32,33	<244>
147,19,17,158,77,69,77,9,82		,0,32,115,0,201,65,240,7,201	
32047 DATA89,32,68,85,77,80,58,32,32,36,32,48,48	<167>	32090 DATA69,240,71,76,8,175,169,131,141,2,3,169	<25>
,48,13,13,165,203,201,3,240,11		,164,141,3,3,162,0,165,0,189	
32048 DATA201,4,240,10,201,60,240,9,76,222,195,7	<105>	32091 DATA82,199,32,210,255,232,201,0,208,245,76	<153>
6,135,196,76,156,196,76,182,205		,228,167,13,158,83,80,69,73,67	
32049 DATA234,185,27,196,141,55,4,200,185,27,196	<19>	32092 DATA72,69,82,80,76,65,84,90,32,65,78,90,69	<120>
,141,70,196,200,192,34,208,5		,73,71,69,32,65,85,83,71,69,83	
32050 DATA160,0,208,234,234,162,0,32,179,238,232	<220>	32093 DATA67,72,65,76,84,69,84,32,33,0,162,0,165	<47>
,208,250,76,60,196,48,0,49,16		,0,189,140,199,32,210,255,232	
32051 DATA50,32,51,48,52,64,53,80,54,96,55,112,5	<57>	32094 DATA201,0,208,245,76,176,199,13,158,83,80,	<123>
6,128,57,144,1,160,2,176,3,192		69,73,67,72,69,82,80,76,65,84	
32052 DATA4,208,5,224,6,240,0,162,0,165,0,152,72	<142>	32095 DATA90,32,65,78,90,69,73,71,69,32,65,75,84	<71>
,160,15,189,0,33,32,210,255,232		,73,86,73,69,82,84,33,13,0,169	
32053 DATA224,255,208,245,238,70,196,136,208,22,	<202>	32096 DATA186,141,2,3,169,199,141,3,3,152,72,138	<220>
104,168,165,203,201,3,240,11		,72,166,214,134,252,169,230,160	
32054 DATA201,4,240,7,201,60,240,3,76,87,196,76,	<99>	32097 DATA199,32,30,171,165,51,229,49,170,165,52	<84>
181,195,165,203,201,3,240,11		,229,50,32,205,189,166,252,160	
32055 DATA201,4,240,7,201,60,240,11,76,107,196,1	<180>	32098 DATA0,132,199,24,32,240,255,104,168,104,17	<74>
69,0,141,145,2,76,68,196,76,182		0,76,131,164,19,18,159,69,65	
32056 DATA205,169,200,141,255,195,141,6,196,169,	<62>	32099 DATA83,89,45,66,65,83,73,67,32,83,89,83,84	<162>
34,141,8,196,169,0,141,12,196		,69,77,29,158,32,32,32,32,32	
32057 DATA76,249,195,169,136,141,255,195,141,6,1	<219>	32100 DATA159,29,66,65,83,73,67,32,66,89,84,69,8	<161>
96,169,2,141,8,196,169,34,141		3,32,70,82,69,69,19,29,29,29	
32058 DATA12,196,76,249,195,234,165,0,162,0,189,	<200>	32101 DATA29,29,29,29,29,29,29,29,29,29,29,29,29	<55>
196,196,32,210,255,232,201,0		,29,29,158,0,32,93,194,169,8	
32059 DATA208,245,76,224,196,13,158,83,80,65,67,	<22>	32102 DATA133,186,169,96,133,185,169,1,133,183,1	<137>
69,32,75,73,76,76,69,82,32,65		,69,36,133,253,169,253,133,187	
32060 DATA75,84,73,86,73,69,82,84,32,33,0,136,16	<26>	32103 DATA169,0,133,188,32,213,243,165,186,32,18	<55>
9,0,166,43,202,134,167,134,169		0,255,165,185,32,150,255,169	
32061 DATA166,44,134,168,134,170,160,0,132,2,177	<195>	32104 DATA0,133,144,160,3,132,253,32,165,255,133	<215>
,167,240,31,201,34,208,8,170		,254,164,144,208,47,32,165,255	
32062 DATA165,2,73,255,133,2,138,166,2,208,4,201		32105 DATA164,144,208,40,164,253,136,208,233,166	

programme

,254,32,205,189,169,32,32,210
 32106 DATA255,32,165,255,166,144,208,18,170,240,
 6,32,210,255,76,119,200,169,13
 32107 DATA32,210,255,160,2,208,198,32,171,255,76
 ,228,167,32,115,0,201,69,240
 32108 DATA7,201,65,240,47,76,8,175,169,15,141,24
 ,212,162,0,165,0,189,187,200
 32109 DATA32,210,255,232,201,0,208,245,76,228,16
 7,13,158,76,65,85,84,83,84,65
 32110 DATA69,82,75,69,32,58,32,49,53,32,33,0,169
 ,0,141,24,212,162,0,165,0,189
 32111 DATA231,200,32,210,255,232,201,0,208,245,7
 6,228,167,13,158,76,65,85,84
 32112 DATA83,84,65,69,82,75,69,32,58,32,48,48,32
 ,33,0,32,68,229,162,0,165,0,189
 32113 DATA29,192,32,210,255,232,201,0,208,245,16
 2,0,165,0,189,32,201,32,210,255
 32114 DATA32,201,0,208,245,76,20,202,19,17,17,1
 7,29,29,29,29,29,159,42,42,42
 32115 DATA42,32,5,66,69,70,69,72,76,83,32,45,32,
 85,69,66,69,82,83,73,67,72,84
 32116 DATA32,159,42,42,42,13,13,158,95,65,32,61,
 32,65,85,84,79,78,85,77,66,69
 32117 DATA82,13,95,66,32,61,32,83,80,69,73,67,72
 ,69,82,80,76,65,84,90,32,65,78
 32118 DATA90,69,73,71,69,32,40,69,47,65,41,13,95
 ,68,32,61,32,68,73,82,69,67,84
 32119 DATA79,82,89,13,95,70,32,61,32,70,69,72,76
 ,69,82,45,66,76,79,67,75,65,68
 32120 DATA69,32,40,69,47,65,41,13,95,71,32,61,32
 ,71,82,65,80,72,73,67,32,40,69
 32121 DATA47,65,41,13,95,72,32,61,32,72,69,76,80
 ,13,95,73,32,61,32,73,78,73,84
 32122 DATA73,65,76,32,68,73,83,75,13,95,75,32,61
 ,32,83,80,65,67,69,32,73,76
 32123 DATA76,69,82,13,95,77,32,61,32,77,69,77,79
 ,82,89,32,68,85,77,80,13,95,79
 32124 DATA32,61,32,79,76,68,13,95,81,32,61,32,81
 ,85,73,84,32,83,89,83,84,69,77
 32125 DATA0,0,162,0,165,0,189,38,202,32,210,255,
 232,201,0,208,245,76,205,205
 32126 DATA13,95,82,32,61,32,82,69,78,65,77,69,32
 ,75,69,89,83,13,95,83,32,61,32
 32127 DATA68,73,83,75,32,83,84,65,84,85,83,13,95
 ,84,32,61,32,84,65,83,84,65,84
 32128 DATA85,82,80,73,69,80,83,32,40,69,47,65,41
 ,13,95,86,32,61,32,76,65,85,84
 32129 DATA83,84,65,69,82,75,69,13,95,90,32,61,32
 ,90,69,73,76,69,78,68,69,76,69
 32130 DATA84,69,13,95,49,32,61,32,72,73,78,84,69
 ,82,71,82,85,78,68,32,70,65,82
 32131 DATA66,69,13,95,50,32,61,32,70,79,82,68,69
 ,82,71,82,85,78,68,32,70,65,82
 32132 DATA66,69,13,18,29,29,29,29,29,29,29,29,29
 ,29,29,87,69,73,84,69,82,32,77
 32133 DATA73,84,32,159,70,55,158,32,33,33,146,0,
 165,203,201,3,240,3,76,211,202
 32134 DATA32,68,229,162,0,165,0,189,241,202,32,2
 10,255,232,201,0,208,245,76,225
 32135 DATA205,95,78,32,61,32,78,69,85,83,84,65,8
 2,84,13,83,72,73,70,84,32,61
 32136 DATA32,76,73,83,84,32,83,84,79,80,13,70,49
 ,45,70,55,32,61,32,74,69,32,78
 32137 DATA65,67,72,32,66,69,76,69,71,85,78,71,13
 ,13,158,0,169,255,141,6,212,141
 32138 DATA24,212,169,9,141,5,212,169,144,141,1,2
 12,169,33,141,0,212,120,169,87
 32139 DATA162,203,141,20,3,142,21,3,160,192,132,
 56,88,76,40,207,72,165,203,201
 32140 DATA64,208,9,162,0,142,4,212,104,76,49,234
 ,160,17,140,4,212,165,203,201
 32141 DATA3,240,15,201,4,240,14,201,5,240,13,201
 ,6,240,12,76,99,203,76,21,204
 32142 DATA76,141,203,76,208,203,76,231,203,32,68
 ,229,169,12,162,15,160,1,141
 32143 DATA0,4,142,1,4,140,2,4,169,4,162,34,160,4
 4,141,3,4,142,4,4,142,21,4,140
 32144 DATA22,4,140,24,4,169,56,162,49,141,23,4,1
 42,25,4,169,1,141,154,0,169,0
 32145 DATA141,214,0,169,5,141,211,0,76,99,203,32
 ,68,229,169,19,162,1,160,22,141
 32146 DATA0,4,142,1,4,140,2,4,169,5,76,161,203,3
 2,68,229,169,12,162,9,160,19
 32147 DATA141,0,4,142,1,4,140,2,4,169,20,162,32,
 141,3,4,142,4,4,169,19,162,13
 32148 DATA160,2,141,119,2,142,120,2,140,198,0,76

<49> ,99,203,32,68,229,169,18,162
 <138> 32149 DATA21,160,14,141,0,4,142,1,4,140,2,4,169,
 32,162,32,141,3,4,142,4,4,76
 <76> 32150 DATA3,204,32,115,0,201,53,240,7,201,55,240
 ,8,76,8,175,160,181,76,73,204
 <89> 32151 DATA160,183,152,72,32,68,229,162,0,165,0,1
 89,101,204,32,210,255,232,201
 <37> 32152 DATA0,208,245,104,168,141,26,4,76,149,204,
 19,18,158,78,69,85,69,32,66,69
 <91> 32153 DATA76,69,71,85,78,71,32,70,85,69,82,32,84
 ,65,83,84,69,32,159,70,32,158
 <156> 32154 DATA32,40,53,32,90,69,73,67,72,69,78,41,13
 ,13,0,0,169,1,141,154,0,169,4
 <121> 32155 DATA141,214,0,169,0,141,211,0,169,177,141,
 8,3,169,204,141,9,3,76,228,167
 <179> 32156 DATA162,0,165,0,189,80,4,157,36,205,232,22
 4,5,208,245,173,26,4,201,181
 <16> 32157 DATA240,7,201,183,240,46,76,228,167,173,36
 ,205,141,235,203,173,37,205,141
 <53> 32158 DATA237,203,173,38,205,141,239,203,173,39,
 205,141,250,203,173,40,205,141
 <139> 32159 DATA252,203,169,240,141,8,3,169,192,141,9,
 3,76,228,167,173,36,205,141,25
 <105> 32160 DATA204,173,37,205,141,27,204,173,38,205,1
 41,29,204,173,39,205,141,40,204
 <152> 32161 DATA173,40,205,141,42,204,169,240,140,8,3,
 169,192,141,9,3,76,228,167,12
 <168> 32162 DATA9,19,20,32,234,32,115,0,201,69,240,7,2
 01,65,240,54,76,8,175,162,0,165
 <193> 32163 DATA0,189,79,205,32,210,255,232,201,0,208,
 245,169,17,141,104,203,76,228
 <166> 32164 DATA167,13,158,84,65,83,84,65,84,85,82,80,
 73,69,80,83,32,65,75,84,73,86
 <23> 32165 DATA73,69,82,84,32,33,0,162,0,165,0,189,13
 0,205,32,210,255,232,201,0,208
 <196> 32166 DATA245,169,0,141,104,203,76,228,167,13,15
 8,84,65,83,84,65,84,85,82,80
 <209> 32167 DATA73,69,80,83,32,65,85,83,71,69,83,67,72
 ,65,76,84,69,84,32,33,0,120,169
 <187> 32168 DATA49,162,234,141,20,3,142,21,3,169,0,141
 ,4,212,88,76,172,195,120,169
 <205> 32169 DATA87,162,203,141,20,3,142,21,3,169,17,14
 1,4,212,88,32,0,192,76,228,167
 <102> 32170 DATA120,169,49,162,234,141,20,3,142,21,3,1
 69,0,141,4,212,88,76,211,202
 <231> 32171 DATA162,0,32,179,238,232,208,250,120,169,8
 7,162,203,141,20,3,142,21,3,169
 <66> 32172 DATA17,141,4,212,88,76,228,167,234,201,81,
 240,13,201,67,240,12,201,80,240
 <16> 32173 DATA11,234,76,8,175,234,76,25,206,76,151,2
 06,76,228,167,234,162,0,165,0
 <188> 32174 DATA189,43,206,32,210,255,232,201,0,208,24
 5,76,65,206,13,158,81,85,73,84
 <114> 32175 DATA32,83,89,83,84,69,77,32,40,74,47,78,41
 ,32,63,0,169,78,162,206,141,8
 <83> 32176 DATA3,142,9,3,76,228,167,32,115,0,201,74,2
 40,23,201,78,240,3,76,25,206
 <6> 32177 DATA169,240,162,192,141,8,3,142,9,3,32,0,1
 92,76,228,167,169,228,162,167
 <46> 32178 DATA141,8,3,142,9,3,169,139,162,227,141,0,
 3,142,1,3,169,49,162,234,141
 <162> 32179 DATA20,3,142,21,3,169,0,141,4,212,169,255,
 141,5,128,76,226,252,32,115,0
 <144> 32180 DATA201,69,240,7,201,65,240,47,76,8,175,16
 2,0,165,0,189,183,206,32,210
 <169> 32181 DATA255,232,201,0,208,245,76,1,207,13,158,
 67,79,76,79,85,82,45,76,73,83
 <57> 32182 DATA84,32,65,75,84,73,86,73,69,82,84,32,33
 ,0,162,0,165,0,189,227,206,32
 <37> 32183 DATA210,255,232,201,0,208,245,76,20,207,13
 ,158,67,79,76,79,85,82,45,76
 <244> 32184 DATA73,83,84,32,65,85,83,71,69,83,67,72,65
 ,76,84,69,84,32,33,0,169,14,162
 <128> 32185 DATA207,141,101,203,142,102,203,76,228,167
 ,238,134,2,76,49,234,120,169
 <40> 32186 DATA49,162,234,141,101,203,142,102,203,88,
 169,7,141,134,2,76,228,167,162
 <183> 32187 DATA53,160,207,142,6,3,140,7,3,76,65,207,7
 2,173,141,2,201,1,240,249,104
 <196> 32188 DATA76,26,167,169,68,162,229,160,68,141,0,
 128,142,1,128,140,2,128,169,229
 <251> 32189 DATA162,195,160,194,141,3,128,142,4,128,14
 0,5,128,169,205,162,56,160,48
 <152> 32190 DATA141,6,128,142,7,128,140,8,128,32,93,19
 4,96,0

ENDE DES LISTINGS

Ski-Jumping

Ski-Jumping ist eine tolle Ski-flug Simulation (passend zur Jahreszeit), bei der die realitätsnahe Darstellung im Vordergrund stand. Daher wurden über 40 Sprites und zwei komplette Zeichensätze definiert.

Nach dem Start sehen Sie zunächst die Sprungschanze, auf der die Spielfigur herabsaut. Sie müssen nun den richtigen Abprungpunkt finden (mit dem Feuerknopf). Ist dies geglückt, so sehen Sie den unteren Teil der Anlage, auf der der Springer durch die Luft segelt.

Weiter oben ist noch eine Vergrößerung des Skispringers abgebildet, in der Sie die Haltung besser erkennen können.

Durch Joystickbewegungen nach links oder rechts können Sie den Springer nach hinten oder vorne legen. Die Mittelstellung ist allerdings optimal. Dem oft auftre-

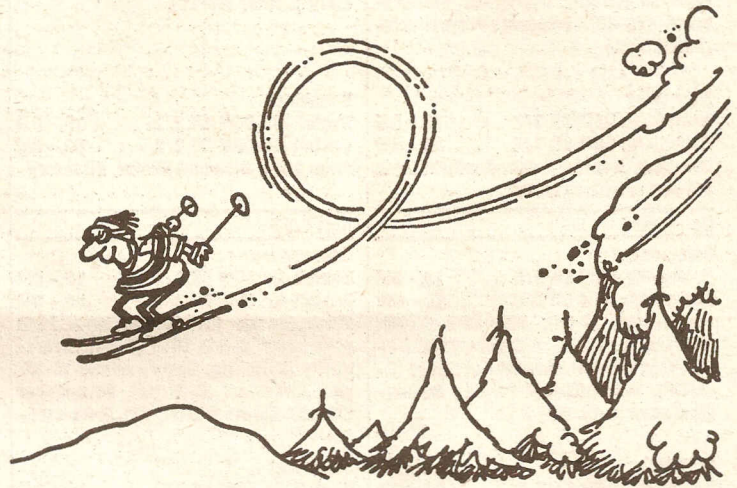
nen Querstellen der Skier können Sie durch Hochdrücken des Joysticks begegnen.

Kurz vor Erreichen des Bodens sollten Sie den Feuerknopf drücken, wodurch Sie den Springer veranlassen, zu Telemark-Landung anzusetzen.

Kurz danach erscheint die Auswertung, auf der Weite und Haltungspunkte aufgeführt werden.

Eingabetips:

Zuerst müssen die Teile "SKI-JUMPING" und "SKI-DATAS" abgetippt und getrennt gespeichert werden. Dann wird "SKI-DATAS" gestartet, wodurch der File "SKI MC" auf Diskette geschrieben wird. Von nun an muß man nur noch "SKI-JUMPING" laden und starten, wodurch der Rest automatisch von Diskette nachgeladen wird. Bei Verwendung einer Datensette müssen die Deviceadressen natürlich entsprechend geändert werden.



...und ab geht die Post!!!

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 IFA=OTHENA=1:LOAD"SKI<SPACE>MC",B,1:REM DATASE
TTE : ,1,1
1 GOTO90
10 *****
11 *
12 * SKI JUMPING *
13 * *
14 * (C) 1987 BY *
15 * *
16 * RADDISH-SOFT GERMANY *
17 * *
18 * WRITTEN BY *
19 * *
20 * CHRISTIAN RDUCH *
21 * *
22 * TEL. 02365/59008 *
23 * *
24 * *
25 *****
90 PRINT"<WHITE CLEAR>":POKE53280,0:POKE53281,0
91 PRINT"<DOWN3 RIGHT2>SKI<SPACE2>JUMPING"
92 PRINT"<DOWN3 RIGHT2>(C)<SPACE>1987<SPACE>BY<S
PACE>"
93 PRINT"<DOWN2 RIGHT2>RADDISH-SOFT<SPACE>GERMAN
Y"
94 PRINT"<DOWN4 RIGHT2>PLEASE<SPACE>WAIT<SPACE>5
.0<SPACE>MINUTES"
95 FORT=1TO2500:NEXT
100 A=0:DIMH(75):FORI=OTO9
101 H(48+I)=I:H(65+I)=I+10:NEXT
102 FORI=50176TO52736:READA$
103 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))

```

```

<30>
<140>
<130>
<15>
<40>
<17>
<50>
<19>
<138>
<21>
<165>
<23>
<87>
<25>
<137>
<27>
<28>
<145>
<200>
<159>
<57>
<96>
<63>
<234>
<162>
<116>
<163>
104 D=H(H)*16+H(L):S=S+D:POKEI,D <249>
105 A=A+1:IFA<20THENNEXT:A=-1 <68>
106 PRINT"ZEILE:";111+Z; <247>
107 READV:Z=Z+1:IFV=STHEN109 <251>
108 PRINT"PRUEFSUMMENFEHLER<SPACE>!";110+Z:STOP <12>
109 IFA<OTHEN1000 <60>
110 S=0:A=0:PRINT:NEXT:GOTO1000 <196>
111 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00, <202>
00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,
112 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00, <211>
00,3C,00,00,3C,00,00,120
113 DATA F0,00,00,F0,00,00,C0,00,00,F0,00,00,30, <197>
00,00,30,00,00,30,0C,1068
114 DATA FF,FF,F0,00,00,00,00,00,00,00,00,00,00,00,00, <154>
00,00,00,00,00,00,00,00,750
115 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00, <54>
00,00,00,00,00,00,30,00,48
116 DATA 00,FC,00,C0,FF,C0,30,33,00,0C,F0,00,03, <42>
C0,00,00,C0,00,00,30,1677
117 DATA 00,00,0C,00,00,03,C0,00,00,00,00,00,00,00, <57>
00,00,00,00,00,00,207
118 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00, <148>
00,00,00,30,00,C0,FC,492
119 DATA 00,30,FF,00,0C,33,C0,0C,F3,00,03,C0,00, <154>
00,C0,00,00,C0,00,00,1392
120 DATA 30,00,00,0C,00,00,0C,00,00,03,C0,00,00, <140>
00,00,00,00,00,00,267
121 DATA 00,00,00,00,00,00,00,00,C0,00,00,C0,00,00, <88>
C3,C0,00,33,F0,00,33,1113
122 DATA 30,00,3F,3C,00,0F,0C,00,0C,00,00,0C,00, <121>
00,03,00,00,03,00,00,228
123 DATA 03,00,00,00,C0,00,00,C0,00,00,C0,00,00, <251>
30,00,00,00,00,00,00,627

```


Zeitschriften und Programme von 1986/87

Heft 1/86

Commodore
 Bestell-Nr. COM CK 1/2 16,- DM
 Bestell-Nr. COM CD 1/2 20,- DM
 Checksummer 1.0 C, Astronaut Battle (C-64), Nueycobra (C-16/116), Hardcopy-routine (C-16/116), Weimut und die Zeitmaschine (C-16/116), Lecas (VC-20), Rock'me (VC-20), Charly der Raketenmann (VC-20), Interrupt-Programmierung (VC-20)

Schneider

Bestell-Nr. COM SK 1/2 16,- DM
 Bestell-Nr. COM SD 1/2 39,- DM
 Defender, Boulder Dash, Basic-Befehle auf den Funktionstasten

Heft 5/86

Commodore
 Bestell-Nr. COM CK 5/2 16,- DM
 Bestell-Nr. COM CD 5/2 20,- DM
 Space-Devil (C-64), Disk-Eddi (C-64), ABC-Logen (V-20), 3-D-Labyrinth (VC-20), Tron (VC-20), Invaders of Space (C-16/116), Submarine (C-16/116), Balkendiagramm (C-16/116)

Schneider

Bestell-Nr. COM SK 5/2 16,- DM
 Bestell-Nr. COM SD 5/2 39,- DM
 Startank, Grafik-Designer, Cross-Reference, Itext

Heft 9/86

Commodore
 Bestell-Nr. COM CK 9/2 16,- DM
 Bestell-Nr. COM CD 9/2 20,- DM
 Moonpatrol (VC20), Falcons Lander (VC20), Frogger (VC20), The Games (C64), Cave of Death (C64) Sprites and Sound mit dem UMH-System (C64), Diggy (C16/116), Tapedirectory (C16/116), Reversi (C16/116)

Schneider

Bestell-Nr. COM SK 9/2 16,- DM
 Bestell-Nr. COM SD 9/2 39,- DM
 Creepy, Briefeschreiben, Frutty Man, Mirror Battle

Heft 1/87

Commodore
 Bestell-Nr. COM CK 1/3 16,- DM
 Bestell-Nr. COM CD 1/3 20,- DM
 Torch (VC20), Bomberman (VC20), Note Screen (C16/116), Champ (C16/116), Checksummer (C16/116), Checksummer (C64), Pyramid (C16/116), Turbo-Disc (C64), IRQ-Routine (C64), Rylan (C64), Ice Racing (C64)

Schneider

Bestell-Nr. COM SK 1/3 16,- DM
 Bestell-Nr. COM SD 1/3 39,- DM
 Electric Gladiators, Bildschirmtoolbox, Suprabasic, Spritegenerator, Madgoon's Race, Star Trek, Checksummer

Heft 2/86

Commodore
 Bestell-Nr. COM CK 2/2 16,- DM
 Bestell-Nr. COM CD 2/2 20,- DM
 Robo (C-64), Basic-Befehlsweiterung (C-64), Raketenstart (C-16/116), Dreikampf (C-16/116), Headline (C-16/116), Defender (VC-20), Formel 1 (VC-20), Memory-Display (VC-20)

Schneider

Bestell-Nr. COM SK 2/2 16,- DM
 Bestell-Nr. COM SD 2/2 39,- DM
 Alien 2001, Diamond Hunter, Directory

Heft 6/86

Commodore
 Bestell-Nr. COM CK 6/2 16,- DM
 Bestell-Nr. COM CD 6/2 20,- DM
 Music-Master (C-64), The little Duck goes home (C-64), EDDY (C-64), Invers-Utility (C-16/116), Checksummer (C-16/116), Affenwurf (C-16/116), Data-Maker (VC-20), Karate-Kid (VC-20), Q-Bert (VC-20)

Schneider

Bestell-Nr. COM SK 6/2 16,- DM
 Bestell-Nr. COM SD 6/2 39,- DM
 Disksort, Mini-Forth, Starcommand

Heft 10/86

Commodore
 Bestell-Nr. COM CK 10/2 16,- DM
 Bestell-Nr. COM CD 10/2 20,- DM
 Laufschrift (VC20), Checksummer (VC20), Treasure Robber (VC20), Scrumble (VC20), Refusal-Basiclader (C-16), Hardy-Copy-Blitz (C-16), Autostart (C-16), Fill-Utility (C-16), Railway (C-16), Domino (C-16), Sprite Turn/Character-Switch (C-64), Checksummer (C-64), Horse Games (C-64), Patrouille (C-64)

Schneider

Bestell-Nr. COM SK 10/2 16,- DM
 Bestell-Nr. COM SD 10/2 39,- DM
 Checksummer, Panic-Car, Scurfy Joe, Drawgenerator, Airfighter

Aus diesem Heft:

Commodore

Bestell-Nr. COM CK 2/3 16,- DM
 Bestell-Nr. COM CD 2/3 20,- DM

Schneider

Bestell-Nr. COM SK 2/3 16,- DM
 Bestell-Nr. COM SD 2/3 39,- DM

Heft 3/86

Commodore
 Bestell-Nr. COM CK 3/2 16,- DM
 Bestell-Nr. COM CD 3/2 20,- DM
 Minor-Jumper (C-16/116), Hero (C-16/116), Data-Maker (C-16/116), Befehlsweiterung (C-64), Starchip (C-64), Mini-Car-Race (C-64), Text-Scroller (C-64), Text-Editor (C-64), Checksummer (VC-20), Robin at Ravenwreck (VC-20), Mini-Karate (VC-20), Autonumber-Routine (VC-20)

Schneider

Bestell-Nr. COM SK 3/2 16,- DM
 Bestell-Nr. COM SD 3/2 39,- DM
 Smily, Screpi, Die verrückte Fabrik, Disc-Help

Heft 7/86

Commodore
 Bestell-Nr. COM CK 7/2 16,- DM
 Bestell-Nr. COM CD 7/2 20,- DM
 VC-20-Sound-Demo, Mad Rush (VC-20), Ghost-Neil (VC-20), Roadrunner (C-64), Basic Extension (C-64), Meteoritensturm (C-64), Checksummer C-16/116, Airwolf (C-16/116), Break-Out (C-16/116)

Schneider

Bestell-Nr. COM SK 7/2 16,- DM
 Bestell-Nr. COM SD 7/2 39,- DM
 Poker, Cave-Walker, Diskettenarchiv

Heft 11/86

Commodore
 Bestell-Nr. COM CK 11/2 16,- DM
 Bestell-Nr. COM CD 11/2 20,- DM
 Cooky (VC20), Gambol (VC20), Wostok (VC20), Topball (C16), Zauberpuzzle (C16), Schieberoutine (C16), Atlantis (C64), Astronomie (C64), Deutsche Fehlermeldung (C64), Remzeilenkiller (C64)

Schneider

Bestell-Nr. COM SK 11/2 16,- DM
 Bestell-Nr. COM SD 11/2 39,- DM
 Diskettenhilfe, Horror Driver, Disc of Tron, Space Hawk, Morsedecoder

Heft 4/86

Commodore
 Bestell-Nr. COM CK 4/2 16,- DM
 Bestell-Nr. COM CD 4/2 20,- DM
 Shuttle (VC-20), Josef in den Höhlen (VC-20), Merge-Routine (VC-20), Shoot it (VC-20), Fast Trust (C-64), Data-Text 64 (C-64), Gary, der schnelle Kabelverleger (C-16/116), SDI (C-16/116), Beep-Routine (C-16/116)

Schneider

Bestell-Nr. COM SK 4/2 16,- DM
 Bestell-Nr. COM SD 4/2 39,- DM
 Space-Rescue, Texteditor, Starfight, Programm-Retter

Heft 8/86

Commodore
 Bestell-Nr. COM CK 8/2 16,- DM
 Bestell-Nr. COM CD 8/2 20,- DM
 Death Desert (C-64), 3-D-Labyrinth (C-64), Klick (C-64), Bahnhof (VC-20), Omega Attack (VC-20), Sound (VC-20), Platinenlayout (C-16/116), Balloon Shoot (16/116), Hardcopy (C-16/116)

Schneider

Bestell-Nr. COM SK 8/2 16,- DM
 Bestell-Nr. COM SD 8/2 39,- DM

Heft 12/86

Commodore
 Bestell-Nr. COM CK 12/2 16,- DM
 Bestell-Nr. COM CD 12/2 20,- DM
 Ballon Hopper (VC20), Fight Night (VC20), King of Kings (C16/116), Galaxis (C16/116), Sprites (C16/116), Atom (C64), Extended-UMH (C64), IRQ-Directory (C64)

Schneider

Bestell-Nr. COM SK 12/2 16,- DM
 Bestell-Nr. COM SD 12/2 39,- DM
 Alien Attack, Lucky Luck, Mouse, Checksummer, Tracer

Superpakete!

Beachten Sie bitte unsere neu geschnürten Programmpakete. ... und das zu einem Superpreis!

Alle hier aufgeführten Kassetten und Disketten können jederzeit nachbestellt werden.

Bei den Zeitschriften sind die Ausgaben 5, 12 (85) und 5, 6, 7, 8, 9, 10, 11, 12 (86), 1 (87) noch vorhanden.

Neu: Superpakete zum Superpreis!

C=64

Superpaket 1 Spiele

Monster Attack, Projekt, High Noon, Brieftaube, Cadilon, Turbo Jet, Sea Attack, The Maze, Graze Game, Galaxi, Space Comets, Sheet, Vampir, Memory

SP-C-K-1 Kassette 20,-
SP-C-D-1 Diskette 20,-

Superpaket 2 Spiele

Tank Duell, Astronaut, Jungle 1, Jungle 2, Sufo, Fire-Legion, Spiders, Star Battle, Super Breakout, Mission Rescue, Roadpainter, Moonsweeper, Scotti

SP-C-K-2 Kassette 20,-
SP-C-D-2 Diskette 20,-

Superpaket 3 Anwender

The Basic, Basicreset, Data-Generator, Print-Generator, Supermon-64, Restore N, Disky 10, Fill-Loader, Funktionstasten-Belegung, Autostart, Multibasic, Bildeditor, Spritehilfe + Demo, Autonumber, Micro-Tools + Mloader

SP-C-K-3 Kassette 20,-
SP-C-D-3 Diskette 20,-

Programm	Preis Kass./Disk.	Bestell-Nr.
Tron/Mercurious	24,50/29,50	C 13
Interceptor		
Base/Schotter	24,50/29,50	C 23
Jump Man/		
The Way/Space Taxi	24,50/29,50	C 33
Fight Night/Monster-		
jagd/Colossus Mühle	24,50/29,50	C 43
Mr. Postman/Moon-		
tunnel/Moonralies/		
Steinschlag	24,50/29,50	C 53
Stone Age comic		
Deadly Mission	24,50/29,50	C 63
Taxi Driver/		
Space Odyssey		C 14

C-16

Superpaket 1

Horror Castle, Fantasy County, Senso, Jumping Joe, Schatztaucher, Shaolin, City Runner

SP-O-K-1 Kassette 20,-
SP-O-D-1 Diskette 20,-

Superpaket 2 Genesis-Super-Assembler für C-16/116&plus 4

Komplett mit Anleitung und Demo-Programm

SP-O-K-2 Kassette 15,-
SP-O-D-2 Diskette 15,-

Programm	Preis Kass./Disk.	Bestell-Nr.
Goldrausch/		
Moon Fighter	18,- /23,-	O 33
Frogger/Humor	18,- /23,-	O 43
Cosmic Terror/		
Schwimmen/Break In	18,- /23,-	O 53
Panic/Phantom/		
Meteorstorm	18,- /23,-	O 63
Flower/Hyperspace	18,- /23,-	O 14

VIC-20

Superpaket 1

Flowers, Tanker, Jupiter Rescue, Einbrecher, Hospital, Alien, Flipperball, Sub-Zone, Dr. Cerp, Geo-Zayxon, Froggies Night Mare, Punktefresser, Starwars, Salavage Crew

SP-V-1 nur Kassette 20,-

Programm	Preis Kass.	Bestell-Nr.
Brückenbau/Jango	16,-	V 13
Miner, der Fassaden-		
streicher/Inkaschatz	16,-	V 23
Bergshooting/		
Bobby In Action	16,-	V 33
Josef in den		
Katakomben	8,50	V 43
Hot Food/		
Men-Rescue/		
Soundprogramm	14,50	V 53
Castle Dracula/		
UFO-Destroyer	14,50	V 63
Care Rider/		
Cosmic Bonncer	14,50	V 14

Schneider

Superpaket 1 Spiele

Alien Hunter, Catch, Interceptor 3 D, City Lander, Conan's Castle, Copter Patrol, CPC Bert, Crash, Demon Attack, Duell, Etis, Labyrinth, Miner, Tower, Snider's Maze, Geister Schloß, Fallschirmspringer, Secret Valley, Aladin, Hounded Castle, Cave runner, Cobra, Erwin, Kruehi, Q-Bert

SP-SR-K-1 Kassette 20,-
SP-SR-D-1 Diskette 27,-

Superpaket 2 Anwender

Back up, Data gen, Datenverwaltung, Discopy, Monitor, Painter, Remkiller, SU-File, Tastaturbelegung

SP-SR-K-2 Kassette 20,-
SP-SR-D-2 Diskette 27,-

Programm	Preis Kass./Disk.	Bestell-Nr.
Schneider Panik/		
Killer Ship	18,50/28,50	SR 13
Midnight/Horror Caves	18,50/28,50	SR 23
Spider Maze	18,50/28,50	SR 33
Triton/Traumland	18,50/28,50	SR 43
Starpatrol/		
Helicopter Pilot	18,50/28,50	SR 53
Red Alert/		
Speedy-Frogs/Aurlon	18,50/28,50	SR 63
Space Battle	18,50/28,50	SR 14

software service

Der Tronic-Verlag bietet Spitzensoftware für jedermann.

Überzeugen Sie sich von unseren Angeboten und bestellen Sie noch heute. Alle Bestellungen werden innerhalb nur

1 Woche

nach Eingang bearbeitet. Telefonische Bestellungen unter der Rufnummer

056 51/300 11

Anleitungen zu unseren Programmen erhalten Sie aus unseren Heften. Bei ausverkauften Heften erhalten Sie zum Heftpreis Fotokopien der Spielbeschreibungen.


