

"Compute mit"

Sonderheft 2/87
für C-16/116/plus 4

jetzt überall im Handel!

monatlich

Compute mit

COMMODORE & SCHNEIDER

4/87

VC-64, VC-20, C-16/116, plus 4, CPC-464/664/6128

3,80 DM

33 öS

3,80 sFr

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

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

VC 20:

Moth
Jeep
Mini-Monitor

C16/116:

Elfmeterschießen
Super-Q-Bert
Memory-Dump
u. a.

C64:

Ram-Disc-Monitor
Startrip
u. a.

CPC:

Die schwarzen
Diamanten
3-D-PacMan
u. a.

AIR COMBAT EMULATOR



Jetzt geht die Post ab..

Super Angebote



Pro.Kartei Version II

Die einfachste Methode, Daten zu verwalten. Einfach zu Handhaben. Verarbeitet enorme Datenmengen. Und so preiswert. Nur von S+S Soft.

Ein klasse Programm zu einem Spitzenpreis! nur 19,80 DM

Die Trickliste

Jetzt über 100 16 Programme! Eine ganze Diskette voll mit Tips und Tricks, unbekannte Tricks, Tabellen, Anwendungen, Pests und Fakes, Spitzkürs, und, und, und.

nur 19,80 DM

Pro.Plan 64 Version II

Das komplette Büro für den C64. (Nur auf Diskette erhältlich). Jetzt stark verbessert. Warten Sie auf Graphikmenüs zwischen Textverarbeitung, Datenverwaltung, Terminplaner, Adressverwaltung, und, und, und.

Leistung für nur 39,-DM

ANGEBOT

Über 300 Programme schon ab 1,- DM..2,- DM..3,- DM

Unglaublich aber...
Skat
Das Prof-Skript von S+S Soft. Ein starker Skatcomputer mit hervorragender Grafik. Sie können gegen 2 Computer-Spieler mit Unterstützung der Punkte- und Zeit-Sensoren.
nur 19,80 DM

Neu Basic-Kurs Neu
Sie brauchen sich nun nicht mehr vor der Programmiersprache Basic zu fürchten, denn Sie können nun problemlos die Grundlagen des Basic erlernen und verstehen. Das ideale Programm für jeden, der Basic erlernen möchte. Auch für Kinder geeignet. Ein klasse Programm.
nur 5,- DM

Diskmagic
Haben Sie Probleme mit Ihren Disketten? Ab sofort lösen Sie Diskmagic und haben alle wichtigen Programme. Sie finden den Umgang mit der Diskette ganz einfach. Wenn Sie Diskmagic haben, können Sie Disketten ganz leicht formatieren, kopieren, verschieben, löschen, und, und, und. Diese Disketten sind immer in der Hand.
nur 19,80 DM

Endlich ist sie da.

A.L.S.-TEXT-DRESS

Eine Textverarbeitung und Adressverwaltung in einem. Die Erstellung von Rundschreiben oder ähnlichen ist nun kein Problem mehr. Sie brauchen den Brief nur noch einmal zu schreiben und Ihr Computer druckt ihn für jede Person abgestimmt aus. Mit Sortier- u. Suchroutinen, komfortable Menüsteuerung und Funktionsassistentenführung. Einfach Super.
Mit Etikettendrucker 14,80-111 DM

AB SOFORT: GRATIS-FÜR SIE

Flammneuer Katalog + Gratiediskette!

Anforderungscoupon

☐ Schicken Sie bitte kostenlos und unverbindlich Ihre Gratiediskette + Katalog

☐ Gleichzeitig bestelle ich:

ich arbeite mit:
☐ Diskette
☐ Diskette
☐ Lieferung soll per Nachnahme erfolgen (1,- 4,50)
☐ Bei Porto in Verrechnung
☐ Bitte fügen Sie Briefmarken ein

Anschrift: Name _____
Straße _____
Ort _____

Heute noch Gratis anfordern!

S + S SOFT Vertriebs GmbH INDUSTRIESTR. 10 4620 CASTROP-RAUXEL 9

ACHTUNG!! Commodore 64!! C 128!!

Leser- + Meckeredke - Report - Kurse
Softwareservice - Tips & Tricks
Großer Kleinanzeigenmarkt

EDITORIAL

Lieber Leser!

Das Vorwort zu diesem Heft möchte ich in besonders eindringlicher Weise dazu benutzen, diejenigen unter unseren Lesern, die selbst Programme entwickeln zur Mitarbeit anzuregen. Es scheint der Irrtum sehr verbreitet zu sein, daß wir nur Spiele veröffentlichen wollen, so daß wir für den C-16 z. B. 5 Einsendungen gleichen Inhalts verzeichnen können, aber sich all jene, die Anwenderprogramme oder auch nur Tips und Tricks beisteuern könnten, in Schweigen hüllen. Auch Textbeiträge, die Programmierhilfen oder ähnliches geben könnten, sind außerordentlich willkommen. Der Aufruf zur Mitarbeit gilt insbesondere für die Funk-

amateure unter unseren Lesern. Interessierte Leser fragen ständig nach Utilities für Funkamateure. Angeblich sollen diese ja so schreibfaul sein, daß sie ja funken können. Programme lassen sich jedoch noch nicht per Funk zum Tronic-Verlag übermitteln. Deshalb würde ich mich freuen, wenn ich durch diesen Aufruf neue Mitstreiter an der Homecomputerfront gewinnen könnte.

Mit freundlichen Grüßen

Uwe Knierrim

Uwe Knierrim
Chefredakteur

Report

- News 3
- Pawn – Ein Traum im Supermarkt? .. 4
- Buchvorstellung, Software 5
- A.C.E – Flugsimulator 7

Rubriken

- Klartext 6
- Updates (Korrekturen) 29

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 Knierrim
Redakteure: Frank Brall, Ottfried Schmidt, Thomas Brandt, Martina Strack

Gesamtherstellung:
Druckhaus Dienrichs 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, Mikrofilm, 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 service 32
- Checksummer (C16/116/plus 4) 39
- Interruptprogrammierung (C 64) 40
- Kleinanzeigen 59

Software

Commodore

- Moth (VC 20) 8
- Jeeper (VC 20) 10
- Monitor(VC 20)..... 12
- Elfmeterschießen (C 16/116/plus 4) 12
- Super-Q-Bert (C 16/116/plus 4) 19
- Froggy (C 16/116/plus 4) 22
- Memory Dump (C16/116/plus 4) 24
- Ram-Disc-Monitor (C 64), Teil 1 25
- Startrip (C 64) 30
- Immigration (C 64)..... 44
- Run Version Generator (C 64) 46

Schneider

- Find (464) 48
- Tape Backup System (464) 48
- Die schwarzen Diamanten
(464, 664, 6128) 51
- 3-D-Pac Man (464, 664, 6128) 55

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) 300 13

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.

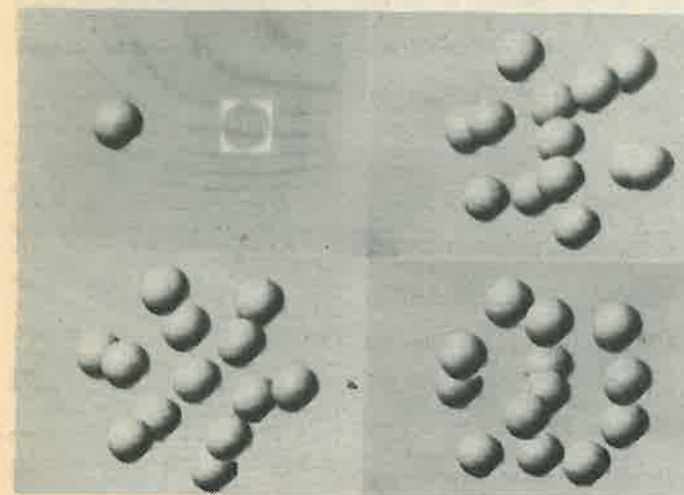
Rücksendung erfolgt nur gegen Erstattung der Kosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:
Kopierfähige Kasette oder Diskette mit dem Programm (Computerbezeichnung nicht vergessen!), von Drucker erstelltes Listing (keine Schreibmaschinenlistings!), evtl. Bildschirmfotos 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

Molekülmodelle mit AMIGA



Neue Wege, die Wirkung von chemischen Substanzen zu analysieren und verstehen, geht das Institut für Physikalische Chemie an der Technischen Hochschule in Darmstadt. Eine Projektgruppe nutzt die Rechenleistung und die Grafikfähigkeit des AMIGA, um physikalisch-chemische Experimente im Computer nachzuvollziehen. Ziel ist es, das Verhalten von Wirkstoffen auf molekularer Ebene zu verstehen und dadurch die Funktionalität zu effektivieren.

Im Rahmen eines vom Bundesforschungsministeriums finanzierten Projektes, führen die Wissenschaftler auf Großrechnern Experimente mit chemischen Stoffen aus, die nur in Form von mathematischen Modellen existieren. Die dabei entstandenen Zahlenberge verarbeitet der AMIGA anschließend zu bewegten Farbgrafiken, die die Molekülmodelle und deren Reaktionsweisen optisch darstellen. Durch diese

Vorgehensweise kann man Molekularstrukturen verschiedener Substanzen darstellen und durch dreidimensional Darstellung eine räumliche Vorstellung komplexer Vorgänge gewinnen.
"Chemiker haben schon immer mit Modellen gearbeitet, um wissenschaftliche Sachverhalte zu visualisieren" erklärt Michael Teschner vom Institut. Die modernen Grafikcomputer lösen somit die einfachen Drahtgebilde, wie auch die Vektorgrafiken der Großrechenanlagen ab.
Weitere Möglichkeiten bieten sich, indem man bestimmte Eigenschaften der Moleküle durch Farbcodierungen hervorheben kann. Ferner ist es möglich, in die Reaktionsabläufe der Moleküle einzugreifen, und so interaktive Forschung zu betreiben, die zum Beispiel in der pharmazeutischen Industrie zu einer Reduzierung der Tierversuche führen könnte.

Quelle: Commodore



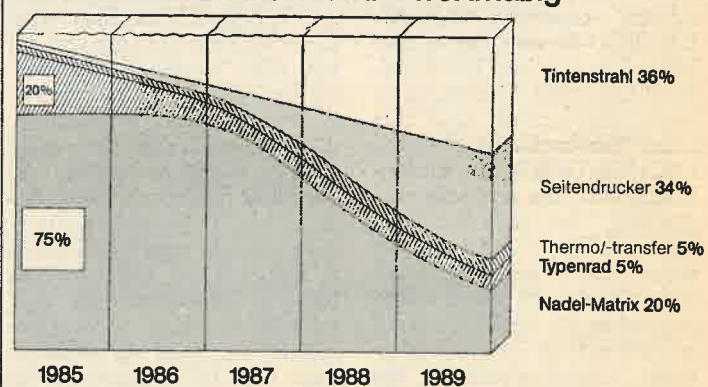
Neue Wege der Computerfinanzierung

Möglichen Finanzierungsproblemen beim Kauf eines Schneider-Computers soll der Fachhandel nun durch den sogenannten Computer-Kredit entgegenwirken. In Zusammenarbeit der SKG Bank und der Schneider-Computer-Division wurde diese direkte Finanzierungsmöglichkeit entwickelt. Der Vertragsabschluß kann gleich beim Händler, durch den Abschluss eines vorgefertigten Kreditvertrages, abgewickelt

werden. Die einzelnen Monatsraten liegen für die CPC und die Joyce-Geräte zwischen 47 und 97 DM. Der Computer-Kredit betrifft auch das jüngste PC-Modell von Schneider. Weiter Informationen über den Schneider-Computer-Kredit werden vom Fachhandel und von der SKG Bank GmbH, Cecilienstraße 4, 6600 Saarbrücken, Telefon 06 81/30 30 1, erteilt.

Druckertrends

Marktentwicklung nach Drucktechnologien in Deutschland – wertmäßig



Grafik EPSON

Bei Computerdruckern geht, gemessen am 100% - "Kuchen", der Anteil der Typenrad- und der Nadeldrucker, wertmäßig zurück. Dies ergab eine Untersuchung im Auftrag der Firma Epson. Stark gewinnen werden laut dieser Untersuchung die Tintenstrahldrucker. In absoluten Zahlen gesehen, wird die Anzahl der Matrixdrucker weiter steigen. Günstige Preise und die neuen Schön-schreibdrucker mit 24 Nadeln tragen dazu bei.

4.500,-DM noch zu hoch gegriffen. Der Preisverfall bei den Laserdruckern, die wiederum Seitendrucker erhalten (Liquid Cristal-, Ionen- und Hochlichtdrucker) wird sich auch auf die Tintenstrahldrucker auswirken. In absoluten Zahlen gesehen, wird die Anzahl der Matrixdrucker weiter steigen. Günstige Preise und die neuen Schön-schreibdrucker mit 24 Nadeln tragen dazu bei.

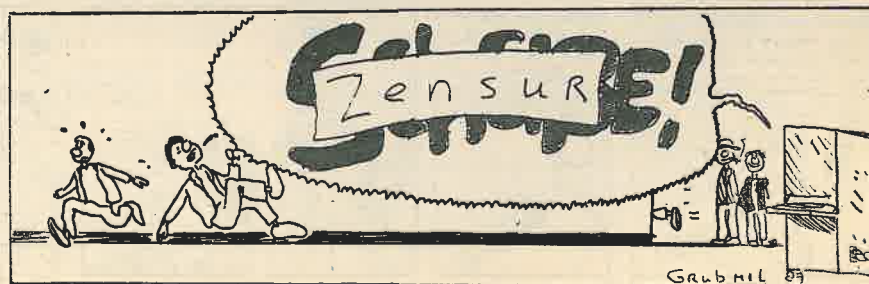
Quelle: Epson

Computern mit LEGO!?



Selber mit den Anfängen des LEGO-Systems als "Kind" in Berührung gekommen, folgendes: Der LEGO-Baustein war nur zum Häuserbauen gedacht und konzipiert! Um etwas Technisches zu bauen, blieb nur der Griff zum Metall-Stabilbaukasten. Die FISCHER-Technik hat dies erkannt, was dann auch zur Verdrängung der Stabil-Baukästen führte. LEGO, in Zugzwang, hat seinen Bausteinen dann mehr und mehr "technische Funktionen" zugeordnet, für die das System eigentlich nicht geeignet ist. Die optische Gegenüberstellung von LEGO-Technik und FISCHER-Technik macht dies nur zu deutlich! Auch preislich ist LEGO im Nachteil. (LEGO Motor ca. 50 DM; FISCHER Motor ca. 40 DM!) Daß einem Schüler der Umgang mit den LEGO-Steinen vertraut ist, stimmt zwar. Auch daß an Schulen, besonders berufsbezogenen Schulen, nur noch mit Steckmodulen (z.B. in der Elektrotechnik, Hydraulik und Pneumatik) gearbeitet wird, macht das System sich zu Nutze. Nur, bei den Jahrgängen der Schüler, die in der Schule (wenn überhaupt) mit dem Computer konfrontiert wer-

den, wird es wohl so sein, daß sie von Haus aus durch eigene Rechner ein gerütteltes Maß von Grundwissen mitbringen und daher dies "Spielzeug" ablehnen dürften. Vermittlung von EDV-Grundrissen z.B. für kaufmännische Berufe, läßt sich wohl schlecht damit vermitteln. Das Berufsbild in den technischen Berufen wird in nächster Zukunft erweitert; d.h. berufliche Schulen müssen sich mit CNC-Technik befassen. Dafür kommt nur eine herstellerabhängige Anwendersprache in Frage (größtenteils nach DIN 66025 genormt), die weit über Basic hinausgeht. Fazit: Wenn überhaupt, ist dieses System nur für Grundschulen geeignet; und auch da nur für Jahrgänge, die noch spielerisch lernen können. Wie schon gesagt, der Baustein eignet sich nicht besonders, technische Modelle und Abläufe wirklichkeitsnah darzustellen. Die Fa. LEGO versucht offensichtlich, verlorene Marktanteile zurückzuholen. Da sie von vornherein das System für Schulen anbietet, dürfte es LEGO wohl bekannt sein, daß sie damit auf dem privaten Markt nicht landen kann. U. Klein



The Pawn - Ein Traum im Supermarkt ?

Also, das war wirklich eine seltsame Geschichte. Keiner wird sie mir jemals glauben. Dennoch werde ich sie erzählen, so oft es mir möglich sein wird. Denn vielleicht treffe ich sie eines Tages doch noch, den sagenumwobenen König Erik, den Zauberer, und die vielen anderen Bewohner von Kerovnia. Es war einmal in einer Stadt wie jeder anderen an einem Morgen, wie jeder andere auch.

Ich ging zum Supermarkt und ahnte nicht, was mir bevorstand. Ein seltsam aussehender Typ schlug mich nieder, ich war bewußtlos und wachte in einer seltsamen Welt jenseits unseres Vorstellungsvermögens wieder auf ... in Kerovnia. Eine Welt mit düsteren Wäldern, Lichtungen, Felsvorsprüngen, Grasslandschaften, steinernen Brücken und wunderbaren Palastgärten. Seltsame Schilder begegneten mir hin und wieder und teilten mir mit, daß ich den Adventure - Bereich verlassen würde. Ich hatte Jeans und ein T-Shirt mit dem Bild einer seltsamen Pflanze an. In jenem Land gab es Radios, Telefon und Milchshakes, ebenso wie Drachen und mystische Magiere. Ich begegnete einem mysteriösen Baumstumpf und einem Guru, der manchmal austritt. Ein gewisser Honest John wollte mir allen möglichen Plunder andrehen, obwohl ich kein Geld hatte. Mit seinem Messer ging John auch nicht gerade zimperlich um. Schließlich fand ich zwei Schlüssel, einen hölzernen und einen aus Metall. Mit dem hölzernen konnte ich ein stabiles Baumhaus öffnen, indem sich allerdings nichts interessantes befand, wie mir von jener seltsamen Adventure-Stimme gesagt wurde. Der Schlüssel aus Metall öffnete einen Geräteschuppen im Palastgarten. Dort setzte ich mich erschöpft auf eine Bank, unter der

ich einen Blumentopf und eine seltsame Pflanze fand, die der auf meinem T-Shirt verblüffend glich. Gelegentlich begegnete ich auch einem Abenteurer. Das sind Menschen einer ganz besonderen Art, die sich dadurch hervortun, daß sie solch unverständliches Zeug wie "nimm Schwert" oder "zünd' Laterne an" murmeln. Doch auch ein Pferd ohne Beine, das offensichtlich fliegen kann, zeichnet einen echten Abenteurer besonders aus. Kronos, das ist der Name eines Magiers, der mir beharrlich eine Nachricht für Erik, den König, mitgeben wollte. Doch Erik war über die Nachricht gar nicht erfreut und schmiß mich im hohen Bogen wieder raus. Die Nachricht schien auch recht tödlich gewesen zu sein, jedenfalls werde ich keinen versiegelten Brief mehr öffnen. Der Knig Erik hatte aber auch wirklich genug Probleme. Zwerge hatten seine Frau getötet, Erik verwies die Zwerge daraufhin des Landes. Das wiederum gefiel dem Volk von Kerovnia gar nicht, den die Zwerge produzierten den besten Whiskey weit und breit. Das alles schien mir doch ziemlich verworren zu sein, vor allem, wenn ich mir die Landkarte von Kerovnia näher anschauete - über den schneebedeckten Bergen im Hintergrund ragt nämlich ein dämonischer Kopf mit leuchtenden Riesenaugen... Ich glaube mich erinnern zu können, daß ich kurz vor der Lösung des Rätsels stand, als ich plötzlich von der Verkäuferin des Supermarktes geweckt wurde. Ich hatte mich nur hingeworfen, während meine Frau an der Wurst-Theke anstand. Ich muß wohl eingenickt sein und alles nur geträumt haben. Seltsam, woher stammte aber dann nur die Beule an meinem Hinterkopf?

Patrick Schmitz

Die Buchvorstellung:

Schneider CPC/Commodore 128 Arbeiten mit Turbo Pascal



Autor: Karl-Hermann-Rollke
Preis: 48,- / 42,- DM
Verlag: Sybex-Verlag GmbH,
Postfach 30 09 61,
4000 Düsseldorf 30

Neben BASIC dürfte Pascal, insbesondere Turbo Pascal, wohl die beliebteste Programmiersprache auf dem Markt sein. Zahlreiche Verlage haben dies schon seit langem erkannt und entsprechende Bücher zu diesem Thema veröffentlicht. Auch der Sybex-Verlag ist mit mehreren Büchern gezielt auf diesen neuen Pascal-Dialekt eingegangen. Mittlerweile gibt es ein spezielles "Turbo-Buch" für die Systeme Schneider CPC, Commodore 128 sowie ein allgemeines Buch für alle Systeme. Neben dem Autor ist auch der größte Teil des Inhaltes bei allen Büchern identisch, lediglich einige Seiten wurden verändert oder kamen hinzu. Aus diesem Grund widmet sich diese Buchvorstellung allen drei Büchern. Der Autor versucht offensichtlich in diesem Buch den Inhalt, der für den Anfänger teilweise etwas schwer verständlichen Handbuchs, in vereinfachter Form wiederzugeben. Im Vorspann wird in beiden Büchern auf die Zusammenarbeit zwischen Turbo-Pascal und CP/M-Betriebssystem eingegangen. Es wird auf einfachste Weise dargestellt, wie man sich eine Arbeitsdiskette erstellt und das Turbo-System aufruft.

Anschließend erfolgt eine recht umfangreiche und vor allem übersichtliche Darstellung aller Editoreigenschaften und Funktionen. Nach einer kurzen Beschreibung des Compilers wird man schrittweise in den Dialekt Turbo eingeführt. Die Einführung wurde so geschrieben, daß selbst Pascal-Neulinge keine Schwierigkeiten haben dürften, hier ihren Einstieg zu finden. Zahlreiche Beispielprogramme, wie eine Adressverwaltung oder ein komfortables File-Kopierprogramm, erleichtern dem Anfänger die schnelle Gewöhnung an die Syntax und Struktur eines Pascal-Programmes. Neben der Beschreibung aller Prozeduren und Funktionen wird auch auf die erhältliche Turbo-Toolbox eingegangen. Beim CPC werden zusätzlich auch die Funktionen des Grafikpaketes erläutert. Besonders erwähnen möchte ich, daß auch die Turtle-Befehle erläutert werden, die beispielsweise von Heimsoeth übernommen wurden. Besonders gut ist auch der Anhang des Buches gelungen, welcher mittels zahlreichen Tabellen schnell einen Überblick über vorhandene Funktionen, Prozeduren und die Compiler-Fehlermeldungen gestattet. Alles in allem kann man sagen, daß dieses Buch zwar einen leichten Einstieg in die Programmiersprache Turbo Pascal erlaubt, jedoch das Original-Handbuch nicht ersetzt. Frank Brall

Software:

Die Trickkiste auf Diskette

Programm: Die Trickkiste
System: C64
Preis: auf Anfrage
Bezugsquelle: Frank Schmitz,
Kurzer Weg 1, 5206 Neunkirchen,
Tel. 02247/8111

Die Trickkiste ist keine neue Rubrik in unserer "Compute Mit", sondern ein neues Programm vom Elektronikhandel Frank Schmitz. Wie der Name schon sagt, handelt es sich hier um eine Sammlung von verschiedenen Programmierhilfen, so zum Beispiel interessante POKE und PEEK-Befehle. Das Wort interessant müßte ich eigentlich in Anführungszeichen setzen, denn mittlerweile dürfte nicht zuletzt auch durch die "Compute Mit" sowie zahlreiche Konkurrenzprodukte das Innenleben des C64 ausreichend beleuchtet worden sein. Etwa 136 KByte an Informationen lassen sich über zahlreiche Menüs von der Diskette abrufen. Die Trickkiste ist in 6 Rubriken unterteilt: Tips & Tricks, Der C-64, Peeks u. Pokes I, Peeks u. Pokes II, Toolcreator Anl., Toolcreator. Diese Rubriken lassen sich wahlweise über ein Hauptmenü oder auch direkt aktivieren. Hier ein kurzer Überblick, der die Leistungen und Informationen der einzelnen Rubriken darstellt:

Tips u. Tricks

Bei dieser Rubrik handelt es sich um die üblichen Tips/Tricks wie sie meist unter der gleichnamigen Rubrik veröffentlicht werden oder wurden. Hier lassen sich ebenfalls über Menü interessante POKE/PEEK oder SYS-Befehle für Bildschirm, Tastatur, Basicpointer und ähnliches abrufen. Insgesamt 4 Menüs geben Auskunft über wichtige Disk-, Sprite-, I/O-, Interrupt- und Joystick-adressen. Selbstverständlich werden auch eine Reihe von Listenschutzverfahren besprochen.

Der C64

Diese Rubrik informiert über grundlegende Eigenschaften des C64 sowie des Basicinterpreters.

Unter anderem wird hier der Aufbau bzw. die Funktion des Speichers, der Variablen, der Basic-Befehle u.v.m. besprochen. Da es sich hier ausschließlich um Textinformationen handelt, scheint mir ein Buch wesentlich informativer und übersichtlicher.

Peeks und Pokes I und II

Der Inhalt dieser Rubrik braucht eigentlich nicht näher erläutert zu werden, da der Begriff PEEK und POKE allgemein bekannt ist. Hier handelt es sich wie schon anfangs erwähnt um bekannte POKE's für Bildschirm, Basicpointer u.s.w. Entgegen den vorherigen Informationen kann die POKE-Liste I nur auf dem Drucker ausgegeben werden, die POKE-Liste II wird dagegen auf dem Bildschirm ausgegeben.

Toolcreator und Anleitung

Diese beiden Rubriken bzw. dieses Programm ist sicherlich das interessanteste der Toolbox. Der Toolcreator ist ein Programm, welches 100% in Maschinencode geschrieben wurde, und in der Lage ist kleine Utilities auf Diskette zu erzeugen. Unter den Utilities sind Funktionen wie OLD, AUTOSTART, DOS-HELP und ähnliches. Etwas unklar ist, warum der Toolcreator diese Files erst erstellen muß, wäre da eine kompakte Basic-Erweiterung nicht sinnvoller?

Alles in allem kann man wohl sagen, daß die Trickkiste vielleicht für den Einsteiger eine nützliche Hilfe ist, für den Insider dürfte das Programm jedoch kaum neue Informationen oder Hilfen bringen.

Frank Brall

Positiv:
Einfache Bedienung durch Menüsteuerung
Interessante Toolbox

Negativ:
Kaum Informationen für Insider
Teilweise etwas lange Ladezeiten
(kein Diskettenbeschleuniger)

Neu für alle Commodore Rechner C 64/VC 20 und C 16

Auf vielfachen Leserwunsch haben wir uns entschlossen, unser Druckverfahren für Commodore-Listings zu ändern. Steuerzeichen sowie alle Grafikzeichen werden zukünftig durch Klartext ersetzt. Der Ausdruck ist kompatibel zum bisherigen Druckverfahren, ein neuer Checksummer ist aus diesem Grund nicht erforderlich.

Eingabehinweise für alle Commodore Rechner!

Wer sich schon etwas näher mit den Commodore Rechnern befaßt hat, wird sicherlich wissen, daß der Grafiksatz des C64 sehr viele ähnliche Symbole enthält. Aus diesem Grund ist es oft sehr schwer, einzelne Zeichen, wie zum Beispiel horizontale Linien, voneinander zu unterscheiden. Besonders macht sich dieser Nachteil in Listings bemerkbar, welche die Bildschirmgrafik mit PRINT-Zeilen aufbauen.

Bisher haben wir in unserem Ausdruck lediglich die Steuersequenzen wie CTRL-RVS ON oder CTRL-BLUE durch Klartext ersetzt, ab dieser Ausgabe werden jedoch auch alle Grafikzeichen durch ein Schlüsselcode ersetzt. Durch diese Methode sind Verwechselungen praktisch völlig ausgeschlossen.

Wie arbeitet unser Klartext?

Wer schon nach dem alten Verfahren Programme eingegeben hat, wird sicherlich keine Schwierigkeiten mit der Umstellung haben.

Für die neu hinzugekommenen Leser erkläre ich jedoch das Verfahren noch einmal grundlegend.

Alle sogenannten Steuerzeichen (z.B. eine Farbe) sowie ein Grafikzeichen, werden in unseren LISTINGS durch ein CODEWORD, welches die Taste bzw. Tastenkombination kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin (z.B. Computronic o. Compute Mit). Das folgende Beispiel zeigt den Unterschied zwischen unserem alten und dem neuen Druckverfahren:

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```
10 REM BISHERIGER AUSDRUCK      <118>
20 PRINT "DOWN4"                <52>
30 PRINT "SA SB SC SD SE SF SG SH SJ SK" <85>
40 PRINT "CA CB CC CD CE CF CG" <149>
50 PRINT "ENDE"                  <246>
ENDE DES LISTINGS
[DRUCK:F.B.]
```

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

```
10 REM NEUER AUSDRUCK          <162>
20 PRINT "DOWN4 SD CY9 SP3"    <52>
30 PRINT "SA SB SC SD SE SF SG SH SJ SK" <85>
40 PRINT "CA CB CC CD CE CF CG" <149>
50 PRINT "ENDE"                  <246>
ENDE DES LISTINGS
[DRUCK:F.B./D.S.]
```

Wie in dem Beispiel zu erkennen ist, werden alle Steuertasten durch die entsprechende Tastenbeschreibung markiert. Das Wort DOWN steht in diesem Fall für Cursor nach unten. Die Zahl dahinter gibt an wie oft die Taste betätigt werden muß. In unserem Beispiel mußte also die Taste CURSOR abwärts viermal betätigt werden. Die genaue Beschreibung, welche Taste gemeint ist, entnehmen Sie bitte der Klartext-Tabelle, welche am Ende dieser Beschreibung abgedruckt ist. Achtung für C16 und C64 existieren zwei verschiedene Tabellen!

Wie ebenfalls aus dem Beispiel zu erkennen ist, werden auch alle Grafikzeichen durch einen 2-Buchstaben-Code ersetzt. Der erste Buchstabe ist entweder ein »S« oder ein »C«. Das »C« steht für die »COMMODORE TASTE« und das »S« für die »SHIFT TASTE«. Der zweite Buchstabe bezeichnet die Aufschrift der Taste. Das »SA« in unserem Beispiel bedeutet also, daß die Taste SHIFT zusammen mit dem Buchstaben »A« betätigt werden muß. »C+« würde bedeuten, daß die Taste COMMODORE zusammen mit dem PLUS Zeichen gedrückt wird. Wird direkt hinter dieser Buchstabenkombination eine Zahl angegeben, so gibt diese die Anzahl der jeweiligen Zeichen an. Das Zeichen COMMODORE Y wurde also in unserem Beispiel 9 mal eingegeben. Durch diese Methode entfällt zukünftig das lästige Abzählen von mehrfachen Zeichen.

Sollten Sie einmal nicht wissen, welche Taste gemeint ist, so hilft Ihnen ein Blick in unsere Klartext-Tabelle weiter.

Um im Listing Tastenbezeichnungen von normalen Zeichen zu unterscheiden, werden alle Tastenkennzeichnungen in geschweifte Klammern gesetzt. Diese dürfen selbstverständlich nicht eingegeben werden. Auch Leerzeichen innerhalb geschweiften Klammern dienen nur zur Trennung einzelner Tastenfunktionen und dürfen ebenfalls nicht eingegeben werden. Um die Tastenfunktionen noch besser hervorzuheben, werden diese in unterstrichener Kursivschrift (Schrägschrift) dargestellt. Alle Zeichen außerhalb der geschweiften Klammern werden normal abgedruckt und auch eingegeben.

Auf den ersten Blick hört sich das sicher etwas kompliziert an, ist jedoch in der Praxis ganz einfach. Wenn man sich erst einmal an die in Klartext geschriebenen Zeichen gewöhnt hat, wird man den großen Vorteil dieser Schreibweise erkennen.

Frank Brall

Noch eine wichtige Anmerkung!

Aus technischen Gründen können drei Zeichen von unserm Ausgabegerät nicht verarbeitet werden. Diese Zeichen sind: Pfeil nach links, engl. Pfund, Pfeil nach oben

- Steht für den Pfeil nach links
- £ Steht für das englische Pfund-Symbol
- ^ Steht für den Pfeil nach oben

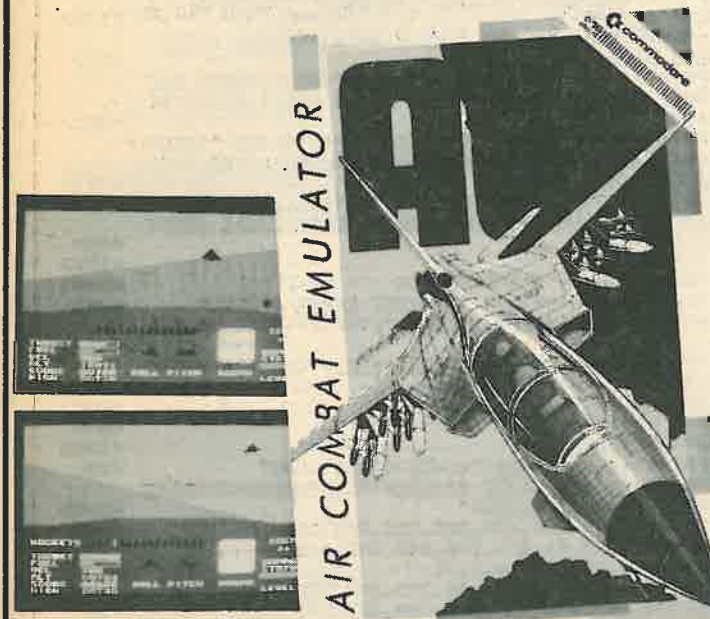
Klartext-Tabelle für Commodore 64 und VC 20

DOWN UP	CURSOR ABWÄRTS CURSOR HINAUF	TASTE NEBEN RECHTEN SHIFT
CLR	CLEAR SCHIRM	HIT-TASTE & TASTE NEBEN RECHTEM SHIFT
INST	EINFÜGEN	SHIFT-TASTE & 2. TASTE GANZ RECHTS OBEN
HOME	CURSOR IN ECKE	SHIFT-TASTE & TASTE GANZ RECHTS OBEN
DEL	DELETE	2. TASTE VON GANZ RECHTS OBEN
RIGHT	CURSOR RECHTS	TASTE GANZ RECHTS OBEN
LEFT	CURSOR LINKS	TASTE GANZ RECHTS UNTEN
SPACE	LEERZEICHEN	SHIFT-TASTE & TASTE UNTEN RECHTS
F1	FUNKTIONSTASTE	LEERTASTE (GRÖSSTE TASTE)
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F1
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F2
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F3
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F4
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F5
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F6
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F7
BLACK	SCHWARZ	FUNKTIONSTASTE F8
WHITE	WEISS	CONTROL-TASTE & 1
RED	ROT	CONTROL-TASTE & 2
CYAN	TUERKIS	CONTROL-TASTE & 3
PURPLE	PURPUR	CONTROL-TASTE & 4
GREEN	GRÜN	CONTROL-TASTE & 5
BLUE	BLAU	CONTROL-TASTE & 6
YELLOW	GELB	CONTROL-TASTE & 7
RVS ON	INVERSE EIN	CONTROL-TASTE & 8
RVS OFF	INVERSE AUS	CONTROL-TASTE & 9
ORANGE	ORANGE	CONTROL-TASTE & 0
BROWN	BRAUN	COMMODORE-TASTE & 1
LIG. RED	HELLROT	COMMODORE-TASTE & 2
DGREY	DUNKELGRAU	COMMODORE-TASTE & 3
MGREY	MITTELGRAU	COMMODORE-TASTE & 4
LIG. GREEN	HELLGRÜN	COMMODORE-TASTE & 5
LIG. BLUE	HELLBLAU	COMMODORE-TASTE & 6
HGREY	HELLGRAU	COMMODORE-TASTE & 7
CTRL ...	CTRL-ZEICHEN	COMMODORE-TASTE & 8
S ...	GRAFIKZEICHEN	CONTROL-TASTE UND DAS ... ZEICHEN
C ...	GRAFIKZEICHEN	SHIFT-TASTE UND DAS ... ZEICHEN
SHIFTSPACE	UNSICHTBARER CODE	COMMODORE-TASTE UND DAS ... ZEICHEN
		SHIFT-TASTE UND SPACE

Klartext-Tabelle für Commodore C16 und Plus 4

DOWN UP	CURSOR ABWÄRTS CURSOR HINAUF	TASTE NEBEN INST DEL: RECHTS OBEN
CLR	CLEAR SCHIRM	3. TASTE NEBEN DOWN VON RECHTS OBEN
INST	EINFÜGEN	TASTE ÜBER RETURN ZUSAMMEN MIT SHIFT
HOME	CURSOR IN ECKE	SHIFT-TASTE & TASTE GANZ RECHTS OBEN
DEL	DELETE	2. TASTE VON GANZ RECHTS OBEN
RIGHT	CURSOR RECHTS	TASTE GANZ RECHTS OBEN
LEFT	CURSOR LINKS	4. TASTE VON RECHTS OBEN
SPACE	LEERZEICHEN	5. TASTE VON RECHTS OBEN
F1	FUNKTIONSTASTE	LEERTASTE (GRÖSSTE TASTE)
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F1 (VORHER DEFINIEREN)
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F2 (VORHER DEFINIEREN)
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F3 (VORHER DEFINIEREN)
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F4 (VORHER DEFINIEREN)
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F5 (VORHER DEFINIEREN)
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F6 (VORHER DEFINIEREN)
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F7 (VORHER DEFINIEREN)
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8 (VORHER DEFINIEREN)
BLACK	SCHWARZ	CONTROL-TASTE & 1
WHITE	WEISS	CONTROL-TASTE & 2
RED	ROT	CONTROL-TASTE & 3
CYAN	TUERKIS	CONTROL-TASTE & 4
PURPLE	PURPUR	CONTROL-TASTE & 5
GREEN	GRÜN	CONTROL-TASTE & 6
BLUE	BLAU	CONTROL-TASTE & 7
YELLOW	GELB	CONTROL-TASTE & 8
RVS ON	INVERSE EIN	CONTROL-TASTE & 9
RVS OFF	INVERSE AUS	CONTROL-TASTE & 0
ORNG	ORANGE	COMMODORE-TASTE & 1
BRN	BRAUN	COMMODORE-TASTE & 2
YLGN	GELBGRÜN	COMMODORE-TASTE & 3
PINK	ROSA	COMMODORE-TASTE & 4
BLGRN	BLAUGRÜN	COMMODORE-TASTE & 5
LBLU	HELLBLAU	COMMODORE-TASTE & 6
DBLU	DUNKELBLAU	COMMODORE-TASTE & 7
LGRN	HELLGRÜN	COMMODORE-TASTE & 8
CTRL ...	CTRL-ZEICHEN	COMMODORE-TASTE & 9
S ...	GRAFIKZEICHEN	CONTROL-TASTE UND DAS ... ZEICHEN
C ...	GRAFIKZEICHEN	SHIFT-TASTE UND DAS ... ZEICHEN
FLASH ON	BLINKEN EIN	COMMODORE-TASTE UND DAS ... ZEICHEN
FLASH OFF	BLINKEN AUS	CONTROL-TASTE UND ...
SHIFTSPACE	UNSICHTBARER CODE	CONTROL-TASTE UND ...
		SHIFT-TASTE UND SPACE

A.C.E. - Flugsimulation der Extraklasse



Programm: A.C.E., System: C-16/Plus4, Preis: 34,95 DM, Hersteller: Cascade, Vertrieb: Ariola-Soft.

Air Combat Emulator (ACE) ist kein Flugsimulator in bisher gekanntem Stil. Es vereint vielmehr das Know how einer guten Flugsimulation mit den erforderlichen schnellen Reaktionen eines "Ballerspiels". Dazu kommt die sehr hohe Geschwindigkeit des Spielgeschehens. Die Grafik, in vergleichbaren Spielen eher ein Schwachpunkt, ist ganz einfach Top. Da gibt es kein Flimmern oder störende Farbverfälschungen während des Spiels. Zudem ist sie dem schnellen Spielgeschehen in keiner Weise unterlegen. Jede Steuerbewegung des Jets wird sauber und präzise umgesetzt, so das der Jetpilot schnell vergißt, daß es sich ja eigentlich nur um ein Spiel handelt. Der Eindruck, den der Spieler bei seinem Flug vermittelt bekommt, läßt fast den Wunsch aufkommen, ein solches Fluggerät in Natura zu fliegen. Zugegeben die Version für die 64K-User ist noch um einiges besser (z. B. kann in der "großen" Version in der Luft nachgetankt werden, oder aber können hier verschiedene Flughäfen angefliegen werden.), schmä-

lert aber nicht den guten bis sehr guten Gesamteindruck der "kleinen" Version. Nun aber zu dem Spielgeschehen. Nach dem Laden (über 8 Min.) werden zunächst die verschiedenen Gegner dargestellt. Dementsprechend gibt es beim evtl. Abschuss eines Feindes auch unterschiedliche Punkte. Jetzt noch die Auswahl des Levels. (Flug bei schönem Wetter, bis 4 = Nachtflug), und auf ins Gefecht. Oder nicht? Zunächst einmal passiert gar nichts. Vor mir sehe ich den Horizont. Geteilt in hellem Blau (der Himmel) und Grün (die Landschaft). Ein Blick auf die Kontrollen gibt mir Auskunft über meine momentane Fluggeschwindigkeit, Flughöhe und vorhandene Bewaffnung (normale und Infrarot-Raketen). Daneben sehe ich an 2 weiteren Anzeigen meine Fluglage (Roll- und Pitchposition). Kurze Pieper lenken meinen Blick auf den Radarschirm, der ständig die aktuellen Positionen meiner Gegner wieder gibt. Aha, zwei sind über und einer ist unter meiner Flughöhe. Also den Steuerknüppel noch vorn, den Feuerknopf leicht berührt, und schon geht es abwärts. Jetzt richte ich meine ganze Aufmerksamkeit auf das Radar, da ich beim Blick aus dem Cockpit nur noch die Landschaft

sehe. Ein Auge schielt dabei auf die Höhenanzeige. Da. Gleiche Flughöhe. Ein leichtes Ziehen am Joy, verbunden mit einer Rechtskurve. Das Radar zeigt mir den Feind nun direkt vor mir. Ich schaue aus dem Cockpit und suche meinen Gegner in der angezeigten Richtung auszumachen. Fehlanzeige. Am Radar sehe ich, daß der Lummel seine Position verändert hat und jetzt genau hinter meiner Maschine herfliegt. Rummms. Schon fliegen mir die ersten Geschosse um die Ohren. Einige explodieren so nah, daß mein Jet sich richtig schüttelt (der Bildschirm bebzt tatsächlich) Tack - Tack - Tack. So rauscht es schon wieder heran. Aber jetzt habe ich meine Überraschung überwunden. Doch der nächste Schreck. Ich habe nur noch eine Höhe von 100 Fuß und so reiße ich instinktiv den Steuerknüppel nach hinten und jage jetzt mit über 700 Meilen in den Himmel. Doch fast zu spät, denn das ist wohl der geeignete Zeitpunkt für meinen Gegner. Ich ziehe unwillkürlich den Kopf ein als seine Maschine von oben durch mein Blickfeld in einer Linkskurve nach unten zischt. Aber dadurch habe ich ihn endlich vor mir und erwiedere das Feuer. Toll, wie sein Flugzeug durch Seiten und Rücklagen meinen Raketen entgeht. Es entsteht jetzt eine wilde Kurbelei mit dem Steuergriff. Immer wieder schieße ich eine Rakete hinter ihm her, und eine trifft. In einem roten Feuerball vergeht das gegnerische Flugzeug. Aufatmend bringe ich meinen Jet in eine ruhige Lage und will mir gerade über die Stirn wischen, da taucht im Hintergrund ein neuer Gegner auf. Rasend schnell wird er größer und eröffnet sofort das Feuer. Im gleichen Moment fegt ein zweiter von der rechten Seite heran. Im Eifer des Gefechtes haben sich die beiden von mir unbemerkt, herangemogelt und haben mich jetzt in der Zange. Rummms. Wieder habe ich einen Treffer einstecken müssen. Ein Blick auf die Anzeige sagt mir, daß ich nicht mehr viel abbekommen darf, will ich nicht ein vorzeitiges Ende finden. Also jage ich eine Rakete nach der anderen hinterher. Aber (wegen des schnellen Programmablaufs), irgendwie bekomme ich die beiden Angreifer nicht so richtig ins Fadenkreuz. So schieße ich eine Rakete mit IR-Suchkopf ab und habe Glück. Einen der beiden er-

wischt es voll. Und der letzte meiner drei Gegner mich. Nun wird es aber kritisch. Noch ein Volltreffer in meinen Jet und die ganze Sache ist für mich gelaufen. Wieder ein Blick von mir zum Höhenmesser und zum Radar, Höhe 8000 Fuß. Der Gegner ist hinter mir. Und jetzt kommt meine große Stunde, ich zwingte meine Maschine in einen gekonnten Looping. (Faszinierend, wie sauber und gekonnt Scrolling und Grafik "mitspielen"). Durch dieses Flugmanöver habe ich den Letzten meiner Gegner direkt vor mir. Feuer! Feuer! Feuer! Rummms. Auch dieser vergeht in einer gewaltigen Explosion. Huui. Tief Luft holend schaue ich zum Radar. Es zeigt mir den nächsten Pulk an. Doch ich möchte es nicht glauben, senkt sich die Nase meiner Maschine nach unten. Auch mit dem Steuerknüppel ist nichts mehr zu machen. Überrascht betrachte ich die Cockpitinstrumente. Und siehe da, durch meine wilde Fliegerei sind meine Treibstoffvorräte zu Ende. Nur das Seitenruder reagiert noch. Auch während meines Absturzes werden ich noch wild attackiert. Ohnmächtig verschleie ich meine letzten Raketen, die aber nun keine Wirkung mehr erzielen. Also zähle ich, während die Anzeige des Höhenmessers immer bedrohlicher wird, von 10 an abwärts ... 4, 3, 2 ... Rummms. Ein greller Blitz und ein lauter Knall sind mein unrühmliches Ende. Am Bildschirm bekomme ich die lapide Auskunft: Your Fighter is crashed!!! Gleichzeitig ertönt wieder im Menü eine ruhige Melodie, die ich vor diesem Flug als schwach eingestuft hatte. Doch jetzt ist sie so richtig beruhigend und verharmlost die ganze Angelegenheit. Noch während ich geschafft vor meinem Plus 4 sitze, schaltet das Programm in den Demo-Mode (Diesen sollte sich jeder vor dem ersten Flugversuch mal anschauen. Hier wird kurz gezeigt, was in diesem Programm alles steckt). Ich kann es nicht lassen, ich wähle nun Level 4.

Fazit:

ACE ist ein absolutes Top-Programm. Wer aber nicht den nötigen Ernst mitbringt, landet schnell wieder auf dem Boden der Tatsachen. Auch der relativ hohe Preis von DM 34,95 ist vollauf für dieses Superspiel gerechtfertigt.

Helmut Niedermeier

MOTH

Giftige Schmetterlinge sind Ihre Gegner!

Erinnern Sie sich noch an Roby, den Helden aus der Steinzeit? Nun ist aus seiner Ahnenreihe ein neuer Held gewachsen. Dieser heißt auch Roby, in dessen Rolle Sie nun schlüpfen. Es gibt einen Wissenschaftler der gerne die Welt erobern möchte und der eingesehen hat, daß mit brutalen Waffen nichts zu erreichen ist, so hat er sich Gehilfen geschaffen. Diese Gehilfen sind Schmetterlinge deren Körper mit einer giftigen Substanz getränkt sind. Jeder, der diese Schmetterlinge berührt, stirbt.

Sie sind nun auf der Insel dieses verrückten Wissenschaftlers. Ausgerechnet jetzt entweicht aus dem Labor ein ganzer Schwarm Schmetterlinge. Da Sie ein Freund der Menschheit sind und ein Nachfahre des legendären Roby, machen Sie sich daran diese Schmetterlinge wieder einzufangen.

Ausgerüstet mit einem Catcher und viel Mut suchen Sie nun die Schmetterlinge und Sie finden sie auch.

Nun versuchen Sie sie zu catchen, und zwar folgendermaßen: Sie warten bis ein Schmetterling in Reichweite ihres Catchers ist (direkt über ihrem Kopf) und drücken dann den Feuerknopf. Aber Vorsicht! Denn dieses Ungenießer ist unberechenbar, er fliegt wie es will. Ihren Roby steuern Sie mit dem Joystick nach links und rechts.

Beim Eintippen ist darauf zu achten, daß das Programm in zwei Teile geschrieben wurde. Wenn Sie eine 3 k Byte-Erweiterung haben, können sie die Teile zusammenfügen, die Zeilennummern sind hierfür vorgesehen. Ansonsten müssen Sie das erste Programm laden, starten und dann das zweite laden. Und nun: Weidmanns heil!

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

1 REM***** <12B>
2 REM* M O T H * <130>
3 REM* C O P Y R I G H T * <250>
4 REM* 1987 * <230>
5 REM* B Y * <228>
6 REM*EJOERN GROEGER* <209>
7 REM* U N D * <113>
8 REM* T I M M H A A S E * <208>
9 REM***** <136>
10 : : : : : : : : : : : : : : : : <22>
11 : : : : : : : : : : : : : : : : <23>
12 : : : : : : : : : : : : : : : : <24>
13 POKES6879,25:PRINT" {CLEAR}" <88>
14 POKES1,0:POKES2,28:POKES5,0:POKES6,28:CLR:CB= <91>
7168 <247>
15 READA:IFA=-1THENPOKE36869,255:GOTO18 <181>
16 FORN=0TO7:READB:POKECB+A*B+N,B:NEXT <155>
17 GOTO15
18 PRINT" {DOWN3 SPACE3 GREEN} { { { {SPACE} { { { {SPACE}
E} { { { {SPACE} { { { {SPACE} { <185>
19 PRINT" {SPACE3 BLUE} ' ' ' {SPACE} ' ' ' {SPACE} ' ' ' {S
PACE} ' {SPACE} ' <180>

```

```

20 PRINT"{SPACE3}"(RVSON C# S\ RVSOFF)"(SPACE)"(SPACE3)"(SPACE2)"(SPACE)"
21 PRINT"(SPACE3)"(SPACE2)"(SPACE)"(SPACE)"(SPACE)"(SPACE2)"(SPACE2)"
22 PRINT"(SPACE3)"(SPACE2)"(SPACE)"(SPACE)"(SPACE)"(SPACE2)"(SPACE)"
23 PRINT"(SPACE3)"(SPACE2)"(SPACE)"(SPACE2)"(SPACE)"
24 PRINT"(HOME YELLOW DOWN RIGHT12)\(DOWN LEFT7)\(DOWN RIGHT3)\
25 PRINT"(HOME DOWN RED)1(BLACK)^(RED)_(DOWN LEFT3 RED):(BLACK)(RED)#(DOWN LEFT3 RED)#(BLACK)%(RED)%"
26 PRINT"(HOME DOWN13 RIGHT17 RED)1(BLACK)^(RED)_(DOWN LEFT3 RED):(BLACK)(RED)#(DOWN LEFT3 RED)#(BLACK)%(RED)%"
27 PRINT"(HOME DOWN15 RIGHT6 RED)1(BLACK)^(RED)_(DOWN LEFT3 RED):(BLACK)(RED)#(DOWN LEFT3 RED)#(BLACK)%(RED)%"
28 REM
29 READX$:IFX$="#"(THENGOTO33
30 FORE=1TOLEN(X$):PRINT"(HOME DOWN2 RVSON)"
31 PRINT"(HOME DOWN21 RVSON)";TAB(20)MID$(X$,E,1)
32 PRINT"(HOME DOWN21 RIGHT RVSON)"CHR$(20):FORV=1TO110:NEXT:TEXTE:GOTO28
33 POKE36869,240
34 PRINT"(CLEAR)"
35 PRINT"(DOWN9)"
36 PRINT"(RED SPACES)PLEASE(SPACE)LOAD"
37 PRINT"(BLACK DOWN2 SPACE2)THE(SPACE)MAIN(SPACE)PROGRAMM(BLUE DOWN3)"
38 DATA16,56,56,120,253,255,255,255
39 DATA1,0,32,96,224,240,240,248,248
40 DATA2,0,0,0,1,3,7,71
41 DATA3,3,35,51,255,255,255,255
42 DATA4,255,255,255,255,255,255,255
43 DATA5,252,254,254,255,255,255,255
44 DATA6,0,0,0,192,248,240,224,240
45 DATA7,228,248,255,255,255,255,255
46 DATA8,231,255,255,255,255,255,255
47 DATA9,0,1,9,27,31,127,127,255
48 DATA10,0,0,192,236,253,254,255,255
49 DATA11,1,1,3,6,7,11,14,15
50 DATA12,128,128,192,224,160,112,240,240
51 DATA13,27,31,29,63,59,127,125,118
52 DATA14,216,248,184,124,236,254,182,254
53 DATA15,118,255,239,253,255,127,63,31
54 DATA16,182,251,95,255,223,126,248,240
55 DATA17,3,3,7,7,7,15,15
56 DATA18,224,192,192,224,224,224,192,192
57 DATA19,15,15,7,7,15,63,63,127
58 DATA20,192,224,224,224,224,240,248,252
59 DATA21,24,60,254,46,126,60,28,28
60 DATA22,31,63,62,254,254,254,126,60
61 DATA23,60,60,24,24,24,60,120
62 DATA24,60,60,60,62,182,247,182,36
63 DATA25,0,6,15,31,63,126,192,128
64 DATA26,60,62,63,63,62,24,24,24
65 DATA28,36,24,219,255,126,255,219,48
66 DATA29,3,3,0,0,56,124,124,230
67 DATA30,0,0,129,129,66,66,36,36
68 DATA31,192,192,0,0,28,62,62,102
69 DATA33,231,255,127,127,63,31,31,63
70 DATA27,24,60,165,189,255,231,255,231
71 DATA35,231,255,254,254,252,248,248,252
72 DATA36,127,207,195,230,254,60,0,0
73 DATA37,189,60,36,60,36,60,60,24
74 DATA38,254,243,195,71,126,60,0,0
75 DATA39,255,255,255,255,255,255,255
76 DATA40,3,100,40,145,82,98,90,127
77 DATA41,255,255,127,127,63,31,15,7
78 DATA42,192,224,240,248,252,252,254,255
79 DATA43,255,254,252,252,248,240,224,192
80 DATA32,0,0,0,0,0,0,0,0
81 DATA=1
82 DATA"(SIE(SPACE)HABEN(SPACE)DIE(SPACE)MOEGLICHKEIT,SICH(SPACE)ALS"
83 DATA"(SPACE)SCHMETTERLINGSFAENGER(SPACE)ZU(SPACE)VERSUCHEN"
84 DATA"-(SPACE)AUS(SPACE)IHREN(SPACE)LABOR(SPACE)SIND(SPACE)VERSEUCHTE(SPACE)"
85 DATA"SCHMETTERLINGE(SPACE)ENTWICHEN,DIE(SPACE)SIE(SPACE)"
86 DATA"NUN(SPACE)WIEDER(SPACE)EINFANGEN(SPACE)MUESSEN,ABER(SPACE)"

```

```

87 DATA"VORSICHT!!!{SPACE}BERUEHRT{SPACE}EIN{SPA
CE}SCHMETTERLING{SPACE}SIE,SIND{SPACE}SIE{SPACE}
DEM{SPACE}" <51>
88 DATA"LIEBEN{SPACE}GOTT{SPACE}EIN{SPACE}STUECK
{SPACE}NAEHER.{SPACE}ALSD" <252>
89 DATA",AUF{SPACE}GEHT{SPACE}DIE{SPACE}JAGT!!!{
SPACE3}" <172>
90 DATA"COPYRIGHT{SPACE}1987{SPACE}BY{SPACE}BJOE
RN{SPACE}GROEGER{SPACE}" <16>
91 DATA"UND{SPACE}TIMM{SPACE}HAASE{SPACE}!!!{SPA
CE}*****" <225>
92 DATA"{SPACE22}" <149>
93 DATA"#" <21>

```

Teil 2

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

94 REM COPYRIGHT          <80>
95 REM 1987                <238>
96 REM BY                  <126>
97 REMBJOERN GROEGER      <109>
98 REM &                   <61>
99 REM TIMM HAASE         <70>
100 PRINT"CLEAR":POKE36879,25:POKE36869,255:A=
19                          <67>
101 S2(1)=7887:S2(2)=7879:S2(3)=7896:POKE36878,1
5                          <221>
102 GOSUB 160              <183>
103 POKE198,0:WAIT198,1    <245>
104 A$="(GREEN)KL(DOWN LEFT2)MN(DOWN LEFT2)OP(DO
WN LEFT2 RED)QR(DOWN LEFT2)ST" <110>
105 PRINT"CLEAR":POKE36879,8 <193>
106 PRINT"(HOME RVSON YELLOW)S(PURPLE)C(GREEN)OC
RED)R(CYAN)E(WHITE SPACE)"; <190>
107 PRINT"(BLUE SPACE10)BA(SPACE8)1J"; <239>
108 PRINT"(SPACE2)BA(SPACE5)QDDF(SPACE5)BSDD"; <67>
109 PRINT"(SPACE)BHEF(SPACE3)IDDDDE(SPACE3)IDDD
D"; <121>
110 PRINT"SDDDEF(SPACE)SDDDDDDG(SPACE)IDDDDD"; <31>
111 PRINT"DDDDDDGDDDDDDDDDDDDDD"; <227>
112 PRINT"(DOWN4 RIGHT2)"A$ <122>
113 PRINT"(HOME DOWN7 RIGHT10)"A$ <243>
114 PRINT"(HOME DOWN9 RIGHT12)"A$ <19>
115 PRINT"(HOME DOWN18)" <102>
116 A$(1)="(YELLOW)U(CYAN)Y(DOWN LEFT2 RED)V(DO
WN LEFT CYAN)W" <3>
117 A$(2)="(YELLOW)U(CYAN)Y(DOWN LEFT2 RED)V(DO
WN LEFT CYAN)X" <28>
118 A$(3)="(YELLOW SPACE DOWN LEFT RED SPACE DO
WN LEFT CYAN SPACE)" <40>
119 A$(4)="(YELLOW SPACE2 DOWN LEFT2 RED SPACE D
OWN LEFT CYAN SPACE)" <170>
120 A$(5)="(CYAN)Z(DOWN LEFT YELLOW)U(SPACE DO
WN LEFT2 RED)V(DOWN LEFT CYAN)W" <174>
121 J9=PEEK(37137):POKE37154,127:J8=PEEK(37152):
POKE37154,255:POKE36876,0:POKE36877,0 <54>
122 IF(J9AND32)=0THENB=5:GOSUB151 <133>
123 IF(J9AND16)=0THENA=A+1:B=1:POKE36877,170:POK
E36877,0 <230>
124 IF(J8AND128)=0THENA=A+1:B=2:POKE36877,150:PO
KE36877,0 <79>
125 C=C+1:IFC=1THENPRINTTAB(A)"(UP3)"A$(1) <136>
126 IFC=2THENPRINTTAB(A)"(UP3)"A$(2):C=0 <177>
127 IFB=2THENPRINTTAB(A-1)"(UP3)"A$(3) <30>
128 IFB=1THENPRINTTAB(A+1)"(UP3)"A$(4) <38>
129 IFB=5THENPRINTTAB(A)"(UP4)"A$(5) <76>
130 IFA>19THENA=19:PRINT"(HOME DOWN16 RIGHT20 SP
ACE2 DOWN LEFT2 SPACE2 DOWN LEFT2 SPACE)" <222>
131 IFA<2THENA=1:PRINT"(HOME DOWN16 SPACE DOWN L
EFT SPACE DOWN LEFT SPACE)" <154>
132 REM BUTTERBEWEGUNG <82>
133 FORI=1TO3             <216>
134 SI(I)=INT(RND(1)*4) <88>
135 IFSI(I)=1THENS1=-22 <229>
136 IFSI(I)=2THENS1=-22 <225>
137 IFSI(I)=3THENS1=-22 <234>
138 IFSI(I)=4THENS1=-22 <0>
139 SI(I)=INT(RND(1)*4) <99>
140 IFSL(I)=1THENS1=-1 <247>
141 IFSL(I)=2THENS1=-1 <243>

```

```

142 IFPEEK(S2(I)+S1)<>32THENS1=0 <27>
143 IFS2(I)>100THENS1=-22 <70>
144 IFPEEK(S2(I)+S1)<>32THENS1=0 <29>
145 IFPEEK(S2(I)+S1)>20ANDPEEK(S2(I)+S1)<25THENG
0T0182 <9>
146 POKES2(I),32:S2(I)=82(I)+S1:POKES2(I),28 <17>
147 IFPEEK(S2(I))>20ANDPEEK(S2(I))<24THENG0T0182 <107>
148 NEXT <22>
149 FORI=1T03:IFPEEK(S2(I)+S1)>20ANDPEEK(S2(I)+S
1)<25THENG0T0182 <148>
150 NEXT:GOTO 121 <11>
151 REM <38>
152 PRINTTAB(A)"(UP4)"A$(5) <170>
153 FORI=1T03:IFPEEK(S2(I))=26THENS1=SC+10:S2(I)
=INT(RND(1)*20)+7812:POKE36877,200:I=0:GOSUB159 <105>
154 NEXTI <174>
155 REM <42>
156 FORI=1T0100:NEXT <209>
157 PRINTTAB(A)"(UP4 LEFT SPACE3 DOWN3)":B=0 <62>
158 RETURN <44>
159 PRINT"(HOME RVSON YELLOW)S(PURPLE)C(GREEN)O(
RED)R(BLUE)E(SPACE WHITE):";SC;"(DOWN18)":RETURN <103>
160 PRINT"(DOWN3 SPACE3 GREEN)((((SPACE) (((SPA
CE) (((SPACE) ((SPACE) ( <71>
161 PRINT"(SPACE3 BLUE)''''(SPACE)'''(SPACE)'''(
SPACE)'(SPACE)' <66>
162 PRINT"(SPACE3)'(RVSON C# S\ RVSOFF)'(SPACE)'
(SPACE)'(SPACE2)'(SPACE2)'(SPACE)' <103>
163 PRINT"(SPACE3)'(SPACE2)'(SPACE)'(SPACE)'(SPA
CE2)'(SPACE2)'' <37>
164 PRINT"(SPACE3)'(SPACE2)'(SPACE)'(SPACE)'(SPA
CE2)'(SPACE2)'(SPACE)' <154>
165 PRINT"(SPACE3)'(SPACE2)'(SPACE)'+(SPACE2)'(
SPACE2)'(SPACE)' <57>
166 PRINT"(HOME YELLOW DOWN RIGHT12)\(DOWN LEFT7
)\(DOWN RIGHT3)\ <229>
167 PRINT"(HOME DOWN RED)I(BLACK)^(RED)_(DOWN LE
FT3 RED)! (BLACK)[(RED)* (DOWN LEFT3 RED)* (BLACK)%
(RED)&" <129>
168 PRINT"(HOME DOWN13 RIGHT17 RED)I(BLACK)^(RED
)_(DOWN LEFT3 RED)! (BLACK)[(RED)* (DOWN LEFT3 RED
)* (BLACK)% (RED)&" <239>
169 PRINT"(HOME DOWN15 RIGHT6 RED)I(BLACK)^(RED)
_(DOWN LEFT3 RED)! (BLACK)[(RED)* (DOWN LEFT3 RED
)* (BLACK)% (RED)&" <185>
170 PRINT"(DOWN3 RVSON BLACK)COPYRIGHT(SPACE)198
7(SPACE)BY(SPACE YELLOW)ZEUS"; <45>
171 PO=PO+1:S2=36876:L=36878:POKE1,15 <87>
172 READP:IFP=-1THENRESTORE:IFPO=2THEN180 <88>
173 IFP=-1ANDPO=1THEN171 <111>
174 READ:POKES2,P:FORN=1TOD:NEXTN <59>
175 GOTO172 <7>
176 DATA147,300,159,160,175,160,170,160,159,160,
183,250,0,25,183,250,0,25,183,160 <97>
177 DATA191,160,170,160,175,160,159,250,0,25,159
,250,0,25,159,160 <71>
178 DATA175,160,170,160,159,160,147,160,201,160,
199,160 <51>
179 DATA191,160,183,160,175,160,170,160,159,160,
147,200,-1 <71>
180 POKES2,0:P=0 <243>
181 RETURN <67>
182 PRINT"(HOME DOWN14 RIGHT5 GREEN RVSON)GAME(S
PACE CYAN)OVER(SPACE YELLOW)!!!" <111>
183 POKE198,0:WAIT198,1:POKE198,0 <23>
184 RUN <66>

```

VC 20-FANS:
Der Tronic-Verlag
sucht noch Autoren
für VC20!

Jeep®

Betätigen Sie sich mal als Umweltsünder

Jeepers Aufgabe ist es, die Blätter eines Baumes umzufärben. Ein Käfer allerdings will diese Arbeit vereiteln, die zudem noch in einer bestimmten Zeit erledigt werden muß. JEEPER ist lauffähig auf einem VC 20 mit zusätzlichen 8 KB.

VC20 LISTING + CHECKSUMMEN. (V. V1.0)

