

L. Oswald

# Programme für CBM



Alle Programme wurden von uns sorgfältig geprüft und getestet, sollten sich trotzdem Fehler oder ein Datenverlust einstellen, kann keine Haftung übernommen werden.

CBM ist ein Warenzeichen der Firma Commodore International Ltd.

Dieses Buch ist eine unabhängige Produktion der Ing. W. Hofacker GmbH

**Titelfotos:**

Werkbild Fuji Photo Film USA Inc.  
Magnetic Tape Division  
und Commodore Pressefoto

ISBN 3-921682-81-9

Es kann keine Gewähr dafür übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen, sowie Programmlistings frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben auch keinen Aufschluß über eventuelle Verfügbarkeit oder Liefermöglichkeit. In jedem Falle sind die Unterlagen der Hersteller zur Information heranzuziehen.

Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen verboten. Programmlistings dürfen weiterhin nicht in irgendeiner Form vervielfältigt oder verbreitet werden. Alle Programmlistings sind Copyright der Fa. Ing. W. Hofacker GmbH. Verboten ist weiterhin die öffentliche Vorführung und Benutzung dieser Programme in Seminaren und Ausstellungen. Irrtum, sowie alle Rechte vorbehalten.

**COPYRIGHT BY ING. W. HOFACKER © 1981, Postfach 75 437,  
8000 München 75**

1. Auflage 1981

Gedruckt in der Bundesrepublik Deutschland – Printed in West-Germany – Imprime'en RFA.

**Programme**  
**für**  
**CBM**



# Inhaltsverzeichnis

Roulette .....	1
Textverarbeitung und Rechnungen schreiben .....	7
Karambolage .....	15
Mäuse .....	19
Musik mit CBM .....	21
Bildschirmausdruck .....	25
Super Drei .....	27
Karusell .....	33
Lotto .....	39
Blockade .....	41
Crash .....	45
String Sort .....	49
Vier gewinnt .....	51
PET als Adressbuch .....	59
Konto 99 .....	63
Auto-Jahresabrechnung .....	67
Word Special .....	71
Laufschrift .....	81
Kreisberechnung .....	85
Roulett-Auswertung .....	87
Parallel- und Serienschaltung .....	89
Rechnungsschreiben .....	91
Poly-Opti .....	95
Druckaufbereitung .....	99
ROM-Vergleichsliste .....	105
Wichtige Routinen in PET-BASIC .....	115



## VORWORT

Das Buch "Programme für CBM" soll Ihnen die Möglichkeit geben leistungsfähige Programme zu minimalen Kosten auf Ihrem System zu implementieren.

Wir haben bei der Zusammenstellung der Programme an den ernsthaften Anwender sowie an den Kreis gedacht, der den Mikrocomputer auch zur Zerstreuung in der Freizeit nutzen möchte. Deshalb sind Rechnungsschreibprogramme, Textverarbeitung usw., aber auch lustige Spielprogramme wie Musik, Mäuse, Lotto usw. dabei.

Die Programme verwenden kaum PEEK und POKE Befehle, so daß die meisten Programme auf allen CBM- und PET-Computern laufen dürften. Trotzdem haben wir am Schluß des Buches noch eine ROM-Vergleichsliste beigefügt, die Ihnen sicherlich gute Dienste leisten kann.

Wir wünschen Ihnen bei der Programmierung viel Spaß und Erfolg.

Holzkirchen, Frühjahr 1981

L. Oswald



# Roulette

## ROULETTE

Dieses Programm stellt eine vollständige Simulation nach den internationalen Vereinbarungen dar. Nach Run spielt das Programm zehn Spiele zur Demonstration. Auf den Bildschirm können Sie an Hand eines Pfeiles und einem Asterik sehen, welche Zahl gefallen ist. Nach dieser kurzen Einleitung werden Sie aufgefordert selbst zu setzen. Sie können jetzt einen Betrag eingeben den Sie insgesamt in dieser Runde "verspielen" möchten. Das Programm führt automatisch Buch über Ihre Gewinne und Verluste.

Sie können auf die Zahlen 1 - 36 setzen. Ebenfalls sind Serien oder Gerade sowie Zahlen, Dutzend, Manque, Passe und Transversalen

setzbar.

TA = Transversale 1-2-3  
TB = Transversale 4-5-6  
IM = Transversale 34-35-36

Bei 0 sind alle Einsätze verloren

P = PAIR (alle geraden Zahlen)  
I = IMPAIR (alle ungeraden Zahlen)  
MA = MANQUE (Serie von 1 - 18)  
PA = PASSE (Serie von 19 - 36)  
D1 = 1. Dutzend von 1 - 12  
D2 = 2. Dutzend von 13 - 24  
D3 = 3. Dutzend von 25 - 36  
R = Rot  
S = Schwarz

```
10 PRINT"☺"  
20 PRINT  
30 PRINT"          ■ R O U L E T T E ■"  
40 PRINT:PRINT  
50 PRINT"EINE VOLLSTAENDIGE SIMULATION NACH IN-  
60 PRINT"TERNATIONALER VEREINBARUNG"  
70 PRINT"XXXXXXXX"  
80 PRINT"FOLGENDE SPEZIELLE INSTRUKTIONEN SIND  
90 PRINT"JEDOCH ZU BEACHTEN:"  
100 PRINT"XXXXXXXX"  
110 FORI=1TO38:PRINT".":NEXT  
120 PRINT:PRINT  
130 PRINT"          ␣SPACE - DRUECKEN■"  
140 GETA$:IFA$=""THEN140  
150 IFA$="" THEN160  
160 PRINT"☺"  
170 PRINT"IN DIESER INTERNATIONALEN VERSION KOEN-  
180 PRINT"EN SIE EINE BELIEBIGE ZAHL VON :"  
190 PRINT"■ 1-36 ■SETZEN."  
200 PRINT  
210 PRINT"EBENFALLS SIND ■SERIEN■, ODER ■GERADE■,"  
220 PRINT"ZAHLN, ■DUTZEND■, ■MANQUE■ UND  
230 PRINT"■PASSE■, SOWIE ■TRANSVERSALE■ SETZBAR (HIER BEDEUTET Z.B.:":PRINT"■"  
240 PRINT"■TA■= TRANSVERSALE 1-2-3  
250 PRINT"■TB■= TRANSVERSALE 4-5-6":PRINT" .":PRINT" .":PRINT" ."  
260 PRINT"■TM■= TRANSVERSALE 34-35-36"  
270 PRINT  
280 PRINT"00....."  
290 PRINT  
300 PRINT"          ␣SPACE - DRUECKEN"  
310 GOSUB2640
```

```

320 PRINT"□"
330 PRINT"DIE ZAHLEN VON 1-36 SIND SPIELZAHLEN= (PLAIN-ZAHLEN)"
340 PRINT
350 PRINT"KOMMT 0, DANN ALLE EINSATZES VERLOREN."
360 PRINT"PAIR (P)=GERADE ZAHLEN
370 PRINT"IMPAIR(I)=UNGER. ZAHLEN
380 PRINT
390 PRINT"MANQUE(M)=SERIE VON 1-18
400 PRINT"PASSE (PA)=SERIE VON 19-36
410 PRINT
420 PRINT"D1 =SERIE VON 1-12
430 PRINT"D2 =SERIE VON 13-24
440 PRINT"D3 =SERIE VON 25-36":PRINT"D"
450 PRINT"ROT R ODER R = ROTE ZAHLEN
460 PRINT"SCHWARZ ODER S = SCHW.ZAHLEN":PRINT"D"
470 PRINT"WERFEN SIE NUN MIT SPACE EINEN BLICK AUF DEN TISCH !"
480 GOSUB2640
490 PRINT"□"
500 GOSUB2210
510 FORZZ=1TO10
520 X$="*":Z$="□":U$="□"
530 X=INT(RND(1)*37)
540 IFX<10THENY$="□":GOTO560
550 Y$="□"
560 GOSUB2180
570 IFG=1THENA$="R":GOTO590
580 A$="S"
590 G=0:PRINT"00000":IFX<10THENPRINTTAB(6)"X"|"A$:GOTO610
600 PRINTTAB(5)"|"X"|"A$
610 IFX>12THEN630
620 ONX+1GOTO660,690,700,710,720,730,740,750,760,770,780,790,800
630 IFX>25THEN650
640 ONX-12GOTO810,820,830,840,850,860,870,880,890,900,910,920,930
650 ONX-25GOTO940,950,960,970,980,990,1000,1010,1020,1030,1040
660 PRINT"00":PRINTTAB(17)X$
670 GOSUB2000
680 PRINTTAB(17)Y$:GOTO1050
690 K=1:GOTO2480
700 K=2:GOTO2480
710 K=3:GOTO2480
720 K=1:GOTO2490
730 K=2:GOTO2490
740 K=3:GOTO2490
750 K=1:GOTO2500
760 K=2:GOTO2500
770 K=3:GOTO2500
780 K=1:GOTO2510
790 K=2:GOTO2510
800 K=3:GOTO2510
810 K=1:GOTO2520
820 K=2:GOTO2520
830 K=3:GOTO2520
840 K=1:GOTO2530
850 K=2:GOTO2530
860 K=3:GOTO2530
870 K=1:GOTO2540
880 K=2:GOTO2540
890 K=3:GOTO2540
900 K=1:GOTO2550
910 K=2:GOTO2550
920 K=3:GOTO2550

```

```

930 K=1:GOTO2560
940 K=2:GOTO2560
950 K=3:GOTO2560
960 K=1:GOTO2570
970 K=2:GOTO2570
980 K=3:GOTO2570
990 K=1:GOTO2580
1000 K=2:GOTO2580
1010 K=3:GOTO2580
1020 K=1:GOTO2590
1030 K=2:GOTO2590
1040 K=3:GOTO2590
1050 NEXTZ
1060 GOSUB2000
1070 PRINT"♣":FORI=1TO20:PRINT:NEXT
1080 PRINT"♣      NUN SIND SIE AN DER REIHE!♣";
1090 GOSUB2000:GOSUB2170
1100 PRINT"♣WIEVIEL GELD HABEN SIE UEBERHAUPT ZUM  WEGSCHMEISSEN BEI SICH ";
1110 INPUTA5:PRINT"♣TT":GOSUB2170
1120 PRINT"♣SETZEN SIE EINE ♣ZAHL♣ ODER ♣SERIE♣:";
1130 INPUTB$:PRINT"♣TT":GOSUB2170
1140 INPUT"♣WIE HOCH WOLLEN SIE SETZEN";C
1150 IF C<=A5THEN1190
1160 PRINT"♣TT":GOSUB2170:PRINT"SOVIEL HABEN SIE NICHT MEHR!♣":GOSUB2000
1170 GOSUB2170:PRINT"♣SETZEN SIE IM RAHMEN IHRER MOEGlichkeit!♣":GOSUB2000
1180 GOSUB2170:PRINT"♣TT":GOTO1140
1190 GOSUB2130:PRINT"♣TT":GOSUB2170
1200 IF VAL(B$)>0AND VAL(B$)<37THEN1790
1210 IF B$="P"OR B$="PAIR"THEN1400
1220 IF B$="I"OR B$="IMPAIR"THEN1450
1230 IF B$="MA"OR B$="MANQUE"THEN1500
1240 IF B$="PA"OR B$="PASSE"THEN1530
1250 IF B$="D1"OR B$="D2"OR B$="D3"THEN1570
1260 IF B$="R"OR B$="ROT"THEN1620
1270 IF B$="S"OR B$="SCHWARZ"THEN1650
1280 IF B$="T"AND B$<"TN"AND RIGHT$(B$,1)<>"J"THEN1310
1290 PRINT"♣TT":GOSUB2170:PRINT"JUNGUELTIG, BITTE NEU SETZEN!♣":GOSUB2000
1300 GOSUB2170:PRINT"♣TT":GOTO1120
1310 GOSUB2000
1320 IF (B$="TA"AND X<4)OR (B$="TB"AND X>3AND X<7)THEN1390
1330 IF (B$="TC"AND X>6AND X<10)OR (B$="TD"AND X>9AND X<13)THEN1390
1340 IF (B$="TE"AND X>12AND X<16)OR (B$="TF"AND X>15AND X<19)THEN1390
1350 IF (B$="TG"AND X>18AND X<22)OR (B$="TH"AND X>21AND X<25)THEN1390
1360 IF (B$="TI"AND X>24AND X<28)OR (B$="TK"AND X>27AND X<31)THEN1390
1370 IF (B$="TL"AND X>30AND X<34)OR (B$="TM"AND X>33)THEN1390
1380 GOTO1720
1390 W=12:GOTO1680
1400 GOSUB2000
1410 J=1
1420 IF J=XTHEN1720
1430 J=J+2:IF J<36THEN1420
1440 GOTO1670
1450 GOSUB2000
1460 J=1
1470 IF J=XTHEN1670
1480 J=J+2:IF J<36THEN1470
1490 GOTO1720
1500 GOSUB2000
1510 IF X>0AND X<19THEN1670
1520 GOTO1720
1530 GOSUB2000

```



```

2120 GOSUB2000:GOSUB2170:GOTO1120
2130 FORJ=1TO5:PRINT"NICHTS GEHT MEHR!".
2140 FORI=1TO70:NEXT
2150 PRINT"NICHTS GEHT MEHR!":FORI=1TO70:NEXT:NEXT:RETURN
2160 FORI=1TO15:PRINT:NEXT
2170 FORE=0TO120:PRINT" ";NEXT:PRINT"!!":RETURN
2180 IFX=10RX=30RX=50RX=70RX=90RX=120RX=140RX=160RX=180RX=19THENG=1:RETURN
2190 IFX=210RX=230RX=250RX=270RX=300RX=320RX=340RX=36THENG=1
2200 RETURN
2210 PRINT"!"
2220 PRINT"
2230 PRINT"
2240 PRINT"
2250 PRINT"
2260 PRINT"
2270 PRINT"
2280 PRINT"
2290 PRINT"
2300 PRINT" TISCH IM | IP | 13 | 14 | 15 | II |
2310 PRINT" SPIELBE- | IA | 16 | 17 | 18 | IM |
2320 PRINT" TRIEB | II | 19 | 20 | 21 | IP |
2330 PRINT"
2340 PRINT" <LAEUFT
2350 PRINT" 10 MAL | IN | 25 | 26 | 27 | IR |
2360 PRINT" DANN KOENNEN IO | 28 | 29 | 30 | IO | SIE HABEN |
2370 PRINT" SIE SELBST II | 31 | 32 | 33 | IU | NOCH DM: |
2380 PRINT" SPIELEN! IR | 34 | 35 | 36 | IG |
2390 PRINT"
2400 PRINT"
2410 PRINT"
2420 PRINT
2430 RETURN
2440 ONKGOTO2450,2460,2470
2450 GOSUB2020:GOSUB1990:GOTO2030
2460 GOSUB2040:GOSUB1990:GOTO2050
2470 GOSUB2060:GOSUB1990:GOTO2070
2480 PRINT"#####":GOTO2600
2490 PRINT"#####":GOTO2600
2500 PRINT"#####":GOTO2600
2510 PRINT"#####":GOTO2440
2520 PRINT"#####":GOTO2440
2530 PRINT"#####":GOTO2440
2540 PRINT"#####":GOTO2440
2550 PRINT"#####":GOTO2440
2560 PRINT"#####":GOTO2440
2570 PRINT"#####":GOTO2440
2580 PRINT"#####":GOTO2440
2590 PRINT"#####":GOTO2440
2600 ONKGOTO2610,2620,2630
2610 PRINTTAB(16)X$:GOSUB1990:PRINTTAB(16)Y$:GOTO2010
2620 PRINTTAB(19)X$:GOSUB1990:PRINTTAB(19)Y$:GOTO2010
2630 PRINTTAB(22)X$:GOSUB1990:PRINTTAB(22)Y$:GOTO2010
2640 GETA$:IFA$=""THEN2640
2650 RETURN
READY.

```

## NOTIZEN

# Textverarbeitung und Rechnungen schreiben

## Textverarbeitung und Rechnungen schreiben

Dieses Programm erklärt sich nach dem eingeben von RUN fast von selbst. Es erlaubt Ihnen das Erstellen von

1. Rechnungen
2. Lieferscheinen
3. Lieferanten Bestellungen
4. Mahnungen

Ab Zeile 550 sollen Sie zuerst die Daten für Ihren Firmennamen, Konto und Lieferungs- und Zahlungsbedingungen eingeben.

Wenn alle Positionen eingegeben sind, wird als Abschluß eine Menge von 0 eingegeben. Dies führt Sie aus der Abfrageschleife über die einzelnen Positionen heraus. Unser Beispiel enthält drei Positionen und berücksichtigt einen Gesamtrabatt von 20%, der bei jeder Position in gleicher Weise zum Abzug kommt. Es lassen sich auch unterschiedliche Rabatte pro Position berechnen. Das Datum braucht nur einmal zu

Beginn des Rechnungsschreibens eingegeben werden. Es bleibt dann für alle anderen Kunden an diesem Tage bestehen. Wollen Sie das Datum ändern, müssen Sie das Programm anhalten und neu starten.

In Zeilen 550 – 630 finden Sie die Daten über den Absender der Rechnung. Auch die Zeilen 2170 – 2220 und 2740 – 2790 müssen Sie an Ihre Firma und Bedingungen anpassen. Sie können diese nach Ihren eigenen Vorstellungen und Bedingungen gestalten.

Das Programm wurde auf einem CBM 3032 geschrieben und getestet. Das eigentliche Programm benötigt 12K RAM zur Ausführung in diese Version ist ein RAM-Bereich von 32K empfehlenswert, da sonst die DIM-Befehle in Zeile 670 und 680 entfernt werden müssen und eine einwandfreie Funktion nicht mehr gewährleistet ist.

```
110 REM RECHNUNG/LIEFERSCHEIN/EIGENE BE STELLUNG/BRIEFTEXT
120 REM SPEICHERBEDARF 13 KB
130 REM
140 REM
150 POKE59468,14
160 PRINTCHR$(147)
170 PRINT" ***** BEDIENUNGSANWEISUNG! *****"
180 PRINT
190 PRINT"ANTWORTEN SIE BITTE GENAU AUF MEINE  "
200 PRINT"FRAGEN.WENN ICH ZB. MENGE ? FRAGE,DANN"
210 PRINT"DUERFEN SIE NUR MIT ZAHLEN ANTWORTEN !"
220 PRINT"NACHDEM SIE EINE FRAGE BEANTWORTET HA-"
230 PRINT"BEN,DRUECKEN SIE IMMER DIE RETURN-TASTE"
240 PRINT"DAMIT ICH WEISS,WANN IHRE EINGABE BEEN-"
250 PRINT"DET IST,NUR WENN ICH SIE BITTE,EINE ENT"
260 PRINT"SPRECHENDE ZIFFER ZU DRUECKEN,WEISS ICH"
270 PRINT"OHNE RETURN BESCHIED."
280 PRINT"SOLLTEN SIE SICH WAEHREND DER EINGABE"
290 PRINT"VERSCHREIBEN,DANN LOESCHEN SIE MIT DER"
300 PRINT"DEL-TASTE DIE FALSCHEN ZEICHEN!"
310 PRINT"WENN ALLERDINGS EINE NEUE FRAGE ER-"
320 PRINT"SCHIEINT,KOENNEN SIE AN DER VORIGEN ANT-"
330 PRINT"WORT NICHTS MEHR AENDERN."
340 PRINT"SIE MUESSEN DANN BEI MEINER NAECHSTEN"
350 PRINT"FRAGE OHNE ANTWORT AUF RETURN DRUECKEN"
360 PRINT"UND NACH MEINER READY-MELDUNG NEU AN-"
370 PRINT"FANGEN,INDEM SIE RUN SCHREIBEN !"
```

```

380 PRINT:PRINT" FUEER START LEERTASTE DRUECKEN !!"
390 GETS$:IFS$="" THEN390
400 PRINTCHR$(147):PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
410 INPUT" DATUM";DA$:PRINT
420 REM*****
430 REMUMSCHALTUNG AUF GROSSBUCHSTABEN
440 POKE59468,12
450 REM*****
460 REMDISTANZFELDER FUEER TEXTAUDRUCK
470 D$=" " :E$=" " :F$=" " :H$=" " :J$=" " :V$=" "
480 REM*****
490 REM GRAFIK FUEER BRIEFKOPF
500 SS$=" ~~~~~~"
510 SB$=" ~~~~~~"
520 DH$=" ~~~~~~"
530 DJ$=" ~~~~~~"
540 HE$="*****"
550 FI$="ERNST FISCHER GMBH & CO KOMMANDITGES."
560 SW$=" 5138 HEINSBERG-UNTERBRUCH * WASSENBERGERSTRASSE 47-49*TEL 02452/6071"
570 KO$="KONTEN>>"
580 VB$="HEINSBERGER VOLKSBANK KTO 1748"
590 PA$="POSTSCHECKKTO KOELN 211824-505"
600 ZB$="ZAHLBAR REIN NETTO BEI LIEFERUNG!"
610 BE$="BEANSTANDUNGEN NUR INNERHALB 3TAGEN NACH WARENEMPFANG. BIS ZUR VOEL-"
620 BZ$="LIGEN BEZAHLUNG BLEIBT DIE WARE UNSER EIGENTUM. "
630 ER$="ERFUELLUNGSORT UND GERICHTSSTAND FUEER BEIDE TEILE IST HEINSBERG."
640 PRINTCHR$(147):REM LOESCHT BSCHIRM
650 REM*****
660 REM MATRIXFELDER FUEER SPEICHER
670 DIMA$(60 ),ME$(60 ),EP$(60 ),G$(60 ),RA$(60 ),RB$(60 ),NE$(60 )
680 DIM EP$(60 ),G$(60 ),ME$(60 ),RA$(60 ),NE$(60 ),RB$(60 )
690 REM*****
700 REM ADRESSENEINGABE
710 PRINTCHR$(147):PRINT:PRINT:PRINT:PRINT
720 GOSUB3540:PRINT
730 PRINT"X"
740 INPUT" ) NAME";NA$:PRINT
750 PRINT"X"
760 INPUT" ) STRASSE";ST$:PRINT
770 PRINT"X"
780 PRINT" ====="
790 PRINT:GOSUB3540
800 INPUT" ) SORT";O$:PRINT
810 PRINTCHR$(147)
820 REM*****
830 REM ARBEITSENTSCHEIDUNG
840 GE=0:TO=0:SU=0
850 PRINTCHR$(147):GOSUB3540:PRINT
860 PRINT" RECHNUNG 01":PRINT
870 PRINT" LIEFERSCHEIN 02":PRINT
880 PRINT" LIEF.BESTELLUNG 03":PRINT
890 PRINT" MAHNUNG 04":PRINT
900 GOSUB3540:PRINT
910 PRINT"BITTE NUR DIE ZIFFER DRUECKEN !!"
920 GET TA$:IF TA$="" THEN 920
930 IF TA$="1" THEN GOTO 980
940 IF TA$="2" THEN GOTO 1020
950 IF TA$="3" THEN GOTO 1060
960 IF TA$="4" THEN GOTO 1080
970 PRINTCHR$(147):GOTO840
980 PRINT:PRINT"EINZELRABATTE ? (J/N)"