```

211 DATA 00,00,00,00,00,00,00,00,00,00,02,00,00,
02,00,00,02,00,00,02, 8
212 DATA 00,00,00,02,00,00,02,00,00,00,00,00,00,
00,02,00,00,02,00,00, 8
213 DATA 03,00,00,03,00,FF,FF,00,00,00,00,00,00,
00,00,00,00,F0,00,00, 756
214 DATA D0,00,00,50,00,00,B0,00,00,A0,00,00,A0,
00,00,A0,00,00,A0,00, 1056
215 DATA 00,60,00,00,60,00,60,00,80,00,00,80,00,
80,00,00,80,00,00,80, 832
216 DATA 00,00,00,00,00,C0,00,C0,FF,FF,00,00,00,
00,00,00,00,00,00, 894
217 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00, 0
218 DATA 00,00,00,00,00,00,00,00,00,00,00,00,C0,
00,00,30,00,00,0C,20, 284
219 DATA 00,03,EA,00,00,CA,00,00,30,00,00,0C,00,
00,03,00,00,10,00,00, 518
220 DATA 80,00,00,80,00,02,80,00,02,30,00,02,9C,
00,02,9E,40,0A,A0,00, 988
221 DATA 2A,80,00,AB,00,00,A0,00,00,AB,00,00,2A,
00,00,02,00,00,CE,00, 916
222 DATA 00,3C,00,00,0C,00,00,03,00,00,00,C0,00,
00,30,00,00,0F,00,00, 330
223 DATA 00,00,02,00,00,02,00,00,02,00,00,09,00,
00,09,00,00,0A,00,00, 34
224 DATA 01,00,00,02,00,00,02,00,00,02,00,00,00,
00,00,00,00,00,00, 7
225 DATA 00,00,00,00,00,00,00,00,03,00,00,0C,00,00,
30,00,00,C0,00,03,00, 258
226 DATA 00,0C,00,00,00,00,00,00,00,00,00,00,3C,00,00,
34,00,00,94,00,00,A0, 432
227 DATA 00,00,A0,00,00,A0,00,00,80,00,00,80,00,
00,00,00,30,00,00,30, 672
228 DATA 80,00,C0,80,03,00,80,0C,00,80,30,00,80,
C0,00,83,00,00,FC,00, 1470
229 DATA 00,F0,00,00,C0,00,00,00,00,00,02,00,00,
02,00,00,01,00,00,09, 446
230 DATA 00,00,0A,00,00,0A,00,00,00,00,02,00,00,
00,02,00,00,02,00,00, 36
231 DATA 00,00,00,00,00,00,00,00,00,00,00,00,0F,
00,00,F3,00,03,0C,00, 273
232 DATA 3C,30,00,C0,C0,00,03,00,00,0C,00,00,00,
00,00,00,00,00,3C,00, 567
233 DATA 00,34,00,00,94,00,00,A0,00,00,A0,00,00,
A0,00,00,80,00,00,80, 936
234 DATA 00,00,00,00,30,00,00,30,80,00,C0,80,03,
0C,80,0C,0C,80,30,F0, 1127
235 DATA 80,CF,00,83,F0,00,FC,00,00,F0,00,00,C0,
00,00,00,FF,FF,FF,FF, 2410
236 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF, 5100
237 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF, 5100
238 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF, 4845
239 DATA 00, 0
1000 PRINT:AD=57344:FORI=1005TO1071:READA$,PS:C=
0:FORN=1TOLEN(A$)STEP2
1001 A=ASC(MID$(A$,N,1))-48:B=ASC(MID$(A$,N+1,1)
)-48:A=A+7*(A>9):B=B+7*(B>9)
1002 POKEAD,16*A+B:C=C+16*A+B:AD=AD+1
1003 NEXT:PRINTI;"{SPACE}";:IFC<>PSTHENPRINT"ERR
OR":STOP
1004 PRINT"OK":NEXT
1005 DATA FFFFFFFFFEFAFAFFFFFFFFFAAAAAAFFFFFFFF
FFFFFFFFFAFAFAEAEAEABABABAB, 6919
1006 DATA AB800000000000000000000000000000ABAB
AB2A2A2A2A2AABABABABABAB, 2029
1007 DATA ABABABABABABABABABAA2A2A2A2A2A2A2A2A
2A2A2A2A2A2AAAAA, 3332
1008 DATA EAEAEAAA000000000000000000000000
AA0200000000002AAAAA, 2152
1009 DATA AAABABAFBFAFAFEFFFFFFFFFAAAAAAFFFFFF
FFAFBFFFFFFFFFAA0000, 6398
1010 DATA 0000000000AA00020A020200AA002A022A20
2A00AA002A020A022A00AA00, 964
1011 DATA 2222A020200AA002A020A022A00AA002A020B
0B2000AA002A222A2A00AA, 1240
1012 DATA 002A222A022A00AA002A222222A0055555452
4A2AAF5F55551500B0ABBEFB, 2204
1013 DATA 5555551500BF8FBF5555551514EA2FEF524B4F
2EBE2BAEEFC1COE0B0BBFEAA, 3842
1014 DATA FF551500000000ABBE15150505525A982FBFB
FFAAFFFAFA7AABBEBAABBE, 4018
1015 DATA BEBEB300B0BAFFEEEEEFAE120ABEEEEEE2EEE

```

```

<146> AEFFBF689266E6BCF2CA2BAF, 5520 <206>
1016 DATA AAFEFBFBFB2A0BBAABBB0B0B000AAFFFEFF
4496ACFAFFAFFFAFB08BBDAA, 5510 <57>
<27> 1017 DATA AAB0A2A2B2FFCCAAAA0B02BE3ABEDAAAAF2C
BC13B6D3AAAAFF3F30EE4C32, 4628 <204>
<176> 1018 DATA F7AAAAF3C0BEB793BFAABFB0B3BB3B51B7AAFF
3030EFAFAA00C0000000A2AA, 4540 <252>
<126> 1019 DATA AA000000000000BAAAA0000000002CAAAA0000
000000F0F3AAAA00000000F3, 1968 <190>
<9> 1020 DATA F3AAAA00000000B0B3BFAA00000000FF30FFFA
0000000055555555555545450, 2700 <1>
<194> 1021 DATA 5555555555551505B0BCB0B0B0B0B0300C03
000000000000B030C000C00, 1822 <106>
<52> 1022 DATA 00000000C030000000003000003000003000C0
B0BCB0B0B0B0B0B0300C0300, 1819 <121>
<137> 1023 DATA 00002F0B020000C00000BFEEFBF2F0B0202FF
3FACAFABCFBFFFFFFFBFFB3, 3799 <118>
<249> 1024 DATA FFAFFFEFFFEFFFEFFFEFFFEFFFEFFFEFFFEFF
FFFF3F2FFFFBFFFD5F5F5F5, 6974 <96>
<112> 1025 DATA F5ADBDBF5555555696F6F6A5555555555A5E4
E2000000300000000BF2FOA, 2911 <245>
<184> 1026 DATA 0200000000FFFFFFFBF2B0200EFFFFAFBFEF
AF2BFFFF3FFF3FFFFFFFBF, 4916 <84>
<39> 1027 DATA BCFBFBFBFBFBFBFBFBFBFBFBFBFBFBFBFBFB
C5B5BD656565656565656565, 5643 <105>
<155> 1028 DATA 2A6B6E6F6F3E55FFAFA0A000000000BFBFFFAF
0A000000EFFFFFFFBF2A0000, 3183 <101>
<110> 1029 DATA FFBFB3BFFFAF0B0055D1C5F5F5D5A50A555555
5555555A96C636A6F6E6E6E, 4144 <209>
<90> 1030 DATA 6E84FFAFAFFBEEBEEF02000000000C000AA00
000000000000A50A00000000, 2319 <55>
<35> 1031 DATA 00006FAA000000000000FEAFA0A0B0B0B0B95
252509090202005555555555, 1439 <127>
<147> 1032 DATA 5555950000C000000000555555555555795
5555595F777FFF7F5F3FFFF, 2727 <99>
<100> 1033 DATA FFFFFFFE553CFD35FFBF3EEF55550555F0FFF
FC5555577FFF3FFFFE55555, 4945 <251>
<225> 1034 DATA 55D5F5EFFF9525250B0B0200007F9595F2BFFF
BFBFBFBFBFBFBFBFBFBFB, 4609 <186>
<233> 1035 DATA 3FFFC3FF3FFFFFFFBFBFBFBFBFBFBFBFBFB
FEFFF3FFFF3FBFBFBFBFB, 7031 <80>
<216> 1036 DATA D5FDFF3F3FCFCF555555555D5F5D50000C00
030000002B2F0B0202000000, 2477 <187>
<8> 1037 DATA E3FEBFBFBFBFBFBFBFBFBFBFBFBFBFBFBFBFB
FFFCEEF2BFBFBFBFBFBFB, 6788 <88>
<40> 1038 DATA FFFFEF3FEFFFBFBFBFBFBFBFBFBFBFBFBFB
F5F5D7FDCDF555555411555, 6469 <70>
<110> 1039 DATA D5D5FFFFFFF5F5F5F5F5F5F5F5F5F5F5F5F5
00110040410544FFFCFC05, 5744 <211>
<8> 1040 DATA C515544415105515551055415544554415555
FFFFFFFFFFFFFFFFDFFFFFFF, 4363 <48>
<76> 1041 DATA FDF55555FC1C115055555554555055555555
55555555555555553F4F5F, 3222 <98>
<77> 1042 DATA 4755545551FFFFFFF7F5F5FFFFFFF7F5F5
FFF5FFFFFFFDD54555FFD, 6176 <56>
<212> 1043 DATA F5555551555D55155545555555445555554
555555FFFF4F53455554FF, 3257 <25>
<2> 1044 DATA FFFFFFFF4F5F57FFFFFFF5F50FFFFFFF
FFFFFFF55555555555555, 6604 <19>
<139> 1045 DATA FFFF4D555551555D554555555555555555455
4555555557F5F515555555, 3292 <10>
<20> 1046 DATA 5FFFFF7F57455554550550551555555505
4154055555557F7F4F5F, 3307 <142>
<135> 1047 DATA 5755FFFFFFFDFDF1F1DFD1D55415554551FF
FFFCFCFCFC0F0C3F0F0F03, 5276 <214>
<121> 1048 DATA 03400454555555559525FF7F7F4F5F505454
FFFFFFFFFFFF3F0FFFFFFF, 4325 <33>
<138> 1049 DATA FFFFFFFCFFFCFC1C5C405415555455155554
5555555554555555FFFF, 4466 <120>
<221> 1050 DATA FFFCFCFC1C0C5C010045555415544055551511
5055FF3F3F4F531354540902, 3167 <14>
<175> 1051 DATA 0000C000000555A509020000034F50505455
952525DF1F5101505515415, 1959 <192>
<91> 1052 DATA 1555555554555555455555455555555555
54555555FFFFDFCF1D55515, 3367 <77>
<224> 1053 DATA C515551555555511505545555555455515151
5551555555411555155555, 2380 <178>
<74> 1054 DATA 553F3F5F43535354540902020000000035555
55952525090255445455555, 1688 <115>
<138> 1055 DATA 55551555555505544555555555555555155155
555555551555155455555515, 2213 <92>
<221> 1056 DATA 55555455555555551555555555551555555
7F3F0F5F53535555952525, 2538 <202>
<215> 1057 DATA 0902C2005454555555555555555555555555
F0FFFFFF3F0F0303FFFF, 4517 <101>
1058 DATA FFFFFFFF5555555A50A0000555555555A5

```


1119 DATA 034C9078EEFBC34C9078203C7FA90085FBA9C0 85FCAD00D0E9184A4A4AB5FD, 4128	<66>	1152 DATA EEB17F60EEB37FEEB37F60EAA92B8DF9C3A92C 8DFAC3A94EBD03D0A94ABD05, 4640	<21>
1120 DATA 18A5FB65FDB5FBAD01D0E9324A4A4AB5FEA6FE 18A5FB692885FBA5FC690085, 4406	<26>	1153 DATA 0A9D58D02D0A9ED8D04D060EAA92D8DF9C3A9 2E8DFAC3A94EBD03D0A94ABD, 4615	<253>
1121 DATA FCCAD0F018A5FB692885FBA5FC690085FCA000 B1FBC9EAF0034C9078A9018D, 4695	<144>	1154 DATA 05D0A9D58D02D0A9ED8D04D0ADB17FC9FAB004 EER17F60EEB37F60A9158D18, 4446	<56>
1122 DATA C47F4C907878A9308501A2F9BD00E0A8BDD06A 9D00E0989DD06ABDFAE0A8BD, 4653	<140>	1155 DATA 0DAD00DD09038D00DDA9048D8802AD34C1E9C4 8DCC7FAD32C1E9C48DCB7FAD, 4236	<119>
1123 DATA CA6B9DFAE0989DCA6BDDF4E1ABDDC46C9DF4E1 989DC46CBDEEE2A8BDBE6D9D, 5582	<18>	1156 DATA 31C1E9C48DCB7FADCA7F8DE27FADCB7F8DE37F ADCC7F8DE57F18ADCD7F6930, 4818	<187>
1124 DATA EEE2989DBE6DBDEBE3ABDBB6E9DEBE3989DBB 6ERDE2E4A8BDB26F9DE2E498, 5642	<120>	1157 DATA 8D2080ADCE7F6930BD2180ADCF7F6930BD2380 18ADCC7F6DC7F8D5280ADCB, 3924	<207>
1125 DATA 9DB26FBDCE5A8BDAC709DDCE5989DAC70BDD6 E6ABDDA6719DD6E6989DA671, 5382	<13>	1158 DATA 7F6DCE7F8D5080ADCA7F6DCD7F8D4F80A0FAA9 2099FF0399F90499F30599ED, 4428	<9>
1126 DATA CAE0FFD08BA93785015860ADC47FC900F00CC9 01F052C902F001604C2D7B20, 3859	<13>	1159 DATA 0699FFD799F9DB99F3D999EDDAB8D0E3A90E8D 21D0A9008D15D0AD5280C93A, 4785	<47>
1127 DATA 0C7CABD17FC9FAB01DEEB17FEEB17F18ADB07F 6DB17F8DB07FB004EE00D060, 4432	<244>	1160 DATA 9008E0A8D5280E5080AD5080C93A9008E90A 8D5080EE4F80AD3180C93A90, 3774	<208>
1128 DATA EEO1D0EE00D060EEB37FEEB37F18ADB07F6DB3 7F8DB07FB007EE00D0EE01D0, 4512	<32>	1161 DATA 0AE90ABD4F80A9308D4E80A90E8DBE7F8DBD7F A01BB9D07F979E048BD0F7A0, 4004	<118>
1129 DATA 60EE00D0EE01D0EE01D060AD00D0C9FE900ECE C77FADC77FF00160A9018DCB, 4405	<108>	1162 DATA 0FB9EB7F99CE048BD0F7A018B9FA7F9946058B D0F7A018B9128099960588D0, 4248	<55>
1130 DATA 7FAD01D0C9BE9066C9C907031A9108DFBC3A92F 8DF9C38DFAC3A94ABD03D0A9, 4665	<99>	1163 DATA 7A01FB92AB09936068BD0F7A011B949809986 0688D0F7A019B95AB099FE06, 4015	<227>
1131 DATA 4ABD05D0A9308DFAC3EE00D0CEC97FADC97FF0 0160A90C8DC97FEE01D060A9, 4411	<207>	1164 DATA 8BD0F7A015B973809926078BD0F7A01DB988B0 994E078BD0F7A016B9A58099, 4269	<122>
1132 DATA 108DFBC3A92F8DF9C38DFAC3A94ABD03D0A94A 8D05D0A9308DFAC3EE00D0CE, 4634	<101>	1165 DATA 76078BD0F7A011B98B80999E078BD0F7AD00DC 2908F018AD00DC2904F019AD, 3884	<24>
1133 DATA C97FADC97FF00160A9048DC97FEE01D060A91B 8DFBC3A9318DF9C38DFAC3A9, 4686	<138>	1166 DATA 00DC2902F01AAD00DC2901F01B4CDB7EA90ABD D07F4C1A7FA9118DD07F4C1A, 3300	<54>
1134 DATA 4ABD03D0A9518D05D0A9328DFAC3EE00D0CEC9 7FADC97FF00160A9018DC97F, 4292	<188>	1167 DATA 7FA918BDD07F4C1A7FA91EBDD07F4C1A7FA901 BDRE7F8DBD7FA9FF8BD15D0AD, 3981	<255>
1135 DATA EEO1D060A9178DFBC3A9218DF9C3A9228DFAC3 A9008DCD7F8DCE7F8DCE7FA9, 4655	<168>	1168 DATA 00DD29FC8D00DDA9C08D8802A908BD18D04C8D 75CECF7FADCF7FC9FFF00160, 4240	<176>
1136 DATA 4ABD03D0A948D05D0CEC977FADC77FF00160A9 018DCB7F60A90085FBA9C085, 4172	<115>	1169 DATA CECE7FA9098DCE7FADCE7FC9FFF00160CECD7F A9098DCE7FADCF7FC9FFF001, 4883	<49>
1137 DATA FCAD00D0E9184A4A4AB5FD18A5FB65FDB5FBAD 01D0E9324A4A4AB5FEA6FE18, 4697	<46>	1170 DATA 60A9008DCE7F8DCE7F8DCE7F60001716141412 120F0F0D0D0D0C0C0C0A09, 2045	<112>
1138 DATA A5FB692885FBA5FC690085FCCAD0F0A000B1FB C9EAF00160A9028DC47F60AD, 4766	<127>	1171 DATA 0202011011213121114150419960406320304 1E0302A50202150207120214, 671	<81>
1139 DATA 04DCC9049005C909903B60ADB77FC901F00160 A5A22901F016CEB77FACB77F, 3738	<122>	1172 DATA 48020DAA0001AA0101000000D7FB0400151B19 000500006A00010128286400, 1269	<228>
1140 DATA B9B478DFBC3ADC57FD00160EEFBC360EEB77F ACB77FB9B478DFBC3ADC57F, 5258	<218>	1173 DATA 000010C6400013236300208031E1309052013 090E0420040905130D010C20, 547	<180>
1141 DATA D00160EEFBC360ADC57FF00160A9018DC57FA9 0C8DC67FEFBC360A200A000, 4297	<171>	1174 DATA 32362C30200D05140512170509142007051310 12150E07050E061505122009, 595	<4>
1142 DATA CBD0FDEBCEB07FD0F560AD34C1C9F9B0066904 8D34C160AD34C1E9068D34C1, 4697	<172>	1175 DATA 08120520080F051210051208010C14150E0705 1208010C14050E2013090520, 414	<190>
1143 DATA EE32C1AD32C1C9FEF00160AD32C1E90ABD32C1 EE31C160ADC47FF00160ADFB, 4562	<252>	1176 DATA 30302C302010150E0B14050401132005120709 02142005090E052007051301, 553	<126>
1144 DATA C3C915F021C916F059C919F016C91AF009C91B F00BC91CF007604C0D7D4C43, 3611	<27>	1177 DATA 0D1410150E0B14A01080C160F0E202032362C 302010150E0B14050E041215, 649	<215>
1145 DATA 7D4C637D4CDD7C203C7FA9338DF9C3A9348DFA C3A94EBD03D0A948BD05D0A9, 4040	<254>	1178 DATA 05030B050E201309052004050E200A0F191314 09030B120115062020200615, 471	<167>
1146 DATA D58D02D0A9EDBD04D0ADB17FC9FAB00A18ADB1 7F69028DB17F6018ADB37F69, 4194	<233>	1179 DATA 051220010E0601050E07051212150E14051220 0615051220060F1214070513, 426	<120>
1147 DATA 028DB37F60203C7FA9338DF9C3A9368DFAC3A9 4E8D03D0A948BD05D0A9D58D, 4097	<95>	1180 DATA 030812091414050E050E0C090E0B1320200615 05122013050D092D10120F06, 473	<165>
1148 DATA 02D0A9EDBD04D0ADB17FC9FAB00A18ADB17F69 038DB17F60ADB37F69038DB3, 4140	<242>	1181 DATA 0912050308141320061505122010120F060900 0000000000000000000000, 260	<201>
1149 DATA 7F60203C7FA91FBDF9C3A9208DFAC3A94EBD03 D0A94EBD05D0A9D58D02D0A9, 4116	<43>	1182 DATA 00 000000000000000000000000, 0	<37>
1150 DATA E8BD04D0ADB17FC9FAB004EEB17F60EEB37F60 203C7FA918DF9C3A91E8DFA, 4568	<43>	1183 DATA 000000FF00BFBFC0F7FFFFFFF7FFFFFFF7FFFFFFF FFFFFFFFFFFFFFFFFFFFFFFF, 6622	<93>
1151 DATA C3A94EBD03D0A9508D05D0A9D58D02D0A9EDBD 04D0ADB17FC9FAB007EEB17F, 4542	<165>	1184 DATA FFFFFFFF, 765	<59>

ENDE DES LISTINGS

JOB RACE

Sie wollen doch nicht
zu spät zur Arbeit
kommen?
Na, dann drehen Sie
Ihre Kiste mal auf!

programme

Wir schreiben das Jahr 3486. Sie als Spieler sind (mal wieder) arbeitslos, was ja kein seltenes Problem ist. ABER: Beim allabendlichen Durchstöbern Ihrer Zeitung, der Ghetto-Gazette, entdecken Sie die Anzeige des Cinort-Galrev Verlages, der eine Stelle als Obermottz veräußert.

ALSO: Nichts wie rein in Ihren O.S.-GTI, ein in dieser Zeit sehr gebräuchlicher Nuklearkleinwagen mit automatisch geregelter

Geschwindigkeitsbegrenzung, die Ihnen allerdings vorgestern kaputt gegangen ist.

Insgesamt gilt es, die 9 Levels des Programmes mit den fünf zur Verfügung stehenden Wagen zu durchfahren, was aber (natürlich) nicht zu einfach ist.

Eingabe Hinweise:

- Teil 1 LOADER
- Teil 2 VEC
- Teil 3 Dataset
- Teil 4 J.R. - MAIN

Teil 1

C-64 LISTING + CHECKSUMMEN (C V1.0)

```
0 REM ***** JOB RACE - LOADER ***** <231>
1 PRINT"(CLEAR YELLOW RIGHT2)PARAVISION(SPACE)CO <11>
MPANY(SPACE)PROUDLY(SPACE)PRESENTS:";A$=CHR$(34)
2 PRINTTAB(13)"(GREEN)ULRICH(SPACE)MUEHL" S" <197>
3 PRINTTAB(7)"(DOWN RED)J(SPACE)O(SPACE)B(SPACE) <214>
E4)R(SPACE)A(SPACE)C(SPACE)E"
4 PRINT"(WHITE)LOADING...(BLACK DOWN2)"POKE 532 <121>
81,0:POKE 53280,0
5 PRINT"LOAD"A$"VEC"A$,8" <215>
6 PRINT"(DOWN4)RUN" <13>
7 PRINT"(DOWN2)LOAD"A$"DATASET"A$,8" <62>
8 PRINT"(DOWN4)RUN" <15>
9 PRINT"(UP17)";POKE631,13:POKE632,13:POKE633,13 <7>
:POKE634,13 <162>
10 POKE 198,4
ENDE DES LISTINGS
```

Teil 2

C-64 LISTING + CHECKSUMMEN (C V1.0)

```
0 REM VEKTOREN AENDERN : BASIC-START <236>
1 POKE16385,0:POKE 16386,0:POKE43,2:POKE44,64:NE <211>
W
ENDE DES LISTINGS
```

Teil 3

C-64 LISTING + CHECKSUMMEN (C V1.0)

```
0 REM"(SU CH)PARAVISION(SPACE)COMPANY(SPACE)PRES <208>
ENTS:(CO SI
1 REM"(SB SPACE)JOB(SPACE)RACE:A(SPACE)STORY(SPA <22>
CE)OUT(SPACE)OF(SPACE)LIFE(SPACE SB
2 REM"(SB)WRITTEN(SPACE)IN(SPACE)'86(SPACE)BY(SP <58>
ACE)ULRICH(SPACE)MUEHL(SB
3 REM"(SB SPACE)COPYRIGHT(SPACE)1986(SPACE)BY(SP <142>
ACE)PARAVISION(SPACE SB
4 REM"(SB)DR. BEUERMANNSTR. 23(SB)TELEFON:(SPACE3 <88>
SB
5 REM"(SB)3440(SPACE)ESCHWEGE(SU SC4 SK SPACE)05 <213>
651/5534(SB
6 REM"(SJ SC13 CE SC16 SK <188>
7 POKE 53281,0:POKE 53280,0:PRINT"(CLEAR YELLOW
DOWN2 RIGHT6)PARAVISION(SPACE)COMPANY(SPACE)PRES <91>
ENTS(SPACE):"
8 PRINT"(GREEN DOWN2 RIGHT14)'JOB(SPACE)RACE" <33>
9 PRINT"(DOWN2 RIGHT9)A(SPACE)STORY(SPACE)OUT(SP <169>
ACE)OF(SPACE)LIFE..."
10 PRINT"(DOWN7 RIGHT7 CYAN)GUESS(SPACE)WHAT(SPA <82>
CE)I'M(SPACE)READING..."
11 POKE 54277,75:POKE 54278,58:POKE 54277+7,9:PO <217>
KE 54278+7,42:POKE 54277+14,6
12 POKE 54278+14,0:POKE 54275+14,0:POKE 54275,15 <72>
:POKE 54275+7,15
```