```

1 PRINT "CLEAR WHITE DOWN3 RIGHTS"PLEASE(SPACE)IN$
ATT":POKE36879,8:POKE36878,15
2 FORI=0TO80
3 READX:POKEI+828,X:CC=CC+X
4 NEXT
5 REACT: IFCC<>XTHENPRINT "CLEARERROR(SPACE)IN$
PACE)3LINES(SPACE)6-16":END
6 DATA120,169,78,141,20,3,169,3
7 DATA141,21,3,88,96,0,0,0,0
8 DATA169,0,141,19,145,169,127
9 DATA141,34,145,173,17,145,41
10 DATA4,141,73,3,173,17,145,41
11 DATA8,141,74,3,173,17,145,41
12 DATA16,141,75,3,173,32,145,41
13 DATA128,141,76,3,173,17,145
14 DATA41,32,141,77,3,169,255
15 DATA141,34,145,169,128,141
16 DATA19,145,76,191,234,7167
17 FORI=5120T05847:READA:POKEI,A:NEXT
18 FORI=6400T06407:POKEI,255:NEXT:GOTO1800
19 DATA36,20,18,11,5,7,30,36
20 DATA68,70,138,146,17,32,32,32
21 DATA36,48,72,208,160,224,120,36
22 DATA34,98,81,73,136,4,4,4
23 DATA32,32,32,17,146,138,70,68
24 DATA36,30,7,5,11,18,20,36
25 DATA4,4,4,136,73,81,98,34
26 DATA36,120,224,160,208,72,40,36
27 DATA0,0,129,70,72,243,108,56
28 DATA12,3,224,24,4,3,14,16
29 DATA56,108,243,72,70,129,0,0
30 DATA16,14,3,4,24,224,3,12
31 DATA0,0,129,98,18,207,54,28
32 DATA28,54,207,18,98,129,0,0
33 DATA48,192,7,24,32,192,112,8
34 DATA8,112,192,32,24,7,192,48
35 DATA31,63,118,57,29,30,12,15
36 DATA13,14,7,7,3,0,0,0
37 DATA0,128,192,240,184,124,254,110
38 DATA158,60,208,248,228,10,5,2
40 DATA0,24,134,97,28,7,9,27
41 DATA21,12,7,28,97,134,24,0
42 DATA0,1,2,132,212,248,164,190
43 DATA190,164,248,212,132,2,1,0
44 DATA13,14,7,7,3,0,0,0
45 SC=0:LV=1:C=0:PRINT "CLEAR GREEN DOWN3":POKE
36869,205:POKE36879,8:FA=9
46 0$=" (SPACE2)PRPRPRPRPRPRPRPRPRPR(SPACE4)QSQSQSQS
SQSQSQSQSQSQSQSQSQSQ(SPACE UP)"

```

```

47 FOR=1T0B:X$=Q+S$<183>
48 PRINTX$<229>
49 NEXT<179>
50 F=33792:POKE36878,15:T=7<214>
55 M=4514+14:M1=M-22:M2=M1+1:M3=M2+22<83>
56 G=4272+2+132:G1=B+1:G2=G+22:G3=G2+1:T1$=""0000
00"<95>
57 POKEM,1:POKEM+F,1:POKEM1,0:POKEM1+F,1:POKEM2,<80>
2:POKEM2+F,1:POKEM3,3:POKEM3+F,1:SYS828<17>
58 IFC=>70THEN1000
59 PRINT"(HOME RIGHT2 PURPLE S* SA SB SC SD SPAC<0>
E YELLOW)"SC"(HOME DOWN2 RIGHT2 PURPLE SE SF SG<24>
SD SPACE YELLOW SPACE2)"MID$(TI$,3,2)+"(SPACE)"M<81>
ID$(TI$,5,2)"(HOME DOWN RIGHT2 PURPLE SH SD SI S
D SH SPACE YELLOW)"LV
60 IFPEEK(841)=0THENM=M-44:M1=M1-44:M2=M2-44:M3=M<24>
3-44:GOSUB69:GOTO64
61 IFPEEK(842)=0THENM=M+44:M1=M1+44:M2=M2+44:M3=M<81>
3+44:GOSUB100:GOTO64
62 IFPEEK(843)=0THENM=M-2:M1=M1-2:M2=M2-2:M3=M3-<42>
2:GOSUB160:GOTO64
63 IFPEEK(844)=0THENM=M+2:M1=M1+2:M2=M2+2:M3=M3+<241>
2:GOSUB200:GOTO64
64 X=INT(RND(1)*4):IFX=2THEN2100:GOTO65<99>
65 IFVAL(TI$)=000040THEN1700:GOTO66<233>
66 GOT058<221>
67 IFVAL(TI$)=000040THENGOTO1700<0>
69 IFPEEK(M)=32THEN800<77>
70 IFPEEK(M)=24THENSC=SC+0:C=C+0<57>
71 IFPEEK(M)=17THENSC=SC+1:C=C+1<113>
72 IFC=>70THEN1000<31>
73 IFM=G2THENS00<39>
76 POKEM,1:POKEM+F,1:POKEM1,0:POKEM1+F,1:POKEM2,<35>
2:POKEM2+F,1:POKEM3,3:POKEM3+F,1
77 POKEM+44,24:POKEM+44+F,3:POKEM3+44,19:POKEM3+<147>
44+F,3:POKEM2+44,18:POKEM2+44+F,3
78 POKEM1+44,16:POKEM1+44+F,3<128>
79 GOSUB900<158>
80 RETURN<222>
100 IFPEEK(M)=32THEN840<152>
101 IFPEEK(M)=24THENSC=SC+0:C=C+0<88>
102 IFPEEK(M)=17THENSC=SC+1:C=C+1<144>
103 IFC=>70THEN1000<62>
104 IFM=G2THENS00<78>
107 IFVAL(TI$)=000040THENGOTO1700<40>
120 POKEM,5:POKEM+F,1:POKEM1,4:POKEM1+F,1:POKEM2,<211>
6:POKEM2+F,1:POKEM3,7:POKEM3+F,1
130 POKEM-44,24:POKEM-44+F,3:POKEM3-44,19:POKEM3-<113>
44+F,3:POKEM2-44,18:POKEM2-44+F,3
140 POKEM1-44,16:POKEM1-44+F,3<208>
149 GOSUB900<228>
150 RETURN<36>
160 IFPEEK(M)=32THEN820<190>
161 IFPEEK(M)=17THENSC=SC+1:C=C+1<203>
162 IFC=>70THEN1000<122>
163 IFPEEK(M)=24THENSC=SC+0:C=C+0<150>
167 IFVAL(TI$)=000040THENGOTO1700<100>
168 IFM=G2THENS00<134>
170 POKEM1,12:POKEM1+F,1:POKEM,13:POKEM+F,1:POKE<34>
M2,14:POKEM2+F,1:POKEM3,15:POKEM3+F,1
175 POKEM+2,24:POKEM+2+F,3:POKEM1+2,16:POKEM1+2+<206>
F,3:POKEM2+2,18:POKEM2+2+F,3
176 POKEM3+2,19:POKEM3+2+F,3<28>
189 GOSUB900<12>
190 RETURN<76>
200 IFPEEK(M)=32THEN860<18>
201 IFPEEK(M)=17THENSC=SC+1:C=C+1<243>
202 IFC=>70THEN1000<161>
203 IFPEEK(M)=24THENSC=SC+0:C=C+0<190>
207 IFVAL(TI$)=000040THENGOTO1700<140>
208 IFM1=GTTHENS00<64>
210 POKEM1,9:POKEM1+F,1:POKEM,11:POKEM+F,1:POKEM<143>
2,B:POKEM2+F,1:POKEM3,10:POKEM3+F,1
225 POKEM-2,24:POKEM-2+F,3:POKEM1-2,16:POKEM1-2+<154>
F,3:POKEM2-2,18:POKEM2-2+F,3
226 POKEM3-2,19:POKEM3-2+F,3<95>
249 GOSUB900<72>
250 RETURN<136>
500 FORU=1STO0STEP-3<116>
510 POKE36878,U<189>
520 POKE36877,140<86>
530 POKE36879,42<146>
540 FORI=0TO50:NEXTI<218>
550 POKE36879,0<4>
560 FORI=0TO50:NEXTI<238>
570 NEXTU<104>
```

```

580 POKE36877,0;POKE36878,15;GOTO1971 <21>
599 END <217>
800 POKEM+44,24;POKEM+44+F,3;POKEM3+44,19;POKEM3 <105>
+44+F,3;POKEM2+44,18;POKEM2+44+F,3 <169>
805 POKEM1+44,16;POKEM1+44+F,3;GOTO950
820 POKEM2,24;POKEM2+2+F,3;POKEM3+2,19;POKEM3+2+ <3>
F,3;POKEM2+2,18;POKEM2+2+F,3 <255>
825 POKEM1+2,16;POKEM1+2+F,3;GOTO950
840 POKEM-44,24;POKEM-44+F,3;POKEM3-44,19;POKEM3 <58>
-44+F,3;POKEM2-44,18;POKEM2-44+F,3 <227>
845 POKEM1-44,16;POKEM1-44+F,3;GOTO950
860 POKEM-2,24;POKEM-2+F,3;POKEM3-2,19;POKEM3-2+ <197>
F,3;POKEM2-2,18;POKEM2-2+F,3 <56>
865 POKEM1-2,16;POKEM1-2+F,3;GOTO950 <32>
900 POKE36877,135;POKE36877,0 <31>
910 RETURN
950 FORI=1TO10;POKE36878,15;POKE36877,140+T;NEXT <173>
POKE36877,0
952 FORI=1TO22;POKE36865,38;POKE36865,39;POKE368 <152>
65,40;POKE36865,39 <62>
953 NEXT <36>
954 GOTO1971 <83>
960 BOT045
1000 BN=INT (VAL (TI$));BD=INT (SC/BN+LV);SC=SC+10+ <107>
BD;LV=LV+1;PRINT "CLEAR GREEN DOWN3" <125>
1040 POKE36869,205 <85>
1050 C=0 <36>
1060 FA=FA+1;IFFA=11THENFA=14 <100>
1061 IFFA=15THENFA=8 <1>
1062 IFFA=9THENFA=10 <199>
1070 POKE36879,FA <207>
1080 GOTO46
1500 POKEG2-2,17;POKEG2-2+F,5;POKEG3-2,19;POKEG3 <107>
-2+F,5;POKEG-2,16;POKEG-2+F,5 <19>
1505 POKEG1-2,18;POKEG1-2+F,5 <194>
1510 G=4272+2+132;G1=G+1;G2=G+22;G3=G2+1 <129>
1520 R=INT (RND (1)*5) <232>
1530 ONRGOTO1540,1560,1550,1570
1540 G=4272+2+176;G1=G+1;G2=G+22;G3=G2+1;R=0;GOT <22>
058
1550 G=4272+2-88;G1=G+1;G2=G+22;G3=G2+1;R=0;GOTO <90>
58 <1>
1560 G=4272+2;G1=G+1;G2=G+22;G3=G2+1;R=0;GOTO58
1570 G=4272+2+44;G1=G+1;G2=G+22;G3=G2+1;R=0;GOTO <17>
58 <211>
1580 R=0;GOTO58
1700 POKE36878,15;FORA=1TO7;FORT=1TO30STEP6;POKE <218>
36876,150+T;NEXT;POKE36876,0;NEXTA <27>
1710 GOTO1971
1800 POKE36865,140;POKE36869,205;POKE36878,15;PR <14>
INT "CLEAR DOWNS RIGHT PURPLE"Y[C][I];# <167>
1810 PRINT "RIGHT"Z\^\ \SPACE$ <51>
1820 PRINT "DOWN RED RIGHT"X% <156>
1830 PRINT "DOWN YELLOW RIGHT" '()*+,-./" <162>
1840 PRINT "DOWN BLUE RIGHT" (SPACE)1984"
1841 FORI=140TO38STEP-1;POKE36865,I;POKE36876,11 <183>
0+I;POKE36876,0;NEXT
1900 SYSB28;B$="{HOME DOWN16}";C$="{SPACE5};<=>? <244>
SPACE12}" <88>
1910 D$=RIGHT$(C$,1)+LEFT$(C$,21);C$=D$ <208>
1920 PRINTB*C$ <70>
1930 POKE646,INT (RND (1)*7)+1 <91>
1940 IFPEEK (845)=0THEN1970
1950 FDRA=1TO2;POKE36876,200;FORT=1TO2;POKE36876 <99>
,0;NEXTT,A <249>
1960 GOTO1910 <73>
1970 GOTO45
1971 L$="{HOME WHITE DOWN20}";J$="{<=>?SPACE3 S <251>
R SS ST SK SPACE SU SV SK SM SPACE5}" <173>
1972 N$=RIGHT$(J$,21)+LEFT$(J$,1);J$=N$ <143>
1973 PRINTL$;J$ <223>
1974 IFPEEK (845)=0THEN1977 <235>
1975 FORT=1TO60;NEXT <43>
1976 GOTO1972 <79>
1977 REM <36>
1978 FORT=1TO500;NEXT <31>
1979 GOTO960 <203>
2100 REM <154>
2105 G=G+2;G1=G1+2;G2=G2+2;G3=G3+2 <151>
2110 IFPEEK (G)=32THEN1500 <132>
2120 IFPEEK (G2)=24THENC=C-1
2121 IFG1=MORG1=MORG1=MORG3=MORG3=MORG3=M10RG3=M <18>
20RG3=M3THEN500
2130 POKEG,20;POKEG+F,7;POKEG1,22;POKEG1+F,7;POK <103>
EG2,21;POKEG2+F,7;POKEG3,23;POKEG3+F,7
2140 POKEG-2,16;POKEG-2+F,5;POKEG1-2,18;POKEG1-2

```

```

4F,5:POKEG2-2,17:POKEG2-2+F,5:POKEG3-2,19          (250)
2145 POKE36876,240:POKE36876,0                          (202)
2150 POKEG3-2+F,5                                         (245)
2160 X=0:GOTO58                                             (31)
3000 DATA0,3,3,3,3,3,3,3                                (108)
3010 DATA3,3,3,195,195,102,126,60                       (192)
3020 DATA0,127,127,96,96,96,96,124                     (190)
3030 DATA124,96,96,96,96,96,96,127,127                 (46)
3040 DATA0,124,126,99,99,99,99,99                       (175)
3050 DATA126,124,96,96,96,96,96,96                     (23)
3060 DATA126,124,102,102,99,99,99,99                     (202)
3070 DATA0,0,0,0,0,0,0,0                                (224)
3080 DATA0,12,24,48,0,0,0,0                             (139)
3090 DATA0,0,0,0,0,0,0,0                                (245)
3100 DATA0,60,126,195,195,192,192,120                  (245)
3110 DATA60,6,3,3,195,195,126,60                        (198)
3120 DATA0,124,66,66,124,66,66,124                      (255)
3130 DATA0,68,68,68,56,16,16,16                         (125)
3140 DATA24,36,66,66,126,66,66,231                      (47)
3150 DATA0,0,0,0,0,0,48,48                              (65)
3160 DATA231,66,98,82,74,70,66,231                      (55)
3170 DATA127,65,64,68,124,68,65,127                    (12)
3180 DATA56,16,16,16,16,16,16,16                       (239)
3190 DATA248,68,66,66,66,66,68,248                      (219)
3200 DATA231,66,66,66,126,66,66,231                     (186)
3210 DATA248,68,66,66,68,120,68,238                     (115)
3230 DATA127,8,8,8,8,8,8,8,8                            (89)
3240 DATA0,60,66,70,90,98,66,60                         (42)
3250 DATA0,16,48,80,16,16,16,124                        (213)
3260 DATA0,60,66,2,12,48,64,126                         (134)
3270 DATA0,60,66,2,28,2,66,60                           (171)
3280 DATA0,4,12,20,36,126,4,4                           (32)
3290 DATA0,126,64,120,4,2,68,56                         (166)
3300 DATA0,30,32,64,124,66,66,60                        (119)
3310 DATA0,126,66,4,8,16,16,16                           (41)
3320 DATA0,60,66,66,60,66,66,60                         (57)
3330 DATA0,60,66,66,62,2,4,56                           (18)
3340 DATA0,124,130,186,162,186,130,124                  (20)
3350 DATA0,0,119,85,119,70,71,0                         (116)
3360 DATA0,0,119,68,103,65,119,0                        (54)
3370 DATA0,3,114,66,115,18,114,2                       (221)
3380 DATA0,215,20,20,23,21,20,20                        (144)
3390 DATA0,188,160,160,184,32,160,188                   (86)
3400 DATA0,60,66,64,60,2,66,60                          (116)
3410 DATA0,28,34,64,64,64,34,28                         (134)
3420 DATA0,24,36,66,66,60,36,24                         (172)
3430 DATA0,124,66,66,124,72,68,66                       (188)
3440 DATA0,126,64,64,120,64,64,126                      (196)
3450 DATA0,127,8,8,8,8,8,8                               (102)
3460 DATA0,56,16,16,16,16,16,56                         (99)
3470 DATA0,130,198,170,146,130,130,130                 (134)
3480 DATA0,64,64,64,64,64,64,126                       (134)
3490 DATA0,66,66,66,36,36,24,24                         (143)
3500 DATA15,6,6,6,198,198,108,56                        (87)
3510 DATA126,126,96,120,120,96,126,126                 (21)
3520 DATA15,6,6,6,198,198,108,56                        (107)
3530 DATA126,126,96,120,120,96,126,126                 (41)
3540 DATA126,126,99,99,126,126,96,96                     (29)
3550 DATA126,126,99,99,126,126,99,99                     (215)
3560 DATA0,12,24,48,0,0,0,0                             (108)
3570 DATA60,102,195,120,30,195,102,60                   (118)
3580 DATA62,127,192,199,195,195,126,60                 (168)
3590 DATA28,62,99,99,127,127,99,99                      (37)
3600 DATA99,99,119,127,107,99,99,99                     (187)
3610 DATA60,126,195,195,195,195,126,60                  (159)
3620 DATA99,99,99,54,54,54,28,28                       (198)
3630 DATA248,252,198,198,252,248,220,206               (93)
3640 DATA0,0,0,0,0,0,0,0                                (29)
3650 DATA0,0,0,0,0,0,0,0                                (39)
3660 DATA0,0,0,0,0,0,0,0                                (49)

```

**Den VC20-Checksummer
können Sie beim Verlag
bestellen
(Kassette 10,- DM)!**

MONITOR VC20

Das vorliegende kleine Programm dient dazu, Maschinensprachedaten komfortabel in den Speicher des VC 20 einzugeben oder Manipulationen in den Speichereinstellen oder in Maschinensprachprogrammen durchzuführen. Gestartet wird der MONITOR mit SYS 7241, (SPEICHERSTELLE). Der Inhalt der Speicherstelle wird nun ausgegeben. Der blinkende Cursor erscheint unmittelbar dahinter. Geben Sie nun eine neue

zweistellige Zahl im Hexadezimalformat ein, nimmt die Speicherstelle diesen neuen Wert an. Die einzelnen Unterprogramme kann man auch separat nutzen: Mit der Routine DEZ-HEX kann man eine dezimale Zahl (0-63999) in eine vierstellige Hexadezimale Zahl umwandeln. Sie gehen wie folgt vor:
1. SYS 680, (DEZ. Zahl)
2. SYS 695
In dieser Weise kann man auch die anderen Unterprogramme nutzen.

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

```
0 PRINT "CLEAR SPACE6" * MONITOR * (SPACE7 DOWN3) ** B
Y (SPACE) MARCUS (SPACE) GOERLING **
1 FORT=680761: READA: POKET, A: NEXTT
2 FORT=716807399: READA: POKET, A: NEXTT
3 PRINT "DOWN2" * (SPACE) AUSGABE (SPACE) = (SPACE RVS
ON) 7241 (RVSOFF), (RVSON) ADR (RVSOFF SPACE) * (DOWN) *
(SPACE) EINGABE (SPACE) = (SPACE RVSON) 7296 (RVSOFF),
(RVSON) ADR (RVSOFF SPACE) *
4 PRINT "DOWN3" * (SPACE) SYS4109 (UP3) : POKE631, 13:
POKE198, 1: NEW
10 REM * EINGABE (680) *
20 DATA32, 253, 206, 32, 107, 201, 165, 20, 133, 251, 165,
21, 133, 252, 96
100 REM * HEX-CODE (695) *
110 DATA165, 252, 32, 190, 2, 165, 251, 72, 74, 74, 74,
```

```
32, 201, 2, 104, 41, 15, 201, 10, 144, 2
120 DATA105, 6, 105, 48, 76, 210, 255
300 REM * HEX-CODE (724) *
310 DATA164, 251, 133, 251, 32, 190, 2, 132, 251, 96
400 REM * UM 4 ERHOEHEN (734) *
410 DATA162, 0, 230, 251, 208, 2, 230, 252, 224, 3, 240, 4,
232, 76, 224, 2, 96
500 REM * WARTEN (751) *
510 DATA32, 228, 255, 201, 32, 240, 3, 76, 239, 2, 96
1000 REM * HAUPTROUTINE (7168) *
1010 DATA32, 168, 2, 32, 183, 2, 162, 0, 169, 32, 32, 210, 2
55, 165, 251, 141, 24, 28, 165, 252, 141, 25, 28
1020 DATA189, 0, 0, 149, 0, 32, 212, 2, 224, 3, 240, 4, 232,
76, 8, 28, 169, 32, 32, 210, 255, 162, 0, 181, 0
1030 DATA32, 86, 28, 224, 3, 240, 4, 232, 76, 46, 28, 169, 1
3, 32, 210, 255, 32, 239, 2, 32, 222, 2, 76, 3, 28
1300 REM * AUSGABE HEX-CODE (7241), ADR *
1310 DATA169, 5, 32, 210, 255, 169, 238, 141, 15, 144, 76,
0, 28
1400 REM * ZEICHEN (7254) *
1410 DATA162, 0, 181, 0, 201, 126, 16, 7, 201, 160, 48, 10,
76, 120, 28, 201, 32, 48, 3, 76, 120, 28
1420 DATA32, 210, 255, 224, 3, 240, 4, 232, 76, 88, 28, 96,
169, 46, 32, 210, 255, 76, 111, 28
2000 REM * EINGABE HEX-CODE (7296), ADR *
2010 DATA32, 168, 2, 169, 62, 32, 210, 255, 32, 183, 2, 169
58, 32, 210, 255, 32, 207, 255, 133, 254
2020 DATA32, 165, 28, 32, 158, 28, 76, 131, 28
2500 REM * UM 1 ERHOEHEN (7324) *
2510 DATA230, 251, 208, 2, 230, 252, 96
3000 REM * DEZ-CODE (7333) *
3010 DATA165, 254, 32, 193, 28, 10, 10, 10, 10, 133, 254, 3
2, 207, 255, 133, 255, 165, 255, 32, 193, 28
3020 DATA234, 234, 133, 255, 76, 203, 28
3200 REM * UMRECH. DEZ-WERT (7359) *
3210 DATA201, 64, 16, 3, 105, 208, 96, 105, 200, 96
3300 REM * POKEN DEZ-WERT (7371) *
3310 DATA165, 254, 101, 255, 133, 253, 198, 253, 165, 251
141, 224, 28, 165, 252, 141, 225, 28, 165, 253
3320 DATA141, 0, 0, 169, 13, 32, 210, 255, 96
ENDE DES LISTINGS
```

```
<44>
<2>
<218>
<231>
<151>
<60>
<72>
<127>
<143>
<236>
<189>
<210>
<133>
<224>
<52>
<36>
<190>
<221>
<40>
<227>
<157>
<28>
<189>
<245>
<95>
<237>
<239>
<210>
<174>
<125>
```

C-16
plus 4

Toooooor!

ELFMETERSCHIESSEN

für den C-16

Bestimmt haben Sie schon einmal bei einem Elfmeterschießen mitgefeiert und wollten am liebsten in die Rolle des Schützen oder des Tormanns schlüpfen, aber Sie konnten nicht. Dieses Programm ermöglicht es Ihnen, ein Elfmeterschießen zwischen zwei Nationalmannschaften durchzuführen. Zu Beginn muß man die Spielerzahl mit "RETURN" einstellen, wählt man "1 PLAYER", so spielt man gegen den Computer. Anschließend wählt man eine von vier möglichen Nationen, indem man mit dem Joystick den Pfeil nach links oder rechts auf die Nation fährt, für die man spielen will und den Feuerknopf drückt, wobei der obere Pfeil für den ersten Spieler bestimmt ist. Sobald die

Nationen ausgewählt sind, beginnt das Spiel. Damit man weiß, wer zu schießen hat, begleitet den Schützen seine Nationalhymne. Zuerst hat jeder Spieler fünf Schüsse, die jeweils hintereinander "abgefeuert" werden. Sollte es dann unentschieden stehen, so geht es im K.O.-System weiter, bis eine Entscheidung fällt. Nun zum wichtigsten Teil, der Steuerung. Sowohl der Schütze als auch der Torwart müssen den Joystick, nachdem der Schiedsrichter gepfiffen hat und während der Schütze anläuft, diagonal in eine der vier Ecken drücken. Drückt man nicht, zu spät oder nicht genau, so geht der Schuß über das Tor bzw. der Torwart bleibt auf der Linie stehen.

Eingabehinweise:

Zuerst tippt man das Hauptprogramm (Basic-Listing) ab und speichert es. Der zweite Teil, der aus reinem Maschinencode besteht, wird nun mit dem Maschinensprache-Monitor eingetippt und von dort aus mit

"ELFMETER/DATAS", 01,1E00,4000 auf Kassette abgespeichert. Wer eine Floppy besitzt, muß die Zahl 01 in 08 ändern. Wenn Sie nun spielen wollen, brauchen Sie nur das Hauptprogramm laden, weil der 2. Teil automatisch nachgeladen wird.

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

```
100 REM *****
101 REM *
102 REM * ELFMETER-SCHIESSEN *
103 REM *
104 REM *****
105 REM *
106 REM * (C) SEPT. 1986 BY *
107 REM *
108 REM * MICHAEL POTSCHE *
109 REM *
110 REM *****
111 REM *
112 REM * GRAPHIC: C. AUMEIER *
```

```
<205>
<198>
<91>
<200>
<209>
<202>
<126>
<204>
<34>
<206>
<215>
<208>
<84>
```

```
113 REM * M. POTSCHE *
114 REM *
115 REM *****
116 POKE56, 30: POKE52, 30: POKE55, 0: POKE51, 0
117 IFPEEK(210)=64 THEN 120
118 PRINT "CLEAR": COLOR0, 1: COLOR4, 1: CHAR1, 11, 10
119 "WHITE" LOAD (SPACE) ELFMETER-CODE: COLOR1, 8, 4
119 POKE210, 64: CHAR1, 0, 19, "": LOAD "ELFMETER/DATAS"
120 A=CHR$(178)+CHR$(179)+" (DOWN LEFT2)" + CHR$(180)+CHR$(181)
121 B=CHR$(182)+CHR$(183)+" (DOWN LEFT2)" + CHR$(184)+CHR$(185): C=" (SPACE2 DOWN LEFT2 SPACE2)": BY
S10502
122 C=RND(TI): PRINT "CLEAR": SYS10151: SYS10354: C
OLOR1, 5, 4: P1=16: P2=24: B=1: C=1: M1=0: M2=0: X=0
123 IFPEEK(198)=1 THEN B=B+1: ELSE 126
124 IFB/2=INT(B/2) THEN B=SYS10386: C=2
125 IFB/2<>INT(B/2) THEN B=SYS10354: C=1
126 IFC=2 THEN CHAR1, P2, 22, B
127 IFC=1 THEN CHAR1, P2, 22, C
128 CHAR1, P1, 17, A: GOSUB165
129 IFM2=M1 THEN M2=0
130 IFC=2 THEN IFM1>0 AND M2>0 THEN 133
131 IFC=1 THEN IFM1>0 THEN 133
132 GOTO123
133 SI=0: S2=0: H=1: SYS9076: IFC=1 THEN M2=INT(RND(1)
+4)+1: IFM2=M1 THEN 133: H=1
134 GOSUB174
135 PRINT "CLEAR": POKE65299, 48: COLOR0, 6, 3: COLOR
4, 6, 4: POKE65302, 0: POKE65303, 107
136 SYS7688: POKE208, 120: SYS8539: SYS8282: POKE3118
28: POKE3198, 28: TW=0: AD=0
137 ONM1GOTO140, 141, 138, 139
138 SYS8964: GOTO142
139 SYS8978: GOTO142
140 SYS8992: GOTO142
141 SYS9006
142 ONM2GOTO145, 146, 143, 144
143 SYS9020: GOTO147
144 SYS9034: GOTO147
145 SYS9048: GOTO147
146 SYS9062
147 S3=INT(S1/10): POKE3196, S3: POKE3197, S1-S3*10
148 S3=INT(S2/10): POKE3199, S3: POKE3200, S2-S3*10
149 FORT=0T04000: NEXT
150 SYS8523: FORT=0T015: VOLB=SOUND1, 1000+T, 1.5: S0
UND2, 1020-T, 1.5: NEXT
151 FORT=120T040STEP-40: POKE208, T: SYS8539: GOSUB1
80: SYS8582: GOSUB180: NEXT
152 SYS8625: GOSUB180: SYS8666: GOSUB180
153 IFTW=0 THEN 154: ELSE SYS8370: SYS87M
154 IFAD=0 THEN 155: ELSE SYS85AD: GOTO157
155 SA=3847: POKE3885, 175: FORT=0T019: PR=SA-39: Z=P
EEK(PR): POKEPR, 158
156 FORA=0T010: NEXT: SA=SA-39: POKE3A, Z: NEXT: POKE2
09, 0
157 IFH=1 AND PEEK(209)=255 THEN S1=S1+1: GOSUB200
158 IFH=2 AND PEEK(209)=255 THEN S2=S2+1: GOSUB200: G0
T0160
159 FORT=0T01000: NEXT
160 X=X+1: IFY=0 THEN IFX=5 AND H=1 THEN H=2: X=0: GOSUB1
79: GOTO162
161 IFY=0 THEN IFX=5 AND H=2 THEN H=1: X=0: GOSUB212: Y=1
: X=0: H=1: GOSUB174: GOTO135
162 IFY=1 THEN IFX=1 AND H=1 THEN H=2: X=0: GOSUB179: G0
T0164
163 IFY=1 THEN IFX=1 AND H=2 THEN H=1: X=0: GOSUB212: X=0
: H=1: GOSUB174
164 GOTO135
165 IFM1<>0 THEN 169
166 IFJOY(1)=7 THEN IFP1>15 THEN CHAR1, P1, 17, C: P1=P
1-8
167 IFJOY(1)=3 THEN IFP1<25 THEN CHAR1, P1, 17, C: P1=P
1+8
168 IFJOY(1)>127 THEN M1=P1/8
169 IFC=10 RM2>0 THEN RETURN
170 IFJOY(2)=7 THEN IFP2>15 THEN CHAR1, P2, 22, C: P2=P
2-8
171 IFJOY(2)=3 THEN IFP2<25 THEN CHAR1, P2, 22, C: P2=P
2+8
172 IFJOY(2)>127 THEN M2=P2/8
173 RETURN
174 ONM1GOTO175, 176, 178, 177
175 SYS10837: RETURN
176 SYS11298: RETURN
```

```
<58>
<211>
<220>
<239>
<25>
<192>
<178>
<38>
<158>
<45>
<249>
<130>
<10>
<183>
<195>
<61>
<106>
<235>
<137>
<209>
<110>
<234>
<219>
<240>
<149>
<138>
<163>
<142>
<251>
<109>
<132>
<157>
<182>
<4>
<132>
<93>
<98>
<1>
<18>
<163>
<242>
<207>
<6>
<30>
<121>
<36>
<82>
<230>
<10>
<161>
<18>
<252>
<247>
<24>
<1>
<200>
<8>
<31>
<8>
<235>
<59>
<158>
<110>
<126>
177 SYS11693: RETURN
178 SYS12116: RETURN
179 ONM2GOTO175, 176, 178, 177
180 IFH=2 THEN 196
181 IFJOY(1)=0 OR JOY(1)>127 THEN 187
182 ONJOY(1) GOTO187, 183, 187, 184, 187, 185, 187, 186
183 AD=9105: GOTO188
184 AD=9389: GOTO188
185 AD=9692: GOTO188
186 AD=9813: GOTO188
187 GOTO188
188 IFH=2 THEN 198
189 IFC=1 THEN JO=INT(RND(1)*4+1)*2: ELSE JO=JOY(2):
IFJOY(2)=0 OR JOY(2)>127 THEN 195
190 ONJOY(2) GOTO195, 191, 195, 192, 195, 193, 195, 194
191 TW=8448: RETURN
192 TW=8422: RETURN
193 TW=8402: RETURN
194 TW=8478: RETURN
195 RETURN
196 IFC=1 THEN JO=INT(RND(1)*4+1)*2: ELSE JO=JOY(2):
IFJO=0 OR JO>127 THEN 187
197 ONJOY(2) GOTO187, 183, 187, 184, 187, 185, 187, 186
198 IFJOY(1)=0 OR JOY(1)>127 THEN 195
199 ONJOY(1) GOTO195, 191, 195, 192, 195, 193, 195, 194
200 SYS8707: FORT=0T02: SYS8282: SYS8307: GOSUB202: S
YS8282: SYS8282: SYS8347
201 GOSUB202: SYS8321: GOSUB202: NEXT: GOSUB202: RETU
RN
202 FORA=0T0250: NEXT: RETURN
203 SYS8282: SYS8894: ONM1GOTO204, 205, 206, 207
204 SYS8936: GOTO208
205 SYS8950: GOTO208
206 SYS8908: GOTO208
207 SYS8922
208 FORT=0T0100: SYS8776: FORA=0T0100: NEXT: SYS8836
209 IFJOY(1)>127 OR JOY(2)>127 THEN SYS8707: RUN
210 FORA=0T040: NEXT: A, T: GOTO208
211 SYS8282: SYS8894: ONM2GOTO204, 205, 206, 207
212 IFB1>52 THEN GOSUB174: GOTO203
213 IFB1<52 THEN GOSUB179: GOTO211
214 RETURN
ENDE DES LISTINGS
```

M. C.-Programm

ready.

monitor

```
pc sr ac xr yr sp
ffff 00 ff ff ff f8
```

```
>1e00 a9 79 a0 00 99 00 08 99
>1e08 00 09 99 00 0a 99 00 0b
>1e10 c8 d0 f1 a9 00 a0 00 99
>1e18 00 08 99 28 08 99 50 08
>1e20 99 78 08 99 a0 08 c8 c0
>1e28 0d d0 e0 a9 3a a0 00 99
>1e30 0f 08 99 39 08 99 64 08
>1e38 99 8f 08 99 b9 08 99 e4
>1e40 08 99 0f 09 c8 c0 09 d0
>1e48 e6 a0 00 99 39 09 c8 c0
>1e50 07 d0 f8 a0 00 99 64 09
>1e58 c8 c0 04 d0 f8 8d 8f 09
>1e60 a9 2c a0 00 99 db 0a 99
>1e68 03 0b 99 2b 0b 99 53 0b
>1e70 99 7b 0b 99 a3 0b 99 cb
>1e78 0b c8 c0 03 d0 e6 a9 79
>1e80 8d 2d 0b a0 00 b9 f2 3d
>1e88 99 0f 0c 99 8f 0c 99 0f
>1e90 0d b9 fb 3d 99 39 0c 99
>1e98 b9 0c b9 04 3e 99 64 0c
>1ea0 99 e4 0c c8 c0 09 d0 dd
>1ea8 a0 00 b9 fb 3d 99 39 0d
```



```

>1eb0 c8 c0 07 d0 f5 a0 00 b9
>1eb8 04 3e 99 64 0d c8 c0 04
>1ec0 d0 f5 a9 8b 8d 8f 0d a0
>1ec8 00 b9 0d 3e 99 c8 0c b9
>1ed0 11 3e 99 70 0f b9 15 3e
>1ed8 99 c6 0f b9 19 3e 99 cb
>1ee0 0d b9 1d 3e 99 4c 0e b9
>1ee8 21 3e 99 77 0e b9 25 3e
>1ef0 99 a2 0e b9 29 3e 99 0f
>1ef8 0e b9 2d 3e 99 3a 0e b9
>1f00 31 3e 99 65 0e 99 90 0e
>1f08 b9 35 3e 99 3c 0f b9 39
>1f10 3e 99 d0 0d c8 c0 04 d0
>1f18 b0 a0 00 b9 3d 3e 99 9a
>1f20 0f b9 42 3e 99 79 0d c8
>1f28 c0 05 d0 ef a0 00 b9 47
>1f30 3e 99 59 0d b9 4d 3e 99
>1f38 d9 0c c8 c0 06 d0 ef a0
>1f40 00 b9 53 3e 99 21 0e b9
>1f48 5a 3e 99 2e 0d b9 61 3e
>1f50 99 03 0d c8 c0 07 d0 e9
>1f58 a0 00 b9 68 3e 99 f4 0c
>1f60 99 e6 0e b9 6b 3e 99 1f
>1f68 0d b9 6e 3e 99 4a 0d b9
>1f70 71 3e 99 75 0d b9 74 3e
>1f78 99 a0 0d b9 77 3e 99 f6
>1f80 0d b9 7a 3e 99 cd 0e b9
>1f88 7d 3e 99 97 0d b9 8c 3e
>1f90 99 11 0f b9 83 3e 99 d5
>1f98 0d b9 86 3e 99 84 0d b9
>1fa0 89 3e 99 a6 0d b9 77 3e
>1fa8 99 b0 0e c8 c0 03 d0 aa
>1fb0 a0 00 b9 8f 3e 99 c3 0f
>1fb8 b9 91 3e 99 71 0d b9 93
>1fc0 3e 99 be 0d b9 95 3e 99
>1fc8 e6 0d b9 97 3e 99 17 0f
>1fd0 b9 9d 3e 99 f0 0e b9 9b
>1fd8 3e 99 ca 0e b9 9d 3e 99
>1fe0 b2 0c b9 9f 3e 99 ad 0d
>1feb b9 a1 3e 99 fc 0d c8 c0
>1ff0 02 d0 bf a9 5d 0d 01 0d
>1ff8 0d 29 0d 8d 51 0d a9 3b
>2000 8d 4f 0f a9 66 8d ff 0d
>2008 a9 2f 8d fe 0d a9 6d 8d
>2010 2b 0d a9 6e 8d 53 0d a9
>2018 db 8d 55 0e a9 da 8d 7c
>2020 0e a9 dc 8d 7d 0e a9 e1
>2028 8d 7e 0e a9 df 8d a5 0e
>2030 a9 e0 8d cd 0e a0 00 b9
>2038 a3 3e 99 7e 0d b9 a6 3e
>2040 99 a6 0d b9 a9 3e 99 ce
>2048 0d b9 ac 3e 99 f6 0d c8
>2050 c0 03 d0 e3 a9 9e 8d 2d
>2058 0f 60 a0 00 a9 1e 99 00
>2060 0c 99 28 0c 99 50 0c 99
>2068 78 0c 99 a0 0c c8 c0 0d
>2070 d0 0c 99 a0 00 b9 af 3e
>2078 99 55 0c c8 c0 03 d0 f5
>2080 60 a0 00 b9 af 3e 99 29
>2088 0c 99 79 0c 99 55 0c 99
>2090 31 0c 99 81 0c c8 c0 03
>2098 d0 e9 60 a0 00 b9 af 3e
>20a0 99 29 0c 99 79 0c 99 31
>20a8 0c 99 81 0c c8 c0 03 d0
>20b0 ec 60 a0 00 b9 b2 3e 99
>20b8 7e 0d b9 b5 3e 99 a6 0d
>20c0 b9 b8 3e 99 ce 0d b9 bb
>20c8 3e 99 f6 0d c8 c0 03 d0
>20d0 e3 60 a0 00 b9 be 3e 99
>20d8 a2 0d b9 c4 3e 99 ca 0d
>20e0 c8 c0 06 d0 ef 0d a0 00
>20e8 b9 ca 3e 99 cf 0d b9 d0
>20f0 3e 99 f7 0d b9 d6 3e 99

```

```

>20f8 1f 0e c8 c0 06 d0 e9 60
>2100 a0 00 b9 dc 3e 99 a8 0d
>2108 b9 e1 3e 99 d0 0d c8 c0
>2110 05 d0 ef a9 70 8d 83 0d
>2118 a9 be 8d 84 0d 60 a0 00
>2120 b9 e6 3e 99 02 0d b9 ea
>2128 3e 99 2a 0d b9 ee 3e 99
>2130 52 0d b9 f2 3e 99 7a 0d
>2138 c8 c0 04 d0 e3 a0 00 b9
>2140 f6 3e 99 a2 0d c8 c0 05
>2148 d0 f5 60 a9 dd 8d 55 0e
>2150 a9 20 8d 7c 0e a9 de 8d
>2158 7d 0e 60 a4 d0 a9 a6 99
>2160 dc 0e a9 f8 99 03 0f a9
>2168 a7 99 04 0f a9 f9 99 2b
>2170 0f a9 a8 99 2c 0f a9 a9
>2178 99 54 0f a9 20 99 7c 0f
>2180 a9 20 99 db 0e 60 a4 d0
>2188 a9 70 99 db 0e a9 a1 99
>2190 dc 0e a9 a2 99 03 0f a9
>2198 a3 99 04 0f a9 20 99 2b
>21a0 0f a9 a4 99 2c 0f a9 a5
>21a8 99 54 0f a9 20 99 7c 0f
>21b0 60 a9 20 8d db 0e a9 aa
>21b8 8d dc 0e a9 20 8d 03 0f
>21c0 a9 ab 8d 04 0f a9 20 8d
>21c8 2b 0f a9 ac 8d 2c 0f a9
>21d0 ad 8d 54 0f a9 20 8d 7c
>21d8 0f 60 a9 ae 8d dc 0e a9
>21e0 fa 8d 03 0f a9 b0 8d 04
>21e8 0f a9 b1 8d 05 0f a9 b2
>21f0 8d 2c 0f a9 b3 8d 2d 0f
>21f8 a9 b4 8d 54 0f a9 b5 8d
>2200 55 0f 60 a0 00 b9 fb 3e
>2208 99 0f 0c 99 8f 0c 99 0f
>2210 0d b9 04 3f 99 99 0c 99
>2218 b9 0c b9 0d 3f 99 64 0c
>2220 99 e4 0c c8 c0 09 d0 dd
>2228 a0 00 b9 04 3f 99 99 0d
>2230 c8 c0 07 d0 f5 a0 00 b9
>2238 0d 3f 99 64 0d c8 c0 04
>2240 d0 f5 a9 8b 8d 8f 0d 60
>2248 a9 21 8d 00 0c a0 00 a9
>2250 22 99 01 0c a9 32 99 a1
>2258 0c c8 c0 0b d0 f1 a9 1f
>2260 8d a0 0c a9 24 8d 28 0c
>2268 8d 50 0c 8d 78 0c a9 23
>2270 8d 0c 0c a9 25 8d 34 0c
>2278 8d 5c 0c 8d 84 0c a9 21
>2280 8d a0 0c 60 a0 00 a9 27
>2288 99 01 0c a9 2b 99 a1 0c
>2290 c8 c0 0b d0 f1 a9 26 8d
>2298 0c 0c a9 28 8d 0c 0c 8d
>22a0 a0 0c a9 2c 8d ac 0c a9
>22a8 29 8d 28 0c 8d 50 0c 8d
>22b0 78 0c a9 2a 8d 34 0c 8d
>22b8 5c 0c 8d 84 0c 60 a0 00
>22c0 b9 16 3f 99 29 0c c8 c0
>22c8 0b d0 f5 60 a0 00 b9 21
>22d0 3f 99 7b 0c c8 c0 07 d0
>22d8 f5 60 a0 00 b9 28 3f 99
>22e0 7d 0c c8 c0 03 d0 f5 60
>22e8 a0 00 b9 2b 3f 99 7b 0c
>22f0 c8 c0 06 d0 f5 60 a0 00
>22f8 b9 31 3f 99 7b 0c c8 c0
>2300 07 d0 f5 60 a0 00 b9 38
>2308 3f 99 2b 0c c8 c0 03 d0
>2310 f5 60 a0 00 b9 28 3f 99
>2318 2b 0c c8 c0 03 d0 f5 60
>2320 a0 00 b9 2b 3f 99 2b 0c
>2328 c8 c0 03 d0 f5 60 a0 00
>2330 b9 31 3f 99 2b 0c c8 c0
>2338 03 d0 f5 60 a0 00 b9 38

```

```

>2340 3f 99 2f 0c c8 c0 03 d0
>2348 f5 60 a0 00 b9 28 3f 99
>2350 2f 0c c8 c0 03 d0 f5 60
>2358 a0 00 b9 2b 3f 99 2f 0c
>2360 c8 c0 03 d0 f5 60 a0 00
>2368 b9 31 3f 99 2f 0c c8 c0
>2370 03 d0 f5 60 78 a9 0e 8d
>2378 14 03 a9 ce 8d 15 03 a9
>2380 00 8d 11 ff 58 60 a0 20
>2388 a2 ff ca d0 fd 88 d0 fa
>2390 60 a9 af 8d 2d 0f a9 9e
>2398 8d 06 0f 20 86 23 a9 20
>23a0 8d 06 0f a9 9f 8d 07 0f
>23a8 20 86 23 a9 20 8d 07 0f
>23b0 a9 9f 8d e0 0e 20 86 23
>23b8 a9 20 8d e0 0e a9 9e 8d
>23c0 b9 0e 20 86 23 a9 20 8d
>23c8 b9 0e a9 9f 8d ba 0e 20
>23d0 86 23 a9 20 8d ba 0e a9
>23d8 9f 8d 93 0e 20 86 23 a9
>23e0 2d 8d 93 0e a9 a0 8d 6c
>23e8 0e 20 86 23 a9 20 8d 6c
>23f0 0e a9 9e 8d 45 0e 20 86
>23f8 23 a9 20 8d 45 0e a9 9f
>2400 8d 46 0e 20 86 23 a9 20
>2408 8d 46 0e a9 9e 8d 1f 0e
>2410 20 86 23 a9 20 8d 1f 0e
>2418 a9 9d 8d 20 0e 20 86 23
>2420 a9 20 8d 20 0e ad f9 0d
>2428 c9 d1 0f 34 a9 9d 8d f9
>2430 0d 20 86 23 a9 d1 8d f9
>2438 0d a9 a0 8d d2 0d 20 86
>2440 23 a9 34 8d d2 0d a9 9e
>2448 8d ab 0d 20 86 23 a9 20
>2450 8d ab 0d a9 fc 8d 84 0d
>2458 20 86 23 a9 ff 85 d1 60
>2460 a9 9d 8d f9 0d 20 86 23
>2468 a9 20 8d f9 0d ad d2 0d
>2470 c9 b8 0d 2c a9 a0 8d d2
>2478 0d 20 86 23 a9 b8 8d d2
>2480 0d a9 9d 8d ab 0d 20 86
>2488 23 a9 bc 8d ab 0d 20 86
>2490 23 a9 bf 8d 84 0d a9 c0
>2498 8d ac 0d a9 80 85 d1 60
>24a0 a9 9d 8d f9 0d 20 86 23
>24a8 a9 20 4c 36 24 a9 af 8d
>24b0 2d 0f a9 9f 8d 2e 0f 20
>24b8 86 23 a9 20 8d 2e 0f a9
>24c0 9e 8d 07 0f 20 86 23 a9
>24c8 20 8d 07 0f a9 9f 8d 08
>24d0 0f 20 86 23 a9 20 8d 08
>24d8 0f a9 9e 8d e1 0e 20 86
>24e0 23 a9 20 8d e1 0e a9 9f
>24e8 8d e2 0e 20 86 23 a9 20
>24f0 8d e2 0e a9 9e 8d bb 0e
>24f8 20 86 23 a9 36 8d bb 0e
>2500 a9 9f 8d bc 0e 20 86 23
>2508 a9 2e 8d bc 0e a9 9e 8d
>2510 95 0e 20 86 23 a9 20 8d
>2518 95 0e a9 9f 8d 96 0e 20
>2520 86 23 a9 20 8d 96 0e a9
>2528 9e 8d 6f 0e 20 86 23 a9
>2530 20 8d 6f 0e a9 9f 8d 70
>2538 0e 20 86 23 a9 20 8d 70
>2540 0e a9 9e 8d 49 0e 20 86
>2548 23 a9 20 8d 49 0e a9 9f
>2550 8d 4a 0e 20 86 23 a9 20
>2558 8d 4a 0e ad 23 0e c9 3a
>2560 d0 31 a9 9e 8d 23 0e 20
>2568 86 23 a9 3a 8d 23 0e a9
>2570 9f 8d 24 0e 20 86 23 a9
>2578 2d 8d 24 0e a9 9e 8d fd
>2580 0d 20 86 23 a9 63 8d fd

```

```

>2588 0d a9 fd 8d fe 0d a9 ff
>2590 85 d1 60 a9 9e 8d 23 0e
>2598 20 86 23 a9 d7 8d 23 0e
>25a0 a9 d9 8d 24 0e a9 d5 8d
>25a8 fc 0d a9 00 85 d1 60 a9
>25b0 af 8d 2d 0f a9 a0 8d 06
>25b8 0f 20 86 23 a9 20 8d 06
>25c0 0f a9 a0 8d de 0e 20 86
>25c8 23 a9 20 8d de 0e a9 a0
>25d0 8d b6 0e 20 86 23 a9 20
>25d8 8d b6 0e 20 86 23 a9 20
>25e0 9e 8d 0e 0e 20 86 23 a9
>25e8 20 8d 8e 0e a9 9d 8d 67
>25f0 0e 20 86 23 a9 3a 8d 67
>25f8 0e a9 9f 8d 3f 0e 20 86
>2600 23 a9 20 8d 3f 0e a9 9d
>2608 8d 18 0e 20 86 23 a9 20
>2610 8d 18 0e a9 9f 8d f0 0d
>2618 20 86 23 a9 20 8d f0 0d
>2620 a9 9d 8d c9 0d 20 86 23
>2628 a9 20 8d c9 0d ad 2d 0d
>2630 c9 43 d0 17 a9 9f 8d a2
>2638 0d 20 86 23 a9 43 8d a2
>2640 0d a9 fe 8d 7b 0d a9 ff
>2648 85 d1 60 a9 c6 8d a2 0d
>2650 a9 00 85 d1 60 20 af 25
>2658 a9 9e 8d 8e 0e 20 86 23
>2660 a9 20 8d 8e 0e a9 9f 8d
>2668 66 0e 20 86 23 a9 2e 8d
>2670 66 0e a9 a0 8d 17 0e 20
>2678 86 23 a9 20 8d 17 0e a9
>2680 a0 8d ef 0d 20 86 23 a9
>2688 20 8d ef 0d a9 9e 8d c7
>2690 0d 20 86 23 a9 20 8d c7
>2698 0d a9 9d 8d a0 8d 20 86
>26a0 23 a9 30 8d a0 8d a9 9f
>26a8 8d 78 0d 20 86 23 a9 20
>26b0 8d 78 0d a9 9d 8d 51 0d
>26b8 20 86 23 a9 5d 8d 51 0d
>26c0 a9 9f 8d 29 0d 20 86 23
>26c8 a9 5d 8d 29 0d ad 02 0d
>26d0 c9 20 d0 17 a9 9f 8d 02
>26d8 0d 20 86 23 a9 20 8d 02
>26e0 0d a9 ff 8d db 0c a9 ff
>26e8 85 d1 60 a9 9f 8d 02 0d
>26f0 20 86 23 a9 e7 8d 02 0d
>26f8 a9 00 85 d1 60 2c 31 33
>2700 31 2c 31 33 31 2c 31 33
>2708 31 2c 31 35 32 2c 31 39
>2710 32 2c 31 39 32 2c 31 35
>2718 32 00 30 27 33 01 83 31
>2720 33 31 2c 38 34 2c 31 33
>2728 31 2c 31 35 32 2c 31 35
>2730 32 2c 31 35 32 2c 31 33
>2738 31 2c 38 34 00 59 27 34
>2740 01 83 33 30 2c 34 2c 31
>2748 39 37 2c 34 2c 33 30 2c
>2750 36 2c 38 39 2c 31 32 37
>2758 00 79 27 35 01 83 31 32
>2760 37 2c 33 30 2c 33 30 2c
>2768 33 30 2c 34 2c 31 39 37
>2770 2c 31 39 37 2c 31 39 37
>2778 00 98 27 36 01 83 38 34
>2780 2c 31 35 32 2c 31 33 31
>2788 2c 38 34 2c 33 30 2c 33
>2790 30 2c 34 2c 31 32 37 00
>2798 ba 27 37 01 83 31 32 37
>27a0 2c 33 30 2c 33 30 2c a9
>27a8 00 8d 12 ff a9 38 8d 13
>27b0 ff a9 98 8d 07 ff a9 00
>27b8 8d 15 ff 8d 19 ff a9 42
>27c0 8d 16 ff a9 36 8d 17 ff
>27c8 a9 69 a0 00 99 78 08 c8

```

```

>27d0 c0 78 d0 f8 a9 61 a0 00
>27d8 99 18 09 c8 c0 28 d0 f8
>27e0 a9 43 a0 00 99 90 09 c8
>27e8 c0 28 d0 f8 a9 45 a0 00
>27f0 99 30 0a c8 c0 50 d0 f8
>27f8 a9 44 a0 00 99 a0 0a 99
>2800 70 0b c8 c0 50 d0 f5 a9
>2808 79 a0 00 99 f8 0a c8 c0
>2810 78 d0 f8 a9 6f a0 00 99
>2818 05 0b 99 2d 0b 99 55 0b
>2820 c8 c0 06 d0 f2 a0 00 b9
>2828 f0 3c 99 81 0c b9 07 3d
>2830 99 a9 0c b9 1e 3d 99 d1
>2838 0c c8 c0 17 d0 e9 a0 00
>2840 b9 35 3d 99 28 0d c8 c0
>2848 09 d0 f5 a0 00 b9 3e 3d
>2850 99 97 0d c8 c0 1a d0 f5
>2858 a0 00 b9 58 3d 99 fd 0e
>2860 b9 76 3d 99 25 0f b9 94
>2868 3d 99 4d 0f c8 c0 1e d0
>2870 e9 60 a0 00 b9 e2 3d 99
>2878 3a 0e b9 ea 3d 99 62 0e
>2880 b9 c2 3d 99 46 0e b9 ca
>2888 3d 99 6e 0e c8 c0 08 d0
>2890 e3 60 a0 00 b9 b2 3d 99
>2898 3a 0e b9 ba 3d 99 62 0e
>28a0 b9 d2 3d 99 46 0e b9 da
>28a8 3d 99 6e 0e c8 c0 08 d0
>28b0 e3 60 ee 5e 06 ae 5e 06
>28b8 e0 2b f0 03 4c 0e ce a9
>28c0 04 8d 11 ff a0 05 a2 ff
>28c8 ca d0 fd 8d d0 fa a9 32
>28d0 8d 11 ff ac 5f 06 b9 2d
>28d8 29 8d 10 ff 18 09 08 8d
>28e0 12 ff b9 6b 29 8d 0e ff
>28e8 38 e9 04 8d 0f ff b9 b6
>28f0 29 8d b9 28 c8 c0 4b d0
>28f8 02 a0 00 8c 5f 06 a2 00
>2900 8e 5e 06 4c 0e ce 78 a9
>2908 b2 8d 14 03 a9 28 8d 15
>2910 03 a9 00 8d 5e 06 8d 5f
>2918 06 a9 05 8d b9 28 58 60
>2920 01 01 01 02 02 02 02 01
>2928 01 01 01 02 02 02 02 02
>2930 02 01 02 02 02 02 02 02
>2940 01 01 02 01 03 01 02 01
>2948 01 02 01 01 01 00 01 02
>2950 01 02 02 01 02 02 02 03
>2958 02 02 02 02 02 02 02 02
>2960 01 02 02 02 02 02 02 02
>2968 02 01 03 c5 c5 c5 54 54
>2970 54 04 7f a9 c5 c5 c5 54
>2978 54 54 83 3b c5 1e 1e 1e
>2980 98 98 98 83 1e 7f 2f 2f
>2988 2f e4 e4 e4 c5 98 c5 fe
>2990 59 04 c5 59 04 c5 59 59
>2998 a9 59 04 c5 54 04 c5 3b
>29a0 3b 04 fe 04 83 54 04 83
>29a8 54 04 04 7f 04 83 54 c0
>29b0 83 54 3b 3b c5 fe 15 15
>29b8 15 15 15 15 2b 2b 2b 15
>29c0 15 15 15 15 15 2b 2b 2b
>29c8 15 15 15 15 15 2b 2b 2b
>29d0 2b 15 15 15 15 15 2b 2b
>29d8 2b 6d 15 15 15 15 15 15
>29e0 15 2b 15 41 15 15 15 15
>29e8 15 15 2b 15 41 15 15 15
>29f0 15 15 15 15 2b 15 41 15
>29f8 15 15 15 15 15 2b 15 41
>2a00 2b ee 5e 06 ae 5e 06 e0
>2a08 61 0f 0e 04 0e ce a9 04
>2a10 8d 11 ff a0 05 a2 ff ca

```

```

>2a18 d0 fd 88 d0 fa a9 32 8d
>2a20 11 ff ac 5f 06 b9 6f 2a
>2a28 8d 10 ff 18 69 00 8d 12
>2a30 ff b9 e4 2a 8d 0e ff 38
>2a38 e9 04 8d 0f ff b9 59 2b
>2a40 8d 08 2a c8 c0 75 d0 02
>2a48 a0 00 8c 5f 06 a2 00 8e
>2a50 5e 06 4c 0e ce 78 a9 01
>2a58 8d 14 03 a9 2a 8d 15 03
>2a60 a9 00 8d 5e 06 8d 5f 06
>2a68 a9 05 8d 08 2a 58 60 01
>2a70 01 01 01 01 02 02 02 02
>2a78 01 01 02 01 01 02 02 01
>2a80 01 03 01 02 02 02 02 02
>2a88 02 02 02 02 02 02 02 02
>2a90 02 02 02 02 03 02 02 02
>2a98 02 01 02 02 01 01 03 01
>2aa0 01 01 02 02 02 01 01 01
>2aa8 01 01 01 01 01 01 01 02
>2ab0 03 02 02 02 02 02 02 02
>2ab8 02 03 02 02 01 01 01 02
>2ac0 02 01 01 01 01 03 02 02
>2ac8 02 02 01 02 03 02 02 02
>2ad0 02 01 02 03 01 01 02 02
>2ad8 03 02 02 02 02 02 02 02
>2ae0 02 02 01 03 06 06 06 c5
>2ae8 c5 04 04 83 3b c5 c5 3b
>2af0 c5 59 54 04 a2 c5 fe c5
>2af8 04 3b 3b 3b 54 3b 3b 04
>2b00 04 04 3b 54 54 54 83 54
>2b08 3b fe 83 83 83 3b c5 83
>2b10 3b c5 06 fe 06 06 a2 04
>2b18 54 04 a2 c5 c5 7f 59 c5
>2b20 c5 c5 a2 c5 04 fe 04 1e
>2b28 3b 04 3b 54 83 04 fe 1e
>2b30 04 c5 c5 c5 1e 04 c5 c5
>2b38 a2 a2 fe 83 83 83 3b c5
>2b40 04 fe 83 83 83 3b c5 04
>2b48 fe 06 c5 04 3b fe 54 83
>2b50 ad 04 ad 83 3b 54 04 c5
>2b58 fe 08 22 08 2f 2f 2f 2f
>2b60 2f 2f 22 08 22 08 2f 61
>2b68 22 08 7a 15 22 08 2f 2f
>2b70 2f 22 08 15 15 61 22 08
>2b78 2f 2f 2f 22 08 61 15 22
>2b80 08 2f 22 08 2f 22 08 61
>2b88 22 08 15 15 61 2f 15 15
>2b90 2f 2f 61 2f 15 15 2f 15
>2b98 15 61 2f 2f 48 15 15 15
>2ba0 15 15 15 15 15 15 48 15
>2ba8 15 15 15 15 15 15 2f 55
>2bb0 08 87 08 22 08 61 55 08
>2bb8 87 08 22 08 61 2f 2f 94
>2bc0 2f 94 2f 61 2f 2f 94 2f
>2bc8 87 08 22 08 87 c7 ee 5e
>2bd0 06 ae 5e 06 e0 81 f0 03
>2bd8 4c 0e ce a9 04 8d 11 ff
>2be0 a0 05 a2 ff oa d0 fd 08
>2be8 d0 fa a9 32 8d 11 ff a0
>2bf0 5f 06 b9 3c 2c 8d 10 ff
>2bf8 18 69 00 8d 12 ff b9 9b
>2c00 2c 8d 0e ff 38 e9 04 8d
>2c08 0f ff b9 fa 2c 8d d5 2b
>2c10 c8 c0 5f d0 02 a0 00 8c
>2c18 5f 06 a2 d0 08 5e 06 4c
>2c20 0e ce 78 a9 ce 8d 14 03
>2c28 a9 2b 8d 15 03 a9 00 
```



```

>2c60 01 01 00 01 01 01 01 00
>2c68 02 01 01 01 02 02 02 02
>2c70 02 02 02 02 02 02 02 02
>2c78 01 01 01 01 01 02 02 01
>2c80 01 01 01 01 01 02 02 02
>2c88 02 02 02 02 02 01 01 01
>2c90 01 01 02 02 01 01 01 01
>2c98 01 01 03 2f 7f c5 7f e4
>2ca0 c5 7f 06 2f 54 1e e4 c5
>2ca8 7f c5 2f 1e 2f 7f c5 7f
>2cb0 e4 c5 7f 06 2f 54 1e e4
>2cb8 c5 7f c5 2f 1e 7f c5 7f
>2cc0 06 3e e4 c5 7f 06 3e 1e
>2cc8 e4 c5 c5 04 04 1e 1e 98
>2cd0 83 83 54 1e 54 1e 1e e4
>2cd8 2f c5 c5 e4 1e 54 e4 7f
>2ce0 c5 c5 7f 2f 98 83 83 54
>2ce8 1e 54 1e 1e e4 c5 7f c5
>2cf0 e4 1e 54 e4 7f 2f c5 7f
>2cf8 2f fe 60 1e 3f 3f 3f 3f
>2d00 1e 1e 3f 3f 3f 3f 3f 3f
>2d08 1e 1e 81 60 1e 3f 3f 3f
>2d10 3f 1e 1e 3f 3f 3f 3f 3f
>2d18 3f 1e 1e 81 3f 3f 3f 1e
>2d20 3f 3f 3f 1e 1e 3f 3f 3f
>2d28 60 1e 3f 1e 1e 81 60 1e
>2d30 1e 1e 3f 60 1e 1e 1e 3f
>2d38 60 0e 0e 1e 1e 1e 1e 3f
>2d40 1e 1e 81 60 1e 1e 1e 3f
>2d48 60 1e 1e 1e 3f 60 0e 0e
>2d50 1e 1e 1e 1e 3f 1e 1e 81
>2d58 d9 ee 5e 06 ae 5e 06 e0
>2d60 67 f0 03 4c 0e ce a9 04
>2d68 8d 11 ff a0 05 a2 ff ca
>2d70 d0 fd 88 d0 fa a9 32 8d
>2d78 11 ff a0 5f 06 b9 c7 2d
>2d80 8d 10 ff 18 69 00 8d 12
>2d88 ff b9 2f 2e 8d 0e ff 38
>2d90 e9 84 8d 0f ff b9 97 2e
>2d98 8d 60 2d c8 c0 68 d0 02
>2da0 a0 00 8c 5f 06 a2 00 8e
>2da8 5e 06 4c 0e ce 78 a9 59
>2db0 8d 14 03 a9 2d 8d 15 03
>2db8 a9 00 8d 5e 06 8d 5f 06
>2dc0 a9 05 8d 60 2d 58 60 01
>2dc8 01 00 01 01 02 02 02 02
>2dd0 01 01 01 01 01 02 02 02
>2dd8 02 01 02 02 02 01 01 00
>2de0 03 01 01 00 01 01 02 02
>2de8 02 02 01 01 01 01 01 02
>2df0 02 02 02 01 02 02 02 01
>2df8 01 00 02 02 02 02 02 02
>2e00 02 02 02 02 02 02 02 02
>2e08 02 02 02 01 02 02 01 01
>2e10 01 01 02 02 02 02 01 01
>2e18 01 02 02 02 02 02 02 02
>2e20 01 01 02 02 02 02 02 02
>2e28 02 02 02 02 02 02 03 7f
>2e30 06 3e 06 7f 1e 83 54 1f
>2e38 06 59 7f 7f 7f 83 54 1f
>2e40 04 c5 04 1f 1f 7f 06 3e
>2e48 fe 7f 06 3e 06 7f 1e 83
>2e50 54 1f 06 59 7f 7f 7f 83
>2e58 54 1f 04 c5 04 1f 1f 7f
>2e60 06 3e 83 83 83 98 c0 c0
>2e68 98 83 54 83 98 98 98 83
>2e70 54 1e 04 c5 04 1e 06 59
>2e78 7f 7f 1e 1e 1e 04 c5 c5
>2e80 c5 54 98 83 54 1e 1e 04
>2e88 7f 7f 1e 1e 54 83 98 c0
>2e90 83 1e 83 98 54 1e fe 18
>2e98 18 32 32 32 67 18 18 32
>2ea0 32 32 67 18 18 4c 18 32

```