```



```

1600 REM*****
1610 REM EVTL.RABATTE & AUFSCHLAEGE
1620 PRINT"NACHLASS NEIN 30"
1630 PRINT" IN % 31"
1640 PRINT" IN DM 32"
1650 PRINT"AUFSCHLAG 13%MWST 33":T0=SU
1660 PRINT:PRINT"ABZ.% + 13% MWST 34"
1670 PRINT:GOSUB3540:PRINT
1680 GETTN$:IFTN$=""THENGOTO1680
1690 IFTN$="0"THENPOKE32989,211:GOTO1940
1700 IFTN$="1"THENPOKE33069,211:GOTO1790
1710 IFTN$="2"THENPOKE33149,211:GOTO1750
1720 IFTN$="3"THENPOKE33229,211:GOTO1840
1730 IFTN$="4"THENPOKE33309,211:GOTO1790
1740 PRINTCHR$(147):GOTO1590
1750 INPUT" NACHLASS IN DM: ";GL
1760 X=GL:GOSUB3250:GL$=X$:GL=VAL(GL$)
1770 T0=SU-GL
1780 GOTO1880
1790 INPUT" NACHLASS IN % : ";PS:PS$=STR$(PS)
1800 NL=(SU*PS)/100
1810 X=NL:GOSUB3250:NL$=X$:NL=VAL(NL$)
1820 T0=SU-NL
1830 IFTN$="1"THENGOTO1880
1840 MW=(T0*13)/100:X=MW:GOSUB3250:MW$=X$
1850 MW=VAL(MW$):T0=SU+MW
1860 IFTN$="3"THENGOTO1880
1870 T0=SU-NL+MW
1880 PRINT:PRINT
1890 PRINT" SUMME =DM";SU$
1900 PRINT" -----"
1910 IFTN$="2"THENPRINT" NACHLASS=DM";GL$
1920 IFTN$="1"ORTN$="4"THENPRINT" NACHLASS=DM";NL$
1930 IFTN$="3"ORTN$="4"THENPRINT" 13%MWST=DM";MW$
1940 PRINT" -----"
1950 IFTN$="0"THENT0$=SU$
1960 X=T0:GOSUB3250:T0$=X$:T0=VAL(T0$)
1970 PRINT" TOTAL =DM";T0$
1980 PRINT" ====="
1990 REM*****
2000 REM WARENWERT UND MWST
2010 WW=T0/1.13:MS=T0-WW
2020 X=WW:GOSUB3250:WW$=X$
2030 X=MS:GOSUB3250:MS$=X$
2040 PRINT
2050 PRINT" WARENWERT=";WW$
2060 PRINT" 13% MWST=";MS$
2070 RETURN
2080 OPEN1,4:REM NORMALER AUSDRUCK
2090 OPEN6,4,6:REM ZEILENABSTAND
2100 REM*****
2110 REM UNTERSTREICHEN DES WOHNORT
2120 FORLE=1TOLEN(0$):OX$=" "+OX$:NEXT
2130 REM*****
2140 REM ANZAHL DER KOPIEN
2150 FOR Y=1TO1
2160 REM*****
2170 REM AUSDRUCK-BEFEHLE
2180 PRINT#1,CHR$(1);" ";DH$
2190 PRINT#1,CHR$(1);" ";FI$
2200 PRINT#1,CHR$(17);" ";SW$

```





```

3410 RETURN
3420 REM**==**==**==**==**==**==**==**==**==**==
3430 REM ADDRESS-AUFKLEBER DRUCKEN
3440 OPEN1,4:OPEN4,4
3450 PRINT#4
3460 PRINT#1,NA$:SPC(40-LEN(NA$));NA$
3470 PRINT#1,ST$:SPC(40-LEN(ST$));ST$
3480 PRINT#1,0$:SPC(40-LEN(0$));0$
3490 PRINT#1,OX$:SPC(40-LEN(OX$));OX$
3500 CLOSE4:CLOSE1
3510 RETURN
3520 REM**==**==**==**==**==**==**==**==**==**==
3530 REM UMRANDUNGEN
3540 FOR Z=1TO40:PRINTCHR$(230);:NEXT
3550 RETURN
3560 REM**==**==**==**==**==**==**==**==**==**==
3570 REM MAHNUNGS-ROUTINE
3580 PRINTCHR$(147):PRINT:PRINT"          *** MAHNUNG ***"
3590 PRINT:PRINT:PRINT
3600 INPUT"RECHNUNG VOM ";YX$
3610 PRINT:PRINT
3620 INPUT"MAHNBETRAG ";MD$
3630 PRINT:PRINT
3640 GOSUB3540:PRINT
3650 PRINT" 1.MAHNUNG      11":PRINT
3660 PRINT" 2.MAHNUNG      22":PRINT
3670 PRINT" 3.MAHNUNG      33":PRINT
3680 GOSUB3540
3690 PRINT:PRINT" BITTE ENTSPRECHENDE ZIFFER DRUECKEN!"
3700 GETTM$:IFTM$=""THEN3700
3710 IFTM$="1"THENGOTO3750
3720 IFTM$="2"THENGOTO3820
3730 IFTM$="3"THENGOTO3890
3740 GOTO3580
3750 U$="          MAHNUNG"
3760 M1$="SIE HABEN ES SICHER UEBERSEHEN,DIE UNTENSTEHENDE FORDERUNG"
3770 M2$="ZU BEGLEICHEN."
3780 M3$="BITTE NEHMEN SIE DIE ERINNERUNG NICHT UEBEL."
3790 M4$="          MIT BESTER EMPFEHLUNG"
3800 M5$=""
3810 RETURN
3820 U$="          2.MAHNUNG"
3830 M1$="IST DIE ERSTE ANFRAGE BEZUEGLICH DES FAELLIGEN BETRAGES"
3840 M2$="NICHT IN IHREN BESITZ GELANGT,ODER IST DER BETRAG"
3850 M3$="SCHON UNTERWEGS?"
3860 M4$="          MIT BESTER EMPFEHLUNG"
3870 M5$=""
3880 RETURN
3890 U$="          LETZTE MAHNUNG"
3900 M1$="WISSEN SIE,DASS IHNEN SCHON ZWEI MAHNUNGEN ZUGINGEN?"
3910 M2$="SIE HABEN GEWISS VERSTAEANDNIS DAFUER,DASS JEDER AUF DEN EINGANG"
3920 M3$="SEINER AUSSENSTRENDE ANGEWIESEN IST."
3930 M4$="BITTE UEBERWEISEN SIE DEN BETRAG NUN ABER BESTIMMT UND SOFORT!"
3940 M5$="          MIT BESTER EMPFEHLUNG"
3950 RETURN

```

READY.

**ERNST FISCHER GMBH & CO KOMMANDITGES.**  
5138 heinsberg-unterbruch \* wassenbergsstrasse 47-49\*tel 02452/6071

FIRMA ALBERT SAMPLE  
NICHTSTRASSE 12  
8000 MUENCHEN 22

3.1.1981

RECHNUNG

MENGE	ARTIKEL - BEZEICHNUNG	DM EINZEL	DM GESAMT
10	AUSGLEICHSGEWICHTE.....	19.80	198.00
10	WASSERWAAGEN.....	55.95	559.50
15	HANDBUECHER.....	12.80	192.00
		SUMME	949.50
		ABZUEGLICH 20%	189.90
		TOTAL DM	759.60

!!WIR BEDANKEN UNS FUER IHREN AUFTRAG!!

WW DM 672.21\*\* 13% MWST DM 87.39  
ZAHLBAR REIN NETTO BEI LIEFERUNG! BETRAG ERHALTEN:

konten>>heinsberger volksbank kto 1748\*\*postscheckkto koeln 211824-505  
beanstandungen nur innerhalb 8tagen nach wareneingang. bis zur voel-  
ligen bezahlung bleibt die ware unser eigentum.  
erfuellungsort und gerichtstand fuer beide teile ist heinsberg.

# Karambolage

## KARAMBOLAGE

Dies ist ein Billiardähnliches Spiel. Sie geben die Anzahl der Spieler und deren Namen ein. Dann wird die Anzahl der Stöße pro Spieler und Durchgang eingegeben. Auf dem "Billardtisch"

erscheinen zwei Ziele und ein Ball. Durch eine geschickte Wahl des Winkels muß jedes Ziel so getroffen werden, daß der Ball durch die Mitte des Zielobjektes läuft.

```
400 GOSUB1000
410 GOSUB9000
420 GOSUB1500
500 P=P+1:IFP>NPTHENP=1:I=I+1
510 IFI>NITHEN7500
520 GOSUB2000
530 GOSUB2500
540 GOSUB3000
550 GOSUB3500
560 GOSUB4000
570 GOSUB4500
700 GOSUB5000
710 GOSUB5500
720 IFNB=>3THENGOSUB6000:GOTO500
730 GOSUB6500
740 IFCU=>8THENGOSUB6000:GOTO500
750 GOTO700
1000 REM***
1010 RX=0:DX=0:RY=0:DY=0:TX=0:TY=0:XR=0:YU=0:CU=0:IX=0:IY=0:X1=0
1020 Y1=0:X2=0:Y2=0:A=0:B=0:NB=0:I=0:S1=0:S2=0:P=2:DIMP$(2):RETURN
1500 REM***
1510 GOSUB8800:GOSUB8400:PRINT"KARAMBOLAGE":PRINT
1520 INPUT"WIEVIELE SPIELER(1/2)";NP
1530 NP=INT(NP):IFNP<1ORNP>2THEN1520
1540 IFNP=1THENINPUT"DEIN NAME";P$(1):GOTO1560
1550 INPUT"NAME VON SPIELER 1";P$(1):INPUT"NAME VON SPIELER 2";P$(2)
1560 INPUT"WIEVIELE STÖESSE ";NI
1570 NX=39:NY=18:SR=.865
1590 K=RND(TI):XR=NX-2
1600 YU=NY-2:RETURN
2000 REM***
2010 GOSUB8800:GOSUB8400:PRINT"ALLES KLAR ";P$(P):PRINT:PRINT
2020 PRINT" SCHMIER DEINEN KOEHN EIN!":GOSUB8900:GOSUB8800
2030 GOSUB8400:PRINT:PRINT
2040 PRINT"VERSUCHMI";I;" VON ";NI;" *** STOSS ZU, ";P$(P):PRINT
2050 PRINT"WERTUNG: ";P$(1);"-";S1:IFNP=2THENPRINT" ";P$(2);"-";S2
2060 RETURN
2500 REM***
2510 GOSUB8600:A=X:FORK=1TONX:PRINT"■";:GOSUB8100:NEXT:A=L:B=U:C=X
2520 FORK=1TONY-1:GOSUB8300:NEXT:GOSUB8600:A=U:B=X:C=L
2530 FORK=1TONY-1:GOSUB8300:NEXT:A=X:FORK=1TONX-1:GOSUB8100
2540 NEXT:PRINT"■";:RETURN
3000 REM***
3010 X1=INT(RND(1)*(NX-3))+1:Y1=INT(RND(1)*(NY-3))+2
3020 X2=INT(RND(1)*(NX-3))+1:Y2=INT(RND(1)*(NY-3))+2
```

```

3030 K=ABS(X1-X2):Q=ABS(Y1-Y2):IFK<3ANDQ<3THEN3020
3040 XB=INT(RND(1)*(NX-4))+2:YB=INT(RND(1)*(NY-4))+2
3050 K=(X1-XB)^2+(Y1-YB)^2:IFK<4THEN3040
3060 K=(X2-XB)^2+(Y2-YB)^2:IFK<4THEN3040
3070 RETURN
3500 REM***
3510 LX=X1:LY=Y1:GOSUB7000:A=OH:B=OH:GOSUB8200
3520 A=L:B=L:C=D:GOSUB8300:A=OH:B=OH:GOSUB8200
3530 LX=X2:LY=Y2:GOSUB7000:A=OH:B=OH:GOSUB8200
3540 A=L:B=L:C=D:GOSUB8300:A=OH:B=OH:GOSUB8200
3550 LX=XB:LY=YB:GOSUB7000:A=AS:GOSUB8100:RETURN
4000 REM***
4010 GOSUB8400:INPUT"SCHUSSWINKEL (0-360)":G:G=G*π/180
4020 DX=COS(G):DY=SIN(G):IFSR>1THENDX=DX/SR
4030 IFSR<1THENDY=DY*SR
4040 IFABS(DX)<.5ANDABS(DY)<.5THENDX=DX*2:DY=DY*2:GOTO4040
4050 RETURN
4500 REM***
4510 LX=XB:LY=YB:GOSUB7000:NB=0:CU=0:IX=XB:IY=YB:RX=XB+.49:RY=YB+.49
4520 GOSUB8900:A=SP:B=L:GOSUB8200:RETURN
5000 REM***
5010 RX=RX+DX:RY=RY+DY:TX=INT(RX):TY=INT(RY)
5020 A=N:B=N:IFTX<IXTHENA=R:GOTO5040
5030 IFTX<IXTHENA=L
5040 IFTY>IYTHENB=U:GOTO5060
5050 IFTY<IYTHENB=D
5060 GOSUB8200:IX=TX:IY=TY
5070 A=AS:B=L:GOSUB8200
5080 A=SP:B=L:GOSUB8200:RETURN
5500 REM***
5510 IFNB>1THEN5540
5520 IFIX<X2ORIX>X2+10RIY<Y2-10RIY>Y2THEN5540
5530 A=PL:B=L:GOSUB8200:NB=NB+2
5540 IFNB=10RNB=3THEN5570
5550 IFIX<X1ORIX>X1+10RIY<Y1-10RIY>Y1THEN5570
5560 A=PL:B=L:GOSUB8200:NB=NB+1
5570 RETURN
6000 REM***
6010 GOSUB8400:PRINT
6020 IFNB<3THENPRINT"KEINE WERTUNG!":GOTO6070
6030 SC=4-ABS(CU-4):PRINT"DIESER STOSS ZAEHLT: ";SC:IFP=1THENS1=S1+SC
6040 IFP=2THENS2=S2+SC
6050 PRINT:PRINT:PRINT"PUNKTE: ";P$(1);"-";S1
6060 IFNP=2THENPRINT" ";P$(2);"-";S2
6070 FORQ=1TO3:GOSUB8900:NEXT:RETURN
6500 REM***
6510 IFIX>1ANDIX<XRTHEN6530
6520 CU=CU+1:A=BL:GOSUB8100:DX=DX*-1
6530 IFIY>1ANDIY<YUTHEN6550
6540 CU=CU+1:A=BL:GOSUB8100:DY=DY*-1
6550 RETURN
7000 REM***
7010 GOSUB8600:A=R:FORK=1TOLX:GOSUB8100:NEXT
7020 A=U:FORK=1TOLY:GOSUB8100:NEXT:RETURN
7500 REM***
7510 PRINT:PRINT:PRINTTAB(10);"DAS WAR'S -- GUTES SPIEL! " :END
8000 REM***
8100 REM***
8110 PRINTCHR$(A):RETURN
8200 REM***
8210 PRINTCHR$(A);CHR$(B):RETURN

```

```
8300 REM***
8310 PRINTCHR$(A);CHR$(B);
8320 PRINTCHR$(C);:RETURN
8400 REM***
8410 A=19:GOSUB8100:RETURN
8600 REM***
8610 GOSUB8400:A=17:FORK=1TO24:GOSUB8100:NEXT:RETURN
8800 REM***
8810 A=147:GOSUB8100:RETURN
8900 REM***
8910 FORK=1TO2000:NEXT:RETURN
8920 REM***
9000 REM***
9010 REM***
9020 REM***
9030 REM***
9040 R=29:L=157:U=145:D=17
9050 X=32:OH=215:AS=209:SP=32:BL=0:N=0:PL=43
9060 REM
9070 RETURN
READY.
```



**CBM 3001 Serie mit ROM Version 3**

## NOTIZEN



```

1135 FORI=1T0100:GETP$:NEXT:GOTO68
1140 PRINT"XXXXXXXXXXXXXXXXDAS WAR SPITZE !"
1150 FORI=1T05000:NEXT
1160 PRINT"MEIN NEUES MATCH BEGINNT MIT TASTE N !"
1170 GETN$:IFN$="N"THENZL=0:ZR=0:GOTO68
1180 GOTO1170
2000 FORI=1T050:NEXT
2010 RETURN
3000 GETA$
3010 IFA$<"W"GOTO3100
3020 L$=A$:RETURN
3100 IFA$<"A"GOTO3200
3120 L$=A$:RETURN
3200 IFA$<"X"GOTO3300
3220 L$=A$:RETURN
3300 IFA$<"D"GOTO3400
3320 L$=A$:RETURN
3400 IFA$<"2"GOTO3500
3420 R$=A$:RETURN
3500 IFA$<"4"GOTO3600
3520 R$=A$:RETURN
3600 IFA$<"8"GOTO3700
3620 R$=A$:RETURN
3700 IFA$<"6"GOTO3800
3720 R$=A$:RETURN
3800 RETURN
4000 S=32768:REM RAHMEN
4005 PRINT"□":V=0
4010 FORH=0T039:GOSUB5000:NEXT:H=39
4020 FORV=0T023:GOSUB5000:NEXT:V=23
4030 FORH=39T00STEP-1:GOSUB5000:NEXT:H=0
4040 FORV=23T00STEP-1:GOSUB5000:NEXT
4042 LX= 1:LV=23:REM LINKS POS
4043 RX=38:RV=23:REM RECHTS POS
4045 L$="W":R$="8"
4050 RETURN
5000 POKE(S+V*40+H),102
5010 RETURN
READY.

```

# Musik mit CBM

## Musik mit CBM

Tonerzeugung mit Ihrem CBM- bzw. PET-Computer.

Das Schieberegister im 6522 VIA-Baustein des CBM eignet sich bestens zur Erzeugung von Toneffekten und Musik.

Was Sie zur Erzeugung von Musik brauchen, ist ein kleiner NF-Verstärker, den Sie an den CB2-Anschluß des User-Ports anschließen können. Sie können sich auch einen Musikboxbausatz unter

der Bestell-Nr. 444 zu 199,00 DM beim Hofacker Verlag bestellen. Dieses Gerät braucht keine eigene Stromversorgung. Es wird vom +5V Versorgungsteil für den zweiten Cassettenrecorder gespeist.

Beim Abtippen des Listings brauchen Sie die vielen Grafikzeichen in den PRINT-Statements nicht alle unbedingt auch einzugeben. Sie machen lediglich die Bedienung des Programmes komfortabler und eleganter.

```
50 DIMLO$(5):LO$(1)=" ***KLICKS***"
55 LO$(2)=" ***TOENE***"
60 LO$(3)=" ***LASER***"
65 LO$(4)=" ***SIRENE***"
70 LO$(5)=" ***GERAUESCHE***"
75 SW$="  ***TONSCHAU PROGRAMMIERBEISPIELE***"
80 ME$=" EXPERIMENTE MIT VERSCHIEDENEN WERTEN"
90 GOTO105
98 POKE59464,75:POKE59467,16:POKE59466,15:POKE59467,0:RETURN
105 B2$="#####!"
109 B3$=" "
110 B$="#####"
120 POKE59468,12:GOSUB6000
130 PRINTB$"DRUECKE 1 FUER KLICKS 4 FUER SIRENEN
140 PRINT " 2 FUER TOENE 5 FUER ZUFALLSTOENE
145 PRINT " 3 FUER LASER 6 THEORIE
150 GETG$:IFG$=""THEN150
160 IFG$="1"THEN400
170 IFG$="2"THEN500
180 IFG$="3"THEN600
190 IFG$="4"THEN700
195 IFG$="5"THEN800
197 IFG$="6"THEN7300
200 GOTO150
400 REM
402 PRINTSW$:PRINT:PRINTB2$: LO$(VAL(G$))
403 POKE59468,14
405 PRINT" DIESES UNTERPROGRAMM:"
410 PRINT"1000 POKE 59464,75
411 PRINT"1010 POKE 59467,16
412 PRINT"1020 POKE 59466,15
413 PRINT"1030 POKE 59467,0
415 PRINT"1040 RETURN
420 PRINT" ERZEUGT KLICKSEN, WOBEI (1TO255)";
425 PRINT" DIE TONHOEHE DES KLICKS BESTIMMT.
427 A=50:GOSUB1000
430 PRINTME$;
```

```

432 PRINT"FUER A# (Z.B. 200 RETURN)";
434 GOSUB910:PRINTB$
435 INPUT" A#=-          ";A$=VAL(A$)
440 IFAC10RA>255THENPRINT" A# BITTE 1-255";:FORN=1TO2000:NEXT:GOTO434
450 GOSUB1000:GOSUB950
465 POKE548,1:IFA$="R"THEN450
470 IFA$="N"THEN434
475 IFA$="M"THEN120
480 GOTO450
500 REM
502 PRINTSW$:PRINT:PRINTB2$; LO$(VAL(G$))
503 POKE59468,14
505 PRINT" DIESES UNTERPROGRAMM:"
510 PRINT"1100 POKE59466,0:POKE59464, A#:POKE59467,          16:POKE59466, C#"
513 PRINT"1110 FOR N=1TO B#:NEXT
515 PRINT"1120 POKE59467,0:RETURN"
520 PRINT"ERZEUGT TONE#, WOBEI A#(1-255) DIE:
522 PRINT"TONHOEHE, B#(1-64000) DIE TONDAUER IST UND";
524 PRINT" C#(1-255) BESTIMMT DIE OKTAVE U.DAS TEMPER.#
530 A=100:B=1000:C=15:GOSUB1100
535 PRINTME$;
540 PRINT"FOR A#, B#, C# (Z.B. 100,1000,15RETURN). "
550 GOSUB910:PRINTB$
555 INPUT" A#, B#, C#=-          ";A$,B$,C$
556 A=VAL(A$):B=VAL(B$):C=VAL(C$)
558 IFAC10RA>255THENPRINT" A# MUSS 1-255 SEIN";:FORN=1TO2000:NEXT:GOTO550
559 IFCC10RC>255THENPRINT" C# MUST BE 1-255";:FORN=1TO2000:NEXT:GOTO550
570 GOSUB1100:GOSUB950
585 POKE548,1:IFA$="R"THEN570
587 IFA$="N"THEN550
590 IFA$="M"THEN120
595 GOTO570
600 REM LASER
602 PRINTSW$:PRINT:PRINTB2$; LO$(VAL(G$))
603 POKE59468,14
605 PRINT" DIESES UNTERPROGRAMM"
606 PRINT"1000 POKE59466,0:POKE59467,16:POKE59466,          15";
607 PRINT"FORN=1TO A#:FORNN=A#TO B#STEP C#
608 PRINT"1010 POKE59464,NN:NEXT:NEXT
609 PRINT"1020 POKE59467,0:RETURN
610 PRINT"ERZEUGT LASER TONE#, WOBEI A#(1-255) DIE
612 PRINT"ANFANGSTONHOEHE IST, B=1-255 ENTSPRICHT
614 PRINT"DER ENDTONHOEHE, C=-255 BIS 255 IST DIE SCHRITTWEITE.
616 PRINT"UND D#(1-255) IST DIE ANZAHL DER ZYKLEN.#
620 A=30:B=255:C=10:D=15:GOSUB1200
625 PRINTME$;
627 PRINT"FUER A#, B#, C#, D# (Z.B. 30,255,10,15RETURN). "
630 GOSUB910:PRINTB$
635 INPUT" A#, B#, C#, D#=-          ";A$,B$,C$,D$
636 A=VAL(A$):B=VAL(B$):C=VAL(C$):D=VAL(D$)
662 IFA<10RA>255ORB<10RB>255THENGOSUB692:GOTO650
666 IFCC<-255ORC>255THENPRINT" C# MUST BE -255,255";:FORN=1TO2000:NEXT:GOTO650
670 GOSUB1200:GOSUB950
675 POKE548,1:IFA$="R"THEN670
680 IFA$="N"THEN650
685 IFA$="M"THEN120
690 GOTO670
692 PRINT" A#, B# MUSS 1-255 SEIN";:FORN=1TO2000:NEXT:RETURN
700 REM SIRENEN
702 PRINTSW$:PRINT:PRINTB2$; LO$(VAL(G$))
703 POKE59468,14

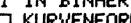
```

```

704 PRINT"DIIESE UNTERPROGRAMM :M
705 PRINT"1000 POKE59466,0:POKE59467,16:POKE59466,      15:FORN=1TO0M";
706 PRINT"FORNN=MTO0MSTEP-M
707 PRINT"1010 POKE59464,NN:NEXT:FORNN=MTO0MSTEP-M
708 PRINT"1020 POKE59464,NN:NEXT:POKE59467,0
709 PRINT"1030 RETURN
710 PRINT"ERZEUGT EINEN MIRENEN, TON WOBELN(1-255)DIE ANFANGS-
712 PRINT"TONHOEHE IST UND DIE ENDTONHOEHE DURCH
714 PRINT"B=(1-255)BESTIMMT WIRD,C=(1-255) IST DIE SCHRITTWEITE DER TONHOEHE"
716 PRINT"UND M(1-255) IST DIE ANZAHL DER ZYKLEN
720 A=255:B=30:C=2:D=4:GOSUB1300
725 PRINTM$;
727 PRINT"FOR M$,B$,C$,M$ <Z.B.255,80,2,4RETURNM$,"
750 GOSUB910:PRINTB$
755 INPUT" M$,B$,C$,M$=          "A$,B$,C$,D$
760 A=VAL(A$):B=VAL(B$):C=VAL(C$):D=VAL(D$)
762 IFAC<10RC>255ORBC<10RB>255ORC<10RC>255ORD<10RD>255THENGOSUB692:GOTO750
770 GOSUB1300:GOSUB950
775 POKE548,1:IFA$="R"THEN770
780 IFA$="N"THEN750
785 IFA$="M"THEN120
790 GOTO770
800 REM RANDOM INFO
802 PRINTSW$:PRINT:PRINTB2$; LO$(VAL(G$))
803 POKE59468,14
804 PRINT"DIIESES UNTERPROGRAMM :M
805 PRINT"1000 FORN=1TO0M:RR=INT((M$)*RND(1))+M$
806 PRINT"1010 POKE9466,0:POKE59464,RR:POKE59467,16:POKE59466,15:FORN=1TO0M
807 PRINT"1020 NEXT:POKE59467,0:NEXT:RETURN
810 PRINT"ERZEUGT ZUFALLSTOENE,WOBEI M$(1-255)
812 PRINT"DER HOECHSTE TON IST., M$(1-255)IST DER BEREICH
814 PRINT"DER TONHOEHEN, C$(1-255)IST DIE DAUER
816 PRINT"UND M$(1-255)IST DIE ANZAHL DER ZYKLEN.M
820 A=20:B=40:C=2:D=30:GOSUB1400
825 PRINTM$;
827 PRINT"FOR M$,B$,C$,M$ <Z.B.20,40,2,30RETURNM$,"
850 GOSUB910:PRINTB$
855 INPUT" M$,B$,C$,M$=          "A$,B$,C$,D$
860 A=VAL(A$):B=VAL(B$):C=VAL(C$):D=VAL(D$)
865 IF(A+B)>255THENPRINT"M+A+B MUST BE 1-255";:FORN=1TO2000:NEXT:GOTO850
870 GOSUB1400:GOSUB950
875 POKE548,1:IFA$="R"THEN870
880 IFA$="N"THEN850
885 IFA$="M"THEN120
890 GOTO870
895 RETURN
900 REM ERASE
905 FORN=1TO4000:NEXT
910 PRINTB$;:FORN=1TO4:PRINTB3$:NEXT:RETURN
950 PRINTB$"MUSGEBE R=WIEDERH N=NEU M-MENU !M":POKE548,0
955 GETA$:IFA$=""ORVAL(A$)<0>THEN955
960 RETURN
1000 REM **CLICKS** SUBROUTINE
1005 POKE59464,A:POKE59467,16:POKE59466,15:POKE59467,0:RETURN
1100 REM **TONES** SUBROUTINE
1110 POKE59466,0:POKE59464,A:POKE59467,16:POKE59466,C:FORN=1TOB:NEXT
1120 POKE59467,0:RETURN
1200 REM **LASER**
1205 POKE59466,0:POKE59467,16:POKE59466,15
1210 FORN=1TOD:FORNN=ATOBSTEP:POKE59464,NN:NEXT:NEXT
1220 POKE59467,0:RETURN