```
13 FOR X=49408 TO 49537:READ DT:POKE X,DT:NEXT X <91>
14 DATA58,138,60,21,0,46,118,20,65,24 <116>
15 DATA43,219,30,65,76,46,118,30,65,24 <162>
16 DATA52,39,20,65,24,46,118,60,19,24 <58>
17 DATA34,207,20,65,24,69,157,30,65,76 <150>
18 DATA58,138,30,65,24,52,39,20,65,24 <19>
19 DATA69,157,60,19,24,58,138,20,33,24 <38>
20 DATA65,181,30,33,24,58,130,30,33,24 <19>
21 DATA52,39,20,33,24,32,219,60,65,135 <81>
22 DATA39,18,20,65,24,32,219,10,65,80 <233>
23 DATA39,18,10,65,10,39,18,10,65,25 <223>
24 DATA32,219,10,65,40,39,18,10,65,55 <81>
25 DATA39,18,10,65,70,32,219,10,65,85 <137>
26 DATA39,18,10,65,100,0,0,0,0 <251>
27 FOR X=49152 TO 49351:READ DT:POKE X,DT:NEXT X <248>
28 DATA14,162,10,65,24,14,162,10,65,24 <109>
29 DATA14,162,10,129,24,14,162,10,65,24 <199>
30 DATA14,162,30,65,24,14,162,10,65,24 <129>
31 DATA14,162,30,65,24,14,162,10,65,24 <130>
32 DATA14,162,30,65,24,14,162,10,65,24 <131>
33 DATA11,158,10,65,24,11,158,10,65,24 <200>
34 DATA14,162,10,129,24,14,162,10,65,24 <204>
35 DATA11,158,30,65,24,11,158,10,65,24 <220>
36 DATA11,158,30,65,24,11,158,10,65,24 <221>
37 DATA11,158,30,65,24,11,158,10,65,24 <222>
38 DATA17,103,10,65,24,17,103,10,65,24 <47>
39 DATA14,162,10,129,24,14,162,10,65,24 <209>
40 DATA17,103,30,65,24,17,103,10,65,24 <67>
41 DATA17,103,30,65,24,17,103,10,65,24 <68>
42 DATA17,103,30,65,24,17,103,10,65,24 <69>
43 DATA21,237,30,65,24,21,237,10,65,24 <188>
44 DATA21,237,10,65,24,21,237,10,65,24 <171>
45 DATA50,37,10,129,24,21,237,10,129,24 <14>
46 DATA26,20,30,65,24,21,237,30,65,24 <166>
47 DATA26,20,20,65,24,0,0,0,0 <96>
48 FOR X=49664 TO 49908:READ DT:POKE X,DT:NEXT X <171>
49 DATA11,158,20,65,24,11,158,10,65,8 <27>
50 DATA11,158,20,65,24,11,158,10,65,8 <28>
51 DATA11,158,10,65,24,100,10,10,129,0 <7>
52 DATA11,158,20,65,24,11,158,10,65,8 <30>
53 DATA11,158,20,65,24,11,158,10,65,8 <31>
54 DATA11,158,10,65,24,100,10,10,129,0 <10>
55 DATA9,196,20,65,24,9,196,10,65,8 <47>
56 DATA9,196,20,65,24,9,196,10,65,8 <48>
57 DATA9,196,10,65,24,100,10,10,129,0 <23>
58 DATA9,196,20,65,24,9,196,10,65,8 <50>
59 DATA9,196,20,65,24,9,196,10,65,8 <51>
60 DATA9,196,10,65,24,100,10,10,129,0 <26>
61 DATA14,162,20,65,24,14,162,10,65,8 <209>
62 DATA14,162,20,65,24,14,162,10,65,8 <210>
63 DATA14,162,10,65,24,100,10,10,129,0 <248>
64 DATA14,162,20,65,24,14,162,10,65,8 <212>
65 DATA14,162,20,65,24,14,162,10,65,8 <213>
66 DATA14,162,10,65,24,100,10,10,129,0 <251>
67 DATA21,237,20,65,24,21,237,10,65,8 <5>
68 DATA21,237,20,65,24,21,237,10,65,8 <6>
69 DATA21,237,10,65,24,100,10,10,129,0 <13>
70 DATA21,237,20,65,24,21,237,20,65,4 <165>
71 DATA21,237,10,65,24,21,237,10,65,4 <132>
72 DATA100,10,10,129,0,100,10,10,129,0 <121>
73 DATA 0,0,0,0,0 <140>
74 FOR X=832 TO 844:READ DT:POKE X,DT:NEXT X <202>
75 DATA120,169,0,141,20,3,169,196,141,21,3,88,96 <51>
76 FOR X=50176 TO 50457:READ DT:POKE X,DT:NEXT X <53>
77 DATA173,255,195,208,71,173,254,195,170,189,0, <40>
193,208,6,169,0,170,141,254
78 DATA195,189,0,193,141,1,212,232,189,0,193,141 <46>
,0,212,232,189,0,193,141,255
79 DATA195,169,0,141,4,212,232,189,0,193,141,4,2 <153>
12,232,189,0,193,141,251,195
80 DATA232,138,141,254,195,169,0,234,141,249,195 <86>
,141,2,212,141,249,195,173
81 DATA249,195,237,251,195,141,2,212,141,249,195 <18>
,206,255,195,173,223,195,208
82 DATA71,173,222,195,170,189,0,192,208,6,169,0, <180>
170,141,222,195,189,0,192
83 DATA141,8,212,232,189,0,192,141,7,212,232,189, <25>
0,192,141,223,195,169,0,141
84 DATA11,212,232,189,0,192,141,11,212,232,189,0, <179>
192,141,219,195,232,138,141
85 DATA222,195,169,0,234,141,217,195,141,9,212,1 <165>
41,217,195,173,217,195,237
86 DATA219,195,141,9,212,141,217,195,206,223,195 <46>
,173,191,195,208,71,173,190
87 DATA195,170,189,0,194,208,6,169,0,170,141,254 <44>
,194,189,0,194,141,15,212
```


88 DATA232,189,0,194,141,14,212,232,189,0,194,14,1,191,195,169,0,141,18,212	<174>	160 DATA 154, 128, 152, 169, 128, 152, 9, 128, 1	<195>
89 DATA232,189,0,194,141,18,212,232,189,0,194,14,1,187,195,232,138,141,190	<5>	161 DATA 9, 128, 152, 9, 128, 152, 9, 96, 152	<255>
90 DATA195,169,0,234,141,185,195,141,16,212,141,185,195,173,185,195,237,187	<13>	162 DATA 2, 98, 88, 2, 98, 98, 2, 98, 98	<103>
91 DATA195,141,16,212,141,185,195,206,191,195,17,3,24,212,141,17,212,76,49	<197>	163 DATA 2, 98, 98, 2, 98, 98, 2, 98, 98	<111>
92 DATA234	<172>	164 DATA 130, 98, 106, 169, 98, 85, 149, 96, 165	<28>
93 POKE 54295,0:POKE 54296,15:FOR X=50104 TO 50175:POKE X,0:NEXT:SYS 832	<15>	165 DATA 149, 96, 10, 149, 128, 0, 42, 0, 0	<117>
94 FOR X=2048 TO 3519:READ DT:POKE X,DT:NEXT X	<171>	166 DATA 0	<137>
95 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<130>	167 DATA 0, 0, 0, 162, 170, 0, 88, 149, 128	<132>
96 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<131>	168 DATA 88, 153, 96, 152, 154, 96, 152, 154, 96	<199>
97 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0*	<132>	169 DATA 152, 153, 96, 152, 153, 128, 152, 153, 128	<54>
98 DATA 0, 236, 0, 0, 168, 0, 2, 170, 0	<37>	170 DATA 152, 149, 160, 96, 154, 88, 96, 37, 88	<234>
99 DATA 2, 86, 0, 2, 118, 0, 10, 118, 128	<209>	171 DATA 96, 38, 152, 96, 38, 150, 96, 38, 38	<197>
100 DATA 9, 117, 128, 9, 253, 128, 41, 85, 160	<8>	172 DATA 96, 38, 38, 96, 38, 150, 96, 38, 88	<199>
101 DATA 42, 170, 160, 14, 238, 192, 1, 69, 0	<126>	173 DATA 130, 37, 96, 0, 170, 128, 0, 0, 0	<14>
102 DATA 0	<73>	174 DATA 0	<145>
103 DATA 0, 40, 0, 0, 190, 0, 2, 170, 128	<125>	175 DATA 10, 128, 0, 175, 224, 0, 46, 224, 10	<205>
104 DATA 2, 130, 128, 10, 0, 160, 8, 0, 32	<232>	176 DATA 46, 248, 11, 46, 184, 47, 46, 184, 46	<66>
105 DATA 8, 60, 32, 8, 235, 32, 15, 255, 240	<206>	177 DATA 46, 184, 46, 46, 248, 46, 47, 224, 46	<26>
106 DATA 21, 85, 84, 42, 170, 168, 21, 153, 148	<25>	178 DATA 175, 128, 46, 187, 128, 184, 187, 128, 184	<78>
107 DATA 25, 153, 152, 21, 149, 152, 26, 166, 14	<11>	179 DATA 186, 224, 186, 186, 224, 191, 186, 224, 186	<191>
108 DATA 42, 170, 168, 21, 85, 84, 59, 187, 184	<226>	180 DATA 184, 184, 184, 184, 186, 224, 184, 186, 224	<149>
109 DATA 46, 238, 236, 21, 170, 84, 63, 0, 252	<58>	181 DATA 168, 170, 224, 0, 2, 224, 0, 2, 160	<147>
110 DATA 0	<81>	182 DATA 0	<153>
111 DATA 0, 84, 0, 0, 100, 0, 0, 100, 0	<80>	183 DATA 0, 0, 0, 0, 0, 0, 160, 0, 168	<212>
112 DATA 0, 168, 0, 0, 168, 0, 0, 168, 0	<193>	184 DATA 224, 10, 254, 248, 11, 235, 184, 11, 13	<93>
113 DATA 0, 168, 0, 2, 170, 0, 2, 170, 0	<252>	185 DATA 184, 11, 128, 184, 11, 128, 184, 11, 12	<106>
114 DATA 2, 186, 0, 2, 186, 0, 10, 254, 128	<88>	186 DATA 184, 11, 128, 46, 47, 128, 46, 46, 0	<77>
115 DATA 10, 254, 128, 10, 254, 128, 43, 255, 16	<34>	187 DATA 174, 46, 0, 254, 46, 0, 174, 46, 0	<134>
116 DATA 43, 255, 160, 170, 254, 168, 42, 170, 1	<173>	188 DATA 46, 46, 0, 11, 174, 0, 11, 174, 2	<163>
117 DATA 6, 170, 64, 1, 101, 0, 0, 0, 0	<134>	189 DATA 11, 175, 171, 10, 138, 254, 0, 0, 168	<96>
118 DATA 0	<89>	190 DATA 0	<161>
119 DATA 26, 0, 164, 91, 190, 229, 171, 170, 234	<8>	191 DATA 0, 42, 34, 10, 190, 34, 11, 250, 34	<93>
120 DATA 171, 170, 234, 43, 170, 232, 46, 150, 1	<185>	192 DATA 139, 160, 42, 139, 128, 0, 139, 128, 34	<99>
121 DATA 42, 105, 168, 9, 235, 96, 9, 190, 96	<215>	193 DATA 11, 128, 42, 11, 128, 34, 11, 170, 34	<238>
122 DATA 9, 235, 96, 9, 190, 96, 89, 150, 101	<109>	194 DATA 11, 254, 34, 2, 234, 0, 2, 224, 0	<28>
123 DATA 173, 105, 122, 174, 170, 186, 171, 170, 234	<170>	195 DATA 2, 224, 0, 2, 224, 0, 2, 224, 0	<48>
124 DATA 171, 170, 234, 107, 170, 233, 27, 170, 228	<154>	196 DATA 2, 224, 0, 2, 234, 128, 130, 239, 128	<248>
125 DATA 62, 170, 188, 0, 40, 0, 0, 0, 0	<102>	197 DATA 130, 250, 128, 128, 160, 0, 0, 0, 0	<242>
126 DATA 0	<97>	198 DATA 0	<169>
127 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<162>	199 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<234>
128 DATA 0, 168, 0, 10, 118, 128, 42, 86, 160	<232>	200 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<235>
129 DATA 41, 153, 160, 153, 153, 152, 153, 101, 152	<74>	201 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<236>
130 DATA 153, 101, 152, 153, 185, 152, 153, 185, 152	<237>	202 DATA 0, 212, 0, 0, 100, 0, 2, 170, 0	<173>
131 DATA 153, 185, 152, 154, 254, 152, 42, 254, 160	<118>	203 DATA 2, 85, 0, 1, 118, 0, 9, 118, 64	<241>
132 DATA 43, 255, 160, 10, 254, 128, 0, 168, 0	<224>	204 DATA 9, 117, 128, 5, 253, 64, 37, 85, 160	<226>
133 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<168>	205 DATA 41, 106, 96, 14, 238, 192, 1, 69, 0	<165>
134 DATA 0	<105>	206 DATA 0	<177>
135 DATA 170, 173, 0, 180, 11, 64, 180, 11, 64	<224>	207 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<242>
136 DATA 180, 11, 64, 180, 11, 64, 180, 11, 64	<4>	208 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<243>
137 DATA 180, 11, 64, 180, 11, 64, 170, 173, 0	<186>	209 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<244>
138 DATA 180, 0, 0, 180, 0, 0, 180, 0, 0	<139>	210 DATA 0, 212, 0, 0, 84, 0, 0, 238, 0	<12>
139 DATA 180, 0, 0, 180, 0, 0, 180, 0, 0	<140>	211 DATA 2, 85, 0, 3, 242, 0, 13, 246, 0	<217>
140 DATA 180, 0, 0, 180, 0, 0, 180, 0, 0	<141>	212 DATA 5, 5, 128, 4, 252, 64, 5, 71, 64	<38>
141 DATA 180, 0, 0, 180, 0, 0, 0, 0, 0	<36>	213 DATA 1, 110, 64, 14, 238, 192, 1, 69, 0	<247>
142 DATA 0	<113>	214 DATA 0	<185>
143 DATA 45, 1, 224, 45, 1, 224, 45, 1, 224	<254>	215 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<250>
144 DATA 45, 1, 224, 11, 71, 128, 11, 71, 128	<117>	216 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<251>
145 DATA 11, 71, 128, 11, 71, 128, 2, 222, 0	<216>	217 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<252>
146 DATA 2, 222, 0, 2, 222, 0, 2, 222, 0	<171>	218 DATA 0, 4, 0, 0, 92, 0, 0, 46, 0	<105>
147 DATA 0, 184, 0, 0, 184, 0, 0, 184, 0	<138>	219 DATA 2, 117, 0, 3, 243, 0, 13, 245, 0	<129>
148 DATA 0, 184, 0, 0, 32, 0, 0, 32, 0	<174>	220 DATA 5, 13, 128, 4, 252, 64, 5, 71, 64	<42>
149 DATA 0, 32, 0, 0, 32, 0, 0, 0, 0	<43>	221 DATA 3, 96, 0, 0, 221, 192, 1, 64, 0	<5>
150 DATA 0	<121>	222 DATA 0	<193>
151 DATA 30, 170, 0, 120, 7, 128, 120, 0, 0	<177>	223 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<2>
152 DATA 120, 0, 0, 120, 0, 0, 120, 0, 0	<211>	224 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<3>
153 DATA 120, 0, 0, 120, 0, 0, 120, 0, 0	<212>	225 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<4>
154 DATA 120, 0, 0, 120, 0, 0, 120, 0, 0	<213>	226 DATA 0, 4, 0, 0, 92, 0, 0, 0, 0	<5>
155 DATA 120, 0, 0, 120, 0, 0, 120, 0, 0	<214>	227 DATA 2, 116, 0, 3, 243, 0, 13, 245, 0	<131>
156 DATA 120, 0, 0, 120, 0, 0, 120, 0, 0	<215>	228 DATA 1, 0, 128, 0, 252, 0, 0, 71, 64	<57>
157 DATA 120, 7, 128, 30, 170, 0, 0, 0, 0	<59>	229 DATA 0, 64, 0, 0, 205, 192, 0, 0, 0	<168>
158 DATA 0	<129>	230 DATA 0	<201>
159 DATA 0, 168, 42, 10, 88, 149, 165, 168, 154	<81>	231 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<10>
		232 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<11>
		233 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<12>
		234 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<13>
		235 DATA 0, 52, 0, 3, 243, 0, 12, 0, 0	<5>
		236 DATA 1, 0, 0, 0, 252, 0, 0, 7, 64	<222>
		237 DATA 0, 0, 0, 0, 204, 192, 0, 0, 0	<49>
		238 DATA 0	<209>

programme

239 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 240 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 241 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 242 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 243 DATA 0, 0, 0, 42, 8, 42, 34, 34, 32
 244 DATA 42, 42, 40, 32, 34, 32, 32, 34, 32
 245 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 246 DATA 0
 247 DATA 170, 170, 170, 170, 239, 171, 186, 238, 238
 248 DATA 186, 238, 238, 186, 238, 238, 186, 238, 238
 249 DATA 190, 238, 235, 170, 170, 170, 128, 0, 0
 250 DATA 128, 0, 0, 128, 0, 0, 128, 0, 0
 251 DATA 128, 0, 0, 128, 0, 0, 128, 0, 0
 252 DATA 128, 0, 0, 128, 0, 0, 128, 0, 0
 253 DATA 128, 0, 0, 128, 0, 0, 128, 0, 0
 254 DATA 0
 255 DATA 170, 170, 170, 175, 234, 238, 235, 170
 256 DATA 239, 171, 170, 238, 235, 174, 238, 235, 170
 257 DATA 174, 235, 170, 170, 170, 170, 0, 0, 0
 258 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 259 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 260 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 261 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 262 DATA 0
 263 DATA 170, 170, 170, 174, 174, 186, 187, 187, 186
 264 DATA 186, 187, 186, 191, 191, 186, 187, 187, 186
 265 DATA 174, 187, 190, 170, 170, 170, 0, 0, 0
 266 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 267 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 268 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 269 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
 270 DATA 0
 271 DATA 170, 170, 170, 250, 254, 238, 238, 234, 238
 272 DATA 250, 250, 238, 238, 234, 238, 238, 234, 238
 273 DATA 238, 254, 186, 170, 170, 170, 0, 0, 2
 274 DATA 0, 0, 2, 0, 0, 2, 0, 0, 2
 275 DATA 0, 0, 2, 0, 0, 2, 0, 0, 2
 276 DATA 0, 0, 2, 0, 0, 2, 0, 0, 2
 277 DATA 0, 0, 2, 0, 0, 2, 0, 0, 2
 278 DATA 0
 279 FOR X=12288 TO 13567:READ DT:POKE X,DT:NEXT X
 32000 DATA 9, 61, 54, 116, 60, 0, 60, 12, 30, 51, 63, 102, 102, 102, 0, 62, 51, 51, 62, 102, 102
 32001 DATA 124, 0, 30, 51, 48, 48, 96, 102, 60, 0, 60, 54, 51, 51, 102, 108, 120, 0, 63, 48, 48, 60
 32002 DATA 96, 96, 126, 0, 63, 48, 48, 60, 96, 96, 96, 0, 30, 51, 48, 55, 102, 102, 60, 0, 51, 51, 51
 32003 DATA 63, 102, 102, 102, 0, 30, 12, 12, 12, 24, 24, 60, 0, 15, 6, 6, 6, 12, 108, 56, 0, 51, 54
 32004 DATA 60, 56, 120, 108, 102, 0, 48, 48, 48, 96, 96, 126, 0, 49, 59, 63, 53, 99, 99, 99, 0
 32005 DATA 51, 59, 63, 63, 110, 102, 102, 0, 30, 51, 51, 51, 102, 102, 60, 0, 62, 51, 51, 62, 96, 96
 32006 DATA 96, 0, 30, 51, 51, 51, 102, 60, 14, 0, 62, 51, 51, 62, 120, 108, 102, 0, 30, 51, 48, 30
 32007 DATA 6, 102, 60, 0, 63, 12, 12, 12, 24, 24, 24, 0, 51, 51, 1, 51, 51, 102, 102, 60, 0, 51, 51, 51
 32008 DATA 51, 102, 60, 24, 0, 49, 49, 53, 127, 119, 99, 0, 51, 51, 30, 12, 60, 102, 102, 0, 51
 32009 DATA 51, 51, 30, 24, 24, 24, 0, 63, 3, 6, 12, 48, 96, 12, 6, 0, 30, 24, 24, 48, 48, 60, 0, 6
 32010 DATA 9, 24, 62, 48, 98, 252, 0, 30, 6, 6, 6, 12, 12, 60, 0, 0, 12, 30, 63, 24, 24, 24, 0, 8
 32011 DATA 24, 63, 127, 48, 16, 0, 0, 0, 0, 0, 0, 0, 0, 12, 1, 2, 12, 12, 0, 0, 24, 0, 51, 51, 51, 0, 0
 32012 DATA 0, 0, 51, 51, 127, 51, 255, 102, 102, 0, 12, 31, 48, 30, 6, 124, 24, 0, 49, 51, 6, 12
 32013 DATA 48, 102, 70, 0, 30, 51, 30, 28, 103, 102, 63, 0, 3, 6, 12, 0, 0, 0, 0, 6, 12, 24, 24, 48
 32014 DATA 24, 12, 0, 24, 12, 6, 6, 12, 24, 48, 0, 0, 51, 30, 1, 27, 60, 102, 0, 0, 12, 12, 63, 24, 24
 32015 DATA 0, 0, 0, 0, 0, 0, 24, 48, 0, 0, 63, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 24, 0, 0, 1, 3, 6, 24
 32016 DATA 48, 96, 0, 30, 51, 55, 59, 102, 102, 60, 0, 12, 12, 28, 12, 24, 24, 126, 0, 30, 51, 3, 6
 32017 DATA 48, 96, 126, 0, 30, 51, 3, 14, 6, 102, 60, 0, 3, 7, 15, 51, 127, 6, 6, 0, 63, 48, 62, 3, 6

<18> 32018 DATA 102, 60, 0, 30, 51, 48, 62, 102, 102, 60, 0, 63, 5
 <19> 1, 6, 12, 24, 24, 24, 0, 30, 51, 51, 30
 <20> 32019 DATA 102, 102, 60, 0, 30, 51, 51, 31, 6, 102, 60, 0, 0,
 <21> 0, 12, 0, 0, 24, 0, 0, 0, 12, 0, 0, 24
 <22> 32020 DATA 24, 48, 7, 12, 24, 48, 48, 24, 14, 0, 0, 0, 63, 0, 1
 <42> 26, 0, 0, 0, 56, 12, 6, 3, 12, 24, 112
 <24> 32021 DATA 0, 30, 51, 3, 6, 24, 0, 24, 0, 0, 0, 127, 255, 0,
 <217> 0, 0, 47, 171, 170, 138, 42, 139, 139
 <228> 32022 DATA 47, 60, 48, 60, 172, 175, 131, 163, 163, 143, 14
 0, 172, 188, 188, 172, 12, 12, 255, 255
 <7> 32023 DATA 255, 247, 213, 213, 255, 247, 255, 247, 255, 24
 7, 255, 255, 255, 255, 87, 119, 87, 87
 <216> 32024 DATA 87, 87, 87, 87, 85, 85, 85, 85, 85, 85, 85, 85
 <85> , 119, 85, 85, 85, 85, 85, 85, 85, 85
 <86> 32025 DATA 85, 85, 85, 85, 85, 255, 191, 251, 255, 255,
 <87> 251, 255, 255, 251, 255, 255, 239, 255
 <88> 32026 DATA 191, 251, 255, 255, 239, 255, 255, 254, 255, 19
 <225> 1, 255, 171, 187, 187, 187, 171, 191
 <191> 32027 DATA 191, 255, 187, 187, 187, 187, 187, 239, 25
 5, 239, 187, 191, 191, 191, 187, 239
 <241> 32028 DATA 255, 239, 171, 171, 171, 171, 171, 255, 251, 25
 1, 234, 251, 251, 251, 255, 255
 <201> 32029 DATA 251, 255, 238, 255, 255, 255, 255, 175, 23
 <38> 5, 250, 190, 175, 235, 250, 190, 175
 <39> 32030 DATA 235, 250, 254, 255, 251, 190, 175, 190, 250, 23
 <40> 5, 175, 190, 250, 235, 175, 195, 231
 <41> 32031 DATA 126, 60, 60, 126, 231, 195, 0, 60, 126, 102, 102
 <234> , 126, 60, 0, 24, 24, 102, 102, 24, 24
 <58> 32032 DATA 60, 0, 6, 6, 6, 6, 6, 6, 6, 8, 28, 62, 127, 62, 28
 , 8, 0, 24, 24, 24, 255, 255, 24, 24, 24
 <192> 32033 DATA 192, 192, 48, 48, 192, 192, 48, 48, 24, 24, 24, 2
 <20> 4, 24, 24, 24, 0, 0, 3, 62, 118, 54
 <46> 32034 DATA 54, 0, 255, 127, 63, 31, 15, 7, 3, 1, 0, 0, 0, 0,
 <47> 0, 0, 0, 240, 240, 240, 240, 240, 240
 <48> 32035 DATA 240, 240, 0, 0, 0, 0, 255, 255, 255, 255, 255, 0,
 <49> 0, 0, 0, 0, 0, 192, 224, 241, 251, 223
 <242> 32036 DATA 206, 196, 192, 97, 225, 227, 99, 98, 102, 1
 03, 199, 199, 192, 96, 96, 32, 48, 240
 <62> 32037 DATA 255, 255, 48, 48, 48, 48, 190, 191, 51, 4
 8, 48, 51, 63, 62, 27, 155, 219, 217
 <173> 32038 DATA 217, 216, 152, 25, 1, 131, 199, 239, 255, 254, 2
 <116> 54, 255, 128, 128, 128, 0, 0, 0, 0
 <126> 32039 DATA 192, 192, 192, 192, 0, 64, 117, 85, 103, 110, 10
 <127> 8, 108, 0, 0, 40, 40, 240, 56, 24, 24
 <128> 32040 DATA 0, 0, 136, 216, 48, 48, 48, 48, 0, 0, 0, 55, 55,
 <129> 51, 50, 0, 3, 9, 57, 25, 27, 155, 27, 128
 <250> 32041 DATA 131, 135, 198, 239, 199, 128, 23, 29, 16, 16, 29
 , 0, 128, 0, 0, 128, 128, 128, 119
 <42> 32042 DATA 1, 127, 0, 0, 46, 106, 46, 40, 40, 58, 0, 0, 238, 1
 68, 238, 248, 168, 138, 0, 1, 3, 3, 1
 <77> 32043 DATA 0, 7, 63, 255, 255, 252, 240, 192, 241, 224, 192
 , 192, 192, 228, 236, 252, 192, 195
 <165> 32044 DATA 227, 227, 97, 113, 116, 62, 143, 140, 140, 198,
 198, 6, 0, 0, 34, 46, 0, 0, 0, 0, 170
 <165> 32045 DATA 238, 0, 0, 0, 0, 0, 1, 3, 3, 2, 0, 0, 0, 248, 24
 0, 240, 56, 56, 56, 28, 28, 28, 0, 0
 <190> 32046 DATA 0, 0, 0, 195, 153, 145, 145, 159, 153, 195, 25
 5, 231, 195, 153, 129, 153, 153, 153
 <167> 32047 DATA 255, 131, 153, 153, 131, 153, 153, 131, 255, 19
 5, 153, 159, 159, 159, 153, 195, 255
 <121> 32048 DATA 135, 147, 153, 153, 153, 147, 135, 255, 129, 15
 9, 159, 135, 159, 159, 129, 255, 129
 <226> 32049 DATA 159, 159, 135, 159, 159, 159, 255, 195, 153, 15
 9, 145, 153, 153, 195, 255, 153, 153
 <47> 32050 DATA 153, 129, 153, 153, 153, 255, 195, 231, 231, 23
 1, 231, 231, 195, 255, 225, 243, 243
 <163> 32051 DATA 243, 243, 147, 199, 255, 153, 147, 135, 143, 13
 5, 147, 153, 255, 159, 159, 159, 159
 <77> 32052 DATA 159, 159, 129, 255, 156, 136, 128, 148, 156, 15
 6, 156, 255, 153, 137, 129, 129, 145
 <129> 32053 DATA 153, 153, 255, 195, 153, 153, 153, 153, 19
 5, 255, 131, 153, 153, 131, 159, 159
 <67> 32054 DATA 159, 255, 195, 153, 153, 153, 153, 195, 241, 25
 5, 131, 153, 153, 131, 135, 147, 153
 <61> 32055 DATA 255, 195, 153, 159, 195, 249, 153, 195, 255, 12
 9, 231, 231, 231, 231, 231, 255
 <51> 32056 DATA 153, 153, 153, 153, 153, 153, 195, 255, 153, 15
 3, 153, 153, 195, 231, 255, 156
 <5> 32057 DATA 156, 156, 148, 128, 136, 156, 255, 153, 153, 19
 5, 231, 195, 153, 153, 255, 153, 153
 <227> 32058 DATA 153, 195, 231, 231, 231, 255, 129, 249, 243, 23
 1, 207, 159, 129, 255, 195, 207, 207
 <117> 32059 DATA 207, 207, 195, 255, 243, 237, 207, 131, 20
 7, 157, 3, 255, 195, 243, 243, 243
 <252> 32060 DATA 243, 195, 255, 255, 231, 195, 129, 231, 231, 23
 1, 231, 255, 239, 207, 128, 128, 207

<147>
 <150>
 <89>
 <162>
 <37>
 <233>
 <25>
 <93>
 <226>
 <100>
 <247>
 <22>
 <253>
 <217>
 <0>
 <183>
 <57>
 <66>
 <45>
 <27>
 <147>
 <112>
 <250>
 <202>
 <14>
 <186>
 <3>
 <87>
 <76>
 <144>
 <38>
 <9>
 <181>
 <78>
 <7>
 <138>
 <197>
 <108>
 <89>
 <148>
 <68>
 <41>


```

32061 DATA239,255
33000 FOR X=12288 TO 12288+1023:POKE X+1024,255-
PEEK(X):NEXT X
33001 FOR X=845 TO 857:READ DT:POKE X,DT:NEXT
33002 DATA120,169,49,141,20,3,169,234,141,21,3,8
8,96
33999 FOR X=8192 TO 8988:READ DT:POKE X,DT:NEXT
34000 DATA165,2,201,5,208,24,169,0,133,2,173,31,
50,72,162,23,189,7,50,157,8,50
34001 DATA202,138,208,246,104,141,8,50,173,47,50
,72,162,15,189,31,50,157,32,50
34002 DATA202,138,208,246,104,141,32,50,173,63,5
0,72,162,15,189,47,50,157,48
34003 DATA50,202,138,208,246,104,141,48,50,173,7
9,50,72,162,15,189,63,50,157
34004 DATA64,50,202,138,208,246,104,141,64,50,23
0,2,96,32,253,174,32,158,183
34005 DATA142,0,208,32,253,174,32,158,183,142,1,
208,32,253,174,32,158,183,142
34006 DATA2,208,32,253,174,32,158,183,142,3,208,
32,253,174,32,158,183,142,4,208
34007 DATA32,253,174,32,158,183,142,5,208,32,253
,174,32,158,183,142,6,208,32
34008 DATA253,174,32,158,183,142,7,208,32,253,17
4,32,158,183,142,8,208,32,253
34009 DATA174,32,158,183,142,9,208,32,253,174,32
,158,183,142,10,208,32,253,174
34010 DATA32,158,183,142,11,208,32,253,174,32,15
8,183,142,12,208,32,253,174,32
34011 DATA158,183,142,13,208,32,253,174,32,158,1
83,142,14,208,32,253,174,32,158
34012 DATA183,142,15,208,96,32,253,174,32,158,18
3,142,37,208,32,253,174,32,158
34013 DATA183,142,38,208,32,253,174,32,158,183,1
42,39,208,32,253,174,32,158,183
34014 DATA142,40,208,32,253,174,32,158,183,142,4
1,208,32,253,174,32,158,183,142
34015 DATA42,208,32,253,174,32,158,183,142,43,20
8,32,253,174,32,158,183,142,44
34016 DATA208,32,253,174,32,158,183,142,45,208,3
2,253,174,32,158,183,142,46,208
34017 DATA96,32,253,174,32,158,183,142,248,7,32,
253,174,32,158,183,142,249,7
34018 DATA32,253,174,32,158,183,142,250,7,32,253
,174,32,158,183,142,251,7,32
34019 DATA253,174,32,158,183,142,252,7,32,253,17
4,32,158,183,142,253,7,32,253
34020 DATA174,32,158,183,142,254,7,32,253,174,32
,158,183,142,255,7,96,162,14
34021 DATA189,1,208,24,24,125,255,22,157,1,208,2
02,138,208,241,96,173,0,220,41
34022 DATA1,208,13,206,1,208,173,1,208,201,83,20
8,3,238,1,208,173,0,220,41,2
34023 DATA208,13,238,1,208,173,1,208,201,224,208
,3,206,1,208,173,0,220,41,4,208
34024 DATA13,206,0,208,173,0,208,201,96,208,3,23
8,0,208,173,0,220,41,8,208,13
34025 DATA238,0,208,173,0,208,201,254,208,3,206,
0,208,96,173,0,220,41,16,208
34026 DATA1,96,169,0,234,141,39,208,141,40,208,1
41,41,208,141,42,208,141,43,208
34027 DATA162,3,160,255,136,152,208,252,202,138,
208,246,32,0,32,165,161,141,38
34028 DATA208,73,255,141,37,208,162,255,173,18,2
08,141,38,208,73,255,141,37,208
34029 DATA202,138,208,241,76,245,33,32,146,33,17
3,2,208,201,168,208,5,169,255
34030 DATA141,0,23,173,2,208,201,96,208,5,169,1,
141,0,23,32,164,33,173,4,208
34031 DATA201,248,208,5,169,255,141,2,23,173,4,2
08,201,184,208,5,169,1,141,2
34032 DATA23,32,164,33,173,6,208,201,248,208,5,1
69,255,141,4,23,173,6,208,201
34033 DATA96,208,5,169,1,141,4,23,32,164,33,173,
8,208,201,248,208,5,169,255,141
34034 DATA6,23,173,8,208,201,96,208,5,169,1,141,
6,23,32,164,33,32,0,32,32,0,32
34035 DATA32,0,32,173,10,208,201,216,208,5,169,2
55,141,8,23,173,10,208,201,192
34036 DATA208,5,169,1,141,8,23,96,169,15,141,255
,31,141,254,31,32,60,34,162,5
34037 DATA160,255,136,152,208,252,202,138,208,24
6,173,30,208,24,74,144,6,169
34038 DATA1,141,253,31,96,206,255,31,173,255,31,
208,20,169,249,141,255,31,234
34039 DATA206,254,31,173,254,31,208,6,169,0,141,
<65> 253,31,96,76,213,34,238,1,208
<192> 34040 DATA238,3,208,238,5,208,238,7,208,96
<63> 35000 FOR X=15 TO 0 STEP-1:POKE 54296,X:FOR WT=0
TO 250:NEXT WT:NEXT X:SYS 845
35001 FOR X=49408 TO 49660:READ DT:POKE X,DT:NEX
T
35010 DATA 14,162,16,65,24,15,129,8,65,24,15,129
,8,65,24,17,103,8,65,24
35011 DATA 17,103,8,65,24,19,137,8,65,24,19,137,
8,65,24
35012 DATA 14,162,16,65,24,15,129,8,65,24,15,129
,8,65,24,17,103,8,65,24
35013 DATA 17,103,8,65,24,19,137,8,65,24,19,137,
8,65,24
35014 DATA 14,162,16,65,24,15,129,8,65,24,15,129
,8,65,24,17,103,8,65,24
35015 DATA 17,103,8,65,24,19,137,8,65,24,19,137,
8,65,24
35016 DATA 14,162,32,65,24,14,162,16,65,24,14,16
2,8,65,24,14,162,8,65,24
35017 DATA 19,137,16,65,24,21,237,8,65,24,21,237
,8,65,24,23,59,8,65,24
35018 DATA 23,59,8,65,24,26,20,8,65,24,26,20,8,6
5,24
35019 DATA 19,137,16,65,24,21,237,8,65,24,21,237
,8,65,24,23,59,8,65,24
35020 DATA 23,59,8,65,24,26,20,8,65,24,26,20,8,6
5,24
35021 DATA 19,137,16,65,24,21,237,8,65,24,21,237
,8,65,24,23,59,8,65,24
35022 DATA 23,59,8,65,24,26,20,8,65,24,26,20,8,6
5,24
35023 DATA 19,137,32,65,24,19,137,16,65,24,19,13
7,8,65,24,19,137,8,65,24,0,0
35100 FOR X=49152 TO 49324:READ DT:POKE X,DT:NEX
T
35110 DATA 7,81,16,65,80,100,100,8,129,56
35111 DATA 8,180,24,65,80
35112 DATA 9,196,16,65,80
35113 DATA 7,81,16,65,80,100,100,8,129,56
35114 DATA 8,180,24,65,80
35115 DATA 9,196,16,65,80
35116 DATA 7,81,16,65,80,100,100,8,129,56
35117 DATA 8,180,24,65,80
35118 DATA 9,196,16,65,80
35119 DATA 7,81,32,65,80
35120 DATA 7,81,8,65,80
35121 DATA 7,81,8,65,80
35122 DATA 7,81,8,65,80
35123 DATA 7,81,8,65,80
35125 DATA 19,137,16,65,80,100,100,8,129,56
35126 DATA 23,59,24,65,80
35127 DATA 26,20,16,65,80
35128 DATA 19,137,16,65,80,100,100,8,129,56
35129 DATA 23,59,24,65,80
35130 DATA 26,20,16,65,80
35131 DATA 19,137,16,65,80,100,100,8,129,56
35132 DATA 23,59,24,65,80
35133 DATA 26,20,16,65,80
35134 DATA 19,137,32,65,80
35135 DATA 19,137,8,65,80
35136 DATA 19,137,8,65,80
35137 DATA 19,137,8,65,80
35138 DATA 19,137,8,65,80,0,0,0
35150 FOR X=49664 TO 49796:READ DT:POKE X,DT:NEX
T
35160 DATA 20,50,16,129,0,150,150,16,129,0
35161 DATA 20,50,8,129,0,20,50,8,129,0
35162 DATA 150,150,16,129,0,20,50,8,129,0
35163 DATA 150,150,8,129,0,20,50,8,129,0
35164 DATA 20,50,8,129,0,20,50,8,129,0
35165 DATA 150,150,16,129,0,150,150,8,129,0
35166 DATA 150,150,8,129,0,150,150,8,129,0
35167 DATA 150,150,8,129,0,150,150,8,129,0
35168 DATA 120,120,8,129,0,120,120,8,129,0
35169 DATA 120,120,8,129,0,120,120,8,129,0
35170 DATA 120,120,32,129,0,120,120,8,129,0
35171 DATA 120,120,8,129,0,120,120,8,129,0
35172 DATA 120,120,4,129,0,150,150,4,129,0,0,0,0
35175 POKE 54295,0:FOR X=50104 TO 50175:POKE X,0
:NEXT
35200 PRINT"(CLEAR DOWN2)":A$=CHR$(34)
35210 PRINT"LOAD"A$"J.R.-MAIN"A$,8"
35220 PRINT"(DOWN4)RUN(UP8)":POKE 631,13:POKE 63
2,13:POKE 198,2
ENDE DES LISTINGS

```


Teil 4

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 FOR X=50104 TO 50175:POKE X,0:NEXT X <28>
1 POKE53272,PEEK(53272)AND2410R12:POKE53270,PEEK <32>
(53270)OR16:POKE54296,15
2 FOR X=12904 TO 12927:POKE X-96,PEEK(X):NEXT:FD <236>
RX=5888T05904:POKEX,1:NEXT
3 POKE 2040,40:POKE 2041,41:POKE 2042,42:POKE 20 <229>
43,43:POKE 2044,44:POKE2045,37
4 POKE 2046,38:POKE 2047,39:POKE54295,2 <151>
8 FOR X=0 TO 11:PRINT"(BROWN RIGHT8 SD SF SH SF
SH SF SH SF SH SF SH SF SH SF SH SF SH SF SH SF
SH SF SH SD)"
9 PRINT"(RIGHT8 SE SG SI SG SI SG SI SG SI SG SI <138>
SG SI SG SI SG SI SG SI SG SI SE)"
10 NEXT X <212>
11 POKE 53282,12:POKE 53283,1 <60>
12 PRINT"(RVSON WHITE SPACE3)SCORE:000000(SPACE2 <149>
)LIVES:5(SPAC2)PARAVISION(SPAC3 HOME BROWN)":P
RINT"(UP2 LIG.BLUE)":
13 POKE 2023,160:POKE 56295,1 <182>
14 FOR X=0 TO 7:PRINT"(SA SB SC SA SB SC SA SB)" <218>
:PRINT"(SB SC SA SB SC SA SB SC)":PRINT"(SC SA S
B SC SA SB SC SA)"
16 NEXT X:PRINT"(HOME)":FOR X=0 TO 7 <127>
17 PRINTTAB(32)"(SA SB SC SA SB SC SA)":PRINTTAB <111>
(32)"(SB SC SA SB SC SA SB)":PRINTTAB(32)"(SC SA
SB SC SA SB SC)"
19 NEXT X:FOR X=1063 TO 1943 STEP120:POKE X,66:P <208>
OKE X+40,67:POKE X+80,65
20 POKE 54272+X,14:POKE 54312+X,14:POKE 54352+X, <166>
14:NEXT X
21 SYS 8285,62,160,110,160,158,160,206,160,254,1 <118>
60,105,70,160,70,230,70
22 SYS 8430,9,2,0,0,0,0,0,7,7,7:POKE53269,255:PO <192>
KE53271,255:POKE53277,255
23 POKE 53276,255 <188>
24 PRINT"(YELLOW HOME DOWN8 RIGHT10)PARAVISION(R <131>
IGHT3)COMPANY"
25 PRINT"(YELLOW RIGHT16 DOWN2)PRESENTS:" <163>
26 PRINT"(DOWN8 RIGHT4 YELLOW)WRITTEN(SPAC2)IN(S <239>
PACE)1986(SPAC2)BY:(SPAC2)ULRICH(SPAC2)MUEHL"
27 PRINT"(DOWN RIGHT2)COPYRIGHT(SPAC2)1986(SPAC <69>
2)BY(SPAC2)PARAVISION(SPAC2)COMPANY"
28 POKE 53280,0:POKE 56322,224:SYS 832 <56>
30 SYS 8693:LI=4:SC=0 <111>
31 FOR LE=1 TO 9 <8>
32 POKE 54295,0:SYS 832 <43>
33 X=INT(2*RND(1)) <32>
34 IF X=0 THEN FOR X=12880 TO 12903:POKE X-72,PE <17>
EK(X):NEXT
35 IF X=1 THEN FOR X=12928 TO 12951:POKE X-120,P <190>
EEK(X):NEXT
40 POKE 53277,0:POKE 53271,0:8YS8521,32,33,34,35 <197>
,36,33,34,35
41 POKE 53269,0:FOR X=0 TO 11:PRINT"(BROWN RIGHT <67>
8 SD SF SH SF SH SF SH SF SH SF SH SF SH SF
E SH SF SH SF SH SF SH SD)"
42 PRINT"(RIGHT8 SE SG SI SG SI SG SI SG SI SG SI <222>
I SG SI SG SI SG SI SG SI SG SI SE)"
43 NEXT X <245>
44 POKE 53282,12:POKE 53283,1 <93>
46 PRINT"(HOME UP2 LIG.BLUE)": <182>
47 FOR X=0 TO 7:PRINT"(SA SB SC SA SB SC SA SB)" <191>
:PRINT"(SB SC SA SB SC SA SB SC)":PRINT"(SC SA S
B SC SA SB SC SA)"
48 NEXT X:PRINT"(HOME)":FOR X=0 TO 7 <160>
49 PRINTTAB(32)"(SA SB SC SA SB SC SA)":PRINTTAB <43>
(32)"(SB SC SA SB SC SA SB)":PRINTTAB(32)"(SC SA
SB SC SA SB SC)"
50 NEXT X:FOR X=1063 TO 1943 STEP120:POKE X,66:P <240>
OKE X+40,67:POKE X+80,65
51 POKE 54272+X,14:POKE 54312+X,14:POKE 54352+X, <197>
14:NEXT X
52 SYS 8430,0,1,2,3,4,5,6,7,8,0:POKE 5898,0:POKE <149>
5900,0:POKE 5896,1
53 FOR X=1984 TO 2023:POKE X,160:POKE 54272+X,11 <185>
NEXT X
54 ON LE GOSUB 200,220,240,260,280,300,320,340,3 <76>
60
55 SYS8285,150,200,100,30,140,180,200,90,110,120 <8>
,147,150,220,60,115,210 <248>
56 POKE53278,0:FOR WT=0 TO 1000:NEXT WT <32>
57 8YS8909 <207>
58 IF PEEK(8189)=0 THEN GOTO 500 <153>
59 FOR X=450 TO 500:POKE 2040,INT(X/10):SYS 8192 <184>
:SYS 8192:NEXT X
60 FOR X=0 TO 50:SYS 8192:FOR WT=0 TO 10:NEXT WT <173>
:NEXT X:LI=LI-1
61 POKE 53269,1:PRINT"(CLEAR)":IF LI=-1 THEN 100 <51>
0
62 PRINTTAB(7)"(WHITE DOWN8)YOU(SPAC2)HAVE"LI"LI <215>
VES(SPAC2)LEFT(SPAC2)!!!":FORWT=0T01000:NEXTWT:G
OTO32 <71>
199 END <113>
200 POKE 53269,31:RETURN <243>
220 POKE 53269,181:POKE53271,192:RETURN <168>
240 POKE 53269,221:POKE 53271,0:RETURN <164>
260 POKE 53269,255:RETURN <69>
280 POKE 53269,255:POKE53271,192:RETURN <49>
300 POKE 53269,63:POKE53271,200:RETURN <83>
320 POKE 53269,127:POKE53271,26:RETURN <139>
340 POKE 53269,255:POKE53271,18:RETURN <119>
360 POKE 53269,255:POKE53271,1:RETURN <243>
370 END
500 POKE 53269,0:PRINT"(CLEAR DOWN8 RIGHT10 WHIT <215>
E)YOU(SPAC2)MADE(SPAC2)LEVEL"LE"(SPAC2)!!!":POKE
54295,7
510 PRINT"(YELLOW DOWN8 RIGHT8)LEVEL(SPAC2)BONUS <35>
:LE*1000"POINTS":SC=SC+(1000*LE)
520 FOR WT=0 TO 1000:NEXT WT:POKE 54295,0:NEXT L <132>
E
530 FOR X=0 TO 11:PRINT"(BROWN RIGHT8 SD SF SH S <150>
F SH SF SH SF SH SF SH SF SH SF SH SF SH SF
F SH SF SH SD)"
540 PRINT"(RIGHT8 SE SG SI SG SI SG SI SG SI SG <233>
SI SG SI SG SI SG SI SG SI SG SI SE)"
550 NEXT X <90>
560 POKE 53282,12:POKE 53283,1 <188>
570 PRINT"(RVSON WHITE SPACE3)SCORE:000000(SPAC <230>
2)LIVES:5(SPAC2)PARAVISION(SPAC3 HOME BROWN)":
PRINT"(UP2 LIG.BLUE)": <19>
580 POKE 2023,160:POKE 56295,1
590 FOR X=0 TO 7:PRINT"(SA SB SC SA SB SC SA SB) <193>
:PRINT"(SB SC SA SB SC SA SB SC)":PRINT"(SC SA
SB SC SA SB SC SA)"
600 NEXT X:PRINT"(HOME)":FOR X=0 TO 7 <85>
610 PRINTTAB(32)"(SA SB SC SA SB SC SA)":PRINTTA
B(32)"(SB SC SA SB SC SA SB)":PRINTTAB(32)"(SC S <35>
A SB SC SA SB SC)"
620 NEXT X:FOR X=1063 TO 1943 STEP120:POKE X,66: <1>
POKE X+40,67:POKE X+80,65
630 POKE 54272+X,14:POKE 54312+X,14:POKE 54352+X <218>
,14:NEXT X
640 POKE 2040,51:POKE 2041,52:POKE 2042,53:POKE <190>
2043,54:POKE 54295,6
650 POKE 53271,255:POKE 53277,255 <55>
660 SYS 8285,88,0,136,0,184,0,232,0,254,160,105, <37>
70,160,70,230,70
670 POKE 53269,15:POKE 53287,0:POKE 53288,0:POKE <13>
53289,0:POKE 53290,0
675 PRINT"(UP15 WHITE RIGHT11)CONGRATULATIONS!!! <106>
"
680 FOR X=0 TO 220:SYS 8976:SYS 8192:NEXT X <97>
690 POKE 53269,0:FOR WT=0 TO 2000:NEXT WT:SC=SC+ <56>
2000:GOTO 1000
999 END <106>
1000 PRINTTAB(15)"(DOWN8 WHITE)GAME(SPAC2)OVER!" <103>
:POKE 54295,3
1010 POKE 54296,0:POKE 54295,7:SYS 845 <64>
1020 PRINT"(DOWN RIGHT8)YOUR(SPAC2)SCORE(SPAC2)I <250>
S"SC"POINTS":FOR WT=0 TO 2000:NEXT WT:RUN
2000 REM***** <220>
2010 REM* * <66>
2020 REM* 'JOB RACE' * <122>
2030 REM* ----- * <202>
2040 REM* * <96>
2050 REM* WRITTEN IN 1986 BY U. MUEHL * <122>
2060 REM* * <117>
2070 REM* (C)1986 BY PARAVISION * <20>
2080 REM* * <137>
2090 REM* TOTAL LENGTH: 101 BLOCKS * <31>
2100 REM* * <157>
2110 REM* WHAT MEANS 25,25 KBYTE RAM! * <110>
2120 REM* * <177>
2130 REM***** <95>
ENDE DES LISTINGS