```

>2ea8 67 18 18 32 32 32 32 32
>2eb0 18 18 18 32 32 32 67 18
>2eb8 18 32 32 32 67 18 18 4c
>2ec0 18 32 67 18 18 32 32 32
>2ec8 32 32 18 18 32 32 32 67
>2ed0 18 18 32 32 32 67 32 67
>2ed8 18 18 67 18 18 32 32 32
>2ee0 67 32 32 32 18 18 32 32
>2ee8 32 32 18 18 18 18 32 32
>2ef0 18 18 32 18 18 18 32 32
>2ef8 32 32 4c 18 32 67 d2 17
>2f00 ee 5e 06 ae 5e 06 e0 64
>2f08 f0 03 4c 0e ce a9 04 8d
>2f10 11 ff a0 05 a2 ff ca d0
>2f18 fd 88 d0 fa a9 32 8d 11
>2f20 ff a0 5f 06 b9 6e 2f 8d
>2f28 10 ff 18 69 00 8d 12 ff
>2f30 b9 9a 2f 8d 0e ff 38 e9
>2f38 84 8d 0f ff b9 c6 2f 8d
>2f40 07 2f c8 c0 2c d0 02 a0
>2f48 00 8c 5f 06 a2 00 8e 5e
>2f50 06 4c 0e ce 78 a9 00 8d
>2f58 14 03 2f 8d 15 03 a9
>2f60 00 8d 5e 06 8d 5f 06 a9
>2f68 05 8d 07 2f 58 60 01 01
>2f70 01 01 01 01 02 02 02 02
>2f80 02 02 02 02 02 02 02 02
>2f88 02 02 01 02 02 02 01 01
>2f90 02 02 02 02 02 02 02 01
>2f98 01 03 7f 7f c5 59 7f c5
>2fa0 04 04 1f 04 c5 7f c5 7f
>2fa8 59 7f fe 54 54 54 1f
>2fb0 04 1f 1f 1f 04 c5 04
>2fb8 1f 04 c5 7f 04 1f 54 83
>2fc0 54 1f 04 c5 7f fe 42 42
>2fc8 42 4f 42 42 42 42 42 64
>2fd0 1f 42 42 42 42 86 42 42
>2fd8 42 42 64 1f 42 42 42 42
>2fe0 64 1f 42 42 1f 1f 1f 1f
>2fe8 64 1f 42 1f 0e 0e 42 42
>2ff0 cc 07 0f ff ff 0f ff 0f
>2ff8 14 ff 08 01 ff 93 01 ff
>3000 c1 00 98 94 94 8c 00 c1
>3008 e3 e3 c3 83 e3 e3 80 80
>3010 c1 00 98 f1 c7 8f 80 80
>3018 c1 80 fc f1 f1 80 c0 c1
>3020 f9 f1 e1 99 80 80 f9 f9
>3028 80 80 9f 81 c0 fc 80 c1
>3030 c1 80 9f 81 80 9c 80 c1
>3038 80 80 98 f8 f1 f1 e3 e3
>3040 c1 80 9c c1 c1 80 80 c1
>3048 c1 80 9c 80 c0 fc 80 c1
>3050 e3 c1 9c 80 80 9c 9c 9c
>3058 c1 80 9c 9f 9f 9c 80 c1
>3060 81 80 9c 9c 9c 9c 80 81
>3068 80 80 9f 83 83 9f 80 80
>3070 80 80 9f 83 83 9f 9f 9f
>3078 c1 80 9f 98 98 9c 80 c1
>3080 9c 9c 9c 80 80 9c 9c 9c
>3088 c1 c1 e3 e3 e3 e3 c1 c1
>3090 9f 9f 9f 9f 9f 9f 80 80
>3098 9c 80 80 80 94 9c 9c 9c
>30a0 9c 8c 84 80 98 9c 9c 9c
>30a8 c1 80 9c 9c 9c 9c 80 c1
>30b0 81 80 9c 80 81 9f 9f 9f
>30b8 81 80 9c 80 81 93 99 9c
>30c0 c1 80 8f c3 e1 f8 80 c1
>30c8 80 80 e3 e3 e3 e3 e3 e3
>30d0 9c 9c 9c 9c 9c 9c 80 c1
>30d8 9c 9c 9c 80 c1 e3 e3 e3
>30e0 ff ff e7 e7 e7 e7 ff ff
>30e8 f1 e3 c7 ff ff ff ff ff

```

```

>30f0 ff ff ff ff ff ff ff ff
>30f8 cf cf ff ff co co ff ff
>3100 00 00 00 00 00 00 00 00
>3108 ff ff f3 f3 cf cf ff ff
>3110 ff ff 33 33 ff ff ff ff
>3118 ff ff 33 33 ff ff f3 f3
>3120 cf cf ff ff cf cf ff ff
>3128 ff ff f3 f3 ff ff f3 f3
>3130 ff ff co co ff ff cf cf
>3138 ff ff co co ff ff ff ff
>3140 ff ff cf cf f3 f3 ff ff
>3148 ff ff cf cf ff ff cf cf
>3150 f3 f3 ff ff f3 f3 ff ff
>3158 ff ff ff ff 33 33 ff ff
>3160 f3 f3 ff ff 33 33 ff ff
>3168 00 00 00 00 00 00 00 c0
>3170 00 00 00 00 00 00 00 00
>3178 00 00 00 00 00 c0 3c 03
>3180 c0 3c 03 00 00 00 00 00
>3188 00 c0 3c 03 00 00 00 00
>3190 ff ff ff ff co co ff ff
>3198 00 00 00 00 00 00 c0 3c
>31a0 00 00 00 c0 3c 03 00 00
>31a8 00 00 00 00 c0 3c 03 00
>31b0 3c 03 00 00 00 00 00 00
>31b8 03 00 00 00 00 00 00 00
>31c0 f0 0f 00 00 00 00 00 00
>31c8 00 00 00 f0 0f 00 00 00
>31d0 00 00 00 00 00 f0 0f 00
>31d8 00 00 00 00 00 00 f0 0f
>31e0 00 00 00 00 f0 0f 00 00
>31e8 00 f0 0f 00 00 00 00 00
>31f0 00 00 00 00 00 f0 0f 0f
>31f8 00 03 0f 00 30 30 c0 c0
>3200 f0 0f 00 00 03 03 c0 3c
>3208 00 00 c0 fc 03 00 00 00
>3210 00 c0 co 3f 0f 03 00 00
>3218 3c 00 c0 00 00 c0 3c 03
>3220 0f 00 00 00 00 00 00 00
>3228 0c cf 3f 03 00 00 00 00
>3230 c0 c0 00 00 c0 3c 03 00
>3238 00 c0 3c 03 03 0c 0c 30
>3240 30 30 c0 c0 3c 03 00 00
>3248 00 00 00 03 03 fc 0c 0c
>3250 00 00 00 c0 30 0f 0c 0c
>3258 03 03 03 00 00 00 00 00
>3260 03 00 00 c0 c0 c0 30 30
>3268 00 00 00 00 c0 c0 c0 30
>3270 00 00 00 00 00 03 03 0c
>3278 03 0c 30 c0 30 c0 00 00
>3280 00 03 0f 0c 30 f0 c0 00
>3288 c0 00 00 00 00 00 00 00
>3290 00 00 03 03 0c 0c 30 f0
>3298 30 c0 c0 00 00 00 00 00
>32a0 30 f0 c0 00 00 00 00 00
>32a8 00 00 00 00 00 03 0f 0c
>32b0 0c 0c 30 f0 c0 00 00 00
>32b8 00 00 00 00 c0 00 03 03
>32c0 03 03 0c 3c 30 c0 c0 00
>32c8 00 00 03 0f 0c 30 30 c0
>32d0 30 0c 0c 0f 03 03 00 00
>32d8 00 00 00 00 00 00 c0 30
>32e0 f0 c0 00 00 00 00 00 00
>32e8 0c 0c 0c 0c 0c 0c 0c 0c
>32f0 00 00 03 03 0f 0f 0c 0c
>32f8 33 c3 03 03 03 c3 3f 03
>3300 f0 cf c0 c0 c0 c0 c0 c0
>3308 00 00 00 00 00 00 00 c0
>3310 00 03 f3 0c 0c 0c 0c 0c
>3318 0c 0c 0c f0 0f 0c 0c 0c
>3320 00 00 00 00 00 03 0c 30
>3328 f0 30 f0 c0 00 00 00 00
>3330 c0 c0 c0 30 30 30 30 30

```

```

>3338 00 00 c0 c0 c0 c0 c0 c0
>3340 03 03 00 00 00 00 00 00
>3348 03 03 03 03 03 03 03 03
>3350 cc 0c 0c 0c 0c 0c 0c 0c
>3358 00 00 00 00 f0 0c 3c 3c
>3360 00 00 00 00 00 03 0f 3f
>3368 c0 c0 30 30 30 30 30 30
>3370 30 30 30 0c 0c 0c 0c 0c
>3378 0c 0c 3f 30 c0 c0 00 00
>3380 00 00 00 00 00 00 00 03
>3388 00 00 00 00 01 05 06 02
>3390 00 00 00 00 00 40 40 40
>3398 2f 2f 8f 8f 57 55 7d 7f
>33a0 00 e0 e0 e0 b0 b0 70 56
>33a8 7f 7f 7f 57 15 01 00 00
>33b0 56 7d 7f 7f 7f 7f 57 15
>33b8 01 00 00 00 00 00 00 00
>33c0 00 04 15 19 09 dd 9f 37
>33c8 37 7d 56 56 7d 7f 7f 7f
>33d0 7f 57 15 01 00 00 00 00
>33d8 00 00 00 00 00 00 c0 c0
>33e0 80 80 c1 c2 48 58 55 7d
>33e8 7f 7f 7f 57 15 01 00 00
>33f0 00 00 00 00 00 00 10 54
>33f8 64 24 e0 fd fd fe fe 7b
>3400 5b 55 7d 7f 7f 7f 7f 57
>3408 15 01 00 00 00 00 00 00
>3410 00 00 00 01 05 06 02 2f
>3418 2f 8f 8f 57 55 7d 7f 7f
>3420 7f 7f 57 15 01 00 00 00
>3428 00 00 00 40 40 40 40 40
>3430 e0 e0 e3 b2 ba 58 55
>3438 7d 7f 7f 7f 7f 57 15 01
>3440 00 00 00 00 00 00 10 54
>3448 54 64 24 e0 fd ff fe fe
>3450 7b 5b 55 7d 7f 7f 7f 7f
>3458 57 15 01 00 00 00 00 00
>3460 00 00 00 00 81 85 26 22
>3468 2f 0f 0f 4f 57 55 7d 7f
>3470 00 00 00 00 00 40 40 48
>3478 0a e2 e0 f0 f0 f0 70 54
>3480 55 7d 7f 7f 7f 7f 57 15
>3488 00 04 95 99 8d 1d 1f 37
>3490 37 7d 55 55 7d 7f 7f 7f
>3498 00 00 00 00 00 20 e0 c0
>34a0 c0 c0 c1 c2 48 58 55 7d
>34a8 64 24 e0 fd fd ff ff 7f
>34b0 57 55 7d 7f 7f 7f 7f 57
>34b8 20 20 88 89 05 06 06 0f
>34c0 0f 0f 4f 57 55 7d 7f 7f
>34c8 00 00 00 00 40 40 40 48
>34d0 02 e2 e2 e3 b0 b0 54 55
>34d8 54 64 24 e0 fc ff ff fe
>34e0 7e 56 55 7d 7f 7f 7f 7f
>34e8 30 fc fc fc 30 00 00 00
>34f0 00 00 00 0c 3f 3f 3f 0c
>34f8 0c 3f 3f 3f 0c 00 00 00
>3500 00 00 00 30 fc fc fc 30
>3508 00 10 54 54 54 20 fc ff
>3510 03 02 02 02 00 00 00 00
>3518 ff fe fe fe 7e 56 54 54
>3520 94 88 88 88 88 88 88 48
>3528 08 08 04 00 00 00 00 00
>3530 00 00 00 00 00 10 54 54
>3538 54 20 fc ff ff fe fe fe
>3540 f4 54 54 54 58 88 88 88
>3548 88 88 88 84 80 80 80 00
>3550 00 00 00 00 00 10 54 54
>3558 54 24 fc ff ff fe fe fe
>3560 fd 5c 54 54 54 84 80 80
>3568 88 88 88 89 80 90 80 00
>3570 00 00 04 15 15 14 18 fc
>3578 00 00 00 00 00 00 00 00

```

```

>3580 ff bf 3f 3f 3f 3f 3d 3d
>3588 c0 c0 90 a0 20 00 00 00
>3590 35 15 15 16 12 20 20 20
>3598 00 00 00 0c bf bf bf 8c
>35a0 20 20 20 20 20 04 00 00
>35a8 10 00 00 00 00 00 00 00
>35b0 02 2a 6a 00 00 0a 1a 00
>35b8 a5 a5 09 29 a0 a0 00 00
>35c0 5f 5c 50 40 3c 03 00 00
>35c8 c0 00 00 00 00 00 f0 0f
>35d0 00 00 00 00 00 01 05 a5
>35d8 00 00 03 0f 3f 7f 5f 5f
>35e0 3f ff fa ea e9 f3 ff fc
>35e8 00 42 4a 68 e0 c0 00 00
>35f0 3c 03 00 00 00 00 ea ea
>35f8 3c 03 00 00 00 00 ee ff
>3600 3f ff ce e8 e0 c0 00 00
>3608 3a ea e0 20 0a 82 3c 03
>3610 b0 bf 0f 5b 6a ab bf 3f
>3618 00 00 f0 ff ff ff ff ff
>3620 00 00 00 c0 f4 f5 f5
>3628 c0 3c 03 00 00 00 a0 a0
>3630 3e ff ff 3f 0e c2 3c 03
>3638 f3 0f 00 00 00 00 00 00
>3640 ff 0f c0 3c 03 00 00 00
>3648 d5 d5 d6 16 06 f0 0f 00
>3650 a8 2a 8a a0 aa 2a 00 c0
>3658 00 00 90 00 a0 40 00 00
>3660 00 00 00 6a 2a 02 00 1a
>3668 03 00 00 00 80 a0 a1 29
>3670 00 f0 0f 00 00 00 00 70
>3678 0a 00 c0 3c 03 00 00 00
>3680 89 a1 a9 2a 0a c2 3c 03
>3688 7c 7f 5f 5f 5f 5f 1f 03
>3690 00 c0 fc ff ff ff ff ff
>3698 00 00 00 c0 fc ff d6 a6
>36a0 c0 3c 03 00 00 00 80 a0
>36a8 c0 3c 03 00 00 00 00 b0
>36b0 ff 0f c0 3c 03 00 00 00
>36b8 e4 f0 fe 0e 00 f0 0f 00
>36c0 28 08 a8 a8 00 00 00 c0
>36c8 fc fc fc b8 00 00 00 c0
>36d0 02 0a 2a a0 00 00 00 00
>36d8 00 04 15 05 09 08 08 15
>36e0 55 55 55 15 15 15 15 16
>36e8 00 04 15 19 19 0c 19
>36f0 59 69 a5 15 15 15 15 16
>36f8 3e 17 15 15 11 21 e2 2e
>3700 22 22 22 12 01 00 00 00
>3708 40 40 80 80 80 80 80 80
>3710 00 00 00 28 2a 82 a0 20
>3718 f0 cf c0 c0 c0 80 80 b0
>3720 28 08 0b 03 03 03 00 00
>3728 f0 fc 5c 6b 6b eb eb ff
>3730 00 00 00 00 00 00 c0 c0
>3738 00 00 30 fc fe fe b0 20
>3740 ff 3f 3f 3f 0f 0f 0f 0f
>3748 c0 f0 f0 fc fc fc fd f5
>3750 00 00 00 00 00 00 40 40
>3758 d5 d5 15 c5 36 0a 0a 02
>3760 40 60 a0 a0 20 20 b0 8a
>3768 02 a2 20 20 00 01 00 00
>3770 00 bc 83 10 00 00 00 00
>3778 00 00 00 00 00 02 0a 0a
>3780 04 15 05 09 09 f8 ff 3f
>3788 c8 3f 0b 08 00 00 00 00
>3790 ff 3f 3f 3f 15 15 15 15
>3798 f0 f8 ca c2 c2 c2 c2 7e
>37a0 15 25 aa a2 a2 82 82 82
>37a8 43 40 40 80 80 00 00 00
>37b0 22 22 40 3c 01 00 00 00
>37b8 00 00 00 00 00 c0 3c 03
>37c0 00 00 00 03 03 02 02 02

```

```

>37c8 02 02 00 00 00 00 00 00
>37d0 00 02 02 02 02 0a 08 08
>37d8 81 80 98 81 81 98 80 81
>37e0 3c 0f 3f 3f 3f 0c 00 00
>37e8 00 0c 3f 3f 3f cc 3c 03
>37f0 0c 3f 3f 3f cc c0 00 00
>37f8 0f 3f 3f 3f 0c 00 00 00
>3800 0f 35 d5 ff ff ff ff fd
>3808 fd ff fd ff ff ff ff fd
>3810 fd ff fd ff ff ff ff 00
>3818 ff 55 57 fd fd ff fc c0
>3820 ff 57 5d f5 f7 fc c0 c0
>3828 ff 55 57 fd fd ff fc 00
>3830 f0 f0 73 73 c3 03 03 03
>3838 c3 c3 c3 c3 03 03 03 03
>3840 f3 73 73 c3 03 03 03 00
>3848 3f db 5f f7 f7 f7 f7 f7
>3850 f7 f7 f7 f7 f7 f7 f7 f7
>3858 f7 fd f5 ff ff ff ff 00
>3860 ff 57 5d f5 f7 fc f0 00
>3868 00 03 0d 0f 0f 0f 0f 0f
>3870 0f 0f 0f 0f 0f 0f 0f 0f
>3878 cf cf cf cf 0f 0f 0f 00
>3880 ff 55 55 ff ff ff ff dd
>3888 df df d5 ff ff ff ff dc
>3890 dc dc dc dc f0 c0 00 00
>3898 ff 5f 77 db dc f0 c0 00
>38a0 fc 7c dc 5c 70 c0 00 00
>38a8 03 0d 3c 3f 3f 3f 3f 3f
>38b0 3f 3f 3f 3f 3f 3f 3f 3f
>38b8 3f 3f 3f 3f 3f 3f 3f 3f
>38c0 f0 70 70 7f 7f ff ff ff
>38c8 ff 7f 7f 7f 7f 70 70 70
>38d0 70 70 70 70 70 c0 00 00
>38d8 03 0d 35 ff ff ff ff ff
>38e0 ff ff ff ff 3f 3f 3f 3f
>38e8 f0 70 f3 73 73 73 73 73
>38f0 73 73 73 73 73 73 73
>38f8 73 73 73 73 c3 03 00
>3900 00 00 00 00 00 00 00 00
>3908 3f d5 55 ff ff ff ff f7
>3910 00 00 00 00 00 00 00 00
>3918 f7 fd f5 ff ff ff ff f7
>3920 f7 fd f5 ff ff ff ff 00
>3928 ff 57 5d f5 f7 fc f0 00
>3930 ff 5f 77 d7 dc f0 c0 00
>3938 ff 57 5d f5 f7 fc f0 00
>3940 c0 c3 cd cf 0f 0f 0f 00
>3948 c0 c0 c0 c0 00 00 00 00
>3950 ff 55 55 ff ff ff ff fd
>3958 fd fd fd fd fd fd fd fd
>3960 fd fd fd fd fd ff fc 00
>3968 ff 5f 77 d7 dc f0 c0 c0
>3970 c0 c0 c0 c0 c0 c0 c0 c0
>3978 c0 c0 c0 c0 c0 00 00 00
>3980 ff 55 55 ff ff ff ff 70
>3988 7f d5 57 fd fd ff fc 70
>3990 7f d5 55 ff ff ff ff 00
>3998 fc 7c dc 5c 70 c0 00 00
>39a0 f0 70 70 c0 00 00 00 00
>39a8 fc 7c dc 5c 70 c0 00 00
>39b0 0f 35 d5 ff ff ff ff fd
>39b8 fd fd fd ff ff ff ff ff
>39c0 ff fd fd fd fd ff fc 00
>39c8 ff 57 5d f5 fd fd ff 0f
>39d0 03 ff 5f ff ff fd f7 d7
>39d8 d5 f5 f5 fd fd 0f 0f 00
>39e0 c0 c0 f0 70 7c 5c 5c 5c
>39f0 c0 c0 70 70 f0 c0 00 00
>39f8 3e 7f 70 3c 1e 07 7f 3e
>3a00 3e 7f 63 60 60 63 7f 3e
>3a08 63 63 63 7f 7f 63 63 63

```



```

>3a10 3e 3e 1c 1c 1c 1c 3e 3e
>3a18 7f 7f 60 7c 7c 60 7f 7f
>3a20 63 73 7b 7f 6f 67 63 63
>3a28 1c 38 70 70 70 70 38 1c
>3a30 38 1c 0e 0e 0e 0e 1c 38
>3a38 1c 1c 3c 7c 1c 1c 7f 7f
>3a40 3e 7f 63 7f 3f 03 7f 3e
>3a48 3e 7f 63 3e 3e 63 7f 3e
>3a50 3e 7f 60 7e 7f 63 7f 3e
>3a58 7e 7f 67 7e 7e 67 7f 7e
>3a60 63 63 63 7f 3e 1c 1c 1c
>3a68 63 7f 7f 7f 6b 63 63 63
>3a70 1c 3e 63 7f 7f 63 63 63
>3a78 60 60 60 60 60 60 7f 7f
>3a80 7e 7f 63 7f 7e 60 60 60
>3a88 3e 7f 63 63 63 63 7f 3e
>3a90 7f 7f 1c 1c 1c 1c 1c 1c
>3a98 aa aa aa aa aa aa aa aa
>3aa0 ff ff ff ff ff ff ff ff
>3aa8 55 55 55 55 55 55 55 55
>3ab0 aa 80 80 80 80 80 80 80
>3ab8 aa 80 80 80 80 80 80 80
>3ac0 aa 02 02 02 02 02 02 02
>3ac8 7a 5e d7 b5 ad ab aa aa
>3ad0 aa aa aa aa aa aa aa aa
>3ad8 b5 b5 b5 b5 b5 b5 b5 f5
>3ae0 5e 5e 5e 5e 5e 5e 5e 5f
>3ae8 aa aa aa ab ad b5 d7 5e
>3af0 ad b5 d7 5e 7a ea aa aa
>3af8 aa ff 55 55 55 55 ff aa
>3b00 00 00 00 00 00 00 00 00
>3b08 ad ff 55 55 55 55 ff ad
>3b10 75 f5 55 55 55 55 f5 75
>3b18 5d 5f 55 55 55 55 5f 5d
>3b20 7a ff 55 55 55 55 ff 7a
>3b28 aa aa ab ad b5 d7 5e 7a
>3b30 b5 d7 5e 7a ea aa aa aa
>3b38 f5 b5 b5 b5 b5 b5 b5 b5
>3b40 5f 5e 5e 5e 5e 5e 5e 5e
>3b48 5e d7 b5 ad ab aa aa aa
>3b50 aa ea 7a 5e d7 b5 ad
>3b58 aa ee aa bb aa ee aa bb
>3b60 af ef af a5 a5 e5 af af
>3b68 ff ff ff 55 55 55 ff ff
>3b70 aa ee aa 55 ff ff ff 55
>3b78 af e5 a5 55 ff ff ff 55
>3b80 ff 55 55 55 ff ff ff 55
>3b88 55 55 ff ff ff 55 55 55
>3b90 00 00 00 00 00 00 00 01
>3b98 00 04 0e 1c 38 70 e0 c0
>3ba0 43 67 7e 7c 7c 7e 7f 00
>3ba8 80 00 00 00 00 00 00 00
>3bb0 00 7f 7e 7c 7c 7e 67 43
>3bb8 00 00 00 00 00 00 00 00
>3bc0 01 00 00 00 00 00 00 00
>3bc8 c0 e0 70 38 1c 0e 04 00
>3bd0 00 00 18 18 18 38 38 18
>3bd8 00 00 7c 7c 66 66 66 7c
>3be0 00 00 60 60 60 60 60 60
>3be8 00 00 18 18 3c 3c 66 7e
>3bf0 00 00 66 66 66 66 66 3c
>3bf8 00 00 7e 7e 60 60 60 78
>3c00 00 00 7c 7c 66 66 66 7c
>3c08 18 18 18 18 7e 7e 00 00
>3c10 7c 60 60 60 60 60 00 00
>3c18 60 60 60 60 7e 7e 00 00
>3c20 7e 66 66 66 66 66 00 00
>3c28 3c 18 18 18 18 18 00 00
>3c30 78 60 60 60 7e 7e 00 00
>3c38 7c 78 6c 6c 66 66 00 00
>3c40 00 00 3c 3c 66 66 0c 0c
>3c48 30 30 60 60 7e 7e 00 00
>3c50 ff ff c3 c3 99 99 f3 f3

```

```

>3c58 ff ff 83 83 99 99 99 83
>3c60 ff ff 9f 9f 9f 9f 9f 9f
>3c68 ff ff e7 e7 c3 c3 99 81
>3c70 ff ff 99 99 99 99 99 c3
>3c78 ff ff 81 81 9f 9f 9f 87
>3c80 ff ff 83 83 99 99 99 83
>3c88 cf cf 9f 9f 81 81 ff ff
>3c90 8f 9f 9f 9f 81 81 ff ff
>3c98 9f 9f 9f 9f 81 81 ff ff
>3ca0 81 99 99 99 99 99 ff ff
>3ca8 c3 e7 e7 e7 e7 ff ff
>3cb0 87 9f 9f 9f 81 81 ff ff
>3cb8 83 87 93 93 99 99 ff ff
>3cc0 ff ff e7 e7 e7 c7 c7 e7
>3cc8 e7 e7 e7 e7 81 81 ff ff
>3cd0 ff 0f ff 0f 0f ff 0f ff
>3cd8 ff 0f ff 0f 0f ff 0f ff
>3ce0 ff 0f ff 0f 0f ff 0f ff
>3ce8 ff 0f ff 0f 0f ff 0f ff
>3cf0 00 03 06 09 20 0d 10 13
>3cf8 15 18 1b 1d 21 25 28 2a
>3d00 2d 15 30 33 36 39 3c 01
>3d08 04 07 0a 20 0e 11 14 16
>3d10 19 1c 1e 23 26 2b 2e
>3d18 16 31 34 37 3a 3d 02 05
>3d20 08 0b 0c 0f 12 20 17 1a
>3d28 17 1f 24 27 29 2c 2f 17
>3d30 32 35 38 3b 3e 3f 40 41
>3d38 42 43 3f 3f 43 44 45 40
>3d40 46 20 47 48 49 4a 20 4b
>3d48 4c 20 4d 42 40 41 4e 43
>3d50 4f 20 50 51 52 3f 40 41
>3d58 53 53 54 54 55 55 20 20
>3d60 56 57 57 57 57 58 20 20
>3d68 59 5a 5b 5c 5d 5e 20 20
>3d70 6b 6b 6c 6d 6d 6d 53 53
>3d78 54 54 55 55 20 20 55 55
>3d80 55 55 55 55 20 20 5f 61
>3d88 62 63 64 5f 20 20 6e 6e
>3d90 6f 70 70 70 53 53 54 54
>3d98 55 55 20 20 54 54 54 54
>3da0 54 54 20 20 65 66 67 68
>3da8 69 6a 20 20 71 71 71 71
>3db0 71 71 7a 20 7b 7c 7d 7e
>3db8 7f 80 81 20 82 83 84 85
>3dc0 86 87 88 20 7b 7c 7d 7e
>3dc8 7f 80 89 20 82 83 84 85
>3dd0 86 87 8a 54 8b 8c 8d 8e
>3dd8 8f 90 91 54 92 93 94 95
>3de0 96 97 98 54 8b 8c 8d 8e
>3de8 8f 90 94 54 92 93 94 95
>3df0 96 97 8b 75 76 79 7c 7f
>3df8 82 85 88 77 7a 7d 80 83
>3e00 86 89 71 72 81 84 87 8a
>3e08 73 74 78 7b 7e 3f 36 2e
>3e10 2f 44 31 4a 33 37 31 3c
>3e18 5b 38 2e 3a 2d 36 45 46
>3e20 3b 44 31 3c 33 37 47 3c
>3e28 33 37 31 3c 33 37 31 35
>3e30 2d 36 2e 3a 2d 30 39 49
>3e38 54 37 3d 34 3e 4b 4c 31
>3e40 3c 30 33 5d 55 6f 34 3e 38
>3e48 2e 3a 61 31 6b 5e 5f 37
>3e50 31 3c 33 38 2e 3a 2d 5d
>3e58 64 65 36 2e 2f 44 31 35
>3e60 33 60 2e 3a 61 31 35 3b
>3e68 38 2e 2f 30 39 2f 40 41
>3e70 2f 30 39 2f 30 42 43 36
>3e78 2e 2f 37 3d 34 70 50 51
>3e80 30 2e 3e 5d 68 67 36 62
>3e88 6a 30 34 33 30 34 3e 5a
>3e90 4d 52 53 4e 4f 48 33 55
>3e98 56 57 58 59 5c 6c 33 5d

```

```

>3ea0 69 30 63 ef f0 2d f1 f2
>3ea8 f3 2d f4 f5 36 f6 f7 19
>3eb0 15 17 20 20 20 30 34 33
>3eb8 2d 20 37 36 2e 2f c1 c2
>3ec0 c3 c4 c5 34 20 c7 c8 c9
>3ec8 ca cb cc cd ce 34 3e 20
>3ed0 cf d0 d1 d2 d3 d4 20 20
>3ed8 38 d6 d7 d8 33 ba bb bc
>3ee0 bd b6 b7 b8 b9 20 e2 e3
>3ee8 2e 3a e4 e5 e6 20 20 e8
>3ef0 e9 ea 55 6f eb ec 43 20
>3ef8 20 ed ee 8b 75 90 92 94
>3f00 95 97 99 88 77 7a 7d 96
>3f08 98 9a 9b 8c 8e 81 84 87
>3f10 9c 8d 8f 91 93 7e 0b 10
>3f18 0a 13 16 11 15 14 1d 08
>3f20 06 0d 14 0f 12 0a 14 0c
>3f28 1a 18 0a 0e 17 0a 14 0b
>3f30 0d 0f 0d 17 13 0a 14 1b
>3f38 0f fb 17 00 00 f0 00 f0
>3f40 f0 00 f0 00 00 f0 00 f0
>3f48 f0 00 f0 00 00 f0 00 f0
>3f50 f0 00 f0 00 00 f0 00 f0
>3f58 f0 00 f0 00 00 f0 00 f0
>3f60 f0 00 f0 00 00 f0 00 f0
>3f68 f0 00 f0 00 00 f0 00 f0
>3f70 f0 00 f0 00 00 f0 00 f0
>3f78 f0 00 f0 00 00 f0 00 f0
>3f80 ff 0f ff 0f 0f ff 0f ff
>3f88 ff 0f ff 0f 0f ff 0f ff
>3f90 ff 0f ff 0f 0f ff 0f ff
>3f98 ff 0f ff 0f 0f ff 0f ff
>3fa0 ff 0f ff 0f 0f ff 0f ff
>3fa8 ff 0f ff 0f 0f ff 0f ff
>3fb0 ff 0f ff 0f 0f ff 0f ff
>3fb8 ff 0f ff 0f 0f ff 0f ff
>3fc0 ff 0f ff 0f 0f ff 0f ff
>3fc8 ff 0f ff 0f 0f ff 0f ff
>3fd0 ff 0f ff 0f 0f ff 0f ff
>3fd8 ff 0f ff 0f 0f ff 0f ff
>3fe0 ff 0f ff 0f 0f ff 45 4c
>3fe8 46 4d 45 54 45 52 2d 48
>3ff0 41 55 50 54 0e ff 8d 3e
>3ff8 ff 4c a4 f2 f6 ff 0f f0
>4000 ff ff ff ff ff ff ff ff

```

**Beachten Sie
bitte auch
das Sonderheft
für
C16/116/plus 4!**

**Jetzt überall
beim Zeitschriften-
händler.**

Q-BERT

für C16/116/plus 4

Nach den Abenteuern der Hexe Waltraut (THE WITCH, veröffentlicht in "Compute mit", SA 3/86) nun ein neues Spiel für C-16 User mit dem Titel Q-Bert, ein rasantes Geschicklichkeitsspiel in extended-color-grafic und mehreren Schwierigkeitsgraden, sowie screens.

SPIELVERLAUFSBESCHREIBUNG:

Q-BERT liebt die Farbe blau über alles, weiß ist ihm ein Greuel, ein unhaltbarer Zustand, den er unbedingt ändern muß. Von der Spitze einer Pyramide aus frönt er seiner "Blauen Schaffensperiode"

und färbt nacheinander die Oberflächen der 21 Würfel um. Seinen beiden Widersachern, den BLACK-BERTS, zwei notorischen Schwarz-Weiß-Denkern, ist dieser Farbrausch natürlich ein Dorn in ihren verkniffenen Augen. Mit Hinterlist und Tücke versuchen sie ihn an seinem Werk zu hindern und ihre eigene Ordnung wiederherzustellen. In der Wahl ihrer Mittel nicht zimperlich, färben die BLACK-BERTS die Felder wieder um und versuchen Q-BERT unschädlich zu machen. Helfen Sie unserem farbenfrohen Helden und bewahren Sie ihn vor dem Absturz und seinen Gegnern. Viel Spaß!

Bitte beachten Sie die Änderungen für Diskettenbenutzer am Ende des Listing.

Teil 1

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

```

10 REM *****
20 REM ** SUPER Q-BERT 1 **
30 REM *****
40 REM
50 PRINT "CLEAR DOWN2 WHITE" LOAD (HOME)
60 POKE 1319,13:POKE 1320,82:POKE 1321,213
70 POKE 1322,13:POKE 239,4:NEW
ENDE DES LISTINGS

```

Teil 2

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

```

10 REM *****
20 REM ** SUPER Q-BERT **
30 REM *****
40 REM ***** WRITTEN 8.06 BY *****
50 REM ***** ANDREAS NIKISCH *****
60 REM ***** C16 C116 +4 *****
70 REM *****
80 REM
90 POKE 55,0:POKE 56,60:CLR
100 COLOR 0,1:COLOR 4,1:COLOR 1,7,2
110 COLOR 2,6,1:COLOR 3,15,0
120 SCNL:PRINT "WHITE DOWN RIGHT2 SPACE" PLAY (SPACE) TASTE (SPACE) GEDRUECKT (SPACE) LASSEN (SPACE) !!
130 PRINT CHR$(142)
140 POKE 1351,128
150 FORT=12288:TO 12935:READA:IFA)--1 THEN POKE T,A:NE
XT:SCNL:
160 POKE 65298,50:POKE 65299,PEEK (65299) AND 30R48
170 POKE 65286,PEEK (65286) OR 64
180 DATA 060,066,153,161,161,153,066,060
190 DATA 124,254,198,198,254,198,198,000
200 DATA 252,198,198,252,198,198,252,000

```

```

210 DATA 254,192,192,192,192,192,254,000
220 DATA 252,194,194,194,194,194,252,000
230 DATA 254,192,192,248,192,192,254,000
240 DATA 254,192,192,248,192,192,254,000
250 DATA 060,192,192,206,206,194,060,000
260 DATA 198,198,198,254,198,198,198,000
270 DATA 024,024,024,024,024,024,024,000
280 DATA 254,134,006,006,134,134,126,000
290 DATA 198,204,216,240,216,204,198,000
300 DATA 192,192,192,192,192,192,254,000
310 DATA 198,238,254,214,198,198,198,000
320 DATA 198,230,246,222,206,198,198,000
330 DATA 254,198,198,198,198,198,254,000
340 DATA 254,198,198,254,192,192,192,000
350 DATA 254,198,198,198,214,206,254,001
360 DATA 254,198,198,254,240,216,204,000
370 DATA 254,192,192,254,006,006,254,000
380 DATA 254,016,016,016,016,016,016,000
390 DATA 198,198,198,198,198,198,124,000
400 DATA 198,198,198,198,108,056,016,000
410 DATA 198,198,198,214,252,238,198,000
420 DATA 198,108,056,016,056,108,198,000
430 DATA 198,108,056,016,016,016,016,000
440 DATA 254,006,012,060,048,096,254,000
450 DATA 240,224,192,140,158,146,158
460 DATA 015,007,003,051,121,073,073,121
470 DATA 192,196,231,240,248,249,129
480 DATA 001,035,231,015,031,159,159,129
490 DATA 240,224,192,144,152,156,158,153
500 DATA 000,000,000,000,000,000,000,000
510 DATA 015,007,003,009,025,057,121,153
520 DATA 153,192,227,243,248,249,129
530 DATA 153,003,199,207,031,159,159,129
540 DATA 128,224,248,254,255,255,255,255
550 DATA 000,000,000,000,128,224,248,254
560 DATA 127,031,007,001,000,000,000,000
570 DATA 255,255,255,255,127,031,007,001
580 DATA 001,007,031,127,255,255,255,255
590 DATA 000,000,000,000,001,007,031,127
600 DATA 254,248,224,128,000,000,000,000
610 DATA 255,255,255,255,254,248,224,128
620 DATA 255,255,255,255,255,255,255,255
630 DATA 000,000,000,255,255,000,000,000
640 DATA 000,000,000,000,000,024,024,000
650 DATA 000,000,000,000,024,000,000,024,000
660 DATA 254,198,206,214,230,198,254,000
670 DATA 056,120,024,024,024,024,024,000
680 DATA 254,006,006,254,192,192,254,000
690 DATA 254,006,006,062,006,006,254,000
700 DATA 192,192,200,200,254,000,000,000
710 DATA 254,192,192,254,002,002,254,000
720 DATA 254,192,192,254,198,198,254,000
730 DATA 254,006,012,060,048,096,192,000
740 DATA 124,198,198,124,198,198,124,000
750 DATA 254,198,198,254,006,006,254,000
760 DATA 000,024,024,000,000,024,024,000
770 DATA 000,126,126,126,126,126,126,000
780 DATA 000,000,000,000,000,000,192,192
790 DATA 000,000,255,000,255,000,000,000
800 DATA 002,002,000,000,000,000,000,000
810 DATA 153,192,227,243,248,249,249,129
820 DATA -01
830 SCNL:
840 POKE 65286,PEEK (65286) AND 191
850 POKE 65299,208:POKE 65298,196
860 COLOR 1,1
870 PRINT "DOWN2" LOAD (HOME)
875 REM 2. TEIL MIT "SAVE" (CLEAR WHITE SPACE) Q-BERT (BLACK)
880 POKE 1319,13:POKE 1320,82:POKE 1321,213
890 POKE 1322,13:POKE 239,4
900 NEW
ENDE DES LISTINGS

```

Teil 3

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

```

10 REM *****
20 REM ** SUPER Q-BERT **
30 REM *****
40 REM ***** WRITTEN 1986 BY *****
50 REM * ANDREAS NIKISCH *
60 REM *****

```



```
70 REM * C16/C116 & PLUS4 *
```

```
80 REM *****
```

```
90 REM
```

```
100 COLOR4,1:VOLB,COLOR5,1:COLOR3,15,0
```

```
110 POKE65298,50:POKE65299,PEEK(65299)AND3OR4B:P
```

```
OKE65304,239:POKE65303,21
```

```
120 POKE65286,PEEK(65286)OR64
```

```
130 SCNCRLR:POKE65286,PEEK(65286)AND239
```

```
140 PRINTCHR$(142);POKE1351,128
```

```
150 COLOR1,9,3
```

```
160 PRINT"(RIGHT3 SPACE RIGHT);";;(SPACE);(SPAC
```

```
E3);(SPACE);";;(SPACE);";;(SPACE);";;"COLOR1
```

```
,9,4
```

```
170 PRINT"(RIGHT2 SPACE RIGHT2);(SPACES);(SPACE3
```

```
);(SPACE);(SPACE3);(SPACE);(SPACE3);(SPACE3);"C
```

```
OLDR1,9,5
```

```
180 PRINT"(RIGHT2 SPACE RIGHT2);";;(SPACE);(SPA
```

```
C E3);(SPACE);";;(SPACE);";;(SPACE2);";;"COLOR
```

```
1,9,6
```

```
190 PRINT"(RIGHT4 SPACE RIGHT4);(SPACE);(SPACE3)
```

```
; (SPACE);(SPACE3);(SPACE5);";"COLOR1,9,7
```

```
200 PRINT"(RIGHT SPACE RIGHT3);";;(SPACE);";;(S
```

```
PACE);(SPACE5);";;(SPACE);(SPACE);";"COLOR1,6
```

```
,3
```

```
210 PRINT"(DOWN SPACE2);";;(SPACE6);";;(SPACE2
```

```
)";;(SPACE);";;(SPACE);";;(SPACE);";"COLOR1,6,4
```

```
220 PRINT"(SPACE2);(SPACE3);(SPACE6);(SPACE4);($
```

```
PAGE);(SPACE5);(SPACE3);(SPACE3);"COLOR1,6,5
```

```
230 PRINT"(SPACE2);(SPACE3);(SPACE);";;(SPACE);"
```

```
";;(SPACE2);";;(SPACE2);";;(SPACE3);"COLOR1,6,
```

```
6
```

```
240 PRINT"(SPACE2);(SPACE2);(SPACE6)(SPACE4);(S
```

```
PAGE);(SPACE5);(SPACE6);"COLOR1,6,7
```

```
250 PRINT"(SPACE2);";;(SPACE5);";;(SPACE2);";;
```

```
(SPACE);(SPACE);";(SPACE3)"
```

```
260 PRINT"(DOWN3 WRITTEN(SPACE)FOR(SPACE)C(SPACE
```

```
)16(SPACE)1986(SPACE)BY(SPACE)ANDREAS(SPACE)NIKI
```

```
SCH"
```

```
270 PRINT"(DOWN3 RIGHT SPACE RIGHT SPACE RIGHT S
```

```
PACE RIGHT SPACE RIGHT SPACE RIGHT)PRESS(SPACE)F
```

```
I RE(SPACE)TO(SPACE)PLAY":POKE65286,PEEK(65286)OR
```

```
16
```

```
280 IFJOY(1)=128ORJOY(2)=128THEN300
```

```
290 GOTO280
```

```
300 FORX=500TO900STEP10:SOUND1,X,2:SOUND2,X+100,
```

```
2:NEXT
```

```
310 SCNCCLR
```

```
320 COLOR1,2
```

```
330 F*(1)="(UP)%(DOWN LEFT4);(,%*(DOWN LEFT6 R
```

```
VSON)&'(RVSOFF),,(RVSON CQ CN DOWN LEFT4)&(CN RV
```

```
SOFF)"
```

```
340 F*(2)="(UP)(RVSON)%(DOWN LEFT4 RVSOFF);(,(
```

```
RVSON)%*(DOWN LEFT6)&'(RVSOFF),,(RVSON CQ CN DOH
```

```
N LEFT4)&(CN RVSOFF)"
```

```
350 F*(3)="(UP RVSON S\ RVSOFF)%(DOWN LEFT4 RVSO
```

```
N S\ C\ RVSOFF),,%*(DOWN LEFT6 RVSON)&'(RVSOFF),
```

```
(RVSON CQ CN DOWN LEFT4)&(CN RVSOFF)"
```

```
360 F*(4)="(UP RVSON S\%(DOWN LEFT4 S\ C\ RVSOFF
```

```
F),(RVSON)%*(DOWN LEFT6)&'(RVSOFF),,(RVSON CQ C
```

```
N DOWN LEFT4)&(CN RVSOFF)"
```

```
370 L$="(DOWN LEFT2),,(RVSOFF)"
```

```
380 B*(1)="(UP)%(DOWN LEFT4);(!_*(DOWN LEFT6 R
```

```
VSON)&'(RVSOFF)?#(RVSON CQ CN DOWN LEFT4)&(CN RV
```

```
SOFF)"
```

```
390 B*(2)="(UP)(RVSON)%(DOWN LEFT4 RVSOFF);(!_*
```

```
RVSON)%*(DOWN LEFT6)&'(RVSOFF)?#(RVSON CQ CN DON
```

```
H LEFT4)&(CN RVSOFF)"
```

```
400 B*(3)="(UP RVSON S\ RVSOFF)%(DOWN LEFT4 RVSD
```

```
N S\ C\ RVSOFF)!%*(DOWN LEFT6 RVSON)&'(RVSOFF)?
```

```
#(RVSON CQ CN DOWN LEFT4)&(CN RVSOFF)"
```

```
410 B*(4)="(UP RVSON S\%(DOWN LEFT4 S\ C\ RVSOFF
```

```
F):(RVSON)%*(DOWN LEFT6)&'(RVSOFF)?#(RVSON CQ C
```

```
N DOWN LEFT4)&(CN RVSOFF)"
```

```
420 D*(1)="(UP)%(DOWN LEFT4)((S+ C-)*%*(DOWN LE
```

```
FT6 RVSON)&'(RVSOFF S-)π(RVSON CQ CN DOWN LEFT4)
```

```
&(CN RVSOFF)"
```

```
430 D*(2)="(UP)(RVSON)%(DOWN LEFT4 RVSOFF);((S+
```

```
C- RVSON)%*(DOWN LEFT6)&'(RVSOFF S-)π(RVSON CQ
```

```
CN DOWN LEFT4)&(CN RVSOFF)"
```

```
440 Q*(3)="(UP RVSON S\ RVSOFF)%(DOWN LEFT4 RVSD
```

```
N S\ C\ RVSOFF S+ C-)*%*(DOWN LEFT6 RVSON)&'(RVSD
```

```
FE S):(RVSON CQ CN DOWN LEFT4)&(CN RVSOFF)"
```

```
450 Q*(4)="(UP RVSON S\%(DOWN LEFT4 S\ C\ RVSOFF
```

```
F S+ C- RVSON)%*(DOWN LEFT6)&'(RVSOFF S-)π(RVSON
```

```
CQ CN DOWN LEFT4)&(CN RVSOFF)"
```

<219>	460 SCX=0:LIX=5:F%=0:C1%=2:C2%=6:LEX=1	<125>
<155>	470 POKE65286,PEEK(65286)AND239:Q1=19:Q2=4:G1=4:	
<233>	62=19:G3=34:G4=19:C1%=2:C2%=6	<41>
<22>	480 SCNLCL:COLOR1,2,6:IFLEX=STHEN1580	<165>
<127>	490 CHAR1,19,4,Q\$(1):CHAR1,16,7,F\$(2):CHAR1,22,7	
<22>	,F\$(3):CHAR1,13,10,F\$(2)	<94>
<195>	500 CHAR1,19,10,F\$(4):CHAR1,25,10,F\$(3):CHAR1,10	
<144>	,13,F\$(2):CHAR1,16,13,F\$(4)	<32>
<85>	510 CHAR1,22,13,F\$(4):CHAR1,28,13,F\$(3):CHAR1,7,	
	16,F\$(2):CHAR1,13,16,F\$(4)	<38>
	520 CHAR1,19,16,F\$(4):CHAR1,25,16,F\$(4):CHAR1,31	
	,16,F\$(3):CHAR1,4,19,G\$(2)	<103>
<47>	530 CHAR1,10,19,F\$(4):CHAR1,16,19,F\$(4):CHAR1,22	
	,19,F\$(4):CHAR1,28,19,F\$(4)	<187>
	540 CHAR1,34,19,G\$(3):CHAR1,18,6,"(RVSON SPACE)"	
<54>	:CHAR1,21,6,"(RVSON SHIFTSPEACE)"CHAR1,15,9,"(RV	
	SON SPACE)"	<35>
<110>	550 CHAR1,18,9,"(RVSON SHIFTSPEACE)":CHAR1,21,9,"	
	(RVSON SPACE)":CHAR1,24,9,"(RVSON SHIFTSPEACE)":C	<243>
<245>	HAR1,12,12,"(RVSON SPACE)"	
	560 CHAR1,15,12,"(RVSON SHIFTSPEACE)":CHAR1,18,12	
	,,"(RVSON SPACE)":CHAR1,21,12,"(RVSON SHIFTSPEACE)	
	:CHAR1,24,12,"(RVSON SPACE)"	<38>
<190>	570 CHAR1,27,12,"(RVSON SHIFTSPEACE)":CHAR1,9,15,	
	"(RVSON SPACE)":CHAR1,12,15,"(RVSON SHIFTSPEACE)"	
<242>	:CHAR1,15,15,"(RVSON SPACE)"	<25>
	580 CHAR1,18,15,"(RVSON SHIFTSPEACE)":CHAR1,21,15	
<118>	,,"(RVSON SPACE)":CHAR1,24,15,"(RVSON SHIFTSPEACE)	
	:CHAR1,27,15,"(RVSON SPACE)"	<34>
<66>	590 CHAR1,30,15,"(RVSON SHIFTSPEACE)":CHAR1,6,18,	
	"(RVSON SPACE)":CHAR1,9,18,"(RVSON SHIFTSPEACE)":	
	CHAR1,12,18,"(RVSON SPACE)"	<145>
<107>	600 CHAR1,15,18,"(RVSON SHIFTSPEACE)":CHAR1,18,18	
	,,"(RVSON SPACE)":CHAR1,21,18,"(RVSON SHIFTSPEACE)	
<229>	:CHAR1,24,18,"(RVSON SPACE)"	<6>
	610 CHAR1,27,18,"(RVSON SHIFTSPEACE)":CHAR1,30,18	
	,,"(RVSON SPACE)":CHAR1,33,18,"(RVSON SHIFTSPEACE)	
<48>	:CHAR1,3,21,"(RVSON SPACE)"	<80>
	620 CHAR1,6,21,"(RVSON SHIFTSPEACE)":CHAR1,9,21,"	
	(RVSON SPACE)":CHAR1,12,21,"(RVSON SHIFTSPEACE)":	
	CHAR1,15,21,"(RVSON SPACE)"	<222>
<195>	630 CHAR1,18,21,"(RVSON SHIFTSPEACE)":CHAR1,21,21	
<244>	,,"(RVSON SPACE)":CHAR1,24,21,"(RVSON SHIFTSPEACE)	
<120>	:CHAR1,27,21,"(RVSON SPACE)"	<244>
	640 CHAR1,30,21,"(RVSON SHIFTSPEACE)":CHAR1,33,21	
<128>	,,"(RVSON SPACE)":CHAR1,36,21,"(RVSON SHIFTSPEACE)	
<31>	:COLOR1,1:CHAR1,0,21,"	<203>
<214>	650 PRINT"(RIGHT2 RVSON)%<SPACE RIGHT2 SHIFTSPEAC	
	E S\)%<SPACE RIGHT2 SHIFTSPEACE S\)%<SPACE RIGHT2	
	SHIFTSPEACE S\)%<SPACE RIGHT2 SHIFTSPEACE S\)%<SP	
<71>	ACE RIGHT2 SHIFTSPEACE S\)%<SPACE RIGHT2 SHIFTSPEAC	
	CE S\)"	<134>
	660 CHAR1,0,22,"":PRINT"(RVSON RIGHT3)%\$(S\ C\ R	
<159>	IGHT2)%\$(S\ C\ RIGHT2)%\$(S\ C\ RIGHT2)%\$(S\ C\ R	
	IGHT2)%\$(S\ C\ RIGHT2)%\$(S\ C\)"	<235>
<221>	670 CHAR1,2,0,"(RED)":PRINTUSING"SCORE:(SPACE)##	
	####":SCX:CHAR1,18,0,"(BLUE)"	<180>
	680 PRINTUSING"LIVES:(SPACE)##":LIX:CHAR1,28,0,"(
	YELLOW)":PRINT"LEVEL:LE	<78>
<37>	690 C3%=C1%C5%=C1%C4%=C2%C6%=C2%	<188>
<153>	700 POKE65286,PEEK(65286)OR16	<43>
	710 CHAR1,2,0,"(RED)":PRINTUSING"SCORE:(SPACE)##	
	####":SCX:CHAR1,18,0,"(BLUE)"	<220>
<128>	720 PRINTUSING"LIVES:(SPACE)##":LIX:CHAR1,28,0,"(
	YELLOW)":PRINT"LEVEL:LE	<118>
	730 COLOR1,C1%,C2%:IFFX=21THEN1410	<3>
<60>	740 IFPEEK(3072+Q1+(Q2*40))<91THEN1300	<25>
	750 IFJOY(1)=2ORJOY(2)=2THENCHAR1,Q1,Q2,L\$:Q1=Q1	
	+3:Q2=Q2-3:GOTO810	<186>
<222>	760 IFJOY(1)=3ORJOY(2)=3THENCHAR1,Q1,Q2,L\$:Q1=Q1	
	+6:GOTO810	<127>
	770 IFJOY(1)=4ORJOY(2)=4THENCHAR1,Q1,Q2,L\$:Q1=Q1	
<194>	+3:Q2=Q2+3:GOTO810	<208>
	780 IFJOY(1)=6ORJOY(2)=6THENCHAR1,Q1,Q2,L\$:Q1=Q1	
	-3:Q2=Q2+3:GOTO810	<37>
<225>	790 IFJOY(1)=7ORJOY(2)=7THENCHAR1,Q1,Q2,L\$:Q1=Q1	
	-6:GOTO810	<19>
	800 IFJOY(1)=8ORJOY(2)=8THENCHAR1,Q1,Q2,L\$:Q1=Q1	
<255>	-3:Q2=Q2-3:GOTO810:ELSE890	<60>
	810 IFPEEK(2048+Q1+(Q2*40))=77THENC1%=2:C2%=6:FX	
	=FX-1:SOUND3,950,10:GOTO830	<14>
<3>	820 C1%=1:C2%=4:SOUND1,500,10:SOUND2,502,10:FX=	
	FX+1:SCX=SCX+10	<0>
	830 IFPEEK(3072+Q1+(Q2*40))=32THEN1340	<6>
<133>	840 IFPEEK(3072+Q1+(Q2*40)-40)=41ANDPEEK(3072+Q1	

```

+ (Q2*40) -40+1)=37THENK%=1:GOTO870
850 IFPEEK (3072+Q1+(Q2*40)-40)=41THENK%=2:GOTO870
860 IFPEEK (3072+Q1+(Q2*40)-40+1)=37THENK%=3:ELSEK%=4
870 IFPEEK (3072+Q1+(Q2*40))>44THEN1300
880 COLOR1,C1%,C2%,CHAR1,Q1,Q2,Q$(K%)
890 G3=6+1:ON6GOTO710,710,710,710,900
900 ONINT (RND (1)*2)+1GOTO100,910
910 G%-7-LEX:COLOR1,C3%,C4%,CHAR1,G1,G2,L$:ONINT (RND (1)*6)GOTO920,940,960,980,1000,1020
920 G1=G1+3:G2=G2-3:P%=PEEK (3072+G1+(G2*40))
930 IFP%=31ORP%=32THENG1=G1-3:G2=G2+3:GOTO1060:ELSE1040
940 G1=G1+6:P%=PEEK (3072+G1+(G2*40)):IFP%=31ORP%=32THENG1=G1-6:GOTO1060
950 GOTO1040
960 G1=G1+3:G2=G2+3:P%=PEEK (3072+G1+(G2*40))
970 IFP%=31ORP%=32THENG1=G1-3:G2=G2-3:GOTO1060:ELSE1040
980 G1=G1-3:G2=G2+3:P%=PEEK (3072+G1+(G2*40))
990 IFP%=31ORP%=32THENG1=G1+3:G2=G2-3:GOTO1060:ELSE1040
1000 G1=G1-6:P%=PEEK (3072+G1+(G2*40)):IFP%=31ORP%=32THENG1=G1+6:GOTO1060
1010 GOTO1040
1020 G1=G1-3:G2=G2-3:P%=PEEK (3072+G1+(G2*40))
1030 IFP%=31ORP%=32THENG1=G1+3:G2=G2+3:GOTO1060
1040 IFPEEK (2048+G1+(G2*40))=77THENC3%=14:C4%=4:ELSEC3%=2:C4%=6:GOTO1060
1050 IFINT (RND (1)*7-LEX)+1=1THENC3%=2:C4%=6:SOUND1,700,5:SOUND2,800,5:F%=F%-1
1060 IFPEEK (3072+G1+(G2*40)-40)=41ANDPEEK (3072+G1+(G2*40)-39)=37THENK%=1:GOTO1070
1070 IFPEEK (3072+G1+(G2*40)-40)=41THENK%=2:GOTO1090
1080 IFPEEK (3072+G1+(G2*40)-39)=37THENK%=3:ELSEK%=4
1090 COLOR1,C3%,C4%,CHAR1,G1,G2,G$(K%):GOTO710
1100 COLOR1,C5%,C6%,CHAR1,G3,G4,L$
1110 ONINT (RND (1)*6)+1GOTO1120,1140,1160,1180,1200,1220
1120 G3=G3+3:G4=G4-3:P%=PEEK (3072+G3+(G4*40))
1130 IFP%=31ORP%=32THENG3=G3-3:G4=G4+3:GOTO1260:ELSE1240
1140 G3=G3+6:P%=PEEK (3072+G3+(G4*40)):IFP%=31ORP%=32THENG3=G3-6:GOTO1260
1150 GOTO1240
1160 G3=G3+3:G4=G4+3:P%=PEEK (3072+G3+(G4*40))
1170 IFP%=31ORP%=32THENG3=G3-3:G4=G4-3:GOTO1260:ELSE1240
1180 G3=G3-3:G4=G4+3:P%=PEEK (3072+G3+(G4*40))
1190 IFP%=31ORP%=32THENG3=G3+3:G4=G4-3:GOTO1260:ELSE1240
1200 G3=G3-6:P%=PEEK (3072+G3+(G4*40)):IFP%=31ORP%=32THENG3=G3+6:GOTO1260
1210 GOTO1240
1220 G3=G3-3:G4=G4-3:P%=PEEK (3072+G3+(G4*40))
1230 IFP%=31ORP%=32THENG3=G3+3:G4=G4+3:GOTO1260
1240 IFPEEK (2048+G3+(G4*40))=77THENC5%=14:C6%=4:ELSEC5%=2:C6%=6:GOTO1260
1250 IFINT (RND (1)*7-LEX)=1THENC5%=2:C6%=6:SOUND1,700,5:SOUND2,800,5:F%=F%-1
1260 IFPEEK (3072+G3+(G4*40)-40)=41ANDPEEK (3072+G3+(G4*40)-39)=37THENK%=1:GOTO1290
1270 IFPEEK (3072+G3+(G4*40)-40)=41THENK%=2:GOTO1290
1280 IFPEEK (3072+G3+(G4*40)-39)=37THENK%=3:ELSEK%=4
1290 COLOR1,C5%,C6%,CHAR1,G3,G4,G$(K%):GOTO710
1300 COLOR1,C1%,C2%,CHAR1,Q1,Q2,Q$(K%):CHAR1,Q1,Q2,"(DOWN LEFT)?"
1310 FORD=900TO100STEP-20:SOUND3,D,2:NEXT
1320 L1=L1X-1:IFL1X=0THEN1450
1330 F%-0:GOTO470
1340 FORD=900TO100STEP-20:SOUND1,D,2:SOUND2,D+100,2:NEXT
1350 L1X=L1X-1:IFL1X=0THEN1450
1360 F%-0:GOTO470
1370 COLOR1,8,7:CHAR1,Q1,Q2,Q$=SOUND1,750,100:60UND2,570,100:COLOR1,1
1380 CHAR1,Q1,Q2,Q$=FORD=600TO200STEP-10:SOUND1,D,2:SOUND2,D+100,2:NEXT
1390 L1=L1X-1:IFL1X=0THEN1450
1400 F%-0:GOTO470

```

<109>	1410 FORD=100T0900STEP20:SOUND1,D,2:SOUND2,D+100,2	<80>
<203>	1420 NEXT:LIX=LIX+1:F=0:SC%=SC%+LEX*100:FORD=1T01000:NEXT:LIX=LEX+1:G%=7-LEX	<108>
<101>	1430 IFLEX=6THEN1470	<2>
<115>	1440 GOTO470	<251>
<57>	1450 CHAR1,15,13,"BLACK RVSON SG SA SM SE SHIFT	
<103>	SPACE2 SO SV SE SR)"	<85>
<18>	1460 FORD=900T0100STEP-10:SOUND1,D,2:SOUND2,D+100,2:NEXT:RUN	<49>
<77>	1470 FORD=100T0900STEP20:SOUND1,D,3:SOUND2,D+100,3:NEXT:SCNCLR	<45>
<111>	1480 POKE65286,PEEK(65286)AND239:COLOR1,7,2:PRINT" (DOWN3)"	<176>
<18>	1490 PRINT" (SPACE10)";;;; (SPACE); (SPACE);;;; (SPACE);;;; " :COLOR1,7,3	<53>
<244>	1500 PRINT" (SPACE10); (SPACE5); (SPACE); (SPACE5); " :COLOR1,7,4	<228>
<178>	1510 PRINT" (SPACE10)";;;; (SPACE); (SPACE);;;; (SPACE); (SPACE2); " :COLOR1,7,5	<148>
<71>	1520 PRINT" (SPACE14); (SPACE); (SPACE); (SPACE5); (SPACE3); " :COLOR1,7,6	<147>
<179>	1530 PRINT" (SPACE10)";;;; (SPACE); (SPACE);;;; (SPACE);;;; " 1540 PRINT" (DOWN3 YELLOW SPACE3)SIE (SPACE)HABEN (SPACE)DIE (SPACE)GEGNER (SPACE)BEZWUNGEN (SPACE)"	<18>
<35>	1550 PRINT" (DOWN2 CYAN SPACE3) IHR (SPACE)SCORE: "B CX	<172>
<48>	1560 PRINT" (DOWN4 RED SPACE14 RVSON SP SR SE SS2 SHIFTSPACE SF SI SR SE)" :POKE65286,PEEK(65286)O R16	<132>
<125>	1570 IFJOY(1)=128ORJOY(2)=128THENRUN:ELSEGOTO1570	<156>
<62>	1580 CHAR1,4,4,G\$(1):CHAR1,10,4,F\$(1):CHAR1,16,4,F\$(1):CHAR1,22,4,F\$(1):RESTORE	<137>
<39>	1590 CHAR1,28,4,F\$(1):CHAR1,34,4,G\$(1)	<161>
<192>	1600 FORD=1T015:READA,B:CHAR1,A,B,F\$(4):NEXT	<217>
<242>	1610 DATA 7,13,7,19,7,25,7,31,7,10,10,16,10,22,10,28,10,13,13,19,13,25,13	<132>
<192>	1620 DATA 16,16,22,16,19,19	<121>
<239>	1630 A\$="(RVSON SPACE)":FORD=1T021:READA,B:CHAR1,A,B,A\$:NEXT	<239>
<122>	1640 DATA 3,6,9,6,15,6,21,6,27,6,33,6,6,9,12,9,18,9,24,9,30,9,9,12,15,12,21,12	<5>
<46>	1650 DATA 27,12,12,15,18,15,24,15,15,18,21,18,18,21	<73>
<144>	1660 A\$="(RVSON SHIFTSPEACE)":FORD=1T021:READA,B:CHAR1,A,B,A\$:NEXT	<86>
<103>	1670 DATA 6,6,12,6,18,6,24,6,30,6,36,6,9,9,15,9,21,9,27,9,33,9,12,18,12,24,12	<35>
<195>	1680 DATA 30,12,15,15,21,15,27,15,18,18,24,18,21,21	<68>
<170>	1690 A\$="(RVSON BLACK)% (DOWN)%%":FORD=1T06:READA,B:CHAR1,A,B,A\$:NEXT	<41>
<171>	1700 DATA 2,6,5,9,8,12,11,15,14,18,17,21	<156>
<196>	1710 A\$="(RVSON BLACK SI CI UP SI)":FORD=1T06:READA,B:CHAR1,A,B,A\$:NEXT	<124>
<172>	1720 DATA 35,7,32,10,29,13,26,16,23,19,20,22	<157>
<120>	1730 Q1=19:Q2=19:G1=4:G2=4:G3=34:G4=4:COLOR1,C1% ,C2%:CHAR1,19,19,G\$(4)	<254>
<255>	1740 GOTO670	<206>
<250>	1750 REM ***** E N D E *****	<44>
<234>	ENDE DES LISTINGS	<198>

Änderungen für Diskettenbenutzer

Mit folgenden Änderungen läuft das Programm SUPER Q-BERT auch auf Diskette. Der erste Teil entfällt und wird durch Teil 1.1 ersetzt. In Teil 2 ist die Zeile 870 durch die entsprechende Änderung zu ersetzen. Dieser wird mit DSAVE "SUPER Q-BERT 2" abgespeichert. Teil 3 braucht nicht geändert zu werden und wird mit DSAVE "SUPER Q-BERT 3" abgespeichert. Zum Spielen wird nur Teil 1 geladen und mit RUN gestartet. Dieser lädt die Teile 2 und 3 nach.

Teil 1.1

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

```
10 PRINT "CLEAR DOWN3" DLOAD "CHR$(34) "SUPER (SPACE
10-BERT (SPACE) 2"
20 PRINT "DOWN4" RUN (HOME)
30 POKE1319,13:POKE1320,13:POKE1321,13:POKE1322,
13:POKE1323,13:POKE239,5
ENDE DES LISTINGS
```

Änderung in Teil 2

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

```
870 PRINT "DOWN3" DLOAD "CHR$(34) "SUPER (SPACE) Q-BE
RT (SPACE) 3 "CHR$(34) " (HOME) "
ENDE DES LISTINGS
```

FROGGY!

Übersteht er die unheilvolle Nacht?

C-16/116
plus 4

In einem Teich sitzt auf Seerosenblättern ein Frosch, der jede Nacht von einem Schwarm Zauberfliegen heimgesucht wird. Diese Fliegen sind in der Lage seine Seerosenblätter wegzuschneiden. Auf diese Weise veranlassen sie den Frosch, der seltsamerweise nicht schwimmen kann, den Fliegen nachzustellen bevor das letzte Blatt verschwunden ist. Dann nämlich fiele er ins Wasser und das Spiel wäre leider beendet. Ist der Frosch geschickt so fängt er möglichst viele Fliegen ohne naß zu werden und übersteht diese unheilvolle Nacht.

Steuerung mit TASTATUR:
4= 4 Schritte = Sprung nach links
5= 1 Schritt nach links
8= 1 Schritt nach rechts
9= 4 Schritte = Sprung nach rechts
6/7= Der Frosch schnappt nach der Fliege

oder mit dem JOYSTICK:

Feuertaste betätigen und den Joystick nach links bewegen: Der Frosch springt nach links und in die Höhe nach der Fliege.

Feuertaste betätigen und den Joystick nach rechts bewegen: Der Frosch springt nach rechts und in die Höhe nach der Fliege.

Die Bewegungen des Frosches werden über Tastatur oder mit dem Joystick gesteuert.

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