```

```

1300 REM **SIREN**
1310 POKE59466,0:POKE59467,16:POKE59466,15
1320 FORN=1TOD:FORNN=ATOBSTEP-C:POKE59464,NN:NEXT
1330 FORNN=BTOASTEP-C:POKE59464,NN:NEXT:NEXT
1340 POKE59467,0:RETURN
1400 REM **RANDOM SUBROUTINE**
1410 FORN=1TOD
1420 RR=INT((B)*RND(1)+A)
1430 POKE59466,0:POKE59464,RR:POKE59467,16:POKE59466,15
1435 FORNN=1TODC:NEXT:POKE59467,0:NEXT
1440 RETURN
6000 REM LOGO
6005 PRINT"      ERZEUGEN VON TOENEN   "
6010 PRINT"AM PORT MUSS EIN SCHALTKREIS ZUR"
6020 PRINT"MUSIKBOX ETC.
6030 PRINT"ANGESCHLOSSEN WERDEN"
6040 GOSUB 98
7000 REM
7090 RETURN
7300 PRINT"      *** TONERZEUGUNGS-THEORIE F. CBM ***"
7303 A=50:GOSUB1000
7305 PRINT"CB-2 IST EIN 8 BIT SCHIEBEREGISTER UND"
7307 PRINT"KANN VON BASIC UEBER POKE PROGRAMMIERT"
7310 PRINT"WERDENPOKE59464,N(N=1-255)BEST.DIE SCHIEBEREGISTERFREQUENZ"
7320 PRINT"POKE59467,16 BRINGT CB2 ZUM LEERLAUF
7330 PRINT"POKE59466,15 BESTIMMT DIE CB2 KURVE"
7335 PRINT"      15= 00001111 IN BINAER
7337 PRINT"      UND 15=  KURVENFORM
7338 PRINT"      51= 00110011 IN BINAER
7339 PRINT"      UND 51=  KURVENFORM
7340 PRINT"POKE59467,0 =CB2 AUS
7350 PRINT"***!!!!* CB2 MUSS AUSGESCHALTET SEIN"
7360 PRINT"      BEVOR DER CASSETTENRECORDER VERWENDET WERDEN SOLL.
7370 PRINT"GEBEN SIE RETURN UM INS MENUE ZU GEHEN"
7380 GETG$:IFG$=""THEN7380
7390 GOTO120
READY.

```

# Bildschirmausdruck

## Bildschirmausdruck

Dieses Unterprogramm druckt Buchstaben und Zahlen auf dem Bildschirm auf dem Drucker aus.

```
100 OPEN 5,5:REM UNLESS ALREADY OPEN
50000 FOR NY=0 TO 24
50010 FOR NX=0 TO 39
50020 NS=PEEK(32768+NY*40+NX)
50030 NS#=CHR$(NS)
50040 IF NS>=0 AND NS<32 THEN NS#=CHR$(
NS+64)
50050 IF NS>63 AND NS<96 THEN NS#=CHR$(
NS+32)
50060 PRINT#4, NS#;
50070 NEXT NX
50080 PRINT#4:FOR NV=0 TO 5:PRINT#4,CHR
$(0);:NEXT NV:NEXT NY
READY.
```

## NOTIZEN

# Super Drei

## Super Drei

Super Drei ist ein Denk-Spiel bei dem Sie die Kästchen in einer ganz bestimmten Reihenfolge anordnen müssen. Die Spielregeln sind

aufgeführt im 1. Programm des Programmsystems, welches insgesamt aus 2 Teilen besteht, enthalten.

```
230 GOSUB1370
240 PRINT"*****
250 PRINT"***** #1. DAS SPIELFELD"
260 PRINT"***** DAS SPIELFELD BESTEHT AUS 9 KAESTCHEN"
270 PRINT"***** MIT JE 9 POSITIONEN."
280 PRINT"***** DIE ZAHL JEDES KAESTCHENS STEHT IN"
290 PRINT"***** REVERSER SCHREIBWEISE IN DER MITTE DES"
300 PRINT"***** JEWEIFIGEN KAESTCHEN, DIE VON 3-11
310 PRINT"***** POSITIONEN JEDES KAESTCHENS SIND
320 PRINT"***** EBENFALLS VON 3 BIS 11 NUMMERIERT.
330 PRINT"***** SIE SIND INNERHALB DES KAESTCHENS"
340 PRINT"***** FORTLAUFEND NUMMERIERT, MIT AUSNAHME DER"
350 PRINT"***** MITTELPOSITION, DIE DIE ZAHL DES"
360 GOSUB1340:GOSUB1370
370 PRINT"***** #2. ZIEL DES SPIELES"
380 PRINT"***** ZIEL DES SPIELES IST ES, ENTWEDER DREI "
390 PRINT"***** KAESTCHEN IN EINER REIHE, EGAL OB "
400 PRINT"***** WAAGERECHT, SENKRECHT ODER DIAGONAL."
410 PRINT"***** ODER FUEF BELIEBIGE KAESTCHEN ZU
420 PRINT"***** DIE KAESTCHEN WERDEN EBENFALLS ENTWEDER"
430 PRINT"***** DURCH BESETZEN VON DREI POSITIONEN IN"
440 PRINT"***** EINER REIHE - WAAGERECHT, SENKRECHT ODER"
450 PRINT"***** DIAGONAL - ODER VON FUEF BELIEBIGEN"
460 PRINT"***** POSITIONEN INNERHALB EINES KAESTCHENS
470 GOSUB1340:GOSUB1370
480 PRINT"***** #3. SPIELREGELN"
490 PRINT"***** DIE BEIDEN SPIELER LEGEN FEST, WER"
500 PRINT"***** SPIELER 1 (IHM GEHOERT WAERHEND DES"
510 PRINT"***** SPIELS DAS ZEICHEN *) UND WER SPIELER 2 (MIT DEM ZEICHEN %) IST.
520 PRINT"***** NACHDEM DER COMPUTER GEWUERFEL HAT."
530 PRINT"***** ADDIERT DER SPIELER DIE BEIDEN AUGEN -"
540 PRINT"***** ZAHLEN UND DARF NUN EINE BELIEBIGE"
550 PRINT"***** POSITION MIT DIESER ZAHL IRGENDWO AUF"
560 PRINT"***** DEM SPIELFELD ODER EINE BELIEBIGE"
570 PRINT"***** POSITION IN DEM KAESTCHEN MIT DIESER"
580 PRINT"***** ZAHL, MIT AUSNAHME DER MITTELPOSITION,"
590 PRINT"***** BESETZEN."
600 GOSUB1340:GOSUB1370
610 PRINT"***** BEI EINEM WURF MIT EINER 4 UND EINER 6"
620 PRINT"***** ZUM BEISPIEL, ALSO EINER 10, DARF JETZT"
630 PRINT"***** EINE BELIEBIGE 10-ER POSITION (JEDES"
640 PRINT"***** KAESTCHEN HAT EINE DAVON) ODER EINE"
650 PRINT"***** BELIEBIGE POSITION IM KAESTCHEN MIT DER"
660 PRINT"***** NUMMER 10 BELEGT WERDEN, MIT AUSNAHME"
670 PRINT"***** DER MITTELPOSITION."
680 PRINT"***** DIE EINGABE ERFOLGT IN FOLGENDER FORM:"
```





```

440 INPUT "#####I          I#####"; I, J
450 IFZ=2THEN1270
460 IFZ=7THEN490
470 IFZ=12THEN500
480 IFJ=7THEN1260
490 IFI<>ZANDJ<>ZTHEN1260
500 IFI<30RJ<3THEN1260
510 IFI>110RJ>11THEN1260
520 IFT%(I, J)<>0THEN1260
530 PRINT "#####"; ONI-2GOTO540, 670, 680, 690, 700, 710, 720, 730, 740
540 ONJ-2GOTO640, 550, 560, 570, 750, 580, 590, 600, 610
550 PRINTSPC(4); :GOTO640
560 PRINTSPC(8); :GOTO640
570 PRINT "###"; :GOTO640
580 PRINT "###"; SPC(8); :GOTO640
590 PRINT "#####"; :GOTO640
600 PRINT "#####"; SPC(3); :GOTO620
610 PRINT "#####"; SPC(7);
620 IFA=1THENPRINT "###"; :GOTO660
630 PRINT "###"; :GOTO660
640 IFA=1THENPRINT "###"; :GOTO660
650 PRINT "###"
660 TZ(I, J)=A+4; :GOTO790
670 PRINTSPC(13); :GOTO540
680 PRINTSPC(26); :GOTO540
690 PRINT "#####"; :GOTO540
700 PRINT "#####"; SPC(13); :GOTO540
710 PRINT "#####"; SPC(26); :GOTO540
720 PRINT "#####"; :GOTO540
730 PRINT "#####"; SPC(13); :GOTO540
740 PRINT "#####"; SPC(26); :GOTO540
750 PRINT "#####"; : IFI>9ANDA=1THENPRINT "###"; :GOTO660
760 IFI>9THENPRINT "###"; :GOTO660
770 IFA=1THENPRINT "###"; :GOTO660
780 PRINT "###"; :GOTO660
790 FORR=3TO11 : IFT%(I, R)=5THENL=L+1
800 NEXT : IFL=5THENL=0; :GOTO940
810 L=0; FORR=3TO11 : IFT%(I, R)=6THENL=L+1
820 NEXT : IFL=5THENL=0; :GOTO940
830 L=0; C=3
840 FORR=CTOC+2; Q=Q+TZ(I, R); NEXT; :GOSUB910
850 C=C+3; IF C=12 THENC=2; :GOTO870
860 GOTO840
870 C=C+1; IF C=6THEN890
880 FORR=CTOC+6STEP3; Q=Q+TZ(I, R); NEXT; :GOSUB910; :GOTO870
890 FORR=3TO11STEP4; Q=Q+TZ(I, R); NEXT; :GOSUB910
900 FORR=5TO9STEP2; Q=Q+TZ(I, R); NEXT; :GOSUB910; :GOTO1210
910 IFQ=150RQ=18THEN940
920 Q=0
930 RETURN
940 PRINT "###";
950 ONI-2GOTO1040, 960, 970, 980, 990, 1000, 1010, 1020, 1030
960 PRINTSPC(13); :GOTO1040
970 PRINTSPC(26); :GOTO1040
980 PRINT "#####"; :GOTO1040
990 PRINT "#####"; SPC(13); :GOTO1040
1000 PRINT "#####"; SPC(26); :GOTO1040
1010 PRINT "#####"; :GOTO1040
1020 PRINT "#####"; SPC(13); :GOTO1040
1030 PRINT "#####"; SPC(26);
1040 IFA=2THEN1070

```

```

1050 FORR=1T05:F0RV=1T012:PRINT"8";:NEXTV:PRINT"#####00";:NEXTR
1060 T%(I,2)=5:FORR=3T011:T%(I,R)=A+4:NEXT:GOTO1110
1070 FORR=1T05:F0RV=1T012:PRINT"5";:NEXTV:PRINT"#####00";:NEXTR
1080 T%(I,2)=6:FORR=3T011:T%(I,R)=A+4:NEXT:FORR=3T011:IFT%(R,2)=6THENQ=Q+1
1090 NEXT:IFQ=5THEN1690
1100 GOTO1130
1110 FORR=3T011:IFT%(R,2)=5THENQ=Q+1
1120 NEXT:IFQ=5THEN1690
1130 Q=3:V=0
1140 FORR=QTOQ+2:V=V+T%(R,2):NEXT:GOSUB1240
1150 Q=Q+3:IFQ=12THENQ=2:GOTO1170
1160 GOTO1140
1170 Q=Q+1:IFQ=6THEN1190
1180 FORR=QTOQ+6STEP3:V=V+T%(R,2):NEXT:GOSUB1240:GOTO1170
1190 FORR=3T011STEP4:V=V+T%(R,2):NEXT:GOSUB1240
1200 FORR=5T09STEP2:V=V+T%(R,2):NEXT:GOSUB1240
1210 IFZ=2THEN210
1220 A=A+1:IFA=3THENA=1
1230 PRINT"11":GOTO210
1240 IFV=150RV=18THEN1690
1250 V=0:RETURN
1260 PRINTSPC(13)"REGELVERSTOSS#####1111":GOTO430
1270 IFI<30RJ<30RI>110RJ>11THEN1260
1280 IFT%(I,2)<>0THEN1260
1290 IFT%(I,J)=0THEN1260
1300 IFA=1ANDT%(I,J)=5THEN1260
1310 IFA=2ANDT%(I,J)=6THEN1260
1320 PRINT"3001";
1330 ONI-2GOTO1340,1430,1440,1450,1460,1470,1480,1490,1500
1340 ONJ-2GOTO1350,1360,1370,1380,1590,1390,1400,1410,1420
1350 PRINT"3";:GOTO1510
1360 PRINTSPC(4)"4";:GOTO1510
1370 PRINTSPC(8)"5";:GOTO1510
1380 PRINT"005";:GOTO1510
1390 PRINT"001";SPC(8)"8";:GOTO1510
1400 PRINT"00009";:GOTO1510
1410 PRINT"000000010";:GOTO1510
1420 PRINT"00000";SPC(7)"11";:GOTO1510
1430 PRINTSPC(13);:GOTO1340
1440 PRINTSPC(26);:GOTO1340
1450 PRINT"0000000";:GOTO1340
1460 PRINT"0000000";SPC(13);:GOTO1340
1470 PRINT"0000000";SPC(26);:GOTO1340
1480 PRINT"000000000000";:GOTO1340
1490 PRINT"0000000000000";SPC(13);:GOTO1340
1500 PRINT"0000000000000";SPC(26);:GOTO1340
1510 T%(I,J)=0:GOTO210
1520 IFA=1THEND=6:GOTO1540
1530 D=5
1540 FORR=3T011:F0RV=3T011:IFT%(R,2)=50RT%(R,2)=6THEN1570
1550 IFT%(R,V)=DTHENM=0:GOTO340
1560 NEXT
1570 NEXTR:M=M+1:IFM=3THEN1210
1580 GOTO210
1590 PRINT"3001";:ONI-2GOTO1600,1610,1620,1630,1640,1650,1660,1670,1680
1600 PRINT" (<300>";:GOTO1510
1610 PRINT" (<340>";:GOTO1510
1620 PRINT" (<350>";:GOTO1510
1630 PRINT" (<360>";:GOTO1510
1640 PRINT" (<370>";:GOTO1510
1650 PRINT" (<380>";:GOTO1510

```

```

1660 PRINT" (29)";:GOTO1510
1670 PRINT" (310)";:GOTO1510
1680 PRINT" (311)";:GOTO1510
1690 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXX";:PRINT"
1700 PRINT"DAMIT HAT SPIELER";A;"GEWONNEN"
1710 PRINT"
1720 PRINT"HERZLICHEN GLUECKWUNSCH !
1730 PRINT"
1740 GOTO1740
READY.
"
"
"
"

```





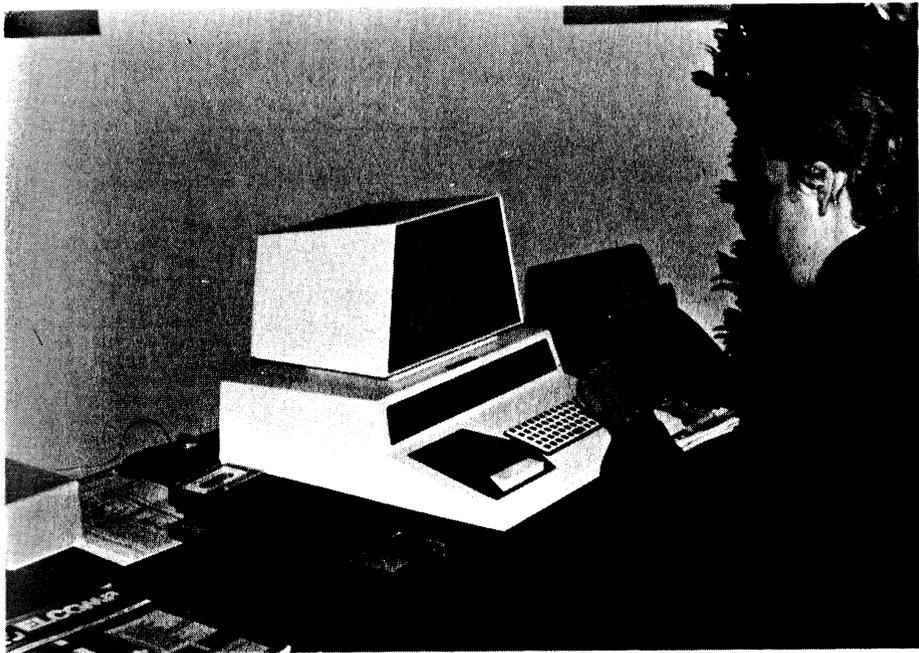
```

3110 REM *** ZAEHLER AUSWERTEN ***
3120 IF W1=12 THEN A=1:GOTO 4000
3130 IF W2=12 THEN A=2:GOTO 4000
3140 W1=0:REM *** ZAEHLER FUER DIE
3150 W2=0:REM *** BESETZTEN FELDER
3160 RETURN
4000 REM ***
4010 REM *** EIN SPIELER HAT GEWONNEN
4020 REM ***
4030 PRINT"XDAMIT HAT"
4040 PRINT"SPIELER"A"GEWONNEN"
4050 REM *** NEUEN SPIELSTAND ANZEIGEN
4060 IF W1=12 THEN S1=S1+1:GOTO 4080
4070 IF W2=12 THEN S2=S2+1
4080 PRINT"XNEUER SPIELSTAND:"
4090 PRINT" "S1" : "S2
4100 W1=0
4110 W2=0
4120 Y=0
4130 F1=1:REM *** SIEG-FLAG SETZEN ***
4140 RETURN
5000 REM ***
5010 REM *** SPIELSCHLUSS
5020 REM ***
5030 PRINT"XNOCH EIN SPIEL ?"
5040 GET A$: IF A$="" THEN 5040
5050 IF A$="J" THEN 180
5060 IF A$<>"N" THEN 5040
5070 GOSUB 7000:REM * KOPF ZEICHNEN *
5080 PRINT"SCHADE..."
5090 PRINT"XBIS ZUM NAECHSTEN MAL..."
5100 END
6000 REM ***
6010 REM *** SPIELREGELN
6020 REM ***
6030 GOSUB 7000:REM ** KOPF ZEICHNEN **
6040 PRINT"KARUSSELL IST EIN STRATEGIESPIEL FUER"
6050 PRINT"ZWEI PERSONEN."
6060 PRINT
6070 PRINT"JEDER SPIELER ERHAELT EINE HAELFTE DES"
6080 PRINT"SPIELFELDES MIT 12 FELDERN. SIEGER IST,"
6090 PRINT"WER SEINE 12 FELDER ZUERST BESETZT HAT."
6100 PRINT
6110 PRINT"BEIM BESETZEN DER FELDER GILT FOLGENDE"
6120 PRINT"REGEL:"
6130 PRINT"NACHDEM GEWUERFELT WURDE, SUCHT SICH DER"
6140 PRINT"SPIELER EINE DER NUMMERIERTEN POSITIONEN"
6150 PRINT"RAUS. VON DER AUS DANN SEIN ZUG"
6160 PRINT"AUSGEFUEHRT WIRD (DIESE POSITION MUSS"
6170 PRINT"ABER IN SEINER HAELFTE LIEGEN !). "
6180 GOSUB 8000:REM * SEITE UMBLAETTERN
6190 GOSUB 7000:REM * KOPF ZEICHNEN
6200 PRINT"VON DER GEWAELHTEN POSITION AUS WERDEN"
6210 PRINT"DANN IN PFEILRICHTUNG UND VON AUSSEN"
6220 PRINT"NACH INNEN SO VIELE FELDER BELEGT, WIE"
6230 PRINT"AUGEN GEWUERFELT WURDEN."
6240 PRINT
6250 PRINT"ABER VORSICHT ! SIND ALLE FELDER EINER"
6260 PRINT"POSITION BELEGT, SO WERDEN SIE BEIM"
6270 PRINT"NAECHSTEN ZUG ALLE WIEDER GELEERT..."
6280 PRINT"XUND NUN VIEL SPASS !!"