```


ASSEMBLERKURS

Teil 21

Nachdem wir uns im letzten Teil mit dem Videochip VIC sowie dem Aufbau und der Handhabung von Sprites beschäftigt haben, möchten wir nun diese Kenntnisse verwerten.

Bevor wir jedoch mit der Programmierung einiger Spritebefehle beginnen, sollten Sie mit dem Spritedesigner aus dem letzten Heft einige Sprites entworfen haben. Wie wir im letzten Teil unseres Kurses erfahren haben, benötigt jedes Sprite genau 64 Bytes an Daten. Die Frage ist nun: Wo legen wir die Daten eines Sprites im Speicher ab?

In der Standardkonfiguration, also ohne Verschiebung des 16k Videobereiches stehen hier nur wenige Bereiche zur Verfügung:

\$2C0 - \$2FE 704 - 766 Platz für Sprite 11
 \$340 - \$37E 832 - 894 Platz für Sprite 13
 \$380 - \$3BE 896 - 958 Platz für Sprite 14
 \$3C0 - \$3FE 960 - 1022 Platz für Sprite 15

Wie aus diesen Angaben zu ersehen ist, lassen sich in der Standardkonfiguration eigentlich nur 4 Spritesblöcke verwenden. Es kommt noch schlimmer! Muß ein Programm von Kassette nachladen, so werden die Spriteblöcke 13 bis 15 zerstört, da diese im sogenannten Kassettenpuffer liegen. Die bisherigen Angaben gelten nur solange wie man in Basic programmiert. Wird ein Programm vollständig in Maschinensprache entwickelt, so steht einem selbstverständlich der gesamte Basic-Speicher von \$800 (2048) für Spritedaten zur Verfügung.

Aber auch in Basic gibt es eine Möglichkeit, mehr Platz für Sprites zu schaffen. Die schon erwähnten Speicherstellen \$2B/\$2C beinhalten nämlich die Adresse des Basicanfangs. Nach dem Einschalten ist hier die Adresse \$800 abgelegt, getrennt nach LOW und HIGH Byte. Um nun mehr Platz für Spritedaten zu schaffen, müssen wir lediglich die Anfangsadresse des Basicspeichers erhöhen. Die maximale Grenze läßt sich leicht bestimmen: 255 Sprites * 64 Bytes = 16320

Wie mache ich ein Sprite sichtbar?

Dank des mageren Commodore Basic sind zahlreiche Poke-Befehle notwendig um ein Sprite sichtbar zu machen. Die folgende Befehlsfolge läßt Sprite 0 an den Koordinaten X=100, Y=100 erscheinen:

```
POKE 2040,13 :REM Block von Sprites 0 festlegen
POKE 53269,1 :REM Sprite 0 einschalten
POKE 53287,2 :REM Farbe von Sprite 0
POKE 53271,0 :REM Keine Y Vergrößerung
POKE 53277,0 :REM Keine X Vergrößerung
POKE 53276,0 :REM Sprites einfarbig
POKE 53248,100 :REM X-Koordinate von Sprite 0
POKE 53249,100 :REM Y-Koordinate von Sprite 0
```

```
2
20: C000 .OPT 00,P1
30: C000 *= $C000
100: C000 TEMP = 704 ;ZWISCHENSPEICHER
115: C000 CKCOM = $AEFD ;AUF KOMMA PRUEFEN
120: C000 GETBYT = $B79E ;ZAHN HOLEN
215: C000 VIC = $D000 ;BASISADRESSE VIDEOCHIP
;
1000: C000 20 FD AE SPREIN JSR CKCOM ;SPRITE EINSCHALTEN
1005: C003 20 9E B7 JSR GETBYT ;NUMMER DES SPRITES HOLEN (0-7)
1010: C006 BD 34 C0 LDA TEIN,X ;EINSCHALTBIT LADEN
1015: C009 0D 15 D0 ORA VIC+21
1020: C00C BD 15 D0 STA VIC+21 ;SPRITE EINSCHALTEN
1025: C00F 60 RTS
;
5100: C010 20 FD AE SPRAUS JSR CKCOM ;SPRITE AUSSCHALTEN
```

Soviel zu den theoretischen Möglichkeiten. Wir wollen uns jedoch in unseren ersten Experimenten auf die wenigen Spriteblöcke beschränken, welche in der Standard Konfiguration zur Verfügung stehen. Das folgende Listing enthält die Einleseschleife sowie die Daten für drei einfache Sprites. Wer eigene Sprites entworfen hat kann diese ebenfalls für unsere Experimente verwenden, sollte jedoch die Einleseschleife der Zeilen 50000-50050 mit übernehmen.

```
50000 REM **** SPRITES EINLESEN **** <139>
50010 FOR I= 832 TO 894:READ A:POKE I,A:NEXT I <59>
50020 FOR I= 896 TO 958:READ A:POKE I,A:NEXT I <135>
50030 FOR I= 960 TO 1022:READ A:POKE I,A:NEXT I <109>
50040 POKE 2040,13:REM SPRITE 0 BLOCK 13 <150>
50041 POKE 2041,14:REM SPRITE 1 BLOCK 14 <205>
50042 POKE 2042,15:REM SPRITE 2 BLOCK 15 <4>
50050 END <197>
60000 REM DREIECK IM KASTEN <53>
60010 DATA255,255,255,128,0,1,128,0,1,128,0,1,12
8,0,1,128 <60>
60020 DATA0,1,128,24,1,128,36,1,128,66,1,128,129
,1,129,0 <217>
60030 DATA129,130,0,65,132,0,33,143,255,241,128,
0,1,128,0,1,128 <238>
60040 DATA0,1,128,0,1,128,0,1,128,0,1,255,255,25
5 <218>
60050 REM FASS <69>
60060 DATA0,0,0,0,0,0,0,0,0,0,60,0,0,66,0,0 <55>
60070 DATA129,0,1,255,128,1,0,128,3,255,192,2,0,
64,2,0 <16>
60080 DATA64,3,255,192,1,0,128,1,255,128,0,129,0
,0,66,0,0 <79>
60090 DATA60,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <83>
60100 REM KREUZ <162>
60110 DATA128,0,8,64,0,16,32,0,32,16,0,64,8,0,12
8,4 <145>
60120 DATA1,0,2,2,0,1,4,0,0,136,0,0,80,0,0,32 <135>
60130 DATA0,0,80,0,0,136,0,1,4,0,2,2,0,4,1,0,8 <52>
60140 DATA0,128,16,0,64,32,0,32,64,0,16,128,0,8 <27>
ENDE DES LISTINGS
```

Wie man sieht, ist es recht umständlich ein Sprite auf dem Schirm sichtbar zu machen. Noch wesentlich komplizierter wird die Technik, wenn mehrere Sprites in verschiedenen Farben und Prioritäten auf dem Bildschirm bewegt werden sollen. Um dieses Problem zu beseitigen, ist es Ziel unseres Assemblerkurses eine Basic-Erweiterung zu entwickeln, welche uns die Steuerung von Sprites vereinfacht.

Wir wissen inzwischen, wie Parameter an Maschinenprogramme übergeben werden können. Wir wissen auch welche Register zur Darstellung von Sprites manipuliert werden müssen. Das nachfolgende Assemblerlisting nutzt unsere bisherigen Kenntnisse für eine kleiner Befehls-Erweiterung.

Das obige Assemblerlisting stellt drei kleine Routinen zur Verfügung: Sprite einschalten, Sprite ausschalten und Spritefarbe setzen

Die Befehle werden jeweils mit einem SYS-Befehl und der entsprechenden Anzahl von Parameter aufgerufen:

```
SYS 49152,0 :REM Sprite 0 einschalten
SYS 49152,1 :REM Sprite 1 einschalten
SYS 49168,0 :REM Sprite 0 ausschalten
SYS 49168,7 :REM Sprite 7 ausschalten
```


programme

```
5105: C013 20 9E B7      JSR GETBYT ;NUMMER DES SPRITES HOLEN (0-7)
5110: C016 BD 3C C0      LDA TAUS,X ;MASKE LADEN
5115: C019 2D 15 D0      AND VIC+21
5120: C01C 8D 15 D0      STA VIC+21 ;SPRITE AUSSCHALTEN
5125: C01F 60              RTS

5200: C020 20 FD AE SETCOLOR JSR CKCOM ;SPRITEFARBE SETZEN
5205: C023 20 9E B7      JSR GETBYT ;NUMMER DES SPRITES HOLEN (0-7)
5210: C026 8E C0 02      STX TEMP ;MERKE NUMMER
5215: C029 20 FD AE      JSR CKCOM ;HOLE FARBE
5220: C02C 8A              TXA
5225: C02D AE C0 02      LDX TEMP
5230: C030 9D 27 D0      STA 532B7,X
5235: C033 60              RTS

10000: C034 01 02 04 TEIN .BYTE1,2,4,8,16,32,64,128
10001: C03C FE FD FB TAUS .BYTE254,253,251,247,239,223,191,127
```

SYS 49184,0,2 :REM Sprite 0 erhält die Farbe 2 (rot)
SYS 49184,1,0 :REM Sprite 1 erhält die Farbe 0 (schwarz)

Die Arbeitsweise und Funktion unseres Maschinenprogrammes erklärt sich praktisch von selbst. Wir könnten die Anzahl der Befehlsroutinen fast unbegrenzt erweitern. Dies wäre praktisch die einfachste Form einer sogenannten Basicerweiterung. Da jedoch durch die zahlreichen unter-

schiedlichen SYS-Aufrufe die Übersichtlichkeit nicht viel verbessert wird, gehen wir einen anderen Weg.

Ziel ist es eine Befehlsweiterung zu programmieren, welche ohne Sys-Aufrufe auskommt. Wie dies möglich ist erfahren wir im nächsten Heft.

Frank Brall

Literatur:

6502 Microcomputer-Programmierung, Peter Heuer, Hofacker-Verlag

64 Intern, Angershausen, Becker, Englisch, Gerits, Data Becker Buch

C64 Programmieren in Maschinensprache, Winfried Kassera, Frank Kassera, Markt & Technik

Profihandbuch, H.L. Schneider, Werner Eberl, Markt & Technik

Hot-line:

Bei Fragen zu den Listings steht Ihnen unsere Programmierabteilung für telefonische Anfragen zur Verfügung. Bitte wählen Sie nur diese Nummer und rufen Sie nur zu den angegebenen Zeiten an: Montag-Freitag, 14-16 Uhr!

0 56 51/3 00 13

Bewerbung als Programmautor

Name: _____ Vorname: _____ Alter: _____

Straße: _____ Wohnort: _____

Titel des Programmes: _____ Computersystem: _____

Erforderlicher Speicherplatz: _____ KBytes

Erforderliche Peripherie: _____

Ich versichere hiermit, daß ich der Autor des oben genannten Programmes bin und alle Rechte besitze. Ich bin damit einverstanden, daß Sie mein Programm in einer Zeitschrift abdrucken und/oder dieses verkaufen. Das einmalige Honorar beträgt 120 DM pro abgedruckter Seite. Sollte das Programm nicht veröffentlicht werden, bitte ich um Rücksendung meiner Unterlagen. Sollte mein Programm in der Zwischenzeit von anderer Seite erworben worden sein, werde ich Sie unverzüglich benachrichtigen. Ich bestätige durch meine Unterschrift, daß ich Kenntnis davon habe, daß der Tronic-Verlag gegen Personen, die Programme einschicken, deren Urheberrechte sie nicht besitzen, rechtliche Schritte einleitet.

Unterschrift des Programmautors (bei Minderjährigen auch die eines Erziehungsberechtigten): _____

Folgende Unterlagen liegen diesem Schreiben bei:

Kassette: _____ Diskette: _____ Dokumentation: _____ Listing: _____ Rückporto: _____

Unterschrift des Programmautors (bei Minderjährigen auch die eines Erziehungsberechtigten): _____

(Bitte zweimal unterschreiben!)

Einsenden an: Tronic-Verlag Abt. Programmentwicklung, Am Stad, 3440 Eschwege

Interruptprogrammierung auf dem Commodore 64

Artikelserie von Daniel Durstewitz und Waldemar Raaz

Teil 3

Im letzten Heft haben wir begonnen die Auswertung der Interruptimpulse zu behandeln. In dieser Ausgabe fahren wir mit diesem Kapitel fort:

Im einzelnen sieht eine Erweiterung der NMI-Routine im Betriebssystem also folgendermaßen aus:

1. Man erstellt ein Erweiterungsprogramm, das nach erfolgtem NMI die Interruptquellen überprüft und entscheidet, ob auf den Interrupt reagiert werden soll. An seinen Ausgängen sollte das Programm mit einem JMP \$FE47 (NMI-Routine im ROM) oder mit RTI (Return From Interrupt) enden.

2. Man bringt das Erweiterungsprogramm in den Speicher.

3. Man legt in der Speicherzelle 792 (hex. \$0318) das Low-Byte der Startadresse des Erweiterungsprogramms ab und in 793 (hex. \$0319) entsprechend das High-Byte.

Nun verzweigt das Betriebssystem bei einem NMI zuerst in die eigene Routine, in der unter Umständen entsprechend reagiert wird, und erst dann in die »echte« NMI-Routine.

Demoprogramm-Nr.1

Dies alles soll nun anhand eines Demoprogrammes einmal in der Praxis vorgeführt werden. Geben Sie dazu das Programmlisting 1 ein. Sie können dabei alle REM-Anweisungen sowie die Kommentare nach dem Doppelpunkt in den Data-Zeilen weglassen. Sie sind für den Programmablauf irrelevant und dienen im Listing nur der Erläuterung und Über-

sichtlichkeit. In die Data-Zeilen wurde ein vollständig kommentiertes Assemblerlisting integriert, das dem Leser es erleichtern soll, das Programm nachzuvollziehen. Speichern Sie das Programm vor dem Starten unbedingt ab, damit es durch falschen Programmablauf nicht versehentlich zerstört wird, und Sie ohne Sicherheitskopie dastehen. Wenn Sie das Programm dann mit RUN starten, werden die Daten für das Maschinenprogramm eingelesen; haben Sie einen Fehler begangen, so teilt Ihnen das Programm dieses mit, und Sie sollten die Data-Zeilen noch einmal überprüfen. Ansonsten erscheint auf dem Bildschirm noch einmal eine Kurzerklärung der Routine, und Sie werden aufgefordert, eine Taste zu drücken. Ist dies erfolgt, so werden Alarm- und Uhrzeit auf vordefinierte Werte eingestellt: die Alarmzeit auf genau 8 Uhr und die Uhrzeit auf 10 Sekunden vor acht. Dies geschieht in den Programmzeilen 245 bis 260. Zunächst wird Bit 7 im Register 15 der CIA #2 (Basissadresse 56576 - hex. \$DD00) gelöscht, was der CIA signalisiert, daß jetzt die Zeit gesetzt werden soll. Danach werden die Werte für die Uhrzeit in folgende Register geschrieben:

Stunden	- Register 11
Minuten	- Register 10
Sekunden	- Register 9
Zehntelsekunden	- Register 8.

Alle Register sind im BCD-Format organisiert, was beim Beschreiben der Register unbedingt beachtet werden muß. Bit 7 des Stundenregisters 11 dient zusätzlich als AM/PM-Flagge: 0 = vormittags (AM), 1 = nachmittags (PM). Um nun die Alarmzeit einzustellen, wird Bit 7 im Register 15 gesetzt (Zeile 255), und in die-

selben Register wie oben wird diesmal die Alarmzeit geschrieben. Alarm- und Uhrzeit belegen dasselbe Register, weshalb auch extra ein Schaltbild (Bit 7 im Register 15) benötigt wird. In Zeile 265 wird noch Bit 7 im Register 14 gesetzt, um sicherzustellen, daß die Echtzeituhr auch wirklich mit 50 Hz getaktet wird und richtig geht, sowie die Interruptroutine initialisiert. Dazu wird die Startadresse der »erweiterten« NMI-Routine (NMINEU) in die Speicherzellen \$0318/\$0319 geschrieben und Bit 2 im INT MASK-Register gesetzt, um den Alarm-NMI freizugeben (Zeilen 280 - 325). Die eigentliche NMI-Routine testet zu Beginn Bit 2 im INT DATA-Register. Ist es nicht gesetzt, so ist eine andere Interruptquelle für den NMI verantwortlich, und es wird der NMI-Routine im Betriebssystem übergeben. Ist es gesetzt, so stimmen Alarm- und Uhrzeit überein, und es wird ein pulsierender Piepston erzeugt, um den Alarm anzuzeigen. Hier kann man ganz nach Belieben natürlich auch eine optische Alarmanzeige programmieren, oder optisch und akustisch gemischt. Gleichzeitig wird aber auch noch die Tastatur kontrolliert, so daß ein Tastendruck den Alarm verstummen läßt. Besonders zu beachten sind zwei Programmzeilen: zum einen Zeile 395, das CLI in dieser Zeile ist absolut notwendig, da das Programm sonst beim Eintreten eines Alarms abstürzen würde. Und zwar gibt CLI den Systeminterrupt wieder frei, der vom Rechner bei jedem NMI gesperrt wird.

Ohne Systeminterrupt läuft beim C64 aber sehr wenig bis gar nichts, so daß er unbedingt wieder freigegeben werden muß, um beispielsweise rie nachfolgende Tastaturabfrage zu ermöglichen.

Die andere wichtige Zeile ist 530: die Piepston-Routine schließt mit einem RTI und nicht - wie üblich - mit einem RTS ab. Dies ist ebenfalls unbedingt notwendig, da die Routine quasi per Interrupt aufgerufen wurde. Wie schon oben erwähnt, werden bei jedem Interrupt Programmzähler und Prozessorstatus auf den Stack gerettet. Dort müssen Sie nach Ablauf der Interruptroutine natürlich auch wieder abgeholt werden, wofür der Befehl RTI sorgt.

4. Der IRQ

Die vierte Art von Interrupt ist schließlich der IRQ (Interrupt ReQuest), und es ist auch die nützlichste und vielseitigste. Im Gegensatz zum NMI und Reset ist der IRQ maskierbar, d.h. man kann ihn softwaremäßig verbieten, nämlich indem man im Prozessorstatusregister das Interruptflag auf 1 setzt, was mit dem Assemblerbefehl SEI (Set Interruptflag) möglich ist. Der Befehl CLI (Clear Interrupt) setzt das Interruptflag auf 0 zurück, erlaubt also den IRQ wieder. Die Interruptquellen des IRQ sind jedoch teilweise denen des NMI ähnlich. Erst einmal wieder liegt die -IRQ-Leitung am Expansion Port am Pin 4 an und braucht nur mit Masse (GND) kurzgeschlossen zu werden, um einen IRQ auszulösen, was wiederum eher für externe Baugruppen gedacht ist und uns hier nicht weiter interessieren soll.

Weiterhin kann man über die entsprechende Programmierung des ICR in der CIA #1 einen IRQ auslösen; und zwar entspricht das ICR in der CIA #1 genau dem in der CIA #2, nur daß es eben statt eines NMI einen IRQ auslöst. Entsprechend gilt natürlich das unter NMI-Programmierung über das ICR in der CIA #2

Gesagte auch für die IRQ - Programmierung mit Hilfe des ICR in der CIA #1 und soll darum an dieser Stelle nicht noch einmal wiederholt werden. Das Besondere am IRQ ist jedoch, daß er vom Betriebssystem aktiv genutzt wird.

Der vom Timer ausgelöste IRQ
 Dazu benutzt das Betriebssystem Bit 0 des ICR in der CIA #1, wodurch der IRQ jedesmal bei einem Unterlauf von Timer A ausgelöst wird. Effektiv heißt das, daß 60 mal in der Sekunde (also alle 16 Millisekunden) das laufende Programm in seiner Ausführung unterbrochen wird, ein IRQ ausgelöst und eine Routine im ROM mit der Adresse \$ EA31 (dez. 59953) angesprungen wird. Die dabei angesprungene IRQ-Routine nutzt das Betriebssystem für verschiedene »Wartungsaufgaben«. So wird z.B. die interne Uhr weitergestellt, die Tastatur abgefragt und auch das Blinken des Cursors wird hier erzeugt. Es liegt natürlich auf der Hand, daß eine Interruptroutine, die 60 mal in der Sekunde durchlaufen wird, sich für eigene Erweiterungen gerade so aufdrängt, da sich Basicprogramme (aber auch Maschinenprogramme) durch solche Maschinensprache-Erweiterungen enorm beschleunigen lassen (gerade Spiele: zum Beispiel eine Routine, die die Joystickeingänge abfragt und entsprechend die Sprites kontrolliert). Auch läßt sich auf diese Weise quasi ein Multitasking auf dem C 64 verwirklichen. Multitasking ist ein Begriff, der aus dem Feld der Computer-Großanlagen stammt, wo ein Computer scheinbar in der Lage ist, mehrere Programme auf einmal zu bearbeiten bzw. mehrere Steuerungs- oder Überwachungsaufgaben parallel. In Wirklichkeit ist dies so realisiert, daß der Computer eine kurze Zeit (Sekundenbruchteile) an einem Programm arbeitet, dann alle Variablen abspeichert und zum nächsten Programm weitergeht. Ist er dann wieder beim ersten Programm angelangt, holt er die Variablen aus dem Arbeitsspeicher zurück und fährt eine kurze Zeit lang mit der Bearbeitung fort. Dies alles regelt der Computer so schnell, daß ein Benutzer von all dem nichts bemerkt und für ihn alles gleichzeitig zu laufen scheint. Einen ähnlichen Effekt erreicht

man eben über die IRQ - Programmierung auf dem C 64; denn einmal an die IRQ - Routine angehängt, wird eine eigene Maschinenroutine unabhängig von anderen gerade laufenden Basic - oder Maschinenprogrammen 60 mal in der Sekunde abgearbeitet. Eine nette Idee wäre da doch eine IRQ-Routine, die eine über den Userport angeschlossene Alarmanlage kontrolliert und bei Störung Alarm schlägt. So könnte das Wachpersonal währenddessen mit Telespielen unterhalten werden; und beides lief »gleichzeitig« auf unserem C 64. Eine andere bekannte Anwendung, die diese Möglichkeit ausnutzt, ist z.B. eine Interrupt-Uhr. Es dürfte eigentlich keine Zeitschrift geben, in der C 64-Programme veröffentlicht werden und noch keine Interrupt - Uhr für diesen Computer abgedruckt wurde, so daß wir hier auf ein entsprechendes Beispiel-Programm verzichten wollen. Nur in Kürze zum Funktionsprinzip eines solchen Programmes: Die IRQ - Einsprung - Adresse, die sich (auch im RAM) über die beiden Speicherzellen \$ 0314 (dez. 788, LO - Byte der Adresse) und \$ 0315 (dez. 789, HI - Byte der Adresse) festsetzen läßt, wird einfach auf eine eigene Routine gestellt, die dann im Interrupt zunächst die aktuelle Zeit aus den RAM - Adressen \$ A0 (dez. 160), \$ A1 (dez. 161) und \$ A2 (dez. 162) ausliest. Diese Adressen enthalten den Wert der Zeit als 24 - Bit - Binärzahl in 1/60 - Sekunden, wobei \$ A0 die niederwertigen acht Bits dieser Binärzahl enthält. Die Zeit kann nicht über die Betriebssystem - Routine RDTIM (Read Time) ausgelesen werden, da diese mit dem Befehl CLI endet, was innerhalb unser IRQ-Routine einen Absturz des Computer verursachen würde. Die Umwandlung der 24-Bit-Binärzahl in Bildschirmcodes kann nun entweder über eigene Routinen oder unter Ausnutzung vorhandener ROM - Routinen geschehen. Falls Sie ROM - Routinen verwenden wollen, finden sie eine zur Umwandlung geeignete Befehlsfolge in den ROM - Adressen \$ AF48 bis \$ AF58, die die entsprechenden Bildschirmcodes ab Adresse \$ 100 (dez. 256) ablegt und die Tabelle mit dem Code # \$00 abschließt. Allerdings muß der erste Befehl von JSR \$ AF84; in JSR \$ AF87 geändert werden. Auch sollte vor

der Abarbeitung dieser Umwandlungsroutine die gesamte Zero - Page (RAM - Speicherzellen \$ 00 bis \$ FF) in einen Zwischenspeicher gerettet und hinterher zurückgeholt werden. Die Bildschirmcode-Tabelle braucht jetzt nur noch in die rechte obere Bildschirmcke übertragen zu werden. Das Programm muß dann mit einem Sprung zur ursprünglichen IRQ - Routine mit der Adresse \$ EA31 abgeschlossen werden, sonst würde sich Ihr Computer bis auf weiteres leider verabschieden, da ja z.B. die Tastatur nicht mehr abgefragt würde. Ganz so einfach ist das Verändern der IRQ-Einsprung-Adresse übrigens auch nicht: dazu muß nämlich erst einmal der Interrupt ausgeschaltet sein, denn es kann ja niemals das Lo-Byte der Adresse in \$ 0314 genau gleichzeitig mit dem Hi-Byte in \$ 0315 geändert werden und sobald ein Wert geändert wäre, würde der Computer beim folgenden IRQ mit Sicherheit abstürzen, da dieser bestimmt ausgelöst wird, bevor auch der zweite Wert geändert ist. Dies läßt sich also nur mit einem kleinen Maschinenprogramm regeln, daß zunächst mit dem Assemblerbefehl SEI jede IRQ-Möglichkeit ausschließt. Ein solches Maschinenprogramm wird u.a. auch in Listing 2 vorgestellt. Damit wären wir auch gleich bei unserem ersten Beispiel-Programm zum IRQ, das während des Interrupts eine Laufschrift in der ersten Zeile erzeugt. Neben der Einschalt-Routine ist der neuen IRQ-Routine noch eine Ausschalt-Routine vorangestellt, so daß außer mit der Tastenkombination STOP/RESTORE auch innerhalb eines Programmes wieder auf die ROM-IRQ-Routine zurückgeschaltet werden kann. Die eigentliche IRQ-Routine muß natürlich nicht mehr mit dem Befehl SEI beginnen, da das Interrupt-Flag durch die Auslösung des IRQ bereits gesetzt wird. Im übrigen ist ein erneuter IRQ noch während der Abarbeitung einer IRQ-Routine durchaus denkbar (bei den im nächsten Kapitel vorgestellten Beispiel-Programmen ist diese Maßnahme, wie Sie sehen werden, sogar erforderlich).

Das Interrupt Request Register und das Interrupt Mask Register
 Es gibt noch vier weitere Möglichkeiten, einen IRQ auszulösen, und damit wird es erst richtig in

teressant. Diese weiteren vier Möglichkeiten sind durch das Register 26 im VIC gegeben, es ist das sogenannte Interrupt Mask Register (IMR) mit der Adresse \$ D01A (dez. 53274). Hier kann genau wie im ICR der CIA #1 auch, ein gesetztes Bit eine Interrupt-Möglichkeit bestimmen. Allerdings werden hier nicht wie im ICR durch Schreib- und Lesezugriff zwei verschiedene Register angesprochen. Um erfahren zu können, wodurch ein Interrupt ausgelöst wurde, gibt es hier nämlich ein zweites Register, das Interrupt Request Register (IRR) mit der Adresse \$ D019 (dez. 53273, Register 25 des VIC). Dieses Register entspricht also in seiner Funktion dem INT DATA-Register beim Lesezugriff auf das ICR. Die Bitbelegung des IRR und des IMR ist genau die gleiche und soll im folgenden vorgestellt werden:
 Bit 0: 1 = Rasterzeilen-IRQ, ausgelöst durch Register 18 des VIC
 Bit 1: 1 = Sprite-Hintergrund-Kollision, IRQ-Auslöser: Register 31
 Bit 2: 1 = Sprite-Sprite-Kollision, IRQ-Auslöser: Register 30
 Bit 3: 1 = Impuls vom Lightpen oder auch Feuerknopf eines Joysticks an Pin 6 des Control Port 1
 Bit 4 - Bit 6: unbenutzt
 Bit 7: 1 = mindestens eins der ersten vier Bits ist gesetzt.
 Wird im IMR ein Bit gesetzt und durch ein entsprechendes Ereignis auch das korrespondierende Bit im IRR, so wird ein IRQ ausgelöst. Zu beachten ist, daß das IRR nicht wie das ICR durch einfaches Laden des Registerinhaltes gelöscht wird, sondern extra gelöscht werden muß (sonst folgt dauernd ein Interrupt, was einem Absturz des Computers gleichkommt). Gelöscht wird das IRR dadurch, daß derselbe Wert, der aus dem Register herausgelesen wurde, auch in selbiges wieder hineingeschrieben wird (Sie können das bei den Beispielprogrammen Listing 3 bis Listing 6 sehen). Die Möglichkeit bei einer Sprite-Sprite- oder Sprite-Hintergrund-Kollision einen IRQ auszulösen, hat den Vorteil, daß das betreffende Ereignis sofort behandelt werden kann, ohne daß erst andere Programmteile bis zu einer entsprechenden Abfrage durchlaufen werden müssen. Der gleiche Vorteil bietet sich auch beim Lightpen, so daß z.B. ein von erwähntem Gerät angesprochener Bild-

schirmpunkt sofort gesetzt werden kann. Ein Impuls an Pin 6 des Control Port 1, d.h. ein Kurzschluß zwischen Pin 6 (Button A/LP) und Pin 8 (GND = Masse), wie er auch vom Lightpen bewirkt wird, kann auch durch

Drücken der Feuerknopfes eines Joysticks oder Paddles erzeugt werden, was besonders für Telespiele interessant ist. So könnte durch das Drücken des Feuerknopfes ein IRQ ausgelöst werden, in dem z.B. der Abschluß

einer Rakete organisiert wird, die dann während des IRQ parallel zum laufenden Hauptprogramm gesteuert wird. Aus diesen Möglichkeiten können auch größere Geschwindigkeitsvorteile resultieren, da ja sich

dauernd wiederholende Abfragen im Programm entfallen und zwei Dinge quasi parallel zueinander ablaufen können.