```
3 COLOR0,1:COLOR1,2:COLOR4,1:GOTO40000
10 KEY1,"(BLACK)M(SO)" +CHR$(13) + "TD000(SPACE)D7F
F(SPACE)3400" +CHR$(13) + "X" +CHR$(13) + "G(SO)27 (WHI
TE)" +CHR$(13)
11 KEY2,">FF12(SPACE)C0(SPACE)30" +CHR$(13) + ">031
4(SPACE)00(SPACE)3C" +CHR$(13) + "X" +CHR$(13) + "G(SO
)28" +CHR$(13)
12 PRINT "CLEAR":VOL2:RESTORE14
13 FORI=0TO15:READX:POKE15360+I,X:NEXT
14 DATA133,208,173,18,255,41,0,141,18,255,165,20
8,76,14,206
26 PRINT "CLEAR WHITE DOWN2 RIGHT WHITE" BITTE (SP
ACE) 'F1' (SPACE) DRUECKEN (BLUE) " :END
27 PRINT "DOWN2 RIGHT WHITE" BITTE (SPACE) 'F2' (SPA
CE) DRUECKEN (BLUE) " :MONITOR
28 RESTORE50
29 POKE788,00:POKE789,60:POKE65298,192:POKE65299
,56
30 KEY2,"
40 FORQ=0TO423:READX:POKE13312+Q+0,X: SOUND1,4*X,
0:PRINT "WHITE CLEAR" X,0:NEXT
50 DATA0,12,18,23,63,53,24,15,15,31,223,255,127,
63,231,0
52 DATA0,48,72,216,252,236,24,240,240,248,251,25
5,254,252,231,0
54 DATA0,0,195,231,102,24,24,36
56 DATA7,28,16,16,93,206,103,51,31,15,15,31,31,3
1,15,7,15,30,62,124,240,224,112,48
```

```
58 DATA56,24,8,24,60,56,0,0,224,56,8,136,58,115,
230,204,248,240,240,248,248,248,240,240,240
60 DATA240,120,124,62,15,7,14,12,28,24,16,24,60,
28,0,0
62 DATA82,255,255,235,255,255,255,170
64 DATA18,26,77,37,213,107,23,95,64,80,84,171,17
4,188,248,218
66 DATA0,3,9,9,12,102,52,165,167,172,184,227,206
,248,195,222
68 DATA210,211,217,209,252,199,208,218,203,202,2
03,206,216,211,217,220
70 DATA213,247,209,221,197,204,202,201,192,254,2
02,218,218,211,217,213
72 DATA213,212,198,196,196,226,240,252
74 DATA38,20,13,135,206,88,115,195,31,121,108,16
7,48,158,208,88
76 DATA65,115,82,71,37,141,201,9,200,8,152,159,1
6,144,31,16
78 DATA16,16,33,33,33,33,33,32,34,34,39,39,39,39
,34,74
80 DATA94,94,94,94,94,74,74,229,0,64,192,4,6,214
,209,29,0,0,0,0,0,0
82 DATA68,224,32,128,0,60,36,228,2,2,2,2,0,0,0
84 DATA0,0,128,192,0,62,192,0,136,156,220,220,22
0,220,220,136
86 DATA168,250,250,255,255,175,175,175,170,170,1
70,170,170,170,170
88 DATA0,143,1,150,2,157,23,150,29,157,30,150,31
,143,0,0
90 DATA0,7,15,15,63,127,127,127,63,31,127,127,25
5,255,243,47,31
92 DATA63,127,127,127,127,126,60,24,0,192,224,24
0,252,254,254,254
94 DATA224,252,252,252,252,250,198,254,254,254,2
52,148,224,0,0,0
96 DATA0,207,239,239,239,239,215,251,251,251,251
,247,251,251,251,251
98 DATA251,251,199,255,251,0,0,0,0,239,223,223,2
23,223,239,239
100 DATA239,239,243,239,223,223,223,223,199,255,
255,255,231,0,0
102 DATA255,255,255,255,255,255,255,255,255,255
105 INPUT "CLEAR DOWN2" SCHWIERIGKEITSGRAD (SPACE)
(5=LEICHT (SPACE) - (SPACE) 1=SCHWER) " :SG:
110 POKE788,00:POKE789,60:POKE65298,192:POKE6529
9,52
140 VOL5:POKE1343,1
150 PRINT "CLEAR WHITE DOWN4 RIGHT5" 3,DOWN LEF
T4)+33(DOWN LEFT3)+1.(RIGHT3 UP3) 33,DOWN LEFT
4)+33-(DOWN LEFT4)+33-(DOWN LEFT4)+11."
152 PRINT "RIGHT17 UP4" 3,DOWN LEFT3)+33(DOWN L
EFT3)+1.(RIGHT4 UP) 33,DOWN LEFT4)+33-(DOWN LEF
T4)+33-(DOWN LEFT4)+11."
155 CHAR1,1,13,"(GREEN)OX_DOWN LEFT3)RY!(DOWN L
EFT3)SZ"+CHR$(34)
156 PRINTCHR$(27)+"0"
157 CHAR1,1,16,"T(CDOWN LEFT3)U*(DOWN LEFT3)VJ
X(DOWN LEFT3)W^&"
```

```
158 CHAR1,31,13,"X_XQ(DOWN LEFT4)Y!YR(DOWN LEFT4
)Z"+CHR$(34)
159 PRINTCHR$(27)+"0"
160 CHAR1,33,15,"ZS(DOWN LEFT4)C*IT(DOWN LEFT4)\
*U(DOWN LEFT4)1XJV(DOWN LEFT4)^&^N"
162 CHAR1,1,20,"(BRN RVSON SPACE3 RVSOFF D.BLUE)
NNN(BL.GRN)55555(D.BLUE)NNN(BL.GRN)55555(D.BLUE)
NNN(BL.GRN)55555(D.BLUE)NNN(BRN RVSON SPACE4 RV
SOFF)"
165 IH=13:SH=14:FH=18:FX=16:IX=22:B=8:T=1:F=3:F
R="(SPACE3 S-3)"
170 GOSUB600:GOSUB800:GOSUB200:GOSUB200
180 GETK$
185 IFK$="5"ORJOY(1)=7THEN400
190 IFK$="4"ORJOY(1)=7+128THEN410
195 IFK$="8"ORJOY(1)=3THEN420
200 IFK$="9"ORJOY(1)=3+128THEN430
210 IFK$="6"ORJOY(1)=7"ORJOY(1)=1THEN6000
215 POKE13312+14*8+1,PEEK(65282)
220 GOSUB7000
230 GOSUB8000
250 GOTO180
260 END
400 IFFX<4THENFX=4
405 GOSUB5040:FX=FX-1:GOSUB600:GOTO250
410 IFFX<8THENGOSUB5040:FX=4:GOSUB600:GOTO250
415 GOSUB5040:FX=FX-4:GOSUB600:GOTO250
420 IFFX>26THENFX=26
425 GOSUB5040:FX=FX+1:GOSUB600:GOTO250
430 IFFX>22THENGOSUB5040:FX=29:GOSUB600:GOTO250
435 GOSUB5040:FX=FX+4:GOSUB600:GOTO250
599 END
600 CHAR1,FX,FH,"(L.GRN)AC(DOWN LEFT2)BD"
610 IFPEEK((3072+FX+40*FH)+80)=14ANDPEEK((3072+FX
+40*FH)+81)=14THENGOSUB5000:GOTO10000
620 RETURN
5000 CHAR1,FX,FH,"(L.GRN)FJ(DOWN LEFT2 BLUE)OP"
5010 FORS=750TO980STEP10: SOUND3,S,0:NEXTS
5020 CHAR1,FX,FH,"(SPACE2 DOWN LEFT2 BLUE)OP"
5030 FORI=1TO100:NEXTI
5040 CHAR1,FX,FH,"(SPACE2 DOWN LEFT2 SPACE2)"
5050 RETURN
6000 CHAR1,FX,FH,"(L.GRN)FJ(DOWN LEFT2)BD"
6010 CHAR1,FX,SH+2,"(L.GRN)FJ(DOWN LEFT2)GK(DOWN
LEFT2)HL(DOWN LEFT2)IM"
6020 CHAR1,FX,SH,"(L.GRN)FJ(DOWN LEFT2)GK(DOWN L
EFT2)HL(DOWN LEFT2)IM(DOWN LEFT2 SPACE2 DOWN LEF
T2 SPACE2)"
6025 SOUND1,300,3
6030 IFPEEK((3072+FX+40*FH)-39)=50RPEEK((3072+FX
+40*FH)-40)=5THENGOSUB20000
6040 CHAR1,FX,SH,"(L.GRN SPACE2 DOWN LEFT2 SPACE
2 DOWN LEFT2 SPACE2 DOWN LEFT2 SPACE2 DOWN LEFT2
)AC(DOWN LEFT2)BD"
6050 GOTO250
7000 CHAR1,IX,IH,"(PURPLE)E"
7010 ONINT(RND(1)*SG)+1GOTO7011,7012,7013,7014,7
015
7011 GOTO7016
7012 RETURN
7013 RETURN
7014 RETURN
7015 RETURN
7016 GOSUB7100:IX=IX+INT(RND(1)*3)-1
7020 IFIX<4THENIX=30
7025 IFIX>30THENIX=4
7030 CHAR1,IX,IH,"(PURPLE)E"
7035 SOUND3,400,0
7040 RETURN
7100 CHAR1,IX,IH,"(SPACE)"
7110 RETURN
8000 IFTI<80=SGTHENRETURN
8010 B=8-1:IFB=0THEN30000
8020 ONBOTO8022,8024,8026,8028,8030,8032,8034,8
036
8021 GOTO8040
8022 RESTORE8050
8023 GOTO8040
8024 RESTORE8051
8025 GOTO8040
8026 RESTORE8052
8027 GOTO8040
8028 RESTORE8053
8029 GOTO8040
8030 RESTORE8054
8031 GOTO8040
```

```
<196>
<224>
<60>
<57>
<102>
<202>
<87>
<231>
<47>
<237>
<73>
<9>
<57>
<23>
<35>
<77>
<133>
<94>
<131>
<57>
<180>
<65>
<139>
<191>
<188>
<217>
<191>
<89>
<252>
<28>
<77>
<50>
<17>
<117>
<91>
<131>
<215>
<122>
<8>
<42>
<73>
<5>
<230>
<16>
<215>
<13>
<14>
<15>
<16>
<105>
<117>
<181>
<4>
<238>
<41>
<152>
<111>
<88>
<66>
<81>
<189>
<197>
<191>
<204>
<193>
<211>
<195>
<218>
<197>
<225>
<199>
8032 RESTORE8055
8033 GOTO8040
8034 RESTORE8056
8035 GOTO8040
8036 RESTORE8057
8037 GOTO8040
8038 RESTORE8058
8040 FORI=0TO7:READA:POKE13312+53*8+I,A:NEXT
8045 SOUND2,900,3
8050 DATA0,0,0,0,0,0,0,0
8051 DATA255,0,0,0,0,0,0,0
8052 DATA255,255,0,0,0,0,0,0
8053 DATA255,255,255,0,0,0,0,0
8054 DATA255,255,255,255,0,0,0,0
8055 DATA255,255,255,255,255,0,0,0
8056 DATA255,255,255,255,255,255,0,0
8057 DATA255,255,255,255,255,255,255,0
8058 DATA255,255,255,255,255,255,255,255
8060 T=TI:RETURN
10000 FORI=1TO200:NEXT
10010 FORI=100TO0STEP-20: SOUND1,1,1:NEXTI
10020 F=F-1:IFF=0THENGOSUB20020:GOTO30020
10025 FX=16:GOSUB600
10030 GOSUB20020:RETURN
20000 GOSUB7100:B=8:GOSUB8038
20010 SC=SC+1:IX=INT(RND(1)*22)+4
20020 CHAR1,0,0,"AC(DOWN LEFT2)BD(SPACE)" +MID$(
FR$,F+1,3)
20030 CHAR1,20,1,"(RVSON RED SPACES RVSOFF WHITE
SPACE)=(SPACE RED)55555(RVSOFF)"
20040 RETURN
30000 CHAR1,1,20,"(BRN RVSON SPACE3 RVSOFF D.BLU
E)NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
RVSON SPACE4 R
VSOFF)"
30002 CHAR1,IX,IH,"(PURPLE FLASHON)E(FLASHOFF)"
30003 FORI=1TO300:NEXTI
30004 CHAR1,IX,IH,"(SPACE)"
30010 GOSUB5000
30020 FORI=1TO200
30025 POKE13312+14*8+1,PEEK(65282)
30030 NEXT
30035 COLOR1,2,2
30040 FORI=17TO7STEP-1
30045 CHAR1,FX,I,"AC(DOWN LEFT2)BD(DOWN LEFT2 SP
ACE2)"
30050 NEXT
30055 CHAR1,FX,I,"(SPACE2 DOWN LEFT2 SPACE2 DOWN
LEFT2 SPACE2)"
30060 PRINT "WHITE CLEAR":POKE65299,56
30070 PRINT "RIGHT2 DOWN2 SD)ER(SPACE SF)ROSC(S
PACE)TAT(SPACE)SEIN(SPACE SB)ESTES(SPACE)!"
30071 IFSC=0THENPRINT "DOWN2 RIGHT2 SA)BER(SPACE
)ER(SPACE)WAR(SPACE)ERFOLGLOS(SPACE)":GOTO30000
30075 PRINT "DOWN2 RIGHT2 SE)S(SPACE)GELANG(SPA
CE)IHM(SPACE),(SPACE)IMMERHIN(SPACE)"SC"(RIGHT11
DOWN2 SZ)AUBERFLIEGEN(SPACE)ZU(SPACE)VERNICHEN(S
PACE)!"
30080 PRINT "RIGHT4 DOWN3)"F=100+SC"(SPACE SP)UN
KTE(SPACE)"
30090 INPUT "DOWN2 RIGHT2 SN)OCHMAL(SPACE) (JA/NE
IN) (SPACE)":K$
30100 IFK$="JA"THENRUN105
30110 PRINT "RIGHTS DOWN3 RIGHT4 SE)NDE(BLACK)":
END
40000 PRINT "CLEAR DOWN5"
40002 PRINT "C+5 SPACE C+4 SPACE3 C+3 SPACE3 C+3
SPACE3 C+3 SPACE3 C+3 SPACE3 C+3"
40004 PRINT "C+ SPACES C+ SPACES C+ SPACE C+ SPA
CES C+ SPACE C+ SPACES C+ SPACE6 C+ SPACE3 C+)"
40006 PRINT "C+3 SHIFTSPEACES C+4 SPACE2 C+ SPACE
3 C+ SPACE C+ SPACE2 C+2 SPACE C+ SPACE2 C+2 SPA
CES C+ SPACE C+ SPACE)"
40008 PRINT "C+ SHIFTSPEACES C+ SPACE C+ SPACE3 C
+ SPACE3 C+ SPACE C+ SPACE3 C+ SPACE C+ SPACE3 C
+ SPACE4 C+ SPACE2)"
40010 PRINT "C+ SHIFTSPEACES C+ SHIFTSPEACES C+ SH
IFTSPEACES C+3 SHIFTSPEACES C+3 SPACE3 C+3 SPACES
C+ SPACE2)"
40015 GETKEY$
40020 PRINT "DOWN5)EIN(SPACE)SCHWARM(SPACE)BOESA
RTIGER(SPACE)ZAUBERFLIEGEN"
40022 PRINT "DOWN)IST(SPACE)AUSGEBROCHEN(SPACE).
(SPACE)SIE(SPACE)ZAUBERN(SPACE)DEM(SPACE)"
40024 PRINT "DOWN)FROSCH(SPACE)MITTEN(SPACE)IN(S
PACE)DER(SPACE)NACHT(SPACE)SEINE(SPACE)SEE-"
40026 PRINT "DOWN)ROSENBLAETTER(SPACE)WEG(SPACE)
```



```

, (SPACE)UM (SPACE) IHN (SPACE) ZU (SPACE) AERGERN"
40030 BETKEYA$
40032 PRINT " (CLEAR) DIE (SPACE) FLIEGEN (SPACE) MUESS
EN (SPACE) MIT (SPACE) DEN (SPACE) TABTEN"
40034 PRINT "4= (SPACE) 4 (SPACE) SCHRITTE (SPACE) (SP
RUNG) (SPACE) LINKS"
40035 PRINT "5= (SPACE) 1 (SPACE) SCHRITT (SPACE) 2 LIN
KS"
40036 PRINT "8= (SPACE) 1 (SPACE) SCHRITT (SPACE) 2 REC
HTS"
40037 PRINT "9= (SPACE) 4 (SPACE) SCHRITTE (SPACE) (SP
RUNG) (SPACE) RECHTS"
40038 PRINT "6/7= (SPACE) 2 SPRUNG (SPACE) NACH (SPACE)

```

MEMORY DUMP

Das Maschinenprogramm "MEMORY DUMP" ermöglicht es, die genaue Speicherbelegung des Basic-Speichers zu erfahren. Das Programm zeigt folgendes an:

PRG: Länge des Programmes
VAR: Variablen
ARR: Arrays
STR: Strings
SPE: Basic-Speicher
FRE: Bytes free

Es wird aufgerufen durch "SYS 1630".

Das Maschinenprogramm selbst ist 137 Bytes lang und belegt den Bereich von 1630-1767 (\$065E-\$06E7). Außerdem werden noch die Zeropagestellen 208-212 belegt. Das Ablegen des Programmes in diesem Bereich hat den Vorteil, daß kein Basic-Speicher verloren geht, denn Sinn der Routine ist es, Speicherplatz zu sparen.

Es sind zwar schon mehrere ähnliche Routinen veröffentlicht worden, aber diese Routine hat meiner Meinung nach folgende Vorteile:

- man erfährt die Speicherbelegung im einzelnen,
 - es wird kein Basic-Speicher belegt,
 - die Routine ist in Grafik und in allen Ausbaustufen (16-64KB) verwendbar
 - die Routine ist extrem kurz (137 Bytes)
 - sie ist sowohl für Kassette als auch für Diskette verwendbar.
- Im Basicloader ist auch eine Save-Routine für das Maschinenprogramm integriert. Wenn der Speicher völlig leer ist, zeigt die Routine bei "PGR" 2 Bytes an. Dies ist jedoch völlig normal. Es bleibt noch zu erwähnen, daß die Routine mit "Genesis" erstellt wurde.

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

```

10 REM *****
20 REM * MEMORY - DUMP *
30 REM *
40 REM * <C> 1987 BY ASTERIX-SOFT *
50 REM *
60 REM * WRITTEN BY *
70 REM *
80 REM * MANFRED ZIMMERMANN *
90 REM * ROTHERMELSTR. 24 *
100 REM * 6976 KOENIGHEIM *
110 REM *
120 REM * BELEGT WIRD DER SPEICHER-
130 REM *
140 REM * BEREICH $065E-$06E7.
150 REM *
160 REM * AUFRUF DURCH SYS 1630
170 REM *
180 REM * PRG - PROGRAMM
190 REM * VAR - VARIABLEN
200 REM * ARR - ARRAYS
210 REM * STR - STRINGS
220 REM * SPE - BASIC-SPEICHER

```

```

<109> FLIEGE"
<224> 40040 PRINT " (DOWN2 RIGHT2) ODER (DOWN2 RIGHT2) "
<83> 40042 PRINT "JOYSTICK (SPACE) "
<236> 40044 PRINT " (DOWN) FEUER+LINKS (SPACE) 2, (SPACE) 2 LI
NKS (SPACE) 2 "
<180> 40046 PRINT " (DOWN SPACE) HOCH (SPACE) "
<0> 40048 PRINT " (DOWN) FEUER+RECHTS (SPACE) 2, (SPACE) RE
CHTS "
<200> 40050 PRINT " (DOWN) GEFANGEN (SPACE) WERDEN (SPACE) . "
40052 PRINT "DER (SPACE) FROSCH (SPACE) DARF (SPACE) NI
CHT (SPACE) INS (SPACE) WASSER (SPACE) FALLEN! "
ENDE DES LISTINGS

```

```

230 REM * FRE - BYTES FREE *
240 REM *****
250 :
260 DATA 2,06,06,03,02,01,00,06, 1034
270 DATA 06,00,04,01,02,09,00,06, 1228
280 DATA 06,04,05,09,00,02,02, 1115
290 DATA 06,00,01,00,02,02,0F, 1297
300 DATA 06,00,04,00,0F,06,00,06, 1518
310 DATA 06,06,00,00,00,00,01, 1421
320 DATA 04,0A,0E,04,06,0E,0E,0A, 1239
330 DATA 06,0B,01,04,0A,0E,0A,06, 1081
340 DATA 0E,0E,00,06,10,0A,33,0E, 1026
350 DATA 0A,0E,31,0A,0A,34,0E,32, 1082
360 DATA 20,0F,04,06,04,06,04,06, 1341
370 DATA 03,0A,06,03,00,06,09,00, 1330
380 DATA 0C,02,0F,05,02,07,0A,06, 918
390 DATA 01,02,0A,01,02,02,0A,03, 575
400 DATA 04,02,0A,03,00,05,0A,06, 584
410 DATA 02,05,0A,02,02,0F,02,01, 438
420 DATA 2F,37,33,37,2B,33,31,33, 402
430 DATA 31,0F,0F,0F,0F,0F,0F,0F, 1834
440 :
450 FOR I=1630 TO 1767 STEP 8:P=0
460 FOR J=0 TO 7:READ A$:A=DEC(A$):P=P+A:POKE I+J,A:
NEXT J
470 READ:IF P<>0 THEN PRINT "PRUEF SUMMENFEHLER (SPACE
E) IN" PEEK (63) + PEEK (64) * 256:END
480 NEXT:PRINT "DATAS (SPACE) RICHTIG (SPACE) EINBELE
SEN (SPACE) !!! "
490 :
500 REM SAVE - ROUTINE
510 :
520 PRINT " (CLEAR DOWN SPACE) DAS (SPACE) PROGRAMM (S
PACE) WIRD (SPACE) ALS (SPACE) REINE (SPACE) MASCHINEN-
"
530 PRINT " (SPACE) ROUTINE (SPACE) ABGESPEICHERT. "
540 PRINT " (SPACE) BITTE (SPACE) KASSETTE/DISKETTE (S
PACE) EINLEGEN "
550 PRINT " (DOWN4 SPACE2) < (SPACE) T (SPACE) A (SPACE)
S (SPACE) T (SPACE) E (SPACE) > "
560 GETKEY K$
570 :
580 SYS 43115 "MEMORY (SPACE) DUMP",1:REM BEI DISK
...
590 POKE 38,94:POKE 39,6:POKE 2035,234:POKE 2036,6
600 POKE 2034,38:SYS 65496
610 PRINT:PRINT " (DOWN2 SPACE) DAS (SPACE) REINE (S
PACE) MASCHINENPROGRAMM (SPACE) MUSS "
620 PRINT " (SPACE) ABSOLUT (SPACE) GELADEN (SPACE) WER
DEN (SPACE) (LOAD) +CHR$ (34) +CHR$ (34) + " ,1,1)
630 END
ENDE DES LISTINGS

```

Hier ist er - unser Super-RAM-DISC-MONITOR

C=64

FCMON
ein Floppy - Computermontitor mit Miniassembler zum Abtippen.
Teil 1 (für Commodore 64 und Floppy VC 1541)

Als ich mit der Programmierung des FCMON's begann, war mir natürlich bewußt, daß mein Programm zumindest den Leistungsstandard erreichen mußte, den die derzeit auf dem Markt befindlichen Monitore aufweisen. Ich hatte schon mit vielen excellenten Monitoren gearbeitet und auch Diskmonitore waren mir nicht fremd. Was mir aber an allen Programmen mißfiel war, daß sie alle nur auf ein Gerät konzipiert waren. Der Maschinensprachemonitor ließ die Floppy links liegen und umgekehrt. Meine Idee war es nun, ein Maschinensprache "tool" zu schaffen, welches Computer und Floppy als Einheit betrachtet und beide Devices gleichermaßen nutzt.

Was kann FCMON?

In diesem Heft wird Teil 1 des Programms veröffentlicht. Hier werden "nur" die Befehle "A" - Assemble, "D" - Disassemble und "K" - Quit implementiert. Der komplette FCMON wird dann vom Directory ohne Programmverlust über READ/WRITE-Block bis zu Verschiebe- und Suchbefehlen vieles bieten, was sich das Programmiererherz so wünscht. Mit ihm werden Sie dann ein Werkzeug in den Händen haben, mit dem Sie Computer und Floppy auf Maschinensprachebene mühelos ansprechen können.

Wenn Sie jedoch die Länge des ersten Teils (2700 Bytes) erschrecken sollte, so kann ich Sie so fern beruhigen, daß in diesem Teil alle benötigten Ein- und Ausgaberroutinen integriert sind. Die folgenden Erweiterungen können dann auf diese Routinen zurückgreifen und werden daher von essentieller Kürze sein; d. h. viele Befehle auf wenig Bytes.

Hinweise zum Abtippen

FCMON wurde in 100 % Maschinensprache geschrieben. Der abzutippende Basic-Data-Loader wird nur einmal benutzt. Nach einem fehlerfreiem Durchlauf und einem Tastendruck von Ihnen wird FCMON auf Diskette abgespeichert. Sollte jedoch ein Datafehler auftreten, so wird dies durch eine Prüfsumme festgestellt. Der Computer meldet Ihnen dann die Prüfsummen Differenz und den Bereich, in dem der Fehler auftrat. Durch Vergleich mit dem Listing müssen Sie dann das falsche Datum korrigieren.

Ist FCMON als Maschinensprache auf Diskette, speichern Sie sicherheitshalber das Basiclisting auch ab, denn versteckte Fehler (z. B. Vertauschungen) werden nicht erkannt und müssen nachkorrigiert werden.

Laden und Starten von FCMON

Das Programm belegt nun 11 Blöcke auf Diskette. Da er in Maschinensprache erstellt wurde, muß man FCMON absolut laden. (= "LOAD

FCMON TEIL 1 "8,1"

Nach dem Laden empfiehlt es sich "NEW" einzugeben, damit diverse Basicvektoren zurückgesetzt werden. Ansonsten können Sie öfters völlig unbegründet "? OUT OF MEMORY ERROR" erhalten. FCMON liegt ab Adresse 28672 (= \$7000) im Speicher. Starten wir ihn also mit "SYS 28672". Die Rahmenfarbe wechselt auf blau, die Zeichenfarbe auf gelb. Mit der ausgegebenen Startmeldung erhalten wir Gewißheit, daß wir uns im Monitor befinden.

Die Befehle des FCMON

Der eigentliche FCMON ist als Monitor konzipiert. Dieser erste Teil ist gengenommen "nur" eine Erweiterung, ein Miniassembler mit den Kommandos "Assemble" und Dis- (= Un-) assemble. Für die Bedienung des Assemblers sind Kenntnisse in Maschinensprache Voraussetzung. Für die etwas unerfahrenen Programmierer unter Ihnen ist jedoch FCMON nicht ganz unbrauchbar, denn auch ich habe durch spielerischen Umgang mit einem abgetippten Monitor die ersten Maschinenspracheschritte gemacht. Auch FCMON bietet die Gelegenheit, Maschinensprache "spielend" zu erlernen und ist ein geeignetes Gerät für kleinere und größere "Experimente".

Assemble

erlaubt die Erstellung von Programmen in Maschinensprache. Eingeben werden die verschiedenen Befehle in sogenannten "Memmonics", das sind allgemeingültige, dreistellige Buchstabenkombinationen. Diese werden dann von FCMON in die entsprechenden Operationscodes umgewandelt. In Wahrheit startet der Befehl "Assemble" nur eine Eingabeschleife. Die Verwaltung und Verarbeitung übernimmt eine andere Routine.

So startet zum Beispiel

AC 5000

die Eingabeschleife ab \$5000 (= Dez. 20480). Der größte Teil der Befehle FCMON's sind zweistellig. Der erste Buchstabe definiert den Befehl (hier: A = Assemble). Da aber der Monitor sowohl für Computer als auch für die Floppy anwendbar ist, muß eine zweite Eingabe bestimmen, welches Gerät angesprochen wird. Dies übernimmt hier der auf den Befehl folgende Buchstabe. "C" steht für Computer, "F" für Floppy. Der oben angeführte Befehl lautet also genau: ASSEMBLE Ab \$5000 in den Computer.

AF0500 dasselbe ab \$5000 in das Floppy-RAM.

Probieren wir es doch einfach aus!

Eingabe: AC5000 (RETURN)

Eine Zeile tiefer wird die Adresse noch einmal ausgegeben und ein blinkender Cursor ermuntert uns zur Eingabe. Wir können jetzt mit der Maschinenspracheprogrammierung in Form von Memmonics beginnen. Alle Eingaben erfolgen normalerweise im hexadezimalen Zahlenformat. FCMON bietet aber auch andere Möglichkeiten der Eingabe. Bei direkter Adressierung sind zum Beispiel folgende Kombinationen

erlaubt:

```
..LDA #0F
  Lade den Akku mit $0F = dez. 15.
..LDY #A
  Lade Y-Register mit dem ASCII-Code von "A". Ein zweites Anführungszeichen ist nicht erforderlich.
..LDX # %01000010
  Lade X-Register mit der Binärzahl %01000010 = $42 = dez. 66.
```

Oft ist es erforderlich, daß beim Assemblerprogrammieren nicht Memmonics sondern einzelne Bytes eingegeben werden können. FCMON akzeptiert dies, wenn dem Byte ein "." (Punkt) vorangestellt wird. z. B.

```
.03
  speichert Byte $03 ab.
.%01001111
das Ganze funktioniert auch binär
..TEXT bis 32 Zeichen"
  und mit ganzen Texten. Hier ist ein abschließendes "Gänsefüßchen" erforderlich.
```

Ansonsten erfolgt die Eingabe der Memmonics nach der üblichen Norm. FCMON akzeptiert alle für den Prozessor MOS6510 mögliche Adressierungsarten. Nur für die Befehle "ASL", "ROR", "ROL", "LSR" mußte eine Ausnahmeregelung getroffen werden. Üblicherweise wird, wenn sich der Befehl auf den Akku beziehen soll, ein "A" nachgestellt. FCMON erkennt nur ein "*" (Joker) an. Dies ist jedoch die einzige Abweichung.

Labels

FCMON besitzt einen Direktassembler, d. h., der Befehl wird vom Bildschirm übernommen, umgerechnet, und dann direkt in eine Speicherstelle geschrieben. Dies hat zum einen Vorteile (weniger Speicherbedarf, man kann sein Programm sofort ohne extra Assembliervorgang ausprobieren), doch leider auch Nachteile. Insbesondere bei Sprungzeilen, die weiter hinten im Programm liegen und die bei der Eingabe des Sprungbefehls ("JMP", "BNE", ...) noch nicht feststehen, kann es Schwierigkeiten geben. Deshalb wurden bei FCMON "Labels" eingeführt. Labels sind Scheinadressen (Variablen), die erst gesetzt werden, wenn das Sprungziel feststeht. Bei FCMON werden diese durch den Buchstaben "M" gekennzeichnet. Zulässig sind Labels von "MOO" bis "M1F", also deren 32.

"Labels müssen immer erst geöffnet und dann geschlossen werden." (FCMON-Spezialität) z. B.

```
..BNE MO1
  Trifft der Computer auf diese Befehlssequenz, so legt er im Kassettenspeicher eine Tabelle an, die die aktuelle Adresse enthält. Gleichzeitig wird das Label (MO1) in einer anderen Tabelle als belegt gekennzeichnet.
```

Das Label ist "geöffnet".
Taucht irgendwann bei einer weiteren Eingabe der Befehl z. B.

```
.. MO1 RTS
  auf, so wird die Adresse aus der Tabelle geholt und auf das Sprungziel gesetzt. Anschließend wird das Label wieder als "frei" gekennzeichnet ("geschlossen"). Wird versucht, ein Label zweimal zu öffnen oder ein nicht geöffnetes zu schließen, antwortet FCMON mit einem "?", seiner Standardfehlermeldung.
```

Labels stehen immer stellvertretend für 3-Bytes-Befehle (außer bei Branch-Befehlen). Es sind neben "JMP MOO" somit auch Sequenzen wie z. B. "LDA MO1,X" oder "LDY MOF,X" möglich.

Labels erleichtern die Direktprogrammierung natürlich erheblich. Doch Vorsicht! Vergessen Sie nie, alle Labels zu schließen, bevor Sie

aus der Eingabeschleife aussteigen. (Dies geschieht übrigens mit "X"). Die gesetzten Scheinadressen führen sonst beim Probelauf mit 80 %iger Sicherheit zum Absturz!

Das FCMON beim Treffen auf den Buchstaben "M" nur mehr an Labels "denkt", sind einige skurrile Kombinationen möglich, z. B.

```
..BNE MO1
..MO1 JMP MO1
..MO1 RTS
```

Uns interessiert hier nur die mittlere, etwas seltsam anmutende Zeile. Hier wird durch das Setzen der Labels nicht etwa eine Endlosschleife fabriziert, sondern folgendes:

Der Computer beginnt links außen. Er trifft auf das Label "MO1", setzt die Adresse auf den vorliegenden "BNE"-Befehl und schließt es zugleich. Danach wird der Befehl "JMP" erkannt. Das Label "MO1", das vorher geschlossen wurde, wird wieder geöffnet. Diesmal auf eine ganz andere Adresse, welche zur Zeit noch nicht feststeht. Schließlich wird in der letzten Zeile des Label entgültig geschlossen. Dies ermöglicht während des Programmierens eine optimale Ausnutzung der Labels. Mehrfachbenutzung ist besser als mehr Speicherplatz für mehr Labels!

Da ein Label nur einmal "geschlossen" werden kann, ist es unmöglich, mehrmals auf eine Label-Adresse zu springen. Um dies zu gestatten, wurde die Schließroutine rekursiv angelegt, d. h. sie ruft sich selbst solange auf, bis sie auf einen Maschinensprachebefehl in der Eingabezeile stößt. z. B.

```
.. BNE MO1
.. JMP MO4 sollen sich auf dieselbe Adresse beziehen.
```

Die Lösung:
.. MO1 MO4 RTS

Der Computer beginnt wieder von links außen. Er schließt zuerst das Label "MO1" und ruft sich selbst wieder auf. Anschließend trifft er auf "MO4" und schließt auch dieses Label. Zum Schluß wird noch der Befehl "RTS" erkannt und die Zeile wird verlassen. Sie können in einer Zeile so viele Labels hintereinandersetzen, wie Sie Platz haben (rund 10).

Da der Computer immer von links beginnt, kann es, wenn weiter rechts in der Zeile ein Tippfehler auftritt, zu Problemen kommen. FCMON markiert die Fehlerstelle mit einem Fragezeichen und gibt die aktuelle Adresse eine Zeile tiefer neu aus. Nun brauchen Sie nicht mehr die gesamte Zeile einzugeben. Vielmehr dürfen Sie die Labels, die links von der Fehlerstelle stehen und schon abgearbeitet wurden, nicht mehr anführen. Sie sind schon auf die Zeilenadresse gesetzt worden. Geben Sie die dennoch ein, erfolgt eine Fehlermeldung, da versucht wurde, ein "freies" Label zu schließen, z. B.
(Nehmen wir an, alle Labels wurden schon vorher eröffnet.)

```
..MO1 MO3 MOF M1F BRL
```

Statt "BRK" wurde "BRL" geschrieben, FCMON quittiert dies mit einem "?". Bei der korrigierten Eingabe eine Zeile tiefer brauchen Sie nur mehr "BRK" tippen, da alle Adressen schon gesetzt worden sind.

So, das waren die Labels. Etwas zuviel für den Anfang? Ich schlage vor, Sie probieren das Ganze erst einmal aus. Der Assemble-Befehl ist nun schon fast komplett erklärt. Eine Sonderfunktion ist noch zu erwähnen. Oft ist es anschaulich, Schleifen, Programmteile ... am Bildschirm optisch zu trennen. Dies geschieht bei FCMON indem Sie ein "-" (Minus) anstatt eines Befehls eingeben. Der

Computer gibt dann eine Trennzeile aus. Sie wird nicht abgespeichert und dient nur zur temporären Trennung am Bildschirm.

DISASSEMBLE

ist die Umkehrfunktion des eben besprochenen Befehls. Ein im Speicher befindliches Maschinenprogramm wird in Memmonics übersetzt und auf den Bildschirm ausgegeben. Es sind mehrere Ausgabeformate möglich.

DC 5000

Disassembliert den Speicherbereich ab \$5000 im Computer. Auflisten erfolgt mit der (f7)-Taste, aussteigen aus dem Listvorgang mit (RUN/STOP).

DC 5000 5500

Übersetzt den Speicherbereich von \$5000 bis \$5500. Das Listen erfolgt automatisch, kann aber mit (Control) verlangsamt werden. Bei Erreichen der Endadresse (\$5500) hält der Computer automatisch an. Aussteigen während der Ausgabe mit (RUN/STOP).

DC 5000 P

Seitenweise Disassemblierung des Speichers. Es werden immer 17 Zeilen ausgegeben, danach wartet der Computer auf einen (f7)-Druck von Ihnen. Aussteigen wieder mit (RUN/STOP).

Das Ganze gilt natürlich analog für die Floppy (z. B. DF C100).

Zur besseren Übersicht wird nach jedem "JMP"-Befehl (auch BRK, RTS ...) eine Trennzeile ausgegeben. Änderungen erfolgen durch einfaches Überschreiben der Befehle, nicht der rechts ausgegebenen Bytes. Sie müssen aber immer darauf achten, daß der korrigierte Befehl dieselbe Länge (in Bytes) hat, wie der vorhergehende. Sonst wird ein Teil Ihres Maschinenprogramms überschrieben.

Trifft FCMON beim Disassemblieren auf Codes, die keinem Assemblerbefehl entsprechen, werden drei Fragezeichen "???" ausgegeben. Es handelt sich hierbei meistens um Texte oder Tabellen u. ä.

Sie werden sich sicher fragen: "Jetzt kann ich Maschinenprogramme erstellen und ansehen, doch wie bringe ich sie zum Laufen?". Hier müssen Sie sich bis zur Veröffentlichung des nächsten Teils mit einer Notlösung begnügen. Um das Programm zu starten, müssen Sie aus FCMON aussteigen, die Hexadezimaladresse in eine Dezimalzahl umrechnen und schließlich mit "SYS" in Ihr Programm einspringen. Die Umrechnung erfolgt in der Art, daß Sie die erste Ziffer der Hex-Zahl mit 4096 (16 hoch 3) multiplizieren. Hierzu wird das Produkt aus zweiter Ziffer mal 256 (16 hoch 2) addiert. Dasselbe geschieht mit der dritten Ziffer mal 16 (16 hoch 01). Die vierte Ziffer wird mit 1 (16 hoch 0) multipliziert, also einfach dazugezählt.
Aus z. B. \$2223 erhalten Sie dann dez. 8739 (8192 + 512 + 32 + 3)

FCMON wurde absichtlich in den Bereich \$7000 - \$8000 (dez. 28672 - 32768) gelegt, um den Bereich \$C000 - \$CFFF (dez. 49152 - 53247) für selbstgestellte Programme frei zu lassen. Der Monitor liegt somit mitten im Basic-Speicher. Um ihn vor Überschreibung zu schützen, können Sie ihn entweder nur mit "kurzen" Basicprogrammen benutzen, oder zwei Pokes + "NEW" nach dem Laden eingeben, die das Basic heruntersetzen. Natürlich haben Sie dann weniger "Bytes free". Die Eingabe lautet: "Poke 55,255:Poke 56,111:NEW". Das Basicende ist nun auf dez. 28671 heruntergesetzt, der freie Basic-Speicherplatz beträgt nun mehr 26620 Bytes, FCMON ist vor Basicprogrammen geschützt.

Doch bis dahin ist noch viel Zeit. Nützen Sie sie, indem Sie sich mit dem Monitor so gut wie möglich vertraut machen. FCMON ist vor allem für jene Leute gedacht, die sich zwar halbwegs in Maschinensprache auskennen, bisher aber keine eigenen Programme geschrieben, da sie den Aufwand, jeden einzelnen Befehl "per Hand" in den Opcode umzurechnen und in den Speicher zu poken scheuten. Und wenn Sie sich in Maschinensprache noch nicht so gut auskennen, so kann ich Ihnen nur raten: Üben, üben, üben ... Es ist bisher noch kein Programmierprofi vom Himmel gefallen.

"FCMON TEIL 2" wird Ihnen dann noch eine Menge weitere Befehle zur Verfügung stellen. Es ist also nur zu Ihrem Vorteil, wenn Sie bis zur nächsten Veröffentlichung den derzeitigen Befehlssatz beherrschen.

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

10 REM		<9>
12 REM	FLOPPY - COMPUTER MONITOR	<87>
13 REM		<16>
14 REM	FUER COMMODORE 64 + VC 1541	<170>
16 REM		<15>
18 REM		<21>
20 REM	(C) 1986 BY JOHANN STRAUSS	<94>
22 REM	OSTRIACH 102	<149>
24 REM	A-9570 OESTRICH	<94>
26 REM	TEL. 04243/8138	<88>
28 REM		<27>
30		<88>
32 REM	TEIL 1: SPEICHER \$7000 - \$7AB0	<218>
34 REM	BEFEHLE 'A' + 'D' + 'X'	<83>
36		<94>
38 DIM	B(26):FORT = 0 TO 26: READ B(T):NEXT	<83>
39		<97>
40 REM	EINLESESCHLEIFE: PRO 100 BYTES	<234>
42 REM	WIRD EINE PRUEFSUMME ERMITTELT	<183>
44 REM		<47>
46		<104>
50 FORT	= 0 TO 27	<18>
52 PZ=0	: PRINT	<18>
54		<112>
60	: FORT = 0 TO 99	<163>
62	: READ A: IFA=-1 THEN 100	<174>
64	: PZ=PZ+A:POKE 28672+100*T+TT,A	<193>
66	: NEXT	<128>
68		<126>
70 IF	PZ=PS(T) THEN NEXT	<14>
72		<130>
80 REM	FEHLERBEHANDLUNG	<160>
82		<140>
84 PRINT	"(CLEAR DOWN)PRUEFSUMMENFEHLER(SPACE)!"	<11>
86 PRINT	"(DOWN SPACE)BEREICH(T*100)(SPACE)-(T+1)	<79>
88	: *100	<239>
90 PRINT	"(DOWN)DIFF.:PS(T)-PZ	<146>
92		<232>
94		<181>
100 IF	PZ<> 4223 THEN 84	<232>
102		<152>
110 REM	ALLES O.K.	<45>
112		<160>
113 PRINT	"O.K.":PRINT"(DOWN)BITTE(SPACE)DISKETTE	<230>
114	: (SPACE)EINLEGEN(SPACE)!"	<170>
115 REM	SAVE PROGRAMM	<5>
116		<172>
120 T=0		<181>
122 READ	A: IFA=-1 THEN 130	<174>
124 POKE	32768+T,A: T=T+1: GOTO 122	<168>
126		<61>
130 POKE	198,0:WAIT 198,1	<184>
132		<16>
134 SYS	32768	<190>
136		<90>
140 PRINT	"(DOWN SPACE)SAVE(SPACE)"CHR\$(34)"FCMON	<194>
142	: (SPACE)TL1.DAT"CHR\$(34)"(SPACE)": END	<241>
150		<208>
1000		<37>
1002 REM	PRUEFSUMMEN	<166>

programme

```

1004 :
1010 DATA 7462,5891,7201,8131,15705,11920,10332,
10698,12102,11326,11499,11109
1012 DATA 10422,12103,12892,12743,12997,13483,12
037,12161,11293,13039,11773
1014 DATA 10966,11454,14456,9720
1020 :
30000 :
30002 REM DATA,DATAS,DATAS,.....
32000 DATA160,7,140,134,2,136,140,32,200,140,33,
200,32,68,229,169,35,160,112
32001 DATA32,30,171,169,241,141,22,3,169,113,141
,23,3,76,241,113,17,32,32,42
32002 DATA42,42,42,32,70,76,79,80,80,89,32,45,32
,67,79,77,80,85,84,69,82,32,77
32003 DATA79,78,73,84,79,82,32,42,42,42,42,13,17
,32,32,42,42,42,32,32,32,32,32
32004 DATA32,32,32,66,89,32,74,46,83,84,82,65,85
,83,83,32,77,32,32,32,32,32,32
32005 DATA42,42,42,42,13,13,0,32,61,65,90,68,59,
77,58,75,39,85,63,42,86,76,83
32006 DATA71,84,69,79,35,64,82,87,66,70,36,88,95
,72,67,0,76,73,112,84,114,0,112
32007 DATA60,119,250,114,0,112,0,112,0,112,0,112
,0,112,0,112,0,112,0,112,0,112
32008 DATA0,112,0,112,0,112,0,112,0,112,0,112,0
,112,0,112,0,112,0,112,0,112,0
32009 DATA112,71,114,0,112,0,112,0,112,0,112,0,1
12,74,83,65,68,67,65,78,68,65
32010 DATA83,76,66,67,67,66,67,83,66,69,81,66,73
,84,16,77,73,66,78,69,66,80,76
32011 DATA66,82,75,66,86,67,66,86,83,67,76,67,67
,76,68,67,76,73,67,76,86,67,77
32012 DATA80,67,80,88,67,80,89,68,69,67,68,69,88
,68,69,89,69,79,82,73,78,67,73
32013 DATA78,88,73,78,89,74,77,80,74,83,82,76,68
,65,76,68,88,76,68,89,76,83,82
32014 DATA78,79,80,79,82,65,80,72,65,80,72,80,80
,76,65,80,76,80,82,79,76,82,79
32015 DATA82,82,84,73,82,84,83,83,66,67,83,69,67
,83,69,68,83,69,73,83,84,65,83
32016 DATA84,88,83,84,87,84,65,88,84,65,89,84,83
,88,84,88,65,84,88,83,84,89,65
32017 DATA94,94,94,97,33,2,144,176,240,36,48,208
,16,0,80,112,24,216,88,184,193
32018 DATA224,192,198,202,136,65,230,232,200,76,
32,161,162,160,66,234,1,72,0
32019 DATA104,40,34,98,64,96,225,56,248,120,129,
134,128,170,168,186,138,154,152
32020 DATA227,227,227,0,0,0,3,0,0,0,255,0,0,255,
255,255,255,227,19,19,35,255
32021 DATA255,227,35,255,255,223,247,227,67,67,0
3,255,227,255,255,255,255,83
32022 DATA83,255,255,227,255,255,255,227,99,99,2
55,255,255,255,255,255,169,255
32023 DATA141,138,2,169,8,133,2,169,0,133,208,32
,50,230,169,10,133,208,169,0
32024 DATA133,211,32,58,114,162,0,121,116,112,24
0,11,232,224,32,144,246,32,46
32025 DATA114,76,250,113,138,10,170,189,148,112,
133,254,189,149,112,133,255,108
32026 DATA254,0,169,79,160,114,32,30,171,169,0,1
33,208,96,32,50,230,201,32,240
32027 DATA249,96,169,13,76,210,255,169,14,141,32
,208,76,116,164,32,32,63,13,0
32028 DATA32,232,114,32,108,114,32,162,114,169,2
55,133,190,169,59,32,189,114
32029 DATA32,227,114,76,250,113,162,172,32,121,1
14,149,1,32,126,114,149,0,96
32030 DATA32,58,114,208,3,32,50,230,32,174,114,3
2,153,114,10,10,10,10,133,146
32031 DATA32,50,230,32,174,114,32,153,114,5,146,
96,201,58,144,2,105,8,41,15,96
32032 DATA32,58,114,201,13,240,248,104,104,76,24
,114,8,201,47,176,4,40,76,169
32033 DATA114,201,71,176,248,40,96,72,32,66,114,
104,32,210,255,162,172,181,1
32034 DATA32,206,114,181,0,72,74,74,74,32,217
,114,104,41,15,201,10,144,2,105
32035 DATA6,105,48,208,2,169,32,76,210,255,32,58
,114,160,0,132,150,201,67,240
32036 DATA6,201,70,208,178,133,150,96,32,108,114
,32,19,115,32,133,115,32,150
32037 DATA117,165,190,240,3,76,97,114,32,66,114,
76,250,113,32,27,119,240,107
32038 DATA201,45,240,4,201,46,208,3,198,211,96,2

```

<41>	01,77,208,3,76,16,118,201,88	<22>
<18>	32039 DATA208,12,169,0,133,190,32,66,114,104,104,76,250,113,141,64,3,32,50,230	<63>
<89>	32040 DATA141,65,3,32,50,230,141,66,3,162,0,169,214,141,88,115,169,112,141,89	<189>
<24>	32041 DATA115,160,0,185,39,113,217,64,3,208,18,200,192,3,144,243,189,129,113	<188>
<57>	32042 DATA141,147,2,189,185,113,141,148,2,96,160,3,238,88,115,208,3,238,89,115	<140>
<223>	32043 DATA136,208,245,32,224,56,144,210,176,108,32,27,119,240,39,201,35,240	<160>
<76>	32044 DATA61,201,45,240,25,201,40,208,3,76,188,116,201,42,240,110,201,46,240	<26>
<151>	32045 DATA13,201,77,208,3,76,211,117,76,71,116,7,6,62,116,76,31,116,172,148,2	<33>
<189>	32046 DATA208,208,58,32,54,116,173,147,2,76,0,116,32,111,118,76,217,115,32,50	<221>
<15>	32047 DATA230,208,14,32,58,114,201,37,240,238,201,34,240,240,32,129,114,141,87	<124>
<170>	32048 DATA3,32,77,117,173,148,2,201,67,240,30,201,19,240,26,173,148,2,201,227	<131>
<90>	32049 DATA240,3,76,251,116,169,8,133,146,173,148,2,45,147,2,5,146,141,86,3,96	<217>
<185>	32050 DATA32,21,116,169,0,240,235,32,21,116,32,5,4,116,169,8,208,225,169,227,141	<125>
<203>	32051 DATA148,2,96,169,28,208,215,32,58,114,201,34,240,21,201,37,208,6,32,111	<129>
<233>	32052 DATA118,76,51,116,32,129,114,141,86,3,169,1,133,192,96,76,205,117,160,0	<50>
<191>	32053 DATA132,211,132,192,76,176,117,174,148,2,24,255,240,163,32,129,114,133	<248>
<239>	32054 DATA192,32,27,119,240,79,201,44,240,93,32,129,114,32,66,117,32,27,119,240	<113>
<42>	32055 DATA6,201,44,240,28,208,132,174,148,2,208,3,76,124,117,32,21,116,173,147	<247>
<169>	32056 DATA22,201,32,240,4,169,12,208,88,169,0,240,84,32,21,116,32,158,116,208	<240>
<144>	32057 DATA4,169,28,208,72,173,147,2,201,162,240,10,169,24,208,61,32,58,114,201	<203>
<70>	32058 DATA88,96,76,27,116,32,174,116,169,4,208,4,5,165,192,141,87,3,32,77,117	<113>
<198>	32059 DATA76,21,116,76,32,117,32,126,114,133,192,32,58,114,201,44,240,62,201	<55>
<183>	32060 DATA41,240,18,174,147,2,224,76,208,40,32,1,29,114,32,66,117,169,32,76,246	<128>
<131>	32061 DATA115,32,58,114,201,44,208,22,32,58,114,201,89,208,15,165,192,141,87	<68>
<19>	32062 DATA3,32,21,116,32,77,117,169,16,208,224,1,04,104,32,46,114,32,22,119,76	<61>
<224>	32063 DATA97,114,32,58,114,201,88,208,238,32,58,114,201,41,208,231,165,192,141	<230>
<165>	32064 DATA87,3,32,77,117,169,0,240,187,32,174,116,32,158,116,208,4,169,20,208	<61>
<209>	32065 DATA175,173,147,2,201,162,240,11,201,134,240,2,208,194,169,231,141,148	<241>
<71>	32066 DATA2,169,20,208,153,141,87,3,165,192,141,88,3,169,3,44,169,2,133,192,96	<239>
<161>	32067 DATA173,87,3,56,229,172,173,88,3,229,173,144,35,173,87,3,56,229,172,176	<135>
<242>	32068 DATA2,233,255,56,233,2,48,37,141,87,3,32,77,117,169,255,141,148,2,169,0	<117>
<15>	32069 DATA96,32,82,117,76,246,115,165,172,56,237,87,3,133,192,56,169,255,229	<204>
<254>	32070 DATA192,233,1,48,219,76,251,116,165,192,240,17,141,83,3,160,0,185,86,3	<49>
<225>	32071 DATA32,255,118,200,206,83,3,208,244,169,0,133,192,96,160,39,169,45,32,210	<132>
<177>	32072 DATA255,136,208,248,96,160,0,32,50,230,201,34,240,8,32,255,118,200,192	<55>
<107>	32073 DATA32,208,241,96,32,187,117,76,171,117,165,150,208,180,32,126,114,201	<82>
<242>	32074 DATA32,176,181,168,185,192,3,208,51,173,148,2,240,12,32,98,116,169,3,32	<232>
<8>	32075 DATA128,118,133,192,208,13,173,147,2,141,86,3,169,2,32,128,118,133,192	<226>
<88>	32076 DATA152,10,168,165,172,153,128,3,165,173,153,129,3,96,32,126,114,201,32	<217>
<24>	32077 DATA144,	

<22>
 <63>
 <189>
 <188>
 <140>
 <160>
 <26>
 <33>
 <221>
 <124>
 <131>
 <217>
 <125>
 <129>
 <50>
 <248>
 <113>
 <247>
 <240>
 <203>
 <113>
 <55>
 <128>
 <68>
 <61>
 <230>
 <61>
 <241>
 <239>
 <135>
 <117>
 <204>
 <49>
 <132>
 <55>
 <82>
 <232>
 <226>
 <217>
 <68>
 <72>
 <110>
 <60>

2,76,67,118,162,8,32,50,230,201
32082 DATA9,38,146,202,208,246,165,146,96,169,0,
153,192,3,96,165,254,133,172
32083 DATA165,255,133,173,96,169,1,166,150,240,3,
2,72,169,87,141,226,118,104,166
32084 DATA172,164,173,141,229,118,140,228,118,14,
2,227,118,32,204,255,32,203,118
32085 DATA32,219,118,32,230,118,96,166,150,240,1,
6,32,210,255,206,229,118,208
32086 DATA243,32,204,255,169,15,76,195,255,129,1,
72,96,160,15,152,166,2,32,186
32087 DATA255,169,0,32,189,255,76,192,255,162,15,
76,201,255,77,45,82,224,3,50
32088 DATA162,224,160,118,169,6,133,164,132,86,1,
34,85,160,0,177,85,32,210,255
32089 DATA200,198,164,208,246,96,72,140,64,3,32,
141,118,104,32,180,118,172,64
32090 DATA3,162,172,246,0,208,2,246,1,96,169,145,
76,210,255,32,58,114,201,58
32091 DATA240,2,201,13,96,32,232,114,32,108,114,
32,86,119,240,65,32,30,119,240
32092 DATA60,198,211,162,252,76,110,114,32,37,11,
9,169,59,32,189,114,32,114,119
32093 DATA141,83,3,32,173,119,32,195,121,32,209,
121,76,63,119,32,58,114,160,255
32094 DATA132,252,132,253,200,132,247,201,80,208,
11,8,133,247,32,68,229,160,17
32095 DATA132,250,40,96,169,1,32,122,119,76,145,
119,166,150,240,243,141,229,118
32096 DATA169,82,141,226,118,32,154,118,32,204,2,
55,162,15,76,198,255,166,150
32097 DATA240,16,32,207,255,206,229,118,208,10,7,
2,32,192,118,104,76,13,119,161
32098 DATA172,76,13,119,76,78,120,32,179,119,76,
143,120,160,0,140,73,3,32,120
32099 DATA122,176,51,72,41,28,141,73,3,104,32,93,
120,176,55,201,44,240,221,201
32100 DATA150,240,90,201,108,240,58,201,142,240,
82,162,22,142,148,2,41,227,141
32101 DATA72,3,201,194,240,80,201,226,240,85,32,
93,120,176,16,169,63,141,64,3
32102 DATA141,65,3,141,66,3,169,255,141,148,2,32,
227,114,162,0,142,67,3,160,3
32103 DATA169,64,76,30,171,32,73,120,169,38,133,
164,32,32,120,169,1,141,148,2
32104 DATA208,224,160,3,177,164,153,63,3,136,208,
248,96,32,73,120,169,101,133
32105 DATA164,32,32,120,240,201,32,73,120,169,17,
133,164,208,242,32,73,120,169
32106 DATA29,133,164,208,233,169,113,133,165,96,
169,112,133,165,169,231,133,164
32107 DATA169,12,141,73,3,208,213,160,0,217,129,
113,240,7,200,192,56,144,246
32108 DATA24,96,141,147,2,185,185,113,141,148,2,
152,32,136,120,168,162,0,185
32109 DATA214,112,157,64,3,200,232,224,3,144,244,
96,133,146,10,24,101,146,96
32110 DATA32,227,114,172,148,2,240,61,200,240,49,
173,83,3,201,32,240,48,201,76
32111 DATA240,44,201,108,240,37,173,73,3,74,168

<19>	185,187,120,133,254,185,188,120	<19>
<62>	32112 DATA133,255,108,254,0,250,121,5,121,11,121,32,121,58,121,81,121,74,122	<141>
<213>	32113 DATA82,122,76,129,121,76,118,121,76,32,121,32,97,122,48,23,24,101,172,72	<28>
<236>	32114 DATA165,173,144,2,105,0,32,206,114,32,110,122,104,32,206,114,76,139,121	<175>
<126>	32115 DATA73,255,133,146,56,165,172,230,146,229,146,72,165,173,176,227,233,0	<146>
<209>	32116 DATA76,227,120,32,97,122,76,234,120,173,83,3,41,15,201,10,240,183,32,97	<94>
<106>	32117 DATA122,32,61,122,173,84,3,76,234,120,32,3,7,121,144,102,32,90,122,133,248	<44>
<111>	32118 DATA32,114,119,133,249,141,85,3,162,248,32,199,114,76,107,122,32,97,122	<75>
<255>	32119 DATA32,49,122,173,84,3,32,206,114,32,58,12,2,32,52,122,32,64,122,208,58	<183>
<116>	32120 DATA173,83,3,201,198,240,173,32,97,122,32,206,114,32,52,122,173,83,3,201	<43>
<23>	32121 DATA150,240,4,201,182,208,5,32,64,122,208,26,32,55,122,208,21,32,49,122	<131>
<120>	32122 DATA32,37,121,32,58,122,208,10,173,83,3,20,1,142,240,152,32,113,122,169	<242>
<203>	32123 DATA20,32,186,121,169,58,32,210,255,162,0,189,83,3,32,206,114,32,227,114	<1>
<172>	32124 DATA232,206,87,3,208,241,173,83,3,162,5,22,1,170,122,240,4,202,208,248,96	<71>
<35>	32125 DATA32,66,114,76,176,117,56,32,10,229,168,24,76,10,229,165,252,56,229,172	<77>
<161>	32126 DATA165,253,229,173,176,23,76,50,115,165,2,47,208,17,32,237,246,240,244	<116>
<126>	32127 DATA164,253,208,208,6,165,203,201,3,208,24,0,96,198,250,208,251,32,213,121	<12>
<28>	32128 DATA8,76,105,119,76,11,121,76,5,121,76,81,121,173,83,3,201,162,240,240	<50>
<203>	32129 DATA168,41,15,240,235,152,160,5,217,166,12,2,240,230,136,208,248,201,198	<47>
<185>	32130 DATA240,226,201,150,240,222,32,97,122,32,4,9,122,173,84,3,32,206,114,32	<118>
<23>	32131 DATA52,122,32,55,122,32,58,122,76,139,121,169,40,44,169,44,44,169,88,44	<57>
<80>	32132 DATA169,41,44,169,35,44,169,89,76,210,255,32,37,121,144,234,32,69,122,32	<112>
<204>	32133 DATA64,122,208,220,32,69,122,32,55,122,208,212,32,114,119,141,84,3,96,32	<83>
<6>	32134 DATA90,122,32,110,122,173,84,3,96,169,3,44,169,2,44,169,1,141,87,3,24,96	<186>
<20>	32135 DATA162,35,221,131,122,240,4,202,208,248,2,4,96,2,18,34,50,66,82,98,114	<17>
<35>	32136 DATA178,210,242,26,58,90,218,250,4,20,52,6,8,84,100,116,212,244,128,137	<246>
<126>	32137 DATA12,28,60,92,124,220,252,122,230,36,134,132,0,96,64,76,108,-1	<123>
<69>	33000	<162>
<83>	42000 DATA162,8,32,186,255,162,31,160,128,169,11,32,189,255,162,0,160,112,134	<190>
	42001 DATA251,132,252,169,251,162,176,160,122,76,216,255,70,67,77,79,78,32,84	<201>
	42002 DATA69,73,76,49,-1	<72>
	ENDE DES LISTINGS	

UPDATES

"Adress 16" in "Compute mit" 2/87

Die Änderungen für Kassettenbenutzer lauten

Zeile 2004 vollständig löschen.

Folgende Programmzeilen ändern

dern:

1008

7,1,0,
2222

2008
7.1.1

OPEN

OPEN

"Trampolin Games" in "Compute mit" 3/87

Der **POKE 44,99** muß bei den Startanweisungen noch vorangestellt werden. Durch schlechten Druck waren in einigen Exemplaren einige Werte nicht zu erkennen:

in Teil 2:

1013 DATA 50,00,FD,56,00,FF,70,00,FF,F0,00,3F,F0
 .00,35,C0,00,15,40,00. 1914

und Teil 3:

1033 DATA 67,83,60,AD,62,83,C9,32,90,03,4C,E0,77
AD,5F,83,C9,9E,B0,03, 2486

1117 DATA DB,8D,16,D0,A9,4F,8D,0F,D0,A9,40,8D,0E
 ,D0,A9,80,8D,1B,D0,A9, 2637

STARTRIP

Die Story: "Das war zuviel, eindeutig zuviel, jetzt werden wir ihnen eine Lehre erteilen", waren die letzten verständlichen Worte General Adoks, bevor seine Stimme im rauschenden Beifall des Erdsicherheitsrates unterging. In der Tat, es war schon recht dreist von den Tronicianern, die Erde mit einer Raumschifflotte zu überfallen, und das jeden Montag, Mittwoch und Freitag. (Die anderen Tage der Woche sind sie auf dem Rückweg zu Sammelsurium III, ihrem Heimatplaneten und

Sonntags wird die Niederlage gefeiert.)
Doch nun wird sich alles ändern.
General Adok stattet ihnen (warum gerade ihnen, das weiß niemand) einen etwas veralteten Wazda c-7b v2 xy Raummüllgleiter mit einem handelsüblichen Durchschlagsgeschoßwefer aus und gibt ihnen in feierlicher Stimmung (8:30 abends, 2 Flaschen billigen Wein dabei) den ehrenvollen Auftrag, den Tronicanern ordentlich eins überzubraten.
Um ca. 10:00 am nächsten Mor-

gen treffen sie an der Weltraum-
militärbasis STAD 35 ein und
überzeugen sich von der Funk-
tionsfähigkeit ihrer Ausrüstung.
So, jetzt gibt es kein Zurück
mehr. Ein Druck auf "Trip" und
ab geht's, mitten hinein in die sich
auf dem Rückzug befindliche
Raumflotte der Tronicianer. Den
Umstand, daß sich der "Trip"-
Knopf verklemmt hat und sie
nach mehr oder weniger langen
Jagdperioden in einen anderen
Bereich der 2,753 Lichtjahre lan-
gen Raumflotte versetzt, haben

sie leider nicht bemerkt, wundern sich daher umso mehr über die Folgen. V.V.(Viel Vergnügen)

Zum Programm:
Benötigt wird lediglich ein Joystick in Port 2 und viel Geduld (beim Tippen). Die einzelnen Spielstufen gehen, durch eine etwa einsekündige Unterbrechung gekennzeichnet, ineinander über.

* * *

Teil 1

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

```

1 REM * UNTER 'PRG1' SPEICHERN                <110>
2 IF PEEK(186)<8 THEN 100:REM KASSETTE        <233>
3 A$=CHR$(34)                                  <108>
4 PRINT"(WHITE CLEAR)LOADING... (BLACK DOWN2)":PO
KE 53281,0:POKE 53280,0
5 PRINT"LOAD"A$"PRG2"A$",8"                   <80>
6 PRINT"(DOWN4)RUN"                           <13>
7 PRINT"(DOWN2)LOAD"A$"PRG3"A$",8"           <255>
8 PRINT"(DOWN4)RUN"                           <15>
9 PRINT"(UP17)":POKE631,13:POKE632,13:POKE633,13
:POKE634,13
10 POKE 198,4:END                              <7>
100 REM *** KASSETTE ***                     <56>
105 POKE 631,131:POKE 198,1:END              <184>
      ENDE DES LISTINGS                      <174>

```

Teil 2

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

[illegible]

Teil 3

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

```

0 REM"*****
2 REM"*(SPACE3)S(SPACE2)P(SPACE2)A(SPACE2)C(SPACE2)E(SPACE2)T(SPACE2)R(SPACE2)I(SPACE2)P(SPACE3)
*
4 REM"*****
7 REM"*(SPACE)(C)1986(SPACE)BY(SPACE)PARAVISION(SPACE)COMPANY(SPACE)*
8 REM"*(SPACE)UNDER(SPACE)'PR03'(SPACE)ABSPEIC
HERN! (SPACE)***
9 REM"*****
10 PRINT"CLEAR":POKE 53280,0:POKE 53281,0
20 PRINT"YELLOW RIGHT6 CR SC SI SU SC SI CA SC
SI SU SC SI CR SPACE CR SPACE CR SU SC SI CR SU
SC SI SU SI CS"

```