```



```
11210 IF I=T THEN 11270
11220 PRINT"X001" "
11230 PRINT"X001" "
11240 PRINT"X001" "
11250 PRINT"X01" "
11260 NEXT I
11270 RETURN
12000 REM *** BILDSCHIRMADRESSE
12010 REM *** DER FELDER
12020 REM ***
12030 DATA396,434,472,272,351,430,266,347,428,382,424,466
12040 DATA622,584,546,746,667,588,752,671,590,636,594,552
READY.
```



**"alter" PET mit kleiner Taschenrechnerähnlichen Tastatur**

## NOTIZEN

# Lotto

## LOTTO

Dieses kleine Programm läuft auf Ihrem CBM oder PET Computer. Geben Sie einfach RUN ein

und Sie erhalten einen Vorschlag, wie Sie Ihren nächsten Lottozettel ausfüllen können. Selbstverständlich ohne Gewähr.

```
1 T0$="| | | | | | | |
2 T1$="|
3 T2$="| 01 | 02 | 03 | 04 | 05 | 06 | 07 |
4 T3$="| 08 | 09 | 10 | 11 | 12 | 13 | 14 |
5 T4$="| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
6 T5$="| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
7 T6$="| 29 | 30 | 31 | 32 | 33 | 34 | 35 |
8 T7$="| 36 | 37 | 38 | 39 | 40 | 41 | 42 |
9 T8$="| 43 | 44 | 45 | 46 | 47 | 48 | 49 |
10 T9$="|
11 S$=" "
12 PRINT"□
13 PRINT"☐"S$+T1$
14 PRINTS$+T2$
15 PRINTS$+T0$
16 PRINTS$+T1$
17 PRINTS$+T3$
18 PRINTS$+T0$
19 PRINTS$+T1$
20 PRINTS$+T4$
21 PRINTS$+T0$
22 PRINTS$+T1$
23 PRINTS$+T5$
24 PRINTS$+T0$
25 PRINTS$+T1$
26 PRINTS$+T6$
27 PRINTS$+T0$
28 PRINTS$+T1$
29 PRINTS$+T7$
30 PRINTS$+T0$
31 PRINTS$+T1$
32 PRINTS$+T8$
33 PRINTS$+T9$
34 PRINT"☐ TO CONTINUE PRESS 'N' (=NEW)
35 PRINT" TO FINISH PRESS 'E' (=END)
36 X(0)=99
37 FORZ=1TO6
38 W=0
39 X(Z)=INT(RND(1)*49)
40 FORV=0TOZ-1
41 IFX(Z)=X(V)THENW=1
42 NEXTV
43 IFW=1THEN38
44 XV=120*INT(X(Z)/7)
45 XH=4*(X(Z)-7*INT(X(Z)/7))
46 XX=32774+XV+XH
```

```
47 POKEXX,160
48 POKEXX+1,160
49 POKEXX+1,160
50 POKEXX+2,160
51 POKEXX+3,160
52 POKEXX+40,160
53 POKEXX+43,160
54 POKEXX+80,160
55 POKEXX+81,160
56 POKEXX+82,160
57 POKEXX+83,160
58 D=PEEK(XX+41)
59 POKEXX+41,D+128
60 D=PEEK(XX+42)
61 POKEXX+42,D+128
62 NEXTZ
63 GETA$: IFA$="N" THEN 13
64 IFA$="E" THEN 66
65 GOT063
66 PRINT"IN CASE OF SUCCESS IN THE LOTTERY
67 PRINT"BY USING THIS PROGRAM, PLEASE, DON'T
68 PRINT"FORGET THE POOR AUTHOR:
69 PRINT"XXXXXXXXXXXXXXXXXG.H. BAUER
70 PRINT"XXXXXXXXXXXXXXXXXBG.M.SCHWAIGERSTR. 18
71 PRINT"XXXXXXXXXXXXXXXXX018 GRAFING
72 PRINT"XXXXXXXXXXXXXXXXXGOOD LUCK!XXXXXXXX
73 END
READY.
```



```

205 PRINT"#####"
206 PRINTN$(1);" HAT DIE TASTEN"
207 PRINT"X'A', 'X', 'D' UND 'W'"
208 PRINT"X":FORI=1TO40:PRINT"=";:NEXT
209 PRINT"###N$(2);" HAT DIE TASTEN"
210 PRINT"X'4', '2', '6' UND '8'"
211 GOSUB8000:GOSUB8000
300 PRINT"?"
310 PRINT"MIT WELCHER GESCHWINDIGKEIT WOLLEN SIE"
320 PRINTTAB(15)"###BLOCKADE###"
330 PRINT"XSPIELEN?"
340 PRINT"XGEBEN SIE DIE ENTSPRECHENDE ZAHL EIN:":
345 FORU=1TO10:GETA$:NEXTU
350 PRINT"X(1) ANFAENGER"
360 PRINT"X(2) FORTGESCHRITTENE"
370 PRINT"X(3) MEISTER"
380 PRINT"X(4) BLOCKADE-STARS"
390 GETA$:IFA$=""THEN390
395 IFASC(A$)<49ORASC(A$)>52THENPRINT"BITTE EINE ZAHL VON 1 BIS 4 !":GOTO390
400 G=10+(4-VAL(A$))*8
900 REM
1000 RU=0
1031 S1=0:S2=0
1040 ZZ=127:GOSUB5000
1045 PRINT"###N$(1);TAB(39-LEN(N$(2)))N$(2)"
1050 RU=RU+1
1100 P1=33258:P2=33277
1103 I1=P(INT(RND(1)*4+1))
1104 I2=P(INT(RND(1)*4+1))
1200 FORV=1TOG:GETA$
1300 IFA$="A"THENI1=-1
1400 IFA$="X"THENI1=40
1500 IFA$="D"THENI1=1
1600 IFA$="W"THENI1=-40
1601 IFA$="4"THENI2=-1
1602 IFA$="2"THENI2=40
1603 IFA$="6"THENI2=1
1604 IFA$="8"THENI2=-40
1650 NEXTV
1700 P1=P1+INT(I1)
1800 IF PEEK(P1)<>32 THEN F=1:PP=P1-I1:S2=S2+1:GOTO6000
1810 POKE P1,160
2300 P2=P2+INT(I2)
2400 IF PEEK(P2)<>32 THEN F=2:PP=P2-I2:S1=S1+1:GOTO6000
2450 POKE P2,102
2600 GOTO1200
5000 PRINT"?"
5010 POKE59409,52
5020 FOR I=32768 TO 32807
5030 POKE I,Z%:POKE I+960,Z%
5040 NEXT
5050 FOR I=32768 TO 33728 STEP 40
5060 POKE I,Z%:POKE I+39,Z%
5070 NEXT
5080 POKE59409,60
5090 RETURN
6000 FORI=1TO10:FORJ=1TO100:NEXT:POKEPP,63:FORJ=1TO100:NEXT:POKEPP,32:NEXTI
6001 Z%=42:GOSUB5000
6010 PRINT"#####FEHLER DURCH ";N$(F)
6015 PRINT"###":FORI=1TOLEN(N$(F))+13:PRINT"*":NEXT:GOSUB8000
6019 IFS1=7ORS2=7THEN6090

```





# CRASH

## CRASH

Das Spiel CRASH simuliert ein aufregendes Autorennen gegen den Computer. Sowohl Sie, als auch der Computer sitzen am Steuer eines Rennwagens. Der Computer fährt mit dem grauen Fahrzeug (gerastertes Muster). Sie sitzen im "weißen" bzw. "grünen" Auto. Die Rennbahn besteht aus ineinandergeschachtelten Fahr-

bahnen. Auf jeder Rennbahnseite sind die Fahrbahnen unterbrochen. An diesen Stellen können Sie die Spuren durch Drücken bestimmter Tasten wechseln. Ihre Aufgabe ist es einen Zusammenstoß mit dem vom Computergesteuerten Auto zu verhindern. Eine genaue Anleitung gibt Ihnen das laufende Programm.

```
18 REM
19 POKE219,PEEK(59464):POKE220,PEEK(59465):POKE221,PEEK(514):POKE222,PEEK(513)
20 PRINT" "
21 PRINT"      WILLKOMMEN ZUM COMPUTERSPIEL"
25 PRINTTAB(10)" "
26 PRINTTAB(10)" | C R A S H | "
27 PRINTTAB(10)" | "
30 FORI=1TO3000:NEXT
40 PRINT" "KENNEN SIE DAS SPIEL BEREITS ? (J/N)"
50 GETA$:IFA$=" "THEN50
60 IFA$="N"THEN500
70 IFA$="J"THEN900
80 PRINT" | BITTE TASTE 'J' ODER 'N' DRUECKEN ! ":GOTO50
500 PRINT" | "
510 PRINT" | "
520 PRINT" | S P I E L - A N L E I T U N G | "
530 PRINT" | "
540 PRINT" | DAS SPIEL CRASH IST EIN AUTORENNEN "
550 PRINT" | GEGEN DEN COMPUTER. "
560 PRINT" | SOWOHL SIE, ALS AUCH DER COMPUTER. "
570 PRINT" | LENKEN EIN AUTO. "
580 PRINT" | SIE HABEN DAS WEISSE FAHRZEUG ( ) "
590 PRINT" | DER COMPUTER HAT DAS GRAUE FAHRZEUG (■) "
600 GOSUB6000
610 PRINT" | DAS SPIELFELD BESTEHT AUS VIER - INEIN- "
620 PRINT" | ANDERGESCHACHELTEN - FAHRBAHNEN. "
630 PRINT" | AUF JEDER SPIELFELDSEITE SIND DIE "
640 PRINT" | FAHRSPUREN UNTERBROCHEN - HIER KOENNEN "
650 PRINT" | SIE DIE FAHRSPUREN, DURCH DRUECKEN BE- "
660 PRINT" | STIMMTER TASTEN WECHSELN. "
670 PRINT" | IHRE AUFGABE IST ES, EINEN ZUSAMMENSTOSS "
680 PRINT" | MIT DEM COMPUTER-AUTO ZU VERHINDERN. "
690 GOSUB6000
700 PRINT" | HIER SIND DIE TASTEN, MIT DENEN SIE IHR "
705 PRINT" | AUTO LENKEN KOENNEN: "
710 PRINTTAB(16)" | "
711 PRINTTAB(16)" | 7 | 8 | 9 | "
712 PRINTTAB(16)" | | | | "
713 PRINTTAB(16)" | 4 | 5 | 6 | "
714 PRINTTAB(16)" | | | | "
715 PRINTTAB(16)" | 1 | 2 | 3 | "
716 PRINTTAB(16)" | | | | "
```

```

720 PRINT,"TASTE","LENKUNG NACH"
721 PRINT,"-----","-----X"
722 PRINT,"8","OBEN"
723 PRINT,"4","LINKS"
724 PRINT,"2","UNTEN"
725 PRINT,"6","RECHTSX"
726 GOSUB6000
727 PRINT"DRUECKEN SIE EINE TASTE EINMAL, SO"
728 PRINT"WECHSELN SIE EINE FAHRSPUR IN DIE ENT-"
729 PRINT"SPRECHENDE RICHTUNG, DIE DURCH DIE GE-"
730 PRINT"DRUECKTE TASTE BESTIMMT IST."
731 PRINT"WIRD EINE STEuertASTE ZWEIMAL GEDRUECKT,"
732 PRINT"SO WERDEN ZWEI FAHRSPUREN GEWECHSELT.XXXXXXX"
733 GOSUB6000
740 PRINT"NAUF DEN FAHRSPUREN BEFINDEN SICH "
741 PRINT"MARKIERUNGSPUNKTE, DIE DURCH DAS UEBER-"
742 PRINT"FAHREN MIT IHREM FAHRZEUG GELOESCHT"
743 PRINT"WERDEN."
745 PRINT"ODAS SPIEL IST ZU ENDE, WENN SIE ALLE"
746 PRINT"UNKTE GELOESCHT HABEN ODER ABER DIE"
747 PRINT"SPIELZEIT ABGELAUFEN IST. SIE HABEN"
748 PRINT"1000 ZEITEINHEITEN ZUR VERFUEGUNG.XXXXXX"
749 GOSUB6000
750 PRINT"WENN SIE JETZT MIT DEM SPIEL BEGINNEN"
751 PRINT"WOLLEN, SO DRUECKEN SIE BITTE DIE"
752 PRINT"TASTE SPACE"
753 PRINT"WENN SIE DIE SPIEL-ANLEITUNG NOCH"
754 PRINT"EMAL LESEN WOLLEN, SO DRUECKEN SIE DIE"
755 PRINT"TASTE RETURN"
756 GETA$:IFA$=""THEN756
757 IFA$<>"THEN500
758 PRINT"UND NUN WUENSCH E ICH IHNEN"
759 PRINTTAB(12)"WIEVIEL VERGNUEGEN !"
800 FORI=1TO3000:NEXT
900 CLR
905 DEFFNM(X)=X-INT(X/4)*4
910 DEFFNR(X)=INT(RND(1)*X)
920 RESTORE
1000 PRINT"XXXXXXXXXXXX":A=32769:B=33729:Z=36:V=4:POKE59459,0
1001 FORI=1TO4:FORJ=0TOZ:POKEA+J,64:POKEB+J,64:POKEA+40+J,46:POKEB-40+J,46:NEXT
J
1002 Z=Z-4:A=A+82:B=B-78:NEXTI:FORJ=0TOZ:POKEA+J,64:POKEB+J,64:NEXTJ
1003 A=32808:B=32846:Z=22:FORI=1TO4:FORJ=0TOZ:POKEA+J*40,93:POKEB+J*40,93
1004 POKEA+J*40+1,46:POKEB+J*40-1,46:NEXTJ
1005 Z=Z-4:A=A+82:B=B+78:NEXTI:FORJ=0TOZ:POKEA+J*40,93:POKEB+J*40,93:NEXTJ
1006 A=32768:B=33728:Z=38
1007 FORI=1TO5:POKEA,112:POKEA+Z,110:POKEB,109:POKEB+Z,125:A=A+82:B=B-78
1008 Z=Z-4:NEXTI
1009 A=32866:B=33666:FORI=1TO3:FORJ=0TO2:POKEA+J,32:POKEB+J,32:NEXTJ
1010 A=A+80:B=B-80:NEXTI
1011 A=33210:B=33244:FORI=1TO3:FORJ=0TO2:POKEA+J*40,32:POKEB+J*40,32:NEXTJ
1012 A=A+2:B=B-2:NEXTI
1015 PRINTTAB(13)"ZEIT:"
1020 DIMR(3),K$(3):FORI=0TO3:READR(I),K$(I):NEXTI
1030 DATA 1,8,-40,4,-1,2,40,6
1040 DIMW(3,3):FORI=0TO3:FORJ=0TO3:READW(I,J):NEXTJ,I
1050 DATA 33707,33285,32827,33249,33627,33283,32907,33251
1060 DATA 33547,33281,32987,33253,33467,33279,33067,33255
1064 WW=46:PU=0:T=999
1065 PC=33175:RC=2:SC=2:TC=3:PP=33709:RP=0:SP=1:TP=0:V=3:Y=1:P1=33709:YY=1
1066 VW=3:L=3

```

```

1070 IFPP=W(TP, SP) THENGOSUB2000
1075 IFV<3THENPP=PP+R(VP):V=V+1:GOTO1100
1080 X=PEEK(PP+R(RP)):IFX=64ORX=93THENRP=FNM(RP+1)
1085 PP=PP+R(RP)
1100 POKEP1,32:IFPEEK(PP)=46THENPU=PU+1
1101 POKEPP,160:P1=PP:IFPU=367THEN5000
1199 IFPEEK(PC)=160THEN4000
1200 IFPC=W(TC, SC) THENGOSUB3000
1210 IFVV<LTHENPC=PC+R(VC):VV=VV+1:GOTO1300
1220 X=PEEK(PC+R(RC)):IFX=64ORX=93THENRC=FNM(RC-1)
1230 PC=PC+R(RC)
1300 POKEP2,WW:WW=PEEK(PC):POKEPC,102:P2=PC
1320 PRINTTAB(19)"T" "T" " " :T=T-1:IFT>0THEN1070
1400 GOTO5000
2000 GETA$:Y=Y+1:IFY=4THEN2030
2010 IFA$=K$(SP)ANDTP<3THENTP=TP+1:V=1:VP=FNM(RP+1):RETURN
2020 IFA$=K$(FNM(SP+2))ANDTP>0THENTP=TP-1:V=1:VP=FNM(RP-1):RETURN
2030 GETA$:GETA$:Y=1:SP=FNM(SP+1):RETURN
3000 VY=VY+1:IFY=30RFNR(4)=0THEN3030
3010 IFTP<T>1ANDFNR(3)>0THENTC=TC+2:L=5:VV=1:VC=FNM(RC-1):RETURN
3020 IFTC<T>1ANDFNR(3)>0THENTC=TC-2:L=5:VV=1:VC=FNM(RC+1):RETURN
3022 IFTP>TCANDFNR(2)=0THENTC=TC+1:L=3:VV=1:VC=FNM(RC-1):RETURN
3024 IFTC>TPANDFNR(2)=0THENTC=TC-1:L=3:VV=1:VC=FNM(RC+1):RETURN
3026 IFTC>0ANDFNR(2)=0THENTC=TC-1:L=3:VV=1:VC=FNM(RC+1):RETURN
3028 IFTC<3ANDFNR(2)=0THENTC=TC+1:L=3:VV=1:VC=FNM(RC-1):RETURN
3030 VY=1:SC=FNM(SC-1):RETURN
4000 CR=CR+1
4010 FORI=1TO5:FORJ=1TO200:NEXTJ:PRINTTAB(13)" " CRASH!!!":POKEPC,32
4020 FORJ=1TO200:NEXTJ:PRINTTAB(16)" " " :POKEPC,42:NEXTI
4030 POKEPC,32:POKEPP,32
4035 IFWW=46THENPOKEPC,46
4040 PRINTTAB(13)" "ZEIT":GOTO1065
5000 PRINTTAB(13)" "SPIEL ZU ENDE"
5010 FORI=1TO3000:NEXTI
5020 PRINT" "
5025 PRINT" "
5030 PRINT" " S P I E L - E R G E B N I S "
5035 PRINT" " :IFT=0THEN5100
5040 PRINT" "GRATULIERE, SIE HABEN ALLE MARKIERUNGEN"
5050 PRINT"GELOESCHT"
5060 PRINT" "ANZAHL DER ZUSAMMENSTOESSE:" ",CR
5070 PRINT" "BENOETIGTE ZEIT:" ",1000-T:GOTO5200
5100 PRINTTAB(6)" "DIE SPIELZEIT IST ABGELAUFEN"
5110 PRINT" "ANZAHL DER ZUSAMMENSTOESSE:" ",CR
5120 PRINT" "ANZAHL GELOESCHTER PUNKTE:" ",PU
5200 PRINT" "NOLLEN SIE NOCH EINMAL SPIELEN ? (J / N)"
5210 GETA$:IFA$=""THEN5210
5220 IFA$="J"THEN900
5230 IFA$<>"N"THENPRINT"BITTE 'J' ODER 'N' DRUECKEN !":GOTO5210
5300 PRINT" "AUF WIEDERSEHEN"BEIM SPIEL"CRASH" :END
6000 PRINT" "ZUM WEITERBLATTTERN EINE TASTE DRUECKEN"
6001 GETA$:IFA$=""THEN6001
6002 RETURN

```

READY.

## NOTIZEN

# String-Sort

## STRING SORT

Dieses Utility Programm sortiert Zeichenketten in alphabetischer Reihenfolge. Auf- oder ab-

steigend. Mit diesem Programm lassen sich maximal 255 Strings sortieren.

```
2000 REM
2010 GOTO4000
2020 PRINT" ":PRINTTAB(8)"|_____|"
2030 PRINTTAB(8)"| S T R I N G - S O R T |"
2040 PRINTTAB(8)"|_____|" :RETURN
2050 PRINT"XXXXXXXXXDIESES PROGRAMM SORTIERT 255 ZEICHENKETTEN"
2060 PRINT"XXXXXXXXXALPHABETISCH AUF- ODER ABSTIEGEND."
2070 PRINT"XXXXXXXXXMAXIMAL KOENNEN MIT DIESEM PROGRAMM"
2080 PRINT"XXXXXXXXX255 STRINGS SORTIERT WERDEN."
2130 FORI=1TO6000:NEXTI:RETURN
4000 GOSUB2020:FORI=1TO10:GETA$:NEXTI
4010 PRINTTAB(5)"XXXXXXXXXKENNEN SIE DAS PROGRAMM BEREITS?"
4020 PRINTTAB(16)"X(J / N)
4030 GETA$: IFA$="" THEN4030
4040 IFA$="N" THENGOSUB2020:GOSUB2050:GOTO5000
4050 IFA$="J" THEN5000
4060 PRINT"XBITTE NUR 'J' FUER 'JA' ODER"
4070 PRINT"X'N' FUER 'NEIN' EINGEBEN!":GOTO4030
5000 PRINT" ":GOSUB2020
5010 DIMA$(255):N=0
5020 PRINT"XBITTE MACHEN SIE JETZT IHRE EINGABEN:"
5030 PRINT"XWENN SIE FERTIG SIND, GEBEN SIE BITTE"
5040 PRINT"XDAS WORT 'ENDE' EIN.XX"
5050 N=N+1:PRINTN,:INPUTX$
5060 IFX$="ENDE" THENN=N-1:GOTO6000
5070 A$(N)=X$:IFN<=254 THEN5050
5080 PRINT"XSIE HABEN BEREITS 255 ELEMENTE EIN-"
5090 PRINT"XGEGEBEN - KEINE WEITEREN EINGABEN"
5100 PRINT"XMOEGLICH":FORI=1TO2000:NEXTI
6000 PRINT" ":
6010 PRINT"XDIE ZEICHENKETTEN WERDEN SORTIERT"
6020 DIMST$(12,2):M=12
6030 S=1:ST$(1,1)=1:ST$(1,2)=N
6040 L=ST$(S,1):R=ST$(S,2):S=S-1
6050 I=L:J=R:X$=A$(INT((L+R)/2))
6060 IFA$(I)<X$ THENI=I+1:GOTO6060
6070 IFX$<A$(J) THENJ=J-1:GOTO6070
6080 IFI<J THENX$=A$(I):A$(I)=A$(J):A$(J)=X$:I=I+1:J=J-1
6090 IFI<J THEN6060
6100 IFI<R THENS=S+1:ST$(S,1)=I:ST$(S,2)=R
6110 R=J:IFL<R THEN6050
6120 IFS<0 THEN6040
6150 PRINT"XSORTIEREN BEENDET"
6160 PRINT"XSOLLEN DIE ZEICHENKETTEN 'AUFSTIEGEND"
6170 PRINT"XODER 'ABSTIEGEND' AUSGEBEN WERDEN ?"
6210 PRINT"XBITTE DRUECKEN SIE TASTE '^' FUER " :FORI=1TO10:GETA$:NEXTI
6220 PRINT"XAUFSTIEGEND, TASTE '^' FUER ABSTIEGEND"
```

```

6230 GETA$: IFA$="" THEN 6230
6240 IFA$="↑" THEN 6500
6250 IFA$="←" THEN 6800
6260 PRINT "BITTE TASTE '↑' ODER '←' DRUECKEN!":GOTO 6230
6500 PRINT "J":FOR I=1 TO N:PRINT I, A$(I):GOSUB 9000:NEXT I:PRINT "N":GOTO 8000
6800 PRINT "J":FOR I=NT01 STEP -1:PRINT I, A$(I):GOSUB 9000:NEXT I:PRINT "N":GOTO 8000
8000 PRINT "WOLLEN SIE DIE ELEMENTE NOCHMAL? ":FOR I=1 TO 10:GETA$:NEXT I
8010 PRINT "AUFLISTEN? (J/N) "
8020 GETA$: IFA$="" THEN 8020
8030 IFA$="N" THEN END
8040 IFA$="J" THEN 8500
8050 PRINT "BITTE 'J' ODER 'N' DRUECKEN!":GOTO 8020
8500 PRINT "WOLLEN SIE SCHNELLE ODER LANGSAME "
8510 PRINT "AUSGABE?"
8520 PRINT "GEBEN SIE EIN 'S' FUER 'SCHNELLE':FOR I=1 TO 10:GETA$:NEXT I
8530 PRINT "UND EIN 'L' FUER 'LANGSAME' AUSGABE EIN"
8540 GETA$: IFA$="" THEN 8540
8550 IFA$="S" THEN B=1:PRINT "J":GOTO 6160
8560 IFA$="L" THEN B=100:PRINT "J":GOTO 6160
8570 PRINT "BITTE EIN 'S' ODER 'L' EINGEBEN!":GOTO 8540
9000 FOR J=1 TO B:NEXT J:RETURN
READY.

```



Commodore CBM System mit Floppy Disk Station

# Vier gewinnt

## Vier gewinnt

Dieses Programm besteht aus zwei Teilen. Laden Sie den ersten Teil und starten Sie mit RUN. Auf Wunsch erhalten Sie jetzt ausführliche Spielregeln.