Fortsetzung folgt!

Demolisting 1

C-64 LISTING + CHECKSUMMEN (C V1.0)		
100 REM * ALARMUHR AUF INTERRUPTBASIS *	<206>	
105 REM	<248>	
110 REM DIESES PROGRAMM DEMONSTRIERT DIE PROGRA		
MMIERUNG EINER ALARMUHR UNTER	<52>	
115 REM AUSNUTZUNG DER ECHTZEIT- UND ALARMUHR		
IN DER CIA #2 UND DEM ZUGE-	<13>	
120 REM HOERIGEN INTERRUPTBIT IM ICR, DAS BEI U		
EBEREINSTIMMUNG DER ALARMZEIT	<119>	
125 REM MIT DER UHRZEIT EINEN NMI AUSLOEST.	<60>	
130 :	<188>	
135 :	<193>	
140 FORN=49152T049252:READA\$:A\$=LEFT\$(A\$,2)	<164>	
145 A1=ASC(LEFT\$(A\$,1)):A2=ASC(RIGHT\$(A\$,1))	<23>	
150 A1=A1-48:IFA1>9THENA1=A1-7	<38>	
155 A2=A2-48:IFA2>9THENA2=A2-7	<99>	
160 A=16*A1+A2:Z=Z+A:POKEN,A:NEXT	<217>	
165 IFZ<>14211THENPRINT"(DOWN)FEHLER{SPACE}IN{SPA	<177>	
CE}DEN{SPACE}DATA-ZEILEN!":END	<228>	
170 :	<233>	
175 :		
180 PRINT"(CLEAR DOWN SPACE)ALARMUHR":PRINT "{SPA	<96>	
CE}=====":PRINT		
185 PRINT "{SPACE}DIESES {SPACE}PROGRAMM {SPACE2}SI	<246>	
MULIERT {SPACE}EINE {SPACE}ALARM-{SPACE}"		
190 PRINT "{SPACE}UHR {SPACE}AUF {SPACE}GENAU {SPACE2}	<102>	
CE}MMODDRE {SPACE}64 {SPACE}UNTER {SPACE}BENUT-{SPAC	<195>	
E}"		
195 PRINT "{SPACE}ZUNG {SPACE}DER {SPACE}ECHTZEITUHR	<255>	
R {SPACE}IN {SPACE}DER {SPACE}CIA {SPACE}#2 {SPACE}UN	<52>	
D {SPACE}"	<76>	
200 PRINT "{SPACE}VERWENDUNG {SPACE2}DER {SPACE}ENT	<84>	
SPRECHENDEN {SPACE2} INTER-{SPACE}"	<167>	
205 PRINT "{SPACE}RUPTMOEGLICHKEITEN. {SPACE3}DIE {	<228>	
SPACE}UHRZEIT {SPACE}WIRD {SPACE}"	<39>	
210 PRINT "{SPACE}AUF {SPACE}7:59:50:0, {SPACE}DIE {	<227>	
SPACE}ALARMZEIT {SPACE}AUF {SPACE}GENAU {SPACE}"	<229>	
215 PRINT "{SPACE}8:00:00:0 {SPACE}EINGESTELLT. {SP	<138>	
ACE}SOBALD {SPACE}SIE {SPACE}EINE {SPACE}"	<35>	
220 PRINT "{SPACE}TASTE {SPACE}DRUECKEN, {SPACE}LAE	<88>	
UFT {SPACE}DIE {SPACE}ZEIT. {SPACE}BITTE {SPACE}"	<177>	
225 PRINT "{SPACE}STOPPEN {SPACE}SIE {SPACE}DIE {SPA	<76>	
CE2}10 {SPACE}SEC. {SPACE}BIS {SPACE}ZUM {SPACE}ALAR	<73>	
M {SPACE}"	<78>	
230 PRINT "{SPACE}RUHIG {SPACE}MIT! {SPACE2}<TASTE>		
"		
235 GETG\$:IFG\$=""THEN235		
240 B=56576 : REM BASISADRESSE DER CIA #2		
245 POKEB+15,PEEK(B+15)AND127:REM ZEIT SETZEN =>		
7:59:50:0		
250 POKEB+11,7:POKEB+10,5*16+9:POKEB+9,5*16:POKE		
B+8,0		
255 POKEB+15,PEEK(B+15)OR128 :REM ALARM SETZEN=>		
8:00:00:0		
260 POKEB+11,8:POKEB+10,0:POKEB+9,0:POKEB+8,0		
265 POKEB+14,PEEK(B+14)OR128:SYS49152:END		
270 :		
275 :		
280 DATA A2,20 : START LDX #<NMI NEU "{SPACE	<166>	
}NEUE {SPACE}NMI-ADRESSE {SPACE}LO-BYTE	<157>	
285 DATA A0,C0 : LDY #<NMI NEU "{SPACE		
}NEUE {SPACE}NMI-ADRESSE {SPACE}HI-BYTE		
290 DATA A9,F4 : LDA #F4 "{SPACE	<97>	
}MASKE {SPACE}FUER {SPACE}ALARM-INTERRUPT {SPACE}FR	<91>	
EI		
295 DATA D0,06 : BNE \$SETZEN "{SPACE	<205>	
}--->{SPACE}IMMER	<216>	
300 DATA A2,47 : LDX #47 "{SPACE		
}ALTE {SPACE}NMI-ADRESSE {SPACE}LO-BYTE	<216>	
305 DATA A0,FE : LDY #FE "{SPACE		
}ALTE {SPACE}NMI-ADRESSE {SPACE}HI-BYTE		
310 DATA A9,04 : LDA #04 "{SPACE		
}MASKE {SPACE}FUER {SPACE}ALARM-INTERRUPT {SPACE}SP		
ERREN	<67>	
315 DATA BE,18,03 : SETZEN STX \$0318 "{SPACE		
}NMI-VEKTOR {SPACE}SETZEN {SPACE}LO-BYTE	<155>	
320 DATA BC,19,03 : STY \$0319 "{SPACE		
}HI-BYTE	<79>	
325 DATA BD,0D,DD : STA \$DD0D "{SPACE		
}ICR {SPACE}IN {SPACE}DER {SPACE}CIA#2 {SPACE}SETZEN	<39>	
330 DATA 60 : RTS "{SPACE		
}ZURUECK {SPACE}INS {SPACE}BASIC	<210>	
335 DATA EA : NOP "	<168>	
340 DATA EA : NOP "	<173>	
345 DATA EA : NOP "	<178>	
350 DATA EA : NOP "	<183>	
355 DATA EA : NOP "	<188>	
360 DATA EA : NOP "	<193>	
365 DATA EA : NOP "	<198>	
370 DATA EA : NOP "	<203>	
375 DATA AD,0D,DD : NMI NEU LDA \$DD0D "{SPACE		
}ICR {SPACE}IN {SPACE}DER {SPACE}CIA {SPACE}#2 {SPACE	<45>	
}LESEN		
380 DATA 29,04 : AND #04 "{SPACE		
}BIT {SPACE}2 {SPACE} (UHRZEIT=ALARMZEIT) {SPACE}GES	<77>	
ETZT {SPACE}?		
385 DATA D0,03 : BNE \$ALARM "{SPACE	<120>	
}JA, {SPACE}DANN {SPACE}ALARMTON {SPACE}ERZEUGEN		
390 DATA 4C,47,FE : JMP \$FE47 "{SPACE	<172>	
}NEIN, {SPACE}DANN {SPACE}ZUR {SPACE}NORMALEN {SPACE		
}NMI-ROUTINE		
395 DATA 58 : ALARM CLI "{SPACE	<58>	
}INTERRUPT {SPACE}WIEDER {SPACE}FREIGEBEN		
400 DATA A9,0F : LDA #0F "{SPACE	<54>	
}SID-REGISTER {SPACE}FUER {SPACE}ALARM {SPACE}VORBE		
REITEN:		
405 DATA BD,18,D4 : STA \$D418 "{SPACE	<189>	
}LAUTSTAERKE	<46>	
410 DATA A9,00 : LDA #00 "		
415 DATA BD,0E,D4 : STA \$D40E "{SPACE	<78>	
}FREQUENZ {SPACE}LO-BYTE	<46>	
420 DATA A9,CB : LDA #CB "		
425 DATA BD,0F,D4 : STA \$D40F "{SPACE	<62>	
}FREQUENZ {SPACE}HI-BYTE	<166>	
430 DATA A9,05 : LDA #05 "		
435 DATA BD,13,D4 : STA \$D413 "{SPACE	<204>	
}ANSCHLAG/ABSCHWELLEN	<20>	
440 DATA A9,F3 : LDA #F3 "		
445 DATA BD,14,D4 : STA \$D414 "{SPACE	<7>	
}HALTEN/AUSKLINGEN	<104>	
450 DATA A9,10 : LDA #10 "		
455 DATA BD,12,D4 : STA \$D412 "{SPACE	<31>	
}TON {SPACE}AUSSCHALTEN		
460 DATA CA : LOOP1 DEX "{SPACE	<129>	
}ZEITSCHLEIFE	<156>	
465 DATA D0,FD : BNE \$LOOP1 "	<203>	
470 DATA 88 : DEY "	<148>	
475 DATA D0,FA : BNE \$LOOP1 "	<154>	
480 DATA A9,11 : LDA #11 "		
485 DATA BD,12,D4 : STA \$D412 "{SPACE	<226>	
}TON {SPACE}EINSCHALTEN		
490 DATA CA : LOOP2 DEX "{SPACE	<168>	
}ZEITSCHLEIFE	<202>	
495 DATA D0,FD : BNE \$LOOP2 "	<233>	
500 DATA 88 : DEY "	<194>	
505 DATA D0,FA : BNE \$LOOP2 "		
510 DATA 20,E4,FF : LEER JSR \$FFE4 "{SPACE	<154>	
}--->{SPACE}TASTATURPUFFER {SPACE}ABFRAGEN		
515 DATA F0,E5 : BEQ \$LEER "{SPACE	<235>	
}WENN {SPACE}LEER, {SPACE}DANN {SPACE}WEITER {SPACE}		
ABFRAGEN		
520 DATA A9,00 : LDA #00 "{SPACE	<221>	
}AUF {SPACE}TASTENDRUCK {SPACE}LAUTSTAERKE {SPACE}A		
UF		
525 DATA BD,18,D4 : STA \$D418 "{SPACE	<165>	
}NULL {SPACE}STELLEN	<24>	
530 DATA 40 : RTI "	<83>	
535 :		
540 REM * WRITTEN 1986 BY WALDEMAR RAAZ *	<250>	
ENDE DES LISTINGS		

The Hero

Schneider CPC 464/664/6128

Ein Actionspiel mit toller Grafik und phantastischen Sprites. Wir schreiben das Jahr 3005. Die HYDRA, ein riesiges Forschungsraumschiff, ist seit 5 Jahren unterwegs, um bislang unbekannte Sektoren der Galaxis zu erforschen. Da es sich um einen friedlichen Auftrag handelt, legte man beim Bau der HYDRA keinen großen Wert auf die militärische Verteidigung. Man gab ihr nur einen Geleitjäger mit, da man nicht an eine Bedrohung glaubte. Sie sind der Pilot dieses Jägers, und hatten bis jetzt nicht viel zu tun. Das ändert sich nun schlagartig. Als die HYDRA in das Gebiet der Zarks einfliegt sehen diese darin einen kriegerischen Akt und greifen die HYDRA sofort an. Sie sind nun die einzige Hoffnung für das wehrlose Schiff! Ihr Auftrag lautet: Verteidigen Sie die HYDRA!

Die HYDRA besteht aus drei Sektoren, die der Reihe nach angefliegen werden, nachdem jeweils 15 Angreifer vernichtet hat, ohne selbst getroffen zu werden. Von Level zu Level wird das Spielgeschehen schneller, so daß

Sie höllisch aufpassen müssen. Sie haben drei Jäger zur Verfügung - sind diese alle zerstört, so ist das Spiel beendet.

Eingabehinweise:
Da durch die angewendete Programmieretechnik frei Basic-Speicher nur noch 16 K umfaßt, mußte das Programm in zwei Teile aufgeteilt werden. Teil1, das Hauptprogramm, lädt drei Binär-Dateien nach, die von Teil 2, dem Data-Lader, erstellt wurden.

Diskettenbenutzer haben es einfach. Sie gehen wie folgt vor:

Teil 1 eingeben und unter dem Namen HERO abspeichern. Nun Teil 2 eingeben und unter DATA abspeichern. Nun Teil 2 laden und mit RUN starten. Ist alles fehlerfrei eingegeben worden, so werden nach kurzer Zeit 3 Dateien auf die Disc geschrieben.

Zum Spielen wird nur noch das Programm HERO geladen und gestartet.

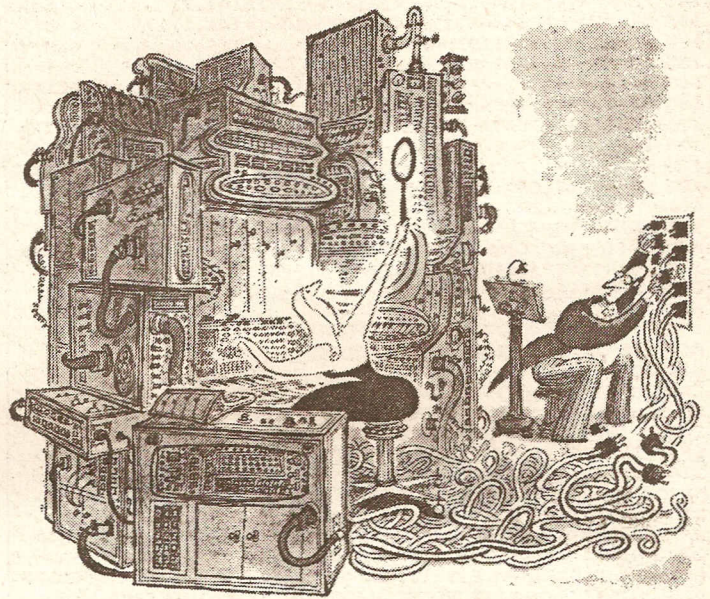
Kassettenbenutzer benutzen am besten zwei Kassetten.

Teil 1 eingeben und auf Kassette 1 abspeichern. Kassette nicht zurückspulen!

Teil 2 eintippen und auf Kassette 2 absaven. Nun Teil 2 laden und anschließend Kassette 1 einlegen. Teil 2 mit RUN starten. Es werden jetzt nach kurzer Zeit die drei Binär-Dateien abgespeichert. Zum Spielen wird nun nur noch

die Kassette 1 benötigt. Teil 1 laden und mit RUN starten. Die Binär-Dateien werden automatisch nachgeladen.

Viel Erfolg beim Eintippen dieses Programmes und viel Spaß!



Schalt jetzt mal »The Hero« ein!

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```

1 *****
2 * THE HERO *
3 * (c) 1986 *
4 * OTTFRIED *
5 * SCHMIDT *
6 *****
7
10 MODE 0:INK 0,0: BORDER 0:PAPE
R 0
20 INK 1,3:INK 2,15: INK 3,25
30 INK 4,0:INK 5,1: INK 6,13: INK
7,24: INK 8,6: INK 9,6,24
40 ENV 1,=11,5000:ENV 2,=11,250
0:ENT-1,3,4,2,3,-4,2,2,2,1
50 hsc=0
60 MEMORY &3FFF
70 PEN 2:LOCATE 4,12:PRINT"BITT
E WARTEN"
80 PEN 3:LOCATE 5,14:PRINT"LADE
DATEN"
90 LOAD"!heromc",&9200
100 LOAD"!graspr",&B200
110 LOAD"!sprite",&A000
120 CALL &BB03
130 CLS:PEN 2:LOCATE 6,10:PRINT
    
```

```

"THE HERO"
140 LOCATE 6,12:PEN 3:PRINT"(C)
1986"
{OD} 150 PEN 5:LOCATE 2,14:PRINT"OTT
{EF} FRIED SCHMIDT"
{LH} 160 PEN 8:LOCATE 5,16:PRINT"PRE
{BL} SS FIRE"
{JH} 170 PEN 9:LOCATE 6,18:PRINT"TO
{DK} START"
{DM} 180 GOSUB 1410:RESTORE
190 FOR i=&A690 TO &A690+&70:PO
{KA} KE i,0:NEXT i
{CJ} 200 CALL &9250:LOCATE 1,20
210 DATA B2,B2,B2,B3,B3,B4
{DB} 220 DATA 00,70,E0,50,C0,30
230 DATA A0,A0,A1,A1,A2,A2,A3,A
{MB} 3
240 DATA 70,E0,50,C0,30,A0,10,B
{BH} 0
{FK} 250 DATA A3,A4,A4,A5,A5,A6
260 DATA F0,60,D0,40,B0,20
{JD} 270 DIM h(6),l(6),ah(B),al(B),e
h(6),el(6)
{PB} 280 FOR i=1 TO 6:READ a$:h(i)=V
{CD} AL("&"+a$):NEXT i
{NH} 290 FOR i=1 TO 6:READ a$:l(i)=V
{EJ} AL("&"+a$):NEXT i
{AE} 300 FOR i=1 TO 8:READ a$:ah(i)=
    
```

```

{LF} VAL("&"+a$):NEXT i (MB)
310 FOR i=1 TO 8:READ a$:al(i)=
{OI} VAL("&"+a$):NEXT i (OP)
320 FOR i=1 TO 6:READ a$:eh(i)=
{AG} VAL("&"+a$):NEXT i (OD)
330 FOR i=1 TO 6:READ a$:el(i)=
{GE} VAL("&"+a$):NEXT i (LH)
340 DATA 129,2,1,60,0,0,15,50,0 (AD)
{NC} 350 DATA 130,2,1,40,0,0,14,45,0 (BH)
{ML} 360 FOR i=&B600 TO &B611:READ a
:POKE i,a:NEXT i (GF)
{DH} 370 sc=0: bild=1:lev1=4:lev2=151
{ML} :lev1=geg=0:liv=3 (HP)
{KH} 380 POKE &9135,lev1:POKE &9136,
{PL} lev2 (CJ)
390 x=2:y=0:IF bild=1 THEN REST
{IM} ORE 460 ELSE IF bild=2 THEN RES
TORE 580 ELSE RESTORE 700 (OI)
{NC} 400 FOR i=1 TO 11:FOR k=1 TO 11 (CI)
{OI} 410 READ a:POKE &9100,y:POKE &9
{HM} 101,x (HP)
420 POKE &9102,1(a):POKE &9103,
{NK} h(a) (FB)
430 CALL &92C7:x=x+7:NEXT k (NJ)
{FI} 440 x=2:y=y+8:NEXT i (FI)
450 'BILD 1 (DO)
{AF} 460 DATA 2,2,2,2,2,2,2,2,2,2 (OM)
470 DATA 2,3,3,3,1,1,1,3,3,3,2 (EO)
    
```



```

480 DATA 2,3,3,3,1,1,1,3,3,3,2
490 DATA 2,3,3,1,1,4,1,1,3,3,2
500 DATA 2,3,3,1,1,4,1,1,3,3,2
510 DATA 2,3,6,5,5,5,5,5,6,3,2
520 DATA 2,3,3,1,1,4,1,1,3,3,2
530 DATA 2,3,3,1,1,4,1,1,3,3,2
540 DATA 2,3,3,3,1,1,1,3,3,3,2
550 DATA 2,3,3,3,1,1,1,3,3,3,2

560 DATA 2,2,2,2,2,2,2,2,2,2,2
570 'BILD 2
580 DATA 3,3,3,3,3,3,3,3,3,3,3
590 DATA 3,2,3,5,5,5,5,5,3,2,3
600 DATA 3,3,3,3,3,3,3,3,3,3,3
610 DATA 3,4,4,4,6,6,6,4,4,4,3
620 DATA 3,1,1,2,2,2,2,2,1,1,3
630 DATA 2,5,5,5,5,5,5,5,5,2,2
640 DATA 3,1,1,2,2,2,2,2,1,1,3
650 DATA 3,4,4,4,6,6,6,4,4,4,3
660 DATA 3,3,3,3,3,3,3,3,3,3,3
670 DATA 3,2,3,5,5,5,5,5,3,2,3
680 DATA 3,3,3,3,3,3,3,3,3,3,3
690 'BILD 3
700 DATA 1,1,1,1,1,1,1,1,1,1,1
710 DATA 1,2,2,2,3,3,3,2,2,2,1
720 DATA 1,2,2,2,2,2,2,2,2,2,1
730 DATA 1,2,2,2,2,2,2,2,2,2,1
740 DATA 1,4,4,4,4,6,4,4,4,4,1
750 DATA 1,2,6,5,5,5,5,5,6,2,1
760 DATA 1,4,4,4,4,6,4,4,4,4,1
770 DATA 1,2,2,2,2,2,2,2,2,2,1
780 DATA 1,2,2,2,2,2,2,2,2,2,1
790 DATA 1,2,2,2,3,3,3,2,2,2,1
800 DATA 1,1,1,1,1,1,1,1,1,1,1
810 CALL &9200
820
830 PEN 3:LOCATE 1,23:PRINT" HI
BH":hsc::LOCATE 13,23:PRINT"LEV
EL":lev;
840 PEN 2:LOCATE 1,25:PRINT"SCO
RE":sc::LOCATE 13,25:PRINT"LIVE
S":liv;
850 POKE &9104,3:POKE &9105,0:P
OKE &9106,45
860 POKE &9108,4:POKE &9111,5:P
OKE &911A,3:REM ZAEHLER
870 POKE &9109,72:POKE &9112,72
:POKE &911B,72:REM X
880 POKE &910E,1:POKE &9117,2:P
OKE &9120,1:REM VEKTOR
890 POKE &910F,4:POKE &9118,5:P
OKE &9121,3:REM VEKTOR ZAEHLER
900 POKE &9110,4:POKE &9119,5:P
OKE &9122,3:REM ALTER VEKTOR ZA
EHLER
910 POKE &9128,0:POKE &912C,0:P
OKE &9130,0:POKE &9134,0:POKE &
9107,0
920 a1=INT(RND*8)+1:a2=INT(RND*
8)+1:a3=INT(RND*8)+1
930 POKE &910C,a1(a1):POKE &910
D,ah(a1)
940 POKE &9115,a1(a2):POKE &911
6,ah(a2)
950 POKE &911E,a1(a3):POKE &911
F,ah(a3):REM SPRITENUMMER
960 a1=INT(RND*80):a2=INT(RND*8
0):a3=INT(RND*80)
970 POKE &910A,a1:POKE &9113,a2
:POKE &911C,a3
980 POKE &9123,0
990 CALL &92D7
1000 IF PEEK(&9123)=1 THEN POKE
&9123,0:GOTO 1040
1010 IF PEEK(&9126)<>0 THEN GOT
O 1090
1020 IF PEEK(&9134)=1 THEN POKE
&9134,0:GOTO 1040
1030 GOTO 990
1040 POKE &9100,PEEK(&9106):POK
E &9101,PEEK(&9105)
1050 POKE &9128,0:POKE &912C,0:
POKE &9130,0
1060 GOSUB 1250:CALL &9233:CALL
&920C:FOR i=1 TO 500:NEXT i:PO
KE &9107,0
1070 liv=liv-1:IF liv<0 THEN GO

```

```

{HF} TO 1340
{CD} 1080 geg=0:GOTO 830
{CN} 1090 x=PEEK(&9126):IF x=1 THEN
adr=&9109 ELSE IF x=2 THEN adr=
{EL} &9112 ELSE adr=&911B
{FC}
{KC} 1100 POKE &9100,PEEK(adr+1):POK
E &9101,PEEK(adr)
{FJ} 1110 GOSUB 1250
{ON} 1120 POKE adr,72:POKE(adr+7),IN
T(RND*5)+1
{EE}
{EK} 1130 x=INT(RND*8)+1:POKE(adr+3)
,al(x):POKE(adr+4),ah(x)
{HB}
{JD} 1140 POKE &9107,0:POKE &9126,0
{JB} 1150 sc=sc+10+lev:LOCATE 6,25:P
{DI} RINT sc::geg=geg+1:IF geg>=15 T
{IM} HEN geg=0:flag=1:lev=lev+1:lev2
=lev2-50:IF lev2<=1 THEN lev2=1
{CN} 51:lev1=lev1-1:IF lev1<=1 THEN
lev1=1
{KE} 1160 IF flag<>1 THEN POKE(adr+1)
,INT(RND*80):GOTO 990
{EL}
{DG} 1170 flag=0:billd=billd+1:IF billd
>3 THEN billd=1
{DG}
{DF} 1180 CALL &9233:LOCATE 2,14:PRI
NT"NOW ENTER LEVEL":lev;
{DF}
{BE} 1190 SOUND 129,60,40,15,1:SOUND
{GP} 1,60,20,15,1:SOUND 1,40,60,15,
{KF} 1
{IN} 1200 WHILE SQ(1)>127:WEND
{JA} 1210 SOUND 1,60,40,15,1:SOUND 1
,60,20,15,1:SOUND 1,40,40,15,1:
{PP} WHILE SQ(1)>128:WEND
{ON} 1220 SOUND 1,47,60,15,1:SOUND 2
,60,60,15,1
{NM}
{DM} 1230 FOR i=1 TO 1000:NEXT i
1240 GOTO 380
1250 SOUND 129,2,80,15,1,0,30
{GP} 1260 FOR i=1 TO 6:CALL &BD19
1270 POKE &9102,el(i):POKE &910
3,eh(i):CALL &92C7
{NB} 1280 SOUND 130,i,5,15,1,0,30-i:
CALL &BD19:NEXT i
{JH} 1290 FOR i=6 TO 1 STEP-1:CALL &
BD19
{PI} 1300 POKE &9102,el(i):POKE &910
3,eh(i):CALL &92C7
{PI} 1310 SOUND 130,i,5,15,1,0,30-i:
CALL &BD19:NEXT i
{PI} 1320 POKE &9102,&90:POKE &9103,
&A6:CALL &92C7
{LH} 1330 RETURN
{BB} 1340 SOUND 129,10,70,15,1,1,30:
CALL &9233:CLS:FOR i=1 TO 24:PE
N INT(RND*10)+1:LOCATE 5,i:PRIN
T"GAME OVER":NEXT i
1350 FOR i=1 TO 500:NEXT i
{HE} 1360 IF hsc<2 THEN hsc=sc
1370 CLS:PEN 2:LOCATE 8,10:PRIN
T hsc:LOCATE 5,12:PRINT"HIGH SC
ORE"
{JB} 1380 PEN 3:LOCATE 8,14:PRINT sc
:LOCATE 5,16:PRINT"LAST SCORE"
{PI} 1390 GOSUB 1410:RESTORE
1400 GOTO 370
{FF} 1410 ENV 3,=11,7000:ENV 4,1,15,
1,15,-1,5:ENT-2,3,0.75,6,3,-0.7
5,6:ENT-3,2,2,4,2,-2,4,1,1,7,1,
-1,7
{EI} 1420 b=4:x=478:RESTORE 1490
{OC} 1430 READ a:IF a=0 THEN RESTORE
{AG} 1500:READ a:IF b=4 THEN b=2 EL
SE b=4
{JB} 1440 IF a<>1 THEN SOUND 130,a*2
,90,7,3,2:SOUND 132,a*b,160,7,3
,3
{OA}
{EF} 1450 SOUND 1,x,24,10,4,3:IF x=4
78 THEN x=379 ELSE x=478
{EK} 1460 IF JOY(0)AND 16 THEN RETUR
N
{CF} 1470 WHILE SQ(1)>127:WEND
1480 GOTO 1430
{LN} 1490 1,1,1
1500 DATA 239,1,239,1,190,159,1
90,1
{CF} 1510 DATA 1,1,239,1,239,1,190,1
59,190,1

```

```

{HF} 1520 DATA 1,1,190,1,190,1,159,1
{PD} 27,159,1
1530 DATA 1,1,284,1,284,1,190,1
59,190,1
{CF} 1540 DATA 1,1
1550 DATA 239,1,190,1,190,1,239
,284,190
{GM} 1560 DATA 1,1,1,190,1,190,1,239
,284,239
{PJ}
{FB} 1570 DATA 1,1,1,1,1,1,0
{GD} ENDE DES LISTINGS
{MD}

```

Teil 2

SCHNEIDER LISTING MIT CHECKSUMMEN V 2.0

```

{II}
{IK}
{LB} 1 '*****
2 * The Hero *
{MD} 3 * Teil 2 *
4 * Datalader *
{DB} 5 '*****
{OP} 6
7
1000 MEMORY &7FFF:MODE 0:LOCATE
4,12:PRINT"LESE DATA's"
{CH} 1010 DATA 3C,3C,3C,3C,3C,3C
{HA} 1020 DATA 3C,7B,FO,FO,FO,FO
{LD} 1030 DATA FO,BO,7B,FO,FO,FO
{GI} 1040 DATA FO,FO,BO,7B,3C,3C
{LO} 1050 DATA 3C,3C,3C,BO,7B,7B
{HG} 1060 DATA FO,FO,FO,BO,BO,7B
{PN} 1070 DATA 7B,FO,FO,FO,BO,BO
{FC} 1080 DATA 7B,7B,B4,3C,7B,BO
{KJ} 1090 DATA BO,7B,7B,B4,FO,70
{AG} 1100 DATA BO,BO,7B,7B,B4,FO
{EK} 1110 DATA 70,BO,BO,7B,7B,B4
{CO} 1120 DATA 30,70,BO,BO,7B,7B
{OE} 1130 DATA FO,FO,FO,BO,BO,7B
{AL} 1140 DATA 7B,FO,FO,FO,BO,BO
{FL} 1150 DATA 7B,3B,30,30,30,30
{GN} 1160 DATA BO,7B,FO,FO,FO,FO
{CD} 1170 DATA FO,BO,7B,FO,FO,FO
{OL} 1180 DATA FO,FO,BO,3B,30,30
{HB} 1190 DATA 30,30,30,30,3C,3C
{AJ} 1200 DATA 3C,3C,3C,3C,3C,7B
1210 DATA FO,FO,34,FO,FO,BO
1220 DATA 7B,FO,FO,34,FO,FO
1230 DATA BO,7B,FO,FO,34,FO
{CE} 1240 DATA FO,BO,7B,FO,FO,34
{JJ} 1250 DATA FO,FO,BO,7B,FO,FO
{JL} 1260 DATA 34,FO,FO,BO,7B,FO
1270 DATA FO,34,FO,FO,BO,7B
1280 DATA FO,FO,34,FO,FO,BO
{FB} 1290 DATA 7B,FO,FO,34,FO,FO
1300 DATA BO,7B,FO,FO,34,FO
{PB} 1310 DATA FO,BO,7B,FO,FO,34
{AB} 1320 DATA FO,FO,BO,7B,FO,FO
{DB} 1330 DATA 34,FO,FO,BO,7B,FO
1340 DATA FO,34,FO,FO,BO,7B
1350 DATA FO,FO,34,FO,FO,BO
1360 DATA 7B,FO,FO,34,FO,FO
{BF} 1370 DATA BO,3B,30,30,30,30
{BK} 1380 DATA 30,30,3C,3C,3C,3C
1390 DATA 3C,3C,3C,7B,FO,FO
1400 DATA 34,FO,FO,BO,7B,FO
{BL} 1410 DATA FO,34,FO,FO,BO,7B
1420 DATA FO,FO,34,FO,FO,BO
1430 DATA 7B,FO,FO,34,FO,FO
{HB} 1440 DATA BO,7B,FO,FO,34,FO
1450 DATA FO,BO,7B,FO,FO,34
{CL} 1460 DATA FO,FO,BO,3B,30,30
1470 DATA 34,30,30,30,3C,3C
{GH} 1480 DATA 3C,34,3C,3C,3B,7B
{CI} 1490 DATA FO,FO,34,FO,FO,BO
{PE} 1500 DATA 7B,FO,FO,34,FO,FO
{CH} 1510 DATA BO,7B,FO,FO,34,FO
1520 DATA FO,BO,7B,FO,FO,34
{CM} 1530 DATA FO,FO,BO,7B,FO,FO
1540 DATA 34,FO,FO,BO,7B,FO
{KD} 1550 DATA FO,34,FO,FO,BO,3B

```


programme