```

30 PRINT"BROWN RIGHT6 CQ SC SK CQ SC CM CQ CR S  

K CQ SC CM SB SPACE SB SPACE SB SJ SC SI SB2 SPA  

C SB4" <161>
40 PRINT"{YELLOW RIGHT6 CE SPACE2 CE SPACE CE CZ  

2 SPACE CE SPACE CE CZ SC CX SPACE CE SJ SC SK C  

E SJ SC SK CZ SJ SK}" <136>
50 PRINT"{YELLOW RIGHT9 SU SC SI SU SC SI SU CR  

SI CA SC SI SU SC SI SU SI CS CA SPACE CS}" <238>
60 PRINT"{BROWN RIGHT9 SB SPACE2 SB SPACE SB4 CQ  

SC SK CQ SC CM SB3 SJ CR SK}" <202>
70 PRINT"{YELLOW RIGHT9 SJ SC SK SJ SC SK CZ SPA  

CE CX CE SPACE2 CE SPACE CE CZ SJ SK SPACE CE}" <235>
80 PRINT"{LIG.RED DOWN RIGHT11}P(SPACER)R(SPACER)E  

(SPACE)S(SPACER)E(SPACER)N(SPACER)T(SPACER)S(SPACER):  

" <227>
100 PRINT"{DOWN2 RIGHT3 PURPLE}*****  

*****" <144>
110 PRINT"{RIGHT3 ORANGE}*(SPACE2 SU SC SI CR SC  

SI SU SC SI SU SC SI CA SC CS CA CR CS CA SC SI  

CA CR CS CR SC SI SPACE2)*" <151>
120 PRINT"{RIGHT3 YELLOW}*(SPACE2 SJ SC SI CQ SC  

SK CQ SC CM SB SPACE2 CQ CM SPACE2 SB SPACE CQ  

CR SK SPACE SB SPACE CQ SC SK SPACE2)*" <61>
130 PRINT"{RIGHT3 LIG.GREEN}*(SPACE2 SJ SC SK CE  

SPACE2 CX SPACE CZ SJ SC SK CZ SC CX SPACE CE S  

PACE CX CZ SPACE CZ CE CX CE SPACE4)*" <122>
140 PRINT"{RIGHT3 GREEN}*****  

*****" <79>
150 PRINT"{LIG.BLUE DOWN RIGHT5}(C)1986(SPACER)BY  

(SPACER)PARAVISION(SPACER)COMPANY" <162>
160 PRINT"{BLUE DOWN RIGHT9}READING(SPACER)CHARAC  

TERDATA" <146>
170 FOR X=12288 TO 13007:READ DT:POKE X,DT:NEXT  

X <50>
180 FOR X=12288 TO 12288+1023:POKE X+1024,255-PE  

EK(X):NEXT X <12>
190 DATA 40,16,198,124,56,56,16,16 <107>
200 DATA 16,56,108,108,108,254,198,198,248,204,2  

04,248,204,198,198,252 <202>
210 DATA124,198,192,192,192,192,198,124,248,204,  

198,198,198,198,198,248,248 <159>
220 DATA254,192,192,248,192,192,192,254,254,192,  

192,248,192,192,192 <135>
230 DATA124,198,192,192,222,214,198,124,198,198,  

198,254,198,198,198,198 <172>
240 DATA 126,24,24,24,24,24,24,126,126,24,24,24,  

24,216,120,48 <96>
250 DATA204,220,248,240,248,220,206,198,192,192,  

192,192,192,192,192,254 <92>
260 DATA198,238,254,214,198,198,198,198,198,230,  

246,222,206,198,198,198 <199>
270 DATA124,198,198,198,198,198,198,124,252,198,  

198,252,192,192,192 <88>
280 DATA124,198,198,198,198,214,206,124,252,198,

```

```

198,252,248,220,206,198
290 DATA 124,198,192,120,28,6,198,124,126,24,24,
24,24,24,24
300 DATA 198,198,198,198,198,198,198,124,198,198
,198,198,198,238,124,56
310 DATA 198,198,198,198,214,254,238,198,198,198
,108,56,56,108,198,198
320 DATA 198,198,110,60,56,112,224,192,254,198,1
2,24,48,96,198,254
330 DATA 198,198,0,198,198,198,198,124,24,80,90,
161,161,90,80,24
340 DATA 24,60,24,90,255,255,231,66,129,64,20,24
,60,16,68,2
350 DATA 224,224,128,56,56,19,6,3,0,0,0,0,0,0,0
0
360 DATA 24,24,24,24,24,0,24,24,204,204,204,0,0,
0,0,0
370 DATA 204,254,204,204,204,204,204,0,24,56,120
,248,248,120,56,24
380 DATA 165,36,231,0,0,231,36,165,219,219,24,25
5,255,24,219,219
390 DATA 24,24,48,0,0,0,0,0,24,48,48,48,48,48,24
,0,48,24,24,24,24,48,0
400 DATA 0,24,60,126,126,60,24,0,0,24,24,126,126
,24,24,0
410 DATA 0,0,0,0,0,24,24,48,0,0,0,254,254,0,0,0,
0,0,0,0,0,0,24,24
420 DATA 0,3,6,12,24,48,96,0
430 DATA 60,230,206,222,246,230,206,120,12,12,12
,12,12,12,12,30
440 DATA 56,120,204,12,12,24,48,254,56,120,204,1
2,30,6,230,60
450 DATA 54,102,102,198,254,6,6,6,126,96,224,252
,14,6,230,60
460 DATA 30,48,96,252,206,198,230,60,254,198,6,1
2,24,24,48,48
470 DATA 60,230,206,120,44,230,230,60,60,230,194
,126,12,12,236,56
480 DATA 0,24,24,0,0,24,24,0,170,86,255,255,255,
255,255,255
490 DATA 255,255,255,255,255,255,85,170,0,254,25
4,0,0,254,254,0
500 DATA 0,96,48,24,24,48,96,0,28,54,38,12,24,24
,0,24,0,0,0,0,0,0,0
510 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
520 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
530 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
540 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
550 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
560 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,24,0
590 PRINT"CUP SPACE10"READING(SPACE2)SPRITEDATA(
SPACE6)"
600 FOR X=8576 TO 10687:READ DT:POKE X,DT:NEXT X
760 DATA 192, 0, 0, 32, 0, 0, 80, 0, 0
761 DATA 160, 0, 0, 192, 0, 0, 0, 0, 0
762 DATA 254, 0, 0, 4, 0, 0, 254, 0, 63
763 DATA 0, 1, 255, 252, 3, 192, 248, 6, 0
764 DATA 176, 6, 0, 160, 6, 0, 160, 6, 0
765 DATA 64, 6, 0, 64, 6, 0, 64, 6, 0
766 DATA 128, 3, 192, 128, 1, 255, 128, 0, 63
767 DATA 0
770 DATA 192, 0, 0, 56, 0, 0, 60, 0, 0
771 DATA 56, 0, 0, 224, 0, 0, 0, 0, 0
772 DATA 254, 0, 0, 4, 0, 0, 254, 0, 63
773 DATA 0, 1, 255, 252, 3, 192, 248, 6, 0
774 DATA 176, 6, 0, 160, 6, 0, 160, 6, 0
775 DATA 64, 6, 0, 64, 6, 0, 64, 6, 0
776 DATA 128, 3, 192, 128, 1, 255, 128, 0, 63
777 DATA 0
780 DATA 0, 16, 0, 0, 84, 0, 0, 84, 0
781 DATA 1, 85, 0, 1, 85, 0, 1, 33, 0
782 DATA 0, 32, 0, 0, 168, 128, 40, 184, 160
783 DATA 168, 184, 168, 152, 184, 152, 152, 184,
152
784 DATA 154, 186, 152, 154, 186, 152, 186, 186,
184

```

<110>	785 DATA 184, 184, 184, 184, 184, 168, 168, 168, 136, 168	<4>
<121>	786 DATA 40, 0, 160, 8, 0, 128, 0, 0, 0	<206>
	787 DATA 0	<249>
<149>	790 DATA 0, 24, 0, 0, 60, 0, 0, 60, 0	<22>
	791 DATA 0, 60, 0, 0, 60, 0, 0, 24, 0	<23>
<9>	792 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<62>
	793 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<63>
<125>	794 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<64>
	795 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<65>
<123>	796 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<66>
	797 DATA 0	<3>
<65>	800 DATA 0, 255, 0, 0, 235, 0, 0, 255, 0	<217>
	801 DATA 0, 235, 0, 0, 235, 0, 0, 255, 0	<208>
<45>	802 DATA 0, 255, 0, 0, 60, 0, 0, 60, 0	<85>
	803 DATA 0, 60, 0, 0, 60, 0, 32, 60, 8	<63>
<28>	804 DATA 32, 60, 8, 48, 255, 12, 48, 255, 12	<6>
	805 DATA 51, 235, 204, 51, 130, 204, 63, 0, 252	<228>
<109>	806 DATA 60, 0, 60, 60, 0, 60, 48, 0, 12	<15>
	807 DATA 0	<13>
<109>	810 DATA 0, 16, 0, 0, 16, 0, 0, 32, 0	<49>
	811 DATA 0, 236, 0, 3, 187, 0, 3, 171, 0	<67>
<210>	812 DATA 3, 187, 0, 3, 171, 0, 14, 170, 192	<239>
	813 DATA 58, 86, 176, 235, 155, 172, 238, 154, 236	
<208>	814 DATA 235, 155, 172, 171, 155, 168, 174, 170, 232	<214>
<18>		<212>
<121>	815 DATA 186, 254, 184, 234, 50, 172, 57, 1, 176	<30>
	816 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<86>
<25>	817 DATA 0	<23>
	820 DATA 57, 1, 176, 234, 50, 172, 186, 254, 184	<194>
<249>	821 DATA 174, 170, 232, 171, 155, 168, 235, 155, 172	<59>
<8>	822 DATA 238, 154, 236, 235, 155, 172, 58, 86, 176	
<39>	823 DATA 14, 170, 192, 3, 171, 0, 3, 187, 0	<217>
	824 DATA 3, 171, 0, 3, 187, 0, 0, 236, 0	<175>
<191>	825 DATA 0, 32, 0, 0, 16, 0, 0, 16, 0	<64>
	826 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<92>
<223>	827 DATA 0	<96>
	830 DATA 15, 170, 240, 58, 170, 172, 234, 190, 171	<33>
<110>		<170>
<83>	831 DATA 234, 235, 171, 235, 170, 235, 234, 170, 171	
	832 DATA 234, 235, 171, 234, 190, 171, 58, 170, 172	<30>
<162>		<218>
	833 DATA 15, 170, 240, 0, 235, 0, 0, 235, 0	<151>
<173>	834 DATA 0, 235, 0, 3, 170, 192, 3, 170, 192	<191>
	835 DATA 14, 170, 176, 14, 170, 176, 58, 235, 172	
<183>		<6>
	836 DATA 58, 235, 172, 234, 235, 171, 255, 255, 255	
<193>		<126>
	837 DATA 0	<43>
<203>	840 DATA 3, 235, 192, 10, 170, 160, 42, 190, 168	<201>
	841 DATA 170, 235, 170, 170, 190, 170, 186, 235, 174	<59>
<213>		
	842 DATA 190, 190, 190, 183, 170, 222, 181, 235, 94	<11>
<223>		<164>
<111>	843 DATA 181, 235, 94, 183, 170, 222, 190, 190, 190	
	844 DATA 186, 235, 174, 170, 170, 170, 170, 119, 170	<52>
<168>		<126>
<232>	845 DATA 41, 221, 232, 7, 119, 112, 0, 221, 0	<116>
<2>	846 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<53>
<175>	847 DATA 0	<82>
<80>	850 DATA 0, 221, 0, 7, 119, 112, 41, 221, 232	
<225>	851 DATA 170, 119, 170, 170, 170, 170, 186, 235, 174	<43>
<212>		
<24>	852 DATA 190, 190, 190, 183, 170, 222, 181, 235, 94	<21>
<26>		<174>
<228>	853 DATA 181, 235, 94, 183, 170, 222, 190, 190, 190	
<187>		
<143>	854 DATA 186, 235, 174, 170, 190, 170, 170, 235, 170	<64>
<91>		<107>
<236>	855 DATA 42, 190, 168, 10, 170, 160, 3, 235, 192	<126>
<223>	856 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<63>
<35>		

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), Hueycopter (C-16/116), Hardcopy-routine (C-16/116), Helmut und die Zeitmaschine (C-16/116), Lucas (VC-20), Rock'n'roll (VC-20), Charly der Raketemann (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-Legen (VC-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, Spritgenerator, 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
Roko (C-64), Basic-Befehlsweiterung (C-64), Raketenstart (C-16/116), Drel-kampf (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
Allen 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 Pack goes home (C-64), EDDY (C-64), Invas-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), Scramble (VC20), Refusal-Basiclader (C-16), Hardy-Copy-Blitz (C-16), Auto-start (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

Heft 2/87
Commodore
Bestell-Nr. COM CK 2/3 16,- DM
Bestell-Nr. COM CD 2/3 20,- DM
Quest for Tires (VC20), Cave Fighters (VC20), Mc Stoney in the Caves (C16/116), Cube (C16/116), Hostage (C16/116), Address 16 (plus4), Easy Basic (C64), Ski-Jumping (C64), Job-Race (C64)

Schneider
Bestell-Nr. COM SK 2/3 16,- DM
Bestell-Nr. COM SD 2/3 39,- DM
The Hero, Smash it, 16 RSX Befehle, Disc Star 2.1

Heft 3/86
Commodore
Bestell-Nr. COM CK 3/2 16,- DM
Bestell-Nr. COM CD 3/2 20,- DM
Miner-Jumper (C-16/116), Hero (C-16/116), Data-Maker (C-16/116), Befehlsweiterung (C-64), Starship (C-64), Mini-Car-Race (C-64), Text-Scroller (C-64), Text-Editor (C-64), Checksummer (VC-20), Robin at Ravenwood (VC-20), Mini-Karate (VC-20), Autochanger-Routine (VC-20)

Schneider
Bestell-Nr. COM SK 3/2 16,- DM
Bestell-Nr. COM SD 3/2 39,- DM
Smily, Script, 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-Hell (VC-20), Roadrunner (C-64), Basic Extension (C-64), Meteoriten-sturm (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), Zauber-puzzle (C16), Schieberoutine (C16), Atlantis (C64), Astronomie (C64), Deutsche Fehlermeldung (C64), Rem-zellenkiller (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 3/87
Commodore
Bestell-Nr. COM CK 3/3 16,- DM
Bestell-Nr. COM CD 3/3 20,- DM
Action Ball (VC20), Star Attack (VC20), Der neue Checksummer (C16/116), Bobdriving (C16/116), Find-Utility (C16/116), Mac Coin (C16/116), SWAP-Utility (C16/116), Druckerroutine (C16/116/plus4), Appleman (C16/116 m. 64K), Trampolin Games (C64), Checksummer (C64), Mission Code (C64), Sprite Tool (C64)

Schneider
Bestell-Nr. COM SK 3/3 16,- DM
Bestell-Nr. COM SD 3/3 39,- DM
Boing Boing, Colors, Druckerproto-koll, Checksummer, Aurion 2

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), Mergo-Routine (VC-20), Shoot It (VC-20), Fast Trust (C-64), Data-Text 64 (C-64), Gary, der schnelle Kachelverleger (C-16/116), SDI (C-16/116), Deep-Routine (C-16/116)

Schneider
Bestell-Nr. COM SK 4/2 16,- DM
Bestell-Nr. COM SD 4/2 39,- DM
Space-Rescue, Texteditor, Starflight, 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), Sasa-ga Attack (VC-20), Sound (VC-20), Plati-nenlayout (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), Galaxix (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

Aus diesem Heft:

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

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

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.

Absender
(Bitte deutlich schreiben)

Vorname/Name

Homecomputer-System

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Absender
(Bitte deutlich schreiben)

Vorname/Name

Homecomputer-System

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Absender
(Bitte deutlich schreiben)

Vorname/Name

Homecomputer-System

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Antwortkarte

Tronic-Verlag GmbH
ABO-Service
Postfach 870

D-3440 Eschwege

Antwortkarte

Tronic-Verlag GmbH
»Compute mit«
Postfach 870

D-3440 Eschwege

Antwortkarte

Tronic-Verlag GmbH
Software-Service
Postfach 870

D-3440 Eschwege

Bitte mit der
jeweils gültigen
Postkartengebühr
freimachen

Bitte mit der
jeweils gültigen
Postkartengebühr
freimachen

Bitte mit der
jeweils gültigen
Postkartengebühr
freimachen

preis!

Schneider
CPC 464

aket 1

Catch, Interceptor 3 D, City
r's Castle, Copter Patrol, CPC
emon Attack, Duell, Etis, Labyrinth,
Snider's Maze, Geister Schloß,
nger, Secret Valley, Aladin,
e, Cave runner, Cobra, Erwin,
t

Kassette 20, -
Diskette 27, -

aket 2
ler

gen, Datenverwaltung, Discopy,
er, Remkiller, SU-File,
ing

Kassette 20, -
Diskette 27, -

	Preis Kass./Disk.	Bestell-Nr.
	18,50/28,50	SR 13
Caves	18,50/28,50	SR 23
	18,50/28,50	SR 33
d	18,50/28,50	SR 43
	18,50/28,50	SR 53
urion	18,50/28,50	SR 63
	18,50/28,50	SR 14

ware service

Tronic-Verlag bietet
Software für jeder-

ugen Sie sich von
n Angeboten und
en Sie noch heute.
stellungen werden
alb nur

Woche*

ingang bearbeitet.
sche Bestellungen
er Rufnummer

6 51/3 00 11

g!


```

870 DATA 0, 0, 0, 2, 0, 128, 10, 130, 160
871 DATA 10, 130, 160, 10, 170, 160, 11, 170, 22
4
872 DATA 11, 150, 224, 43, 150, 232, 46, 150, 10
4
873 DATA 46, 105, 184, 46, 105, 184, 46, 150, 10
4
874 DATA 174, 105, 184, 105, 170, 110, 186, 170,
174
875 DATA 170, 130, 170, 42, 0, 168, 0, 0, 0
876 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
877 DATA 0
880 DATA 0, 84, 0, 1, 185, 0, 6, 238, 64
881 DATA 26, 186, 144, 26, 186, 144, 26, 186, 14
4
882 DATA 26, 238, 144, 6, 170, 64, 1, 169, 0
883 DATA 0, 84, 0, 0, 16, 0, 0, 16, 0
884 DATA 0, 16, 0, 0, 84, 0, 1, 169, 0
885 DATA 6, 186, 64, 26, 238, 144, 27, 171, 144
886 DATA 26, 154, 144, 6, 70, 64, 1, 1, 0
887 DATA 0
890 DATA 14, 238, 224, 43, 187, 184, 170, 170, 1
70
891 DATA 149, 90, 154, 166, 102, 102, 166, 90, 1
02
892 DATA 166, 102, 154, 170, 170, 170, 165, 166,
154
893 DATA 166, 102, 106, 166, 102, 106, 166, 102,
154
894 DATA 170, 170, 170, 47, 170, 248, 10, 235, 1
60
895 DATA 2, 190, 128, 0, 170, 0, 0, 170, 0
896 DATA 2, 170, 128, 2, 170, 128, 10, 170, 160
897 DATA 0
900 DATA 0, 0, 0, 0, 40, 0, 0, 190, 0
901 DATA 0, 190, 0, 2, 255, 120, 11, 255, 120
902 DATA 11, 255, 224, 47, 235, 248, 10, 130, 16
0
903 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0
904 DATA 0, 40, 0, 0, 190, 0, 0, 190, 0
905 DATA 0, 190, 0, 2, 255, 120, 11, 255, 224
906 DATA 47, 235, 248, 10, 130, 160, 0, 0, 0
907 DATA 0
910 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
911 DATA 30, 191, 164, 123, 238, 253, 123, 239,
189
912 DATA 123, 239, 237, 126, 190, 189, 127, 255,
253
913 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
914 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
915 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
916 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84
917 DATA 0
920 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
921 DATA 31, 255, 244, 123, 190, 189, 123, 254,
237
922 DATA 122, 254, 189, 123, 254, 237, 123, 254,
189
923 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
924 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
925 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
926 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84
927 DATA 0
930 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
931 DATA 31, 255, 244, 123, 190, 237, 122, 190,
237
932 DATA 123, 190, 189, 123, 190, 237, 123, 190,
237
933 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
934 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
935 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
936 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84
937 DATA 0
940 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
941 DATA 31, 255, 244, 123, 187, 237, 123, 187,
237
942 DATA 122, 187, 237, 123, 186, 173, 123, 187,
237
943 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
944 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
945 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
946 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84

```

```

<240>
<95>
<218>
<216>
<17>
<159>
<146>
<83>
<197>
<105>
<240>
<180>
<184>
<44>
<37>
<93>
<37>
<229>
<11>
<178>
<83>
<162>
<152>
<103>
<173>
<104>
<131>
<188>
<113>
<241>
<86>
<69>
<31>
<87>
<111>
<130>
<123>
<251>
<158>
<94>
<41>
<97>
<121>
<140>
<133>
<5>
<244>
<85>
<51>
<107>
<131>
<150>
<143>
<15>
<71>
<83>
<61>
<117>
<141>
<160>

```

```

947 DATA 0
950 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
951 DATA 31, 255, 244, 126, 191, 173, 123, 254,
253
952 DATA 126, 254, 189, 127, 190, 237, 122, 255,
173
953 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
954 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
955 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
956 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84
957 DATA 0
960 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
961 DATA 31, 255, 244, 122, 254, 173, 123, 191,
237
962 DATA 122, 255, 189, 123, 190, 253, 122, 254,
173
963 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
964 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
965 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
966 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84
967 DATA 0
970 DATA 5, 85, 80, 31, 255, 244, 31, 255, 244
971 DATA 31, 255, 244, 122, 190, 189, 126, 254,
237
972 DATA 126, 254, 189, 126, 254, 237, 126, 254,
189
973 DATA 127, 255, 253, 31, 255, 244, 31, 255, 2
44
974 DATA 5, 65, 80, 5, 20, 80, 5, 0, 80
975 DATA 5, 0, 80, 1, 65, 64, 1, 85, 64
976 DATA 7, 255, 208, 31, 255, 244, 21, 85, 84
977 DATA 0
980 DATA 0, 0, 0, 0, 0, 0, 0, 32, 0
981 DATA 0, 136, 32, 10, 242, 136, 0, 152, 224
982 DATA 3, 19, 144, 41, 220, 96, 8, 85, 136
983 DATA 35, 69, 224, 0, 87, 232, 1, 45, 128
984 DATA 8, 75, 96, 11, 123, 0, 34, 177, 2
985 DATA 0, 140, 128, 0, 40, 160, 0, 32, 0
986 DATA 0, 8, 0, 0, 0, 0, 0, 0
987 DATA 0
990 DATA 0, 0, 0, 0, 0, 0, 0, 0
991 DATA 0, 0, 0, 2, 240, 128, 0, 144, 224
992 DATA 3, 19, 128, 9, 220, 96, 8, 85, 128
993 DATA 3, 69, 224, 0, 87, 0, 0, 45, 128
994 DATA 0, 72, 64, 3, 120, 0, 2, 177, 0
995 DATA 0, 140, 128, 0, 0, 0, 0, 0
996 DATA 0, 0, 0, 0, 0, 0, 0, 0
997 DATA 0
1000 DATA 0, 0, 0, 0, 0, 0, 0, 0
1001 DATA 0, 0, 0, 0, 0, 0, 0, 0
1002 DATA 0, 0, 0, 0, 220, 0, 0, 20, 128
1003 DATA 0, 69, 192, 0, 23, 0, 0, 45, 128
1004 DATA 0, 72, 0, 0, 0, 0, 0, 0
1005 DATA 0, 0, 0, 0, 0, 0, 0, 0
1006 DATA 0, 0, 0, 0, 0, 0, 0, 0
1007 DATA 0
1010 DATA 0, 0, 0, 0, 0, 0, 0, 0
1011 DATA 0, 0, 0, 0, 0, 0, 0, 0
1012 DATA 0, 0, 0, 0, 0, 0, 0, 0
1013 DATA 0, 0, 0, 0, 0, 0, 0, 0
1014 DATA 0, 0, 0, 0, 0, 0, 0, 0
1015 DATA 0, 0, 0, 0, 0, 0, 0, 0
1016 DATA 0, 0, 0, 0, 0, 0, 0, 0
1017 DATA 0
20000 DATA 0, 127, 243, 3, 255, 243, 7, 255, 243
20001 DATA 15, 255, 231, 31, 255, 231, 31, 255,
231
20002 DATA 15, 255, 207, 7, 255, 207, 1, 255, 20
7
20003 DATA 0, 0, 0, 0, 0, 0, 0, 0
20004 DATA 63, 248, 1, 63, 254, 1, 63, 255, 1
20005 DATA 127, 255, 131, 127, 255, 131, 127, 25
5, 3
20006 DATA 255, 254, 7, 255, 252, 7, 255, 224, 7
20007 DATA 0
20010 DATA 255, 248, 127, 255, 248, 127, 255, 24
8, 255
20011 DATA 255, 240, 255, 255, 241, 255, 255, 24
3, 255
20012 DATA 255, 227, 255, 255, 231, 255, 255, 23
1, 255
20013 DATA 0, 0, 0, 0, 0, 0, 0, 0
20014 DATA 240, 31, 255, 240, 31, 255, 240, 31,
255

```

```

<153>
<25>
<35>
<191>
<71>
<127>
<151>
<170>
<163>
<35>
<57>
<47>
<81>
<137>
<161>
<180>
<173>
<45>
<33>
<36>
<91>
<147>
<171>
<190>
<183>
<112>
<209>
<111>
<152>
<129>
<231>
<32>
<193>
<4>
<11>
<118>
<166>
<245>
<131>
<10>
<203>
<14>
<15>
<227>
<67>
<172>
<19>
<20>
<213>
<24>
<25>
<26>
<27>
<28>
<29>
<30>
<223>
<233>
<89>
<178>
<148>
<85>
<143>
<34>
<88>
<167>
<122>
<51>
<158>
<145>

```

```

20015 DATA 224, 63, 255, 224, 62, 3, 224, 62, 3
20016 DATA 192, 124, 7, 192, 124, 7, 192, 124, 7
20017 DATA 0
20020 DATA 252, 63, 248, 252, 63, 254, 252, 63,
255
20021 DATA 252, 127, 255, 252, 127, 255, 252, 12
7, 255
20022 DATA 252, 255, 254, 252, 255, 252, 252, 25
5, 224
20023 DATA 0, 0, 0, 0, 0, 0, 0, 0
20024 DATA 243, 240, 0, 243, 248, 0, 243, 252, 0
20025 DATA 231, 254, 0, 231, 191, 0, 231, 159, 1
28
20026 DATA 207, 15, 128, 207, 7, 128, 207, 3, 12
8
20027 DATA 0
20030 DATA 15, 255, 231, 15, 255, 231, 15, 255,
231
20031 DATA 159, 255, 207, 159, 255, 207, 31, 255
, 207
20032 DATA 63, 255, 159, 63, 255, 159, 63, 255,
159
20033 DATA 0, 0, 0, 0, 0, 0, 0, 0
20034 DATA 3, 224, 126, 3, 224, 127, 3, 224, 127
20035 DATA 7, 192, 255, 7, 192, 251, 7, 192, 249
20036 DATA 15, 129, 240, 15, 129, 240, 15, 129,
240
20037 DATA 0
20040 DATA 255, 1, 243, 255, 193, 243, 255, 225,
243
20041 DATA 255, 243, 231, 255, 243, 231, 255, 22
7, 231
20042 DATA 255, 199, 207, 255, 135, 207, 252, 7,
207
20043 DATA 0, 0, 0, 0, 0, 0, 0, 0
20044 DATA 0, 31, 62, 0, 31, 62, 128, 31, 62
20045 DATA 192, 62, 124, 224, 62, 124, 240, 62,
124
20046 DATA 248, 124, 248, 120, 124, 248, 56, 124
, 248
20047 DATA 0
20050 DATA 255, 128, 182, 255, 224, 170, 255, 24
0, 226
20051 DATA 255, 248, 0, 255, 248, 0, 255, 248, 0
20052 DATA 255, 224, 0, 255, 192, 0, 254, 0, 0
20053 DATA 0, 0, 0, 0, 0, 0, 0, 0
20054 DATA 0, 0, 0, 0, 0, 0, 0, 0
20055 DATA 0, 0, 0, 0, 0, 0, 0, 0
20056 DATA 0, 0, 0, 0, 0, 0, 0, 0
20057 DATA 0
20060 DATA 0, 16, 0, 0, 0, 0, 0, 0
20061 DATA 0, 16, 0, 0, 0, 0, 0, 0
20062 DATA 0, 0, 0, 16, 0, 0, 0, 56, 0
20063 DATA 37, 125, 72, 0, 56, 0, 0, 16, 0
20064 DATA 0, 0, 0, 16, 0, 0, 0, 0
20065 DATA 0, 16, 0, 0, 0, 0, 0, 0
20066 DATA 0, 16, 0, 0, 0, 0, 0, 0
20067 DATA 0
31000 PRINT "UP SPACE10"READING(SPACE2)M-CODEDAT
A(SPACE3)
31010 FOR X=2048 TO 4102:READ DT:POKE X,DT:NEXT
X
32000 DATA173,207,50,72,162,200,189,7,50,157,0,5
0,202,138,208,246,104,141,0,50
32001 DATA96,32,0,16,142,0,208,32,0,16,142,1,208
,32,0,16,142,2,208,32,0,16,142
32002 DATA3,208,32,0,16,142,4,208,32,0,16,142,5,
208,32,0,16,142,6,208,32,0,16
32003 DATA142,7,208,32,0,16,142,8,208,32,0,16,14
2,9,208,32,0,16,142,10,208,32
32004 DATA0,16,142,11,208,32,0,16,142,12,208,32,
0,16,142,13,208,32,0,16,142,14
32005 DATA208,32,0,16,142,15,208,32,0,16,142,21,
208,96,32,0,16,142,24,7,32,0
32006 DATA16,142,248,7,32,0,16,142,249,7,32,0,16
,142,250,7,32,0,16,142,251,7
32007 DATA32,0,16,142,252,7,32,0,16,142,253,7,32
,0,16,142,254,7,32,0,16,142,255
32008 DATA7,96,32,0,16,142,39,208,32,0,16,142,40
,208,32,0,16,142,41,208,32,0
32009 DATA16,142,42,208,32,0,16,142,43,208,32,0,
16,142,44,208,32,0,16,142,45
32010 DATA208,32,0,16,142,46,208,96,32,0,16,142,
37,208,32,0,16,142,38,208,96
32011 DATA32,0,16,142,20,208,96,32,0,16,142,32,2
08,32,0,25,9,33,208,96,162,14

```

```

<49>
<210>
<98>
<33>
<124>
<5>
<168>
<98>
<172>
<217>
<108>
<200>
<75>
<139>
<178>
<71>
<7>
<218>
<118>
<109>
<112>
<108>
<188>
<10>
<121>
<146>
<128>
<209>
<229>
<209>
<198>
<199>
<200>
<201>
<138>
<99>
<157>
<24>
<87>
<157>
<104>
<105>
<148>
<140>
<106>
<32>
<252>
<68>
<15>
<145>
<42>
<59>
<39>
<26>
<8>
<9>
<150>

```

```

32012 DATA189,1,208,24,234,125,107,9,157,1,208,2
02,138,208,241,96,32,0,16,142
32013 DATA108,9,32,0,16,142,109,9,32,0,16,142,11
0,9,32,0,16,142,111,9,32,0,16
32014 DATA142,112,9,32,0,16,142,113,9,32,0,16,14
2,114,9,32,0,16,142,115,9,32
32015 DATA0,16,142,116,9,32,0,16,142,117,9,32,0,
16,142,118,9,32,0,16,142,119
32016 DATA9,32,0,16,142,120,9,32,0,16,142,121,9,
96,0,250,1,1,253,1,255,1,255
32017 DATA1,254,1,1,1,173,224,7,201,58,208,8,169
,48,141,224,7,238,223,7,173,223
32018 DATA7,201,58,208,8,169,48,141,223,7,238,22
2,7,173,222,7,201,58,208,8,169
32019 DATA48,141,222,7,238,221,7,173,221,7,201,5
8,208,8,169,48,141,221,7,238
32020 DATA220,7,96,169,224,141,2,220,173,0,220,4
1,4,208,10,173,0,208,201,48,240
32021 DATA3,206,0,208,173,0,220,41,8,208,10,173,
0,208,201,240,240,3,238,0,208
32022 DATA173,0,220,41,16,208,19,173,0,208,141,2
,208,169,192,141,3,208,173,21
32023 DATA208,9,2,141,21,208,173,3,208,201,16,17
6,8,173,21,208,41,253,141,21
32024 DATA208,169,255,141,2,220,96,162,7,189,248
,7,221,6,14,208,9,189,255,13
32025 DATA157,248,7,76,35,10,254,248,7,202,138,2
08,232,96,32,0,16,142,0,14,32
32026 DATA0,16,142,1,14,32,0,16,142,2,14,32,0,16
,142,3,14,32,0,16,142,4,14,32
32027 DATA0,16,142,5,14,32,0,16,142,6,14,32,0,16
,142,7,14,32,0,16,142,8,14,32
32028 DATA0,16,142,9,14,32,0,16,142,10,14,32,0,1
6,142,11,14,32,0,16,142,12,14
32029 DATA32,0,16,142,13,14,96,0,173,30,208,141,
125,10,41,1,240,6,169,255,141
32030 DATA125,10,96,173,125,10,41,6,201,6,208,16
,169,156,141,250,7,169,159,141
32031 DATA1,14,141,8,14,238,224,7,173,125,10,41,
10,201,10,208,16,169,156,141
32032 DATA251,7,169,159,141,2,14,141,9,14,238,22
4,7,173,125,10,41,18,201,18,208
32033 DATA16,169,156,141,252,7,169,159,141,3,14,
141,10,14,238,224,7,173,125,10
32034 DATA41,34,201,34,208,16,169,156,141,253,7,
169,159,141,4,14,141,11,14,238
32035 DATA224,7,173,125,10,41,66,201,66,208,16,1
69,156,141,254,7,169,159,141
32036 DATA5,14,141,12,14,238,224,7,173,125,10,41
,130,201,130,208,16,169,156,141
32037 DATA255,7,169,159,141,6,14,141,13,14,238,2
24,7,96,173,5,208,201,1,240,7
32038 DATA173,5,208,201,0,208,16,173,18,208,41,1
5,105,137,141,250,7,141,1,14
32039 DATA141,8,14,173,7,208,201,1,240,7,173,7,2
08,201,0,208,16,173,18,208,41
32040 DATA15,105,137,141,251,7,141,2,14,141,9,14
,173,9,208,201,1,240,7,173,9
32041 DATA208,201,0,208,16,173,18,208,41,15,105,
137,141,252,7,141,3,14,141,10
32042 DATA14,173,11,208,201,1,240,7,173,11,208,2
01,0,208,16,173,18,208,41,15
32043 DATA105,137,141,253,7,141,4,14,141,11,14,1
73,13,208,201,1,240,7,173,13
32044 DATA208,201,0,208,16,173,18,208,41,15,105,
137,141,254,7,141,5,14,141,12
32045 DATA14,173,15,208,201,1,240,7,173,15,208,2
01,0,208,16,173,18,208,41,15
32046 DATA105,137,141,255,7,141,6,14,141,13,14,9
6,32,5,9,32,237,11,32,13,10,32
32047 DATA126,10,32,122,9,32,37,11,96,169,224,14
1,2,220,173,0,220,41,4,208,10
32048 DATA173,0,208,201,94,240,3,206,0,208,173,0
,220,41,8,208,10,173,0,208,201
32049 DATA240,240,3,238,0,208,173,0,220,41,16,20
8,26,173,21,208,41,2,208,19,173
32050 DATA0,208,141,2,208,169,176,141,3,208,173,
21,208,9,2,141,21,208,173,3,208
32051 DATA201,16,176,8,173,21,208,41,253,141,21,
208,169,255,141,2,220,173,3,208
32052 DATA141,1,212,173,21,208,41,2,240,8,169,21
,141,4,212,76,99,12,169,20,141
32053 DATA4,212,173,18,208,41,4,240,8,169,21,141
,11,212,76,119,12,169,20,141
32054 DATA11,212,173,18,208,41,47,141,8,212,162,
12,189,2,208,201,249,144,11
```



```

32055 DATA18,208,41,3,56,233,4,157,108,9,189,2,2
08,201,94,176,8,173,18,208,41
32056 DATA3,157,108,9,202,202,138,208,218,96,32,
0,8,32,218,11,32,0,8,32,237,11
32057 DATA96,32,0,16,142,114,9,32,0,16,142,115,9
,32,0,16,142,116,9,32,0,16,142
32058 DATA117,9,32,0,16,142,118,9,32,0,16,142,11
9,9,32,0,16,142,120,9,32,0,16
32059 DATA142,121,9,96,0,250,0,1,0,1,0,1,0,1,0,1
,0,1,173,224,7,201,58,208,8,169
32060 DATA48,141,224,7,238,223,7,173,223,7,201,5
8,208,8,169,48,141,223,7,238
32061 DATA222,7,173,222,7,201,58,208,8,169,48,14
1,222,7,238,221,7,0,23,0,4,239
32062 DATA16,0,249,0,0,6,4,0,0,247,4,0,16,0,4,0,
16,145,16,0,0,0,249,7,0,247
32063 DATA0,16,255,16,22,16,16,255,16,16,16,241,
255,247,16,127,16,251,16,247
32064 DATA16,239,16,23,191,16,16,183,16,255,20,1
6,255,255,16,182,16,16,22,255
32065 DATA247,255,16,255,255,16,16,143,255,16,16
,16,150,16,16,16,249,255,150
32066 DATA255,16,16,16,255,0,0,0,4,0,0,0,0,0,0,0
,0,0,0,0,0,16,0,65,0,0,0,0
32067 DATA0,0,0,0,0,4,0,0,0,64,0,0,0,0,16,0,0,
16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
32068 DATA32,16,0,0,0,0,0,0,249,0,1,0,0,0,0,0,0,
0,0,0,0,0,97,0,1,0,255,16,0,1
32069 DATA0,1,16,255,0,0,0,97,0,0,0,1,0,1,0,0,1,
0,0,4,0,0,97,0,0,0,0,0,253
32070 DATA4,0,64,0,0,0,0,0,0,64,64,81,137,149,14
7,139,148,150,159,137,149,147
32071 DATA139,148,150,159,0,255,255,255,215,255,
213,134,16,255,16,247,16,0,16
32072 DATA0,16,255,20,0,22,0,20,239,16,0,249,0,1
6,6,16,0,16,247,20,0,16,0,20
32073 DATA0,16,145,16,6,0,8,248,73,0,247,0,16,25
5,16,22,16,16,255,16,16,16,241
32074 DATA255,247,16,127,16,251,16,247,16,239,16
,151,191,16,16,182,16,255,16
32075 DATA16,255,251,16,182,16,16,22,255,247,255
,16,255,255,16,16,143,255,16
32076 DATA16,16,150,16,16,16,249,255,150,255,16,
16,16,251,0,0,0,4,0,0,0,0,0,0
32077 DATA0,0,0,0,0,0,0,16,0,65,0,0,0,0,0,0,0,0,
0,0,4,0,0,0,64,0,0,0,0,0,16,0,0
32078 DATA16,0,0,0,0,0,0,0,0,0,0,0,0,0,16,0,0,
0,0,0,0,249,0,64,0,0,0,0,0,0
32079 DATA0,0,0,0,0,65,0,0,0,255,16,0,0,0,0,24,2
55,0,0,0,97,0,0,0,1,0,0,0,0,0
32080 DATA0,0,4,0,0,0,0,65,0,0,0,0,0,253,4,0,64,0,
0,0,0,0,64,64,81,249,231,255
32081 DATA255,20,255,16,255,150,231,6,255,255,13
5,16,255,255,247,215,255,213
32082 DATA134,16,255,16,247,16,0,0,0,0,255,20,0,
22,0,20,239,16,0,249,0,16,6,20
32083 DATA0,16,247,4,0,16,0,0,16,145,16,6,0,0,
249,9,0,247,0,16,255,16,22,16
32084 DATA16,255,16,16,16,241,255,247,16,127,16,
251,16,247,16,239,16,151,191
32085 DATA16,16,182,16,255,20,16,255,251,16,150,
16,16,22,255,247,255,16,255,255
32086 DATA16,16,143,255,16,16,150,16,16,16,24
9,255,150,255,16,16,16,255,0
32087 DATA0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,65
,0,0,0,0,0,0,0,0,0,0,4,0,0,0
32088 DATA64,0,0,0,0,16,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,32,16,0,0,0,0,0,0,249
32089 DATA0,1,0,0,0,0,0,0,0,0,0,0,0,65,0,1,0,255
,16,0,1,0,0,16,255,0,0,97,0
32090 DATA0,0,1,0,1,0,0,1,0,0,4,0,0,0,97,0,0,0,0
,0,253,4,0,64,0,0,0,0,0,64
32091 DATA64,81,32,253,174,32,158,183,96
33000 PRINT "CLEAR BLACK":AS=CHR$(34)
33005 IF PEEK(186)<8 THEN 33040
33010 PRINT "DONNEE2"LOAD"AS"PRG4"AS",8"
33020 PRINT "DONNEE"RUN
33030 PRINT "HOMME":POKE 631,13:POKE 632,13:POKE
198,2:END
33040 POKE 631,131:POKE 198,1:NEW
ENDE DES LISTINGS

```

Teil 4

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

```

0 REM UNTER 'PRG4' SPEICHERN
1 FOR X=49152 TO 50836:READ DT:POKE X,DT:NEXT X
2 DATA120,169,43,141,20,3,169,0,133,249,133,247,
133,251,169,192,141,21,3
3 DATA169,205,133,250,169,202,133,248,169,199,13
3,252,169,32,141,255,20,141
4 DATA254,20,141,253,20,96,173,255,20,208,84,160
,0,177,249,208,8,169,0,133
5 DATA249,169,205,133,250,177,249,141,1,21,230,2
49,165,249,208,2,230,250
6 DATA177,249,141,0,21,230,249,165,249,208,2,230
,250,177,249,141,21,21,230
7 DATA249,165,249,208,2,230,250,177,249,141,255,
20,230,249,165,249,208,2
8 DATA230,250,177,249,162,0,142,4,212,141,4,21,2
30,249,165,249,208,2,230
9 DATA250,206,255,20,173,254,20,208,84,160,0,177
,247,208,8,169,0,133,247
10 DATA169,202,133,248,177,247,141,8,21,230,247,
165,247,208,2,230,248,177
11 DATA247,141,7,21,230,247,165,247,208,2,230,24
8,177,247,141,22,21,230,247
12 DATA165,247,208,2,230,248,177,247,141,254,20,
230,247,165,247,208,2,230
13 DATA248,177,247,162,0,142,11,212,141,11,21,23
0,247,165,247,208,2,230,248
14 DATA206,254,20,173,253,20,208,84,160,0,177,25
1,208,8,169,0,133,251,169
15 DATA199,133,252,177,251,141,15,21,230,251,165
,251,208,2,230,252,177,251
16 DATA141,14,21,230,251,165,251,208,2,230,252,1
77,251,141,23,21,230,251,165
17 DATA251,208,2,230,252,177,251,141,253,20,230,
251,165,251,208,2,230,252
18 DATA177,251,162,0,142,18,212,141,18,21,230,25
1,165,251,208,2,230,252,206
19 DATA253,20,162,21,189,255,20,157,255,211,202,
138,208,246,230,2,173,21,21
20 DATA201,1,208,10,169,0,141,5,21,169,0,141,6,2
1,173,21,21,201,2,208,10,169
21 DATA10,141,5,21,169,0,141,6,21,173,21,21,201,
3,208,10,169,249,141,5,21
22 DATA169,233,141,6,21,173,21,21,201,4,208,34,1
65,2,41,4,208,14,173,0,21
23 DATA24,105,16,141,0,21,144,3,238,1,21,173,0,2
1,56,233,8,141,0,21,176,3
24 DATA206,1,21,173,21,21,201,5,208,34,165,2,41,
8,208,14,173,0,21,24,105,14
25 DATA141,0,21,144,3,238,1,21,173,0,21,56,233,7
,141,0,21,176,3,206,1,21,173
26 DATA21,201,6,208,34,165,2,41,16,208,14,173
,2,21,24,105,32,141,2,21,144
27 DATA3,238,3,21,173,2,21,56,233,16,141,2,21,17
6,3,206,3,21,173,21,21,201
28 DATA7,208,13,169,8,141,2,21,141,3,21,169,154,
141,5,21,173,21,21,201,8,208
29 DATA14,173,0,21,24,105,48,141,0,21,144,3,238,
1,21,173,21,21,201,9,208,14
30 DATA173,0,21,56,233,48,141,0,21,176,3,206,1,2
1,173,21,21,201,10,208,11
31 DATA165,2,41,1,208,5,169,129,141,4,212,173,21
,21,201,11,208,14,169,2,141
32 DATA3,21,173,2,21,24,105,5,141,2,21,173,21,21
,201,12,208,42,165,2,41,4
33 DATA208,14,173,0,21,74,141,0,212,173,1,21,74,
141,1,212,165,2,41,6,208,16
34 DATA173,0,21,74,74,141,0,212,173,1,21,74,74,1
41,1,212,173,21,21,201,13
35 DATA208,42,165,2,41,4,208,14,173,2,21,74,141,
2,212,173,3,21,74,141,3,212
36 DATA165,8,41,6,208,16,173,2,21,74,74,141,2,21
2,173,3,21,74,74,141,3,212
37 DATA173,21,21,201,14,208,8,169,8,141,2,21,141
,3,21,173,21,21,201,15,208
38 DATA25,165,2,41,4,208,11,169,3,141,3,21,238,2
,21,76,49,234,169,8,141,3
39 DATA21,206,2,21,173,21,21,201,16,208,8,169,12
8,141,2,21,238,3,21,173,22
40 DATA21,201,1,208,10,169,8,141,12,21,169,0,141

```

```

,13,21,173,22,21,201,2,208
41 DATA10,169,10,141,12,21,169,0,141,13,21,173,2
2,21,201,3,208,10,169,249
42 DATA141,12,21,169,233,141,13,21,173,22,21,201
,4,208,34,165,2,41,4,208,14
43 DATA173,7,21,24,105,16,141,7,21,144,3,238,8,2
1,173,7,21,56,233,8,141,7
44 DATA21,176,3,206,8,21,173,22,21,201,5,208,34,
165,2,41,8,208,14,173,7,21
45 DATA24,105,14,141,7,21,144,3,238,8,21,173,7,2
1,56,233,7,141,7,21,176,3
46 DATA206,8,21,173,22,21,201,6,208,34,165,2,41,
16,208,14,173,9,21,24,105
47 DATA32,141,9,21,144,3,238,10,21,173,9,21,56,2
33,16,141,9,21,176,3,206,10
48 DATA21,173,22,21,201,7,208,13,169,8,141,9,21,
141,10,21,169,154,141,12,21
49 DATA173,22,21,201,8,208,14,173,7,21,24,105,48
,141,7,21,144,3,238,8,21,173
50 DATA22,21,201,9,208,14,173,7,21,56,233,48,141
,7,21,176,3,206,8,21,173,22
51 DATA21,201,10,208,11,165,2,41,1,208,5,169,129
,141,11,212,173,21,201
52 DATA11,208,14,169,2,141,10,21,173,9,21,24,105
,5,141,9,21,173,22,21,201
53 DATA12,208,42,165,2,41,4,208,14,173,7,21,74,1
41,7,212,173,8,21,74,141,8
54 DATA12,165,2,41,6,208,16,173,7,21,74,74,141,
7,212,173,8,21,74,74,141,8
55 DATA12,173,22,21,201,13,208,42,165,2,41,4,20
8,14,173,9,21,74,141,9,212
56 DATA173,10,21,74,141,10,212,165,8,41,6,208,16
,173,9,21,74,74,141,9,212
57 DATA173,10,21,74,74,141,10,212,173,22,21,201,
14,208,8,169,8,141,9,21,141
58 DATA10,21,173,22,21,201,15,208,25,165,2,41,4,
208,11,169,3,141,10,21,238
59 DATA9,21,76,179,196,169,8,141,10,21,206,9,21,
173,22,21,201,16,208,8,169
60 DATA128,141,9,21,238,10,21,173,21,21,201,17,2
08,15,173,2,0,41,8,208,8,169
61 DATA0,141,0,212,141,1,212,173,23,21,201,1,208
,10,169,8,141,19,21,169,0
62 DATA141,20,21,173,23,21,201,2,208,10,169,10,1
41,19,21,169,0,141,20,21,173
63 DATA23,21,201,3,208,10,169,249,141,19,21,169,
233,141,20,21,173,23,21,201
64 DATA4,208,34,165,2,41,4,208,14,173,14,21,24,1
05,16,141,14,21,144,3,238
65 DATA15,21,173,14,21,56,233,8,141,14,21,176,3,
206,15,21,173,23,21,201,5
66 DATA208,34,165,2,41,8,208,14,173,14,21,24,105
,14,141,14,21,144,3,238,15
67 DATA21,173,14,21,56,233,7,141,14,21,176,3,206
,15,21,173,23,21,201,6,208
68 DATA34,165,2,41,16,208,14,173,16,21,24,105,32
,141,16,21,144,3,238,17,21
69 DATA173,16,21,56,233,16,141,16,21,176,3,206,1
7,21,173,23,21,201,7,208,13
70 DATA169,8,141,16,21,141,17,21,169,154,141,19,
21,173,23,21,201,8,208,14
71 DATA173,14,21,24,105,48,141,14,21,144,3,238,1
5,21,173,23,21,201,9,208,14
72 DATA173,14,21,56,233,48,141,14,21,176,3,206,1
5,21,173,23,21,201,10,208
73 DATA11,165,2,41,1,208,5,169,129,141,11,212,17
3,23,21,201,11,208,14,169
74 DATA2,141,17,21,173,16,21,24,105,5,141,16,21,
173,23,21,201,12,208,42,165
75 DATA2,41,4,208,14,173,14,21,74,141,7,212,173,
15,21,74,141,8,212,165,2,41
76 DATA6,208,16,173,14,21,74,74,141,7,212,173,15
,21,74,74,141,8,212,173,23
77 DATA21,201,13,208,42,165,2,41,4,208,14,173,16
,21,74,141,9,212,173,17,21
78 DATA74,141,10,212,165,8,41,6,208,16,173,16,21
,74,74,141,9,212,173,17,21
79 DATA74,74,141,10,212,173,23,21,201,14,208,8,1
69,8,141,16,21,141,17,21,173
80 DATA23,21,201,15,208,25,165,2,41,4,208,11,169
,3,141,17,21,238,16,21,76
81 DATA124,198,169,8,141,17,21,206,16,21,173,23,
21,201,16,208,8,169,128,141
82 DATA16,21,238,17,21,173,23,21,201,17,208,15,1
73,2,0,41,8,208,8,169,0,141
83 DATA14,212,141,15,212,76,49,234

```

```

<89> 84 FOR X=52480 TO52444:READ DT
85 POKE X,DT:NEXT X
<254> 86 DATA 1,1,2,1,0,36,59,17,161,129:REM
87 DATA 1,1,2,1,0,36,59,17,191,129:REM
<90> 88 DATA 1,1,2,1,0,36,59,17,191,129:REM
89 DATA 1,1,2,1,0,36,59,17,191,129:REM
<162> 90 DATA 1,1,2,1,0,36,59,17,191,129:REM
91 DATA 1,1,3,1,0,36,59,17,221,129:REM
<28> 92 DATA 58,138,5,192,17
93 DATA 69,157,5,12,17
<73> 94 DATA 65,181,5,12,17
95 DATA 58,138,5,24,17
<232> 96 DATA 65,181,5,24,17
97 DATA 69,157,5,24,17
<13> 98 DATA 78,36,5,96,17
99 DATA 49,58,5,48,17
<234> 100 DATA 43,219,5,72,17
101 DATA 39,18,5,72,17
<246> 102 DATA 58,138,4,192,17
103 DATA 69,157,4,12,17
<161> 104 DATA 65,181,4,12,17
105 DATA 58,138,4,24,17
<161> 106 DATA 65,181,4,24,17
107 DATA 69,157,4,24,17
<240> 108 DATA 78,36,4,96,17
109 DATA 49,58,4,48,17
<215> 110 DATA 43,219,4,72,17
111 DATA 39,18,4,72,17
<157> 112 DATA 0,0,0,0
113 FOR X=51712 TO51931:READ DT
115 POKE X,DT:NEXT X
<153> 116 DATA 1,1,2,1,0,36,59,9,191,129:REM
117 DATA 36,59,9,192,129:REM
<144> 118 DATA 36,59,9,192,129:REM
119 DATA 1,1,14,1,0,14,162,16,191,65
<146> 120 DATA 21,237,16,12,65
121 DATA 19,137,16,12,65
<222> 122 DATA 21,237,16,24,65
123 DATA 23,59,16,24,65
<210> 124 DATA 26,20,16,24,65
125 DATA 23,59,16,192,65
126 DATA 29,69,16,12,65
<12> 127 DATA 26,20,16,12,65
128 DATA 29,69,16,24,65
<78> 129 DATA 26,20,16,24,65
130 DATA 23,59,16,24,65
<30> 131 DATA 1,1,14,1,0,14,162,16,191,65
132 DATA 21,237,16,12,65
<49> 133 DATA 19,137,16,12,65
134 DATA 21,237,16,24,65
<168> 135 DATA 23,59,16,24,65
136 DATA 26,20,16,24,65
<4> 137 DATA 23,59,16,192,65
138 DATA 29,69,16,12,65
<163> 139 DATA 26,20,16,12,65
140 DATA 29,69,16,24,65
<24> 141 DATA 26,20,16,24,65
142 DATA 23,59,16,24,65
<252> 143 DATA 1,1,14,1,0,14,162,16,191,65
144 DATA 21,237,16,12,65
<30> 145 DATA 19,137,16,12,65
146 DATA 21,237,16,24,65
<73> 147 DATA 23,59,16,24,65
148 DATA 26,20,16,24,65
<161> 149 DATA 23,59,16,192,65
150 DATA 29,69,16,12,65
<39> 151 DATA 26,20,16,12,65
152 DATA 29,69,16,24,65
<159> 153 DATA 26,20,16,24,65
154 DATA 23,59,16,24,65
<27> 155 DATA 0,0,0,0
157 FOR X=50944 TO51218:READ DT
159 POKE X,DT:NEXT X
<10> 159 DATA 1,1,2,1,0,5,207,6,23,65:REM
160 DATA 1,1,14,1,0,5,207,6,11,65
<173> 161 DATA 1,1,14,1,0,5,207,6,23,65
162 DATA 1,1,14,1,0,5,207,6,11,65
<42> 163 DATA 1,1,14,1,0,5,207,6,23,65
164 DATA 1,1,14,1,0,5,207,6,11,65
<10> 165 DATA 1,1,14,1,0,5,207,6,23,65
166 DATA 1,1,14,1,0,5,207,6,11,65
<185> 167 DATA 1,1,14,1,0,5,207,6,23,65
168 DATA 1,1,14,1,0,5,207,6,11,65
<7> 169 DATA 1,1,14,1,0,5,207,6,23,65
170 DATA 1,1,14,1,0,5,207,6,11,65
<107> 171 DATA 7,81,15,24,65

```



```

172 DATA 8,55,15,24,65
173 DATA 8,180,15,24,65
174 DATA 1,1,14,1,0,8,55,6,23,65:REM
175 DATA 1,1,14,1,0,8,55,6,11,65
176 DATA 1,1,14,1,0,8,55,6,23,65
177 DATA 1,1,14,1,0,8,55,6,11,65
178 DATA 1,1,14,1,0,8,55,6,23,65
179 DATA 1,1,14,1,0,8,55,6,11,65
180 DATA 1,1,14,1,0,8,55,6,23,65
181 DATA 1,1,14,1,0,8,55,6,11,65
182 DATA 1,1,14,1,0,8,55,6,23,65
183 DATA 1,1,14,1,0,8,55,6,11,65
184 DATA 1,1,14,1,0,8,55,6,23,65
185 DATA 1,1,14,1,0,8,55,6,11,65
186 DATA 9,196,15,24,65
187 DATA 10,247,15,24,65
188 DATA 11,158,15,24,65
189 DATA 0,0,0,0,0
190 IF PEEK(186)<8 THEN 240
200 PRINT "CLEAR":A$=CHR$(34)
210 PRINT "DOHN2)LOAD:A$="PRG5"A$,"B"
220 PRINT "DOHN4)RUN"
230 PRINT "HOME)":POKE 631,13:POKE 632,13:POKE 1
98,2:END
240 POKE 631,13:POKE 198,1:END
ENDE DES LISTINGS

```

Teil 5

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

```

10 POKES3272,PEEK(53272)AND2410R12:LI=5:LE=1:POK
E 53276,0
20 GOSUB 10000:POKE 53281,0:POKE 53280,0
30 PRINT "HOME YELLOW RIGHT6)PARAVISION(SPACE)CO
MPANY(SPACE)PRESENTS:"
40 PRINT "GREEN DOWN RIGHT13)ULRICH(SPACE)MCHL'S
"
50 PRINT "DOWN16 RIGHT6 YELLOW)\86(SPACE)BY(SPA
CE)PARAVISION(SPACE)COMPANY"
55 PRINT "DOWN RIGHT12 GREEN)SPACE(SPACE)TO(SPA
CE)PLAY(SPACE)!"
60 SYS 2069,116,101,187,101,110,110,134,110,158,
110,182,110,206,110,230,110,255
70 SYS 2227,1,1,12,12,12,12,12
80 SYS 2178,166,166,166,166,161,162,163,164,165
85 FOR X=2024 TO 2032:POKE X,48:NEXT X
86 POKE 54296,15:SYS 49152
90 PRINT "HOME)":GET A$:IF A$<>"(SPACE)" THEN BY
S 2048:GOTO 90
91 PRINT "CLEAR)":POKE53269,0:GOSUB10000:PRINTTA
B(20)"YELLOW)SCORE:(SPACE)00000000(U22)"
92 FOR X=2025 TO 2032:POKE X-14,PEEK(X):NEXT X
93 IFLI>1THENFOR X=2 TO LI:POKE 1985+X*2,29:POKE
56257+X*2,5:NEXT X
94 POKE 2685,0
101 POKE 54277,0:POKE 54278,248:POKE 54277+7,0:P
OKE 54278+7,248:POKE 54296,15
115 SYS 2600,137,148,147,149,147,150,155,137,148
,147,149,147,150,155
130 POKE 53281,0:POKE 53280,0
140 SYS 2289,253
165 POKE 54296,0:GOSUB 500:TR=2685:POKE TR,0:POK
E 53278,0
170 POKE 54296,15:FOR S=1TO6L:SYS3240:IFPEEK(TR)=
255THEN185
180 NEXTS:POKE 54296,0:SYS 2426
181 FOR X=2011 TO 2019:POKE X+14,PEEK(X):NEXT X:
LE=LE+1:GOTO 91
185 FOR X=2011 TO 2019:POKE X+14,PEEK(X):NEXT X
190 LI=LI-1:POKE53269,1:POKE54276,129:POKE54283,
129:POKE2040,156
200 FOR WT=0 TO 500:NEXT WT:POKE 54296,0:IF LI>0
THEN 91
210 POKE 53269,0:PRINT "HOME GREEN DOWN10 RIGHT1
5)GAME(RIGHT)OVER"
220 FOR WT=0 TO 500:NEXT WT:RUN
500 REM PREPARE LEVEL
510 ON LE GOSUB 20000,20500,21000,21500,22000,22
500,23000,23500,24000
520 RETURN
9999 END
10000 REM STERNE AUFBAUEN
10001 PRINT "CLEAR WHITE SQ SH SE SR ST SY SU SI
SQ SP SL SK SF SG SH SJ SD SS SA SE SX SC SV SB

```

```

<119> SN SM SR SG SY SH SF SB SH ST SG SJ SU SR SU SB
<82> )"
10002 PRINT "UP SR SX SF SS SU SA SV SJ SP SQ SM
SL SG SH SI SK SE ST SB SF SY SD SM SC SO SH SS
SH SA SI SG SC ST SU SH SK SV SS SV SC)"
10003 PRINT "UP SS SY SG ST SV SB SH SK SQ SR SH
SM SH SI SJ SL SF SU SC SG SA SE SX SD SP SQ ST
SI SB SJ SH SD SJ SV SI SL SM ST SH SD)"
10004 PRINT "UP ST SA SH SU SH SC SX SL SR SS SO
SH SI SJ SK SM SG SV SD SH SB SF SY SE SQ SP SU
SJ SC SK SI SE SK SM SJ SH SX SU SX SE)"
10005 PRINT "UP SU SB SI SV SX SD SY SM SS ST SP
SO SJ SK SL SM SH SM SE SI SC SG SA SF SR SQ SV
SK SD SL SJ SF SL SX SK SM SJ SV SY SF)"
10006 PRINT "UP SV SC SJ SM SY SE SA SH ST SU SQ
SP SK SL SM SO SI SX SF SJ SD SH SB SG SS SR SH
SL SE SM SK SG SM SY SL SO SA SH SA SG)"
10007 PRINT "UP SM SD SK SX SA SF SO SO SU SV SR
SQ SL SM SM SP SJ SY SG SK SE SI SC SH ST SS SX
SH SF SM SL SM SM SA SM SP SB SX SB SH)"
10008 PRINT "UP SX SE SL SY SG SJ SD SI SU ST SY
SR SM SM SO SQ SK SA SM SL SF SJ SD SI SU ST SY
SH SO SO SM SI SO SB SM SO SC SY SC SI)"
10009 PRINT "UP SY SF SM SA SM SC SY SV SY SF)"
SS SM SO SP SR SL SB SI SM SG SK SE SJ SV SU SA
SO SM SP SM SJ SP SC SO SR SO SA SO SJ)"
10010 PRINT "UP SA SO SM SB SO SI SE SR SX SY SU
ST SO SP SO SS SM SC SJ SM SH SL SF SK SM SV SB
SP SI SO SO SK SO SD SP SS SE SB SE SK)"
10011 PRINT "UP SB SH SO SC SE SJ SF SS SY SA SV
SU SP SO SR ST SM SO SK SO SI SM SO SL SX SH SC
SQ SJ SR SP SL SR SE SQ ST SF SC SF SL)"
10012 PRINT "UP SC SI SP SO SF SK SO ST SA SB SH
SY SQ SR SS SU SO SE SL SP SJ SM SH SM SX SX SD
SR SK SS SO SM SS SF SR SU SO SO SM)"
10013 PRINT "UP SD SJ SO SE SO SL SH SJ SB SC SX
SH SR SS ST SV SP SF SM SO SK SO SI SM SA SY SE
SS SL ST SR SM ST SO SS SV SH SE SH)"
10014 PRINT "UP SE SK SR SF SH SM SI SV SC SD SY
SX SS ST SU SM SO SG SM SR SL SP SJ SO SB SA SE
ST SH SU SO SU SH ST SM SI SF SI SO)"
10015 PRINT "UP SF SL SS SO SI SM SJ SM SO SE SA
SY ST SU SV SX SR SH SO SS SM SO SK SP SC SB SO
SU SM ST SP SV SI SU SX SJ SO SP)"
10016 PRINT "UP SO SM ST SH SJ SO SK SX SE SF SB
SA SU SV SH SY SS SI SP ST SM SR SL SO SO SC SH
SV SO SH SU SO SM SJ SV SY SK SH SK SQ)"
10017 PRINT "UP SH SM SI SK SP SL SY SF SO SC
SB SV SH SX SA ST SJ SO SU SO SS SM SR SE SO SI
SH SP SX SV SR SX SK SM SA SL SI SL SR)"
10018 PRINT "UP SI SO SV SJ SL SO SM SA SO SM SD
SC SM SX SY SB SU SK SR SV SP ST SM SS SF SE SJ
SX SQ SY SM SS SY SL SX SB SM SJ SM SS)"
10019 PRINT "UP SJ SP SM SK SM SR SM SH SI SE
SD SX SY SA SC SV SL SS SM SO SO ST SO SE SK
SY SR SA SX ST SA SM SY SC SM SK SM ST)"
10020 PRINT "UP SK SO SX SL SM SS SO SC SI SJ SF
SE SY SA SB SD SM SM ST SX SR SV SP SU SH SO SL
SA SS SB SJ SU SB SM SA SO SO SL SO SU)"
10021 PRINT "UP SL SR SY SM SO ST SP SD SJ SK SO
SF SA SB SC SE SX SM SY SS SM SO SV SI SM SM
SB ST SC SA SV SC SO SB SE SP SM SP SV)"
10022 PRINT "UP SM SS SA SM SP SU SO SE SK SL SM
SG SB SC SO SF SY SO SV SA ST SX SM SJ SL SM
SC SU SD SB SM SD SP SC SF SO SM SO SM)"
10023 PRINT "UP SM ST SB SO SO ST SR SF SL SM SI
SH SC SO SE SG SA SP SM SB SU SY SS SX SK SJ SO
SD SV SE SC SK SE SO SD SG SR SO SR SX)"
10024 PRINT "UP SO SU SC SP SR SY SS SO SM SM SJ
SI SD SE SF SH SB SO SX SC SV SA ST SY SL SK SP
SE SM SF SO SY SF SR SE SH SS SP SS)"
10025 RETURN
20000 REM LEVEL I
20010 SYS 2178,136,137,145,146,147,141,142,143
20015 SYS 2227,13,7,12,12,15,15,14,13
20016 SYS 2276,5,3
20020 SYS 2069,150,185,150,150,50,0,80,30,110,60
,140,90,170,120,200,200,203
20030 SYS 2327,0,250,1,1,1,1,1,1,1,1,1,1
20040 BL=750:RETURN
20050 REM LEVEL II
200510 SYS 2178,136,137,145,146,147,141,142,143
200515 SYS 2227,10,5,1,1,5,5,14,13
200516 SYS 2276,2,14
200520 SYS 2069,150,185,150,150,50,0,80,30,110,60
,140,90,170,120,200,200,175

```

Weiter auf Seite 46.

Der neue Checksummer für C16/116/plus 4

Endlich ist es soweit, der Checksummer OV 2.0 ist fertig!

Obwohl schon durch die alte Checksummer-Version OV1.0 das Eingeben von Listings erheblich vereinfacht wurde, gab es immer noch kleine Mängel in der Bedienung. Insbesondere die Tatsache, daß Listings in einem "Stück" eingegeben werden mußten, wirkte sich sehr nachteilig aus. Programme die am Vortag begonnen wurden, mußten später ohne Checksummer vervollständigt werden. Lediglich die Besitzer eines Floppylaufwerkes kannten diese Probleme nicht. Ein weiterer Nachteil trat beim nachträglichen Editieren von Programmzeilen auf, welche länger als 40 Zeichen waren.

Alle diese Nachteile sind in der neuen Version OV2.0 nicht mehr vorhanden. Jetzt können Programme, welche bereits eingegeben wurden, nachträglich mit dem Checksummer kontrolliert werden. Um dies zu bewerkstelligen läßt man zuerst den Checksummer von Kassette bzw. Diskette, startet ihn mit RUN und läßt das zu prüfende Basisprogramm nach. Jetzt wird der Checksummer mit SYS 4097 aktiviert. Um eine Zeile zu überprüfen, fährt man mit dem Cursor auf diese und betätigt die RETURN-Taste. Danach vergleicht man die Checksumme am linken oberen Bildschirmrand mit der abgedruckten Checksumme im Heft. Stimmt diese Summe nicht überein, so ist die eingegebene Zeile fehlerhaft und sollte korrigiert werden. Beachten Sie bitte, daß die Checksummen der neuen Version identisch mit der alten Version sind, diese jedoch jetzt formatiert (mit führenden Nullen) ausgegeben werden. Das bedeutet die Checksumme <008> entspricht der Checksumme <8>.

Die Arbeitsweise und Funktion des Checksummers ist bis auf die erwähnten Merkmale identisch geblieben. Da der neue Checksummer jedoch in einem anderen Bereich liegt (4097 bis 4287) ändert sich der Basicanfang sowie die Checksummer-Startadresse. Die neue Startadresse lautet nun SYS 4097. Vor einem LOAD bzw. SAVE Befehl sollte der Checksummer jedoch weiterhin mit SYS 62158:SYS 33047 abgeschaltet werden.

Da der Checksummer den Basicanfang versetzt, stehen dem Anwender 255 Bytes weniger Speicher zur Verfügung. In der Regel macht sich dies jedoch nicht nachteilig bemerkbar, da selten der gesamte Speicher benötigt wird. Sollte dies jedoch einmal vorkommen, so sollte man das Programm nach der Eingabe mittels Checksummer abspeichern, danach den RESET-Taster betätigen und das Programm wieder einladen. Jetzt kann das Programm ohne Probleme gestartet werden.

Achtung!

Wenn Sie den neuen Checksummer eingegeben haben, so sollten Sie diesen unbedingt vor einem Start abspeichern, da er sich automatisch löscht. Um den Checksummer möglichst fehlerfrei eingeben zu können, sollten Sie den alten Checksummer benutzen.

Für alle, die nicht gerne abtippen, ist der Checksummer unter folgender Bestellnummer zu beziehen:

OV 20 K Kassette 10 DM
OV 20 D Diskette 15 DM

Die alte Version OV1.0 ist nicht mehr erhältlich.

Frank Brall

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