Auch hier handelt es sich wieder um ein sehr interessantes Strategiespiel. Das Spielfeld (Bildschirm) besteht aus sieben Spalten mit je sechs Feldern. Die Spieler werfen abwechselnd Ihre Steine von oben in eine Spalte. Die Steine der einzelnen Spieler sind unterschiedlich gekenn-

zeichnet, so daß diese leicht unterscheiden können. Das Ziel des Spieles ist so schnell wie möglich vier Steine in einer Reihe anzuordnen. Entweder senkrecht, waagrecht oder diagonal.

Beim ersten Spiel bestimmt der Computer, wer zuerst beginnt, anschließend wird immer abwechselnd angefangen. Für einen Sieg gibt es zwei Punkte – für ein Unentschieden gibt es einen Punkt.

```
100 DIMB%(7,6),B$(2),Y$(7),I%(7),C%(2)
110 GOSUB1740
120 INPUT"WOOLDKENNEN SIE DIE SPIELREGELN BEREITS":A$
130 IFLEFT$(A$,1)="J"THEN540
140 IFLEFT$(A$,1)<>"N"THEN110
150 GOSUB1740
160 PRINT"SPIELREGELN
170 PRINT"DAS SPIELFELD BESTEHT AUS 7 SPALTEN
180 PRINT
190 PRINT"(VON 1-7 DURCHNUMMERIERT) MIT JE 6
200 PRINT
210 PRINT"FELDERN.
220 PRINT
230 PRINT"DIE SPIELER WERFEN ABWECHSELND IHRE
240 PRINT
250 PRINT"STEINE (X FUER SPIELER 1, % FUER
260 PRINT
270 PRINT"SPIELER 2) IN EINE DER SPALTEN.
280 PRINT
290 PRINT"SIEGER IST, WER ZUERST VIER STEINE
300 PRINT
310 PRINT"WAAGERECHT, SENKRECHT ODER DIAGONAL IN
320 PRINT
330 PRINT"EINER REIHE HAT.
340 GOSUB1790
350 GOSUB1740
360 PRINT"IM ERSTEN SPIEL WAEHLE ICH DEN SPIELER
370 PRINT
380 PRINT"AUS, DER ANFANGT. DANACH WIRD IMMER
390 PRINT
400 PRINT"ABWECHSELND ANGEFANGEN.
410 PRINT
420 PRINT"NACH JEDEM SPIEL ZEIGE ICH DEN SPIEL-
430 PRINT
440 PRINT"STAND AN (FUER EINEN SIEG GIBT ES ZWEI
450 PRINT
460 PRINT"PUNKTE, BEI EINEM UNENTSCHEIDEN BEKOMMT
```

```

470 PRINT
480 PRINT"JEDER EINEN PUNKT). DANACH KANN DURCH
490 PRINT
500 PRINT"DRUECKEN DER RETURN-TASTE DAS NAECHSTE
510 PRINT
520 PRINT"SPIEL GESTARTET WERDEN.
530 GOSUB1790
540 GOSUB1740
550 INPUT"MOECHTEN SIE GEGEN MICH SPIELEN";A$
560 ILEFT$(A$,1)="N"THEN790
570 ILEFT$(A$,1)<>"J"THEN540
580 GOSUB1740
590 PRINT"WENN SIE GEGEN MICH SPIELEN, BEKOMMEM
600 PRINT
610 PRINT"SIE DAS ZEICHEN #. ICH FANGE AN (ICH
620 PRINT
630 PRINT"SETZE AUS TAKTISCHEN GRUENDEN IMMER IN
640 PRINT
650 PRINT"SPALTE 4), DANACH WIRD ABWECHSELND ANGE-
660 PRINT"FANGEN. NACHDEM SIE GESETZT HABEN,
670 PRINT
680 PRINT"BILLIGEN SIE MIR BITTE CA. 20 SEKUNDEN
690 PRINT
700 PRINT"BEDENKZEIT ZU, BEVOR ICH MICH FUER EINEN
710 PRINT"ZUG ENTSCHEIDE.
720 GOSUB1790
730 GOSUB1740
740 PRINT"MLADEN SIE JETZT BITTE DEN 2. TEIL
750 PRINT
760 PRINT"DES PROGRAMMES
770 PRINT"LOAD"CHR$(34)"VIER GEWINNT 2"CHR$(34)".TTI
780 END
790 B$(1)="#####":B$(2)="#####":Z$=" "
800 Y$(1)="#####"
810 Y$(2)="#####"
820 Y$(3)="#####"
830 Y$(4)="#####"
840 Y$(5)="#####"
850 Y$(6)="#####"
860 Y$(7)="#####"
870 Q=INT(RND(1)*2+1)
880 FORI=1TO7
890 IZ(I)=1
900 NEXT:IFPEEK(639)<>94THENNEW:PRINT"#####
910 FORI=1TO7
920 FORJ=1TO6
930 BZ(I,J)=0
940 NEXTJ,I
950 Z=0:IFPEEK(1229)<>75THENNEW:PRINT"#####
960 Q=Q+1
970 IFQ=3THENQ=1
980 A=Q
990 PRINT"#####";
1000 FORI=1TO7
1010 PRINT"  #";I;"#";
1020 NEXT
1030 PRINT"
1040 PRINT"
1050 FORI=1TO6
1060 FORJ=1TO2

```



```

1690 C%(A)=C%(A)+2
1700 REM ANZEIGE DES NEUEN SPIELSTANDES
1710 PRINT " SPIELSTAND: ";C%(1); " ";C%(2);"TT"
1720 GETA$: IFA$<>CHR$(13)THEN1720
1730 GOTO880
1740 PRINT"  30*****
1750 PRINT"  30 VIER GEWINNT 30
1760 PRINT"  30*****
1770 PRINT"
1780 RETURN
1790 PRINT"3000000000000000000000000000000000"SPC(29)"30<<RETURN>>
1800 GETA$: IFA$<>CHR$(13)THEN1800
1810 RETURN
READY.

```

READY.

```

10 DIMB%(7,6),Y$(7),H%(7),I%(7),B$(5),C%(2):B$(1)="#####":B$(2)="#####":Z$=" "
20 B$(3)="N SIE ":B$(4)=" ICH ":B$(5)="#####":V=1
30 Y$(1)="#####":Y$(3)="#####":Y$(6)="#####":Y$(7)="#####":
40 Y$(4)="#####":Y$(2)="#####":Y$(5)="#####":
70 V=V+1:IFV=3THENV=1
80 A=V:PRINT"#####":FORI=1TO7:FORJ=1TO6:B%(I,J)=0:NEXT:I%(I)=1:PRINT"  #";I;"#";
:NEXT:Z=0
90 PRINT"#####":FORI=1TO6:FORJ=1TO2
:NEXT
110 PRINT"  # " :NEXT:GOTO190
120 PRINT"#####":PRINTY$(J):B$(A):PRINTY$(J):B$(A):IFI%(J)=6THENRETURN
130 FORI=5TOI%(J)STEP-1:PRINT"TT"Y$(J):Z$:PRINT"TT"Y$(J):B$(A):PRI
NTY$(J):B$(A):NEXT
140 RETURN
150 PRINTB$(5)" DAMIT HABE"B$(A+2)"GEWONNEN":C%(A)=C%(A)+2
160 PRINT" SPIELSTAND: ";C%(1); " ";C%(2);"TT"
170 GETA$: IFA$<>CHR$(13)THEN170
180 Z=1:RETURN
190 IFA=1THEN220
200 J=4:GOSUB120:B%(4,I%(4))=2:I%(4)=2
210 A=1
220 GETA$: IFA$=""THEN220
230 J=VAL(A$):IFJ<10RJ>7THEN220
240 IFI%(J)=7THEN220
250 IFB%(J,I%(J))<>0THEN220
270 B%(J,I%(J))=1:GOSUB120:M=4:GOSUB360:IFZ=1THEN70
280 I%(J)=I%(J)+1:FORI=1TO7:IFI%(I)<7THEN300
290 NEXT:GOTO330
300 GOSUB350:B%(J,I%(J))=2:GOSUB120:M=4:GOSUB360:IFZ=1THEN70
310 I%(J)=I%(J)+1:FORI=1TO7:IFI%(I)<7THEN210
320 NEXTI
330 PRINTB$(5)" DAS IST EIN UNENTSCHEIDEN !"
340 C%(1)=C%(1)+1:C%(2)=C%(2)+1:GOSUB160:IFZ=1THEN70
350 M=3:A=2:X=0
360 D=0:FORG=1TO6:FORC=1TO7:IFB%(C,G)=ATHEND=D+1:IFD=MTHEN670
370 IFB%(C,G)<>ATHEND=0
380 NEXTC:D=0:NEXTG:D=0:FORC=1TO7:FORG=1TO6:IFB%(C,G)=ATHEND=D+1:IFD=MTHEN1870
390 IFB%(C,G)<>ATHEND=0
400 NEXTG:D=0:NEXTC:D=0:FORH=1TO4:C=H:FORG=1TO6:IFB%(C,G)=ATHEND=D+1:IFD=MTHEN8
70

```

```

410 IFB%(C,G)<>ATHEND=0
420 C=C+1:IFC=8THEN440
430 NEXTG
440 D=0:NEXTH:D=0:FORH=1TO2:C=1:FORG=H+1TO6:IFB%(C,G)=ATHEND=D+1:IFD=MTHEN990
450 IFB%(C,G)<>ATHEND=0
460 C=C+1
470 NEXTG:D=0:NEXTH:D=0:FORH=7TO4STEP-1:C=H:FORG=1TO6:IFB%(C,G)=ATHEND=D+1:IFD=
MTHEN1100
480 IFB%(C,G)<>ATHEND=0
490 C=C-1:IFC=0THEN510
500 NEXTG
510 D=0:NEXTH:D=0:FORH=1TO2:C=7:FORG=H+1TO6:IFB%(C,G)=ATHEND=D+1:IFD=MTHEN1220
520 IFB%(C,G)<>ATHEND=0
530 C=C-1:NEXTG:D=0:NEXTH:D=0:IFM=4THENRETURN
540 IFX=0ANDM=3ANDA=2THENM=2:GOTO360
550 IFX=0ANDM=2ANDA=2THEN2320
560 IFX=0ANDM=3ANDA=1THENM=2:GOTO360
570 IFX=0ANDM=2ANDA=1THEN2320
580 IFX=1ANDM=2ANDA=2THENA=1:GOTO360
590 A=2:X=0:FORI=1TO7:IFH%(I)=0ANDI%(I)<7THEN610
600 NEXT:GOTO640
610 J=INT(RND(1)*7+1):IFI%(J)>6THEN610
620 IFH%(J)<>0THEN610
630 GOTO1860
640 FORI=1TO7:IFH%(I)=AANDI%(I)<7THENJ=I:GOTO1860
650 NEXT:FORI=1TO7:IFH%(I)<>AANDI%(I)<7THENJ=I:GOTO1860
660 NEXT
670 IFM=4THEN150
680 IFA=1ANDB%(4,G)=2THEN860
690 IFA=2ANDB%(4,G)=1THEN860
700 IFM=3THEN1330
710 IFX=1THEN2210
720 IFC>5THEN800
730 IFB%(C+1,G)<>0THEN790
740 IFG=1THEN780
750 IFB%(C+2,G)=AANDB%(C+1,G-1)<>0THENJ=C+1:GOTO1860
760 IFB%(C+2,G)=AAND(G=2ORB%(C+1,G-2)<>0)THENH%(C+1)=A
770 GOTO790
780 IFB%(C+2,1)=ATHENJ=C+1:GOTO1860
790 IFC<4THEN860
800 IFB%(C-2,G)<>0THEN860
810 IFG=1THEN850
820 IFB%(C-3,G)=AANDB%(C-2,G-1)<>0THENJ=C-2:GOTO1860
830 IFB%(C-3,G)=AAND(G=2ORB%(C-2,G-2)<>0)THENH%(C-2)=A
840 GOTO860
850 IFB%(C-2,1)=0ANDB%(C-3,1)=ATHENJ=C-2:GOTO1860
860 D=0:GOTO380
870 IFM=4THEN150
880 IFM=3THEN1460
890 IFX=1THEN1930
900 IFC>5ORG>4THEN940
910 IFB%(C+1,G+1)<>0THEN940
920 IFB%(C+2,G+2)=AANDB%(C+1,G)<>0THENJ=C+1:GOTO1860
930 IFB%(C+2,G+2)=AANDB%(C+1,G-1)<>0THENH%(C+1)=A
940 IFG<4THEN980
950 IFB%(C-2,G-2)<>0THEN980
960 IFB%(C-3,G-3)=AANDB%(C-2,G-3)<>0THENJ=C-2:GOTO1860
970 IFB%(C-3,G-3)=AAND(G=4ORB%(C-2,G-4)<>0)THENH%(C-2)=A
980 D=0:GOTO420
990 IFM=4THEN150
1000 IFM=2THEN1560

```

```

1010 IFG=6THEN1050
1020 IFB%(C+1,G+1)⟨0THEN1050
1030 IFB%(C+1,G)⟨0THENJ=C+1:GOTO1860
1040 IFB%(C+1,G-1)⟨0THENH%(C+1)=A
1050 IFC=3THEN1090
1060 IFB%(C-3,G-3)⟨0THEN1090
1070 IFB%(C-3,G-4)⟨0THENJ=C-3:GOTO1860
1080 IFG=5ORB%(C-3,G-5)⟨0THENH%(C-3)=A
1090 D=0:GOTO460
1100 IFM=4THEN150
1110 IFM=2THEN1660
1120 IFC=1ORG=6THEN1160
1130 IFB%(C-1,G+1)⟨0THEN1160
1140 IFB%(C-1,G)⟨0THENJ=C-1:GOTO1860
1150 IFB%(C-1,G-1)⟨0THENH%(C-1)=A
1160 IFG<4THEN1210
1170 IFB%(C+3,G-3)⟨0THEN1210
1180 IFG=4THENJ=C+3:GOTO1860
1190 IFB%(C+3,G-4)⟨0THENJ=C+3:GOTO1860
1200 IFG=5ORB%(C+3,G-5)⟨0THENH%(C+3)=A
1210 D=0:GOTO490
1220 IFM=4THEN150
1230 IFM=2THEN1760
1240 IFG=6THEN1280
1250 IFB%(C-1,G+1)⟨0THEN1280
1260 IFB%(C-1,G)⟨0THENJ=C-1:GOTO1860
1270 IFB%(C-1,G-1)⟨0THENH%(C-1)=A
1280 IFC=5THEN1320
1290 IFB%(C+3,G-3)⟨0THEN1320
1300 IFB%(C+3,G-4)⟨0THENJ=C+3:GOTO1860
1310 IFG=5ORB%(C+3,G-5)⟨0THENH%(C+3)=A
1320 D=0:GOTO530
1330 IFC=7THEN1400
1340 IFG=1THEN1380
1350 IFB%(C+1,G)=0ANDB%(C+1,G-1)⟨0THENJ=C+1:GOTO1860
1360 IFB%(C+1,G-2)⟨0ORG=2ANDB%(C+1,1)=0THENH%(C+1)=A
1370 GOTO1390
1380 IFB%(C+1,1)=0THENJ=C+1:GOTO1860
1390 IFC<4THEN1450
1400 IFG=1THEN1440
1410 IFB%(C-3,G)=0ANDB%(C-3,G-1)⟨0THENJ=C-3:GOTO1860
1420 IFB%(C-3,G-2)⟨0ORG=2ANDB%(C-3,1)=0THENH%(C-3)=A
1430 GOTO1450
1440 IFB%(C-3,1)=0THENJ=C-3:GOTO1860
1450 D=0:GOTO380
1460 IFC=7ORG=6THEN1500
1470 IFB%(C+1,G+1)⟨0THEN1500
1480 IFB%(C+1,G)⟨0THENJ=C+1:GOTO1860
1490 IFB%(C+1,G-1)⟨0THENH%(C+1)=A
1500 IFG=3THEN1550
1510 IFB%(C-3,G-3)⟨0THEN1550
1520 IFG=4THENJ=C-3:GOTO1860
1530 IFB%(C-3,G-4)⟨0THENJ=C-3:GOTO1860
1540 IFG=5ORB%(C-3,1)⟨0THENH%(C-3)=A
1550 D=0:GOTO420
1560 IFX=1THEN2000
1570 IFG>4THEN1610
1580 IFB%(C+1,G+1)⟨0THEN1610
1590 IFB%(C+2,G+2)=AANDB%(C+1,G)⟨0THENJ=C+1:GOTO1860
1600 IFB%(C+2,G+2)=AANDB%(C+1,G-1)⟨0THENH%(C+1)=A
1610 IFC<4THEN1650

```

```

1620 IFB%(C-2,G-2) <> 0 THEN 1650
1630 IFB%(C-3,G-3)=ARANDB%(C-2,G-3) <> 0 THEN J=C-2:GOTO1860
1640 IFB%(C-3,G-3)=ARANDB%(C-2,G-4) <> 0 THEN H%(C-2)=A
1650 D=0:GOTO460
1660 IFX=1 THEN 2070
1670 IFC<30RG>4 THEN 1710
1680 IFB%(C-1,G+1) <> 0 THEN 1710
1690 IFB%(C-2,G+2)=ARANDB%(C-1,G) <> 0 THEN J=C-1:GOTO1860
1700 IFB%(C-2,G+2)=ARANDB%(C-1,G-1) <> 0 THEN H%(C-1)=A
1710 IFG<4 THEN 1750
1720 IFB%(C+2,G-2) <> 0 THEN 1750
1730 IFB%(C+3,G-3)=ARANDB%(C+2,G-3) <> 0 THEN J=C+2:GOTO1860
1740 IFB%(C+3,G-3)=ARAND<G=40RB%(C+2,G-4) <> 0 THEN H%(C+2)=A
1750 D=0:GOTO490
1760 IFX=1 THEN 2140
1770 IFG>4 THEN 1810
1780 IFB%(C-1,G+1) <> 0 THEN 1810
1790 IFB%(C-2,G+2)=ARANDB%(C-1,G) <> 0 THEN J=C-1:GOTO1860
1800 IFB%(C-2,G+2)=ARANDB%(C-1,G-1) <> 0 THEN H%(C-1)=A
1810 IFC>4 THEN 1850
1820 IFB%(C+2,G-2) <> 0 THEN 1850
1830 IFB%(C+3,G-3)=ARANDB%(C+2,G-3) <> 0 THEN J=C+2:GOTO1860
1840 IFB%(C+3,G-3)=ARANDB%(C+2,G-4) <> 0 THEN H%(C+2)=A
1850 D=0:GOTO530
1860 FOR I=1 TO 7:HZ(I)=0:NEXT A: A=2: RETURN
1870 IFM=4 THEN 150
1880 IF I%(C)=7 OR G=M+3 THEN 400
1890 IFB%(C,G+1) <> 0 THEN 400
1900 IFM=3 THEN J=C:GOTO1860
1910 IFX=1 AND H%(C)=0 THEN J=C:GOTO1860
1920 GOTO400
1930 IFG>50 RC>6 THEN 1960
1940 IFB%(C+1,G+1) <> 0 THEN 1960
1950 IFB%(C+1,G) <> 0 AND H%(C+1)=0 THEN J=C+1:GOTO1860
1960 IFG<30 RC<3 THEN 1990
1970 IFB%(C-2,G-2) <> 0 THEN 1990
1980 IF(B%(C-2,G-3) <> 0 OR G=3) AND H%(C-2)=0 THEN J=C-2:GOTO1860
1990 D=0:GOTO420
2000 IFG=6 THEN 2030
2010 IFB%(C+1,G+1) <> 0 THEN 2030
2020 IFB%(C+1,G) <> 0 AND H%(C+1)=0 THEN J=C+1:GOTO1860
2030 IFC=2 THEN 2060
2040 IFB%(C-2,G-2) <> 0 THEN 2060
2050 IFB%(C-2,G-3) <> 0 AND H%(C-2)=0 THEN J=C-2:GOTO1860
2060 D=0:GOTO460
2070 IFC=10 RG=6 THEN 2100
2080 IFB%(C-1,G+1) <> 0 THEN 2100
2090 IFB%(C-1,G) <> 0 AND H%(C-1)=0 THEN J=C-1:GOTO1860
2100 IFG<30 RC>5 THEN 2130
2110 IFB%(C+2,G-2) <> 0 THEN 2130
2120 IF(B%(C+2,G-3) <> 0 OR G=3) AND H%(C+2)=0 THEN J=C+2:GOTO1860
2130 D=0:GOTO490
2140 IFG=6 THEN 2170
2150 IFB%(C-1,G+1) <> 0 THEN 2170
2160 IFB%(C-1,G) <> 0 AND H%(C-1)=0 THEN J=C-1:GOTO1860
2170 IFC>5 THEN 2200
2180 IFB%(C+2,G-2) <> 0 THEN 2200
2190 IFB%(C+2,G-3) <> 0 AND H%(C+2)=0 THEN J=C+2:GOTO1860
2200 D=0:GOTO530
2210 IFC>6 THEN 2270
2220 IFG=1 THEN 2250

```

```

2230 IFB%(C+1,G)=0ANDB%(C+1,G-1)<>0ANDH%(C+1)=0THENJ=C+1:GOTO1860
2240 GOTO2260
2250 IFB%(C+1,1)=0ANDH%(C+1)=0THENJ=C+1:GOTO1860
2260 IFC<3THEN2310
2270 IFG=1THEN2300
2280 IFB%(C-2,G)=0ANDB%(C-2,G-1)<>0ANDH%(C-2)=0THENJ=C-2:GOTO1860
2290 GOTO2310
2300 IFB%(C-2,1)=0ANDH%(C-2)=0THENJ=C-2:GOTO1860
2310 D=0:GOTO380
2320 FORG=1TO6
2325 IFB%(4,G)=0ANDB%(3,G)=AANDB%(5,G)=AANDB%(2,G)=0ANDB%(6,G)=0ANDH%(4)=0THEN2
350
2330 IFB%(4,G)=AANDB%(3,G)=0ANDB%(2,G)=AANDB%(1,G)=0ANDB%(5,G)=0ANDH%(3)=0THEN2
400
2340 IFB%(4,G)=AANDB%(5,G)=0ANDB%(6,G)=AANDB%(7,G)=0ANDB%(3,G)=0ANDH%(5)=0THEN2
450
2342 NEXT:IFA=2THENA=1:M=3:GOTO360
2345 X=1:A=2:GOTO360
2350 IFG=10RB%(2,G-1)<>0ANDB%(6,G-1)<>0ANDB%(4,G-1)<>0THENJ=4:GOTO1860
2360 IFB%(2,G-1)<>0ANDB%(6,G-1)<>0AND(G=20RB%(4,G-2)<>0)THENH%(4)=A
2370 GOTO2342
2400 IFG=10RB%(1,G-1)<>0ANDB%(3,G-1)<>0ANDB%(5,G-1)<>0THENJ=3:GOTO1860
2420 GOTO2342
2450 IFG=10RB%(3,G-1)<>0ANDB%(5,G-1)<>0ANDB%(7,G-1)<>0THENJ=5:GOTO1860
2460 GOTO2342
READY.