```

1560 DATA 30,30,30,30,30,30
1570 DATA 3C,3C,3C,3C,3C,3C
1580 DATA 3C,7B,F0,F0,F0,F0
1590 DATA F0,B0,7B,F0,F0,F0
1600 DATA F0,F0,B0,7B,F0,B4
1610 DATA 3C,7B,F0,B0,7B,F0
1620 DATA 7B,F0,B0,F0,B0,7B
1630 DATA B4,F0,F0,F0,70,B0
1640 DATA 7B,B4,F0,30,F0,70
1650 DATA B0,7B,B4,B0,C3,7B
1660 DATA 70,B0,7B,B4,B0,C3
1670 DATA 7B,70,B0,7B,B4,F0
1680 DATA 3C,F0,70,B0,7B,B4
1690 DATA F0,F0,F0,70,B0,7B
1700 DATA F0,70,F0,B0,F0,B0
1710 DATA 7B,F0,B0,30,F0,F0
1720 DATA B0,7B,F0,F0,F0,F0
1730 DATA F0,B0,7B,F0,F0,F0
1740 DATA F0,F0,B0,3B,30,3C
1750 DATA 30,30,30,30,3C,3C
1760 DATA 3C,3C,3C,3C,3C,7B
1770 DATA F0,F0,F0,F0,F0,B0
1780 DATA 30,30,30,30,30,30
1790 DATA 30,F0,F0,F0,F0,F0
1800 DATA F0,F0,F0,F0,F0,F0
1810 DATA F0,F0,F0,7B,F0,B4
1820 DATA F0,F0,7B,F0,B4,F0
1830 DATA F0,7B,F0,B4,F0,F0
1840 DATA 7B,F0,B4,F0,F0,7B
1850 DATA F0,7B,F0,B4,F0,F0
1860 DATA 7B,B4,F0,F0,7B,F0
1870 DATA B4,F0,7B,F0,B4,F0
1880 DATA F0,7B,F0,F0,F0,F0
1890 DATA F0,F0,F0,F0,F0,F0
1900 DATA F0,F0,F0,F0,F0,3C
1910 DATA 3C,3C,3C,3C,3C,3C
1920 DATA 7B,F0,F0,F0,F0,F0
1930 DATA B0,30,30,30,30,30
1940 DATA 30,30,3C,3C,3C,3C
1950 DATA 3C,3C,3C,7B,F0,F0
1960 DATA F0,F0,F0,B0,7B,F0
1970 DATA F0,F0,F0,F0,B0,7B
1980 DATA 30,30,30,30,30,B0
1990 DATA 7B,70,F0,F0,F0,B4
2000 DATA B0,7B,70,30,30,30
2010 DATA B4,B0,7B,70,30,30
2020 DATA 34,B4,B0,7B,70,30
2030 DATA F0,3C,B4,B0,7B,70
2040 DATA 30,F0,3C,B4,B0,7B
2050 DATA 70,30,3C,3C,B4,B0
2060 DATA 7B,70,34,3C,3C,B4
2070 DATA B0,7B,70,F0,F0,F0
2080 DATA B4,B0,7B,34,3C,3C
2090 DATA 3C,3C,B0,7B,F0,F0
2100 DATA F0,F0,F0,B0,7B,F0
2110 DATA F0,F0,F0,F0,3B,30
2120 DATA 30,30,30,30,30,3C
2130 DATA 00,CC,8B,00,C6,CC
2140 DATA 00,CC,8B,C9,CC,00
2150 DATA 44,C9,00,C9,CC,00
2160 DATA 00,00,00,00,00,00
2170 DATA 00,00,00,00,00,00
2180 DATA 00,00,00,00,00,00
2190 DATA 00,00,00,00,00,00
2200
2210 RESTORE 1010:x=0
2220 FOR i=&B400 TO &B6C9
2230 READ a$
2240 POKE i,VAL("&"+a$)
2250 x=x+VAL("&"+a$)
2260 NEXT i
2270 IF x<109319 THEN CLS:PRIN
T"FEHLER in DATA's":END
2280 DATA 11,00,40,21,00,C0
2290 DATA 01,00,40,ED,B0,C9
2300 DATA 21,00,C0,06,03,11
2310 DATA 03,00,E5,CD,19,BD
2320 DATA CD,20,92,E1,23,10
2330 DATA F5,C9,CB,BC,7E,CB
2340 DATA FC,77,CD,2D,92,19
2350 DATA DB,1B,F3,3E,0A,3D
2360 DATA 20,FD,C9,21,00,C0
2370 DATA 06,03,11,03,00,E5
2380 DATA CD,19,BD,CD,47,92
2390 DATA E1,23,10,F5,C9,AF
2400 DATA 77,CD,2D,92,19,DB
(IM) 2410 DATA 1B,F7,21,00,C0,DD
(ED) 2420 DATA 21,00,80,FD,21,00
(NJ) 2430 DATA B1,06,64,7C,DD,77
(CM) 2440 DATA 00,7D,FD,77,00,DD
(JI) 2450 DATA 23,FD,23,CD,26,BC
(PC) 2460 DATA CD,26,BC,10,EC,C9
(ML) 2470 DATA DD,21,00,80,FD,21
(OH) 2480 DATA 00,81,7A,32,83,92
(HC) 2490 DATA 32,87,92,DD,7E,00
(BG) 2500 DATA 67,FD,7E,00,6F,16
(AJ) 2510 DATA 00,19,E5,16,10,1E
(JL) 2520 DATA 07,0A,FE,00,2B,03
(BJ) 2530 DATA 77,1B,06,CB,BC,7E
(IC) 2540 DATA CB,FC,77,03,23,1D
(AF) 2550 DATA 20,ED,E1,7C,C6,0B
(GM) 2560 DATA 67,E6,3B,20,14,7C
(CL) 2570 DATA D6,40,67,7D,C6,50
(OE) 2580 DATA 6F,30,0A,24,7C,E6
(BB) 2590 DATA 07,20,04,7C,D6,0B
(IP) 2600 DATA 67,E5,15,20,CA,E1
(FF) 2610 DATA C9,3A,00,91,57,3A
(DP) 2620 DATA 01,91,5F,ED,4B,02
(IM) 2630 DATA 07,0A,FE,00,2B,03
(CI) 2640 DATA 04,91,3D,2B,05,32
(HB) 2650 DATA 04,91,1B,0B,3E,02
(EF) 2660 DATA 32,04,91,CD,B6,93
(OE) 2670 DATA 3A,0B,91,3D,2B,05
(AN) 2680 DATA 32,0B,91,1B,0E,3E
(JL) 2690 DATA 03,32,0B,91,CD,09
(AH) 2700 DATA 94,3A,23,91,FE,00
(LJ) 2710 DATA C0,3A,11,91,3D,2B
(ML) 2720 DATA 05,32,11,91,1B,0E
(JA) 2730 DATA 3E,03,32,11,91,CD
(PM) 2740 DATA 0F,94,3A,23,91,FE
(FI) 2750 DATA 00,C0,3A,1A,91,3D
(BJ) 2760 DATA 2B,05,32,1A,91,1B
(KB) 2770 DATA 0E,3E,03,32,1A,91
(JA) 2780 DATA CD,15,94,3A,23,91
(MC) 2790 DATA FE,00,C0,3A,07,91
(EI) 2800 DATA FE,00,C4,F4,94,3A
(LJ) 2810 DATA 26,91,FE,00,C2,D4
(DM) 2820 DATA 95,3A,2B,91,FE,00
(DD) 2830 DATA 2B,1D,3A,29,91,3D
(FB) 2840 DATA 20,14,3E,02,32,29
(FF) 2850 DATA 91,DD,21,2B,91,CD
(NB) 2860 DATA 21,96,3A,34,91,FE
(BD) 2870 DATA 00,C0,1B,03,32,29
(DD) 2880 DATA 91,3A,2C,91,FE,00
(CD) 2890 DATA 2B,1D,3A,2D,91,3D
(DJ) 2900 DATA 20,14,3E,02,32,2D
(LN) 2910 DATA 91,DD,21,2C,91,CD
(JH) 2920 DATA 21,96,3A,34,91,FE
(MK) 2930 DATA 00,C0,1B,03,32,2D
(LI) 2940 DATA 91,3A,30,91,FE,00
(AE) 2950 DATA CA,82,96,3A,31,91
(OD) 2960 DATA 3D,20,15,3E,02,32
(CB) 2970 DATA 31,91,DD,21,30,91
(KN) 2980 DATA CD,21,96,3A,34,91
(GL) 2990 DATA FE,00,C0,C3,82,96
(LM) 3000 DATA 32,31,91,C3,82,96
(HL) 3010 DATA CD,24,8B,CB,44,CC
(FD) 3020 DATA FE,93,CB,4C,CC,F3
(KN) 3030 DATA 93,CB,5C,CC,DD,93
(EA) 3040 DATA CB,54,CC,EB,93,CB
(AB) 3050 DATA 64,C4,D0,94,01,00
(AP) 3060 DATA A0,ED,5B,05,91,CD
(NB) 3070 DATA 72,92,C9,3A,05,91
(MN) 3080 DATA FE,00,CB,3D,32,05
(OB) 3090 DATA 91,C9,3A,05,91,FE
(AE) 3100 DATA 1E,CB,3C,32,05,91
(FK) 3110 DATA C9,3A,06,91,FE,00
3120 DATA CB,3D,32,06,91,C9
3130 DATA 3A,06,91,FE,50,CB
3140 DATA 3C,32,06,91,C9,DD
3150 DATA 21,0B,91,1B,0A,DD
3160 DATA 21,11,91,1B,04,DD
3170 DATA 21,1A,91,DD,7E,01
3180 DATA FE,00,2B,06,3D,DD
3190 DATA 77,01,1B,13,01,90
3200 DATA A6,5F,DD,56,02,DD
(AK) 3210 DATA E5,CD,72,92,DD,E1
(OD) 3220 DATA 3E,4B,DD,77,01,DD
3230 DATA 7E,07,3D,2B,05,DD
(PP) 3240 DATA 77,07,1B,14,DD,7E
(FE) 3250 DATA 0B,DD,77,07,DD,7E
(AI) 3260 DATA 06,67,CB,44,CC,7C
(JF) 3270 DATA 94,CB,4C,CC,8E,94
(JA) 3280 DATA DD,7E,01,5F,DD,7E
(FI) 3290 DATA 02,57,DD,7E,04,4F
(NH) 3300 DATA DD,7E,05,47,DD,E5
(OF) 3310 DATA CD,72,92,DD,E1,CD
(EH) 3320 DATA A0,94,ED,5F,E6,6F
(GL) 3330 DATA FE,0A,CC,DB,95,C9
(HA) 3340 DATA DD,7E,02,FE,00,2B
(AO) 3350 DATA 05,3D,DD,77,02,C9
(CH) 3360 DATA 3E,01,DD,77,06,C9
(DK) 3370 DATA DD,7E,02,FE,50,2B
(DN) 3380 DATA 05,3C,DD,77,02,C9
(AJ) 3390 DATA 3E,02,DD,77,06,C9
(AK) 3400 DATA 3A,05,91,06,07,80
(BO) 3410 DATA 47,DD,7E,01,8B,DD
(CH) 3420 DATA 4F,7B,06,07,90,47
(FI) 3430 DATA 79,8B,DB,3A,06,91
(CI) 3440 DATA 06,0B,80,47,DD,7E
(HB) 3450 DATA 02,8B,DD,4F,7B,06
(PB) 3460 DATA 0B,90,47,79,8B,DB
(MJ) 3470 DATA 3E,01,32,23,91,C9
(NJ) 3480 DATA 3A,07,91,FE,00,C0
(BI) 3490 DATA 3E,01,32,07,91,3A
(PP) 3500 DATA 05,91,06,07,80,32
(EK) 3510 DATA 24,91,3A,06,91,06
(EE) 3520 DATA 03,80,32,25,91,21
(GL) 3530 DATA 00,86,CD,AA,BC,C9
(FN) 3540 DATA 3A,24,91,FE,4B,2B
(NP) 3550 DATA 13,3C,32,24,91,5F
(EK) 3560 DATA 3A,25,91,57,01,A0
(OE) 3570 DATA B4,CD,7F,95,CD,21
(DN) 3580 DATA 95,C9,AF,32,07,91
(JL) 3590 DATA 3A,24,91,5F,3A,25
(PI) 3600 DATA 91,57,01,CO,84,CD
(MI) 3610 DATA 7F,95,C9,DD,21,0B
(GM) 3620 DATA 91,CD,53,95,FE,00
(LF) 3630 DATA 20,06,3E,01,32,26
(PK) 3640 DATA 91,C9,DD,21,11,91
(DF) 3650 DATA CD,53,95,FE,00,20
(MF) 3660 DATA 06,3E,02,32,26,91
(AL) 3670 DATA C9,DD,21,1A,91,CD
(BH) 3680 DATA 53,95,FE,00,C0,3E
(OD) 3690 DATA 03,32,26,91,C9,DD
(CP) 3700 DATA 7E,01,06,07,80,47
(CF) 3710 DATA 3A,24,91,8B,DD,4F
(AO) 3720 DATA 7B,06,07,90,47,79
(CM) 3730 DATA 8B,DB,DD,7E,02,06
(EO) 3740 DATA 0B,80,47,3A,25,91
(PL) 3750 DATA 8B,DD,4F,7B,06,0B
(AL) 3760 DATA 90,47,79,8B,DB,AF
(BP) 3770 DATA C9,DD,21,00,80,FD
(BN) 3780 DATA 21,00,81,7A,32,90
(AI) 3790 DATA 95,32,94,95,DD,7E
(IJ) 3800 DATA 00,67,FD,7E,00,6F
(NP) 3810 DATA 16,00,19,E5,16,03
(PD) 3820 DATA 1E,03,0A,FE,00,2B
(GE) 3830 DATA 03,77,1B,06,CB,BC
(AF) 3840 DATA 7E,CB,FC,77,03,23
(HD) 3850 DATA 1D,20,ED,E1,7C,C6
(BL) 3860 DATA 0B,67,E6,3B,20,14
(FA) 3870 DATA 7C,DD,40,67,7D,C6
(AL) 3880 DATA 50,6F,30,0A,24,7C
(DN) 3890 DATA E6,07,20,04,7C,D6
(BH) 3900 DATA 0B,67,E5,15,20,CB
(IL) 3910 DATA E1,C9,CD,0E,95,C9
(BM) 3920 DATA DD,7E,01,06,19,9B
(CJ) 3930 DATA DB,3A,2B,91,FE,00
(IG) 3940 DATA 2B,0F,3A,2C,91,FE
(OB) 3950 DATA 00,2B,0E,3A,30,91
(FD) 3960 DATA FE,00,2B,0D,C9,FD
(BC) 3970 DATA 21,2B,91,1B,0A,FD
(KJ) 3980 DATA 21,2C,91,1B,04,FD
(OC) 3990 DATA 21,30,91,3E,01,FD
(KL) 4000 DATA 77,00,3C,FD,77,01
(CC) 4010 DATA DD,7E,01,FB,77,02
(LK) 4020 DATA DD,7E,02,FD,77,03
(CJ) 4030 DATA 21,09,86,CD,AA,BC
(CK) 4040 DATA C9,DD,7E,02,FE,00
(DB) 4050 DATA 2B,17,3D,DD,77,02
(KI) 4060 DATA 5F,DD,7E,03,57,01
(PD) 4070 DATA A9,84,DD,E5,CD,7F
(LA) 4080 DATA 95,DD,E1,CD,52,96
(DP) 4090 DATA C9,AF,DD,77,00,DD
(AE) 4100 DATA 7E,02,5F,DD,7E,03
(HL) 4110 DATA 57,01,CO,84,CD,7F
(KD) 4120 DATA 95,C9,3A,05,91,06
(PD) (MH) (GM) (AI) (AF) (BJ) (DP) (MH) (FB) (DH) (PD) (FP) (DL) (GF) (BC) (EN) (FM) (FE) (KP) (HP) (BC) (CL) (JF) (MD) (BA) (PD) (AB) (KJ) (JL) (DF) (KL) (BH) (DE) (NO) (AH) (LF) (LK) (OG) (AC) (BE) (CH) (AC) (LL) (CP) (AK) (EP) (PO) (FA) (DH) (KC) (BM) (EB) (DM) (IP) (MG) (AG) (HD) (ID) (ND) (AG) (CI) (EN) (ON) (KM) (AD) (DC) (EJ) (JB) (JA) (LH) (OA) (NB) (AD) (IP) (FL) (HF) (GC) (HF) (EB) (IJ) (DA) (PN) (JC) (PB)

```



```

4130 DATA 07,80,47,DD,7E,02
4140 DATA 88,00,4F,78,06,07
4150 DATA 90,47,79,88,08,3A
4160 DATA 06,91,06,08,80,47
4170 DATA DD,7E,03,88,00,4F
4180 DATA 78,06,08,90,47,79
4190 DATA 88,08,3E,01,32,34
4200 DATA 91,C9,3A,35,91,47
4210 DATA 48,3A,36,91,47,10
4220 DATA FE,41,10,F6,C3,D7
4230 DATA 92,44,20,22,21,41
4240 DATA 53,53,45,4D,42,4C
4250 DATA 45,2E,42,49,4E,0D
4260 DATA 0A,33,30,20,43,41
4270 DATA 4C,4C,20,26,38,46
4280 DATA 35,42,0D,0A,34,30
4290 DATA 20,44,45,4C,45,54
4300 DATA 45,20,31,30,2D,34
4310 DATA 30,0D,0A,1A,1A,55
4320 DATA 53,41,54,5A,20,48
4330 DATA 49,45,5A,55,22,2C
4340 DATA 31,34,34,0D,0A,22
4350 DATA 31,31,22,2C,22,4B
4360 DATA 46,5A,2D,53,54,45
4370 DATA 55,45,52,22,2C,31
4380 DATA 32,2C,32,32,35,0D
4390 DATA 0A,1A,31,34,39,32
4400 DATA 0D,0A,22,32,00,00
4410
4420 RESTORE 2280:x=0
4430 FOR i=&9200 TO &96FD
4440 READ a$
4450 POKE i,VAL("&" + a$)
4460 x=x+VAL("&" + a$)
4470 NEXT i
4480 IF x<>128237 THEN CLS:PRIN
T"FEHLER in DATA"s":END
4490 DATA 00,00,00,00,00,00
4500 DATA 00,00,00,00,00,00
4510 DATA 00,00,00,CC,CC,CC
4520 DATA 00,00,00,00,44,4C
4530 DATA 00,00,00,00,00,44
4540 DATA 0C,88,00,00,00,00
4550 DATA 8C,8C,4C,88,00,00
4560 DATA 00,8C,4C,8C,4C,00
4570 DATA 00,00,44,8C,0C,8C
4580 DATA CC,00,00,40,84,0C
4590 DATA 84,C0,00,00,84,48
4600 DATA C0,48,00,00,00,84
4610 DATA 84,48,80,00,00,00
4620 DATA 40,0C,80,00,00,00
4630 DATA 00,40,48,00,00,00
4640 DATA 00,00,C0,C0,C0,00
4650 DATA 00,00,00,00,00,00
4660 DATA 00,00,00,00,00,00
4670 DATA 00,00,00,00,00,00
4680 DATA 00,00,00,00,00,00
4690 DATA 00,00,00,00,00,00
4700 DATA 00,44,CC,CC,00,00
4710 DATA 00,00,CC,CC,8C,88
4720 DATA 00,00,44,8C,0C,4C
4730 DATA 4C,00,00,CC,84,0C
4740 DATA 48,0C,88,00,40,48
4750 DATA C0,84,8C,88,00,00
4760 DATA 84,0C,4C,4C,88,00
4770 DATA 00,8C,0C,48,48,80
4780 DATA 00,44,4C,CC,8C,84
4790 DATA 80,00,CC,8C,0C,4C
4800 DATA 0C,80,00,40,84,0C
4810 DATA 48,48,00,00,00,C0
4820 DATA C0,84,80,00,00,00
4830 DATA 40,C0,C0,00,00,00
4840 DATA 00,00,00,00,00,00
4850 DATA 00,00,00,00,00,00
4860 DATA 00,00,00,00,00,00
4870 DATA 00,00,00,00,00,00
4880 DATA 00,00,00,00,44,CC
4890 DATA CC,CC,88,00,00,00
4900 DATA 84,0C,8C,4C,00,00
4910 DATA 00,40,4C,8C,CC,88
4920 DATA 00,00,44,0C,C4,0C
4930 DATA 88,00,00,8C,CC,C8
4940 DATA 0C,C4,00,CC,4C,0C
4950 DATA 4C,C0,4C,00,C0,48
4960 DATA 0C,48,CC,48,00,00
4970 DATA 84,C0,C4,0C,C8,00

(AA) 4980 DATA 00,40,0C,C8,0C,80
(EF) 4990 DATA 00,00,44,48,84,C0
(FC) 5000 DATA 80,00,00,8C,0C,84
(FA) 5010 DATA 48,00,00,44,C0,C0
(CJ) 5020 DATA C0,80,00,00,00,00
(EO) 5030 DATA 00,00,00,00,00,00
(DL) 5040 DATA 00,00,00,00,00,00
(FN) 5050 DATA 00,00,00,00,00,00
(BD) 5060 DATA 00,00,00,00,00,00
(PE) 5070 DATA 00,00,00,00,00,CC
(EA) 5080 DATA CC,CC,00,00,00,00
(HD) 5090 DATA 44,4C,00,00,00,00
(CH) 5100 DATA 00,8C,4C,88,00,00
(LK) 5110 DATA 44,CC,0C,8C,4C,00
(DM) 5120 DATA 00,8C,4C,CC,0C,4C
(CB) 5130 DATA 00,CC,4C,8C,0C,CC
(JB) 5140 DATA 88,00,C0,48,84,0C
(LI) 5150 DATA C0,80,00,00,84,48
(EE) 5160 DATA C0,0C,48,00,00,40
(KN) 5170 DATA C0,0C,84,48,00,00
(IL) 5180 DATA 00,00,84,48,80,00
(OA) 5190 DATA 00,00,00,40,48,00
(AF) 5200 DATA 00,00,00,00,C0,C0
(CD) 5210 DATA C0,00,00,00,00,00
(EJ) 5220 DATA 00,00,00,00,00,00
(JD) 5230 DATA 00,00,00,00,00,00
(OB) 5240 DATA 00,00,00,00,00,00
(GE) 5250 DATA 00,00,00,00,00,00
(PM) 5260 DATA 00,00,CC,CC,00,00
(BI) 5270 DATA 00,00,CC,0C,4C,CC
(LC) 5280 DATA 00,00,44,4C,CC,CC
(NG) 5290 DATA 0C,88,00,8C,48,C0
(KN) 5300 DATA C0,CC,CC,00,8C,CC
(NF) 5310 DATA 8C,CC,4C,4C,00,CC
(GF) 5320 DATA 0C,4C,0C,8C,CC,00
5330 DATA C0,0C,48,0C,84,C0
(PD) 5340 DATA 00,84,C0,84,C0,48
(KP) 5350 DATA 48,00,84,4C,CC,CC
(DD) 5360 DATA C0,C0,00,40,48,C0
(DK) 5370 DATA C0,0C,80,00,00,C0
(CK) 5380 DATA 0C,48,C0,00,00,00
(II) 5390 DATA 00,C0,C0,00,00,00
(CC) 5400 DATA 00,00,00,00,00,00
(HH) 5410 DATA 00,00,00,00,00,00
(NI) 5420 DATA 00,00,00,00,00,00
(MH) 5430 DATA 00,00,00,00,00,00
(KI) 5440 DATA 00,00,00,00,00,44
(OH) 5450 DATA CC,88,00,00,00,00
(MD) 5460 DATA 8C,0C,4C,44,00,00
(MN) 5470 DATA 44,4C,CC,0C,C9,00
(LJ) 5480 DATA 00,8C,8C,0C,C8,40
(GH) 5490 DATA 00,00,8C,84,0C,80
(DN) 5500 DATA 00,00,00,44,48,C0
(HJ) 5510 DATA 4C,88,00,00,40,4C
(KM) 5520 DATA CC,48,80,00,00,8C
(HL) 5530 DATA 8C,0C,88,00,00,00
(FO) 5540 DATA 84,84,0C,C4,44,00
(KN) 5550 DATA 00,40,48,C0,0C,C9
(EH) 5560 DATA 00,00,00,84,0C,48
(DF) 5570 DATA 40,00,00,00,40,C0
(FO) 5580 DATA 80,00,00,00,00,00
(OA) 5590 DATA 00,00,00,00,00,00
(BK) 5600 DATA 00,00,00,00,00,00
(BO) 5610 DATA 00,00,00,00,00,00
(CM) 5620 DATA 00,00,00,00,00,00
(PM) 5630 DATA 00,00,00,00,44,CC
(DH) 5640 DATA 88,00,00,00,CC,8C
(BD) 5650 DATA CC,CC,00,00,00,44
(JI) 5660 DATA 4C,C0,C0,88,00,00
(FI) 5670 DATA 8C,C8,CC,8C,88,00
(FP) 5680 DATA 44,0C,4C,4C,CC,4C
(CB) 5690 DATA 00,8C,4C,8C,8C,4C
(HI) 5700 DATA C8,00,84,48,84,84
(GH) 5710 DATA 48,C0,00,40,0C,48
(MK) 5720 DATA 48,C0,48,00,00,84
(HJ) 5730 DATA C4,C0,84,80,00,00
(KC) 5740 DATA 40,48,CC,CC,80,00
(GB) 5750 DATA 00,C0,84,C0,C0,00
(LI) 5760 DATA 00,00,00,40,C0,80
(DK) 5770 DATA 00,00,00,00,00,00
(EO) 5780 DATA 00,00,00,00,00,00
(DI) 5790 DATA 00,00,00,00,00,00
(OB) 5800 DATA 00,00,00,00,00,00
(LA) 5810 DATA 00,00,00,00,00,00
(CA) 5820 DATA 00,00,00,44,CC,88
(BM) 5830 DATA 00,00,00,CC,8C,CC

(CD) 5840 DATA CC,00,00,00,00,CB
(IH) 5850 DATA 0C,C0,00,00,00,CC
(PL) 5860 DATA 0C,C0,80,00,00,CC
(IK) 5870 DATA 0C,CC,CC,80,00,CC
(GE) 5880 DATA 0C,CC,0C,0C,CC,00
(FO) 5890 DATA C0,0C,C0,0C,0C,C0
(KN) 5900 DATA 00,00,C0,0C,C0,C0
(EA) 5910 DATA 80,00,00,00,C0,C0
(KP) 5920 DATA CC,80,00,00,00,00
(IA) 5930 DATA C0,0C,4C,00,00,00
(JH) 5940 DATA C0,84,C0,C0,00,00
(MN) 5950 DATA 00,00,40,C0,80,00
(GK) 5960 DATA 00,00,00,00,00,00
(LM) 5970 DATA 00,00,00,00,00,00
(DC) 5980 DATA 00,00,00,00,00,00
5990 DATA 00,00,00,00,00,00
(DE) 6000 DATA 00,00,00,00,00,00
(LI) 6010 DATA CC,CC,00,00,00,00
(KA) 6020 DATA CC,CC,C8,88,00,00
(HP) 6030 DATA 44,4C,C0,C4,C4,00
(KE) 6040 DATA 00,8C,0C,0C,4C,84
(OL) 6050 DATA 88,00,C0,84,0C,C8
(NF) 6060 DATA 4C,CC,00,00,40,CC
(MP) 6070 DATA 84,8C,4C,00,00,44
(HJ) 6080 DATA C0,8C,84,48,00,CC
(KM) 6090 DATA 8C,0C,C4,48,C0,00
(HL) 6100 DATA 84,0C,0C,48,C8,80
(FO) 6110 DATA 00,40,48,CC,C8,C0
(LL) 6120 DATA 00,00,00,C0,C0,C0
(DI) 6130 DATA 80,00,00,00,00,C0
(NE) 6140 DATA C0,00,00,00,00,00
(AB) 6150 DATA 00,00,00,00,00,00
(AK) 6160 DATA 00,00,00,00,00,00
(MA) 6170 DATA 00,00,00,00,00,00
(EE) 6180 DATA 00,00,00,00,00,00
(BJ) 6190 DATA 00,00,00,00,00,00
(CO) 6200 DATA 00,00,00,00,00,00
(GC) 6210 DATA 40,80,00,00,00,00
(OP) 6220 DATA 00,81,42,00,00,00
(KK) 6230 DATA 00,40,56,42,00,00
(KM) 6240 DATA 00,40,46,A9,80
(MJ) 6250 DATA 00,00,00,81,CC,03
(GF) 6260 DATA 80,00,00,00,40,56
(HI) 6270 DATA 42,00,00,00,00,40
(GH) 6280 DATA 03,C0,00,00,00,00
(MK) 6290 DATA 00,C0,00,00,00,00
(IJ) 6300 DATA 00,00,00,00,00,00
(FM) 6310 DATA 00,00,00,00,00,00
(KE) 6320 DATA 00,00,00,00,00,00
(AI) 6330 DATA 00,00,00,00,00,00
(AH) 6340 DATA 00,00,00,00,00,00
(PM) 6350 DATA 00,00,00,00,00,00
(MB) 6360 DATA 00,00,00,00,00,00
(EL) 6370 DATA 00,00,00,00,00,00
(LL) 6380 DATA 00,00,00,00,00,00
(GB) 6390 DATA 00,00,00,C0,80,80
(EA) 6400 DATA 00,00,00,00,81,42
(JB) 6410 DATA 42,00,00,00,00,40
(FB) 6420 DATA 03,09,80,00,00,00
(BC) 6430 DATA 00,81,09,80,00,00
(MI) 6440 DATA 00,00,81,89,80,00
(KP) 6450 DATA 00,00,40,46,42,00
(DD) 6460 DATA 00,00,00,81,0C,42
(JG) 6470 DATA 00,00,00,40,06,06
(GF) 6480 DATA 42,00,00,00,40,03
(OQ) 6490 DATA 81,80,00,00,00,00
(PE) 6500 DATA C0,40,00,00,00,00
(HA) 6510 DATA 00,00,00,00,00,00
(CA) 6520 DATA 00,00,00,00,00,00
(EH) 6530 DATA 00,00,00,00,00,00
(HF) 6540 DATA 00,00,00,00,00,00
(ND) 6550 DATA 00,00,00,00,00,00
(HD) 6560 DATA 00,00,00,00,00,00
(JJ) 6570 DATA 40,80,C0,00,00,00
(HB) 6580 DATA 00,81,42,03,80,00
(EF) 6590 DATA 00,00,81,09,42,00
(CA) 6600 DATA 00,00,00,40,06,42
(ND) 6610 DATA 00,00,00,00,81,0C
(JF) 6620 DATA 42,00,00,00,40,06
(HL) 6630 DATA CC,09,80,00,00,40
(FO) 6640 DATA 06,DC,09,80,00,00
(KN) 6650 DATA 00,81,0C,0C,42,00
(OB) 6660 DATA 00,00,40,03,06,42
(GE) 6670 DATA 00,00,00,40,42,81
(GH) 6680 DATA 80,00,00,00,00,80
(PB) 6690 DATA 40,00,00,00,00,00

(AG)
(EG)
(GB)
(PL)
(EJ)
(DC)
(II)
(PM)
(OB)
(FO)
(FM)
(AP)
(JO)
(MK)
(HJ)
(KM)
(HL)
(FO)
(EB)
(EH)
(IH)
(KD)
(GB)
(HK)
(DF)
(NC)
(PM)
(LI)
(DD)
(HO)
(EA)
(KP)
(FC)
(EB)
(LE)
(KM)
(JH)
(FL)
(MO)
(GK)
(AF)
(NP)
(LD)
(GE)
(DA)
(BH)
(MK)
(HL)
(FO)
(KN)
(EA)
(KP)
(JK)
(HH)
(EJ)
(DC)
(MK)
(HI)
(MK)
(HJ)
(KM)
(HL)
(PB)
(BB)
(BI)
(AF)
(DC)
(LA)
(KC)
(II)
(HJ)
(MG)
(JA)
(ED)

```


16 RSX Befehle

Der CPC 464 stellt dem Anwender ein umfangreiches Basic zur Verfügung. Manche Befehle kann man aber nur über chrs-Folgen oder Call-Befehle erreichen. Meine Erweiterung stellt mit einfachen Befehlsnamen eine echte Programmierhilfe dar. Außerdem werden die Programme übersichtlicher.

Die neuen Befehle lauten:

Hinweis: Ö = SHIFT+KLAMMERAFFE (Taste rechts neben P)

1. ÖREV: Schaltet in Reverse Darstellung. Erneutes aufrufen schaltet zurück in normale Darstellung
2. ÖCRACK: Ermöglicht das Einladen geschützter Basic Programme, Anwendung: ÖCRACK:load""list
3. ÖEMPTY: Löscht den Tastaturpuffer (z. B. vor INKEYS)
4. ÖWAIT: Wartet auf Tastendruck
5. ÖXOR: Schaltet auf XOR-Graphikmodus
6. ÖAND: Schaltet auf AND-Graphikmodus
7. ÖOR: Schaltet auf OR-Graphikmodus
8. ÖNORM: Schaltet auf Normalen Graphikmodus
9. ÖFRAME: Wartet auf Strahlenrücklauf. Dadurch werden Pixelweise Bewegungen flüssiger

10. ÖSCRUP: Scrollt Bildschirm nach oben
11. ÖSCRDW: Scrollt Bildschirm nach unten
12. ÖKOMP: Schaltet auf speicherplatzsparenden Komprimiermodus
Aus 10 print
wird 10 print
13. ÖBILD1: Setzt Bildschirmspeicheranfang ab 49152
14. ÖBILD2: Setzt Bildschirmspeicheranfang ab 16384
Bitte vorher mit MEMORY 16383 schützen
15. ÖHOME: Bringt den Textcursor in die linke obere Ecke des Bildschirms ohne diesen zu löschen.
16. ÖSCHUTZ: Das Programm wird bei EDIT oder LIST gelöscht

Der Basiclader mit Demoprogramm belegt 7,32 KB, das Maschinenprogramm genau 333 Bytes.

Das Maschinenprogramm kann mit:

SAVE"RSX BEFEHLE.OBJ",&a000,333,&a000 gespeichert und danach mit MEMORY &9FFF:load"RSX BEFEHLE.OBJ",&a000:call &a000 in eigene Programme geladen werden.

Im Demoprogramm werden für Plotbeschleunigung einige Variablen mit Sin- und Coswerten belegt. Deshalb ist der Bildschirm, nachdem eine Taste gedrückt wurde, für einige Sekunden gelöscht.

Durch ÖBILD2 kann ein Bild im Speicher abgelegt oder durch ÖBILD1 zurück auf den Bildschirm geholt werden, ohne es zu zerstören.

Das ganze funktioniert auch umgekehrt.

SCHNEIDER LISTING MIT CHECKSUMMEN V 2.0