```

1 REM *****
2 REM *
3 REM * C16 CHECKSUMMER OV 2.0 *
4 REM *
5 REM * COPYRIGHT BY FRANK BRALL *
6 REM *
7 REM * (C) 15.12.86 TRONIC VERLAG *
8 REM *
9 REM *****
10 DIM H(75)
20 PRINT "CLEAR DOWN SPACE)***SPACE)CHECKSUMMER
(SPACE)OV(SPACE)2.0(SPACE)***SPACE)C16/PLUS/4"
30 PRINT "SPACE) (C) (SPACE)FRANK (SPACE)BRALL (SPA
CE)1987(SPACE)/(SPACE)SONTRA"
40 PRINT
50 PRINT "CHECKSUMMER(SPACE)EIN(SPACE)=(SPACE)SVS
(SPACE)4097"
60 PRINT "CHECKSUMMER (SPACE)AUS (SPACE)=(SPACE)SVS
(SPACE)62158:SYS(SPACE)33047"
70 PRINT "PRINT " (SPACE)3)ACHTUNG (SPACE)! (SPACE)VOR
(SPACE)DEM(SPACE)LADEN(SPACE)ODER"
80 PRINT " (SPACE)3)SPEICHERN(SPACE)VON(SPACE)PROGR
AMMEN(SPACE)IST(SPACE)DER"
90 PRINT " (SPACE)3)CHECKSUMMER(SPACE)AUSZUSCHALTEN
(SPACE)!"
100 PRINT "PRINT "BITTE (SPACE)BEACHTEN(SPACE)SIE (S
PACE)AUCH(SPACE)DIE (SPACE)HINWEISE"
110 PRINT "IN(SPACE)DEN(SPACE)HEFTEN(SPACE)COMPUT
E (SPACE)MIT, (SPACE)COMPUTRONIC"
120 PRINT "SOWIE (SPACE)DEN(SPACE)AKTUELLEN(SPACE)
SONDERHEFTEN(SPACE)!"
130 PRINT "DIESER(SPACE)CHECKSUMMER (SPACE)IST (SPA
CE)VOLL (SPACE)KOMPATIBEL"
140 PRINT "ZUR (SPACE)VERSION(SPACE)1.0, (SPACE)ERL
AUBT(SPACE)JEDOCHE"
150 PRINT "ZUSÄTZLICH(SPACE)DAS (SPACE)NACHLADEN(
SPACE)VON(SPACE)PROGRAMMEN"
160 PRINT:PRINT
200 PRINT "HABEN(SPACE)SIE (SPACE)DEN(SPACE)CHECKS
UMMER(SPACE)GESPEICHERT (SPACE)?"
210 GET E$:IF E$=" " THEN 210
220 IF E$="N" THEN PRINT "DANN(SPACE)SAVEN(SPACE)
SIE (SPACE)IHN(SPACE)JETZT (SPACE)!" :STOP
1000 PRINT "BITTE (SPACE)WARTEN(SPACE)!"
60000 FOR I=0TO9
60010 H(48+I)=I:H(65+I)=I+10:NEXT
60020 FOR I= 4097 TO 4287 :READA$
60030 H=ASC(LEFT$(A$,1))
60040 L=ASC(RIGHT$(A$,1))
60050 D=H(H)*16+H(L):S=S+D:POKE1,D
60060 A=A+1:IFA(20)THENNEXTA:=-1
60070 READV:Z=Z+1:IFV=STHEN60085
60080 PRINT "DATAFEHLER(SPACE)IN(SPACE)ZEILE (SPAC
E):":60200+Z:END
60085 IFA(0)THEN60100
60090 S=A:A=0:NEXT
60100 PRINT "CLEAR)POKE (SPACE)44,17:POKE (SPACE)1
7*256,0:NEW:"
60110 PRINT "DOHN3)CHECKSUMMER (SPACE)IST (SPACE)E
INGELESEN(SPACE)!"
60120 PRINT "DOHN)START (SPACE)MIT: (SPACE2)SVS (SP
ACE)4097"
60130 POKE 239,2:POKE 1319,19:POKE 1320,13:END
60201 DATA A9,0C,8D,02,03,A9,10,8D,03,03,60,A2,F
F,86,3A,20,5A,8B,86,3B, 1815
60202 DATA 84,3C,20,73,04,AA,F0,EF,90,09,20,53,B
9,20,79,04,4C,D7,8B,20, 2018
60203 DATA 3E,8E,20,53,89,84,0B,A9,00,8D,8B,10,B
D,BC,10,8D,8D,10,18,A5, 1992
60204 DATA 14,65,15,8D,8D,10,A0,FF,CB,B1,3B,F0,2
C,C9,22,D0,0A,AD,8B,10, 2452
60205 DATA 49,01,8D,8B,10,B1,3B,AE,8B,10,D0,04,C
7,20,F0,E4,EE,BC,10,AE, 2560
60206 DATA BC,10,18,B1,3B,6D,8D,10,8D,8D,10,CA,D
0,F4,4C,45,10,A9,BC,8D, 2437
60207 DATA 00,0C,A9,8E,8D,04,0C,A9,00,AE,8D,10,B
5,62,86,63,A2,90,3B,20, 1934
60208 DATA CE,A2,20,71,A4,20,74,9B,20,4E,9C,A2,0
3,AB,8B,20,B0,04,09,80, 2064
60209 DATA 9D,00,0C,CA,8B,10,F4,A9,B0,E0,00,F0,0
4,9D,00,0C,CA,D0,FA,86, 2545
60210 DATA FF,4C,36,87,00,00,00,00,FF,CD, 980
ENDE DES LISTINGS

```


Interruptprogrammierung auf dem Commodore 64

Artikelserie von Daniel Durstewitz und Waldemar Raaz

Teil 5

Hier nun die Fortsetzung des Kurses, den wir einen Monat aussetzen mußten.

Als erstes drucken wir hier das 2. Listing ab, welches sich noch auf den letzten Teil des Kurses bezieht.

Danach geht es weiter mit dem Rasterzeilen IRQ.

Der Rasterzeilen-IRQ

Ungeahnte Perspektiven eröffnet aber erst der Rasterzeilen-IRQ. Mit ihm kann man so schöne Dinge tun, wie Grafik und Text mischen (d.h. einen Grafikbildschirm mit Textzeilen erzeugen) oder einen mehrfarbigen Hintergrund und Rahmen gestalten. Auch mehr als acht Sprites lassen

sich realisieren. Jetzt werden Sie sicherlich sagen, Rasterzeilen ist ja ein wunderschönes Wort, aber was bedeutet es? Zur Klärung ist ein kleiner technischer Exkurs notwendig. Der Bildschirm besteht aus einer Lochrasterplatte mit einer fluoreszierenden Schicht, die durch einen Elektronenstrahl punktuell zum Leuch-

ten angeregt werden kann. Dieser Elektronenstrahl wird am hinteren Ende der Bildröhre erzeugt und von zwei Ablenkplatten-Paaren (im Prinzip Kondensatoren) von oben an Zeile für Zeile, Punkt für Punkt über die ganze Lochrasterplatte gesteuert, wo er, je nach den am UHF-Antenneneingang des Fernsehers anliegenden Informationen, die in unserem Falle vom Video Interface Chip (VIC) des C 64 stammen, einen einzelnen Rasterpunkt zum Leuchten anregt oder übergeht. Unter Rasterzeilen sind also die einzelnen Punktzeilen dieser Lochrasterplatte zu verstehen.

Die meisten Fernsehgeräte verfügen über 625 Rasterzeilen. Da der C 64 jedoch aus Geschwindigkeitsgründen nur Halbbilder erzeugt, d.h. jede zweite Rasterzeile übergeht (wenn Sie sich das Bild einmal genau anschauen, werden Sie feststellen können, daß jede zweite Rasterzeile schwarz ist und bleibt), handelt es sich für uns praktisch nur um etwa 312 Rasterzeilen, wovon jedoch noch ein paar Rasterzeilen am oberen und unteren Bildschirmrand »verschluckt« werden. Die Nummer der ersten und der letzten Rasterzeile variiert also etwa von Fernsehgerät zu Fernsehgerät (bzw. Monitor), meist liegt jedoch der Rasterzeilenanfang etwa bei Zeile 18 und das Rasterzeilenende ungefähr bei Zeile 280. Oben und unten werden dabei noch etwa jeweils 30 Rasterzeilen vom Bildschirmrahmen beansprucht, so daß der für den Benutzer verfügbare Bildschirmraum durchschnittlich von Zeile 48 bis Zeile 250 liegt. Da der VIC nun ein sehr benutzerfreundlicher Chip ist und keine Geheimnisse vor uns hat, teilt er uns über sein Register 18 (Adresse \$D012, dez. 53266) mit, welche Rasterzeile gerade erstellt. Falls dieser Wert größer als 255 wird, wird Bit 7 im Register 17 (Adresse \$D011, dez. 53265) gesetzt. Dieses Bit stellt also das Übertragsbit aus Register 18 dar. Nun wird der gesamte Bildschirm innerhalb von nur etwa 20 Millisekunden (0,02 s) aufgebaut, d.h. eine einzelne Rasterzeile innerhalb von nur etwa 0,08 Millisekunden (oder 0,00008 s). Ein Maschinensprachebefehl braucht aber mindestens »schon« 0,005 Millisekunden (0,000005 s) zur Ausführung (pro Rasterzeile können also bestenfalls etwa 6 bis 10 Maschinensprachebefehle ab-

gearbeitet werden), so daß eine gezielte Abfrage innerhalb längerer Programme auch in Maschinensprache kaum möglich ist, wenn da nicht die Möglichkeit wäre, bei einer bestimmten Rasterzeile einen IRQ auszulösen (einen einzelnen Rasterpunkt anzusprechen, ist allerdings wegen der viel zu kurzen Ausführungszeiten vollkommen unmöglich, so daß diese Möglichkeit vom VIC auch gar nicht erst vorgesehen ist). Dazu muß zunächst im IMR das für einen Rasterzeilen-IRQ zuständige Bit (also Bit 0) auf 1 gesetzt werden.

Nun kann durch Lesen des Registers 18 nicht nur festgestellt werden, in welcher Rasterzeile sich der VIC mit der Ausführung gerade befindet, sondern es kann auch durch Hineinschreiben in dieses Register diejenige Rasterzeile bestimmt werden, bei der ein IRQ ausgelöst werden soll. Ist dieser Wert größer als 255, muß noch Bit 7 im Register 17 gesetzt werden (s.o.). Durch das Schreiben in die Register 17 und 18 läßt sich der VIC übrigens nicht irritieren; nach wie vor erhalten Sie mit dem Lesen der Register die Rasterzeile, die der VIC gerade erstellt (im Normalfall also einen anderen Wert als den hineingeschriebenen).

Damit wäre also geklärt, was ein Rasterzeilen-IRQ ist und wie man ihn auslöst. Wie man diese Möglichkeit geschickt anwendet, soll an vier Beispielprogrammen gezeigt werden.

Mehrfarbiger Hintergrund und Rahmen

Da wäre zunächst einmal Listing 3, das einen dreifarbenen Hintergrund und Rahmen erzeugt, und das geht so: In der Einschaltoutine wird zunächst die neue IRQ-Adresse festgelegt, im IMR das Rasterzeilen-IRQ-Bit gesetzt und die Rasterzeile festgelegt, bei der der erste IRQ erfolgen soll (in unserem Fall ist das sicherheitshalber Rasterzeile 0, noch außerhalb des sichtbaren Bereiches). In der neuen IRQ-Routine wird als erstes das IRR gelöscht und beim aus dem IRR geladenen Wert getestet, ob das Rasterzeilen-IRQ-Bit (Bit 0) gesetzt und damit der Auslöser des IRQ eine bestimmte Rasterzeile ist. Falls nicht, wurde der IRQ von dem Timer A der CIA #1 ausgelöst und es folgt ein Sprung zur normalen IRQ-Routine (\$

EA31). Diesem Sprungbefehl vorangestellt ist der Assemblerbefehl CLI, um auch während der Ausführung der ROM-IRQ-Routine einen Rasterzeilen-IRQ zuzulassen. Würde man auf den Befehl CLI verzichten, d.h. also während der Ausführung der ROM-IRQ-Routine einen Rasterzeilen-IRQ nicht zulassen, könnte es schnell passieren, daß eine Rasterzeile, bei der eigentlich ein IRQ ausgelöst werden sollte, übergangen wird und damit wie in unserem Beispiel »Blau-Rot-Grün« ein erwarteter Farbenwechsel kurzfristig nicht stattfindet. Das hat ein leichtes Blinken

des Bildes zur Folge, das sich stark erhöht, sobald irgendwelche Eingaben über die Tastatur gemacht werden (wir erinnern uns: auch die Tastatur wird innerhalb der ROM-IRQ-Routine kontrolliert).

Bevor aber ein Interrupt mit CLI wieder ermöglicht wird, muß zunächst die IRQ-Ursache, nämlich ein Überlauf des Timers A der CIA #1, im ICR der CIA #1 gelöscht werden, da sonst sofort nach dem Befehl CLI wieder ein IRQ ausgelöst und es schließlich zum Absturz des Computers kommen würde. Als nächstes wird getestet, bei welcher Raster-

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

```

100 REM * LAUFSCHRIFT IM INTERRUPT *
105 REM
110 REM DIESES PROGRAMM ERZEUGT EINE
115 REM LAUFSCHRIFT IN DER ERSTEN
120 REM ZEILE UNABHÄNGIG VON ANDEREN
125 REM LAUFFENDEN PROGRAMMEN.
130 :
135 :
140 FORN=49152TO49265:READD:Z=Z+D:POKEN,D: NEXT
145 IFZ<>11471 THENPRINT"(DOHN)FEHLER(SPACE)IN(SPACE)DEN(SPACE)DATA-ZEILEN!":END
150 PRINT"(DOHN)AKTIVIEREN(SPACE)MIT(SPACE)SYS($SPACE)49152."
155 PRINT"ABSCHALTEN(SPACE)MIT(SPACE)SYS(SPACE)49170.":END
160 :
165 :
170 DATA120 : EIN SEI
E) INTERRUPT(SPACE)VERHINDERN
175 DATA169,10 : LDA #$0A
E) ZAEHLER(SPACE)FUER(SPACE)VERZOGERUNG
180 DATA133,251 : STA $FB
E) DER(SPACE)LAUFSCHRIFT(SPACE)SETZEN
185 DATA169,39 : LDA #$27
E) STARTWERT(SPACE)FUER(SPACE)KOPIEREN
190 DATA133,252 : STA $FC
E) DER(SPACE)TEXTTABELLE(SPACE)SETZEN
195 DATA169,34 : LDA #$19-LD
E) NEUE(SPACE)IRQ-ADRESSE(SPACE)LO-BYTE
200 DATA162,192 : LDX #$19-HI
E) NEUE(SPACE)IRQ-ADRESSE(SPACE)HI-BYTE
205 DATA208,8 : BNE $SET
E) NEUE(SPACE)ADRESSE(SPACE)SETZEN
210 DATA234 : NOP
215 DATA234 : NOP
220 DATA234 : NOP
225 DATA120 : AUS SEI
E) INTERRUPT(SPACE)VERHINDERN
230 DATA169,49 : LDA #$31
E) ALTE(SPACE)IRQ-ADRESSE(SPACE)LO-BYTE
235 DATA162,234 : LDX #$EA
E) ALTE(SPACE)IRQ-ADRESSE(SPACE)HI-BYTE
240 DATA141,20,3 : SET STA $0314
E) IRQ-ADRESSE
245 DATA142,21,3 : STX $0315
E) SETZEN
250 DATA88 : CLI
E) INTERRUPT(SPACE)WIEDER(SPACE)ERMOEGLICHEN
255 DATA96 : RTS
260 DATA234 : NOP
265 DATA234 : NOP
270 DATA234 : NOP
275 DATA198,251 : IRQ DEC $FB
E) LAUFSCHRIFT(SPACE)WEITER
280 DATA16,30 : BPL $ENDE
E) VERSCHIEBEN(SPACE)?
285 DATA169,10 : LDA #$0A
E) JA, (SPACE)VERZOGERUNGSZAEHLER(SPACE)WIEDER
290 DATA133,251 : STA $FB
E) AUF(SPACE)STARTWERT(SPACE)ZURUECKSETZEN
295 DATA164,252 : LDY $FC
E) STARTWERT(SPACE)FUER(SPACE)LAUFSCHRIFT
300 DATA162,39 : LDX #$27
305 DATA189,74,192 : FILL LDA $TEXT,X
E) TEXTTABELLE
310 DATA153,0,4 : STA $0400,Y
E) IN
315 DATA136 : DEY
E) BILDSCHIRMBEREICH
320 DATA16,2 : BPL $WEIT
E) KOPIEREN
325 DATA160,39 : LDY #$27
330 DATA202 : WEIT DEX
335 DATA16,242 : BPL $FILL
340 DATA198,252 : DEC $FC
E) ZEILENSTARTWERT(SPACE)UM(SPACE)EINS(SPACE)NACH(SPACE)LINKS
345 DATA208,4 : BNE $ENDE
E) ZEILE(SPACE)VOLLSTAENDIG(SPACE)VERSCHOBEN(SPACE)?
350 DATA169,39 : LDA #$27
E) JA, (SPACE)DANN(SPACE)STARTWERT(SPACE)FUER
355 DATA133,252 : STA $FC
E) ZEILE(SPACE)WIEDER(SPACE)NEU(SPACE)SETZEN
360 DATA76,49,234 : ENDE JMP $EA31
E) ZUR(SPACE)ALTEN(SPACE)IRQ-ROUTINE
365 DATA234 : NOP
370 DATA234 : NOP
375 DATA234 : NOP
380 :
385 TEXTTABELLE
390 :
395 DATA90,87,42,90,87,42,32,32,12,1,21,6,19,3,8,18,9,6,20,32,9,13,32,9,14
400 DATA20,5,18,18,21,16,20,32,32,90,87,42,90,87,42
405 :
410 + 1986 BY DANIEL DURSTEWITZ +

```

Listing 2

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

```

100 REM * GRUEN-ROT-BLAU *
105 REM
110 REM DIESES PROGRAMM ERZEUGT EINEN
115 REM DREIFARBIGEN HINTERGRUND UND
120 REM RAHMEN DURCH ZWEI
125 REM RASTERZEILEN-IRQ'S.
130 :
135 :
140 FORN=49152TO49265:READD:Z=Z+D:POKEN,D: NEXT
145 IFZ<>12399 THENPRINT"(DOHN)FEHLER(SPACE)IN(SPACE)ACE)DEN(SPACE)DATA-ZEILEN!":END
150 PRINT"(DOHN)BLACK)AKTIVIEREN(SPACE)MIT(SPACE)SYS(SPACE)49152."
155 PRINT"ABSCHALTEN(SPACE)MIT(SPACE)SYS(SPACE)49177.":END
160 :
165 :
170 DATA120 : EIN SEI
E) INTERRUPT(SPACE)VERHINDERN
175 DATA169,127 : LDA #$7F
E) RASTERZEILEN-UEBERTRAG
180 DATA45,17,208 : AND $D011
E) LOESCHEN
185 DATA141,17,208 : STA $D011
190 DATA169,0 : LDA #$00
E) ERSTEN(SPACE)IRQ(SPACE)BEI
195 DATA141,18,208 : STA $D012
E) RASTERZEILE(SPACE)0
200 DATA169,46 : LDA #$19-LD
E) NEUE(SPACE)IRQ-ADRESSE(SPACE)LO-BYTE
205 DATA162,192 : LDX #$19-HI
E) NEUE(SPACE)IRQ-ADRESSE(SPACE)HI-BYTE
210 DATA160,1 : LDY #$01
E) FLAG(SPACE)FUER(SPACE)RASTERZEILEN-IRQ(SPACE)EIN
215 DATA208,10 : BNE $SET
E) NEUE(SPACE)ADRESSE(SPACE)UND(SPACE)FLAG(SPACE)SETZEN
220 DATA234 : NOP
225 DATA234 : NOP
230 DATA234 : NOP
235 DATA120 : AUS SEI
E) INTERRUPT(SPACE)VERHINDERN
240 DATA169,49 : LDA #$31
E) ALTE(SPACE)IRQ-ADRESSE(SPACE)LO-BYTE
245 DATA162,234 : LDX #$EA
E) ALTE(SPACE)IRQ-ADRESSE(SPACE)HI-BYTE
250 DATA160,0 : LDY #$00
E) FLAG(SPACE)FUER(SPACE)RASTERZEILEN-IRQ(SPACE)AUS
255 DATA141,20,3 : SET STA $0314
E) IRQ-ADRESSE
260 DATA142,21,3 : STX $0315
E) UND(SPACE)FLAG(SPACE)FUER
265 DATA140,26,208 : STY $D01A
E) RASTERZEILEN-IRQ(SPACE)EIN/AUS(SPACE)SETZEN
270 DATA88 : CLI
E) INTERRUPT(SPACE)WIEDER(SPACE)ERMOEGLICHEN

```

Listing 3, Teil 1


```

275 DATA96 : RTS " <122>
280 DATA234 : NOP " <58>
285 DATA234 : NOP " <63>
290 DATA234 : NOP " <68>
295 DATA173,25,208 : IRQ LDA $D019 " (SPACE
E) INTERRUPT (SPACE) REQUEST (SPACE) REGISTER (SPACE) (
IRR) <129>
300 DATA141,25,208 : STA $D019 " (SPACE
E) LOESCHEN <191>
305 DATA41,1 : AND #$01 " (SPACE
E) HANDELT (SPACE) ES (SPACE) SICH (SPACE) UM <156>
310 DATA208,7 : BNE $RASZ " (SPACE
E) EINEN (SPACE) RASTERZEILEN-IRQ (SPACE) ? <112>
315 DATA173,13,220 : LDA $D00D " (SPACE
E) NEIN, (SPACE) IRQ-URSACHE (SPACE) IM (SPACE) ICR (SPA
CE) LOESCHEN <41>
320 DATAB8 : CLI " (SPACE
E) RASTERZEILEN-IRQ (SPACE) ERMOEGLICHEN <153>
325 DATA76,49,234 : JMP $EA31 " (SPACE
E) UND (SPACE) ZUR (SPACE) ALTEN (SPACE) IRQ-ROUTINE <153>
330 DATA173,18,208 : RASZ LDA $D012 " (SPACE
E) AKTUELLE (SPACE) RASTERZEILE <174>
335 DATA201,113 : CMP #$71 " (SPACE
E) WURDE (SPACE) DER (SPACE) IRQ (SPACE) BEI <190>
340 DATA176,6 : BCS $FLG1 " (SPACE
E) RASTERZEILE (SPACE) 113 (SPACE) AUSGELOEST (SPACE) ? <178>
345 DATA169,5 : LDA #$05 " (SPACE
E) FARBWERT (SPACE) FUER (SPACE) HINTERGRUND (SPACE) & (
SPACE) RAHMEN <164>
350 DATA162,113 : LDX #$71 " (SPACE
E) NAECHSTER (SPACE) IRQ (SPACE) BEI (SPACE) RASTERZEIL
E (SPACE) 113 <247>
355 DATA208,14 : BNE $F&RS " (SPACE
E) NEUE (SPACE) WERTE (SPACE) SETZEN <89>
360 DATA201,193 : FLG1 CMP #$C1 " (SPACE
E) WURDE (SPACE) DER (SPACE) IRQ (SPACE) BEI <0>
365 DATA176,6 : BCS $FLG2 " (SPACE
E) RASTERZEILE (SPACE) 193 (SPACE) AUSGELOEST (SPACE) ? <210>
370 DATA169,2 : LDA #$02 " (SPACE
E) FARBWERT (SPACE) FUER (SPACE) HINTERGRUND (SPACE) & (
SPACE) RAHMEN <129>
375 DATA162,193 : LDX #$C1 " (SPACE
E) NAECHSTER (SPACE) IRQ (SPACE) BEI (SPACE) RASTERZEIL
E (SPACE) 193 <140>
380 DATA208,4 : BNE $F&RS " (SPACE
E) NEUE (SPACE) WERTE (SPACE) SETZEN <197>
385 DATA169,6 : FLG2 LDA #$06 " (SPACE
E) FARBWERT (SPACE) FUER (SPACE) HINTERGRUND (SPACE) & (
SPACE) RAHMEN <144>
390 DATA162,0 : LDX #$00 " (SPACE
E) NAECHSTER (SPACE) IRQ (SPACE) BEI (SPACE) RASTERZEIL
E (SPACE) 0 <9>
395 DATA141,32,208 : F&RS STA $D020 " (SPACE
E) NEUEN (SPACE) FARBWERT <213>
400 DATA141,33,208 : STA $D021 " (SPACE
E) UND <216>
405 DATA142,18,208 : STX $D012 " (SPACE
E) NEUE (SPACE) RASTERZEILE (SPACE) SETZEN <143>
410 DATA76,129,234 : JMP $EA81 " (SPACE
E) ANS (SPACE) ENDE (SPACE) DER (SPACE) ALTEN (SPACE) IRQ
-ROUTINE <236>
415 : <218>
420 + 1986 BY DANIEL DURSTEWITZ +
ENDE DES LISTINGS <89>

```

Listing 3, Teil 2

zeile der IRQ ausgelöst wurde. Dabei wird sicherheitshalber immer mit dem Assemblerbefehl BCS – auf größer oder gleich geprüft, da sich der VIC mit der Ausführung unter Umständen schon eine Rasterzeile weiter befindet. Je nach dem, bei welcher Rasterzeile der IRQ ausgelöst wurde, wird eine bestimmte Rahmen- und Hintergrundfarbe gesetzt, die aber nur ab der betreffenden Rasterzeile wirksam ist. Die Rahmen- und Hintergrundfarbe des schon erstellten Bild-

schirmteiles wird davon nicht beeinflusst, da zunächst einmal das ganze Bild aufgebaut sein muß, bevor auch im oberen Teil Rahmen- und Hintergrundfarbe wieder geändert werden können. Dazu kommt es aber erst gar nicht, da mit dem letzten Ändern der Rahmen- und Hintergrundfarbe im unteren Teil des Bildschirms auch der nächste Rasterzeilen-IRQ wieder für die Zeile 0 festgelegt wird. Beim folgenden Rasterzeilen-IRQ wird also die Rahmen- und Hintergrundfarbe wieder ge-

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

```

100 REM * 16 SPRITES * <230>
105 REM <248>
110 REM DIESES PROGRAMM ERMOEGLICHT <136>
115 REM DURCH EINEN RASTERZEILEN-IRQ <173>
120 REM DIE DARSTELLUNG VON <129>
125 REM 16 SPRITES GLEICHZEITIG. <179>
130 : <188>
135 : <193>
140 FORN=832TO894: READD: Z=Z+D: POKEN,D: NEXT <178>
145 IFZ<>3601 THEN PRINT " (DOWN) FEHLER (SPACE) IN (SPA
CE) DEN (SPACE) DATA-ZEILEN (SPACE) 170-180! " : END <254>
150 FORN=2040TO2047: POKEN,13: NEXT: POKES3269,255 <49>
155 : <213>
160 REM DATEN FUER DAS SPRITE <147>
165 : <223>
170 DATA1,148,192,2,73,32,4,34,16,8,247,136,9,24 <188>
175 DATA193,224,1,136,192,2,8,32,2,8,32,2,8,32,2, <44>
180 DATA0,136,128,0,136,128,0,73,0,0,42,0,0,28,0 <174>
185 : <243>
190 : <248>
195 FORN=49152TO49326: READD: Y=Y+D: POKEN,D: NEXT <32>
200 IFY<>21063 THEN PRINT " (DOWN) FEHLER (SPACE) IN (SPA
CE) DEN (SPACE) DATA-ZEILEN (SPACE) 230-610! " : END <189>
205 PRINT " (DOWN) AKTIVIEREN (SPACE) MIT (SPACE) SYS (S
PACE) 49152. " <164>
210 PRINT "ABSCHALTEN (SPACE) MIT (SPACE) SYS (SPACE) 4
9177. " <254>
215 POKES3280,0: POKES3281,0: END <189>
220 : <22>
225 : <27>
230 DATA120 : EIN SEI " (SPACE
E) INTERRUPT (SPACE) VERHINDERN <255>
235 DATA169,127 : LDA #$7F " (SPACE
E) RASTERZEILEN-UEBERTRAG <28>
240 DATA45,17,208 : AND $D011 " (SPACE
E) LOESCHEN <138>
245 DATA141,17,208 : STA $D011 " <226>
250 DATA169,0 : LDA #$00 " (SPACE
E) ERSTER (SPACE) IRQ (SPACE) BEI <54>
255 DATA141,18,208 : STA $D012 " (SPACE
E) RASTERZEILE (SPACE) 0 <115>
260 DATA169,46 : LDA #$IRQ-LO " (SPACE
E) NEUE (SPACE) IRQ-ADRESSE (SPACE) LO-BYTE <101>
265 DATA162,192 : LDX #$IRQ-HI " (SPACE
E) NEUE (SPACE) IRQ-ADRESSE (SPACE) HI-BYTE <234>
270 DATA160,1 : LDY #$01 " (SPACE
E) FLAG (SPACE) FUER (SPACE) RASTERZEILEN-IRQ (SPACE) E
IN <51>
275 DATA208,10 : BNE $SET " (SPACE
E) NEUE (SPACE) ADRESSE (SPACE) UND (SPACE) FLAG (SPACE)
SETZEN <161>
280 DATA234 : NOP " <58>
285 DATA234 : NOP " <63>
290 DATA234 : NOP " <68>
295 DATA120 : AUS SEI " (SPACE
E) INTERRUPT (SPACE) VERHINDERN <165>
300 DATA169,49 : LDA #$31 " (SPACE
E) ALTE (SPACE) IRQ-ADRESSE (SPACE) LO-BYTE <58>
305 DATA162,234 : LDX #$EA " (SPACE

```

Listing 4, Teil 1

ändert und der nächste Rasterzeilen-IRQ für 113 Rasterzeilen später festgelegt u.s.w.. Abgeschlossen werden muß die neue IRQ-Routine mit einem Sprung zum Ende der alten (\$EA81), wo die Adresse, bei der der Ausstieg aus dem gerade laufenden Programm erfolgte, wieder vom Stack (Stapel Speicher, auf dem diese Adresse bei der Auslösung des IRQ abgelegt wurde) zurückgeholt und die Rückkehr vom IRQ mit dem Befehl RTI (Return from Interrupt) vollzogen

wird. Ein leichtes Flimmern derjenigen Rasterzeilen, die sich an der Grenze zwischen zwei verschiedenen Rahmen- und Hintergrundfarben befinden, läßt sich leider nicht vermeiden, da, wie schon gesagt, die einzelnen Rasterpunkte so schnell erstellt werden, daß es nicht möglich ist, das Farbende auf einen Rasterpunkt genau festzulegen. Dieser Wert variiert darum immer etwas. Auf gleiche Weise ist auch eine Mischung aus Text und Grafik, wie sie in Listing 4+ gezeigt wird,

```

E) ALTE (SPACE) IRQ-ADRESSE (SPACE) HI-BYTE <187>
310 DATA160,0 : LDY #$00 " (SPACE
E) FLAG (SPACE) FUER (SPACE) RASTERZEILEN-IRQ (SPACE) A
US <153>
315 DATA141,20,3 : SET STA $0314 " (SPACE
E) IRQ-ADRESSE <39>
320 DATA142,21,3 : STX $0315 " (SPACE
E) UND (SPACE) FLAG (SPACE) FUER <145>
325 DATA140,26,208 : STY $D01A " (SPACE
E) RASTERZEILEN-IRQ (SPACE) EIN/AUS (SPACE) SETZEN <79>
330 DATAB8 : CLI " (SPACE
E) INTERRUPT (SPACE) WIEDER (SPACE) ERMOEGLICHEN <65>
335 DATA96 : RTS " <182>
340 DATA234 : NOP " <118>
345 DATA234 : NOP " <123>
350 DATA234 : NOP " <128>
355 DATA173,25,208 : IRQ LDA $D019 " (SPACE
E) INTERRUPT (SPACE) REQUEST (SPACE) REGISTER (SPACE) (
IRR) <189>
360 DATA141,25,208 : STA $D019 " (SPACE
E) LOESCHEN <251>
365 DATA41,1 : AND #$01 " (SPACE
E) HANDELT (SPACE) ES (SPACE) SICH (SPACE) UM <216>
370 DATA208,7 : BNE $RASZ " (SPACE
E) EINEN (SPACE) RASTERZEILEN-INTERRUPT (SPACE) ? <236>
375 DATA173,13,220 : LDA $D00D " (SPACE
E) NEIN, (SPACE) IRQ-URSACHE (SPACE) IM (SPACE) ICR (SPA
CE) LOESCHEN <101>
380 DATAB8 : CLI " (SPACE
E) RASTERZEILEN-IRQ (SPACE) ERMOEGLICHEN <213>
385 DATA76,49,234 : JMP $EA31 " (SPACE
E) UND (SPACE) ZUR (SPACE) ALTEN (SPACE) IRQ-ROUTINE <213>
390 DATA162,16 : RASZ LDX #$10 " (SPACE
E) ZAEHLER (SPACE) FUER (SPACE) POSITIONEN-
<193>
395 DATA160,7 : LDY #$07 " (SPACE
E) UND (SPACE) FARBTABELLE (SPACE) SETZEN <186>
400 DATA173,18,208 : LDA $D012 " (SPACE
E) AKTUELLE (SPACE) RASTERZEILE, (SPACE) IRQ <215>
405 DATA201,155 : CMP #$9B " (SPACE
E) AUSGELOEST (SPACE) BEI (SPACE) RASTERZEILE (SPACE) 1
55 (SPACE) ? <66>
410 DATA176,22 : BCS $FLG1 " (SPACE
E) JA, (SPACE) DANN (SPACE) ZU (SPACE) FLG1 <189>
415 DATA189,125,192: FLG2 LDA $POS1,X " (SPACE
E) DIE (SPACE) ERSTEN (SPACE) ACHT <99>
420 DATA157,0,208 : STA $D000,X " (SPACE
E) SPRITES (SPACE) AUF (SPACE) DEN <225>
425 DATA202 : DEX " (SPACE
E) BILDSCHIRM (SPACE) SETZEN <122>
430 DATA16,247 : BPL $FLG2 " <16>
435 DATA185,142,192: FLG3 LDA $FAR1,Y " (SPACE
E) DIE (SPACE) FARBEN (SPACE) FUER <27>
440 DATA153,39,208 : STA $D027,Y " (SPACE
E) DIE (SPACE) ERSTEN (SPACE) ACHT (SPACE) SPRITES <191>
445 DATA136 : DEY " (SPACE
E) FESTSETZEN <68>
450 DATA16,247 : BPL $FLG3 " <52>
455 DATA169,155 : LDA #$9B " (SPACE
E) NAECHSTER (SPACE) IRQ (SPACE) BEI (SPACE) RASTERZEIL
E (SPACE) 155 <240>
460 DATA208,20 : BNE $RSET " (SPACE
E) SETZEN <130>
465 DATA189,150,192: FLG1 LDA $POS2,X " (SPACE
E) DIE (SPACE) ZWEITEN (SPACE) ACHT <19>
470 DATA157,0,208 : STA $D000,X " (SPACE
E) SPRITES (SPACE) AUF (SPACE) DEN <19>
475 DATA202 : DEX " (SPACE
E) BILDSCHIRM (SPACE) SETZEN <172>
480 DATA16,247 : BPL $FLG1 " <50>
485 DATA185,167,192: FLG4 LDA $FAR2,Y " (SPACE
E) DIE (SPACE) FARBEN (SPACE) FUER <173>
490 DATA153,39,208 : STA $D027,Y " (SPACE
E) DIE (SPACE) ZWEITEN (SPACE) ACHT (SPACE) SPRITES <199>
495 DATA136 : DEY " (SPACE
E) FESTSETZEN <118>
500 DATA16,247 : BPL $FLG4 " <118>
505 DATA169,0 : LDA #$00 " (SPACE
E) NAECHSTER (SPACE) IRQ (SPACE) BEI (SPACE) RASTERZEIL
E (SPACE) 0 <178>
510 DATA141,18,208 : RSET STA $D012 " (SPACE
E) RASTERZEILE (SPACE) FESTSETZEN <171>
515 DATA76,129,234 : JMP $EA81 " (SPACE
E) ANS (SPACE) ENDE (SPACE) DER (SPACE) ALTEN (SPACE) IRQ
-ROUTINE <86>
520 DATA234 : NOP " <43>
525 DATA234 : NOP " <48>
530 DATA234 : NOP " <53>
535 : <83>
540 TABELLE POSITIONEN 1 (POS1) <111>
545 : <93>
550 DATA32,80,64,128,112,64,144,96,192,112,224,1 <220>
555 : <103>
560 TABELLE FARBEN 1 (FAR1) <16>
565 : <113>
570 DATA1,2,3,4,5,6,7,8 <171>
575 : <123>
580 TABELLE POSITIONEN 2 (POS2) <184>
585 : <133>
590 DATA40,184,88,224,120,192,168,176,200,169,23 <169>
595 : <143>
600 TABELLE FARBEN 2 (FAR2) <89>
605 : <153>
610 DATA9,10,11,12,13,14,15,10 <132>
615 : <163>
620 + 1986 BY DANIEL DURSTEWITZ +
ENDE DES LISTINGS <34>

```

Listing 4, Teil 2

möglich. Hier wird nicht wie in Listing 3 zwischen mehreren Farben hin- und hergeschaltet, sondern nach genau demselben Prinzip ab einer bestimmten Rasterzeile vom Grafik- zum Textbildschirm umgeschaltet. Diese Fähigkeit wird vor allem bei Grafik-Adventuren und Funktionszeichnern, aber auch bei vielen anderen Programmen, wo Grafikbildschirme mit Text unterlegt werden bzw. unter einer Grafik Eingaben und Ausgaben im Textmodus möglich sein müssen, genutzt. Als nächstes möchten wir Ihnen zeigen, wie Sie mehr als acht Sprites auf dem Bildschirm darstellen können.

Im nächsten Heft erscheint der letzte Teil dieses Kurses!

Der nächste Kurs kommt bestimmt!

Ab Heft 6/87

erläutern wir anhand der WAIT-Routine
alles Wichtige über:

den Timer und über maskiertes Poken

Ohne viele Worte:

C=64

Immigration - ein schnelles Space-Invaders mit wenig Eintipparbeit!

Hierbei handelt es sich um eine superkurze und superschnelle Version von Space-Invaders für den C 64. Also für alle, die sich nicht die Finger wundtippen wollen: ran an den Speck, mal eben eingetippt und schon kann man spielen.

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

```

999 PRINTCHR*(8)          <54>
1000 REM IMMIGRATION C<64  <120>
1010 REM 22.12.1986 BY CHRISTI
AN PROBST
1015 REM 7813 STAUFEN - GRUNER
N (BRD)
1020 :
1030 S=4*4096+7*256:PRINT"(CLE
AR)"
1035 PRINT"(RVSON BLACK DOWN R
IGHT4)IMMIGRATION(SPACE)C<64"
1036 PRINT"(RVSON WHITE DOWN6
RIGHT4)BITTE(SPACE)CA.(SPACE2)
1,5(SPACE)MIN.(SPACE2)WARTEN(S
PACE)!"
1037 PRINT"(RVSON CYAN DOWN4 R
IGHT4 SPACE)WRITTEN(SPACE)BY(S
PACE)CHRISTIAN(SPACE)PROBST(C)
-SOFT"
1038 PRINT"(RVSON YELLOW RIGHT
4 SPACE)7813(SPACE)STAUFEN-GRU
NERN(SPACE)(BRD)(SPACE)1986(S
PACE)"
1040 READX:IF X=-1THEN1060
1050 A=A+X:POKE8+B,X:B=B+1:GOT
O1040
1060 IF B>2030 THENPRINT"***ZUV
IELE(SPACE)DATEN**":END
1070 IF B<2030 THENPRINT"***ZUM
ENIG(SPACE)DATEN**":END
1080 IFA<>210762THENPRINT"***DA
TEN(SPACE)ANZAHL(SPACE)RICHTIG
**":GOTO1130
1090 PRINT"***ALLES(SPACE)RICHT
IG**"
1100 PRINT"***ZUM(SPACE)START(S
PACE)SPACE**"
1110 GET A$:IFA$<>"(SPACE)"THE
N1110
1120 BYB 8
1130 PRINT"***JEDDOCH(SPACE)EINI
GE(SPACE)FALSCH**":END
1140 :
1150 DATA 169,0,133,254,169
1160 DATA 56,133,255,169,0
1170 DATA 162,13,160,0,145
1180 DATA 254,136,208,251,230
1190 DATA 253,202,208,244,169
1200 DATA 0,133,254,169,216
1210 DATA 133,255,169,6,162
1220 DATA 4,160,0,145,254
1230 DATA 136,208,251,230,255

```

```

1240 DATA 202,208,244,160,63
1250 DATA 185,73,71,153,0
1260 DATA 56,136,16,247,173
1270 DATA 24,208,41,240,24
1280 DATA 105,14,141,24,208
1290 DATA 76,137,71,126,153
1300 DATA 153,255,255,153,153
1310 DATA 129,102,165,36,126
1320 DATA 153,153,255,126,204
1330 DATA 51,204,51,204,51
1340 DATA 204,51,0,0,130
1350 DATA 42,204,51,204,51
1360 DATA 0,0,0,0,0
1370 DATA 145,205,51,0,24
1380 DATA 24,24,24,24,24
1390 DATA 24,16,8,16,8
1400 DATA 16,8,16,8,24
1410 DATA 60,126,24,60,126
1420 DATA 255,102,162,7,189
1430 DATA 151,71,157,184,56
1440 DATA 202,16,247,76,159
1450 DATA 71,66,66,66,255
1460 DATA 66,66,66,255,162
1470 DATA 161,189,71,78,157
1480 DATA 127,57,202,208,247
1490 DATA 76,221,71,141,24
1500 DATA 212,162,5,138,24
1510 DATA 105,58,157,0,4
1520 DATA 157,27,4,202,14
1530 DATA 243,162,3,138,24
1540 DATA 105,64,157,23,4
1550 DATA 202,16,246,96,255
1560 DATA 255,255,255,255,255
1570 DATA 255,255,255,255,255
1580 DATA 2,255,255,255,255
1590 DATA 255,234,32,43,76
1600 DATA 169,0,141,12,64
1610 DATA 141,13,64,141,14
1620 DATA 64,32,222,76,32
1630 DATA 2,76,32,47,72
1640 DATA 32,73,72,32,150
1650 DATA 72,32,240,77,32
1660 DATA 221,72,32,97,76
1670 DATA 169,0,32,13,76
1680 DATA 32,62,73,32,174
1690 DATA 74,32,105,74,32
1700 DATA 5,75,32,34,75
1710 DATA 32,105,74,32,88
1720 DATA 75,32,174,74,32
1730 DATA 105,74,32,179,72
1740 DATA 173,9,64,240,221
1750 DATA 76,76,75,169,32
1760 DATA 162,0,157,0,4
1770 DATA 157,0,5,157,0
1780 DATA 6,157,231,6,202

```

```

1790 DATA 208,241,142,33,208
1800 DATA 142,32,208,96,169
1810 DATA 1,162,39,157,0
1820 DATA 216,202,16,250,169
1830 DATA 7,162,0,157,24
1840 DATA 217,157,0,218,202
1850 DATA 208,247,169,10,162
1860 DATA 255,157,247,218,202
1870 DATA 208,250,169,0,162
1880 DATA 40,157,192,219,202
1890 DATA 208,250,32,24,77
1900 DATA 96,32,32,32,2
1910 DATA 2,2,32,32,32
1920 DATA 2,2,2,32,32
1930 DATA 32,2,2,32
1940 DATA 32,32,32,32,48
1950 DATA 48,48,48,48,48
1960 DATA 32,162,20,189,120
1970 DATA 72,157,32,7,157
1980 DATA 52,7,157,72,7
1990 DATA 157,92,7,157,112
2000 DATA 7,157,132,7,202
2010 DATA 208,232,32,188,72
2020 DATA 169,9,141,152,7
2030 DATA 32,61,78,96,169
2040 DATA 255,133,86,169,3
2050 DATA 133,87,160,1,162
2060 DATA 22,24,165,86,105
2070 DATA 40,133,86,165,87
2080 DATA 105,0,133,87,169
2090 DATA 23,145,86,202,208
2100 DATA 236,96,173,10,64
2110 DATA 24,106,24,106,170
2120 DATA 169,160,133,86,169
2130 DATA 4,133,87,138,240
2140 DATA 17,24,165,86,105
2150 DATA 40,133,86,165,87
2160 DATA 105,0,133,87,202
2170 DATA 76,237,72,160,24
2180 DATA 169,0,145,86,24
2190 DATA 165,86,105,40,133
2200 DATA 88,165,87,105,0
2210 DATA 133,89,169,1,145
2220 DATA 88,24,165,88,105
2230 DATA 40,133,88,165,89
2240 DATA 105,0,133,89,169
2250 DATA 0,145,88,24,165
2260 DATA 88,105,40,133,88
2270 DATA 165,89,105,0,133
2280 DATA 89,169,1,145,88
2290 DATA 136,136,208,197,169
2300 DATA 88,133,86,169,4
2310 DATA 133,87,160,1,169
2320 DATA 63,141,191,7,24
2330 DATA 165,86,105,1,133
2340 DATA 86,165,87,105,0
2350 DATA 133,87,177,86,240
2360 DATA 7,201,63,208,235
2370 DATA 76,168,73,32,135
2380 DATA 75,160,1,169,32
2390 DATA 145,86,160,0,177
2400 DATA 86,201,5,208,11
2410 DATA 234,234,32,209,77
2420 DATA 32,113,77,76,77
2430 DATA 73,201,23,208,11
2440 DATA 169,1,160,41,145
2450 DATA 86,160,1,76,97
2460 DATA 77,169,0,145,86
2470 DATA 160,1,76,77,73
2480 DATA 169,32,145,86,160
2490 DATA 41,169,1,145,86
2500 DATA 160,1,76,77,73
2510 DATA 169,191,133,86,169
2520 DATA 7,133,87,160,0
2530 DATA 169,63,141,40,4
2540 DATA 56,165,86,233,1
2550 DATA 133,86,165,87,233
2560 DATA 0,133,87,160,0
2570 DATA 177,86,201,1,240
2580 DATA 9,201,6,240,87
2590 DATA 201,63,208,227,96
2600 DATA 32,135,75,169,32
2610 DATA 145,86,160,1,177
2620 DATA 86,201,5,208,25
2630 DATA 169,32,145,86,160
2640 DATA 0,32,209,77,32

```

```

<254>
<83>
<159>
<125>
<169>
<87>
<117>
<97>
<181>
<126>
<136>
<172>
<13>
<145>
<131>
<32>
<248>
<243>
<176>
<170>
<207>
<62>
<181>
<163>
<147>
<97>
<204>
<53>
<216>
<177>
<178>
<69>
<113>
<64>
<118>
<146>
<40>
<57>
<107>
<109>
<200>
<47>
<240>
<13>
<174>
<107>
<221>
<32>
<150>
<28>
<23>
<72>
<180>
<203>
<182>
<46>
<68>
<106>
<252>
<125>
<169>
<86>
<232>
<243>
<93>
<178>
<199>
<174>
<110>
<214>
<204>
<228>
<49>
<63>
<149>
<2>
<76>
<144>
<218>
<54>
<126>
<103>
<213>
<5>
<11>

```

```

2650 DATA 161,75,76,183,73
2660 DATA 200,177,86,136,201
2670 DATA 9,240,95,208,15
2680 DATA 201,23,208,242,169
2690 DATA 0,160,39,145,86
2700 DATA 160,0,76,183,73
2710 DATA 169,1,145,86,160
2720 DATA 0,76,183,73,169
2730 DATA 32,145,86,160,40
2740 DATA 169,0,145,86,160
2750 DATA 0,234,234,234,76
2760 DATA 183,73,169,32,145
2770 DATA 86,160,40,177,86
2780 DATA 201,2,208,9,169
2790 DATA 3,145,86,160,0
2800 DATA 76,183,73,201,3
2810 DATA 208,22,169,4,145
2820 DATA 86,160,0,76,183
2830 DATA 73,41,254,240,6
2840 DATA 169,32,145,86,160
2850 DATA 0,76,183,73,201
2860 DATA 7,208,6,169,1
2870 DATA 141,9,64,96,201
2880 DATA 32,240,2,208,227
2890 DATA 76,34,76,160,0
2900 DATA 177,90,201,5,240
2910 DATA 3,76,125,77,76
2920 DATA 136,77,169,32,145
2930 DATA 90,56,165,90,233
2940 DATA 40,133,90,165,91
2950 DATA 233,0,133,91,177
2960 DATA 90,201,0,240,16
2970 DATA 201,1,240,12,201
2980 DATA 8,240,18,201,32
2990 DATA 240,15,76,164,74
3000 DATA 234,32,208,97,32
3010 DATA 161,75,169,32,145
3020 DATA 90,96,169,5,145
3030 DATA 90,96,172,1,220
3040 DATA 152,41,28,201,20
3050 DATA 208,1,96,152,41
3060 DATA 4,240,121,152,41
3070 DATA 8,240,122,152,41
3080 DATA 16,240,123,96,173
3090 DATA 8,64,201,1,208
3100 DATA 1,96,206,8,64
3110 DATA 76,34,75,173,0
3120 DATA 64,201,58,208,1
3130 DATA 96,238,8,64,76
3140 DATA 34,75,173,6,64
3150 DATA 208,1,96,24,169
3160 DATA 152,109,8,64,133
3170 DATA 90,169,7,105,0
3180 DATA 133,91,169,0,141
3190 DATA 6,64,169,5,160
3200 DATA 0,145,90,96,141
3210 DATA 0,64,142,1,64
3220 DATA 140,2,64,162,112
3230 DATA 234,234,32,10,70
3240 DATA 202,208,250,173,0
3250 DATA 64,172,2,64,174
3260 DATA 1,64,96,174,8
3270 DATA 64,169,7,157,152
3280 DATA 7,232,169,32,157
3290 DATA 152,7,202,202,157
3300 DATA 152,7,232,96,32
3310 DATA 201,74,76,190,74
3320 DATA 32,215,74,76,195
3330 DATA 74,76,144,77,234
3340 DATA 234,234,234,234,96
3350 DATA 32,49,78,32,172
3360 DATA 77,32,188,76,76
3370 DATA 236,71,169,42,133
3380 DATA 86,169,4,133,87
3390 DATA 160,0,140,7,64
3400 DATA 24,165,86,105,1
3410 DATA 133,86,165,87,105
3420 DATA 0,133,87,177,86
3430 DATA 201,1,240,14,201
3440 DATA 0,240,10,201,63
3450 DATA 208,229,238,10,64
3460 DATA 76,63,76,96,173
3470 DATA 18,208,201,250,176
3480 DATA 1,96,160,40,177
3490 DATA 86,201,32,240,3
3500 DATA 160,0,96,169,6

```

```

<236>
<132>
<67>
<252>
<182>
<134>
<200>
<2>
<137>
<224>
<10>
<147>
<101>
<1>
<64>
<82>
<41>
<236>
<154>
<214>
<185>
<246>
<251>
<49>
<164>
<235>
<157>
<77>
<156>
<184>
<201>
<182>
<171>
<230>
<26>
<231>
<142>
<239>
<44>
<186>
<141>
<116>
<144>
<213>
<101>
<21>
<35>
<163>
<123>
<39>
<135>
<112>
<182>
<59>
<49>
<94>
<19>
<47>
<160>
<177>
<149>
<227>
<39>
<48>
<47>
<184>
<95>
<175>
<120>
<180>
<20>
<113>
<193>
<112>
<170>
<210>
<67>
<234>
<147>
<153>
<21>
<236>
<200>
<134>
<200>
<176>

```

```

3510 DATA 145,86,160,0,96
3520 DATA 248,24,173,3,64
3530 DATA 101,251,141,3,64
3540 DATA 173,4,64,101,252
3550 DATA 141,4,64,173,5
3560 DATA 64,105,0,141,5
3570 DATA 64,216,173,3,64
3580 DATA 74,74,74,74,24
3590 DATA 105,48,141,10,4
3600 DATA 173,3,64,41,15
3610 DATA 105,48,141,11,4
3620 DATA 173,4,64,74,74
3630 DATA 74,74,24,105,48
3640 DATA 141,8,4,173,4
3650 DATA 64,41,15,105,48
3660 DATA 141,9,4,173,5
3670 DATA 64,74,74,74,74
3680 DATA 24,105,48,141,6
3690 DATA 4,173,5,64,41
3700 DATA 15,105,48,141,7
3710 DATA 4,96,169,0,141
3720 DATA 3,64,141,4,64
3730 DATA 141,5,64,141,9
3740 DATA 64,141,6,64,141
3750 DATA 7,64,141,33,208
3760 DATA 141,32,208,169,20
3770 DATA 141,8,64,96,169
3780 DATA 6,145,86,160,0
3790 DATA 76,183,73,162,11
3800 DATA 169,0,157,0,64
3810 DATA 202,208,250,169,5
3820 DATA 133,251,169,112,141
3830 DATA 15,75,96,24,248
3840 DATA 165,251,105,5,133
3850 DATA 251,165,252,105,0
3860 DATA 133,252,216,238,10
3870 DATA 64,238,10,64,238
3880 DATA 10,64,238,10,64
3890 DATA 32,170,76,234,76
3900 DATA 242,71,162,39,169
3910 DATA 8,157,40,4,202
3920 DATA 208,250,96,234,234
3930 DATA 234,234,234,160,16
3940 DATA 162,7,142,11,64
3950 DATA 173,17,208,41,248
3960 DATA 13,11,64,141,17
3970 DATA 208,32,19,78,202
3980 DATA 16,236,162,0,142
3990 DATA 11,64,173,17,208
4000 DATA 41,248,13,11,64
4010 DATA 141,17,208,32,19
4020 DATA 70,232,224,8,208
4030 DATA 234,136,208,207,169
4040 DATA 27,141,17,208,96
4050 DATA 162,11,206,15,75
4060 DATA 202,208,250,238,10
4070 DATA 64,162,40,169,32
4080 DATA 157,192,7,202,208
4090 DATA 250,206,139,75,206
4100 DATA 139,75,206,139,75
4110 DATA 234,234,160,0,162
4120 DATA 0,142,33,208,140
4130 DATA 32,208,140,15,212
4140 DATA 202,208,244,136,208
4150 DATA 239,96,169,250,141
4160 DATA 139,75,173,5,64
4170 DATA 205,14,64,48,20
4180 DATA 208,22,173,4,64
4190 DATA 205,13,64,48,10
4200 DATA 208,12,173,3,64
4210 DATA 205,12,64,16,4
4220 DATA 32,43,76,96,173
4230 DATA 3,64,141,12,64
4240 DATA 173,4,64,141,13
4250 DATA 64,173,5,64,141
4260 DATA 14,64,76,255,76
4270 DATA 173,12,64,74,74
4280 DATA 74,74,24,105,48
4290 DATA 141,37,4,173,12
4300 DATA 64,41,15,105,48
4310 DATA 141,38,4,173,13
4320 DATA 64,74,74,74,74
4330 DATA 24,105,48,141,35
4340 DATA 4,173,13,64,41
4350 DATA 15,105,48,141,36
4360 DATA 4,173,14,64,74

```

```

<162>
<141>
<127>
<139>
<125>
<11>
<189>
<218>
<54>
<127>
<88>
<35>
<20>
<214>
<6>
<254>
<121>
<3>
<18>
<40>
<9>
<49>
<54>
<47>
<82>
<35>
<72>
<40>
<165>
<108>
<152>
<178>
<61>
<46>
<252>
<254>
<211>
<175>
<111>
<131>
<189>
<91>
<235>
<58>
<255>
<159>
<44>
<197>
<45>
<187>
<222>
<112>
<76>
<217>
<172>
<253>
<110>
<119>
<50>
<29>
<199>
<32>
<102>
<154>
<78>
<12>
<23>
<225>
<24>
<221>
<201>
<254>
<249>
<70>
<9>
<191>
<160>
<63>
<146>
<106>
<5>
<228>
<112>
<10>
<229>

```

ENDE DES LISTINGS

Fortsetzung von Seite 38.