```



```

80 PRINT"SIE SIND ";RT$:PRINT"SIE SIND ";RR$
100 R$="":GOSUB 500
110 IFR$="DARK"THENPOKE59409,52:R$="":GOTO100
120 IFR$="HELL"THENPOKE59409,60:R$="":GOTO100
130 IFR$="SUCH"THENR$="":GOSUB550:GOTO100
140 IF R$="SICHERN"THENR$="":GOTO2000
150 IF R$="LIST"THENGOSUB1000:GOTO100
160 IF R$="SAVE"THEN49
170 IFR$="STOP"THENSTOP
499 PRINT"SYNTAX ERROR":GOTO100
500 P1=167:IFPEEK(50003)=0THENP1=548
501 R$="":POKEP1,0
502 GETE$:IFE$=""THEN502
503 IFE$=CHR$(13)THENPOKEP1,1:PRINT" ■":RETURN
504 IFR$=CHR$(20)ANDLEN(R$)<2THENPRINTE$:GOTO501
505 IFE$=CHR$(20)THENR$=LEFT$(R$,LEN(R$)-1):PRINTE$:GOTO502
524 PRINTE$:R$=R$+E$:GOTO502
540 REM SUCH
550 GT=0:PRINT"ZUSUCHEN EINE(N) NAMENS(N)"
555 GOSUB500
600 RESTORE
601 READNN$
602 IFNN$="ENDE"THEN750
603 READVN$,ST$,WO$,TE$,GE$,HO$,FA$,SO$
605 IFR$<>NN$THEN601
606 GOTO800
609 PRINT"ZUSUCHEN EINES NAMENS(N)"
610 PRINT"GESUCHTER NAME: ",R$
620 PRINT"  NACHNAME   :";NN$
630 PRINT"  VORNAME    :";VN$
640 PRINT"  WOHNORT    :";WO$
650 PRINT"  STRASSE     :";ST$
660 PRINT"  TELEFON     :";TE$
680 IFQW<>88THEN686
681 PRINT"WEITERE ANGABEN?"
682 GETR$:IFR$=""THEN682
683 IFR$="J"THEN700
684 IFR$="N"THEN755
685 GOTO682
686 PRINT"WEITERE ANGABEN NUR AN BERECHTIGTE ■      PERSONEN! ■"
690 GOTO755
700 PRINT"  GEBURTSTAG: ";GE$
710 PRINT"  HOBBIES    : ";HO$
720 PRINT"  FAHRZEUG   : ";FA$
730 PRINT"  SONSTIGES  : ";SO$
749 GOTO755
750 IFGT=0THENPRINT"  WIESER NAME IST NICHT IN DER KARTEI      ENHALTEN. (X)"
751 IFGT=1THENPRINT"  WIESER NAME IST SONST NICHT IN DER   KARTEI ENHALTEN. (X)"
"
752 GOTO795
755 PRINT"WOLLEN SIE DIE DATEN AUSGEDRUCKT?"
756 GETR$:IFR$=""THEN756
757 IFR$="N"THEN795
758 IFR$<>"J"THEN756
759 OPEN1,5:PRINT#1,NN$:PRINT#1,VN$:PRINT#1,ST$:PRINT#1,WO$:PRINT#1,TE$:PRINT#1
760
761 PRINT"  WOLLEN SIE ALLE DATEN AUSGEDRUCKT?"
762 GETR$:IFR$=""THEN762
763 IFR$="N"THEN790
764 IFR$<>"J"THEN762
765 PRINT#1,GE$:PRINT#1,HO$:PRINT#1,FA$:PRINT#1,SO$

```

```

790 FORA=1TO10:PRINT#1:NEXTA
791 CLOSE1
795 PRINT"WO WOLLEN SIE NOCH DATEN EINER ANDEREN PERSON WISSEN?"
796 GETR$:IFTR$=""THEN796
797 IFTR$="J"THEN550
798 IFTR$="N"THENPRINT"J":PRINT"BEFEHLE":RETURN
799 GOTO760
800 PRINT"SUCHEN EINE(N) NAMENS(N)"
810 PRINT"GESUCHTER NAME: ",R$
820 PRINT"  NACHNAME :";NN$
830 PRINT"  VORNAME :";VN$
840 PRINT"  WOHNORT :";WO$
850 PRINT"  STRASSE :";ST$
860 PRINT"  TELEFON :";TE$
870 PRINT"WO IST DIES DER GESUCHTE NAME?"
871 GT=1
880 GETQ$:IFQ$=""THEN880
883 IFQ$="J"THEN609
884 IFQ$="N"THEN601
885 GOTO880
1000 PRINT"ZU LISTEN ALLER NAMEN"
1050 PRINT"X":AS=0:RESTORE
1100 READNN$
1110 IFNN$="ENDE"THEN1500
1120 READVN$,ST$,WO$,TE$,GE$,HO$,FA$,SO$
1200 PRINTNN$,VN$
1205 AS=AS+1:IFAS=20THENAS=0:GOTO1250
1210 GOTO1100
1220 GOTO1100
1250 PRINT"WEITER?"
1260 GETR$:IFR$=""THEN1260
1265 IFR$="J"THEN1000
1270 IFR$<"N"THEN1260
1280 PRINT"BEFERTIG?"
1290 GETR$:IFR$=""THEN1290
1295 IFR$="J"THENPRINT"J":PRINT"BEFEHLE":RETURN
1296 IFR$<"N"THEN1290
1297 GOTO1290
1500 PRINT"ENDE":GOTO1200
2000 PRINT"ZULEGEN SIE EINE ZURUECKGESPULTE CASSETTE EIN."
2010 PRINT"DRUECKEN SIE 'PLAY'UND'RECORD'."
2020 IFPEEK(59411)=61THEN2020
2030 PRINT"OK"
2040 SAVE"DATEI"
2045 GOTO1
2050 PRINT"BEFEHLE":GOTO100
10000 DATANACHNAME,VORNAME,STRASSE,WOHNORT,TELEFON
10001 DATAGEBURTSTAG,HOBBY,FAHRZEUG,SONSTIGES
60000 DATA"ENDE"
READY.

```

## NOTIZEN

# Konto 99

## KONTO 99

Dieses Programm, zeigt wie man eine brauchbare Software aufbaut und sie so gestaltet, daß der Benutzer im Dialogverkehr damit entlastet wird und immer weiß, was er gerade zu tun hat. Wichtig ist auch der Aufbau eines ansprechenden Displays für die Eingabe. Dazu sind einige "Tricks" im Programm enthalten, die erklärt werden sollen. Vermieden wurden alle Maschinenroutinen, man kann also auf alle BASIC-Maschinen damit arbeiten.

Geschrieben wurde auf einer CBM 3032, CBM 3040 FLOPPY, CBM 3022 DRUCKER.

Bei anderen Druckern\* muß evtl. abweichende Ausdruck-Formatierung (Zeilen 1700 - 1760) korrigiert werden.

## Im Einzelnen:

Zeile 5 schaltet die STOP-Taste des Rechners aus. Man kann also durch versehentliche Betätigung derselben nicht aus dem Programm "hinausfliegen". Mancher wird sich ferner schon gegifft haben, wenn er bei einem INPUT-Befehl vergessen hatte, etwas einzugeben und gleich die RETURN-Taste drückt. Dann ist er draußen. Hier geht das nicht.

In den INPUT-Befehlen werden nach dem Text drei GESHIFTETE LEERZEICHEN gegeben, gleich gefolgt von drei CURSOR-RETOUR. Für eine Numerische Variable. Für eine \$-Variable kommt noch dazu. 'IF(N)\$=" THEN PRINT' (2x Cursor hinauf):GOTO xxx (Zeile des Input-Befehls. Das Blank vor 'THEN' muß ebenfalls geschiftet sein.

```
4 POKE182,8:REM BLOCKADE DER HELP-DOS REPET-FUNKTION (INDIVIDUELL !)*
5 POKE144,49:REM BLOCKADE STOP-TASTE*****
6 POKE59468,12:REM GROSS-SCHREIBUNG *****
8 POKE182,8:REM WIE 4 (SICHERUNG !) *****
10 PRINT"K O N T O "
20 PRINT"
30 PRINT"MIT SPEICHER, SUMME, DATUM, SALDO, DRUCK"
32 PRINT"ALLE KONTENFUNKTIONEN (NEG. SALDO !)*"
50 PRINT"
52 PRINT"VERBOTENE ZEICHEN : , : ANF.ZEICH."
60 PRINT"FUER 100 POSTEN "
75 PRINT"
80 PRINT" ABHEBUNGEN - VORZEICHEN !"
89 REM DIMENSIONIERUNG (BELIEBIG HOCH-SPEICHERFRAGE !)*
90 DIMD$(100):DIMA$(100):DIMB$(100):DIMA(100):DIMS(100)
100 Y$=" "
110 YI$="===== "
120 CR$=CHR$(13)
190 GOSUB 490
199 REM EINGABE UND RECHENROUTINEN*****
200 INPUT"MONAT "M$:IFM$=" THENPRINT"TT":GOTO 200
205 INPUT"ZAUSZUG NR. "R$:IFR$=" THENPRINT"TT":GOTO 205
210 INPUT"KONTOSTAND "K
212 W=K:GOSUB 1000:K$=H$:H$="0"
220 GOSUB1000
240 I=1
242 PRINT
244 PRINT"ING.ENDE:0 0 ABHGB. - EINZ. + ! "
```

```

245 PRINT "DAT. BEZEICHNUNG BETRAG
246 PRINT "RET.← RET.← RET.←
247 PRINT "
250 INPUT " I"; I$; I:=IFD$(I)=" THENPRINT " :GOTO 250
252 INPUT " I"; I$; I:=IFD$(I)=" THENPRINT " :GOTO252
254 INPUT " I"; I$; I:=IFD$(I)=" THENPRINT "
260 S(I)=S(I-1)+A(I):S=S(I):GOSUB1100:IFD$(I)<"0" THENI=I+1:PRINT:GOTO250
310 W=S:GOSUB1000:S#=H#:N=K-VAL(S#):W=N:GOSUB1000:N#=H#:PRINT:GOTO 600
399 REM BEFEHLSABFRAGE*****
489 REM BEFEHLS-ABFRAGE
490 PRINT "INITIALISIEREN ? IN * AUSSTEIGEN"
492 GETZ$:IFZ$<"I"ANDZ$<"N"ANDZ$<"A"THEN 492
494 IFZ$="A"THEN 2000
496 IFZ$="I"THEN 1960
500 PRINT "LESEN, SPEICHERN, DRUCKEN, ERGAEENZEN,
510 PRINT " NEUES KONTO, ( ZEIGEN )"
520 GETZ$:IFZ$<"L"ANDZ$<"S"ANDZ$<"D"ANDZ$<"E"ANDZ$<"N"ANDZ$<"Z"THEN 520
530 IFZ$="L"THEN 1500
540 IFZ$="S"THEN 1400
550 IFZ$="D"THEN 1700
552 IFS>0 THENX$="S -"
554 IFS<0 THENX$="S +"
560 IFZ$="E"THEN PRINT: S$="0":N$="0":GOTO 244
570 IFZ$="N"THEN A$(I)="0":B$(I)="0":S$="0":N$="0":K$="0":GOTO 200
580 IFZ$="Z"THEN600
585 END
589 REM BILDSCHIRM-FORMATIERUNG*****
600 IFK<0 THENX$="S +"
602 IFK<0 THENX$="S -"
604 PRINT "AB AUSZ.NR "M$TAB(26)X$TAB(39-LEN(K$))K$
610 PRINT "
615 FORI=1TO100
620 PRINTD$(I)TAB(3)A$(I);
621 IFA(I)<0 THENX$="S -"
622 IFA(I)>0 THENX$="S*+"
625 PRINTTAB(26)X$;:IFD$(I)="0"THEN 650
630 PRINTTAB(39-LEN(B$(I)))B$(I):IFI=100 THEN 650
632 GETX$:IFX$<" " THEN NEXTI
634 PRINT "Forts., SPACE= STOP, ROLL-ENDE= Z"
636 GETX$:IFX$="F"THEN NEXTI
638 IFX$=""THEN 636
640 IFX$="Z"THEN 650
650 PRINTTAB(29)Y$
655 IFW>0 THENX$="S -"
656 IFW<0 THENX$="S +"
660 PRINT "SUMME ABHG.(-)/EINZ.(*)"TAB(26)X$TAB(39-(LEN(S#)))S#
670 PRINT "
675 IFN>0 THENX$="S +"
676 IFN<0 THENX$="S -"
680 PRINT "AKT. STAND: "M$TAB(26)X$TAB(39-LEN(N$))N$
685 PRINTTAB(29)Y1$
690 GOTO 490
699 END
999 REM RUNDUNG UND ERGAEENZUNG (BIS 1190)*****
1000 L=ABS(W):H$=STR$(L):J=L
1010 H$=STR$(INT(J)):H$=H$+"." :J=INT(100*(J-INT(J))+.5)
1020 IFJ=0 THENH$=H$+"00":GOTO1060
1030 IFJ>99 THENJ=VAL(H$)+1:GOTO 1010
1040 C$=STR$(J):IFLEN(C$)=2 THENH$=H$+"0"
1050 H$=H$+RIGHT$(C$,LEN(C$)-1)
1060 RETURN

```

```

1090 END
1100 L=ABS(A(I)):B$(I)=STR$(L):J=L
1110 B$(I)=STR$(INT(J)):B$(I)=B$(I)+".":J=INT(100*(J-INT(J))+.5)
1120 IF J=0 THEN B$(I)=B$(I)+"00":GOTO 1160
1130 IF J>99 THEN J=VAL(B$(I))+1:GOTO 1110
1140 C#=STR$(J):IF LEN(C#)=2 THEN B$(I)=B$(I)+"0"
1150 B$(I)=B$(I)+RIGHT$(C#,LEN(C#)-1)
1160 RETURN
1190 END
1399 REM SEQU.FILE SCHREIBEN *****
1400 INPUT"DISKSTORE: KONTO-NAME   ";F$:IFF$=" " THEN PRINT"TI":GOTO 1400
1410 INPUT"DISK IN DR.NR (0/1)     ";E:CLOSE 4
1430 CLOSE 4:OPEN 4,8,4,"@"+STR$(E)+"":+F$+",S,W":I=1
1431 CLOSE 15:OPEN 15,8,15:GOSUB 1600
1440 PRINT#4,M$;CR$;R$;CR$;STR$(K);CR$;STR$(S);CR$;STR$(N);CR$;
1460 PRINT#4,D$(I);CR$;A$(I);CR$;B$(I);CR$;STR$(A(I));CR$;STR$(S(I));CR$;
1470 IF D$(I)<>"0" THEN I=I+1:GOTO 1460
1485 PRINT"DISKSTORE, OVERWRITE*";F$;" / " :GOSUB 1600
1486 FOR J=1 TO 2800:NEXT
1490 CLOSE 4:PRINT"J":GOTO 600
1495 END
1499 REM SEQU.FILE LESEN *****
1500 INPUT"DISKLOAD: KONTO-NAME   ";F$:IFF$=" " THEN PRINT"TI":GOTO 1500
1510 INPUT"DISK IN DR.NR (0/1)     ";E:
1519 CLOSE 15:OPEN 15,8,15:GOSUB 1600
1520 CLOSE 15:CLOSE 4
1530 OPEN 4,8,4,STR$(E)+"":+F$+",S,R":OPEN 15,8,15:PRINT"*LOAD* " :GOSUB 1600
1540 INPUT#4,M$,R$,K,S,N
1545 I=1
1560 INPUT#4,D$(I),A$(I),B$(I),A(I),S(I):IF D$(I)<>"0" THEN I=I+1:GOTO 1560
1585 PRINT"DISKLOAD*";F$;" / " :GOSUB 1600
1586 FOR J=1 TO 2800:NEXT:CLOSE 15:CLOSE 4
1590 W=K:GOSUB 1000:K#=H$:W=S:GOSUB 1000:S#=H$:W=N:GOSUB 1000:N#=H$:GOTO 600
1591 REM RECHEN-VARIABLE IN DRUCK-VARIABLE VERWANDELN *****
1595 END
1599 REM ERROR KANAL RUFEN *****
1600 INPUT#15,EN$,EM$,ET$,ES$
1630 PRINTEM$;" ";EN$;" ";ET$;" ";ES$
1640 IF EN$="00" THEN RETURN
1650 CLOSE 15:CLOSE 4:GOTO 490
1699 REM AUSDRUCK-FORMAT FUER CBM 3022(3023) *****
1700 OPEN 5,4
1710 OPEN 6,4,1
1720 Z$=CHR$(29):Q$=CHR$(141)
1730 G$=" "
1750 OPEN 7,4,2
1760 PRINT#7,G$;"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
1761 PRINT#5,Q$;"
1762 PRINT#5,Q$;Z$;"          KONTO *?: 000 000 000 - AUSZ.NR: ";Z$;R$
1763 PRINT#5,Q$;"
1764 CLOSE 7
1765 OPEN 7,4,2
1766 PRINT#7,G$;"AAAAAAAAAAAAAAAAAAAAAAAAAAA 9999999.99
1770 IF K>0 THEN X$="S +"
1771 IF K<0 THEN X$="S -"
1775 PRINT#6,Q$;Z$;"AB AUSZ.NR";Z$;M$;Z$;X$;Z$;K$
1780 PRINT#5,Q$;"
1790 CLOSE 7
1800 OPEN 7,4,2
1810 PRINT#7,G$;"AAA AAAAAAAAAAAAAAAAAAAAAAAAAAAA 9999999.99
1815 I=1

```

```

1820 PRINT#6,D$(I);Z$;A$(I),
1822 IFA(I)>0THENX$="S*+"
1823 IFA(I)<0THENX$="S -"
1824 PRINT#6," " ;Z$;X$;Z$;B$(I):IFD$(I)<>"0"THENI=I+1:GOTO 1820
1830 CLOSE7
1840 PRINT#5,Q$;SPC(40)"-----"
1850 OPEN7,4,2
1860 PRINT#7,G$;"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAA 9999999.99
1862 IFS>0THENX$="S +"
1864 IFS<0THENX$="S -"
1870 PRINT#6,Q$;G$;"SUMME:ABHBGN.(-)/EINZ.(*)";Z$;X$;Z$;S$
1880 CLOSE7
1900 OPEN7,4,2
1910 PRINT#7,G$;"AAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAA AAA 9999999.99
1915 PRINT#5,G$;"-----"
1917 IFN>0THENX$="S*+"
1919 IFN<0THENX$="S -"
1920 PRINT#6,Q$;G$;"NEU.STAND:";Z$;M$;Z$;X$;Z$;N$
1930 PRINT#5,Q$;G$;"-----"
1935 PRINT#5,:PRINT#5:PRINT#5
1940 CLOSE5:CLOSE6:CLOSE7:PRINT"J":GOTO490
1950 END
1960 PRINT"X00INITIALISIEREN. FACH NR 301/31"
1962 GETDD$:IFDD$<>"0"ANDDD$<>"1"THEN 1962
1964 CLOSE15,8,15:OPEN15,8,15
1966 IFDD$="1"THEN1970
1968 IFDD$="0"THEN1974
1970 PRINT#15,"I1":PRINT"J"TAB(28)"1"
1972 GOTO 1976
1974 PRINT#15,"I0":PRINT"J"TAB(28)"0"
1976 CLOSE15:PRINT"J":GOTO 500
2000 CLOSE15:END:REM GESAMT AUSSTIEG *****
READY.

```

# Auto-Jahresabrechnung

## Auto-Jahresabrechnung

Das nachfolgende Programm erstellt eine Jahresabrechnung. Über Device 4 kann ein Drucker angeschlossen werden. Die Daten werden in DATA-Statements eingegeben.

Die Reihenfolge ist zu beachten.

```
500 DATA 88302 : REM Anfangs Km-Stand
510 DATA 8892, 41.54, 35.27
      ↑      ↑      ↑
      Km-Stand Liter Preis in DM
```

Die Tankungen werden mit DATA-1 abgeschlossen.

810 DATA-1

```
820 DATA 3, 19.95, 2.5, 20, 3, 19.95, 3, 1995
      3 Liter Öl | DM 19,95 | 2,5 Liter | 20,-
830 DATA-1
```

Mit DATA-1 kann auch die Ölingabe abgeschlossen werden.

Anschließend können noch alle restlichen Kosten eingegeben werden. Siehe z. B. Zeile 830. Hier gibt man einfach die Beträge getrennt durch ein Komma ein. Achten Sie bitte darauf, daß die Dezimal Kennzeichnung als Punkt und nicht als Komma eingegeben wird. Am Ende muß dann wieder mit DATA-1 abgeschlossen werden.

```
10 PRINT "AUTO - JAHRESABRECHNUNG"
11 INPUT "DRUCKEN (J/N) "; A$: P=A$="J"
12 REM
13 REM DEVICE-NR. FUER DRUCKER HIER: 4
14 REM (ZEILE 16)
16 IF THEN OPEN 1, 4: PRINT#1: PRINT#1, " AUTO - JAHRESABRECHNUNG"
20 READ KA: K0=KA
30 PRINT "ANFANGS-KM "; KA: PRINT "TANKUNGEN (KM,L,DM,L/100KM):"
35 IF THEN PRINT#1, "ANFANGS-KM "; KA: PRINT#1, "TANKUNGEN (KM,L,DM,L/100KM):"
40 SL=0: SD=0
50 READ KE: IF KE<K0 THEN 100
60 READ L, DM: SL=SL+L: SD=SD+DM
70 PRINT KE; L; SPC(8-LEN(STR$(L)));
80 PRINT DM; SPC(8-LEN(STR$(DM)));
90 PRINT L*100/(KE-K0): IF NOT P THEN K0=KE: GOT050
92 PRINT#1, KE; SPC(8-LEN(STR$(KE)));
93 PRINT#1, L; SPC(8-LEN(STR$(L)));
94 PRINT#1, DM; SPC(8-LEN(STR$(DM)));
95 PRINT#1, L*100/(KE-K0): K0=KE: GOT050
100 PRINT "GESAMT-KM "; K0-KA: K0=K0-KA
110 PRINT "BENZIN-L "; SL
120 PRINT "BENZIN-DM "; SD
130 PRINT "L/100KM "; 100*SL/K0
140 PRINT "DM/L "; SD/SL
150 PRINT "PF/KM "; 100*SD/K0
170 IF NOT P THEN 200
180 PRINT#1, "GESAMT-KM "; K0
181 PRINT#1, "BENZIN-L "; SL
182 PRINT#1, "BENZIN-DM "; SD
183 PRINT#1, "L/100KM "; 100*SL/K0
184 PRINT#1, "DM/L "; SD/SL
185 PRINT#1, "PF/KM "; 100*SD/K0
200 S0=0: SM=0: REM OEL
```

```

210 READO: IFO<0THEN250
220 READDM: SM=SM+DM: SO=SO+O: GOT0210
250 PRINT "OEL-L          ";SO
260 PRINT "OEL-DM          ";SM
270 PRINT "L/1000KM        ";1000*SO/K0
280 PRINT "OEL-PF/KM       ";100*SM/K0
281 IFNOTPTHEN300
285 PRINT#1,"OEL-L          ";SO
286 PRINT#1, "OEL-DM          ";SM
287 PRINT#1, "L/1000KM        ";1000*SO/K0
288 PRINT#1, "OEL-PF/KM       ";100*SM/K0
300 SO=0: REM SONSTIGES: VERS.,REPAR.,ABSCHREIBUNG
310 READO: IFO<0THEN350
320 SO=SO+O: GOT0310
350 PRINT "SONSTIGES-DM ";SO:SD=SD+SM+SO
400 PRINT "GESAMT-KM      ";K0
410 PRINT "GESAMT-DM      ";SD
415 PRINT "DM/MONAT        ";SD/12
420 PRINT "GESAMT-PF/KM   ";SD*100/K0
430 PRINT
440 IFNOTPTHENEND
450 PRINT#1,"SONSTIGES-DM ";SO
451 PRINT#1,"GESAMT-KM      ";K0
452 PRINT#1, "GESAMT-DM      ";SD
453 PRINT#1, "DM/MONAT        ";SD/12
454 PRINT#1, "GESAMT-PF/KM   ";SD*100/K0
460 PRINT#1,"XDM":CLOSE1
470 REM
471 REM REIHENFOLGE IN DATA-ZEILEN:
472 REM 1. ANFANGS-KM-STAND
473 REM 2. TANKUNGEN (KM,LITER,DM)
474 REM 3. OEL (LITER,DM)
475 REM 4. SONSTIGES (DM)
476 REM POSTEN 2-4 KOENNEN MEHRERE
477 REM EINZELBETRAEGE ENTHALTEN,
478 REM JEMEILS MIT NEGATIVER ZAHL
479 REM ABSCHLIESSEN!
480 REM
500 DATA88302
510 DATA88605,41.54,35.27
515 DATA88926,37.97,32.23
520 DATA89267,40.04,37.40
525 DATA89553,40.00,34.75
530 DATA89855,33.20,28.89
535 DATA90114,29.14,25.32
540 DATA90261,15.70,13.49
545 DATA90578,27.73,24.38
550 DATA91014,39.14,34.79
555 DATA91406,41.05,35.67
560 DATA91711,34.32,30.17
565 DATA91949,23.46,21.10
570 DATA92221,22.52,20.25
575 DATA92562,29.30,26.32
580 DATA92943,38.20,34.29
585 DATA93283,32.87,29.22
590 DATA93656,37.01,33.64
595 DATA94046,39.08,35.53
600 DATA94542,41.68,38.73
605 DATA94952,36.42,34.20
610 DATA95341,36.98,34.35
615 DATA95637,27.20,25.55

```

620 DATA96057,40,25,38,20  
625 DATA96357,29,07,27,89  
630 DATA96680,29,98,29,27  
635 DATA96957,23,24,23,47  
640 DATA97441,40,23,40,63  
645 DATA97840,31,90,32,25  
650 DATA98120,21,49,22,38  
655 DATA98603,33,88,29,80  
660 DATA98983,34,41,34,77  
665 DATA99434,36,93,37,30  
670 DATA99909,40,21,40,61  
675 DATA100383,41,20,41,62  
680 DATA100807,41,20,41,63  
685 DATA101092,29,10,29,40  
690 DATA101550,38,18,38,56  
695 DATA101790,22,47,22,69  
700 DATA102072,21,01,21,22  
705 DATA102530,42,09,42,00  
710 DATA102904,41,94,42,36  
715 DATA103310,35,84,36,20  
720 DATA103633,30,26,30,57  
725 DATA103921,23,70,23,96  
730 DATA104342,40,35,40,76  
735 DATA104714,40,08,40,48  
740 DATA105033,39,59,39,98  
745 DATA105249,26,47,27,00  
750 DATA105556,25,98,26,49  
755 DATA105943,35,21,35,91  
800 DATA106023, 9,09, 9,18:REM LEIHWAGEN  
810 DATA-1: REM ENDE BENZIN  
820 DATA3,19,95,2,5,20,3,19,95,3,19,95  
821 DATA-1: REM ENDE OEL  
825 REM SONSTIGES: REPARATUREN,STEUER,  
826 REM VERSICHERUNG,ABSCHREIBUNG  
830 DATA5,2,796,71,50,4,50,138,2,8,234  
835 DATA6,26,5,9,132,8,8,132,8,88,95  
840 DATA37,6,18,9,8,05,255,47,7,28  
845 DATA124,47,270,4,7,1  
850 DATA188 : REM CA. STEUER  
855 DATA1060 : REM ABSCHREIBUNG  
860 DATA-1  
READY.