```

10 REM *****
*** 16 RSX Befehle *****
*****
20 REM ***** BY ALEX
ANDER BLOSS FROM ABC SOFT *****
*****
30 REM *****
*****
40 DATA &21,&0A,&A0,&01,&0E,&A0
,&CD,&D1,&BC,&C9,&0A,&A0,&0E,&A
0,&4C,&A0,&C3
50 DATA &A1,&A0,&C3,&A5,&A0,&C3
,&BB,&A0,&C3,&BC,&A0,&C3,&C0,&A
0,&C3,&CB,&A0
60 DATA &C3,&D6,&A0,&C3,&E1,&A0
,&C3,&EC,&A0,&C3,&F0,&A0,&C3,&F
4,&A0,&C3,&FB
70 DATA &A0,&C3,&FE,&A0,&C3,&04
,&A1,&C3,&0A,&A1,&C3,&10,&A1,&C
3,&16,&A1,&C3
80 DATA &1C,&A1,&C3,&2F,&A1,&C3
,&3E,&A1,&52,&45,&D6,&43,&52,&4
1,&43,&CB,&45
90 DATA &4D,&50,&54,&D9,&57,&41
,&49,&D4,&5B,&4F,&D2,&41,&4E,&C
4,&4F,&D2,&4E
100 DATA &4F,&52,&CD,&43,&55,&5
2,&53,&4F,&D2,&4F,&46,&C6,&46,&
52,&41,&4D,&C5
110 DATA &53,&43,&52,&55,&D0,&5
3,&43,&52,&44,&D7,&4B,&4F,&4D,&
D0,&42,&49,&4C
120 DATA &44,&B1,&42,&49,&4C,&4
4,&B2,&4B,&4F,&4D,&C5,&53,&43,&
4B,&55,&54,&DA
130 DATA &53,&41,&56,&C5,&4C,&4
F,&41,&C4,&CD,&9C,&BB,&C9,&21,&
01,&AC,&36,&C3
140 DATA &23,&11,&B2,&A0,&73,&2
3,&72,&C9,&21,&45,&AE,&36,&00,&
C9,&CD,&03,&BB
150 DATA &C9,&CD,&1B,&BB,&C9,&3
E,&17,&CD,&5A,&BB,&3E,&01,&CD,&
5A,&BB,&C9,&3E
160 DATA &17,&CD,&5A,&BB,&3E,&0
2,&CD,&5A,&BB,&C9,&3E,&17,&CD,&
5A,&BB,&3E,&03
170 DATA &CD,&5A,&BB,&C9,&3E,&1
7,&CD,&5A,&BB,&3E,&00,&CD,&5A,&
BB,&C9,&CD,&7B
180 DATA &BB,&C9,&CD,&D0,&BD,&C
9,&CD,&19,&BD,&C9,&06,&03,&CD,&

```


DiscStar 2.1

Dieses Programm beinhaltet viele Vorzüge von anderen Programmen dieser Art und noch einige sehr interessante Extras, so z.B. das Formatieren von einzelnen Tracks, das Kopieren oder Austauschen von einzelnen Sektoren, das Setzen von Attributen (R/O, R/W, \$D, \$\$) und das Wiederherstellen von gelöschten Programmen sowie das Wählen eines beliebigen Users (0-255). Das Programm wurde absichtlich in BASIC geschrieben, um es jederzeit an spezielle Aufgaben anzupassen oder es beliebig zu erweitern.

Als erstes wird Listing 1 abgetippt und unter "DSLADER" gespeichert. Listing 2 wird unter "DISCSTAR" gespeichert. Zum Starten wird RUN "DSLADER" eingegeben. Die REM-Zeilen müssen nicht mit abgetippt werden.

Zuerst erscheint eine Informationszeile mit dem Inhalt
Discstar 2.1 Format: DATA ONLY von Drive a auf Drive a

Auf der mittleren Anzeige befindet sich das gerade verwendete Disketten-Format. Die rechte Hälfte zeigt Ihnen, in welchem Laufwerk sich die Quell- und Zieldisketten befinden (besonders wichtig bei COPY, FORMAT, CAT und EX; bei den Befehlen UNVEIL, CAT und DIR wird das Quell-Laufwerk angesprochen). Nun zu der Befehlseingabe. Es werden die Befehle ohne Komma zwischen Befehl und Wert(en) eingegeben. Manche Befehle können nicht im IBM-Format arbeiten, da die dafür benötigte Erfahrung fehlt. (Wer arbeitet schon im IBM-Format?).

Syntax der Befehle:

FORMAT vt-bt<,n>

Formatiert im Format der Disc im Quelllaufwerk die Diskette im Ziellaufwerk von Track vt bis Track bt. (n ist ein formatierter Track der einzulegenden Disc, d.h. es werden die benötigten Werte von Track n gelesen.)

Anm.: Hiermit lassen sich auch die Spuren 40-41 formatieren und man hat theoretisch 9k mehr Speicher. Aber man braucht ein paar Pokes, damit das Betriebssystem diese nutzen kann.(s.u.) Die CP/M Routine formatiert nur von Track 0-39.

Beispiel: Format 0-41 formatiert eine Disc einschl. Track 40 & 41.

COPY vt-bt, vs-bs<,m>

Kopiert von Track vt bis Track bt und von Sector vs bis Sector bs. Mit diesem Befehl können Sie mit unterschiedlichem Format arbeiten, d.h. die Quell-Disc kann ein anderes Format haben als die Ziel-Disc. Mit diesem Befehl kann man ohne Probleme zwischen zwei Formaten überspielen. Er berücksichtigt, daß im DATA ONLY Format auf dem ersten Track das Directory steht und im CP/M - VENDOR auf dem Dritten. Da dies allerdings zu Komplikationen führen könnte, kann man m auf 1 setzen, dann wird diese Regulierung unterlassen (z.B. wenn man ganze DATA- auf VENDOR/CP/M-Discs bzw. umgekehrt oder einzelne gezielte Teile kopiert). Auch läßt sich CP/M nachträglich auf VENDOR - formatierte Disketten hinzukopieren. Falls die Tracks 0 und 1 nicht beschreibbar sind, lassen diese sich durch "FORMAT" nachträglich formatieren. Die beiden Discs können entweder mit einem oder mit zwei Laufwerken kopiert werden.

Beispiel: COPY 0-39,1-9 kopiert eine mit CP/M-Routine formatierte Disc komplett.

COPY 0-1,1-9 kopiert von einer CP/M Disc die Systemspuren auf eine VENDOR-Disc.

UNVEIL"<u2:>n.z.,u<,h<,g>>

Setzt alle Files mit Name n.z., die vor diesem Befehl auf User u2 liegen, auf den neuen User u (somit kann man auch geschützte Programme löschen, wenn u=229), wenn weitere Parameter angegeben auf Hide (h=1) oder UnHide (h=0) und wenn angegeben auf geschützt (g=1) oder ungeschützt (g=0). Hiermit kann man Files löschen bzw. gelöschte wieder herstellen oder auf einen beliebigen User bringen (0 - 255!). Wildcards gibt es nicht. Alle "*" oder "?" werden einfach weggelassen ("*" = "?").

Beispiel: UNVEIL "229;.BIN" setzt alle gelöschten Binar-Files auf User 0, ungeschützt und unhide (gelöschte Programme ohne Disc-Monitor 'retten!').

BL bl

Rechnet bl in Track und Sector der eingelegten Disc um (formatabhängig, da CP/M und DATA unterschiedliche Ergebnisse erzielen). Mit diesem Befehl kann man den Track und Sektor ausrechnen, auf dem sich ein File befindet. Gut geeignet, falls man mit einem Disc-Monitor das File untersuchen will.

Beispiel: BL 10 errechnet den track und Sektor von Block 10 der eingelegten Disc.

CAT <u>

Es wird der CATALOG auf User u aufgelistet. u kann zwischen 0 und 255 liegen. Auf User 229 / hex &E5 liegen alle gelöschten Files.

Beispiel: CAT 229 zeigt alle gelöschten Files, welche noch nicht im Directory überschrieben sind.

DIR

Listet das komplette DIRECTORY mit allen Files (auch gelöschte), zeigt deren Zustand (z.B. geschützt) und gibt Informationen über Länge, Startadresse und belegte Blocks aus. Hierbei sei anzumerken, daß es sich bei der Bezeichnung "TEIL x" um den Eintrag handelt. Ist ein File nämlich länger als 16k, dann wird es auf mehrere Einträge im Directory mit je 16 k verteilt. Bei ASC,COM und gelöschten Files, sowie bei Einträgen über 1 wird die untere Zeile mit Startadresse, Runadresse usw. weggelassen, da die Angaben dann nicht mehr stimmen. Falls man dennoch die untere Zeile bei gelöschten Files haben möchte, muß man einfach mit "UNVEIL" diese auf einen anderen User bringen. Nach einer vollen Bildschirmseite wird auf einen Tastendruck gewartet. Drückt man eine andere Taste als SPACE, so wird zum Eingabemodus zurückgesprungen, ansonsten wird das Directory weiter gezeigt.

CLS

Löscht den Bildschirm.

EX vt-bt,vs-bs

Vertauscht auf einer Disc vom track vt und bt die Sektoren vs-bs (wenn das Quell- und Ziellaufwerk nicht das gleiche ist, auch Austausch zwischen zwei Disketten möglich!).

Beispiel: EX 0-41,1-4 vertauscht Sektor 1 bis 4 der Tracks 0 und 41.

DR vd\$-ad\$

Setzt Quell- und Ziellaufwerk fest. vd\$ und ad\$ können nur "A" oder "B" sein.

Beispiel: DR A-B setzt das Quell-Laufwerk Drive A und als Ziel-Laufwerk Drive B fest.

"<>" der in Klammern stehende Ausdruck kann wahlweise eingegeben werden (die Klammern werden weggelassen!).

n,z n ist Name, z ist Zusatz (z.B. n= DISCSTAR, Z=BAS)

vt, bt können Werte zwischen 0 und 41 annehmen;

vs, bs können Werte zwischen 1 und 9 annehmen;

bl kann Werte zwischen 1 und 199 annehmen;

h,g stellen entweder 0 oder 1 dar

u.u2 Usernummer 0-255

Der Aufbau des Programms:

Zeile	Bedeutung
10-40	Initialisierung
50-150	Eingabeschleife
170	Befehl: CAT
190-370	Befehl: FORMAT
390-660	Befehl: COPY
680-920	Befehl: UNVEIL
940-950	Befehl: CLS
970-1210	Befehl: EX
1230-1280	Befehl: BL
1300-1340	Befehl: DR
1360-1720	Befehl: DIR
1740-1880	SUBroutine von DIR für Fileinformationen
1900-1910	SUBroutine zum Formatfeststellen
1930-2010	SUBroutine für Kopfzeile
2020	ERROR- und BREAK-Aussprungszeile

Nachtrag:

Das Programm wurde auf einem CPC 664 erstellt. Normalerweise dürfte es beim 464/6128 keine Schwierigkeiten geben, außer bei 'DEC\$('. Dieser Befehl muß beim CPC 464 wegen einem Systemfehler 'DEC\$((' lauten.

Die Variablenliste:

a\$,b\$ enthält Eingabe (b\$ in Kleinbuchstaben)
 aadr Anfangsadresse
 bl Block
 bl\$,bl1\$,bl2\$ enthalten Tracks
 bs bis Sektor
 bt bis Track
 bu Buffer
 dirf Anzahl der DIR-Sektoren
 dirs Directory Sektor
 dirt Directory Track
 dr auf Laufwerk
 dr2 auf Laufwerk

programme

fb Formatbuffer
 fdt formatierter Disc-Track
 fo Format der Disc
 fo2 (wie fo)
 fz speichert Stringposition
 h Zähler für 'COPY'
 hd Flag für versteckt
 hs Hilfsvariable
 hv Hilfsvariable
 hw Hilfsvariable
 i,i2 Position eines Zeichens im String
 j Zählvariable
 kz Zähler
 laenge Länge des Files
 laddr Ladeadresse
 m Flag für Trackversatz ein/aus
 n\$ Name
 ot Nummer des Eintrags im Directory
 ot\$ (wie ot)
 pruefs Prüfsumme v. Filekopf und Flag für ASCII-File
 pz Hilfsvariable
 radr RUNadresse
 rw Flag für schreibgeschützt
 sec augenbl. Sektor
 t augenbl. Track
 t\$,t1\$,t2\$ Namen des/der Files bei 'UNVEIL'

top alter Himem
 v vertikale Cursorposition
 vs von Sektor
 vt von Track
 z Zählvariable
 zb Zwischenbuffer

Die oben beim Befehl FORMAT angesprochenen Pokes, welche den Discspeicher um Sk erhöht, sind folgende:

POKE fl1*24,255

um das Login anzuschalten und

POKE fl1+Laufw,PEEK(fl1*256+Laufw.)+9

um die Kapazität um 9 zu erhöhen,

wobei LAUFW sich aus Drivenummer (A=0/B=1)*64 +5 zusammensetzt. fl1 ist PEEK(&BE42)+PEEK(&BE43)*256.

Zu beachten ist, daß vor Eingabe dieser Pokes einmal CAT oder DIR gemacht werden muß, um sonstige Werte an die Diskette anzupassen. Außerdem muß man, wenn man das Discformat wechselt oder Disketten, die NICHT auf TRack 40/41 formatiert sind, das Login einschalten mit

POKE fl1+24,0

SCHNEIDER LISTING MIT CHECKSUMMEN V 2.0

```

10 *****
*****
11 *
*
12 *          DiscStar 2.1
*
13 *
*
14 * Geschrieben im Sept. '86
von: *
15 *          Peter Woelflick
*
16 *          Bertolt-Brecht-St
r 80 *
17 *          8500 Nuernberg 50
*
18 *          0911 / B1 80 1
*
19 *
*
20 *****
*****
30 MEMORY &A59F
40 FOR adr=&A5A0 TO &A650:READ
a$:a=VAL("&"+a$):POKE adr,a:ps=
ps+a:NEXT
50 IF ps<>20706 THEN PRINT"Fehl
er":STOP ELSE CALL &A5A0:RUN"di
scstar"
100 DATA 00,00,07,01,AC,AS,21,D
D
110 DATA AS,C3,D1,BC,CO,AS,C3,F
1
120 DATA AS,C3,16,A6,C3,26,A6,C
3
130 DATA E1,AS,C3,34,A6,C3,40,A
6
  
```

```

140 DATA 53,45,43,CC,53,45,43,D
3
150 DATA 53,49,44,CC,44,46,45,4
B
{CC} 160 DATA 4C,45,D2,46,4F,52,4D,4
1
{HD} 170 DATA D4,50,4F,D3,00,00,00,0
0
{CM} 180 DATA 00,FE,01,CO,21,72,CA,2
2
{OJ} 190 DATA A0,A5,DD,7E,00,DF,A0,A
5
{DH} 200 DATA C9,FE,04,CO,21,66,C6,2
2
{MH} 210 DATA A0,A5,DD,66,03,DD,6E,0
2
{CF} 220 DATA DD,56,04,DD,4E,06,06,0
0
{IB} 230 DATA 1E,00,DD,7E,00,FE,01,2
0
{DA} 240 DATA 01,1C,DF,A0,A5,C9,FE,0
4
{GL} 250 DATA CO,3E,07,32,A2,AS,21,4
E
{CD} 260 DATA C6,22,A0,A5,18,D4,FE,0
1
{BN} 270 DATA CO,21,5D,C5,22,A0,A5,D
D
{DJ} 280 DATA 5E,00,1B,DE,FE,04,CO,2
1
{IJ} 290 DATA 52,C6,22,A0,A5,C3,FA,A
5
{NP} 300 DATA FE,02,CO,21,63,C7,22,A
0
{NJ} 310 DATA AS,DD,5E,00,DD,56,02,1
B
{BJ} 320 DATA C1
ENDE DES LISTINGS
{CB}
  
```

Teil 2

SCHNEIDER LISTING MIT CHECKSUMMEN V 2.0

```

10 ON ERROR GOTO 2020:MODE 2:DE
FINT a,c-z:top!=HIMEM
20 MEMORY &3379:DIM b1(16):INK
0,9:INK 1,25:BORDER 9
30 bu=&339F:fb=&337A:zb=bu+&A00
40 GOSUB 1900:ON BREAK GOSUB 20
20
50 LINE INPUT">":a$:b$=LOWER$(a
$):z=0
60 IF LEFT$(b$,3)="cat"THEN 170
70 IF LEFT$(b$,6)="format"THEN
190
80 IF LEFT$(b$,4)="copy"THEN 39
0
90 IF LEFT$(b$,6)="unveil"THEN
680
100 IF LEFT$(b$,3)="cls"THEN 94
0
110 IF LEFT$(b$,2)="ex"THEN 970
120 IF LEFT$(b$,2)="b1"THEN 123
0
130 IF LEFT$(b$,2)="dr"THEN 130
0
140 IF LEFT$(b$,3)="dir"THEN 13
60
150 IF b$=""THEN 50 ELSE PRINT"
?":b$:GOTO 50
160 *** CAT
170 POKE &A701,VAL(MID$(a$,4))M
OD 256:CAT:GOTO 50
180 *** FORMAT
190 i=INSTR(8,a$,"-"):IF i=0 TH
EN 150
200 vt=VAL(MID$(a$,8))
210 bt=VAL(MID$(a$,i+1))
220 PRINT"Eine formatierte Disc
einlegen":CALL &BB06:GOSUB 190
0
230 i=INSTR(8,a$,"-"):IF i=0 TH
EN fdt=0:GOTO 250
240 fdt=VAL(MID$(a$,i+1))MOD 41
  
```

```

250 fb=&337A:iPOS,fdt,dr
260 FOR t=1 TO 20
270 !SIDL,dr:FOR hw=0 TO 3:POKE
fb+hw,PEEK(&DE4F+hw):NEXT
{OO}
{GN} 280 fb=fb+4:NEXT:fb=&337A
290 IF PEEK(fb+2)=1+fo THEN 310
ELSE fb=fb+4
{LO}
300 IF fb<&337A+&28 THEN 290 EL
SE PRINT"Fehler":GOTO 40
{EA}
310 PRINT"Die zu formatierende
Disc einlegen":CALL &BB06
{II}
{KB} 320 FOR t=vt TO bt
330 FOR j=0 TO 8:POKE fb+4*j,t:
NEXT
{IG}
340 LOCATE 1,24:PRINT"Formattin
g Track"+STR$(t);
{OI}
{PA} 350 !FORMAT,fo+1,t,fb,dr2
360 NEXT:iSECL,1+fo,0,bu,0
{CB}
{FP} 370 GOTO 40
380 *** COPY
{DL}
{DD} 390 GOSUB 1900:FOR hw=6 TO LEN(
a$)
400 IF MID$(a$,hw,1)<>","THEN 4
20
{HN}
{AH} 410 z=z+1:fz(z)=hw
420 NEXT:IF z=0 THEN 150
{FF}
{PJ} 430 IF z=1 THEN fz(2)=LEN(a$)
440 IF z=2 THEN m=VAL(MID$(a$,f
z(2)+1))
{DI}
450 i=INSTR(a$,"-"):IF i=0 THEN
150
{IK}
{OP} 460 vt=VAL(MID$(a$,6))
470 bt=VAL(MID$(a$,i+1),fz(1)-i)
{JE}
{BO} 480 i2=INSTR(fz(1)+1,a$,"-")
490 IF i2=0 THEN 150
{DI}
{IA} 500 vs=VAL(MID$(a$,fz(1)+1))
510 bs=VAL(MID$(a$,i2+1))
{NE}
{CC} 520 f=fz(2)+1:fo2=fo:IDFEHLER,0
530 FOR t=vt TO bt:LOCATE 1,25:
PRINT"Original einlegen";
{OD}
{EA} 540 CALL &BB06:h=5:WHILE t+h>bt
:h=h-1:WEND
{FM}
550 bu=&339E:FOR ht=t TO t+h:LO
CATE 1,24:PRINT"Read Track "+ST
R$(ht)" ":FOR s=vs TO bs
{JB}
{GL} 560 !SECL,s+fo2,ht,bu,dr:bu=bu+
&200
{CH}
  
```