```

20530 SYS 2327,0,250,1,2,1,2,1,2,1,2,1,2,1,2
20540 GL=1000:RETURN
21000 REM LEVEL III
21010 SYS 2178,136,137,145,146,147,141,142,143
21015 SYS 2227,7,13,13,13,3,3,5,14
21016 SYS 2276,5,3
21020 SYS 2069,150,185,150,150,50,0,80,30,110,60
,140,90,170,120,200,200,85
21030 SYS 2327,0,250,1,255,1,255,1,255,1,255,1,2
55,1,255
21040 GL=750:RETURN
21500 REM LEVEL IV
21510 SYS 2178,136,137,145,146,147,141,142,143
21515 SYS 2227,14,5,3,3,4,4,12,15
21516 SYS 2276,2,6
21520 SYS 2069,150,185,150,150,50,0,80,30,110,60
,140,90,170,120,200,200,207
21530 SYS 2327,0,250,1,2,1,2,1,2,1,2,1,2,1,2
21540 GL=1000:RETURN
22000 REM LEVEL V
22010 SYS 2178,136,137,145,146,147,141,142,143
22015 SYS 2227,9,14,12,12,12,12,1,1
22016 SYS 2276,5,3
22020 SYS 2069,150,185,150,150,50,0,80,30,110,60
,140,90,170,120,200,200,85
22030 SYS 2327,0,250,1,3,1,3,1,3,1,3,1,3,1,3
22040 GL=1000:RETURN
22500 REM LEVEL VI
22510 SYS 2178,136,137,145,146,147,141,142,143
22515 SYS 2227,15,5,11,11,4,4,12,15

```

<186>	22520 SYS 2069,150,185,150,150,50,0,80,30,110,60	<37>
<115>	,140,90,170,120,200,200,13	<34>
<161>	22530 SYS 2327,0,250,1,252,1,2,1,2,1,2,1,2	<102>
<162>	22540 GL=750:RETURN	<108>
<70>	23000 REM LEVEL VII	<211>
<146>	23010 SYS 2178,136,137,145,146,147,141,142,143	<121>
	23015 SYS 2227,9,14,12,12,12,12,1,1	<51>
<131>	23016 SYS 2276,5,3	<105>
	23020 SYS 2069,150,185,150,150,50,0,80,30,110,60	
<220>	,140,90,170,120,200,200,255	<132>
<130>	23030 SYS 2327,0,249,1,3,1,3,1,3,1,3,1,3	<118>
<109>	23040 GL=1000:RETURN	<65>
<152>	23500 REM LEVEL VIII	<163>
<126>	23510 SYS 2178,136,137,145,146,147,141,142,143	<111>
<142>	23515 SYS 2227,1,1,3,3,4,4,2,15	<42>
	23516 SYS 2276,4,2	<79>
<226>	23520 SYS 2069,150,185,150,150,50,0,80,30,110,60	
<166>	,140,90,170,120,200,200,13	<14>
<95>	23530 SYS 2327,0,249,1,254,1,254,1,254,1,254,1,2	
<14>	54,1,254	<66>
<141>	23540 GL=1000:RETURN	<54>
<71>	24000 REM LEVEL IX UND FOLGENDE	<65>
<126>	24010 SYS 2178,136,137,145,146,147,141,142,143	<101>
	24015 SYS 2227,9,14,12,12,12,12,1,1	<31>
<111>	24016 SYS 2276,5,3	<85>
<50>	24020 SYS 2069,150,185,150,150,50,0,80,30,110,60	
<85>	,140,90,170,120,200,200,255	<112>
<76>	24030 SYS 2327,0,249,1,4,1,4,1,4,1,4,1,4	<248>
<131>	24040 GL=1250:RETURN	<84>
<233>	ENDE DES LISTINGS	

Ein nützliches Utensil!

Für den C 64

Wer kennt das nicht: da hat man ein sehr nützliches Maschinen-spracheprogramm, kann es aber nicht benutzen, weil man die Startadresse nicht mehr auf-treiben kann. Dieses Programm nun schafft hier Abhilfe. Denn es er-zeugt aus jedem beliebigen Ma-schinspracheprogramm eine Version, die man ganz einfach wie ein Basicprogramm einlädt und mit RUN startet. Der hier abgedruckte Lader bietet sowohl die Möglichkeit, das Maschinen-programm selbst über einen Mo-nitor einzugeben, oder das Pro-gramm aus den Dateilen auszu-lesen. Normalerweise schreibt der Basiclader den erzeugten Code auf Diskette. Arbeitet man mit einer Datensette, muß die Geräte-adresse

```

1000 REM*****
110 REM*
120 REM*      RUN VERSION CREATOR
130 REM*
140 REM*      VON THOMAS GUENTHER
150 REM*      GERANIENSTR.7
160 REM*      4460 NORDHORN
170 REM*****
180 :
190 :
200 :

1000 REM***** DATEN FUER BASICZEILE. (8000-8020)
1010 DATA 20,08,0A,00,9E,20,32,30 :
1020 DATA 30,33,20,3A,20,8F,20,2D :
1030 DATA 20,52,55,4E,20,56,45,52 :
1040 DATA 53,49,4F,4E,20,2D,00,00,00 :
1055 REM***** KOPFPROGRAMM FUER RUN VERSION.
1060 DATA EA :REM 8021 NOP ; BEGINN DES KOPFPROGRAMMES.
1070 DATA A9,42 :REM 8022 LDA #$42 ; ANFANGSADRESSE DES ZU KOPIERENDEN
1080 DATA 85,5F :REM 8024 STA $5F ; BLOCKES (PRG) IN DIE ZEROFAH -
1090 DATA A9,08 :REM 8026 LDA #$08 ; ADRESSEN $5F UND $60.
1100 DATA 85,60 :REM 8028 STA $60 ;
1110 DATA A9,FF :REM 802A LDA $FFF. ; ENDADRESSE NACH $5A UND $5B.
1120 DATA 85,5A :REM 802C STA $5A ; DIE KORREKTEN WERTE WERDEN IN DAS
1130 DATA A9,FF :REM 802E LDA $FFF. ; PROGRAMM ERST SPAETER GESCHRIEBEN.

```

RUN VERSION GENERATOR

```

1140 DATA 85,5B :REM 8030 STA $5B ; $FF - BYTES SIND NUR PLATZHALTER.
1150 DATA A9,FF :REM 8032 LDA $FFF ; ENDADRESSE DES ZIELBLOCKES NACH
1160 DATA 85,5B :REM 8034 STA $5B ; $5B UND $59.
1170 DATA A9,FF :REM 8036 LDA $FFF ;
1180 DATA 85,5F :REM 8038 STA $59 ;
1190 DATA 20,BF,A3 :REM 803A JSR $A3BF ; BLOCKVERSCHIEBEROUTINE.
1200 DATA 4C,00,00 :REM 803D JMP $0000 ; IN KOPIERTES PROGRAMM VERZWEIGEN.
1210 DATA EA :REM 8040 NOP ;
1215 REM***** ADRESSE NACH A/Y HOLEN.
1220 DATA 20,FD,AE :REM 8041 JSR $AEFD ; AUF KOMMA TESTEN.
1230 DATA 20,8A,AD :REM 8044 JSR $AD8A ; ADRESSE IN FAC HOLEN UND
1240 DATA 20,F7,B7 :REM 8047 JSR $B7F7 ; IN INTERFORM NACH $14 UND $15.
1250 DATA A5,14 :REM 804A LDA $14 ; LOWBYTE IN AKKU UND
1260 DATA A4,15 :REM 804C LDY $15 ; HIGHBYTE IN Y-REGISTER HOLEN.
1270 DATA 60 :REM 804E RTS ; ZURUECK INS HAUPTPROGRAMM.
1275 REM***** (PROGRAMMEINSPRUNG) PRG. KOPIEREN.
1280 DATA 20,41,80 :REM 804F JSR $8041 ; ADRESSE (PRG-ANFANG) HOLEN
1290 DATA 48 :REM 8052 PHA ; UND AUF STAPEL LEGEN.
1300 DATA 98 :REM 8053 TYA ;
1310 DATA 48 :REM 8054 PHA ;
1320 DATA 20,41,80 :REM 8055 JSR $8041 ; WEITERE ADRESSE (PRG-ENDE) HOLEN
1330 DATA 85,5A :REM 8058 STA $5A ; UND NACH $5A UND $5B.
1340 DATA 84,5B :REM 805A STY $5B ;
1350 DATA 68 :REM 805C PLA ; ADRESSE VOM STAPEL HOLEN
1360 DATA 85,60 :REM 805D STA $60 ; UND NACH $5F UND $60.
1370 DATA 68 :REM 805F PLA ;
1380 DATA 85,5F :REM 8060 STA $5F ;
1390 DATA A5,5A :REM 8062 LDA $5A ;
1400 DATA 80,33,80 :REM 8064 STA $8033 ; ENDADRESSE AUCH IM KOPFPROGRAMM
1410 DATA 8C,37,80 :REM 8067 STY $8037 ; ABSPEICHERN.
1420 DATA 38 :REM 806A SEC ; DIFFERENZ BEIDER ADRESSEN BERECHNEN
1430 DATA E5,5F :REM 806B SBC $5F ; (PRG-LAENGE) UND NACH $FB UND $FC.
1440 DATA 85,6B :REM 806D STA $FB ;
1450 DATA 98 :REM 806F TYA ;
1460 DATA E5,60 :REM 8070 SBC $60 ;
1470 DATA 85,FC :REM 8072 STA $FC ;
1480 DATA A5,FB :REM 8074 LDA $FB ;
1490 DATA 18 :REM 8076 CLC ; NUN DIE STARTADRESSE IM BASICOFF
1500 DATA 69,42 :REM 8077 ADC $42 ; HINZUADIEREN, UM 80 DIE ENDADRESSE
1510 DATA 8D,2B,80 :REM 8079 STA $802B ; IN DER RUN VERSION ZU ERHALTEN.
1520 DATA 85,58 :REM 807C STA $58 ; DIESE NACH $5B UND $59 UND IN KOPF-
1530 DATA A5,FC :REM 807E LDA $FC ; PROGRAMM.
1540 DATA 69,08 :REM 8080 ADC $08 ;
1550 DATA 8D,2F,80 :REM 8082 STA $802F ;
1560 DATA 85,59 :REM 8085 STA $59 ;
1570 DATA 20,BF,A3 :REM 8087 JSR $A3BF ; PROGRAMM KOPIEREN.
1575 REM***** PROGRAMMKOPF KOPIEREN.
1580 DATA A9,00 :REM 808A LDA $00 ; STARTADRESSE (8000) DES PROGRAMM-
1590 DATA 85,5F :REM 808C STA $5F ; KOPFES NACH $5F UND $60.
1600 DATA A9,80 :REM 808E LDA $80 ;
1610 DATA 85,60 :REM 8090 STA $60 ;
1620 DATA 85,5B :REM 8092 STA $5B ; ENDADRESSE (8040) NACH $5A UND $5B.
1630 DATA A9,40 :REM 8094 LDA $40 ;
1640 DATA 85,5A :REM 8096 STA $5A ;
1650 DATA A9,41 :REM 8098 LDA $41 ;
1660 DATA 85,58 :REM 809A STA $58 ; SCHLIESSLICH NEUE ENDADRESSE
1670 DATA A9,08 :REM 809C LDA $08 ; ($841) NACH $5B UND $59.
1680 DATA 85,59 :REM 809E STA $59 ;
1690 DATA 20,BF,A3 :REM 80A0 JSR $A3BF ; PROGRAMMKOPF KOPIEREN.
1700 DATA 20,41,80 :REM 80A3 JSR $8041 ; ADRESSE (AUFRUFADRESSE) HOLEN
1710 DATA 8D,3F,08 :REM 80A6 STA $803F ; UND IM KOPFPROGRAMM ABLEGEN.
1720 DATA 8C,40,08 :REM 80A9 STY $8040 ;
1730 DATA EA,EA,EA :REM 80AC NOP ; (BEFEHL GESTRICHEN).
1740 DATA F0,03 :REM 80AF BEQ $80B4 ; WENN JA VERZWEIGEN.
1750 DATA A9,4C :REM 80B1 LDA $4C ; CODE FUER JMP.
1760 DATA 0C :REM 80B3 .BYTE 0C ; NAECHSTEN BEFEHL UEBERSPRINGEN.
1770 DATA A9,60 :REM 80B4 LDA $60 ; CODE FUER RTS. (WENN ADRESSE NULL)
1780 DATA 8D,3E,08 :REM 80B6 STA $803E ; CODE IN KOPFPROGRAMM SCHREIBEN.
1785 REM***** SAVEROUTINE.
1790 DATA A9,01 :REM 80B9 LDA $01 ; PROGRAMMSTARTADRESSE NACH $FB UND
1800 DATA 85,FB :REM 80BB STA $FB ; $FC. ($881, DA BASIC.)
1810 DATA A9,08 :REM 80BD LDA $08 ;
1820 DATA 85,FC :REM 80BF STA $FC ;
1830 DATA 20,FD,AE :REM 80C1 JSR $AEFD ; AUF KOMMA TESTEN.
1840 DATA 20,D4,E1 :REM 80C4 JSR $E1D4 ; FILENAMEN HOLEN / PARAMETER SETZEN.
1850 DATA AE,2F,80 :REM 80C7 LDX $802B ; ENDADRESSE DES PRG. IN X/Y.
1860 DATA AC,2F,80 :REM 80CA LDY $802F ;
1870 DATA A9,01 :REM 80CD LDA $01 ; SEKUNDAERADRESSE IST 1.
1880 DATA 85,B9 :REM 80CF STA $B9 ; NACH $B9.
1890 DATA A9,FB :REM 80D1 LDA $FB ; OFFSET FUER ANFANGSADRESSE IN AKKU.
1900 DATA 4C,D8,FF :REM 80D3 JMP $FFD8 ; SAVEROUTINE.
2000 REM*****
2010 :
2020 : POKE53280,0:POKE53281,0:PRINT"UN EINEN AUGENBLICK BITTE, DIE DATEN" :
2030 : S=0:PRINT"5":PRINT" WERDEN ZUNAECHST EINGELESEN !" :FOR N=0 TO 213:
2040 : READ A$: B$= LEFT$(A$,1) : C$= MID$(A$,2,1) : F$=B$: :80B0B 3000 :B=F

```

Kleinanzeigen-
coupon
finden Sie
in der Mitte
dieses Heftes!

FIND

für



SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```
100 REM *****
+++++
110 REM ***** FI
ND *****
120 REM ***** by Uwe B
chaefer *****
130 REM *****
+++++
140 DATA DD,21,04,AC,3E,C3,21,C
3,A4,DD,77,00,DD,75,01,DD,-1779
150 DATA 74,02,C9,DD,21,6A,AS,E
5,2B,7E,FE,21,38,FA,EB,E1,-2295
160 DATA 06,04,1A,E6,7F,CD,8A,F
F,DD,8E,00,1B,DD,23,C2,64,-1779
```

```
2050 : F=C* : GOSUB 3000 : C=F : A=(B*16)+C : S=S+A : POKE N+32768,A : NEXT N:
2060 : IF S<24178 THEN PRINT "MASCHINENCODE ABSPEICHERN? " : GET A$ : IF A$="" THEN 2070:
2070 : PRINT "J" THEN OPEN1,0,1,"R-VERSION " : PRINT#1,CHR$(0);CHR$(128);:GOTO2110:
2080 : IF A$="J" THEN OPEN1,0,1,"R-VERSION " : PRINT#1,CHR$(0);CHR$(128);:GOTO2110:
2100 : PRINT "DER RUN GENERATOR IST BETRIEBSBEREIT!" :
2110 : FOR N= 0 TO 213 :PRINT#1,CHR$(PEEK(N+32768));:NEXT N :CLOSE1 :GOTO2100:
2140 :
2990 REM*****
3000 : IF F$="A" THEN F$="10" : REM * HEX - DEZIMAL UMWANDLUNGSTABELLE *
3010 : IF F$="B" THEN F$="11" : REM *
3020 : IF F$="C" THEN F$="12" : REM * DIE MIT DIESEM PROGRAMM
3030 : IF F$="D" THEN F$="13" : REM * ABGESPEICHERTE VERSION MUSS
3040 : IF F$="E" THEN F$="14" : REM * ABSOLUT GELADEN WERDEN.
3050 : IF F$="F" THEN F$="15" : REM *
3060 : F=VAL(F$) : RETURN : REM * (W) 1986 VON THOMAS GUENTHER
4000 REM*****
4010 REM*
4020 REM* SYNTAX DES RUN - VERSION GENERATORS :
4030 REM*
4040 REM* SYS 32847, ANFANGSADRESSE, ENDADRESSE, STARTADRESSE, "NAME", DEVICE *
4050 REM*
4060 REM*****
READY.
```

Wenn Sie schon einmal versucht haben ein längeres Programm zu schreiben oder sich eine Ihrer älteren Produktionen zu vergegenwärtigen, dann werden Sie den neuen Befehl zu schätzen wissen. FIND ist ein direkter Befehl, er listet alle Zeilen auf, in denen die gesuchte Zeichenfolge vorkommt. Suchen Sie also in einem Programm den Namen HANS,

so geben Sie einfach FIND "HANS" oder AS="HANS": FIND AS ein. Es werden dann alle Basic-Zeilen, die das Wort HANS enthalten, nacheinander aufgelistet. Es werden aber auch alle Zeilen aufgelistet, die das Wort HANSEAT enthalten. Befehlsörter wie PRINT oder REM werden Sie mit FIND allerdings vergeblich

suchen, da diese ja nicht als Wort, sondern lediglich als TOKEN im Speicher abgelegt werden. Wenn Sie das nachstehende Programm einmal laufen gelassen haben können Sie es auch schon wieder löschen (beim ersten Mal sollten Sie es natürlich vorher auf Kassette aufnehmen) - der Befehl FIND steht Ihnen weiterhin zu Diensten.

```
170 DATA AS,10,EF,CD,9F,CE,DS,D
D,E1,32,69,AS,E5,C5,2A,83,-2568
180 DATA AE,ED,4B,81,AE,A7,ED,4
2,44,4D,2A,81,AE,ED,5B,83,-2208
190 DATA AE,E5,A7,1B,1B,ED,5
2,E1,CA,64,C0,DD,7E,00,CD,-2241
200 DATA BA,FF,57,23,4E,23,46,E
5,FD,E1,0B,0B,7E,ED,A1,E2,-2177
210 DATA FD,A4,CD,8A,FF,BA,20,F
4,D5,E5,D1,1B,CD,42,AS,D1,-2800
220 DATA 20,EA,FD,5E,01,FD,56,0
2,09,E5,D5,C1,CD,0D,E1,E1,-2267
230 DATA 1B,8B,DD,E5,C5,3A,69,A
5,47,DD,7E,00,DD,23,E6,7F,-2217
240 DATA CD,8A,FF,4F,1A,13,E6,7
F,CD,8A,FF,B9,20,02,10,E9,-2145
250 DATA C1,DD,E1,C9,3E,02,C3,9
```

```
8,CA,00,44,4E,49,46,FF,FF,-2252 (CM)
260 MEMORY &A4AF (MI)
270 adr=&A4B0 (EL)
280 FOR I=10 TO 21 (MI)
290 READ op$: IF LEFT$(OP$,1)="-" (OI)
"THEN 320 (ME)
300 op=VAL("&"+op$):POKE adr,op (ME)
310 adr=adr+1:psum=psum+op:GOTO 290 (DC)
320 IF psum+VAL(op$)=0 THEN 340 (HK)
330 PRINT "Checksum-Error in (DD)
Zeile "I":END (DG)
340 psum=0:NEXT (OD)
350 CALL &A4B0 (OD)
ENDE DES LISTINGS
```

TAPE BACKUP SYSTEM

Hierbei handelt es sich um ein universelles Kopierprogramm von Disc nach Tape und ein Backupsystem mit Disc Kennzeichnung. Es können Files in unbegrenzter Länge kopiert werden. Die Baudrate ist wählbar zwischen 2000, 3000 und 4000. Die Blocklänge auf Band ist wählbar mit 2 KB oder 36 KB. Dadurch wird eine sehr hohe Aufzeichnungsgeschwindigkeit erreicht.

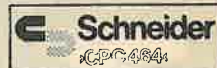
Wird der erste Block auf Band mit 2k und der nachfolgende Block mit 36k gespeichert, kann bei Basic- und Binär-Files das Programm normal mit LOAD vom Band geladen werden:

Gesichert werden alle USER, SYS-Files und PROT-Files gleichzeitig. Beim Laden werden die USER wieder richtig eingetragen, das SYS-Attribut jedoch nicht mehr gesetzt. Der PROT-Schutz

bleibt erhalten. Es können auch Tape-Files kopiert werden die nicht mit dem Backup-System erstellt wurden. Dabei werden die Namen von ungültigen Zeichen entfernt und dafür " " eingetragen.

Das Programm berücksichtigt BASIC 1.0 und BASIC 1.1, ist jedoch nur mit AMSDOS ablauf-fähig.

Backup:
Catalog-Anzeige aller Files die nicht mit Backup gesichert wurden. Alle Files werden mit Blocklänge 36k aufgezeichnet. Am Schluß wird ein EDF File auf Band geschrieben und die Directory mit Backupkennung auf Disc zurückgesichert.



Save:

Abfrage Blocklänge 36k, bei "ja" werden alle Blöcke 36k lang. Alle Blöcke 2k, bei "ja" werden alle Blöcke 2k lang, bei "nein" wird der 1. Block 2k und die nachfolgenden Blöcke 2k lang. Catalog-Anzeige aller Files, CHR\$(8&F7) markiert Backup-Files. Alle ausgewählten Files werden von Disc nach Tape kopiert.

Load:

1-9 Files werden von Tape nach Disc kopiert. Bei ENTER werden alle Files kopiert bis zum EDF-File von Backup.

Verify:

Überprüft Rape-Files. Entspricht dem CAT-Befehl bei TAPE (CAT kann nur Blöcke bis 2 KB verarbeiten, sonst wird Read Error d ausgegeben). Die File Anzahl entspricht der Angabe bei Load.

Catalog:

Disketten Catalog aller User wird ausgegeben auf dem Bildschirm. CHR\$(8&F7) markiert Backup-Files.

Drive: Drive Auswahl A oder B

Speed: Speed Write für Tape wählbar zwischen 2000, 3000, 4000 Baudopiert. Bei ENTER werden alle Files kopiert bis

Ende:

Programm beenden, anschließend kann erneut mit !R gestartet werden.

ESC

Rückkehr zum Menü, wenn nicht gerade ein Datentransfer zwischen DISC oder TAPE ausgeführt wird.

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```
10 *** Startprogramm TAPE BAC
KUP SYSTEM (c) Ekkehard Vossel
er 1.87 *** (CM)
20 * (AC)
30 MODE 1:MEMORY &90FF:PRINT"TA
PE BACKUP SYSTEM loading":PRINT
40 LOAD"backup .bin",&9100:CAL
L &9100 (JK)
50 * (AP)
60 *SAVE"backup":SAVE"backup.bi
n",b,&9100,&920 (AH)
ENDE DES LISTINGS (BP)
```

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```
10 *****
***** (NC)
20 * (CN)
30 * Data Lader -- TAPE BACK
UP SYSTEM 2.0 (c) Ekkehard V
osseler 1.87 * (HN)
40 * (DH)
50 *****
***** (BS)
60 * (AB)
70 CLS:PRINT"Diskette einlegen
und Taste druecken":PRINT:CALL
&B806 (BK)
80 GOSUB 200:IF fehl=1 THEN PRI
NT"Datei nicht gespeichert!!!!"
:GOTO 110 (MJ)
90 PRINT"Datei BACKUP.BIN wird
erzeugt." (OJ)
100 :DISC:SAVE"backup",b,&9100,
&920 (KG)
110 PRINT:END (HH)
120 * (DH)
200 PRINT:PRINT"MC-Programm wir
d erzeugt." (PI)
210 MEMORY &90FF:PRINT (LO)
220 dat=0:sz=0:dz=330:fehl=0 (IC)
230 FOR adr=&9100 TO &9A1C (PH)
240 READ byte$:byte=VAL("&"+byt
e$):dat=dat+1 (LG)
250 sz=sz+byte (OB)
260 POKE adr,byte (PP)
270 IF dat<20 AND adr<&9A1C THE
N 310 (FL)
280 READ chksum$ (BE)
290 IF VAL("&"+chksum$)<>sz THE
N fehl=1:PRINT"Fehler in Zeile
":dz (IP)
300 dz=dz+10:sz=0:dat=0 (JK)
310 NEXT adr (JL)
320 PRINT:RETURN (DH)
330 DATA 01,39,97,21,1C,9A,CD,D
1,BC,0E,00,CD,15,B9,3E,01,BC,20
,0B,21,06EF (EJ)
340 DATA 14,A9,11,28,C1,18,06,2
1,A4,A7,11,2B,C1,22,33,95,ED,53
,19,9A,071B (MF)
350 DATA 11,60,00,ED,53,46,BE,E
D,73,25,9A,3A,01,A7,32,24,9A,3A
,5D,98,07D5 (ML)
360 DATA CD,70,96,CD,8C,94,11,7
C,97,CD,0F,95,21,10,9A,CD,25,97
,1B,18,0709 (BN)
370 DATA CD,71,BC,11,9D,98,CD,0
F,95,CD,06,BB,CD,71,BC,11,4B,9B
,CD,DD,0AD7 (HJ)
380 DATA 94,CD,F4,94,ED,7B,25,9
A,11,95,97,CD,0F,95,CD,7D,BC,CD
,92,BC,0BDF (AJ)
390 DATA AF,32,32,9A,32,33,9A,2
1,00,90,22,2A,9A,22,2C,9A,3A,24
,9A,32,0655 (OO)
400 DATA 01,A7,CD,06,BB,CB,EF,F
E,62,CA,F2,91,FE,6C,CA,75,93,FE
,73,2B,0C72 (CJ)
410 DATA 5E,FE,76,CA,6F,94,FE,7
0,CA,E2,91,FE,63,CA,C1,91,FE,64
,2B,18,0C69 (PO)
420 DATA FE,65,20,D6,11,DB,99,C
D,0F,95,DF,19,9A,11,63,98,CD,DD
,94,CD,0AFB (KI)
430 DATA B4,95,1B,84,3A,00,A7,3
D,2B,02,3C,3C,CD,1D,97,3A,00,A7
,C6,41,070E (NA)
440 DATA 32,61,9B,C3,5C,91,3A,5
D,9B,3D,FE,31,20,02,3E,34,CD,70
,96,C3,0B80 (CN)
```

```
450 DATA 5C,91,11,6E,9B,CD,DD,9
4,3E,01,32,32,9A,1B,1F,11,79,9B
,CD,DD,0B82 (CB)
460 DATA 94,11,DE,9B,CD,3B,95,3
8,11,21,00,0B,22,2A,9A,11,F5,9B
,CD,3B,07B0 (MC)
470 DATA 95,30,03,22,2C,9A,CD,B
4,95,11,80,9B,CD,3B,95,11,05,99
,CD,0F,0B44 (PL)
480 DATA 95,21,80,9A,11,80,A2,0
6,40,ES,D5,C5,3B,33,CD,EF,94,C5
,ES,1A,0A47 (LO)
490 DATA FE,00,28,20,D5,CD,2E,9
6,06,0C,CD,F9,94,11,CD,9B,CD,3B
,95,3B,0760 (MB)
500 DATA 0B,CD,E5,94,D1,AF,12,1
B,07,11,05,99,CD,0F,95,D1,E1,01
,20,00,07F2 (KI)
510 DATA 09,13,C1,10,00,11,17,9
9,CD,3B,95,02,5C,91,CD,E5,94,C1
,D1,E1,0A90 (IA)
520 DATA D5,C5,C5,D5,E5,1A,FE,0
1,CC,E3,92,E1,01,20,00,09,D1,13
,C1,10,0A33 (AM)
530 DATA ED,C1,D1,3A,32,9A,FE,0
0,2B,46,21,8D,9A,1A,FE,00,2B,02
,36,01,07B2 (HK)
540 DATA 13,D5,11,20,00,19,D1,1
0,F0,11,34,9A,21,04,9A,01,0C,00
,ED,00,064B (LI)
550 DATA 06,34,3E,00,12,13,10,F
C,CD,0C,95,3E,FF,32,45,9A,32,4B
,9A,21,069D (FC)
560 DATA 02,00,22,47,9A,CD,1B,9
5,CD,EA,94,CD,8C,95,21,16,9A,CD
,57,96,0946 (HC)
570 DATA C3,50,91,7E,32,01,A7,3
2,73,9A,CD,2E,96,CD,0C,95,21,34
,9A,06,0B2F (GE)
580 DATA 0C,11,00,89,CD,77,BC,D
2,5C,91,E5,ED,43,2E,9A,3E,00,32
,33,9A,0B7F (BD)
590 DATA B8,20,0E,B9,20,0B,3E,0
1,32,33,9A,21,FF,FF,22,2E,9A,E1
,11,40,0743 (IB)
600 DATA 9A,01,0C,00,09,01,33,0
0,ED,00,3E,01,32,44,9A,3E,FF,32
,4B,9A,0624 (AP)
610 DATA ED,5B,2A,9A,B7,2A,2E,9
A,ED,52,28,02,30,09,2A,2E,9A,EB
,3E,FF,0B71 (HD)
620 DATA 32,45,9A,ED,53,47,9A,D
2,2E,9A,D5,CD,1B,95,CD,0C,95,21
,CD,50,09CA (LH)
630 DATA 95,CD,8C,95,3A,45,9A,F
E,FF,2B,0E,3A,44,9A,C6,01,32,44
,9A,ED,09AB (IJ)
640 DATA 5B,2C,9A,1B,C3,CD,7A,B
C,C9,11,7E,9B,CD,CE,96,3E,01,32
,31,9A,095C (ML)
650 DATA CD,0C,95,CD,F8,96,D2,5
9,94,CD,21,95,3A,37,9A,FE,F0,CA
,50,91,0BAF (EI)
660 DATA 3A,44,9A,5F,3A,31,9A,B
B,C2,51,94,FE,01,20,0B,21,34,9A
,11,74,077C (FO)
670 DATA 9A,01,0C,00,ED,0B,21,7
4,9A,11,34,9A,06,0C,1A,BE,C2,51
,94,23,0706 (KE)
680 DATA 13,10,F7,CD,0D,97,D2,5
9,94,CD,71,BC,11,9D,99,CD,0F,95
,3A,73,09A9 (NE)
690 DATA 9A,32,01,A7,3A,44,9A,3
2,31,9A,FE,01,20,4B,11,3F,9A,06
,03,CD,06B3 (EN)
700 DATA A9,96,3E,00,BB,C4,BA,9
6,11,3B,9A,06,0B,CD,A9,96,3E,00
,BB,20,0B5F (IE)
710 DATA 02,04,13,CD,BA,96,06,0
C,21,3C,9A,7E,FE,2E,2B,02,06,0B
,21,34,0576 (JO)
720 DATA 9A,11,80,9A,CD,8C,BC,3
0,57,11,12,00,19,3A,46,9A,77,01
,07,00,0636 (PB)
730 DATA 23,23,23,11,49,9A,EB,E
D,80,3A,31,9A,C6,01,32,31,9A,CD
```



```

.90,96,0BA1
740 DATA 3A,45,9A,FE,FF,C2,80,9
3,CD,8F,BC,CD,0C,95,3A,30,9A,3D
,FE,00,0AB0
750 DATA CA,50,91,32,30,9A,C3,7
B,93,11,8B,99,CD,0F,95,1B,07,FE
,00,2B,0B63
760 DATA 0F,CD,01,95,CD,71,BC,1
1,8E,99,CD,30,95,DA,83,93,C3,5C
,91,11,0A1F
770 DATA 83,9B,CD,CE,96,CD,0C,9
5,CD,8F,96,30,2B,CD,21,95,3A,37
,9A,FE,0AFC
780 DATA F0,CA,50,91,CD,0D,97,3
0,1B,11,02,99,CD,0F,95,3A,45,9A
,FE,FF,0A5A
790 DATA 20,DE,3A,30,9A,3D,FE,0
0,CA,50,91,32,30,9A,18,CD,FE,00
,2B,CD,0BAF
800 DATA CD,01,95,CD,71,BC,11,8
E,99,CD,38,95,3B,BE,18,80,11,4B
,9B,D5,07E6
810 DATA 3A,00,A7,C6,41,32,61,9
8,11,3E,97,06,3E,CD,F9,94,D1,CD
,0F,95,0B09
820 DATA 11,8E,9B,06,0B,CD,F9,9
4,C9,D5,11,96,9B,06,07,18,E0,11
,12,99,0B45
830 DATA 18,25,3E,0A,C3,5A,BB,3
E,0B,C3,5A,BB,11,15,99,18,16,1A
,CD,5A,06AC
840 DATA BB,13,10,F9,C9,11,AE,9
7,CD,0F,95,C6,60,CD,5A,BB,11,0F
,99,F5,0A1F
850 DATA 1A,CD,5A,BB,13,FE,00,2
0,F7,F1,C9,11,35,99,CD,0F,95,11
,34,9A,090D
860 DATA 06,0C,CD,F9,94,11,5B,9
9,CD,0F,95,3A,44,9A,21,00,00,C3
,0B,00,06E6
870 DATA CD,0F,95,CD,06,0B,CD,E
F,FE,6A,37,CB,FE,6E,37,3F,CB,FE
,FC,20,0BE4
880 DATA EE,C3,5C,91,21,00,01,C
D,80,BC,38,1D,FE,1A,2B,19,FE,0F
,20,ED,0B91
890 DATA 2A,47,9A,B7,ED,52,22,4
7,9A,21,00,00,22,2E,9A,3E,FF,32
,45,9A,075D
900 DATA C9,77,23,1B,AF,BA,20,D
7,BB,20,D4,3A,33,9A,FE,01,CD,21
,FF,FF,0A72
910 DATA 22,2E,9A,C9,11,47,99,C
D,1E,95,CD,6E,BC,21,34,9A,11,40
,00,3E,0799
920 DATA 2C,CD,9E,BC,30,AB,ED,5
B,47,9A,21,00,01,3E,16,CD,9E,BC
,30,9D,0BC1
930 DATA CD,71,BC,C9,CD,1A,97,F
E,90,2B,92,3A,00,A7,32,20,9A,FE
,01,20,0975
940 DATA 05,3A,DF,AB,18,03,3A,9
F,AB,32,22,9A,06,00,FE,C1,2B,06
,04,FE,0745
950 DATA 01,2B,01,04,7B,32,21,9
A,21,15,9A,CD,57,96,CD,F4,94,21
,80,9A,07AD
960 DATA 11,80,A2,06,40,C5,D5,E
5,16,00,7E,FE,10,30,26,01,0C,00
,09,7E,06B4
970 DATA FE,00,20,10,23,3A,32,9
A,FE,00,2B,05,7E,FE,01,2B,10,E1
,ES,CD,07D7
980 DATA 2E,96,CD,0F,95,11,76,9
9,CD,0F,95,16,01,01,20,00,E1,09
,7A,01,0733
990 DATA 12,C1,13,10,C4,C9,23,1
1,34,9A,D5,01,0B,00,ED,80,3E,2E
,12,13,0691
1000 DATA 06,03,7E,E6,7F,12,13,
23,10,F8,23,7E,FE,01,3E,F7,2B,0
2,3E,20,0699
1010 DATA 12,13,3E,00,12,D1,C9,
CD,2E,97,21,80,9A,ED,5B,20,9A,3
A,22,9A,07D4
1020 DATA 4F,06,04,DF,27,9A,24,

```

```

(MP) 24,0C,10,F8,C9,32,5D,9B,FE,34,2
B,10,FE,07AD
(FN) 1030 DATA 33,2B,05,21,A7,00,18,
03,21,6F,00,3E,0A,18,05,3E,00,2
1,54,00,02EB
(PB) 1040 DATA CD,6B,BC,C9,21,00,01,
ED,5B,47,9A,7E,CD,95,BC,D2,5C,9
1,23,1B,079E
(PH) 1050 DATA 3E,00,BA,20,F2,8B,20,
EF,C9,1A,FE,00,2B,07,FE,20,C0,1
B,10,F5,08E2
(PH) 1060 DATA C9,3E,20,12,18,F7,C5,
1A,01,12,00,21,F2,99,ED,B1,20,0
3,3E,2D,0712
(PH) 1070 DATA 12,1B,C1,10,ED,C9,CD,
DD,94,11,61,99,CD,0F,95,CD,06,8
B,CD,5A,0A23
(PH) 1080 DATA BB,CD,0C,95,FE,0D,20,
04,3E,FF,18,0C,FE,31,DA,5C,91,F
E,3A,D2,09B9
(PH) 1090 DATA 5C,91,D6,30,32,30,9A,
C9,11,7A,99,CD,0F,95,CD,6E,BC,2
1,34,9A,0933
(PH) 1100 DATA 11,40,00,3E,2C,CD,A1,
BC,C9,21,00,01,ED,5B,47,9A,3E,1
6,CD,A1,07BB
(PH) 1110 DATA BC,C9,3A,00,A7,C6,C1,
32,14,9A,21,14,9A,CD,2E,97,3E,0
0,DF,27,0872
(PH) 1120 DATA 9A,C9,CD,D4,BC,22,27,
9A,79,32,29,9A,C9,17,9A,C3,2F,9
1,04,02,0914
(PH) 1130 DATA 1B,1A,00,4F,00,00,1E,
14,20,20,20,20,54,61,70,65,20,4
2,61,63,03E3
(PH) 1140 DATA 6B,75,70,20,53,79,73,
74,65,6D,20,32,2E,30,1F,34,01,5
3,70,65,0621
(PH) 1150 DATA 65,64,20,34,30,30,30,
20,20,20,20,20,20,44,72,69,7
6,65,20,04A7
(PH) 1160 DATA 1F,3C,01,2B,63,29,20,
20,45,2E,56,6F,73,73,65,6C,65,7
2,20,20,0556
(PH) 1170 DATA 31,2E,38,37,00,1F,1B,
03,42,20,20,20,42,61,63,6B,75,7
0,20,53,0483
(PH) 1180 DATA 69,63,6B,65,72,6B,65,
69,74,73,6B,6F,70,69,65,1F,1B,0
5,53,20,06F2
(PH) 1190 DATA 2D,20,53,61,76,65,20,
20,20,20,20,44,69,73,63,20,2D,3
E,20,54,04FE
(PH) 1200 DATA 61,70,65,1F,1B,07,4C,
20,20,20,4C,6F,61,64,20,20,20,2
0,20,54,04A4
(PH) 1210 DATA 61,70,65,20,2D,3E,20,
44,69,73,63,1F,1B,09,56,20,2D,2
0,56,65,0525
(PH) 1220 DATA 72,69,66,79,20,20,20,
54,61,70,65,20,70,72,75,65,66,6
5,6E,1F,060B
(PH) 1230 DATA 1B,0B,43,20,20,20,43,
61,74,61,6C,6F,67,20,20,44,69,7
3,63,1F,0573
(PH) 1240 DATA 1B,0D,44,20,20,20,44,
72,69,76,65,20,41,2F,42,1F,1B,0
F,53,20,0461
(PH) 1250 DATA 2D,20,53,70,65,65,64,
20,57,72,69,74,65,1F,1B,11,45,2
0,2D,20,0566
(PH) 1260 DATA 45,6E,64,65,0A,0D,00,
1F,22,01,44,69,73,63,20,9A,9A,2
0,54,61,05B1
(PH) 1270 DATA 70,65,1F,3A,01,34,1F,
4B,01,20,00,1F,27,01,43,61,74,6
1,6C,6F,04B9
(PH) 1280 DATA 67,00,1F,27,01,20,42,
61,63,6B,75,70,00,1F,2B,01,F3,0
0,1F,27,04A5
(PH) 1290 DATA 01,F2,00,1F,22,01,56,
65,72,69,66,79,20,00,18,1A,00,4
F,02,1B,0465
(PH) 1300 DATA 1E,14,1B,1A,00,4F,00,
00,1E,0A,0A,0D,54,61,73,74,65,2
0,64,72,03E9

```

```

(FN) 1310 DATA 75,65,63,6B,65,6E,21,
00,0A,0A,0D,41,6C,6C,65,20,46,6
9,6C,65,05DB
(DM) 1320 DATA 73,20,4B,6F,70,69,65,
72,65,6E,20,2B,4A,2F,4E,29,00,2
0,20,4B,0593
(NE) 1330 DATA 6F,70,69,65,72,65,6E,
20,2B,4A,2F,4E,29,00,42,6C,6F,6
3,6B,6C,06B1
(NI) 1340 DATA 61,65,6E,67,65,20,33,
36,6B,20,2B,4A,2F,4E,29,00,00,4
1,6C,6C,0552
(MB) 1350 DATA 65,20,42,6C,6F,65,63,
6B,65,20,32,0A,00,0B,0B,0B,0B,0
8,2E,20,040C
(KM) 1360 DATA 2E,20,2E,0A,0D,00,0D,
14,00,0C,00,0A,0D,54,41,50,45,2
0,41,75,02D7
(PC) 1370 DATA 66,6E,61,6B,6D,65,62,
65,72,65,69,74,20,2B,4A,2F,4E,2
9,0D,0B,063A
(HE) 1380 DATA 00,14,0A,0D,44,49,53,
43,20,4C,6F,61,64,69,6E,67,20,2
0,00,0D,0479
(MJ) 1390 DATA 0B,54,41,50,45,20,53,
61,76,69,6E,67,20,20,20,00,20,2
0,42,6C,050B
(JE) 1400 DATA 6F,63,6B,20,00,46,69,
6C,65,20,41,6E,7A,61,6B,6C,20,2
B,31,2D,0601
(DA) 1410 DATA 39,2C,45,4E,54,29,20,
20,20,00,0D,54,41,50,45,20,4C,6
F,61,64,04AC
(FJ) 1420 DATA 69,6E,67,20,20,14,00,
0D,54,41,50,45,20,46,6F,75,6E,6
4,20,20,0525
(EA) 1430 DATA 20,20,0A,0D,00,0D,44,
49,53,43,20,53,61,76,69,6E,67,2
0,20,20,046F
(PI) 1440 DATA 0D,00,20,20,20,20,52,
65,61,64,20,45,72,72,6F,72,20,0
0,0D,57,04B7
(NH) 1450 DATA 69,65,64,65,72,6B,6F,
6C,65,6E,20,2B,4A,2F,4E,29,14,0
0,20,20,05AB
(MN) 1460 DATA 20,20,4F,6B,0A,0D,00,
04,02,52,75,65,63,6B,6B,65,6B,7
2,20,6D,054B
(CM) 1470 DATA 69,74,20,20,7C,52,0A,
0A,0D,00,3C,3E,2E,2C,3B,3A,3D,5
B,5D,25,046F
(CI) 1480 DATA 7C,2B,29,2F,5C,20,5F,
7F,45,4F,46,6F,20,20,20,20,2E,4
2,41,53,05A4
(CB) 1490 DATA 44,49,53,C3,00,84,85,
D2,00,00,00,00,00,037E
(GI) 2000
(AD) 2010
(KF) 2020
(HF) 2030
(HF) 2040
(AA) 2050
(FB) 2060
(CB) 2070
(FL) 2080
(PL) 3000
(DP) 3010
(DC) 3020
(OK)

```

ENDE DES LISTINGS



DIE SCHWARZEN DIAMANTEN

Wie wäre es mit einem Textadventure?

Den "schwarzen Diamanten" sollen Sie ihr Geheimnis entlocken. Wie? Na, wir werden hier doch nichts verraten! Die Textfiles geben beim Abtippen noch keinen Sinn. Alles, was Sie sonst noch wissen müssen, wird nach dem Starten des Programms erklärt. Auch zum Programm sind keine weiteren Hinweise erforderlich.

Auf in's Abenteuer!



SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```

10 *****
20 *
30 * Die schwarzen *
40 *
50 * Diamanten *
60 *
70 *
80 * (c) 1986 *
90 *
100 * Cebetronic *
110 *
120 *
130 *****
140 INK 0,0:BORDER 0
150 MODE 1
160 INK 1,24:INK 2,18:INK 3,6
170 SYMBOL AFTER 255
180 SYMBOL 255,0,0,129,90,36,90
,129,0
190 DIM ra$(33),du$(33,6),obj$(3
B),obra$(3B),verb$(13),cobj$(3B)
,richt$(6)
200 LOCATE 9,2:PEN 3:PRINT"Die
schwarzen Diamanten"
210 LOCATE 9,3:PEN 2:PRINT"----
"
220 PEN 1
230 LOCATE 5,6:PRINT"Sie muesse
n eine Aufgabe erfuehlen,"
240 LOCATE 5,7:PRINT"die Ihnen
im Laufe des Spieles mit-"
250 LOCATE 5,8:PRINT"geteilt wi

```

```

rd."
260 LOCATE 5,9:PRINT"Sie geben
mir Ihre Anweisungen in"
270 LOCATE 5,10:PRINT"kurzen Sa
etzen.z.B. Nimm Stein"
280 LOCATE 5,11:PRINT"Folgende
Verben verstehe ich:"
290 PEN 2
300 LOCATE 5,12:PRINT"Bewege
Drehe Wurf Nimm"
310 LOCATE 5,13:PRINT"Schlafe
Benuetze Lege Sage"
320 LOCATE 5,14:PRINT"Untersuch
e Warte Ende Hilfe"
330 PEN 1
340 LOCATE 5,15:PRINT"Auf den B
efehl"+CHR$(34)+"Hilfe"+CHR$(34
)+""gebe ich Ihnen"
350 LOCATE 5,16:PRINT"einen Tip
, oder ich weise auf eine "
360 LOCATE 5,17:PRINT"Gefahr hi
n."
370 LOCATE 5,18:PRINT"Ich kann
eine gewisse Anzahl Dinge"
380 LOCATE 5,19:PRINT"tragen un
d kann mich durch Eingabe"
390 LOCATE 5,20:PRINT"von N S O
W Ob U fortbewegen."
400 LOCATE 5,21:PRINT"Dazu muss
jedoch ein geeigneter Weg"
410 LOCATE 5,22:PRINT"existiere
n."
420 PEN 2
430 LOCATE 5,24:PRINT"Alles kla
r?"
440 *****
450 ** Daten lesen **
460 *****
470 RESTORE 1370
480 FOR a=1 TO 33
490 READ ra$(a)
500 FOR b=1 TO 6

```



```

870 IF obra(a)=ra THEN PRINT ob
j$(a)+";:nichts=1
880 NEXT
890 IF nichts=0 THEN PRINT"nich
ts Erwahnenswertes."
900 PRINT:PRINT
910 PEN 2
920 PRINT"ich trage :";
930 PEN 3
940 FOR a=1 TO oanz
950 PRINT tobj$(a)+";:
960 NEXT a
970 IF oanz=0 THEN PRINT"nichts
."
980 PRINT:PRINT:PRINT:PRINT
990 PRINT STRING$(40,255):PRINT
:PRINT
1000 IF ra=1 THEN PRINT"ich hab
e mich im Labyrinth verirrt.
ich bin verhungert.":GOTO 210
0
1010 IF ra=10 THEN PRINT"ich ha
be das Schiff verlassen und bin
auf einen giftigen Seeigel g
etreten.":GOTO 2100
1020 IF ra=11 THEN PRINT"Ein Or
kan hat das Schiff erfasst.
Das Schiff ist gekentert, ic
h bin er- trunken.":GOTO 2100
1030 IF ra=24 AND bed1=0 THEN P
RINT"ich bin abgestuerzt und bi
n auf einem Felsen aufgeschla
gen.Eine Krabbe klet- tert ueb
er mein Gesicht...":GOTO 2400
1040 IF ra=14 THEN PRINT"Der Wl
nd hat dich auf eine Insel ver-
trieben.":FOR a=1 TO 3000:NE
XT a:ra=30:GOTO 720
1050 IF ra=19 THEN PRINT"ich wu
rde in dem Schacht eingeklemmt u
nd Wasser ist eingedrungen...":
FOR a=1 TO 3000:NEXT a:GOTO 210
0
1060 PRINT"Was soll ich tun?";
:FEN 2:INPUT"ant$
1070 ant$=UPPER$(ant$)
1080 IF ant$="N"AND du(ra,1)<>0
THEN ra=du(ra,1):GOTO 720
1090 IF ant$="S"AND du(ra,2)<>0
THEN ra=du(ra,2):GOTO 720
1100 IF ant$="O"AND du(ra,3)<>0
THEN ra=du(ra,3):GOTO 720
1110 IF ant$="W"AND du(ra,4)<>0
THEN ra=du(ra,4):GOTO 720
1120 IF ant$="OB"AND du(ra,5)<>
0 THEN ra=du(ra,5):GOTO 720
1130 IF ant$="U"AND du(ra,6)<>0
THEN ra=du(ra,6):GOTO 720
1140 IF LEN(ant$)<4 THEN PRINT"
Dahin fuehrt kein Weg.":FOR a=1
TO 3000:NEXT:GOTO 720
1150 zeitw$=LEFT$(ant$,4)
1160 richtig=0:FOR a=1 TO 13
1170 IF zeitw$=verb$(a)THEN ueb
erp=a:richtig=1
1180 NEXT a
1190 IF richtig=0 THEN PRINT"Da
s verstehe ich nicht.":FOR a=1
TO 3000:NEXT:GOTO 720
1200 IF ueberp=7 AND ra<23 THE
N PRINT"ich habe den ganzen Tag
gepennt. Aber nichts is
t geschehen.":FOR a=1 TO 4000:N
EXT:GOTO 720 ELSE IF ueberp=7 T
HEN GOTO 2180
1210 IF ueberp=10 THEN GOTO 225
0
1220 IF ueberp=12 THEN END
1230 IF ueberp=13 THEN PRINT"Ei
ne Unmenge Zeit ist vergangen,
aber nichts ist passiert.Fast
waere ich eingeschlafen."
:FOR a=1 TO 5000:NEXT:GOTO 720
1240 ding$=RIGHT$(ant$,LEN(ant$
)-4)
1250 IF LEN(ding$)<4 THEN PRINT

```

```

"Das verstehe ich nicht.":FOR a
=1 TO 3000:NEXT:GOTO 720
1260 IF LEFT$(ding$,1)=" "AND L
EN(ding$)>4 THEN ding$=RIGHT$(d
ing$,LEN(ding$)-1):GOTO 1290
1270 ding$=RIGHT$(ding$,LEN(din
g$)-1)
1280 GOTO 1250
1290 ding$=LEFT$(ding$,4)
1300 FOR a=1 TO 38
1310 IF ding$=cobj$(a)THEN o=a:
richtig=1
1320 NEXT a
1330 IF ueberp=11 THEN PRINT"Da
s verstehe ich nicht.":FOR a=1
TO 3000:NEXT:GOTO 720
1340 IF richtig=0 THEN PRINT"Da
s verstehe ich nicht.":FOR a=1
TO 3000:NEXT:GOTO 720
1350 ON ueberp GOSUB 2530,2960,
3260,3430,3620,3730,1,3900,4030
,1,1,1,1
1360 GOTO 720
1370 DATA in einem Labyrinth.,0
,0,0,0,0,0
1380 DATA in einer Hoehle.,5,1,
1,3,0,0
1390 DATA vor einem Altar in ei
ner Hoehle.,0,0,2,0,0,0
1400 DATA neben einem Felsen.,8
,0,0,0,17,0
1410 DATA im Wald.,9,0,0,0,0,0
1420 DATA am Strand.,0,0,7,0,0,
0
1430 DATA bei den Klippen.,0,0,
0,0,0,0
1440 DATA in einem dichten Wald
.,0,4,9,7,0,0
1450 DATA in einer Bucht.,12,5,
10,8,0,0
1460 DATA im seichten Wasser.,0
,0,0,9,0,0
1470 DATA in einem Orkan.,0,0,0
,0,0,0
1480 DATA auf dem Meer.,15,9,13
,11,0,0
1490 DATA bei einem Riff.,0,0,0
,12,0,0
1500 DATA in einem Sturm.,0,0,0
,0,0,0
1510 DATA auf dem weiten Meer.,
16,12,0,14,0,0
1520 DATA auf dem Meer.,16,15,0
,0,0,0
1530 DATA auf dem Felsen.,0,0,0
,0,0,4
1540 DATA in einem Schacht.,0,0
,0,0,4,19
1550 DATA in einem Schacht.,0,0
,0,0,0,0
1560 DATA in einem Moor.,24,0,0
,0,0,0
1570 DATA auf einem Vulkan.,25,
0,22,0,0,0
1580 DATA im Wald.,0,0,23,21,0,
0
1590 DATA im Wald der Wunder.,2
7,0,0,22,0,0
1600 DATA bei den Klippen.,28,2
0,25,0,0,0
1610 DATA auf einem Pfad.,0,0,0
,24,0,0
1620 DATA in einem verfallenen
Haus.,0,0,27,0,0,33
1630 DATA in einem Garten.,31,2
3,0,26,0,0
1640 DATA an einem weissen Sand
strand.,0,24,29,0,0,0
1650 DATA an der Kueste.,0,0,30
,28,0,0
1660 DATA auf einem morschen St
eg.,0,0,0,29,0,0
1670 DATA auf einer Lichtung.,0
,27,0,0,0,0
1680 DATA in der Tropfsteinhoeh
le.,0,0,0,0,24,0

```

```

1690 DATA im dunklen Keller.,0,
0,0,0,26,0
1700 DATA Seil,SEIL,0
1710 DATA Horn,HORN,0
1720 DATA Brecheisen,BREC,0
1730 DATA Smaragd,SMAR,18
1740 DATA Truhe,TRUH,0
1750 DATA Saphier,SAPH,0
1760 DATA Diamant,DIAM,0
1770 DATA Fledermaus,FLED,2
1780 DATA Pfuete,PfUE,3
1790 DATA Kerze,KERZ,3
1800 DATA Statue,STAT,3
1810 DATA Leiter,LEIT,4
1820 DATA Baum,BAUM,5
1830 DATA Fischernetz,FISC,6
1840 DATA Kraehennest,NEST,7
1850 DATA Kraeheneier,EIER,7
1860 DATA Zweig,ZWEI,8
1870 DATA Schiffwrack,WRAC,9
1880 DATA Knochen,KNOC,9
1890 DATA Meer,MEER,12
1900 DATA Haie,HAIE,13
1910 DATA Totenkopf,KOPF,20
1920 DATA Pfahl,PFAL,20
1930 DATA Schuessel,SCHL,17
1940 DATA Bild,BILD,26
1950 DATA Aepfel,AEFF,27
1960 DATA Palme,PALM,29
1970 DATA Feuerstelle,FEUE,29
1980 DATA Fackel,FACK,29
1990 DATA Wasser,WASS,30
2000 DATA Sand,SAND,30
2010 DATA Skelett,SKEL,30
2020 DATA Steinblock,STEI,32
2030 DATA Schaufel,SCHA,33
2040 DATA Staub,STAU,33
2050 DATA Tropfstein,TROP,32
2060 DATA Fels,FELS,4
2070 DATA THYR,0
2080 DATA UNTE,BEWE,DREH,WIRF,N
IMM,LEGE,SCHL,BENU,SAGE,HILF,BE
FE,ENDE,WART
2090 DATA Norden,Sueden,Osten,W
esten,oben,unten
2100 PRINT:PRINT:PEN 1:PAPER 0:
PRINT" Leertaste fuer N
eustart "
2110 a$=INKEY$
2120 WHILE a$="":a$=INKEY$:WEND
2130 IF a$=" " THEN RUN ELSE END
2140 END
2150 *****
2160 ** Traum **
2170 *****
2180 PRINT"ich schlafe ein und
traeume: ich laufe a
uf einer Insel umher, es ist ei
ne schoene Tropeninsel.Auf der
Insel steht eine grosse Villa.
ich hoere Schritte.Ploetz
lich sehe ich etwas. Thy....."
2190 PRINT"ich wache auf.Was is
t Thy...?":a$=INKEY$:WHILE a$="
":a$=INKEY$:WEND
2200 FOR a=1 TO 3000:NEXT
2210 GOTO 720
2220 *****
2230 ** Hilfe **
2240 *****
2250 IF ra=2 THEN PRINT"Destlic
h und suedlich sehe ich viele
Gaenge.":FOR a=1 TO 3000:NEXT
:GOTO 720
2260 IF ra=3 THEN PRINT"Wie wae
r's mit einem kleinen Opfer?":
FOR a=1 TO 3000:NEXT:GOTO 720
2270 IF ra=4 THEN PRINT"Der Fel
sen wackelt.":FOR a=1 TO 3000:N
EXT:GOTO 720
2280 IF ra=5 THEN PRINT"Sieh di
r doch den Baum an!!!":FOR a=1
TO 3000:NEXT:GOTO 720
2290 IF ra=6 THEN PRINT"Willst
du Fische fangen???":FOR a=1 TO
3000:NEXT:GOTO 720

```

```

2300 IF ra=7 THEN PRINT"Hilf di
r selbst sonst hilft dir keiner
!":FOR a=1 TO 3500:NEXT:GOTO 72
0
2310 IF ra=8 THEN PRINT"ich seh
den Wald vor lauter Baeumen
nicht!!!":FOR a=1 TO 3500:NEX
T:GOTO 720
2320 IF ra=9 THEN PRINT"Der Kno
chen stammt von einem Menschen.
":FOR a=1 TO 3000:NEXT:GOTO 72
0
2330 IF ra=12 THEN PRINT"Kein K
ommentar!!!":FOR a=1 TO 2000:NE
XT:GOTO 720
2340 IF ra=13 THEN PRINT"Die Ha
ie scheinen hungrig zu sein.":
FOR a=1 TO 3000:NEXT:GOTO 720
2350 IF ra=15 THEN PRINT"Im Wes
ten zieht ein Gewitter auf.":F
OR a=1 TO 3000:NEXT:GOTO 720
2360 IF ra=16 THEN PRINT"Versuc
h's mal mit Sueden.":FOR a=1 T
O 3000:NEXT:GOTO 720
2370 IF ra=20 THEN PRINT"oh sch
aurig ist's uebers Moor zu geh'
n.":FOR a=1 TO 3000:NEXT:GOTO 7
20
2380 IF ra=21 THEN PRINT"so ein
Kaminfeuerchen ist warm!":FOR
a=1 TO 3000:NEXT:GOTO 720
2390 IF ra=22 THEN PRINT"Sehr d
unkel ist's hier!!!":FOR a=1 T
O 3000:NEXT:GOTO 720
2400 IF ra=23 THEN PRINT"ich bi
n sehr muede!!! Uahhhh!":FOR a=
1 TO 3000:NEXT:GOTO 720
2410 IF ra=24 THEN PRINT"ich bi
n kein Freikletterer!":FOR a=1
TO 3000:NEXT:GOTO 720
2420 IF ra=25 THEN PRINT"vom Su
eden regnet es Asche!!!":FOR a=
1 TO 3000:NEXT:GOTO 720
2430 IF ra=26 THEN PRINT"Das Bi
ld ist ein Aquarell!":FOR a=1
TO 3000:NEXT:GOTO 720
2440 IF ra=27 THEN PRINT"Erinne
rst du dich an Rotkaeppchen?":
FOR a=1 TO 3000:NEXT:GOTO 720
2450 IF ra=28 THEN PRINT"2m unt
er dir...":FOR a=1 TO 3000:NEXT
:GOTO 720
2460 IF ra=29 THEN PRINT"Palmen
sind selten.":FOR a=1 TO 3000:
NEXT:GOTO 720
2470 IF ra=30 THEN PRINT"Kaum g
elandet und schon so neugierig!
!":FOR a=1 TO 3000:NEXT:GOTO 7
20
2480 IF ra=31 THEN PRINT"Mach m
al Musik!":FOR a=1 TO 3000:NEXT
:GOTO 720
2490 IF ra=32 THEN PRINT"Hinter
dem Stein leuchtet es roetlich
.":FOR a=1 TO 3000:NEXT:GOTO 72
0
2500 IF ra=33 THEN PRINT"Haaats
chiiii!!!Wo ist ein Staubsauge
r!":FOR a=1 TO 2000:NEXT:GOTO 7
20
2510 GOTO 720
2520 *****
2530 **** untersuche ****
2540 *****
2550 IF o=1 AND ra=obra(o)THEN
PRINT"Es ist ein sorgfaeltig ge
flochtenes Hanfseil."
2560 IF o=2 AND ra=obra(o)THEN
PRINT"Das Horn koennte noch fun
ktionieren."
2570 IF o=3 AND ra=obra(o)THEN
PRINT"Es sieht sehr stabil aus.
"
2580 IF o=4 AND ra=obra(o)THEN
PRINT"Sehr wertvoller Stein von
30 Karat."

```

```

2590 IF o=5 AND ra=obra(o)THEN
PRINT"Die Truhe koennte man lei
cht oeffnen."
2600 IF o=6 AND ra=obra(o)THEN
PRINT"Sehr wertvoller Stein von
60 Karat."
2610 IF o=7 AND ra=obra(o)THEN
PRINT"Sehr wertvoller Stein von
15 Karat."
2620 IF o=8 AND ra=obra(o)THEN
PRINT"Die Fledermaus hat etwas
Glaenzendes im Mund."
2630 IF o=9 AND ra=obra(o)THEN
PRINT"Das ist nicht Wasser,sond
ern Blut!!"
2640 IF o=11 AND ra=obra(o)THEN
PRINT"Im Mund und in den Augen
der Statue waren einmal Ed
elsteine."
2650 IF o=12 AND ra=obra(o)THEN
PRINT"Die Leiter fuehrt nach o
ben."
2660 IF o=13 AND ra=obra(o)THEN
PRINT"Der Baum haelt nicht fes
t am Boden."
2670 IF o=14 AND ra=obra(o)THEN
PRINT"Das Netz sollte geflickt
werden."
2680 IF o=15 AND ra=obra(o)THEN
PRINT"Die Kraehe ist gerade we
ggefliegen."
2690 IF o=16 AND ra=obra(o)THEN
PRINT"Die Eier sind noch warm.
"
2700 IF o=17 AND ra=obra(o)THEN
PRINT"Die Zweige sind verbrann
t."
2710 IF o=18 AND ra=obra(o)THEN
PRINT"Das Schiff heisst Espera
nta."
2720 IF o=19 AND ra=obra(o)THEN
PRINT"Er liegt schon lange da.
"
2730 IF o=20 AND ra=obra(o)THEN
PRINT"Das Wasser ist salzig."
2740 IF o=21 AND ra=obra(o)THEN
PRINT"Sie schwimmen mordgierig
um das Boot."
2750 IF o=23 AND ra=obra(o)THEN
PRINT"Der Pfahl wackelt."
2760 IF o=24 AND ra=obra(o)THEN
PRINT"Einem Schuessel kann ma
n immer ge- brauchen."
2770 IF o=25 AND ra=obra(o)THEN
PRINT"Auf dem Bild ist ein alt
er Mann dar- gestellt.Er sag
t: Vor langer Zeit raubte
n ein paar Piraten wertvolle
Steine einer Statue. Br
inge den Diamant, den Saphie
r und den Smaragd dorthin."
2780 IF o=25 AND ra=obra(o)THEN
PRINT"Du wirst dort ein Loesun
gsword er- fahren.Sag es m
ir!":a$=INKEY$:WHILE a$="":a$=I
NKEY$:WEND
2790 IF o=26 AND ra=obra(o)THEN
PRINT"Sie sind auf einer Seite
schoen rot."
2800 IF o=22 AND ra=obra(o)THEN
PRINT"Er grinst dich an."
2810 IF o=27 AND ra=obra(o)THEN
PRINT"Oben liegt ein Seil."rob
ra(1)=29
2820 IF o=28 AND ra=obra(o)THEN
PRINT"Das Feuer flackert lusti
g."
2830 IF o=10 AND ra=obra(o)THEN
PRINT"Sie ist fast nur mehr ei
n Wachsklumpen."
2840 IF o=30 AND ra=obra(o)THEN
PRINT"Na wie schaut Wasser sch
on aus."
2850 IF o=31 AND ra=obra(o)THEN
PRINT"Schoener weissglaenzende
r Sand."

```

```

2860 IF o=32 AND ra=obra(o)THEN
PRINT"Verweste Hautfetzen haen
gen noch daran."
2870 IF o=33 AND ra=obra(o)THEN
PRINT"Er ist leicht wegzubeweg
en."
2880 IF o=34 AND ra=obra(o)THEN
PRINT"Naja ich koennte so etwa
s brauchen."
2890 IF o=29 AND ra=obra(o)OR(
o=36 AND ra=obra(o))OR(o=35 AND
ra=obra(o))THEN PRINT"ich sehe
nichts besonderes."
2900 IF o=37 AND ra=obra(o)THEN
PRINT"Auf den ersten Blick sch
eint ein Spalt zwischen dem Fe
ls und dem Erdboden zu sein."
2910 FOR a=1 TO 3000:NEXT
2920 GOTO 720
2930 *****
2940 ** bewege **
2950 *****
2960 IF o=1 AND ra=obra(o)THEN
PRINT"Das Seil verwickelt sich
sonst ist nichts passiert."
2970 IF o=2 AND ra=obra(o)THEN
PRINT"Warum denn das??"
2980 IF o=5 AND ra=obra(o)THEN
PRINT"Die laesst sich nicht bew
egen."
2990 IF o=8 AND ra=obra(o)THEN
PRINT"Die Fledermaus fliegt dav
on."
3000 IF o=9 AND ra=obra(o)THEN
PRINT"ohne Behaelter geht das n
icht."
3010 IF o=11 AND ra=obra(o)THEN
PRINT"Die ist am Altar angesch
raubt."
3020 IF o=13 AND ra=obra(o)THEN
PRINT"Der Baum wackelt sehr st
ark.Das brint aber nicht den
gewuenschten Erfolg."
3030 IF o=18 AND ra=obra(o)THEN
PRINT"Das ist viel zu schwer."
3040 IF o=19 AND ra=obra(o)THEN
PRINT"ich will ihn nicht angre
ifen!!"
3050 IF o=20 AND ra=obra(o)THEN
PRINT"ich bin kein Gott.Das ka
nn nur der Wind."
3060 IF o=21 AND ra=obra(o)THEN
PRINT"Dabei wuerde ich meine A
rme riskieren."
3070 IF o=22 AND ra=obra(o)THEN
PRINT"ich will ihn nicht angre
ifen!!"
3080 IF o=23 AND ra=obra(o)THEN
PRINT"Unter dem Pfahl kommt ei
n Brecheisen hervor."obra(3
)=28
3090 IF o=24 AND ra=obra(o)THEN
PRINT"Das brint nichts !!!"
3100 IF o=25 AND ra=obra(o)THEN
PRINT"Das Bild faellt herunter
.Sonst ist nichts passiert
."
3110 IF o=26 AND ra=obra(o)THEN
PRINT"Der Apfel faellt vom Bau
m."
3120 IF o=27 AND ra=obra(o)THEN
PRINT"Die laesst sich nicht be
wegen."
3130 IF o=28 AND ra=obra(o)THEN
PRINT"Wie stellst du dir das v
or??"
3140 IF o=29 AND ra=obra(o)THEN
PRINT"Beinahe waere das Feuer
ausgegangen."
3150 IF o=30 AND ra=obra(o)THEN
PRINT"ich bin kein Gott.Das ka
nn nur der Wind."
3160 IF o=31 AND ra=obra(o)THEN
PRINT"Willst du eine Sandburg
bauen??"

```



```

3170 IF o=32 AND ra=obra(o) THEN
PRINT "Ich will das nicht angre-
ifen!!"
3180 IF o=33 AND ra=obra(o) THEN
PRINT "Magma ist unter dem Stei-
nblock aus- getreten. Die Ho-
ehle füllt sich mit Magma.
Ich bin nach oben geflüchtet."
:du(25,2)=21:ra=24:FOR a=1 TO 1
000:NEXT
3190 IF o=37 AND ra=obra(o) THEN
PRINT "Der Fels beginnt sich zu
drehen. Ein Schacht nach unten
wird frei." :du(4,6)=18
3200 IF (o=3 OR o=4 OR o=6 OR o=
7 OR o=10 OR o=12 OR o=14 OR o=
15 OR o=16 OR o=34 OR o=35) AND
ra=obra(o) THEN PRINT "Hab ich, ab-
er es hat nichts gebracht."
3210 IF (o=17 OR o=36) AND ra=obr-
a(o) THEN PRINT "Das geht nicht."
3220 FOR a=1 TO 4000:NEXT:GOTO
720
3230 *****
3240 ** drehe **
3250 *****
3260 IF o=5 AND ra=obra(o) THEN
PRINT "Die Truhe ist viel zu sch-
wer."
3270 IF o=8 AND ra=obra(o) THEN
PRINT "Sie fliegt mir davon."
3280 IF o=9 AND ra=obra(o) THEN
PRINT "Dazu brauchte ich einen
Behälter."
3290 IF o=11 AND ra=obra(o) THEN
PRINT "Sie ist am Altar angesch-
raubt."
3300 IF o=13 AND ra=obra(o) THEN
PRINT "Der Baum laesst sich gan-
z leicht zur Seite drehen. De-
r Eingang in eine Hoehle wird s-
ichtbar." :du(5,2)=2
3310 IF o=18 AND ra=obra(o) THEN
PRINT "Das ist doch viel zu sch-
wer."
3320 IF o=19 AND ra=obra(o) THEN
PRINT "Mir ekelt davor ich will
ihn nicht an- greifen."
3330 IF o=20 AND ra=obra(o) THEN
PRINT "Ich bin kein Gott. Das ka-
nn nur der Wind."
3340 IF o=21 AND ra=obra(o) THEN
PRINT "Der Hai hat nach mir ge-
chnappt. Er reisst mich vom
Boot und zieht mich unter
Wasser....":GOTO 2100
3350 IF (o=22 OR o=32) AND ra=obr-
a(o) THEN PRINT "Mir ekelt davor.
Ich will das nicht an- greifen."
3360 IF o=23 AND ra=obra(o) THEN
PRINT "Er bewegt sich leicht. So
nst passiert nichts."
3370 IF o=25 AND ra=obra(o) THEN
PRINT "Die Rueckseite sieht nic-
ht sehr inter- essant aus."
3380 IF o=30 AND ra=obra(o) THEN
PRINT "Dazu brauchte ich einen
Behälter."
3390 IF o=37 AND ra=obra(o) THEN
PRINT "Er bewegt sich leicht. So
nst passiert nichts."
3400 IF (o=1 OR o=2 OR o=3 OR o=
4 OR o=6 OR o=7 OR o=10 OR o=12
OR o=14 OR o=15 OR o=16 OR o=1
7 OR o=24 OR o=26 OR o=29 OR o=
34) AND ra=obra(o) THEN PRINT "Das
hat nichts Aussergewöhnliches
ge- bracht."
3410 FOR a=1 TO 3500:NEXT:GOTO
720
3420 *****
3430 ** Wirf **
3440 *****
3450 IF o=1 AND (ra=obra(o) OR ob-
ra(o)=1) THEN PRINT "Wohin denn

```

```

(PB)
(EN)
(DB)
(OI)
(DN)
(NP)
(GF)
(FA)
(EH)
(KH)
(CD)
(FL)
(CDP)
(CM)
(CBP)
(COE)
(CPO)
(EL)
(FD)
(CD)
(EF)
(CDB)
(PH)
(GC)
(PI)
(BB)
(OC)
(AH)

```