## NOTIZEN

# Word Special

## WORD SPECIAL

Es handelt sich hier um eine erweiterte und verbesserte Version des Word Processing Program (WPP) von Connecticut microComputer. Das Programm läuft auf den Commodore Computern der Serien 2000 und 3000, falls diese mindestens 16 K Speicher haben.

Sämtliche Druckroutinen des Programms sind auf den Adcomp-X80 ausgelegt. Sie lassen sich jedoch mit wenig Aufwand auf jeden anderen Drucker umschreiben. Das Programm berücksichtigt auch die unterschiedlichen Grafik-ROMs bei den verschiedenen CBM-Versionen. So muss man anfangs einfach die Art der Tastatur des Computers eingeben:

Business (2) – Tastatur = 2  
Grafik (1/3) – Tastatur = 1

Das Programm korrigiert dann automatisch sämtliche Ausgaben auf den Bildschirm, sowie sämtliche Eingaben von der Tastatur. Ferner wandelt es beim CBM-2001 die Zeichen-Codes um, so daß normal die Kleinbuchstaben und mittels Shift die Grossbuchstaben zur Verfügung stehen. Allerdings muß der Drucker auf den entsprechenden Computer-Typ umgeschaltet werden (mittels Dil-Schalter Nr. 6 auf der Steuerungsplatine des Druckers). Dabei ist die Adresse des Druckers mittels der Dil-Schalter 1 – 5 auf die Adresse 5 einzustellen. Andernfalls muß man in sämtlichen OPEN-Befehlen in den folgenden Zeilen die Geräteadressen ändern:

8600 – 8900  
17330  
2400 – 24999

Als nächstes gibt man dann die Uhrzeit in der Form hhmms ein. Will man die Zeitvariable TI\$ nicht ändern, oder ist die schon eingegeben worden und man fängt nach einem BREAK mit RUN neu an, so drückt man einfach Return.

Will man das Programm nur auf einem Rechner einsetzen, so sind folgende Änderungen vorzunehmen:

a) Grafik-Tastatur:  
Zeile 23000 – 24100 löschen  
8660 REM  
1099 REM  
Zeile 120 – 180 löschen  
150 TA=1

b) Business-Tastatur:  
Zeile 120 – 180 löschen  
150 TA=2  
1099 GOTO 23000  
Zeile 1100 – 1415 löschen  
8660 GOTO 24000  
Zeile 8700 – 8790 löschen

Dies spart Speicherplatz. Dies kann besonders den 'kleinen' 16K-Maschinen vorteilhaft sein.

Im folgenden sei '\$' die Abkürzung für ein Space.

Besonderheiten gegenüber Original WPP

1. Blinkender Cursor
2. Fehlermeldungen in Inverse
3. Repeat-Taste (Graffik-Tastatur: RVS-Taste  
Business-Tastatur: ← Taste  
Nach Drücken dieser Taste wird die zuletzt gedruckte Taste solange wiederholt, solange die Repeat-Taste gedrückt wird.)
4. Programm läuft unverändert auf dem 2001 und dem 3001 egal, ob Sie eine Grafik- oder eine Business-Tastatur haben Sie müssen nur anfangs die Art der Tastatur eingeben.
5. Cursor hoch und runter, links und rechts werden als Eingabe nicht anerkannt.
6. Bei Eingaben die länger als 150 Zeichen sind, erfolgt automatisch eine Warnung, d. h. jedes folgende Zeichen wird auf dem Bildschirm invers dargestellt. Bei mehr als 187 Zeichen

nimmt das Programm keine weiteren Zeichen mehr an. Ausnahmen sind Delete und Return.

Gleiche Befehle wie WPP:

c\$/S1/S2/ Tauscht String S1 gegen String S2 in der entsprechenden Zeile  
d\$N Addiert N zum Zeilenzähler  
er\$N Löscht N Zeilen von der aktuellen Zeile aus  
f FRE(0): gibt den verbleibenden Speicherbereich an  
g\$N Bringt Zeilenzähler auf N  
i Eintritt in Input mode  
l Lädt Textfile von Band  
m\$N1\$N2 Versetzt den Textblock von N1 bis N2 hinter die aktuelle Zeile  
p\$N1\$N2\$N3 Druckt N1 Zeilen, beginnend mit der aktuellen Zeile, N3 mal auf Gerät N2 aus  
r Ersetzt aktuelle Zeile durch die als nächste einzugebende Zeile  
s Nimmt Textfile auf Kassette auf  
t\$N1\$N2 Schreibt N1 Textzeilen, beginnend mit der aktuellen Zeile, auf Gerät N2 aus  
u\$N Subtrahiert N vom Zeilenzähler  
x Druckt die Zeilenlänge der aktuellen Zeile aus

Gleiche Direktiven wie WPP

.ce\$S Zentriert und druckt den String S  
.ii\$N Setzt Zeilenlänge auf N  
.lm\$N Druckt N Spaces als linken Rand aus  
.ls\$N Druckt nach jeder gedruckten Zeile N Leerzeilen aus  
.nl  
.pa Wartet bis irgendeine Taste gedruckt wird  
.rc\$N Wiederholt den String der folgenden Zeile N mal  
.sk\$N Druckt N Leerzeichen aus  
. Druckt und löscht den Ausgabe-Puffer  
\$S Druckt und löscht Ausgabe-Puffer, bringt ein Space und den String S in den Ausgabe-Puffer

Zusätzliche und veränderte Befehle:

w watch: druckt Uhrzeit auf Bildschirm aus  
E\$N Eingabe von Adressenfiles für automatischen Briefkopf-Ausdruck und Aufnahme auf Kassette N  
NE Löscht gesamten Text im Speicher  
I\$\$\$N1\$\$\$Name Lädt Textfile Name von Gerät N1  
I\$\$\$\$name Lädt Textfile von Kassette 1  
s\$N1\$N2\$N3\$NName Nimmt Textfile von Zeile N1 an N3 mal auf Gerät N3 unter dem Namen Name auf.  
s\$\$\$\$Name Nimmt gesamtes Textfile unter dem Namen Name auf Kassette 1 auf.

Neue Direktiven:

.du\$N Fertigt von jeder gedruckten Seite N Duplikate an.  
.du keine Duplikate  
.pg\$N Führt nach jeweils N gedruckten Zeilen automatisch einen Seitenvorschub aus, trennt die Seiten durch eine Punktlinie und nummeriert bei mehr als einer Seite die Seiten.  
.pg kein Seitenvorschub  
.ng stellt Drucker auf Normal-Kleinschrift um  
.nk stellt Drucker auf Normal-Grafik-schrift um  
.bg stellt Drucker auf Breit-Grafik-schrift um  
.bk stellt Drucker auf Breit-Kleinschrift um  
.at druckt automatisch Briefkopf aus und holt Adressen von Band 1  
.ae Briefkopfausdruck mit Adresseneingabe von Tastatur, soll alte Adresse erhalten bleiben, einfach RETURN drücken  
.wa\$N druckt und löscht Ausgabe-Puffer, druckt N Spaces, dann Uhrzeit

16 REM 10795 BYTES  
18 REM  
19 REM  
90 POKE59500,14

```

95 IFPEEK(50003)=0THENPOKE59490,62
120 W9=167:IFPEEK(50003)=0THENW9=548
150 PRINT"GRAFIK-ROM:(NOR/SHIFT)POKE14 POKE12"
151 PRINT" 1)> KLEIN/GROSS GROSS/GRAF."
152 PRINT" 2)> GROSS/KLEIN GROSS/GRAF."
153 OPENS,0:IL$="" :GOSUB6040:TA=VAL(IL$):CLOSES
155 IFIL$="" THEN150
160 IFVAL(IL$)<>1ANDVAL(IL$)<>2THEN150
170 TA=VAL(IL$)
180 CLOSES
190 IFTA=1THENPRINT"UHRZEIT (HHMMSS):":IL$="" :OPENS,0:GOSUB6040:CLOSES
191 IFTA=2THENPRINT"UHRZEIT (HHMMSS):":IL$="" :OPENS,0:GOSUB6040:CLOSES
192 IFIL$="" THEN200
193 IFLEN(IL$)<>6THEN190
194 TI$=IL$
200 Y=255
205 W9=167:IFPEEK(50003)=0THENW9=548
209 IFPEEK(50003)=0THENIFY>255THENY=255
210 DIMT$(Y),P(2):NU=5:PRINT" "
1000 A$="" :B$="" :C$="" :D$="" :D=0:CLOSE5:CLOSE3:CLOSE1:XL=0:IFL<0THENL=0
1005 IFTA=1THENPRINT"TOP-BOUNDARY ";Y;"|| "
1006 IFTA=2THENPRINT"TOP-BOUNDARY ";Y;"|| "
1010 LK=0:TT=NOTTT:IFTT=-1THENIFFRE(0)<650THEN16800
1020 PRINT:PRINT" " :IL$="" :OPENS,0:GOSUB6040:CLOSES
1050 IFIL$="" GOTO1000
1090 S2=0:IM$=LEFT$(IL$,1):GOSUB20000:TP=TP-1
1095 IFER=1THENGOSUB16850:GOTO1000
1099 IFTA=2THEN23000
1100 IFIM$="C"GOTO4000
1105 IFIM$="X"THENPRINT"LEN(T$(L)):GOTO1000
1110 IFIM$="F"THENPRINT"FRE(0)|| BYTES FREE":GOTO1000
1112 IFIM$="W"THENPRINT"UHRZEIT "LEFT$(TI$,2)" H ";
1113 IFIM$="W"THENPRINT" MID$(TI$,3,2)" M "RIGHT$(TI$,2)" S":GOTO1000
1115 IFIM$="E"THEN30000
1120 IFIM$="G"THENPRINT:L=P(0):GOSUB62000:GOTO1000
1130 IFIM$="I"THENOPENS,0:GOTO6000
1140 IFIM$="L"THEN11500
1150 IFIM$="M"GOTO16000
1160 IFIM$="D"GOTO13000
1170 IFIM$="P"THENGOSUB8000:GOTO1000
1180 IFIM$="R"GOTO15000
1190 IFIM$="S"GOTO11000
1200 IFIM$="T"GOTO7000
1210 IFIM$="U"GOTO14000
1400 IM$=LEFT$(IL$,2)
1410 IFIM$="ER"THENGOTO5000
1415 IFIM$="NE"THENCLR:GOTO90
1500 PRINT:PRINT"SYNTAX ERROR!":GOTO1000
4000 LL=LEN(IL$)
4010 FORN1=2TOLL-2:D$=MID$(IL$,N1,1):IFD$="" :GOTO4130
4020 NEXTN1:GOTO1500
4130 N1=N1+1:D$=MID$(IL$,N1,1)
4200 FORN=N1+1TOLL
4220 IFMID$(IL$,N,1)=D$GOTO4300
4230 NEXTN
4240 N=LL+1
4300 A$=MID$(IL$,N1+1,N-N1-1)
4400 IFN>LLTHENB$="" :GOTO4600
4420 FORN3=N+1TOLL
4430 IFMID$(IL$,N3,1)=D$GOTO4500
4440 NEXTN3

```

```

4450 N3=LL+1
4500 B$=MID$(IL$,N+1,N3-N-1)
4600 LL=LEN(A$):IFLEN(T$(L))<LLGOTO1500
4620 FORN=1TOLEN(T$(L))-LL+1
4630 IFMID$(T$(L),N,LL)=A$GOTO4700
4640 NEXTN:GOTO1500
4700 L5=LEN(T$(L))-N-LL+1:IFN=1THENA$="":GOTO4720
4710 A$=LEFT$(T$(L),N-1)
4720 IFL5=0THENT$(L)="":GOTO4800
4730 T$(L)=RIGHT$(T$(L),L5)
4800 T$(L)=A$+B$+T$(L)
4810 IFT$(L)=" "THENC=1:GOTO5050
4850 PRINT:GOSUB63000:PRINT:GOTO1000
5000 C=P(0):IFC<1THENC=1
5010 IFL+C>YTHENC=Y-L
5050 FORN=LTOY-C
5070 T$(N)=T$(N+C):T$(N+C)="":NEXTN
5080 FORN=Y-C+1TOY:T$(N)="":NEXTN
5090 IFT$(L)=" "GOTO7140
5100 GOSUB63000:GOTO1000
6000 KT=0:IFL=YTHEN16900
6005 IFT$(L)<>" "THENL=L+1
6010 IFTP=0THENL=0
6020 IFTA=1THENPRINT"¿?INPUT MODE."
6021 IFTA=2THENPRINT"¿?INPUT MODE."
6025 IFL>YTHEN16900
6026 IFFRE(0)<650THEN16800
6030 PRINT"¿L";"ID¿";IL$=""
6033 IFKT=5THEN1000
6035 GOSUB6040:GOTO6170
6040 POKEW9,0
6041 IFXM=0THENGET#5,C$
6042 IFXM=1THENGET#1,C$
6043 IF(ST)AND64THENPOKEW9,1:PRINT"■ ":KT=5:RETURN
6044 IFPEEK(59410)=254THENC$=R$
6045 IFC$=""GOTO6041
6046 R$=C$:CS=ASC(C$)
6050 IFCS=13ORCS=141THENPOKEW9,1:PRINT"■ ":RETURN
6055 IFLEN(IL$)>187ANDCS<>20THEN6041
6060 IFC$="□"THENIL$="" :PRINT"□":RETURN
6070 IFC$="▣"THENIL$="" :PRINT" " :RETURN

6080 IFC$="■"THENC$=""
6090 IFCS=20GOTO6400
6095 IFCS<30OR(CS<158ANDCS>130)THEN6041
6100 IFD<>0ORTA=2GOTO6145
6110 CT=CSAND127:CU=CSAND128
6120 IF(CT<65)OR(CT>90)GOTO6145
6130 IFCU=0THENC$=CHR$(CSOR128)
6140 IFCU=128THENC$=CHR$(CSAND127)
6145 IFLEN(IL$)>150THENPRINT"¿";
6146 IFLEN(IL$)>187THENPRINT:PRINT"¿?STRING TOO LONG ERROR!":GOTO6040
6150 IL$=IL$+C$:PRINTC$;:IFC$=CHR$(34)THENPRINTC$;:PRINT"■ ■";
6160 GOTO6041
6170 IFIL$=""THENL=L-1:GOTO1000
6250 GOSUB26100
6260 PRINT
6300 GOTO6025
6400 IFLEN(IL$)=0GOTO6040
6410 IFLEN(IL$)=1THENIL$="" :GOTO6430
6420 IL$=LEFT$(IL$,LEN(IL$)-1)
6430 PRINT"■ ■■":GOTO6040

```

```

7000 C=P(0):PRINT:D=P(1):IFD=0THEND=3
7010 IFD>=33THENGOSUB16850:GOTO1000
7040 OPEN5,D
7050 FORL=LTOL+C-1
7060 IFL>YGOTO7140
7100 IFT$(L)="GOTO7140
7120 GOSUB63100:NEXTL
7140 L=L-1:GOTO1000
8000 D=P(1):IFD=0THEND=3
8010 OPEN5,D,14
8100 F=P(0):IFF=0THENF=Y
8120 FORN1=1TOP(2):D1=0:GG=0:GA=0
8125 PP=0:PB=0
8150 A$="":LL=79:LM=0:LG=0:QX=1
8210 FORN=0TOL+F-1
8220 GETC$:IFC$<>" "THENN1=999:GOTO8410
8230 IFT$(N)="GOTO8400
8240 L$=LEFT$(T$(N),1)
8250 IFL$="," THENGOSUB8600:GOTO8390
8260 IFN<LGOTO8390
8270 IFL$=" " THENC$=A$:GOSUB8900:A$=T$(N):GOTO8320
8300 IFA$=" " THENA$=T$(N):GOTO8320
8310 A$=A$+" "+T$(N)
8320 IFLEN(A$)<LLGOTO8390
8330 GOSUB8500
8335 IFLS=0THENLS=LL
8340 C$=LEFT$(A$,LS):GOSUB8900
8355 IFLEN(A$)=LSTHENA$="":GOTO8390
8360 A$=RIGHT$(A$,LEN(A$)-LS):GOTO8320
8390 NEXTN
8400 C$=A$:GOSUB8900:IFPB=0THENPRINT#5:PRINT#5:GOTO8400
8401 FORQI=PPTOPB:PRINT#5:NEXTQI
8402 IFQI=1THENPRINT#5,TAB(LL/2-5)"(QX" ):PRINT#5:PRINT#5
8403 IFPB=0THEN8405
8404 FORQI=1TOLL-1:PRINT#5,".":NEXTQI:FORQI=1TOS:PRINT#5:NEXTQI
8405 IFD1>0THENND1=D1-1:IFD1>0THENGOTO8125
8410 NEXTN1:RETURN
8500 FORLS=LLTO1STEP-1
8520 IFMID$(A$,LS,1)=" " THENRETURN
8540 NEXTLS:RETURN
8600 IFA$<>" " THENC$=A$:A$="":GOSUB8900
8620 LB=LEN(T$(N))
8630 IFLB<3THENRETURN
8640 C$=MID$(T$(N),2,2)
8650 IL$=T$(N):GOSUB20000:I=P(0)
8660 IFTA=2THEN24000
8700 IFC$="CE"GOTO9300
8720 IFC$="LL" THENLL=I:IFLL=0THENLL=79
8730 IFC$="LM" THENLM=I
8735 IFC$="DU" THENIFGG=0THEND1=I:GG=5
8740 IFC$="SK" THENFORJ1=1TOI:GOSUB8960:NEXTJ1:RETURN
8745 IFC$="PG" THENPB=I:IFI=0THENPB=0
8750 IFC$="RC"GOTO9500
8753 IFC$="NG" THENCLOSE5:OPEN5,D,12:POKE59500,12:IFLL%>0THENLL=LL%
8754 IFC$="NK" THENCLOSE5:OPEN5,D,14:POKE59500,14:IFLL%>0THENLL=LL%
8755 IFC$="BG" THENCLOSE5:OPEN5,D,16:POKE59500,12:LL%=LL:LL=LL/2
8756 IFC$="BK" THENCLOSE5:OPEN5,D,18:POKE59500,14:LL%=LL:LL=LL/2
8760 IFC$="LS" THENLI=I
8770 IFC$="PA"GOTO9600
8775 IFC$="AT"ORC$="AE" THENIFD1>0THENIFGG=5THENIFGA=5THEN17000
8780 IFC$="AT" THENGOSUB18000:GA=5

```

```

8781 IFC$="RE" THEN GOSUB 18500:GA=5
8785 IFC$="WA" THEN DN=I:GOSUB 22000
8790 IFC$="NL" THEN XL=1
8890 RETURN
8900 IFC$="" THEN RETURN
8910 IFFP>=PB-5 THEN GOSUB 40000
8920 K=S+LM:S=0
8930 IFK<0 THEN FOR J=1 TO K:PRINT#5," ";:NEXT J
8950 PRINT#5,C$:IFXL=1 THEN XL=0:PRINT#5,CHR$(13):PP=PP+1:RETURN
8960 FOR J=1 TO LG:PRINT#5:PP=PP+1:NEXT J:RETURN
8980 RETURN
9300 IFN<L THEN RETURN
9310 W=LEN(T$(N))-4:IFW<1 THEN RETURN
9330 C$=RIGHT$(T$(N),W)
9340 FORM=1 TO W:IF MID$(C$,M,1)=CHR$(8) THEN W=W-2
9345 NEXT M
9350 S=(LL-W)/2:IFS<0 THEN S=0
9360 GOTO 8920
9500 IFN<L THEN RETURN
9510 N=N+1:IFT$(N)="" THEN RETURN
9520 C$="":C=P(0):IFC=0 THEN C=LL
9530 C$=C$+T$(N):W=LEN(C$):IFW<CGOTO 9530
9540 GOTO 9350
9550 IFC<0 THEN FOR J=1 TO C:PRINT#5," ";:NEXT J
9560 GOTO 8950
9600 GETC$:IFC$="" GOTO 9600
9610 RETURN
11000 N1=P(0):D=P(1):N3=P(2):IFD=0 THEN D=1
11005 IFF(1)>=33 THEN GOSUB 16850:GOTO 1000
11007 FOR O0=1 TO N3
11008 PRINT"WRITING "NA$
11010 OPEN5,D,1,NA$
11120 FOR N=1 TO Y
11130 IFT$(N)="" GOTO 11200
11140 PRINT#5,T$(N):T$=T$(N):GOSUB 11300
11150 NEXT N
11200 CLOSE5
11210 NEXT O0
11250 GOTO 1000
11300 SZ=SZ+LEN(T$)+1
11320 IFSZ<192 THEN RETURN
11330 POKE59411,53
11340 T=TI
11350 IF(TI-T)<50 GOTO 11350
11360 POKE59411,61
11370 SZ=SZ-191
11400 RETURN
11500 D=P(1):IFD=0 THEN D=1
11501 IFD>=33 THEN GOSUB 16850:GOTO 1000
11510 PRINT"SEARCHING FOR "NA$
11520 OPEN5,D,0,NA$:GOTO 6000
13000 C=P(0):PRINT:IFC=0 THEN C=1
13010 L=L+C:GOSUB 62000:GOTO 1000
14000 C=P(0):PRINT
14200 IFC=0 THEN C=1
14210 L=L-C:IFL<0 THEN L=0:TP=1
14300 GOSUB 63000:GOTO 1000
15000 PRINT"█L:█D█":IL$="":OPEN5,0:GOSUB 6040
15010 IFIL$="" GOTO 15200
15100 T$(L)=IL$
15200 CLOSE5:GOSUB 63000:GOTO 1000