```

570 NEXT: NEXT
580 PRINT "Kopie einlegen ";: CA
LL &BB06: IF t=vt THEN GOSUB 190
0
590 IF m THEN hw=0: GOTO 630
600 IF fo=&40 AND fo2=&C0 THEN
hw=-2
610 IF fo2=&40 AND fo=&C0 THEN
hw=2
620 IF fo2=fo THEN hw=0
630 bu=&339E: FOR ht=t TO t+h: LO
DATE 1,24: PRINT "Write Track"+ST
R$(ht-hw))" ": FOR s=vs TO bs
640 !SECS,s+fo,ht-hw,bu,dr2: bu=
bu+&200
650 NEXT: NEXT: t=ht-1: NEXT
660 GOTO 30
670 *** UNVEIL
680 i=INSTR(6,a$,CHR$(34)): IF i
=0 THEN 150 ELSE i=i+1
690 i2=INSTR(i,a$,CHR$(34)): IF
i2=0 THEN 150
700 i3=INSTR(i,a$,":"): IF i3>0
THEN user2=VAL(MID$(a$,i)): i=i3
+1
710 t$=UPPER$(MID$(a$,i,2-i)):
i=INSTR(t$,":"): kz=0: IF i THEN
720 ELSE 730
720 t2$=MID$(t$,i+1,3): t1$=LEFT
$(t$,i-1): GOTO 740
730 t1$=LEFT$(t$,8): t2$=""
740 GOSUB 1900: IF fo=&40 THEN d
irt=2
750 FOR dirs=fo+1 TO fo+4: !SECL
,dirs,dirt,bu,dr
760 kz=0: FOR hw=12+1 TO LEN(a$)
770 IF MID$(a$,hw,1)="," THEN kz
=kz+1: k(kz)=hw+1
780 NEXT: user=VAL(MID$(a$,i2+2)
): rw=0: hd=0
790 IF kz>1 THEN IF MID$(a$,k(2
),1)="1" THEN rw=1
800 IF kz>2 THEN IF MID$(a$,k(3
),1)="1" THEN hd=1
810 FOR hw=bu TO bu+511 STEP &2
0
820 FOR hv=hw+1 TO hw+11: n$=n$+
CHR$(PEEK(hv) AND 127): NEXT
830 IF n$="eeeeeeeeee" THEN 920
840 IF NOT(LEFT$(n$,LEN(t1$))=t
1$ AND MID$(n$,9,LEN(t2$))=t2$) T
HEN 920
850 IF PEEK(hw)<>user2 AND i3>0
THEN 920
860 POKE hw,user: IF hd=1 THEN P
OKE hw+10,PEEK(hw+10) OR 128
870 IF hd=0 THEN POKE hw+10,PEE
K(hw+10) AND 127
880 IF rw=1 THEN FOR hv=hw+1 TO
hw+9: POKE hv,PEEK(hv) OR 128: NE
XT
890 IF rw=0 THEN FOR hv=hw+1 TO
hw+9: POKE hv,PEEK(hv) AND 127: N
EXT
900 IF rw=1 THEN POKE hw+11,PEE
K(hw+11) OR 128
910 IF rw=0 THEN POKE hw+11,PEE
K(hw+11) AND 127
920 n$="" : NEXT: !SECS,dirs,dirt,
bu,dr: NEXT: GOTO 40
930 *** CLS
940 GOSUB 1930
950 GOTO 50
960 *** EX
970 GOSUB 1900: FOR hw=3 TO LEN(
a$)
980 IF MID$(a$,hw,1)<>"," THEN 1
000
990 z=z+1: fz(z)=hw
1000 NEXT: IF z=0 THEN 150
1010 IF z=1 THEN fz(2)=LEN(a$)
1020 i=INSTR(a$,"-"): IF i=0 THE
N 150
1030 vt=VAL(MID$(a$,3))
1040 POKE &9100,PEEK(&9106): POK
E &9101,PEEK(&9105)
1050 i2=INSTR(fz(1)+1,a$,"-")
1060 IF i2=0 THEN 150
[EP] 1070 vs=VAL(MID$(a$,fz(1)+1))
1080 bs=VAL(MID$(a$,i2+1))
1090 f=fz(2)+1
[OD] 1100 bu=&339E
[PL] 1110 LOCATE 1,24: PRINT "Read Tra
ck"+STR$(vt): FOR s=vs TO bs
[PO] 1120 !SECL,s+fo,vt,bu,dr: bu=bu+
&200
[CL] 1130 NEXT
[FL] 1140 LOCATE 1,24: PRINT "Read Tra
ck"+STR$(bt): FOR s=vs TO bs
[AC] 1150 !SECL,s+fo,vt,bu,dr2: bu=bu
+&200
[CE] 1160 NEXT
[EO] 1170 bu=&339E: LOCATE 1,24: PRINT
"Save Track"+STR$(bt): FOR s=vs
TO bs
[CA] 1180 !SECS,s+fo,bt,bu,dr: bu=bu+
&200
[AH] 1190 NEXT: LOCATE 1,24: PRINT "Sav
e Track"+STR$(vt): FOR s=vs TO b
s
[II] 1200 !SECS,s+fo,vt,bu,dr2: bu=bu
+&200
[PH] 1210 NEXT: GOTO 40
1220 *** BL
[HO] 1230 b1=VAL(MID$(a$,3)): IF b1<0
OR b1>189 THEN 40 ELSE b1=b1*2
-1
[LF] 1240 IF fo=&40 THEN sec=(b1 MOD
9)+1: tr=(b1\9)+2
[GB] 1250 IF fo=&C0 THEN sec=(b1 MOD
9)+1: tr=b1\9
[JM] 1260 IF fo<>&C0 AND fo<>&40 THE
N PRINT "Disc Format Error": GOTO
40
[MF] 1270 PRINT "Sektor: "+STR$(sec): P
RINT "Track: "+STR$(tr)
[CH] 1280 CALL &BB06: GOTO 50
1290 *** DR
[AH] 1300 i=INSTR(3,a$,"-"): IF i=0 T
HEN 150
[PK] 1310 dr2=(ASC(MID$(a$,i+1))-65)
MOD 2
[EA] 1320 dr=(ASC(MID$(a$,4))-65) MOD
2
1330 GOSUB 1930
1340 GOTO 50
[HK] 1350 *** DIR
1360 dirt=0: dirs=0: dirf=0: z=0: n
$="" : b1=0: user=0: ot=0: zs$="" : la
enge=0: b1$=""
[LN] 1370 GOSUB 1900: bu=&339E
1380 MODE 2: dirs=fo: IF fo=&40 T
HEN dirt=2
[EI] 1390 IF fo=&C0 THEN dirt=0
1400 IF fo=0 THEN PRINT "Leider
nicht lesbar": GOTO 40
[KC] 1410 FOR dirf=0 TO 3: dirs=dirf+
1
[OB] 1420 !SECL,dirs,dirt,bu,dr: hs=0
1430 FOR hw=bu TO bu+511 STEP &
20
[IM] 1440 GOTO 1740
[HC] 1450 hs=hs+1: FOR hv=hw+1 TO hw+
11
[KI] 1460 n$=n$+CHR$(PEEK(hv) AND 127
)
[FB] 1470 NEXT: n$=LEFT$(n$,8)+" : MI
D$(n$,9)
[BI] 1480 FOR hv=hw+&10 TO hw+&1F
[MA] 1490 b1=PEEK(hv): IF b1>0 THEN z
=z+1: b1(z)=b1
[EC] 1500 NEXT: FOR hv=1 TO z: b1$=b1$
+" "+HEX$(b1(hv),2): NEXT
[NC] 1510 b11$=LEFT$(b1$,24): b12$=MI
D$(b1$,25)
[OC] 1520 user=PEEK(hw): laenge=PEEK(
15+hw)
[NI] 1530 ot=PEEK(12+hw): ot$="TEIL"+
STR$(ot+1)
[LE] 1540 v=VPOS(#0): FOR hv=1 TO 12:
IF ASC(MID$(n$,hv,1))<32 THEN 1
680
[IC] 1550 NEXT
[CF] 1560 LOCATE 1,v: PRINT CHR$(133)
+ n$+CHR$(138): n$=""
[LA]
[KA]
[ED] 1570 LOCATE 1,v+1: PRINT STRING$
(14,CHR$(131)):
[PC] 1580 LOCATE 18,v: PRINT ot$:
[PB] 1590 LOCATE 28,v: PRINT zs$:
[CN] 1600 LOCATE 38,v: PRINT "USER "HE
X$(user,2):
[PH] 1610 LOCATE 49,v: PRINT "Blocks "
b11$
[IN] 1620 v=VPOS(#0)
[BH] 1630 LOCATE 56,v: PRINT b12$: v=v
+1: IF ot<>0 OR user=229 OR prue
fs!>0 THEN PRINT: GOTO 1680
[IM] 1640 LOCATE 2,v: PRINT "Anfadr. "D
EC$(aadr!,"#####"):
[BD] 1650 LOCATE 18,v: PRINT "RUNadr.
"DEC$(radr!,"#####"):
[BC] 1660 LOCATE 35,v: PRINT "Records
"HEX$(laenge,2):
[IK] 1670 LOCATE 49,v: PRINT "Laenge "
DEC$(ladr!,"#####") Byte /
"DEC$(INT((ladr!+128)/1024)+1,
"##") kByte"
[LM] 1680 PRINT "STRING$(78,"-"): IF
VPOS(#0)>21 THEN LOCATE 36,25:
PRINT "<TASTE>": WHILE n$="" : n$=
INKEY$: WEND: IF n$<>" " THEN 40 E
LSE CLS
[OJ] 1690 z=0: n$="" : b1=0: user=0: ot=0
: zs$="" : laenge=0: b1$="" : NEXT
[MC] 1700 IF hs=0 THEN dirf=4
[NP] 1710 NEXT
[AN] 1720 LOCATE 36,25: PRINT "<TASTE>
": CALL &BB06: CLS: GOTO 40
[KE] 1730 'Sub -Dir-
[KB] 1740 IF PEEK(hw+9)>127 THEN zs$
="R/O " ELSE zs$="R/W "
[CG] 1750 IF PEEK(hw+10)>127 THEN zs
$=zs$+"$ " ELSE zs$=zs$+"$D"
[NG] 1760 IF PEEK(hw+16)=229 THEN 18
50
[PA] 1770 sec=(PEEK(hw+16)*2) MOD 9+1
1780 track=(PEEK(hw+16)*2)\9: IF
fo=64 THEN track=track+2
[LF] 1790 !SECL,sec+fo,track,zb,dr
1800 pruefs!=0: FOR pz=1 TO 66: p
ruefs!=pruefs!+PEEK(pz+zb): NEXT
1810 IF pruefs!<>(PEEK(zb+67)+P
EEK(zb+68)*256) THEN pruefs!=1 E
LSE pruefs!=0
[KE] 1820 aadr!=PEEK(zb+22)*256+PEEK
(zb+21)
[IC] 1830 laddr!=PEEK(zb+25)*256+PEEK
(zb+24)
[DG] 1840 radr!=PEEK(zb+27)*256+PEEK
(zb+26): IF radr!=0 THEN radr!=a
adr!
[GI] 1850 pz=0: FOR hf=16 TO 31: pz=
pz+PEEK(hw+hf): NEXT
[KI] 1860 IF pz<>3664 THEN 1450
[KB] 1870 IF hw<bu+512-&20 THEN hw=h
w+&20: GOTO 1850
[BD] 1880 GOTO 1700
[DI] 1890 'Sub Format feststellen
[NC] 1900 !DFEHLER,1: WINDOW#0,1,80,2
0,20: PEN 0: !DIR
1910 fo=PEEK(&A9F)-1: PEN 1: !DF
EHLER,0: GOSUB 1930: RETURN
[AP] 1920 'Sub Kopfzeile
[AB] 1930 MODE 2: LOCATE 53,1: PRINT "v
on DRIVE": LOCATE 68,1: PRINT "au
f DRIVE":
[GM] 1940 LOCATE 63,1: PRINT CHR$(65+
dr): POKE &A700,dr
[GL] 1950 LOCATE 78,1: PRINT CHR$(65+
dr2):
[FA] 1960 LOCATE 1,1: PRINT "DiscStar
2.1"
[BL] 1970 LOCATE 23,1: PRINT "Format :
Unbekannt"+STRING$(9,CHR$(8)):
[HO] 1980 IF fo=0 THEN PRINT "IBM
"
[BF] 1990 IF fo=64 THEN PRINT "CP/M /
VENDOR"
[GE] 2000 IF fo=192 THEN PRINT "DATA
ONLY"
[AC] 2010 RETURN
[AD] 2020 MODE 1: CALL &BC02: MEMORY t
op: PEN 1: !USER,0: !A: PRINT "Disc
Star 2.1 beendet"
[GE]

```


* Kleinanzeigen * Kleinanzeigen * Kleinanzeigen *

Biete Hardware

 *** Für C16/116/PLUS 4 ***
 * 3+1 Software in C16/116 *
 * 198 DM *
 * 3+1 Dito in Deutsch 248 DM *
 * Umbauset C16/116 64K 50 DM *
 * Umbau C16/116 64K m. Gar. *
 * 100 DM *
 * LOGO für C64 159 DM:
 * Jetzt 49,50 DM *
 * Elektronik Datasette *
 * nur 98,50 DM *
 * Elektronik-Technik-Peters *
 * Tannenweg 9, *
 * 2351 Trappenkamp *

Verkaufe fast neuen C-16 + Datasette + 2 Bücher + Joystick + rund 20 Programme für 400,- DM. Mehr Information: Tel. 0751/92972
 Schenke kostenlos dazu ZX-81 + 16K + 20 Programme.

Verkaufe C-116 mit Datasette Handbuch und Basiclernkassette für 85,- DM ca. 10 Monate alt. Tel. 05924/1279.

C16 umschaltbar auf 64K mit der Datasette und Datenmithörgerät, viele Spiele und Programme dazu, alles wie neu aus 1986! VB: 199,- DM
 Boye, Tel.: 04522/4134

C16/116 RAM-Umbau 64 KBYTE in 1. Tag für 100 DM nur bei mir kostenl. Durchs. auch Reparatur *** beim Computerfachmann ***
 Noch Fragen? Freiumschlag an: Elektronik-Technik-Peters
 Tannenweg 9, 2351 Trappenkamp

Verkaufe C-16 mit Datasette und ein TOP-GAME, Intern. Karate um 250 DM (C16 2 Monate alt)
 Tel.: 06541/407 Austria (Martin verlangen!)

 * Für C16/116/Plus 4 Preiswert *
 * Turbotape Super (64K) 19,50 DM *
 * Elektronik Datasette 98,50 DM *
 * 3+1 Software Englisch *
 * 198,00 DM *
 * 3+1 Software Deutsch *
 * 248,00 DM *
 * Plus 4 geprüfte Ware 298,00 DM *
 * Plus 4 dito deutsch 348,00 DM *
 * Floppy 1551 Restp. 389,00 DM *
 * Elektronik-Technik-Peters *
 * Tannenweg 9, *
 * 2351 Trappenkamp *

COMMODORE VC-20 zu verkaufen + 25 Modulspielen und 75 auf Cassette. Spiele auch einzeln erhältlich. Tel. 0251/788451.

 * ++Achtung, Achtung!++ *
 * Verkaufe CPC 464, komplett mit *
 * Handbuch, Democassette und *
 * weiteren Programmen. Fast neu- *
 * wertig, für nur 500,- DM. Bitte *
 * schreiben Sie an: *
 * Günter Seebauer, Sportplatz 12, *
 * 8852 Bayerdilling, *
 * Tel.: 09002/3840 *

Verkaufe C16 + Hand u. Lehrbuch + Datasette + 9 Spiele, Gesamtpreis 270 DM, Neuwert 480 DM
 Anfragen: Michael Schulz,
 Tel.: 02101/273420

PLUS/4 + Floppy + Datasette + Interface Neu! Preis: VB. Auch einzeln!!!
 Zuschriften an:
 Frank Reichmann, Kirschenstr. 21,
 5309 Meckenheim, Tel.: 02225/3716

Turbo-PC's, 4 MgH-8 MgH umsch., ohne Schneider-Macken zum Toppreis.
 Infos anfordern: 0561/772408

CPU 464/664
 Suche und tausche Software. Verkaufe Steve D. Smooker, Macadam Bummer a 20,- DM. Suche Dragons Cair usw. Michael Schliitt, Fasanenweg 1, 4459 Uelsen.

Suche Druckknopf für MPS-802 oder defekten Drucker MPS-802. Zahle Gut. Compiliere Basic-Programme mit Austro-Speeder CCA. 3 - 10mal schneller, gegen geringen Kostenbeitrag. Anfragen und Information unter 09353/3864.
 Ps. Tausche C-16-Plus/4 Programme auf Diskette!!!

C-64 Besitzer sucht gebrauchte Floppy 1541!
 Da finanziell sehr am Ende, bitte Preise bis 30,- DM. 0991/25129, nach Roland fragen!!!

Ich suche einen gebrauchten Curran-Microspeech-Sprachsynthesizer. Nur mit Anleitung! Angebote an: R. Buck, Paukengasse 2, 7900 Ulm.

Suche noch guterhaltenen C-64 mit Floppy. Zahle bis 600 DM. Tel.: 06541/497 Austria (Martin verlangen)
 Martin Sommerbichler,
 Hotel Tirolerhof, A-5754 Hinterglemm

***** HILFE !!! *****
 Suche C-16 + 64K Programme Nur Diskette !!!
 An: Hubert Mühlbacher, Plattenberg 3, 8221 Waging, Tel.: 08681/772

Suche



So bestellen Sie:

Der Mindestbestellwert von Software liegt bei 15.00 DM!

Sie können das Geld in bar oder per Scheck Ihrer schriftlichen Bestellung beilegen.

Bei Nachnahme-Versand werden 4.00 DM für Porto-Kosten erhoben.

Auslandsbestellungen werden nur per Vorkasse entgegengenommen!

Rufen Sie uns doch einfach einmal an! Unsere Telefone sind wochentags ab 9.30 Uhr für Sie zu erreichen!

SOFTWARE-VERSAND

ANDREAS BACHLER
 POSTFACH 429
 D-4290 BOCHOLT

(0 28 71) 18 30 88

Wir haben die Software, die Sie suchen!

Commodore 64/128

	Kassette	Diskette
Alleykat	29,95 DM	38,95 DM
Die Erbschaft	68,95 DM	78,95 DM
Eis & Feuer		68,95 DM
Hacker II	38,95 DM	54,95 DM
Knight Games	38,95 DM	38,95 DM
1942	32,95 DM	44,95 DM
Parallax	34,95 DM	
Room 10	29,95 DM	
Sanxion	32,95 DM	
Super Cycle	32,95 DM	44,95 DM
Tass Times		58,95 DM
Thrust	9,95 DM	
Trinity		98,95
Vera Cruz	48,95 DM	58,95 DM
Vietnam	32,95 DM	44,95 DM
Way of the Expl. Fist II	32,95 DM	
World Games	32,95 DM	44,95 DM

Schneider CPC (464, 664, 6128)

	Kassette	Diskette
Eden Blues	32,95 DM	44,95 DM
Jack, the Nipper	32,95 DM	32,95 DM
Saboteur	32,95 DM	32,95 DM
Star Strike II	32,95 DM	
Tarzan	32,95 DM	44,95 DM
Tau Ceti		32,95 DM
Tomahawk	32,95 DM	48,95 DM
Utchimata	32,95 DM	44,95 DM
Winter Games	32,95 DM	54,95 DM

Commodore 16/116/PLUS 4

	Kassette
ACE	32,00 DM
ACE (+ 64K)	34,95 DM
Bandits at Zero	14,95 DM
Bomb Jack	29,00 DM
Classics III	32,95 DM
Computer-Hit (10 Spiele)	32,95 DM
Finders Keepers	9,95 DM
Formula One Sim.	9,95 DM
Frank Bruno's Boxing	25,00 DM
Ghosts'n'Goblins	29,00 DM
G-Man	9,95 DM
International Karate	19,95 DM
Jack Attack (Modul)	14,95 DM
Kikstart	9,95 DM
Matrix/Laser Zone	29,00 DM
Monty on the Run	9,95 DM
Paint Box	25,00 DM
Pod	9,95 DM
Project Nova	25,00 DM
Quiwi (+ 64K)	29,00 DM
ROM-Listing	29,00 DM
Space Pilot	19,95 DM
Speed King	9,95 DM
Text Manager (Disk)	38,95 DM
Text Manager (Kass.)	38,95 DM
Trail Blazer	19,95 DM
Winter Olympiade	29,00 DM
Yie Ar Kung-Fu	29,00 DM

Fordern Sie kostenlos Informationsmaterial über unser Liefer-Programm für folgende Computer an: COMMODORE 64/128, COMMODORE 16/116/PLUS 4, SCHNEIDER 464/664, ATARI 130XE/260ST/520ST/800XL, SINCLAIR SPECTRUM und MSX.
 Bitte geben Sie immer Ihren Computertyp an!



COMPUTERSOFT JONIGK

C64	Cass.	Disk	C16/Plus 4	Disk	Cass.
Go for Gold	14,90	24,90	über 220 verschiedene Programme auf Lager ab	(Plus 4)	49,- 42,-
Yie ar Kung Fu	32,-	39,-	ACE (Air Combat Emulator)	(Plus 4)	49,- 32,-
Pub Games	39,-	39,-	CSJ Games 1 (3 Spiele)	(Plus 4)	39,-
Alleykat	36,-	49,-	Mercenary	(Plus 4)	39,-
Paperboy	36,-	54,-	Bridgehead	(Plus 4)	39,-
Shanghai	54,-	54,-	Saboteur	(Plus 4)	39,-
Movie Monster	34,-	54,-	Scoby Doo	32,-	Dangerzone 9,90
VIZAWRITE	neuer Preis	98,-	Grandmaster	32,-	Into the Deep 9,90
C128			Bounder/Planet Search	29,-	Torpedo Alley 9,90
The Last V8		24,90	Fighting Warriors	24,-	G-Man 9,90
Thai Boxing		49,-	Space 2	19,-	Prospector Pete 9,90
Finanzbuchhaltung		198,-	Aardvark	19,-	Harvey Headbanger 9,90
VIZA CLASSIC (Deutsch)	neuer Preis	298,-			
Amiga			Vorsicht!!!		
Arena		109,-	Über Sog könnte Sie erfassen, beim Laden und Speichern mit CSJ		
Jewels of Darkness		69,-	TURBO TAPE + AZIMUTH TAPE da 12facher Geschwindigkeit.		
Shanghai		98,-	Siehe auch Test in COMPUTE MIT 1/86		DM 29,90
Finanzbuchhaltung		198,-			
VIZAWRITE AMIGA	neuer Preis	498,-			

Mindestbestellwert DM 30,-



3000 Hannover 1
An der Tiefenriede 27
Tel.: Service 05 11/88 63 83
CSJ COMPUTERSOFT JONIGK

Riesenauswahl an Software
sofort CSJ NEWS anfordern
bitte Computertyp angeben
Händleranfragen erwünscht

Plus 4: Suche Speichererweiterung zum Selbstbau. Wer kann helfen? Biete Hard- und Software für C-16 und Plus 4. Tel.: 04209/5390

Suche zuverlässigen Tauschpartner, 100 %ige Antwort an:
Christian Doeres, Fridrichstr. 5d, 5550 Bernkastel-Kues

Kontakte

***** Hallo C-16 Freunde *****
Wir suchen noch Mitglieder für unseren C-16 Spitzenclub!
Info bei: Markus Schaper
Muldweg 7, 3300 Braunschweig

Wer tauscht Software für CPC 464, kaufe auch billige Spiele. Mit Liste an: Jörg Artischewski, Ilsenburgerstr. 56b, 3388 Bad Harzburg
Antworte garantiert!

***** Größter C16/116/ + 4 Club *****
Biete einen tollen Service u.a. Fachzeitschrift auf Datenträger. Informationen gegen 1,30 Porto!
Hacker, Im Wingert 10, 5440 Mayen 14

Hallo 4132 Kali (Richtig gehört)
Suche C16/116 und Plus 4 Anwender im Raum Lintfort, zwecks Software- u. Informationsaustausch.
Detlef Hoppe ab 21 Uhr: 02842/5847

C16/116/ + 4-Anwender und Einsteiger
Endlich ein Club speziell für uns. Kostenlose Informationen bei:
hrc e.V., Laschinsky, Bauerland 15, 4800 Bielefeld 1

Verschiedenes

Für C16/116/Plus 4 kostenlos!
Infos-Tips-Angebote-Umbauten
*** vom Computerfachmann ***
Erweiterungen + Reparaturen
Noch Fragen? Freiumschatz an:
Elektronik-Technik-Peters
Tannenweg 9, 2351 Trappenkamp

Plus 4: Deutsche Schreibmaschinen-tastatur mit allen Umlauten und Sonderzeichen auf E-PROM. Einfacher Austausch. In allen Betriebsarten nutzbar. Tel.: 04209/5390

* **WER SUCHT** *
* VC-20 intern + 19 Compute *
* 60,- DM *
* Tastatur (einwandfrei) 90,- DM *
* Netzgerät VB 50,- DM *
* Platine (defekt) VB 60,- DM *
* Datensette 1530 50,- DM *
* 32 KB Erw. + Prg. 110,- DM *
* Schnell melden unter *
* Tel.: 07822/19768. Vielen Dank! *
* PS: Ralf zahlt auch Porto *

* **DISKETTEN** *
* 5 1/4", 48 tpi, DM 0,99, 2d *
* 3 1/2", 135 tpi, DM 3,99, 1DD *
* auch andere, bew. Garantie *
* Allg. AUSTRO-AG, Ringstr. 10 *
* 8057 Eching, Tel:08133/6116 *

Biete Software

***C16* Games (Topball, Domino, IRA, Cave, Diggy, Diam, Bacugammon, Rom) + Kassette für 10,- DM + Porto.**
Info gegen Rückporto bei Horst Meyer, Overgünne 89, 4600 Dortmund 30.

Software für:
C16/116, C64/128, Spectrum und ZX 81, Atari, Svi 318/328, Laser 110, 210, 310, VZ 200, ab 7,50 DM, Computer angeben. Kostenlos Info: Firma Jupitersoft, Höttinger Str. 34b, 8836 Ellingen.

***** C16/116-Grundversion*****
Textadventure (Eigenproduktion)
Werner Barwig - Tape - Nach Lader
Bodelschwingstr. 19, - Vorkasse -
6418 Hünfeld 1 - 10-DM-Schein.

Achtung! VC-20 Besitzer!
Supersoftware zu unglaublichen Preisen! Info kostenlos!
X-TRA Datendienst, Colmarer Str. 57, 7000 Stuttgart 40.

Wegen Systemwechsel biete ich meine Programm-Sammlung an für C64, suche Amiga-Programme. Liste anfordern bei: Jürgen Klein, Kürtnersr. 46, 5060 Bergisch Gladbach 2

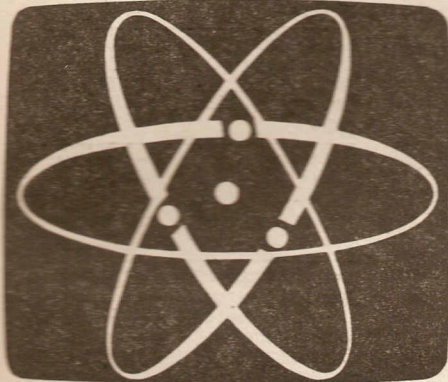
* **GEISLER GmbH: Hotline (0211)** *
* 578081 4000 Düsseldorf 11 *
* Leostr. 1 : Prost Neujahr, lieber *
* Leser! Hier wieder ein Teil *
* unserer Angebote: GPC 20-II *
* XT, kompl. Rechner, 640KB, 20 *
* MB Festplatte, Floppy, Service, *
* Garantie, nur DM 2.499, *
* (unglaublich aber wahr): *
* **COMMODORE AT** und *
* **COMMODORE AMIGA** zu un- *
* glaublichen Preisen. *
* Anrufen und staunen : 20MB *
* Festplatte, Controller, Kabel, *
* wirklich nur DM 1.088,- (toll *
* nicht?) : NEC P& 24 *
* Nadeldrucker, der leise ALLES *
* KÖNNER, DM 1.299,- (da lacht *
* das Kundenherz) : STAR *
* Drucker NL 10 DM 749 (echt *
* super!) : WORD PERFECT 4.1, *
* deutsch, Originalware DM *
* 1.089,- (Hurral) : GEISLER'S *
* **SOFTWARE KATALOG**, die um- *
* fassende Marktübersicht mit *
* **SPEZIALPREISEN**: Schutzge- *
* bühr DM 5,- (Schein oder Brief- *
* marken) : GEISLER ist Ver- *
* tragshändler für COMMODORE, *
* TANDON, BROTHER, THOM *
* SON, Sonderangebotsliste GRA- *
* TIS anfordern! : Beachten Sie *
* auch weiterhin unsere Kleinan- *
* zeigen mit den kleinen Preisen! *
* Wir wünschen Ihnen ein gutes *
* neues Jahr : Ihre GEISLER *
* GmbH : Hotline (0211) 578081 *

Commodore C16/116 Plus/4
Hardcopy - Hardcopy - Hardcopy
druckt Ihre Bildschirmgrafik in max. 60
Sek. auf ca. DIN A4 Größe aus. Für 16K
und 64K Version geeignet. Als Kassette
oder Diskette für nur 19,-DM. Porto per
NN bei: Frank Schmitz, Kurzer Weg 1,
5206 Neunkirchen 2, Tel. 02247/8111.

Biete User-Software für C16 zu
günstigen Preisen. Info gegen
Rückporto bei: Frank Gustik, Am
Ginsterberg 42, 5110 Alsdorf.

C16/Plus/4! Anwender und Spiel-
programme in großer Auswahl für nur
1,- DM/Stück. Liste gegen Rückporto
bei: Jürgen Roumen, Eichenstr. 49,
4054 Nettetal 1.

Zwei Themen - ein Ereignis:



Hobby-tronic

10. Ausstellung für Funk- und Hobby-Elektronik

COMPUTER- SCHAU

3. Ausstellung
für Computer,
Software
und Zubehör

Dortmund
18. - 22. Februar 1987

Die umfassende Marktübersicht für Hobby-Elektroniker und Computeranwender, klar gegliedert:

In Halle 5 das Angebot für CB- und Amateurfunker, Videospiele, DX-er, Radio-, Tonband-, Video- und TV-Amateure, für Elektro-Akustik-Bastler und Elektroniker. Mit dem Actions-Center und Laborversuchen, Experimenten, Demonstrationen und vielen Tips.

In Halle 6 das Superangebot für Computer-anwender in Hobby, Beruf und Ausbildung. Dazu die „Computer-Straße“ als Aktionsbereich, der Wettbewerb „Jugend programmiert“ und die Stände der Computerclubs.

Ausstellungsgelände Westfalenhallen Dortmund täglich 9.00-18.00 Uhr

HARDWARE-SERVICE

Nachdem der Tronic-Verlag mit großem Erfolg eine 64K-Speichererweiterung für den C-16/116 in einer der vorangegangenen Ausgaben von »COMPUTE MIT« angeboten hat, möchten wir auch jetzt wieder auf ein neues Hardware-Angebot hinweisen. Wir planen für den Anfang des Jahres 1987 den Ausbau einer Speichererweiterung auf 256K-Byte. Dazu müssen wir jedoch von Ihnen wissen, ob bei dem User überhaupt ein Interesse dafür vorhanden ist, die Speicherkapazität des C-16/116, C-64 und Plus 4 auf die eines ausgewachsenen Personalcomputers aufzurüsten. Von der technischen Seite sind alle Probleme gelöst, so daß eigentlich sofort mit der Serienproduktion begonnen werden könnte. Wir möchten Sie deshalb bitten uns umgehend mitzuteilen, ob und an welcher Speichererweiterung Sie interessiert sind. Die genaue Aufstellung der Erweiterungen für die einzelnen Systeme und die entstehenden Kosten können Sie der nachfolgenden Gliederung entnehmen:

C-16/116 aufgerüstet auf 256K-Byte

als Bausatz

(Lötungen müssen selbst vorgenommen werden, Bauanleitung und Zubehör wird geliefert) 223,— DM

zum Umrüsten

(Sie senden Ihren Computer zum Umrüsten an eine Firma ein)
263,— DM

C-64/Plus 4 aufgerüstet auf 256K-Byte

als Bausatz 173,— DM

zum Umrüsten 218,— DM

Senden Sie Ihre Mitteilung umgehend an folgende Adresse:

TRONIC-VERLAG GmbH
Hardware-Service
Postfach 870
3440 Eschwege

Bitte senden Sie keine verbindlichen Bestellungen ein. Wir möchten zunächst nur die Resonanz auf unser Angebot testen und werden dann zu einem späteren Zeitpunkt in der Zeitschrift »COMPUTE MIT« ein genaues Angebot und die entsprechenden Bestell-Modalitäten bekanntgeben.

**Aufträge für Ihre persönliche Kleinanzeige
können Sie mit der Auftragskarte
im Innenteil dieser Ausgabe erteilen!**

Bitte beachten:

**Eine Veröffentlichung der Kleinanzeigen
erfolgt nur gegen Vorkasse!!!**

Verkaufe für C16/116: 20 Garantierte Top-Spiele auf Kassette u. a. Dame, Kniffel, U-Boot usw. Zahlung: Vorauskasse DM 20,-. Nachnahme 24,- DM. Bestellung: Telefonisch unter 02773/3411 oder direkt bei Frank Diesner, Industriestr. 48, 6342 Haiger 1; weitere Information unter Tel. 02773/3411 ab 16 Uhr.

VC-20 Programme vorrätig
Angebotsliste gegen 1,30 DM in Briefmarken bei der Basic-Programmbibliothek, Inh. Horst Querfurt, Postfach 84 30 42, 4320 Hattingen 14.

C16/116 preiswerte Software
17+4, Glückspiel, Schiffeversenken, Knobeln, Bakkarat, u. a., Preis a 3,-DM bei Jürgen Schramme, Postfach 16 31, 3260 Fintel 1.

Achtung C16/116, Plus/4-User!!!
Verkaufe Disc vollgepackt mit 24 Spielen und 9 Anwendungsprogrammen für 10,- DM (Poker, Mastermind), Helikopter, (Annon, Kniffel, usw.) Schickt einen 10-Mark-Schein an Michael Grahnies, Stefanstr. 1, 5551 Monzelfeld und Ihr erhaltet umgehend die Diskette. Tel. 06531/3949

Plus/4 C16/116
Super Vokabelprogramm, beliebig viele Vokabeln können eingegeben werden. Kas.: 19,- DM; Dis.: 25,- DM Tel. 02202/30283 (Tobias).

Textedition für C16/116 Plus/4
Ein Super-Textverarbeitungsprogramm nur auf Diskette (64K) für nur 68,- DM (incl. MwSt.) + Porto per NN. Kostenlos Info anfordern bei Firma Frank Schmitz, Kurzer Weg 1, 5206 Neunkirchen 2, Tel. 02247/8111.

Assembler für C16/116 Plus/4
Ein Komfort-Assembler mit vielen Extras nur auf Diskette für nur 45,- DM (incl. MwSt.) + Porto per NN. Kostenlos Info anfordern bei Frank Schmitz, Kurzer Weg 1, 5206 Neunkirchen 2, Tel. 02247/8111.
Auch bei Joysoft erhältlich.

Original Basic 7.0 Compiler
für PC 128 von Data Becker.
Original C-Compiler, Data Becker für PC 128, je 50,— DM.
Burk, Tel. 05 61/77 24 08

VC-20 Software zu Kleinstpreisen. Original Steckmodule und Kassetten bis zu 80% reduziert. Liste gegen 80 Pf. bei Michael Behle, Stryckweg 48, 3542 Willingen.

Lern- und Trainingssoftware für Schneider und Commodore. Psychol.-pädagog.-Praxis, Bergstr. 34, 6900 Heidelberg, Tel. 06221-474711/Info anfordern.

****C16/116** Kfm. Softw., Systemsoftware * Userport 24 I/O-Kanäle.** Info gegen Freiumschlag bei Alfred Lämmerzahl, 4100 Duisburg 17, Saarstr. 85, Tel. 02136/13994.

Spiele * C-16/116/Plus 4 * Spiele

Winter Olympiade	24,90
Computer Hits 10 (10 Progr.)	27,90
Yie are Kung Fu	24,90
Bomb Jack	19,90
Project Nova	24,90
Trail Blazer	19,90
Ghosts 'n Goblins	24,90

Preise: Stand 12/86 zzgl. Versand
Comp. Service, PF 1304, 7913 Senden



COMPUTERSOFT JONIGK

Wir haben für Sie über 150 verschiedene Spiele auf Lager und jeden Tag werden es mehr. 17 verschiedene Mastertronic Games z. B. Heikki, Formula 1, Obido, Speed King je DM 9,90 und natürlich die CSJ Anwendungsprogramme Adress, Datei, Vokabel, Text, Utility je DM 29,90.

C 16/116/Plus 4	Disk	Cass	Robo Knight	Cass
Sun Street (The Newsboy)	39,-	29,-	Shark	9,90
Danger Diamonds	39,-	29,-	Auriga	9,90
Jump Jet	(+ 16K/Plus 4)	39,-	Booby	9,90
Mercenary	(Plus 4)	39,-	Killapepe	9,90
Mercenary Compendium	(Plus 4)	39,-	Varmit	9,90
Saboteur	(Plus 4)	39,-	Guzzler	9,90
Grandmaster	29,-	29,-	Runner	9,90
Winter Events	29,-	29,-	Winnie Witch's Superbroom	11,90
Wimbledon	29,-	29,-	Mission Mars	11,90
Intern. Karate	29,-	29,-	Loco Coco	11,90
Yie ar Kung Fu	39,-	39,-	Lunar Docking	14,90
Dark's Dilema	29,-	29,-	Tazz	19,00
European Games	29,-	29,-	Vox	19,00
Cyborg	19,-	19,-		

Mindestbestellwert DM 30,-

Nicht vergessen CSJ NEWS anfordern



3000 Hannover 1
An der Tiefenriede 27
Tel.: Service 05 11/88 63 83
CSJ COMPUTERSOFT JONIGK

Riesenauswahl an Software
sofort CSJ NEWS anfordern
bitte Computertyp angeben
Händleranfragen erwünscht

*** Für C16/116/PLUS 4 ***
* Umbau C16/116 64K garant. *
* 100 DM *
* Umbauset C16/116 nur 50 DM *
* Turbotape Super (64K) *
* nur 19,50 DM *
* Plus 4 in Deutsch nur 348 DM *
* Elektronikdatasette *
* nur 98,50 DM *
* Floppy 1551 die Reste 389 DM *
* LOGO für C-64 159 DM, *
* nun 49,50 DM *
* Elektronik-Technik-Peters *
* Tannenweg 9, *
* 2351 Trappenkamp *

Französisch auf dem C 16/116
Vocteach Plus mit franz. Accents und deutschen Sonderzeichen! Mit Menüsteuerung, Auswertung etc. Ein Dateisystem garantiert optimale Speichernutzung und sinnvolle Abfragetechnik. Preis im Nachnahmeversand: 15 DM (mit Einführung). Info bzw. Bestellung: B. Jakob, Kehi 4, 8832 Weißenburg 1

* Elite plus V1.0 *** Ka/Di *
* fantastische neue Programmier- *
* sprache für strukturiertes Pro- *
* grammieren auf Ihrem Plus *
* 4/C16/116 (64 KB) *
* 29,- DM mit ausf. Handbuch incl. *
* Porot & Verp. Info gegen Rück- *
* porto bei: TEDsoft, *
* Bohlingerstr. 2, 7700 Singen *

Grafikprogramme - Grafikdemos
für C-64 mit Simon's-Basic und C 16/116. Grafiken aller Art. Supergünstige Grafikdemos mit detaillierten Erklärungen und vielen Beispielen. Initialen in Grafikgroßdarstellung. Info gegen Rückporto, Inform. Cass.: 3,80 DM!
B. Jakob, Kehi 4, 8832 Weißenburg 1

C16 + 116 + Plus 4 Technische Mathe, Schulanwender- und Grafik-Programme zu wirklich realen Preisen. Wahlweise Cass./Disk. z.B. Zahntriebfestigkeit/E. Techn./Hydro-Pneumatik/Werkzeugmaschinen/Bruchrechn./Dreieck/Volumen/Super-Grafik-Programme usw. Katalog mit Prg.-Beschreibung 1 DM Briefmarke Software-Versand A. Ristau, Peetzweg 9, 3320 Salzgitter 1

* WIR LIEFERN NEUE SOFT- *
* WARE FÜR: *
* Commodore C-16/116/Plus 4 *
* C-64/C-128/Amiga, Schneider *
* 464/664/6128, Atari 800/800 XL/ *
* 260 ST/520 ST/1040 ST *
* Liste anfordern von: *
* BERLAU-SOFT, Postfach 1415, *
* 2150 Buxtehude *

C-16/116/Plus 4 Software en Masse!
Verbilligte Preise: Über 200 Artikel Lieferbar! Katalog von Softi Soft, Blanckennagelweg 9, 4770 Soest, Tel.: 02921/4359 RÜCKPORTO !

* NEU-ERÖFFNUNG *
* VERSANDHANDEL *

C-64 Software
Spiel- und Anwenderprogramme
Liste bei
Herbert Müller, Dorfstr. 1
8852 Rain-Unterpeiching

C-64 Super-Spiele-Paket
C-64 nur auf Disk: Alien, Powerstar, Slap, Shot, Phönix, Liebesfilm, Int. Tennis usw. Das Ganze kostet Euch 12 DM, bei Nachnahme 15 DM.
Christian Doeres, Friedrichstr. 5d, 5550 Bernkastel-Kues

* C16/116/PLUS 4 GRAFIK - CAD *
* Zeichen- und Grafikprogramme *
* für 16K-Version. Ein CAD- *
* System (Basissys.) für die 64 K- *
* Version. Anwendbar für tech- *
* nische Zeichnungen u.a. Ein um- *
* fangreicher Befehlssatz ermög- *
* licht einfaches Bearbeiten einer *
* Zeichnung (zoomen, duplizieren, *
* drehen, spiegeln, transferieren, *
* löschen) 16 Ebenen *
* Info gegen Porto: *
* Dipl.-Ing. M. Raetzl, *
* Ulvenbergstr. 6, *
* D-6100 Darmstadt 13 *

***** Software für *****
Atari 800 XL, 130 XE * Atari 260 ST, 520 ST * Commodore VC-20, C-64, C 128, C-16/116/Plus 4 * Commodore Amiga * MSX Computer * Schneider CPC-464/664/6128 + kostenlos Liste anfordern: Frank Schmitz, Kurzer Weg 1, 5026 Neunkirchen 2, Tel.: 02247/8111

CPC SOFTWARE ZU TOPPREISEN!!
tel. best. bei: M+K 07022/6815
Sam Fox Strip Poker 29,90 DM
Werner 29,90 DM
Star Trek (Enterprise) 33,90 DM
Dragon's Lair 33,90 DM
Elite 47,90 DM
Jack the Nipper 31,90 DM
Kane 14,90 DM
Welteroberer 5,90 DM

Verkaufe 11 Spiele und 4 Anwenderprogramme für CPC 464
1. Turbo Tape, 2. Masterfile, 3. Soul Robot, 4. Nonterraqueous, 5. Chiller, 6. Ghostbusters, 7. Yie are Kung Fu, 8. Bruce Lee, 9. Match Point, 10. Match Day, 11. Knight Lore, 12. Winter Games 1 u. 2, 14. Pascal, 15. Tasword. Alle Spiele für 50,- DM auf einer Kassette! Bestellen bei:
Robert Dragojevic, Konstanzerstr. 4, 7730 Villingen, Tel.: 07721/56432

* C-16/116/+ 4 (64K) Programme *
* ACE.....(D) 34,90.....(K) 32,90 *
* Saboteur (normal/64K) *
* Vers.).....27,90 *
* Quiwi.....23,90 *
* Mercenary.....27,90 *
* Jump Jet.....22,90 *
* Karate King.....22,90 *
* Legende im Eis *
* (Adventure).....22,90 *
* Katalog m. Beschreibungen *
* -80 DM *
* Comp. Service, PF 1304, *
* 7913 Senden *

C-16 Low-Cost-Software Plus 4
Varmint-Guzzler-Bandits at Zero-Terra
Cognita-Netrun 2000-Trizone-Tazz-Video
Meanies-P.o.D.-Big Mac
Powerball-Video Poker-Robo Knight
Climb It-Winnie Witch-Shark-Hulk-
Monty on the Run-Speed King-Kane-
One Man and his droid-Kung Fu Kid
Preis pro Spiel nur 9,90 DM
Comp. Service, PF 1304, 7913 Senden

****TOP-SPIELE für C16**:** Gery, Moon-Fighter, Poker, Jump-Man, Pferderennen, Turmspringen, Black-Jack, Cowboy-Duell, Pac-Mac, Freikampf, Shaolin, Affenwurf, Autorenrennen, Malprogramm, Mover, Fressman, Poker II, Tankwars, Oil or Texas, Mimi, Diam, Mistermind, Vitman, SOS, Schiffbruch, Duell, Frog-Match, Soundmaschine. Sie können sich 8 Programme aussuchen für 10,- DM bei Nachnahme 13,60 DM.
Jürgen Arns, Jugendheimstr. 1, 5550 Bernkastel-Kues.

* C-16/116/Plus 4 Anwenderpro- *
* gramme *
* Turbo Tape 15,90 *
* Music Master 22,90 *
* Paint Box 19,90 *
* Graphic Designer 15,90 *
* Utilities C-16 27,90 *
* Text C-16 27,90 *
* Micro Datei/Micro Kalk je 22,90 *
* Tel. Bestellungen: 07307/6230 *
* Comp. Service, PF 1304, *
* 7913 Senden *

Rennersoft
Computer-Distribution
Action · Games
Adventures · Business · Software

Sie suchen preiswerte Software?
für Ihren
Commodore C-16/116 plus 4
Commodore C-64/128
Atari ST

Dann fordern Sie unseren
Gratis-Katalog an!
Bitte unbedingt Computersystem angeben!

Rennersoft
Postfach 920 · 4440 Rheine



Versandhandel R. Lindenschmidt

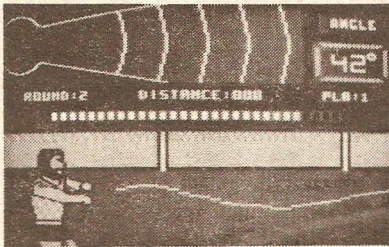
are proudly to present

European Games

5 neue Sportarten für den C-16

- * Hammerwerfen (siehe Foto)
- * Schwimmen
- * Rudern
- * Gewichtheben
- * Weitsprung

DM 19,95



Diese Spiele sind einfach Spitze

Torpedo Run	Cass. 18,95
Speedking	Cass. 9,95
Superhits	Cass. 27,95
Favourite Four	Cass. 26,95
Project Nova	Cass. 24,95
Winter Olympiade	Cass. 29,95
Kikstart	Cass. 9,95

Neuheiten:

Mount Vesuvius	Cass. 26,95
Suicide Run	Cass. 9,95
Winnie the Witch	Cass. 9,95
Trailblazer	Cass. 25,95
Classics III	Cass. 33,95
Kane	Cass. 9,95
Netrun	Cass. 9,95
Terra Cognita	Cass. 9,95
Ghost'n Goblins	Cass. 29,95

64 k Steckmodul

Die Supererweiterung für den C-16

- * 60671 Bytes free
- * Damit ist der C-16 dem 64'er speicherplatzmäßig klar überlegen (22000 Bytes mehr als der C-64)
- * Voll vom Basic her nutzbar
- * Wahnsinnsspiele für die ausgebaut Version

COMMODORE BASIC V3.5 60671 BYTES FREE
READY.

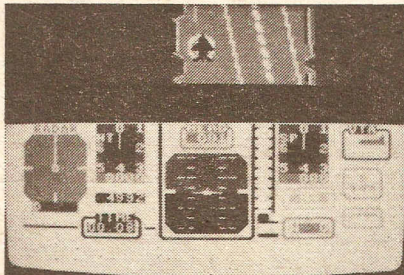
DM 89,-

Jump Jet

(mindestens 16k Erweiterung nötig)

Senkrechtstarter (Flugsimulator)

- * Start vom Flugzeugträger aus
- * Deutsche Anleitung
- * Top Grafik mit verschiedenen Perspektiven
- * Sehr gute Instrumentierung (siehe Foto)
- * Sehr realistische Simulation **DM 33,75**



Superspiele für den ausgebauten C-16

- Operation Thunderstorm**
Ein deutschsprachiges Textadventure, daß es in sich hat. Witzig und locker aufgebaut und nur am Anfang einfach (oder nicht?) Cass. 29,95 Disk. 29,95
- ACE (64k)** Flugsimulator, der die 64k voll ausnutzt!
Einfach super! Cass. 38,95
- Quiwi** Das Quizspiel für die ganze Familie. Cass. 29,95
- Jump Jet** Flugsimulator eines Senkrechtstarters Cass. 33,75
- Kurvendiskussion**
Eine tolle Hilfe für Schule/Beruf. Minima, Maxima, Nullstellen, Polstellen, Wendepunkte, Lupenfunktion, Grafik ständig abrufbar, etc. Cass. 19,95 Disk. 19,95

Literatur und Zubehör

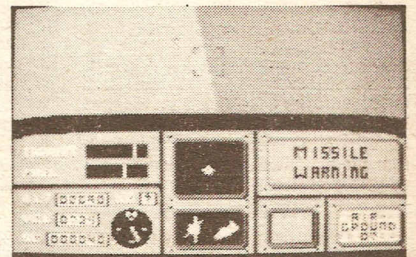
Data Becker: Basic Buch	29,00
Data Becker: Grafikbuch	29,00
Data Becker: Maschinenspr.	29,00
Leercassetten C-15 5 St.	11,95
Leercassetten C-5 5 St.	9,95
C-16 Userport	59,95
Leerdisketten 10 St.	19,95

ACE (64k)

DM 38,95

Ein Kampfflugsimulator der Superlative

- * verschiedene Bewaffnung wählbar
- * Landkartenanzeige
- * Boden, Luft und Schiffskampf
- * Betankung in der Luft möglich
- * Spitzengrafik, wahnsinnig schnell
- * Gegnerischer Raketenangriff wird angezeigt



Unsere Lieferbedingungen

Einfach Spitze: Wir liefern Ihnen die Ware frei Haus - auch nach Österreich, Luxemburg, Belgien, Schweiz, Niederlande (nur der Zoll geht zu Lasten des Empfängers)! Sie bezahlen keine zusätzlichen Nebenkosten, Porto oder Nachnahmegebühren. Einen Mindestbestellwert kennen wir ebenfalls nicht. Durch unseren Telefonservice (keinen Anrufbeantworter - sondern einen Ansprechpartner) können Sie problemlos bei uns bestellen. Unser umfangreiches Lager und das riesige Softwareangebot sorgen für zufriedene Kunden.

Spitzenservice + Superqualität - hierfür bürgt der Name Lindenschmidt!

Versandhandel R. Lindenschmidt · Schulstraße 14 · Postfach 1328 · 4972 Löhne 2 · Telefon (05732) 72849