```

?"
3460 IF o=2 AND (ra=obra(o) OR ob-
ra(o)=1) THEN PRINT "Zu wertvoll
."
3470 IF (o=4 OR o=6 OR o=7) AND (r-
a=obra(o) OR obra(o)=1) THEN PRI-
NT "Viel zu wertvoll."
3480 IF o=9 AND (ra=obra(o) OR ob-
ra(o)=1) THEN PRINT "Dazu braeuc-
hte ich eine Behälter."
3490 IF o=14 AND ra=obra(o) AND obra-
(o)=1 THEN PRINT "Ich habe die F-
ledermaus gefangen. Aus i-
hrem Maul ist ein Edelstein ge-
fallen." :obra(6)=2
3500 IF o=14 AND (ra=obra(o) OR o-
bra(o)=1) AND ra<2 THEN PRINT "
Ich habe mich fast im Netz verf-
angen. So etwas mache ich nicht
nochmal."
3510 IF o=16 AND (ra=obra(o) OR o-
bra(o)=1) THEN PRINT "Klatsch! S-
ie sind alle zerbrochen. M-
usste das sein."
3520 IF (o=19 OR o=22 OR o=32) AND
D(ra=obra(o) OR obra(o)=1) THEN
PRINT "Igitt ist das grauslich."
3530 IF o=23 AND (ra=obra(o) OR o-
bra(o)=1) THEN PRINT "Er steckt
zu fest im Boden."
3540 IF o=26 AND (ra=obra(o) OR o-
bra(o)=1) THEN PRINT "Wohin denn
?"
3550 IF o=30 AND (ra=obra(o) OR o-
bra(o)=1) THEN PRINT "Dazu braeu-
chte ich mindestens einen g-
eeigneten Behälter."
3560 IF o=8 AND (ra=obra(o) OR ob-
ra(o)=1) THEN PRINT "Sie fliegt
weg."
3570 IF (o=3 OR o=10 OR o=12 OR
o=15 OR o=17 OR o=20 OR o=24 OR
o=25 OR o=29 OR o=31 OR o=34 O-
R o=35) AND (ra=obra(o) OR obra(o)
=1) THEN PRINT "Hab ich aber geb-
racht hat es nichts."
3580 IF (o=5 OR o=11 OR o=13 OR
o=18 OR o=21 OR o=27 OR o=28 OR
o=33 OR o=36 OR o=37) AND (ra=ob-
ra(o) OR obra(o)=1) THEN PRINT "D-
as kann ich nicht."
3590 FOR a=1 TO 5000:NEXT
3600 GOTO 720
3610 *****
3620 ** nimm **
3630 *****
3640 IF (o=5 OR o=8 OR o=9 OR o=
11 OR o=13 OR o=18 OR o=20 OR o=
21 OR o=23 OR o=27 OR o=28 OR
o=30 OR o=31 OR o=33 OR o=35 OR
o=36 OR o=37) AND ra=obra(o) THE-
N PRINT "Das geht auf keinen Fal-
l." :FOR a=1 TO 3000:NEXT:GOTO 7
20
3650 IF obra(o)=1 THEN PRINT "D-
as habe ich schon." :FOR a=1 TO
3000:NEXT:GOTO 720
3660 IF obra(o)=ra AND oanz+1<=
5 THEN PRINT "Okay":obra(o)=1:t-
obj$(oanz+1)=obj$(o):oanz=oanz+
1
3670 IF o=1 THEN bed1=1
3680 IF o=29 THEN PRINT "Ich hab-
e sie angezündet." :du(24,6)=32
3690 IF o=12 THEN du(4,5)=0
3700 FOR a=1 TO 2000:NEXT
3710 GOTO 720
3720 *****
3730 ** lege **
3740 *****
3750 IF obra(o)<>1 THEN PRINT "
Das habe ich nicht." :FOR a=1 TO
3000:NEXT:GOTO 720
3760 FOR a=1 TO oanz:oo$(a)=UPP-
ER$(tobj$(a)):oo2$(a)=LEFT$(oo$(
a),4):NEXT

```

```

(MA)
(CL)
(IL)
(FH)
(JB)
(GP)
(NG)
(KN)
(JM)
(NE)
(CDK)
(FA)
(COE)
(HE)
(BP)
(DM)
(DG)
(AI)
(ME)
(OA)
(CF)
(GO)
(MH)
(FN)
(KC)
(AF)
(DD)
(DO)
(ME)
(CG)
(NC)
(LK)
(LH)

```

```

3770 FOR a=1 TO oanz
3780 IF oo2$(a)=ding$ THEN PRINT
"Okay":tobj$(a)=obra(o):ra=o-
anz:oanz=1
3790 NEXT
3800 IF o=29 THEN du(24,6)=0
3810 IF o=1 THEN bed1=0
3820 IF o=12 AND ra=4 THEN du(4
,5)=17
3830 IF obra(4)=3 AND obra(6)=3
AND obra(7)=3 THEN PRINT "Die S-
tatue beginnt zu leben, sie sagt
: Gehe in das verfallene Haus
und sage das Loesungswort Thyri-
s." :FOR a=1 TO 4000:NEXT
3840 IF tobj$(1)=" THEN tobj$(1
)=tobj$(2):tobj$(2)=tobj$(3):to-
bj$(3)=tobj$(4):tobj$(4)=tobj$(
5):tobj$(5)="
3850 IF tobj$(2)=" THEN tobj$(2
)=tobj$(3):tobj$(3)=tobj$(4):to-
bj$(4)=tobj$(5):tobj$(5)="
3860 IF tobj$(3)=" THEN tobj$(3
)=tobj$(4):tobj$(4)=tobj$(5):to-
bj$(5)="
3870 IF tobj$(4)=" THEN tobj$(4
)=tobj$(5):tobj$(5)="
3880 FOR a=1 TO 1000:NEXT:GOTO
720
3890 *****
3900 **** benutze ****
3910 *****
3920 IF o=2 AND ra=31 AND (obra(
o)=ra OR obra(o)=1) THEN PRINT "
Ein Drache kommt angefliegen und
be- foerdert mich auf mein
Schiff." :ra=12
3930 IF o=3 AND ra=28 AND obra(
5)=28 AND obra(o)=ra OR obra(o)
=1 THEN PRINT "Die Truhe laesst
sich oeffnen. Ein Horn und ein
Diamant werden sichtbar." :obra
(2)=28:obra(7)=28
3940 IF o=12 AND ra=7 AND (obra(
o)=7 OR obra(o)=1) THEN ra=6
3950 IF o=34 AND ra=28 AND obra(
o)=ra OR obra(o)=1 THEN PRINT
" In 2m Tiefe habe ich eine Truh-
e gefunden." :obra(5)=28
3960 IF obra(o)<>ra AND obra(o)
<>1 THEN PRINT "Soetwas sehe ic-
h hier nicht."
3970 IF obra(o)=ra OR obra(o)<>
1 THEN PRINT "Ich habe alles Mo-
egliche getan, aber ich habe kei-
nen Erfolg erzielt."
3980 FOR a=1 TO 4000:NEXT a
3990 GOTO 720
4000 *****
4010 ** Loesung **
4020 *****
4030 IF ra=26 AND o=38 THEN PRI-
NT "Sehr gut, du hast die Aufgabe
perfekt ge-loest. Du hast die I-
nsel vom Fluch der schwarzen
Diamanten befreit. Als Lohn f-
uer deine Heldentat erhaelst du
diese Insel, die sich nun wied-
er in ihre Ur- gestalt als";
4040 IF ra=26 AND o=38 THEN PRI-
NT "wunderschoene Tropeninsel ve-
rwandeln wird...."
4050 IF ra=26 AND o=38 THEN PRI-
NT "Plotzlich verschwindet das
Bild und das Haus verwandelt sic-
h in eine Villa."
4060 IF ra=26 AND o=38 THEN PRI-
NT STRING$(40,255):CALL &B18:E-
ND
4070 PRINT "Ich habe es gesagt, a-
ber es hat so gut wie nichts
gebracht."
4080 FOR a=1 TO 4000:NEXT a:GOT
O 720

```

```

(ED)
(PC)
(BB)
(JM)
(BC)
(KC)
(CM)
(DJ)
(AB)
(CB)
(MB)
(CM)
(LI)
(NG)
(PN)
(NF)
(GN)
(MF)
(GM)
(DN)
(MA)
(GN)
(CCE)
(FK)
(CDC)
(CEM)
(AB)
(MD)
(JD)
(JI)
(NP)
(GO)
(COK)
(IK)
(DA)
(IJ)
(IM)

```

ENDE DES LISTINGS

3-D-PAC MAN

Dieses ist eine tolle 3D-Version des bekannten PACMANN-Spiels. Hier geht es darum, beim Durchlaufen des Labyrinths zwanzig verstreute Kisten innerhalb von 30 Minuten einzusammeln, wobei einem zwei Ungeheuer an den Kragen wollen. Wird man erwisch, verliert man eines seiner fünf Computerleben. Zu Beginn erscheinen die TOP-FIVE und das Spielbild; nach Tastendruck beginnt das Spiel. Der

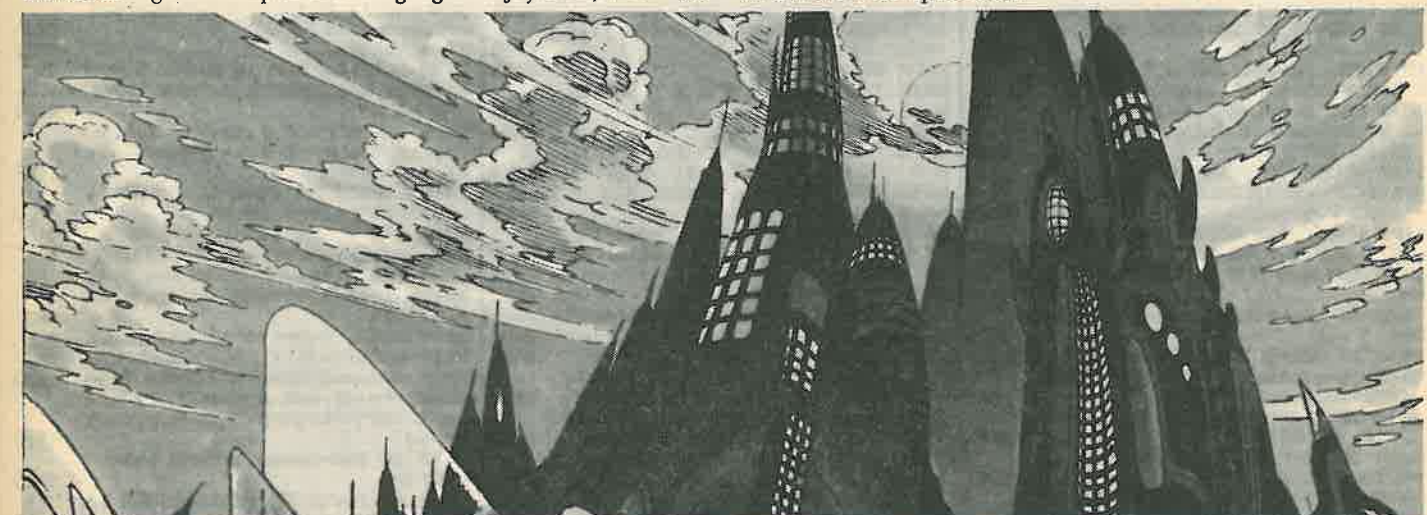
Screen zeigt Ihnen dreidimensional und mehrfarbig, was Sie aus Ihrer Perspektive im Labyrinth sehen, das Window zeigt Ihnen die eigene Position und die der Geister in einer Aufsicht auf das Labyrinth. Außerdem informiert Sie das Window über Ihre Restzeit, Ihre Computerleben und die bereits eingesammelten Kisten.

Die Steuerung erfolgt durch Betätigung des Joysticks, bzw. des

Cursors in die gewünschte Bewegungsrichtung. Durch das Drücken des Fireknopfes können die Kisten eingesammelt werden. Mit der Taste "L" Kann zusätzlich zu den Positionspunkten ein Labyrinthplan abgerufen werden, diese Hilfe ist danach jedoch für bestimmte Zeit gesperrt. Die Begegnung mit einem Geist bewirkt, daß PACMAN wider an den Anfang des Spiels versetzt wird und ein Computerleben we-

niger besitzt. Das Spiel ist beendet, wenn man entweder alle Leben verloren hat oder wenn alle Kisten eingesammelt wurden. Der Gewinner darf sich jedoch in die Hall of Fame eintragen.

Die Spiele sind hinsichtlich der Daten auf Floppy-Betrieb ausgelegt, die Änderungen für den Recorder werden aber keinem Schneider-Fan Kopfzerbrechen bereiten.



SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```

1000 *****
1010 --- PACMAN: THE 3D-VERSIO
N (c) 1986 by ALE
XANDER STROH ---
1020 *****
1030 V O R S P A N N
1040 MEMORY &3FFF:POKE &BDEE,&C
9:CALL &BC02
1050 FOR I=0 TO 3:INK 1,1:NEXT I
MODE 1:BORDER 1:PAPER 0:PEN 1:C
LS
1060 DEFINT A-Z:DIM f(32,20),la
$(6),n$(5),z$(5),v(2),w(2),c(14
),sort(5)
1070 ENV 1,20,-5,2:ENV 2,30,1,1
,1,0,20,50,-3,1:MEMORY &9FFF
1080 LOAD "sb.bin",&A000:CALL &A
000:OPENIN "hall.fil":FOR I=1 TO
5
1090 INPUT#9,n$(1):INPUT#9,z$(1
):NEXT:CLOSEIN:MEMORY &3FFF
1100 LOCATE 10,1:PRINT "PACMAN:
THE 3D-VERSION":PEN 2:LOCATE 19
,3
1110 PRINT CHR$(164)+" by":PEN

```

```

3:LOCATE 16,5:PRINT "SUPRA-SOFT"
1120 PEN 2:PRINT
1130 PRINT " This is the success-
ful game of PACMAN"
1140 PRINT " renewed.Find your w
ay in the horrible"
1150 PRINT " maze.Beware of the
ghosts.Collect the"
1160 PRINT " chests-twenty of th
em are scattered."
1170 PEN 3:PRINT
1180 PRINT " Try to finish it in
thirty minutes.Use"
1190 PRINT " JOYSTICK/CURSORS.Pr
ess FIRE/COPY to"
1200 PRINT " take a chest and pr
ess L for help:the"
1210 PRINT " maze will be displa
yed."
1220 PEN 2:PRINT
1230 PRINT " Good luck.Now selec
t control-keys and"
1240 PRINT " wait then while the
datas are read."
1250 PEN 1:PRINT:PRINT:PRINT:PR
INT " JOYSTICK (J) -- CURSOR
KEYS (C)"
1260 INK 1,18:INK 2,20:INK 3,6
1270 IF INKEY(45)>1 THEN ok=72
:uk=73:lk=74:rk=75:fk=76:GOTO 1
300
1280 IF INKEY(62)>1 THEN ok=0:

```

```

(KM) uk=2:lk=8:rk=1:fk=9:GOTO 1300 (GA)
(CF) 1290 GOTO 1270 (AP)
(ND) 1300 RANDOMIZE TIME:FOR I=1 TO
3000:NEXT: BORDER 0 (ML)
1310 FOR I=0 TO 15:INK 1,0:NEXT
:GOSUB 2780:MODE 0:GOSUB 2460:P
APER 5:CLS (CD)
(CME) 1320 ' 3 D - S C R E E N S (JK)
1330 IBANK,1 (EB)
(CA) 1340 x=1:y=1:GOSUB 2790:' Norma
ler Weg (ND)
(ED) 1350 x=11:y=1:GOSUB 2790:GOSUB
2850:' Sackgasse (PM)
(LM) 1360 x=1:y=9:GOSUB 2790:GOSUB 2
870:' Geradeaus/Links (JL)
(GP) 1370 x=11:y=9:GOSUB 2790:GOSUB
2910:' Geradeaus/Rechts (FP)
(HD) 1380 x=1:y=17:GOSUB 2790:GOSUB
2870:GOSUB 2910:GOSUB 2950:' Re
chts/Links (MM)
(AB) 1390 x=11:y=17:GOSUB 2790:GOSUB
2870:GOSUB 2910:' Kreuzung (IP)
(HC) 1400 IBANK,3 (HF)
(LA) 1410 ' B I L D S C H I R M U
N D I N I T I A L I S I E R U
N G (LN)
(GN) 1420 FOR I=1 TO 20 (CP)
(JB) 1430 k=RND*29+2:j=RND*17+2 (EC)
1440 IF f(k,j)=1 THEN f(k,j)=2
ELSE 1430 (MA)
(IP) 1450 NEXT (BE)
1460 PAPER 5:PEN 13:LOCATE 6,3:

```



```

PRINT"THE SCREEN":ISTRETCH,4,3,
10
1470 LOCATE 2,15:PRINT" * * * WI
NDOW * * * ":ISTRETCH,2,15,10
1480 PLOT 0,0,14:DRAW 0,376:DR
W 638,376:DRAW 638,0:DRAW 0,0
1490 MOVE 0,182:DRAW 638,182:PE
N 7:LOCATE 6,1:PRINT"SUPRA-SOFT
"
1500 LOCATE 2,18:PRINT"POSITION
":LOCATE 12,18:PRINT"CONTROLS"
1510 x=1:y=1:v(1)=17:w(1)=12:v
(2)=23:w(2)=4:be=1:hr=1:ti=1:u=1
:met=0
1520 s=0:ds=0:m=0:dm=3:chest=0:
liv=5:hiz=0:ik=0:FOR 1=0 TO 14:
c(1)=32:NEXT
1530 liut=1:tiut=1:fiut=1
1540 IQUAD,2,9,20,24,3:IQUAD,12
,19,20,24,3
1550 PLOT 116,60,1:PLOT 164,48,
6:PLOT 212,80
1560 PAPER 8:PEN 6:LOCATE 12,20
:PRINT"Ti":LOCATE 12,22:PRINT"L
i"
1570 LOCATE 12,24:PRINT"Ci"
PEN 1:LOCATE 15,20:PRINT"30:00"
1580 LOCATE 19,22:PRINT"5":LOCA
TE 19,24:PRINT"0"
1590 CALL &8000:INK 0,1:INK 1,2
4:INK 2,20:INK 3,6:INK 4,26
1600 INK 6,2:INK 7,8:INK 8,10:I
NK 9,12:INK 10,14:INK 11,16:INK
12,13:INK 13,18
1610 INK 14,26:INK 15,3:WHILE I
NKEY<>"":WEND:CALL &8B18:SOUND
1,50,30,6
1620 EVERY 6,3 GOSUB 2170:EVERY
50,2 GOSUB 2200
1630 ' H A U P T P R O G R A M
M
1640 IF liut*fiut*tiut=0 THEN 2
580
1650 IF met=1 THEN GOSUB 2090
1660 x1=x:y1=y:IF INKEY="" THEN
1910
1670 IF INKEY(36)>-1 THEN GOSUB
2260
1680 IF INKEY(fk)>-1 AND f(x,y)
=2 THEN 2130
1690 IF INKEY(ok)>-1 THEN be=1:
ik=1 ELSE IF INKEY(uk)>-1 THEN
be=2:ik=1
1700 IF INKEY(1k)>-1 THEN be=3:
ik=1 ELSE IF INKEY(rk)>-1 THEN
be=4:ik=1
1710 IF ik=1 THEN ik=0:GOTO 172
0 ELSE 1910
1720 IF be=2 THEN hr=hr+2 ELSE
IF be=3 THEN hr=hr-1 ELSE IF be
=4 THEN hr=hr+1
1730 IF hr=0 THEN hr=4
1740 hr=1+(hr-1)MOD 4
1750 ON hr GOTO 1760,1780,1800,
1820
1760 IF f(x,y-1)<>0 THEN y=y-1
ELSE y=y1:GOSUB 2040:GOTO 1640
1770 te$=STR$(SGN(f(x-1,y)))+ST
R$(SGN(f(x,y-1)))+STR$(SGN(f(x+
1,y))):GOTO 1840
1780 IF f(x+1,y)<>0 THEN x=x+1
ELSE x=x1:GOSUB 2040:GOTO 1640
1790 te$=STR$(SGN(f(x,y-1)))+ST
R$(SGN(f(x+1,y)))+STR$(SGN(f(x,
y+1))):GOTO 1840
1800 IF f(x,y+1)<>0 THEN y=y+1
ELSE y=y1:GOSUB 2040:GOTO 1640
1810 te$=STR$(SGN(f(x+1,y)))+ST
R$(SGN(f(x,y+1)))+STR$(SGN(f(x-
1,y))):GOTO 1840
1820 IF f(x-1,y)<>0 THEN x=x-1
ELSE x=x1:GOSUB 2040:GOTO 1640
1830 te$=STR$(SGN(f(x,y+1)))+ST
R$(SGN(f(x-1,y)))+STR$(SGN(f(x,
y-1))):GOTO 1840
1840 z=0:FOR 1=1 TO 6:IF te$=1a

```

```

(EP) 1,1)+RIGHT$(STR$(s),1)
(BA) 2340 IQUAD,1,20,1,25,240:FOR 1=
(FB) 0 TO 3:INK 1,0:NEXT:PAPER 0:PEN
1:LOCATE 6,6
(PM) 2350 PRINT"EXCELLENT!":PEN 3:LO
CATE 1,9:PRINT"YOU ARE IN THE H
ALL!":PEN 1
(CB) 2360 LOCATE 6,12:PRINT"ENTER NA
ME":PEN 2:LOCATE 4,17:PRINT"---
"
(JJ) 2370 ISTRETCH,6,6,10:ISTRETCH,1
(DB) 9,20:ISTRETCH,6,12,10:ISTRETCH
(HB) 4,17,14
(MI) 2380 WHILE INKEY<>"":WEND:INK
1,18:INK 2,20:INK 3,6
(OI) 2390 k=0:le=13:0=4:p=15:GOSUB 2
510
(JE) 2400 FOR 1=0 TO 13:n$(5)=n$(5)+
CHR$(c(1)):c(1)=32:NEXT
(JK) 2410 FOR 1=1 TO 5:sort(1)=60+VA
L(LEFT$(z$(1),2))+VAL(RIGHT$(z$
(1),2)):NEXT
(JN) 2420 FOR 1=5 TO 2 STEP-1:IF sor
t(1)>sort(1-1) THEN GOSUB 2440
(FA) 2430 NEXT:GOTO 2460
(LD) 2440 IEX,3,0n$(1-1),0n$(1):IEX,
3,0z$(1-1),0z$(1):IEX,2,0sort(1
-1),0sort(1)
(FK) 2450 RETURN
(AE) 2460 CLS:FOR 1=0 TO 3:INK 1,0:N
EXT:PEN 3:LOCATE 5,1:PRINT"HALL
OF FAME"
(AF) 2470 ISTRETCH,5,1,12:PEN 2:FOR
1=1 TO 5:LOCATE 1,1*4:PRINT n$(
1)
(BM) 2480 LOCATE 16,1*4:PRINT z$(1):
ISTRETCH,1,1*4,20:NEXT
(ML) 2490 PEN 1:LOCATE 4,23:PRINT"PR
ESS ANY KEY":ISTRETCH,4,23,14
2500 INK 1,18:INK 2,20:INK 3,6:
WHILE INKEY<>"":WEND:CALL &8B1
8:RETURN
(IN) 2510 WHILE in$="" :in$=INKEY$:WE
ND:in$=ASC(in$)
(HB) 2520 IF in$<>CHR$(127) THEN 2540
2530 k=k-(1 AND k>0):LOCATE 0+k
,p:PRINT" ":ISTRETCH,0+k,p,1:c(
k)=32
(JL) 2540 IF in$=CHR$(13) THEN in$=""
:RETURN
(CE) 2550 IF in$<32 OR in$>122 THEN in
$="" :GOTO 2510
(OK) 2560 LOCATE 0+k,p:IF k=le THEN
PRINT CHR$(in$):ELSE PRINT" "
2570 ISTRETCH,0+k,p,1:c(k)=in$
:IF (1 AND k=le):in$="" :GOTO 25
10
(LF) 2580 ' G A M E O V E R
(JF) 2590 FOR 1=1 TO 4000:NEXT:r=REM
AIN(2):r=REMAIN(3)
(AJ) 2600 IF liut*tiut=0 THEN 2610 E
LSE 2620
(JD) 2610 beg1$="OH,SUCH A SHAME!" :b
eg2$="LOOSER":GOTO 2630
(PC) 2620 GOSUB 2310:beg1$="CONGRATU
LATIONS!":beg2$="WINNER"
(MJ) 2630 IQUAD,1,20,1,25,240:FOR 1=
0 TO 3:INK 1,0:NEXT:PAPER 0:PEN
1
(JC) 2640 LOCATE 3,9:PRINT beg1$:IST
RETCH,3,9,16
(BM) 2650 LOCATE 2,12:PEN 3:PRINT"YO
U ARE THE " :beg2$:ISTRETCH,2,12
,18
(DH) 2660 LOCATE 5,15:PEN 2:PRINT"AG
AIN? (Y/N)":ISTRETCH,5,15,12
(AD) 2670 INK 2,2:INK 3,6:INK 1,18
(AP) 2680 IF INKEY(43)>-1 THEN 2690
ELSE IF INKEY(46)>-1 THEN 2720
ELSE 2680
(PF) 2690 FOR 1=1 TO 32:FOR k=1 TO 2
0:IF f(1,k)=2 THEN f(1,k)=1
(JH) 2700 NEXT k,1:PAPER 5:CLS:FOR 1
=0 TO 15:INK 1,0:NEXT:GOTO 1410
(NC) 2710 ' G O O D B Y E
(LC) 2720 CLS:INK 1,0:PEN 1:LOCATE 7

```

```

,12:PRINT"GOOD BYE":ISTRETCH,7,
12,8:INK 1,18
2730 r$="hall.fil":IERA,0r$:MEM
ORY &8FFF:OPENOUT"hall.fil":FOR
1=1 TO 5
2740 PRINT#9,n$(1):PRINT#9,z$(1
):NEXT:CLOSEOUT:FOR 1=1 TO 3000
:NEXT
2750 IHELP:NEW:END
2760 ' Z E I C H E N R O U T I
N E N
2770
2780 ' G R U N D R O U T I N E
2790 IQUAD,x,x+9,y,y+7,15
2800 b=0:FOR 1=y+4 TO y+7:IQUAD
,x+4-b,x+6+b,1,1,51:b=b+1:NEXT
2810 IQUAD,x+4,x+5,y,y+3,48:b=0
:PAPER 10:PEN 12
2820 FOR 1=y+4 TO y+7:LOCATE x+
3-b,1:PRINT CHR$(214):
2830 LOCATE x+6+b,1:PRINT CHR$(
215):b=b+1:NEXT:RETURN
2840 ' S A C K B A S E
2850 IQUAD,x+4,x+5,y,y+3,0:RETU
RN
2860 ' L I N K S
2870 IQUAD,x+1,x+2,y,y+4,0
2880 PEN 12:LOCATE x+1,y+5:PRIN
T CHR$(143)CHR$(143)
2890 LOCATE x+1,y+6:PRINT CHR$(
143):RETURN
2900 ' R E C H T S
2910 IQUAD,x+7,x+8,y,y+4,0
2920 PEN 12:LOCATE x+7,y+5:PRIN
T CHR$(143)CHR$(143)
2930 LOCATE x+8,y+6:PRINT CHR$(
143):RETURN
2940 ' N U R L I N K S U N
D R E C H T S
2950 IQUAD,x+1,x+8,y,y+4,0:RETU
RN
2960 ' D A T A R O U T I N E N
2970
2980 ' M C F U E R S E L E
C T
2990 RESTORE 3030:basis=&8000:a
dresse=0
3000 READ byte$:IF byte$="" THEN
N 3110
3010 byte$=VAL("&"+byte$)
3020 POKE basis+adresse,byte$:ad
resse=adresse+1:GOTO 3000
3030 DATA 21,00,40,22,05,90,cd,
00,90,c9,0,0,0,0,0,0
3040 DATA 21,28,40,22,05,90,cd,
00,90,c9,0,0,0,0,0,0
3050 DATA 21,00,42,22,05,90,cd,
00,90,c9,0,0,0,0,0,0
3060 DATA 21,00,42,22,05,90,cd,
00,90,c9,0,0,0,0,0,0
3070 DATA 21,00,45,22,05,90,cd,
00,90,c9,0,0,0,0,0,0
3080 DATA 21,28,45,22,05,90,cd,
00,90,c9,0,0,0,0,0,0
3090 DATA *
3100 ' M C F U E R 3 D - S
C R E E N S
3110 RESTORE 3150:basis=&9000:a
dresse=0
3120 READ byte$:IF byte$="" THEN
N 3200
3130 byte$=VAL("&"+byte$)
3140 POKE basis+adresse,byte$:ad
resse=adresse+1:GOTO 3120
3150 DATA 21,a4,c1,dd,21,00,40,
3e,08,f5,11,28,00,0e,08,06,28
3160 DATA dd,7e,00,77,23,dd,23,
05,c2,11,90,19,dd,19,0d,c2,0f,9
0
3170 DATA 11,00,05,19,dd,19,f1,
3d,c2,09,90,c9
3180 DATA *
3190 ' S P R I T E - D A T A S
3200 RESTORE 3240:basis=&9800:a
dresse=0
3210 READ byte$:IF byte$=999 THEN

```

```

3420
(CP) 3220 POKE basis+adresse,byte:ad
resse=adresse+1:GOTO 3210
3230 ' S P R I T E 1
(CP) 3240 DATA 51,51,51,51,51,51,51,
51,51,51,51,51,51,51,51,51,51,5
1,204,204,51
(CM) 3250 DATA 102,204,204,51,204,20
4,204,51,207,207,207
(CK) 3260 DATA 51,51,51,51,51,51,51,51,
51,51,51,51,51,51,51,204,
204,153
(CML) 3270 DATA 204,204,204,187,204,2
04,221,187,207,207,255,187
(CAB) 3280 DATA 51,207,207,207,51,207
,207,207,51,207,207,207,51,207,
207,207,51
(CFA) 3290 DATA 207,207,207,51,207,20
7,207,51,207,207,207,51,207,207
,207
(CMK) 3300 DATA 207,207,255,187,207,2
07,255,187,207,207,255,187,207,
207,255,187
(CAB) 3310 DATA 207,207,255,187,207,2
07,255,187,207,207,255,187,207,
207,255,187
(CNH) 3320 ' S P R I T E 2
(CKL) 3330 DATA 51,51,51,51,51,60,60,
60,51,60,60,60,54,56,48,60,54,5
6,176,60,60
(CBJ) 3340 DATA 56,176,108,60,56,48,1
08,60,60,60,108
(CBJ) 3350 DATA 51,51,51,51,60,60,60,
51,60,60,60,51,60,48,52,57,60,1
12,52,57
(CMF) 3360 DATA 156,112,52,60,156,48,
52,60,156,60,60,60
(CNE) 3370 DATA 60,60,60,60,60,120,20
8,208,60,240,240,240,60,104,224
,224,54,60
(CGO) 3380 DATA 60,60,51,60,60,60,51,
114,177,51,51,240,177,51
(CIK) 3390 DATA 60,60,60,60,208,208,2
08,60,240,240,240,60,224,224,18
0,60,60,60,60
(CGN) 3400 DATA 57,60,60,60,51,51,114
,177,51,51,114,240,51
(CBN) 3410 DATA 999
(CNK) 3420 ' L A B Y R I N T H - D A
T A S
(AH) 3430 RESTORE 3440:FOR L=2 TO 19
:FOR M=2 TO 31:READ F(M,L):NEXT
M:NEXT L
(CKI) 3440 DATA 1,1,1,1,1,1,1,1,0,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1
(CAC) 3450 DATA 0,1,0,1,0,1,0,1,0,0,0
,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
,0,0,1,0
(CNF) 3460 DATA 0,1,0,0,0,1,1,1,1,1,0
,1,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1
(CPD) 3470 DATA 0,1,0,1,0,1,0,0,0,0,0
,1,0,1,0,0,1,0,0,0,0,1,0,0,0,0,0
,0,1,0,0
(CBE) 3480 DATA 1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,0
(CNB) 3490 DATA 0,1,0,0,0,0,0,0,0,0,1
,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,1,0,0
(CDA) 3500 DATA 0,1,0,0,0,1,1,1,1,1,0
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,0
(CAK) 3510 DATA 0,0,0,0,0,1,0,1,0,1,0
,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,1,1,1,1
(CFC) 3520 DATA 0,1,0,0,0,0,0,0,0,0,1
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,1,0,0,0
(CMG) 3530 DATA 0,1,0,0,0,1,1,1,1,1,0
,1,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1
,1,1,1,1
(CEO) 3540 DATA 0,0,0,0,0,1,0,1,0,1,0
,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,1,0,0,0
(CPF) 3550 DATA 1,1,1,1,1,1,0,0,0,1,1,1
,1,1,1,1,1,1,1,1,0,1,0,1,0,1,0,1
,0,0,0,0
(CBH) 3560 DATA 0,1,0,0,0,0,1,0,1,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0
(CBI) 3570 DATA 0,1,0,0,0,0,1,0,1,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,1,1,1
(CDA) 3580 DATA 0,1,0,1,1,1,0,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1
(CPF) 3590 DATA 0,1,0,1,1,1,0,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1
(CME) 3600 DATA 0,1,0,1,1,1,0,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1

```

```

(MC) 3550 DATA 0,0,0,0,0,1,0,1,0,1,0
,1,0,0,0,1,0,1,0,0,1,0,1,0,0,0,0
,0,1,1,1
(BN) 3560 DATA 1,1,1,1,1,1,0,1,0,1,0
,1,0,1,0,1,1,1,1,0,1,0,1,0,1,0,0
,1,0,1,0
(CN) 3570 DATA 0,1,0,1,0,1,1,1,0,0,0
,1,0,1,0,1,0,0,0,0,1,0,1,1,1,0,0
,1,1,1,1
(CBC) 3580 DATA 0,1,0,1,0,1,0,1,0,0,1
,1,1,1,0,1,1,1,1,1,1,0,1,0,1,0,1
,0,0,0,0
(CGB) 3590 DATA 0,0,0,1,0,1,0,1,1,0,0
,0,0,1,0,1,0,0,0,0,1,0,1,0,1,0,1
,0,1,0,1
(CLE) 3600 DATA 0,1,0,0,0,1,0,1,0,0,1
,0,1,0,0,0,1,0,1,1,1,1,1,0,0,1,1
,1,1,1,0
(CFC) 3610 DATA 1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,0,0,0,0,0,0,0,0,0,1,0,0
,1,0,1,0
(CNC) 3620 ' E C K - C H E C K - D A
T A S
(CDJ) 3630 RESTORE 3640:FOR 1=1 TO 6:
READ 1a$(1):NEXT:RETURN
(CGP) 3640 DATA " 0 1 0", " 0 0 0", " 1
1 0", " 0 1 1", " 1 0 1", " 1 1 1
"
(CBN) 3650
(CDF) 3660

```

ENDE DES LISTINGS

Teil 2

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```

1000 '-----
1010 '--- (c) September 1986
by
1020 '--- Alexander Stroh
1030 '-----
1040 MEMORY &9FFF:CALL &8C02:DE
FINIT a=2
1050 RESTORE 1080
1060 FOR 1=&A000 TO &A4D7:READ
a$
1070 a=VAL("&"+a$):POKE 1,a:NEX
T
1080 DATA 01,0f,a0,21,00,a1,cd,
d1,bc,3e
1090 DATA c9,32,00,a0,c9,72,a0,
c3,04,a1
1100 DATA c3,51,a1,c3,63,a1,c3,
74,a1,c3
1110 DATA 85,a1,c3,b0,a1,c3,be,
a1,c3,d7
1120 DATA a1,c3,e7,a1,c3,f7,a1,
c3,1d,a2
1130 DATA c3,58,a2,c3,72,a2,c3,
a8,a2,c3
1140 DATA 17,a3,c3,37,a3,c3,4f,
a3,c3,73
1150 DATA a3,c3,9d,a3,c3,c0,a3,
c3,ea,a3
1160 DATA c3,14,a4,c3,30,a4,c3,
4c,a4,c3
1170 DATA 74,a4,c3,82,a4,c3,91,
a4,c3,9a
1180 DATA a4,c3,a9,a4,c3,bc,a4,
c3,c5,a4
1190 DATA c3,ce,a4,00
1200 DATA 53,50,52,49,54,c5,43,
52,55,4e
1210 DATA 43,c8,43,4f,50,d9,44,
49,53,50
1220 DATA 4c,41,d9,48,45,4c,d0,
46,41,53
1230 DATA d4,42,41,4e,cb,42,49,
47,4f,ce
1240 DATA 42,49,47,4f,46,c6,52,
54,55,52

```


1250 DATA ce,4c,54,55,52,ce,51,
55,41,c4
1260 DATA 42,59,c5,53,54,52,45,
54,43,c8
1270 DATA 46,49,45,4c,44,5d,08,
53,59,c4
1280 DATA 50,49,43,c4,53,5d,9,
4c,46,49
1290 DATA ce,4c,46,4f,55,44,57,
4f,57,ce
1300 DATA 57,55,d0,48,49,44,c5,
53,48,4f
1310 DATA d7,54,52,41,d0,49,4e,
d6,44,4f
1320 DATA 4b,c5,4e,c9,50,45,d9,
43,41,4c
1330 DATA cd,4e,4f,49,53,c5,0,0,
0,0,0,0
1340 DATA 00,00,00,00,fe,03,c2,
00,a5,eb
1350 DATA 06,07,29,10,fd,eb,21,
00,98,19
1360 DATA eb,dd,66,04,dd,6e,02,
d1,3e,02
1370 DATA 3a,a1,cd,1a,bc,0e,08,
06,03,1a
1380 DATA 77,13,23,10,fa,1a,77,
13,d5,11
1390 DATA fd,07,19,d1,0d,20,ec,
c9,e5,cd
1400 DATA 20,a1,e1,24,e5,cd,20,
a1,e1,2c
1410 DATA 25,e5,cd,20,a1,e1,24,
cd,20,a1
1420 DATA c9,fe,00,c2,00,a5,21,
00,ac,af
1430 DATA be,28,03,36,00,c9,36,
01,c9,fe
1440 DATA 00,c2,00,a5,21,00,c0,
11,00,40
1450 DATA 01,00,40,ed,b0,c9,fe,
00,c2,00
1460 DATA a5,21,00,40,11,00,c0,
01,00,40
1470 DATA ed,b0,c9,fe,00,c2,00,
a5,3e,01
1480 DATA cd,0e,bc,cd,02,bc,af,
cd,96,bb
1490 DATA 3e,01,cd,90,bb,cd,14,
bc,cd,00
1500 DATA bb,cd,03,bb,cd,b6,bc,
cd,45,bb
1510 DATA 3e,c3,32,ee,bd,c9,fe,
00,c2,00
1520 DATA a5,3e,0b,21,5f,00,cd,
68,bc,c9
1530 DATA fe,01,c2,00,a5,3e,01,
bb,28,04
1540 DATA 3e,c0,18,02,3e,40,47,
cd,06,bc
1550 DATA 78,32,cb,b1,c9,fe,00,
c2,00,a5
1560 DATA 3e,07,32,0b,a1,3e,3a,
32,1e,a1
1570 DATA c9,fe,00,c2,00,a5,3e,
05,32,0b
1580 DATA a1,3e,20,32,1e,a1,c9,
fe,00,ca
1590 DATA 00,a5,f5,cd,43,a2,11,
4f,c0,19
1600 DATA 06,08,c5,e5,d1,2b,01,
4f,00,1a
1610 DATA ed,b8,12,01,50,08,09,
c1,10,ee
1620 DATA f1,3d,20,e0,c9,fe,00,
ca,00,a5
1630 DATA f5,cd,43,a2,11,00,c0,
19,06,08
1640 DATA c5,e5,d1,23,01,4f,00,
1a,ed,b0
1650 DATA 12,01,b0,07,09,c1,10,
ee,f1,3d
1660 DATA 20,e0,c9,dd,6e,00,2d,
dd,23,dd
1670 DATA 23,af,67,06,04,29,10,
fd,54,5d

(OB)
(JD)
(ME)
(DH)
(DH)
(HL)
(KB)
(KP)
(IM)
(EA)
(NL)
(IL)
(FD)
(CK)
(KK)
(MH)
(IN)
(HJ)
(DB)
(KO)
(OI)
(GI)
(AC)
(KP)
(NJ)
(FO)
(LM)
(KB)
(LF)
(IF)
(KB)
(BJ)
(HP)
(KB)
(FC)
(FF)
(BB)
(IJ)
(PD)
(NB)
(HD)
(GC)
(HJ)

1680 DATA 29,29,19,c9,fe,05,c2,
00,a5,7b
1690 DATA dd,5e,02,dd,6e,04,dd,
56,06,dd
1700 DATA 66,08,1d,15,25,2d,cd,
44,bc,c9
1710 DATA fe,00,c2,00,a5,cd,8a,
a1,cd,88
1720 DATA a2,af,06,ff,21,00,a0,
77,23,10
1730 DATA fc,c9,21,9c,a2,06,0c,
7e,cd,5a
1740 DATA bb,23,10,f9,26,01,2e,
03,cd,75
1750 DATA bb,c9,20,42,41,53,49,
43,20,31
1760 DATA 2e,30,0d,0a,fe,03,c2,
00,a5,7b
1770 DATA 32,76,a5,dd,6e,02,dd,
66,04,25
1780 DATA 2d,e5,cd,1a,bc,11,00,
38,19,22
1790 DATA 72,a5,11,50,00,e5,19,
d1,3e,02
1800 DATA 32,74,a5,dd,21,74,a5,
0e,04,06
1810 DATA 04,1a,77,c5,01,00,08,
ed,42,77
1820 DATA ed,4a,c1,23,13,10,f0,
d5,11,04
1830 DATA 10,ed,52,d1,eb,c5,01,
00,08,ed
1840 DATA 42,c1,2b,2b,2b,2b,eb,
0d,20,d7
1850 DATA 2a,72,a5,11,00,20,ed,
52,eb,2a
1860 DATA 72,a5,dd,35,00,20,c4,
e1,24,dd
1870 DATA 35,02,20,a3,c9,fe,03,
c2,00,a5
1880 DATA cd,78,bb,e5,dd,6e,02,
dd,66,04
1890 DATA cd,75,bb,cd,60,bb,30,
02,eb,77
1900 DATA e1,26,00,cd,75,bb,c9,
fe,03,c2
1910 DATA 00,a5,dd,66,03,dd,6e,
02,dd,46
1920 DATA 04,4e,1a,77,79,12,13,
23,10,f7
1930 DATA c9,57,3e,20,ba,fa,00,
a5,af,67
1940 DATA 6b,06,05,29,10,fd,42,
11,1f,90
1950 DATA 19,05,00,dd,7e,02,77,
2b,dd,23
1960 DATA dd,23,10,f5,af,77,c9,
fe,04,c2
1970 DATA 00,a5,dd,66,06,dd,6e,
04,cd,75
1980 DATA bb,dd,7e,02,cd,90,bb,
af,67,6b
1990 DATA 06,05,29,10,fd,11,00,
90,19,06
2000 DATA 20,7e,cd,5a,bb,23,10,
f9,c9,fe
2010 DATA 00,ca,00,a5,f5,af,67,
dd,6e,00
2020 DATA 06,05,29,10,fd,11,00,
90,19,06
2030 DATA 20,77,23,10,fc,dd,23,
dd,23,f1
2040 DATA 3d,20,e3,c9,fe,02,f2,
00,a5,fe
2050 DATA 01,20,14,3e,10,12,eb,
23,5e,23
2060 DATA 56,21,07,b8,06,10,7e,
12,23,13
2070 DATA 10,fa,c9,21,07,b8,06,
10,7e,cd
2080 DATA 5a,bb,23,10,f9,c9,fe,
02,f2,00
2090 DATA a5,fe,01,20,14,3e,10,
12,eb,23
2100 DATA 5e,23,56,21,4c,b8,06,
10,7e,12

(MN)
(PG)
(OK)
(GI)
(NK)
(HE)
(IN)
(GK)
(LF)
(KH)
(JF)
(AK)
(JM)
(GI)
(PI)
(KP)
(PF)
(DM)
(KF)
(KP)
(KI)
(AO)
(DM)
(IB)
(BL)
(DF)
(NB)
(BF)
(HA)
(CD)
(FA)
(NJ)
(FH)
(AB)
(HK)
(NB)
(BN)
(HI)
(ME)
(GH)
(LM)
(FK)
(BN)

2110 DATA 23,13,10,fa,c9,21,4c,
b8,06,10
2120 DATA 7e,cd,5a,bb,23,10,f9,
c9,fe,05
2130 DATA c2,00,a5,06,00,7b,dd,
5e,02,dd
2140 DATA 6e,04,dd,56,06,dd,66,
08,1d,15
2150 DATA 25,2d,cd,50,bc,c9,fe,
05,c2,00
2160 DATA a5,06,01,7b,dd,5e,02,
dd,6e,04
2170 DATA dd,56,06,dd,66,08,1d,
15,25,2d
2180 DATA cd,50,bc,c9,fe,00,c2,
00,a5,21
2190 DATA 81,ae,11,83,ae,1a,77,
13,23,1a
2200 DATA 77,ed,5b,81,ae,2a,7b,
ae,ed,52
2210 DATA eb,2a,7b,ae,af,77,42,
4b,e5,d1
2220 DATA 1b,ed,b8,c9,fe,00,c2,
00,a5,21
2230 DATA 81,ae,36,6f,23,36,01,
c9,fe,01
2240 DATA c2,00,a5,3e,16,cd,5a,
bb,7b,cd
2250 DATA 5a,bb,c9,fe,00,c2,00,
a5,cd,9c
2260 DATA bb,c9,fe,02,c2,00,a5,
dd,66,03
2270 DATA dd,6e,02,73,23,72,c9,
fe,01,c2
2280 DATA 00,a5,7b,07,47,21,40,
fe,2b,7c
2290 DATA b5,20,fb,10,f6,c9,fe,
00,c2,00
2300 DATA a5,cd,18,bb,c9,fe,00,
c2,00,a5
2310 DATA cd,b6,bc,c9,fe,00,c2,
00,a5,cd
2320 DATA b9,bc,c9,00
2330 RESTORE 2360
2340 FOR i=&A500 TO &A51E:READ
a\$
2350 a=VAL("&"+a\$):POKE 1,a:NEX
T
2360 DATA 21,0d,a5,7e,fe,00,c8,
cd,5a,bb
2370 DATA 23,18,f6,4f,70,65,72,
61,6e,64
2380 DATA 20,6d,69,73,74,61,6b,
65,0a,0d
2390 DATA 00
2400
2410 'Direkt hinter dem
2420 'Hauptprogramm savens
2430
2440 SAVE"sb.bin",b,&A000,&520

ENDE DES LISTINGS

Teil 3

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

10 ***** TOP FIVE *****
11 * Erstellungsprogramm *
12 DIM n\$(5),z\$(5)
13 FOR i=1 TO 5
14 n\$(i)="SUPRA-SOFT"
15 z\$(i)="00:00:1NEXT
16 'Hinter "sb.bin" savens
17 OPENOUT"halli.fil"
18 FOR i=1 TO 5
19 PRINT#9,n\$(i);PRINT#9,z\$(i)
20 NEXT i:CLOSEOUT
ENDE DES LISTINGS

(GA)
(BB)
(DC)
(DB)
(GP)
(DM)
(LB)
(MF)
(NL)
(CD)
(GO)
(KJ)
(OK)
(CF)
(JG)
(KB)
(EP)
(LP)
(LN)
(EP)
(BL)
(KE)
(DM)
(EH)
(BJ)
(FB)
(JN)
(OM)
(BB)
(CC)
(EB)
(DN)
(DD)
(GJ)

* Kleinanzeigen * Kleinanzeigen * Kleinanzeigen *

BIETE SOFTWARE

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

C(1)-16-SUPER-
TEXTVERARBEITUNG I
Auf Kassette oder Diskette gegen 10,-
DM bei MP-SOFT. Bestellungen und
Info bei: M. Klarner, Markgrafenstr. 33,
4600 Dortmund 1, Tel.: 0231/127668

C-64: 50 Nützliche Hilfsprogram-
me, voll listbar z. B. Tools, Stat-
istik ... auf Disk oder Kass. nur
10,- DM (III). Kein Versand ins
Ausland. Nur Vorkasse, keine
Marken! COMPUTER-TECHNIK
Andreas Mlynek, Postfach 22 22,
2990 Papenburg 2

C-16/Plus 4 Verkäufe 17 Originalspiel-
kassetten z.B. Bomb Jack, Berks...!
Außerdem Winter Olympics von Tyne-
soft (Disk) für 20 DM. Info von:
Frank Roulunds, Benraderstr. 135,
4154 Tönisvorst 1

Biete C16 Software
Info gegen Rückporto bei:
Markus Perschon, Birkenhöhe 3,
2057 Wentorf bei Hamburg
(Original Spiele)

Hallo C-16 USER
Verkaufen Games z.B. Dallas, Airwolf,
Poker, usw. über 50 Games zur Aus-
wahl. 15 davon 14,- DM!
Ruft an bei Karsten (02981/2990)
oder bei Torben (02981/2267)
P.S. suchen Kriegsspiele und brauchen
einen Tauschpartner!
Beilgt Euch, wir warten!

Achtung C-16/116/Plus 4-USER!
Verkaufe auf Diskette oder Kassette 24
Spiele und 9 Anwenderprogramme für
20,- DM. Schickt einen 20-Mark-Schein
an Michael Grahneis, Stefanstr. 1,
5551 Monzelfeld und Ihr erhaltet sofort
Eure Software.
Tel.: 06531/3949.
P.S. Auf Kassette sind 3 Programme
mehr drauf!

C-16C-16***C-16***C-16***
Verkaufe Torten- und Balkendiagramm.
Super-Grafik. Die Programme sind in
Turbotape abgespeichert und kosten
nur 15,- DM (Alles auf Kassette und Por-
to frei). Carsten Warnke, Engestr. 14,
2256 Garding (nur gegen Vorkasse)!

C-64: 60 Spielprogramme über-
arbeitete Version für nur noch 10,-
DM (III) - Disk od. Kass. -Sicherer
Disk-Versand in Hartbox! Kein
Versand ins Ausland! Nur Vor-
kasse, keine Marken!
COMPUTER-TECHNIK
Andreas Mlynek, Postfach 22 22,
2990 Papenburg 2.

C-64: 50 Programme auf Disk
oder Kass. für nur 10,- DM! Alles
Basic, ohne Listschutz z. B.
Tools, Floppy-Ceter, Weltzeit,
Grafik, Sound ...
COMPUTER-TECHNIK
Andreas Mlynek, Postfach 22 22,
2990 Papenburg 2
*** VORKASSE! ***

C-64 Programme
C-64 nur auf Disk: z.B. Sanxion/New-
look, Exp. Fist, Speedking, Commando,
Boxing, 8 wirklich Top-Spiele 20,- DM!
Ines ten Hagen, Kückendellstr. 45,
4100 Duisburg 12

Spiele/Anwenderprogramme für
C-16/116/Plus 4/Atari 800 XL, 600 XL,
520 ST, CPC 464 und CPC 664 für nur
1,- DM pro Spiel. Liste gegen Rückporto
anfordern bei:
Thorsten Olbrich, Hauptstr. 36,
6402 Grossenluder 1

C16/116 *** Programme *** C16/116
aller Art (Spiele, Anwendungen, Grafi-
ken, ...) gibt es bei Oliver Dangel,
Riesbergstr. 37, 7157 Murrhardt
!!! Liste gegen -80 DM !!!

Cooler Software für alle Freaks!

Commodore 16/116/Plus 4

	Kassette	Kassette
ACE	32,00	Saboteur (+64 KB) 29,95
ACE (+64K)	34,95	Scooby-Doo 25,00
Ghost'n Goblins	29,00	Space Pilot 19,95
Hypersports	29,00	Sport Show 25,00
International Karate	19,95	The way of the expl. fist 29,95
ROM-Listing (Buch)	29,00	Winter Olympiade 29,00
RAMBO	29,00	Yie ar Kung-Fu 29,00

Amiga

	Diskette
Defender of the Crown	98,90
Hacker II	74,90
Hollywood Poker	79,00
Leader Board	74,90
Marble Madness	78,90
The Halley Projekt	69,90
The Pawn	74,90
World Games	69,90

Atari ST

	Diskette
Championship Wrestling	69,90
Gauntlet	69,90
Hollywood Poker	69,00
Karate Kid II	69,90
Renegade	38,90
Starglider	69,90
Super Cycle	69,00
World Games	69,90

Schneider CPC (464/664/6128)

	Kassette	Diskette
1942	32,95	44,95
Aliens	38,95	48,95
Avenger	32,95	44,95
Footballer of the year	32,95	44,95
Gauntlet	32,95	44,95
Ikari Warriors	32,95	44,95
Scooby-Doo	32,95	44,95
Space Harrier	32,95	44,95
Top Gun	32,95	44,95

Commodore 64/128

	Kassette	Diskette		Kassette	Diskette
1942	29,95	44,95	Ninja	14,95	-
City Cobra	32,95	-	Scooby-Doo	32,95	44,95
Flash Gordon	14,95	-	Super Cycle	32,95	44,95
Gauntlet	32,95	44,95	Trivial Pursuit (deutsch)	38,95	48,95
Destroyer	-	44,95	Vietnam	32,95	44,95
Hollywood Poker	-	49,00	World Games	32,95	44,95
Howard the duck	32,95	48,95	ZUB	14,95	-
Labyrinth	32,95	48,95			



SOFTWARE VERSAND

ANDREAS BACHLER
POSTFACH 429
D-4290 BOCHOLT

(02871) 18 30 88

Versand erfolgt gegen Nachnahme (+ 4,- DM Porto) oder
Vorkasse (Bar, Scheck)

Fordern Sie unter Angabe Ihres Computertyps kostenloses
Informationsmaterial über unser Lieferprogramm an!

Haben Sie Ihren
C-16/116 schon
auf 64 K aufgerüstet?

NEIN!

Dann nutzen Sie schnell
den Hardware-Service
des Tronic-Verlages

Unser Angebot: 64 K-Speichererweiterung

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

Bestell-Nr.: HWS-1-16 (C16) 45,- DM + NN

Bestell-Nr.: HWS-1-116 (C116) 45,- DM + NN

(Bei Vorkasse + 2,- DM Versand, Ausland nur gegen Vorkasse,
Bearbeitungsdauer bis zu 3 Wochen, Bestellungen bitte nur
separat vom Software-Service!)

Bitte Computertyp angeben!

Tronic-Verlag GmbH

Postfach 870 · 3440 Eschwege · Tel. (05651) 300 11

***** HILFE !!! *****

Suche C-16+64K Programme

Nur Diskette !!!

An: Hubert Mühlbacher, Plattenberg 3,

8221 Waging, Tel.: 08681/772

Computer-Fachzeitschriften gesucht.

Zahle 1,- DM pro Stück. In Köln und Um-

gebung hole ich ab.

Wilmes, Titusstr. 22, Köln 1.

Tel. 373198 (Abends bitte vor 20 Uhr)

COMPUTER SERVICE – DER C-16 SPEZIALIST

Postfach 1304 · 7913 Senden (Iller) · Telefon (07307) 6230

COMPUTER SERVICE – DER C-16 SPEZIALIST

SPIELPROGRAMME	K	D	ANWENDERPROGRAMME	K	D
Ace	27,90		Music Master	22,90	22,90
Ace 64K	29,90	34,90	Paint Box	19,90	19,90
Winter Olympiade	24,90	24,90	Micro Detail/Calc/Text	je 22,90	22,90
Hit Pack	27,90		Graphic Designer	15,90	15,90
Plus Paket	24,90	24,90	Turbo Tape	15,90	
Rätsel der 7 Kolonien	(di. graph. Adv.)	15,90	Turbo Plus Modul	34,90	
Legende im Eis	(di. graph. Adv.)	15,90	Utility/Adress/Vorkabel C-16	je 27,90	
Plaid im Dschungel	(di. graph. Adv.)	15,90			
Karate King (64K)	17,90				
Bridgehead	15,90				
The way of the exploding Fist	27,90				
Mercenary (64K)	27,90	39,00			
Sport Show	24,90				
Scoby Duo	20,90				
The Magician Curse	19,90				
Project Nova	24,90				
Saboteur	27,90				
Who dares wins II	19,90	29,00			
European Games	27,90	29,00			
Computer Hits (10 Programme)	27,90				
Quivi (64K)	23,90				
Trail Blazer	19,90				
Melbourne Classics (4 Programme)	27,90				

Kompletter Katalog mit Programmbeschreibungen gegen DM 0,80 in Briefmarken. Versand erfolgt gegen
Vorauszahlung oder Nachnahme. An Versandkosten fallen an: Bis 50,- DM Bestellwert: 2,50/5,- DM
Bis 300,- DM Bestellwert: 5,- /7,50 DM
Ab 300,- DM Bestellwert: 7,- /9,50 DM

***** Osterhasieschmünzelpreise *****

Freeze Frame MK IIIb,
Copymodul C64 115 DM
Freeze Frame
+ orig. Utility-Disc 140 DM
The Power Cartridge, Superpreis 125 DM

+ viele Markenprodukte für C16/116/128

Preise b. Vorkasse ohne Zuschläge f. Porto + Verpackung, Nachnahme + 4,50 DM

ASTRO-VERSAND · Postfach 1330 · 3502 Vellmar

24-Stunden-Bestelltelefon: (0561) 880111

Suche C-64 Zahle bis 150 DM, mit Zu-
behör auch mehr! Melden bei: Moser
Norbert, Uferstr. 17, 5280 Graunau-
Austria

.....
SUCHE:
Floppy 1551, muß O.K. sein.
Bitte melden bei:
Stefan Wuttke, Sonnenweg 51,
2122 Bleckede, Preis: VHS
.....

Suche für C-64 Megamania Angebote
an 0511/666619 oder Wolfgang Pauke,
Isenrahger Str. 74, 3000 Hannover 1.

Floppy 1541 zum günstigen Preis ge-
sucht. Melden bei Thorsten Kleinsorge,
Mittelstraße 16, 5788 Winterberg 3,
Tel. 02981/6631.

Suche Tauschpartner für Spiele und
Programme nur für Commodore VC 20.
Ingo Devoght, Dorfstr. 11, 5441 Nitz.

Suche Kopierprogr. für Plus 4 um
Progr. von Cass. auf Disk. zu kopieren.
Suche auch 64K Progr. Angebote an:
Norbert Göbbers, Kurt-Hubertstr. 3,
4040 Neuss 1.

Suche Floppy für den Plus/4. Biete da-
für bis zu 200,- DM. Angebote an:
Gerhard Lackner, Hölberg 25,
8151 Hiltendorf/Osterreich.
Tel. A-03137/27772.

C 116 : Suche Soft- und Hardware
Schriftliche Angebote, ggf. mit Spielbe-
schreibung bzw. Anwendermöglichkeit
an: Lothar Eckert, Schild 2,
3300 Braunschweig

S * SUCHE * * * SUCHE * S
U * Disk = SLAP SHOT = f. C64 * U
C * Rudolf Preiss * C
H * 8491 Chameray, Rossberg 12 * H
E * SUCHE * * * SUCHE * E

Der Hammer:
Orig. Final Cartridge II,
neueste Version, für jämmerliche 99 DM
Commodore/Schneider-Maus nur 90 DM
Alles-Copy-Adapter f. 2 Datensets 44 DM

+ viele Markenprodukte für C16/116/128

Preise b. Vorkasse ohne Zuschläge f. Porto + Verpackung, Nachnahme + 4,50 DM

ASTRO-VERSAND · Postfach 1330 · 3502 Vellmar

24-Stunden-Bestelltelefon: (0561) 880111

Hallo C 16/116/Plus 4 Besitzer
Suche Tauschpartner für den C 16 ohne
64K. Habe Top Games.
Stefan Kramer, Hauptstr. 30,
5788 Winterberg 1

Achtung: Suche Commodore 128
mit Floppy oder Datensette. Billig und gut
zuhaben. Adresse:
Denny Behle, Hachbergstr. 6,
7867 Wehr, Tel.: 07762/8450

BIETE
HARDWARE

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

Verkaufe C-16 + Datensette + 4 Bücher
+ 2 Joysticks + 5 Spiele, z.B. Winter-
Events + Turbo Tape für 200,- DM.
Tel.: 0209/611329 ab 19.00 UHR

Verkaufe C-116 mit Basic-Kurs + 25
Anwenderpr. + 50 Spiele z.B. Scramble,
Kikstart, Formula one ...
Preis nur 160,- DM ! Melden bei:
Matthias Rutsch, Tel.: 04152/72072

COMPUTER SERVICE – DER C-16 SPEZIALIST

Postfach 1304 · 7913 Senden (Iller) · Telefon (07307) 6230

COMPUTER SERVICE – DER C-16 SPEZIALIST

SPIELPROGRAMME	K	D	SPIELPROGRAMME	K	D
Classic Compilation I, II, III	je 27,90		Tazz	9,90	
Bandits at Zero	9,90		Via are Kung Fu	24,90	
Danger Zone	9,90		Jet Set Willy II	24,90	
Bongo Construction Set	15,90	15,90	Pinpoint	16,90	
Legionaire	15,90	15,90	Video Manias	24,90	
Monty on the Run	9,90		Iran Warriors	9,90	
Rescue on Zylon	9,90		1942	17,90	
Superhits (6 Programme)	24,90		Ghosts'n Goblins	24,90	
King Size Programmsammlung (50 Progr.)	24,90		Fighting Warrior	27,90	
Favourite 4 (4 Programme)	24,90		Video Poker	9,90	
Gunslinger	22,90		XCellar 8	9,90	
International Karate	16,90		Cave Fighter	19,90	
Jet Set Willy	24,90	29,-	Finders Keepers	9,90	
Jump Jet (mind. 32K)	27,90		Formula 1 Simulator	9,90	
Lazer Zone/Matrix	22,90		Speedking	9,90	
Prospector Pete	9,90		G-Man	9,90	
P. O. D.	9,90		Raider	9,90	
Kickstart	9,90		Daley Thompson Star Events	9,90	
Powerball	9,90		Space 2	9,90	
Trizones	9,90		Winni witch's superroom	9,90	
Hustler (Billard)	9,90		Atlantis	19,90	
Torpedorun	19,90		Canoe Slalom	13,90	
Saboteur	27,90		Diagon	10,90	

Programme die mit * gekennzeichnet sind, waren zur Zeit der Drucklegung (15.1.1987) noch nicht lieferbar.
Bestellungen sind schriftlich als auch telefonisch möglich. Außerhalb der Bürozeiten (10.00 bis 18.30) nimmt
unser Anrufbeantworter Ihre Wünsche entgegen. Für aufkommende Fragen stehen wir Ihnen gerne zur Verfügung.
COMPUTER SERVICE – DER C-16 SPEZIALIST

Kopiermodul Mirage-Imager für
Schneider CPC 464/664/6128 und
MAD-Game Knight-Tyme für 120 DM
abzugeben. Nachnahme. C. Reinke,
Otto-Dix-Str. 3, 1000 Berlin 21

.....

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 C16/64K mit Datensette und

ca. 701 Spielen (keine Abgetippten) für

500 DM. Michael Schimpf, Eikelohr

Str. 4, 4782 Erwitte,

Tel.: 02943/3340

.....

C16 + Joystick + Datensette + 55 ab-

geleitete Programme + Basic-Kurs =

150 DM

C116 + Joystick + Datensette + Basic-

Kurs = 85 DM

Tel.: 02833/1297 ab 18 Uhr

.....

Für C16/116/Plus 4 Preiswert

Turbolape Super (64K) 19,50 DM

Elektronik Datensette 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

.....

Verkaufe Commodore Floppy 1541 in

sehr gutem Zustand!!! Preis VB 320,-

DM. Anfragen bei Oliver Dangel, Ries-

bergstr. 37, 7157 Morhardt.

VC 1541/ VC 1541/ VC 1541

.....

Verkaufe VC-20+27/32 KB +

Datensette

ca. 80 - 100 Originalprogramme

für 150,- DM VB. Verkäufe an

Meistbietenden. Christoph

Flesch, Dianastr. 42,

4200 Oberhausen 11,

Tel. 0208/640715.

Verkaufe C-16 mit Datensette und 20

Originalspiele + 2 Bücher + Joy +

Computerhefte für 300,- DM. Anfragen

Robert Salzgeber, Sänststr. 40,

7970 Leutkirch 1

.....

Verkaufe C16-Datensette + SUPER-

SPIELE (z.B. Quivi, Winter Olympiade)

+ 64 KB-Erweiterung.

Michael Plahn, Suedergeest 17,

2241 Wrohm, Tel.: 04802/1066

.....

Commodore 16 mit Datensette, Joystick

und 10 Spielprogrammen wegen

Systemwechsel zu verkaufen. Preis:

100,- DM. Tel.: 0561/514902

.....

Verkaufe C16 + Datensette

dazu einiges Zubehör, sehr billig, Anruf

lohnt bestimmt.

06183/2639 nach 17 Uhr

.....

PLUS 4 + PLUS 4 + PLUS 4

Datenfernübertragung jetzt nur

ein Kinderspiel. DFÜ Komplett-

anlage bestehend aus Adapter,

Modem (ohne FTZ-Nr.) und

Supersoftware. Wählen von der

Computertastatur:

* WAHLWIEDERHOLUNG

* AUTOANSWER

* MAILBOXBETRIEB

229,- DM

Ihr DFÜ-Spezialist

GUSS DATA CONNECTION,

Tel.: 02723/6679

.....

Komplettsystem! Informatikstud. ver-

kauft wegen Systemwechsel Commo-

dore Plus/4 (64K RAM eingebaute Soft-

ware), Datensette 1531, Floppy 1551,

Drucker Citizen 129 D (120 Z/s, NLQ,

200 Schriftarten), 4 Joysticks, gute Lite-

ratur und über 200 Programmen (z. B.

Winter Games, Tom usw.) noch mit Ga-

rantie gegen Gebot! Tel. 08652/64157.

.....

C-16 mit 2 Spielkassetten und ein paar

Magazinen zu verkaufen. Schriftliche

Angebote an: Thomas Adams, Wupper-

Str. 13, 5650 Solingen 1

.....

C16 - DFÜ - C16 - DFÜ - C16 -

Datenfernübertragung wird zum

Kinderspiel!

Erweiterungsplatine >USMOI<

für C16/116 mit RS-232 und zwei

Userports für DM 138,-!

Modem steckfertig mit passen-

der Software (ohne FTZ-Nr.):

* Wahl von der Computertastatur

* Autoanswer

* Mailboxbetrieb DM 148,-

* Speichererweiterung DM 49,-

* Guss Data Connection,

Tel.: 02723/6679

.....

Verkaufe fast neuen C-16 + Datensette

mit Hand- und Basicbuch + Joystick +

20 Top Games + ca. 40 Programmen

für nur 300,- DM. Tel. 0203/746822