```

```

16000 F=P(0):C=P(1)
16001 IFFRE(0)<<(C-F)*180THEN50000
16010 IFC<FTHENC=F
16020 IFT$(F)="GOTO16900
16030 IFT$(C)="GOTO16900
16050 IFL>FGOTO16500
16100 J=1:IFTP=0THENJ=0
16105 FORN1=FTOC
16110 IL#=T$(C)
16120 FORN=CTOL+1+JSTEP-1:T$(N)=T$(N-1):NEXTN
16200 T$(L+J)=IL$
16210 NEXTN1
16300 GOTO1000
16500 FORN1=FTOC
16510 IL#=T$(F)
16520 FORN=FTOL-1:T$(N)=T$(N+1):NEXTN
16600 T$(L)=IL$
16610 NEXTN1
16700 GOTO1000
16800 IFTA=1THENPRINT:PRINT"??OUT OF MEMORY ERROR!":PRINT"??READY.":GOTO1000
16801 IFTA=2THENPRINT:PRINT"??OUT OF MEMORY ERROR!":PRINT"??READY.":GOTO1000
16850 IFTA=1THENPRINT:PRINT"??ILLEGAL QUANTITY ERROR":PRINT"??READY.":RETURN
16851 IFTA=2THENPRINT:PRINT"??ILLEGAL QUANTITY ERROR":PRINT"??READY.":RETURN
16900 IFTA=1THENPRINT:PRINT"??OUT OF RANGE ERROR!":PRINT"??READY.":GOTO 1000
16901 IFTA=2THENPRINT:PRINT"??OUT OF RANGE ERROR!":PRINT"??READY.":GOTO 1000
17000 FORQI=1T06:PRINT#5:NEXTQI
17010 XX=0:IFLL<55THENL8=LL:LL=79:XX=5
17110 IFTA=1THENPRINT#5,"JUERGEN RUED /PF.1325 /D-7800 FREIBURG
17111 IFTA=2THENPRINT#5,JUERGEN RUED /PF.1325 /D-7800 FREIBURG
17120 PRINT#5,"
-----
17200 IFA1$=""THEN17220
17210 PRINT#5,A1$;
17220 IFTA=1THENPRINT#5,TAB(LL-25-LEN(A1$))"FREIBURG,DEN "DA$
17221 IFTA=2THENPRINT#5,TAB(LL-25-LEN(A1$))"FREIBURG,DEN "DA$
17230 IFA2$=""THEN17250
17240 PRINT#5,A2$;
17250 IFTA=1THENPRINT#5,TAB(LL-17-LEN(A2$))"JUERGEN RUED
17251 IFTA=2THENPRINT#5,TAB(LL-17-LEN(A2$))"JUERGEN RUED
17260 IFA3$=""THEN17280
17270 PRINT#5,A3$;
17280 IFTA=1THENPRINT#5,TAB(LL-17-LEN(A3$))"EISENBAHNSTR.66
17281 IFTA=2THENPRINT#5,TAB(LL-17-LEN(A3$))"EISENBAHNSTR.66
17310 IFTA=1THENPRINT#5,TAB(LL-17)"D-7800 FREIBURG
17311 IFTA=2THENPRINT#5,TAB(LL-17)"-7800 FREIBURG
17320 IFA4$=""THEN17350
17330 OPEN1,D,18:PRINT#1,A4$:CLOSE1
17340 GOTO17360
17350 PRINT#5
17360 DN=LL-17:GOSUB22000
17370 FORQI=1T03:PRINT#5:NEXTQI:PRINT#5,"——":PRINT#5:PRINT#5:PRINT#5
17380 PP=PP+20
17385 IFXX=5THENLL=L8
17390 RETURN
18000 D=P(0):IFLK=0THENOPEN3,D,0:LK=1:GOSUB18200
18010 INPUT#3,A1$:IFA1$=""ENDE"THENLK=0:CLOSE3:N1=999:N=999:RETURN
18020 INPUT#3,A2$,A3$,A4$
18030 GOTO17000
18200 OPEN1,0:XM=1:BB=D:D=0
18205 PRINT"??DATUM : "DA$ " ";IL$="":GOSUB6040:IFIL$<>"?THENDA$=IL$
18208 CLOSE1:D=BB:XM=0
18210 RETURN
18500 IFTA=1THENPRINT"??ADRESSEX"

```

```

18501 IFTA=2THENPRINT"ADRESSEX"
18505 OPEN1,0:XM=1:BB=D:D=0
18507 IFTA=1THENIL$="" :PRINT"ADATUM" : ; DA$ " " ; GOSUB6040:IFIL$<>"THENDA$=IL
18508 IFTA=2THENIL$="" :PRINT"ADATUM" : ; DA$ " " ; GOSUB6040:IFIL$<>"THENDA$=IL
18510 IFTA=1THENIL$="" :PRINT"ANAME" : ; A1$ " " ; GOSUB6040:IFIL$<>"THENA1$=IL
18511 IFTA=2THENIL$="" :PRINT"ANAME" : ; A1$ " " ; GOSUB6040:IFIL$<>"THENA1$=IL
18512 IFTA=1THENIL$="" :PRINT"AN" : ; A2$ " " ; GOSUB6040:IFIL$<>"THENA2$=IL
$
18513 IFTA=2THENIL$="" :PRINT"AN" : ; A2$ " " ; GOSUB6040:IFIL$<>"THENA2$=IL
$
18514 IFTA=1THENIL$="" :PRINT"ASTRASSE" : ; A3$ " " ; GOSUB6040:IFIL$<>"THENA3$=IL
18515 IFTA=2THENIL$="" :PRINT"ASTRASSE" : ; A3$ " " ; GOSUB6040:IFIL$<>"THENA3$=IL
18516 IFTA=1THENIL$="" :PRINT"AWOHNORT" : ; A4$ " " ; GOSUB6040:IFIL$<>"THENA4$=IL
18517 IFTA=2THENIL$="" :PRINT"AWOHNORT" : ; A4$ " " ; GOSUB6040:IFIL$<>"THENA4$=IL
18520 CLOSE1:D=BB:XM=0
18540 GOTO17000
20000 NA$="" :ER=0:L1=0:P(0)=0:P(1)=0:P(2)=0
20100 FORK=0T03
20110 L1=L1+1
20120 IFL1=LEN(IL$)GOTO20300
20130 P1$=MID$(IL$,L1,1):IFASC(P1$)<>32GOTO20110
20150 L2=L1
20160 L2=L2+1
20170 IFL2=LEN(IL$)GOTO20210
20180 P1$=MID$(IL$,L2,1)
20190 IFASC(P1$)<>32GOTO20160
20210 IFK<3THENP(K)=VAL(MID$(IL$,L1,L2-L1+1))
20215 IFK=3THENNA$=MID$(IL$,L1,L2-L1+1):IFLEN(NA$)>16THENNA$=LEFT$(NA$,16)
20220 NEXTK
20230 IFP(1)>15THENER=1
20300 RETURN
22000 IFTA=1THENQ$=LEFT$(TI$,2)+" H "+MID$(TI$,3,2)+" M "+RIGHT$(TI$,2)+" S"
22001 IFTA=2THENQ$=LEFT$(TI$,2)+" H "+MID$(TI$,3,2)+" M "+RIGHT$(TI$,2)+" S"
22010 PRINT#5,TAB(DN):Q$:PP=PP+1
22020 RETURN
23000 IFIM$="F"THENPRINT"FRE(O) III BYTES FREE":GOTO1000
23001 IFIM$="C"GOTO4000
23002 IFIM$="W"THENPRINT"WOHRZEIT "LEFT$(TI$,2) H ";
23003 IFIM$="W"THENPRINT"WOHRZEIT "MID$(TI$,3,2) M "RIGHT$(TI$,2) S":GOTO1000
23005 IFIM$="X"THENPRINT"X"LEN(T$(L)):GOTO1000
23015 IFIM$="E"THEN30000
23020 IFIM$="G"THENPRINT:L=P(0):GOSUB62000:GOTO1000
23030 IFIM$="I"THENOPEN5,0:GOTO6000
23040 IFIM$="L"THEN11500
23050 IFIM$="M"GOTO16000
23060 IFIM$="D"GOTO13000
23070 IFIM$="P"THENGOSUB8000:GOTO1000
23080 IFIM$="R"GOTO15000
23090 IFIM$="S"GOTO11000
23098 IFIM$="T"GOTO7000
23099 IFIM$="U"GOTO14000
23100 IM$=LEFT$(IL$,2)
23110 IFIM$="ER"THENGOTO5000
23115 IFIM$="NE"THENCLR:GOTO90
23200 GOTO1500
24000 IFC$="CE"GOTO9300
24020 IFC$="LL"THENLL=I:IFLL=0THENLL=79
24030 IFC$="LM"THENLM=I
24035 IFC$="DU"THENIFGG=0THEND1=I:GG=5
24040 IFC$="SK"THENFORJ1=1TO1:GOSUB8960:NEXTJ1:RETURN
24045 IFC$="PG"THENPB=I:IFI=0THENPB=0

```

```

24050 IFC$="RC"GOTO9500
24053 IFC$="NG" THENCLOSE5:OPENS,D,12:IFLLX>0THENLL=LLX
24054 IFC$="NK" THENCLOSE5:OPENS,D,14:IFLLX>0THENLL=LLX
24055 IFC$="BG" THENCLOSE5:OPENS,D,16:LLX=LL:LL=LL/2
24056 IFC$="BK" THENCLOSE5:OPENS,D,18:LLX=LL:LL=LL/2
24060 IFC$="LS" THENLG=1
24070 IFC$="PA"GOTO9600
24075 IFC$="AT"ORC$="AE" THENIFD1>0THENIFG0=5THENIFGA=5THEN17000
24080 IFC$="AT" THENGOSUB18000:GA=5
24081 IFC$="AE" THENGOSUB18500:GA=5
24085 IFC$="WA" THENDN=I:GOSUB22000
24090 IFC$="NL" THENXL=1
24190 RETURN
26100 FORN=YTOL+1STEP-1
26110 T$(N)=T$(N-1)
26120 NEXT
26200 T$(L)=IL$:L=L+1:RETURN
30000 IFTA=1THENPRINT"☐☐ FILE - EINGABE"
30001 IFTA=2THENPRINT"☐☐ FILE - EINGABE"
30002 IFTA=1THENPRINT"☐NAME=(RETURN)> EINGABEENDE☐☐"
30003 IFTA=2THENPRINT"☐NAME=(RETURN)> EINGABEENDE☐☐"
30005 XM=1:BB=D:D=0:SZ=0
30010 OPENS,1,1,NA$:OPEN1,D
30020 IFTA=1THENIL$="":PRINT"☐NAME"
30021 IFTA=2THENIL$="":PRINT"☐NAME"
30025 IFA1$="":THEN300200
30030 IL$="":PRINT"☐"
30040 IFTA=1THENIL$="":PRINT"☐STRASSE"
30041 IFTA=2THENIL$="":PRINT"☐STRASSE"
30050 IFTA=1THENIL$="":PRINT"☐WOHNORT"
30051 IFTA=2THENIL$="":PRINT"☐WOHNORT"
30100 PRINT#3,A1$:T$=A1$:GOSUB11300
30102 PRINT#3,A2$:T$=A2$:GOSUB11300
30103 PRINT#3,A3$:T$=A3$:GOSUB11300
30104 PRINT#3,A4$:T$=A4$:GOSUB11300
30105 PRINT:PRINT:PRINT
30200 CLOSE1:CLOSE3:D=BB:XM=0
31000 GOTO1000
40000 IFPB=0THENRETURN
40005 PRINT#5:PRINT#5:PRINT#5,TAB(LL/2-5)"(QX" )"
40010 PRINT#5:PRINT#5:PRINT#5:PRINT#5:FORQI=1TOLL-1:PRINT#5,".":NEXTQI:PRINT#5
40015 PRINT#5:PRINT#5:PRINT#5:PRINT#5:PP=0:QX=QX+1:RETURN
62000 IFL>YTHENL=Y
62010 IFL=0THENTP=1:GOTO63000
62020 IFT$(L)="":THENL=L-1:GOTO62010
63000 OPENS,3
63100 GOSUB8980:PRINT#5,"☐L";"ID☐";T$(L):RETURN
READY.

```

Für weitere Informationen empfehle ich Ihnen:

ELCOMP die Fachzeitschrift für Microcomputer Nr. 10/1979, Seite 118

sowie das Originalprogramm Wortprozessor Best.-Nr. 426 zu 96,- DM

zu beziehen durch: Ing. W. Hofacker GmbH, Postfach 75 437, 8000 München 75

Eine Cassette mit dem hier gezeigten Programm kann vom Verfasser direkt gegen eine Gebühr bezogen werden: Juergen Rued, Eisenbahnstr. 66, 7800 Freiburg

## NOTIZEN

# Laufschrift

## LAUFSCHRIFT

Dieses Programm eignet sich bestens zur Erzeugung von Großschrift auf einem Drucker. Das Programm ist auf Dvice 4 ausgelegt. Der Text wird nicht horizontal, sondern vertikal ausge-

geben. Daher ist es zweckmäßig dieses Programm nur mit einem Drucker zusammen zu verwenden.

```
100 DIM QQ$(100),QW$(100),QE$(100),QR$(100),QT$(100)
110 FORA=32TO96
120 READQQ$(A),QW$(A),QE$(A),QR$(A),QT$(A)
130 NEXTA
140 GOTO8000
150 CLOSE5:INPUT"JTEXTHOEHE (7-80):";DF
160 IFDF<7ORDF>80THEN150
170 DF=INT(DF/7)
180 GOSUB5000
200 PRINT"JTEXT<NICHT MEHR ALS 255 ZEICHEN. ["CHR$(34)"]NICHT VERWENDEN!
!):"
204 INPUTRT$
205 IFRT$=""THEN204
207 IFCV=1THENFORA=1TO10:PRINT:NEXTA
208 IFCV=2THENFORA=1TO10:PRINT#4:NEXTA
210 FORA=1TOLEN(RT$)
220 T$=MID$(RT$,A,1)
230 Q=ASC(T$)
240 IFQ=222ORQ=255THENQ=96:GOTO245
242 IFQ<32ORQ>95THENQ=32:GOTO245
245 GOSUB400
246 IFCV=2THENPRINTT$;
250 IFCV=1THENFORTY=1TODF:PRINT:NEXTTY
255 IFCV=2THENFORTY=1TODF:PRINT#4:NEXTTY
260 NEXTA
261 IFCV=1THENFORA=1TO10:PRINT:NEXTA
262 IFCV=2THENFORA=1TO10:PRINT#4:NEXTA
265 GOTO150
400 ER$=QQ$(Q):GOSUB500
410 ER$=QW$(Q):GOSUB500
420 ER$=QE$(Q):GOSUB500
430 ER$=QR$(Q):GOSUB500
440 ER$=QT$(Q):GOSUB500
450 RETURN
500 FORRE=1TODF-1
510 FORW=1TOLEN(ER$)
520 AS$=MID$(ER$,W,1)
524 IFAS$=" "THEN530
526 IFCB=2THENAS$="■"
527 IFCB=1THENAS$=T$
530 IFCV=1THENFORAS=1TODF:PRINTAS$;:NEXTAS
535 IFCV=2THENFORAS=1TODF:PRINT#4,AS$;:NEXTAS
540 NEXTW
545 IFCV=1THENPRINT
```





## NOTIZEN

# LC Kreisberechnung

## KREISBERECHNUNG

```
100 PRINT"*****L-C KREISBERECHNUNG*****"
110 PRINT"DIESES PROGRAMM BERECHNET L- UND C WERTE"
120 PRINT"RESONANZFREQUENZ UND SPULE-ABMESSUNG "
130 PRINT"BITTE WAHLEN SIE AUS FOLGENDES:"
140 PRINT"-1 FREQUENZ BERECHNUNG"
150 PRINT"-2 KAPAZITÄT BERECHNUNG"
160 PRINT"-3 INDUCTIVITÄT BERECHNUNG"
170 PRINT"-4 SPULE-ABMESSUNG"
180 PRINT"-5 ENDE"
190 GETB$:IFB$=""THEN190
200 IFB$="1"THEN260
210 IFB$="2"THEN360
220 IFB$="3"THEN520
230 IFB$="4"THEN670
240 IFB$="5"THEN910
250 PRINT":GOTO130
260 PRINT"*****FREQUENZ BERECHNUNG*"
270 INPUT"INDUCTIVITÄT IN MICROHENRY";L
280 INPUT"CAPACITÄT IN PICO-FARAD";C
290 F=1/(2*PI*SQR(L*C*1E-6))
300 REM FÜR MEHR GENAUE WERTEN 1000 IN 140 ANDEREN
310 FF=F*1000:F=INT(FF+.5)/1000
320 PRINT"*****"
330 PRINT"*****F"MHZ*"
340 PRINT"WAHLEN SIE BITTE NEUE NUMMER":GOTO190
350 END
360 PRINT"*****KAPAZITÄT BERECHNUNG*"
370 B=0
380 INPUT"FREQUENZ IN MHZ";F
390 INPUT"INDUCTIVITÄT IN MICRO-HENRY";L
400 A=1/(2*PI*F*1E6)
410 C=(A^2)/(L*1E-6)
420 IFINT(C)<1THENB=B+1:C=C*1E3:GOTO420
430 IFB=0THENCC$="F"
440 IFB=1THENCC$="MF"
450 IFB=2THENCC$="UF"
460 IFB=3THENCC$="NF"
470 IFB=4THENCC$="PF"
475 IFB>=5THENPRINT"NICHT PRAKTISCH !!!!":GOTO500
480 C=C*10:C=INT(C+.5):C=C/10
490 PRINT"*****C;CC$*"
500 PRINT"WAHLEN SIE BITTE NEUE NUMMER":GOTO190
510 END
520 PRINT"*****INDUCTIVITÄT BERECHNUNG*"
530 B=0
540 INPUT"FREQUENZ IN MHZ";F
550 INPUT"KAPAZITÄT IN PF";C
560 A=1/(2*PI*F*1E6)
570 L=(A^2)/(C*1E-12):GOTO580
580 IFINT(L)<1THENB=B+1:L=L*1E3:GOTO580
```

```

590 IFB=0THEN A$="H"
600 IFB=1THEN A$="MH"
610 IFB=2THEN A$="UH"
620 IFB=3THEN A$="NH"
630 IFB=4THEN A$="PH"
640 L=L*10:L=INT(L):L=L/10
650 PRINT"XXXXXXXXXXXXXXXXXXXX"LA$" "
660 PRINT"XXXXXXXXXXXXXXXXXXXXWAHLEN SIE BITTE NEUE NUMMER":GOTO190
670 PRINT"XXXXXXXXXXXXXXXXXXXXSPULE-ABMESSUNG"
680 INPUT"XXXXXXXXXXXXXXXXXXXXINNENDURCHMESSER IN CM";D
690 INPUT"XXXXXXXXXXXXXXXXXXXXINDUCTIVITÄT IN UH";L
700 LE=1.75*D
710 N=SQR(L*(9*D+20*LE)/(.197*D^2))
720 N=10*N:N=INT(N+.5)/10
730 DD=(LE/(2*N))*10
740 DD=100*DD:DD=INT(DD+.5)/100
750 PRINT"D"
760 PRINT"
770 PRINT"
780 PRINT"
790 PRINT"
800 PRINT"
810 PRINT"
820 PRINT"
830 PRINT"
840 PRINT"
      N4
      N3
      N2
      N1
      ID
      LE"
850 PRINT:PRINT:PRINT" L= INDUCTIVITÄT          ";L;" UH"
860 PRINT:PRINT" D= INNENDURCHMESSER";D;" CM"
870 PRINT" N= ANZAHL WINDUNGEN          ";N
880 PRINT" LE= SPULE LÄNGE";LE;" CM"
890 PRINT" DD= MAX. DRAHTDURCHMESSER ";DD;" MM"
900 PRINT"XXXXXXXXXXXXXXXXXXXXWAHLEN SIE BITTE NEUE NUMMER":GOTO190

```

# Roulette Auswertung

## ROULETTE AUSWERTUNG

```
1 REM * ROULETTPROGRAMM ZUR AUSWERTUNG VON SYSTEMSPIELEN *
2 REM * MIT ERRECHNUNG DER ZEROVERLUSTE *
3 REM * EINFACHE CHANCEN GRUNDMUSTER *
4 REM * MIT PROGRESSION +3 -5 *
5 REM * GRAPHISCHE ANZEIGE DER SERIEN VON SCHWARZ, ROT UND ZERO *
6 REM * ERSCHEINT EINE FARBE MEHR ALS EINMAL, SO WIRD DIESE FARBE NACHGESETZT
9 REM * ERSTELLT AUF CBM 3032,880 BYTES OHNE REM
10 CLR
20 Z=100
30 DIM A(36)
50 FOR I=1 TO 18
60 READ R,N
70 A(N)=1:A(R)=2
80 NEXT I
100 DATA 1,2,3,4,5,6,7,8,9,10,12,11,14,13,16,15,18,17
110 DATA 19,20,21,22,23,24,25,26,27,28,30,29,32,31,34,33,36,35
150 KA=100
160 MA=KA:MI=KA
200 GOTO 450
250 S=X
300 KA=KA-S
310 IF KA<MI THEN MI=KA
450 SP=SP+1
500 W=INT(RND(1)*37)
510 F(A(W))=F(A(W))+1
520 GOSUB 950
530 IF W=0 THEN KA=KA+S/2:VL=VL+S/2:GOTO 850
550 REM * VERGLEICH *
560 IF A(W)=F1 GOTO 710
570 IF A(W)=F2 GOTO 710
600 REM * VERLUST - NEUER SATZ *
650 X=X-5
660 IF X<0 THEN X=0
690 GOTO 800
700 REM * GEWINN - NEUER SATZ *
710 KA=KA+2*S
720 IF KA>MA THEN MA=KA
750 X=X+3
760 IF X>10 THEN X=3
800 ON A(W) GOSUB 980,990
820 PRINT TAB(22)KA
850 IF SP<Z GOTO 250
890 GOTO 5010
900 REM * UNTERPROGRAMME *
950 IF W=0 THEN PRINT SP TAB(8)S TAB(13)W TAB(16)"ZERO"TAB(22)KA+S/2:RETURN
960 PRINT SP TAB(8)S TAB(12)W TAB(16+A(W))"*":RETURN
980 F1=1:F2=0:RETURN
990 F2=2:F1=0:RETURN
5000 REM * ERGEBNISSE AUSDRUCKEN *
```

```

5010 PRINT
5020 PRINT"SPIELE      = "SP
5030 PRINT"SCHWARZ    = "F(1)
5040 PRINT"ROT        = "F(2)
5050 PRINT"ZERO       = "F(0)
5060 PRINT"ZEROVERL.  = "VL
5070 PRINT"KAPIT.MIN. = "MI
5080 PRINT"KAPIT.MAX. = "MA
5090 PRINT"KAPITAL    = "KA
7000 REM * ----- *
7010 REM * VARIABLENLISTE *
7020 REM A(N) = NOIR = SCHWARZ
7030 REM A(R) = ROUGE = ROT
7040 REM F(A(W)) = ZAEHLER FARBE+ZERO
7050 REM F1 UND F2 = VARIABLEN FUER
7060 REM FARBENVERGLEICH
7070 REM KA = KAPITAL
7080 REM MA = MAX. KAPITAL
7090 REM MI = MIN. KAPITAL
8000 REM S = SATZ PRO SPIEL
8010 REM SP = SPIELE
8020 REM VL = VERLUSTE DURCH ZERO
8030 REM W = WURF
8040 REM X = VARIABLE ZUR ERRECHNUNG
8050 REM DER EINSATZE
8060 REM Z = ZAHL DER SPIELE
8070 REM KA UND Z KOENNEN ZU BEGINN
8080 REM BELIEBIG GESETZT WERDEN
9000 REM ERKLAERUNG BILDSCHIRMAUSGABE
9010 REM 1. SPALTE = SPIELNUMMER
9020 REM 2. SPALTE = EINSATZ
9030 REM 3. SPALTE = GEFALLENE ZAHL
9040 REM 4. SPALTE = GRAPHIK
9050 REM 5. SPALTE = KAPITAL
READY.

```

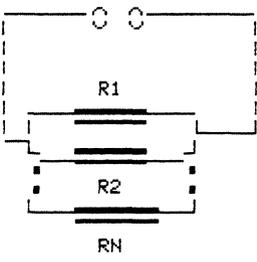
# Serien- und Parallelschaltung

## PARALLEL- UND SERIENSCHALTUNG

```

500 PRINT"WOELLEN SIE ..."
510 PRINT"00A) PARALLELSCHALTUNG ?"
520 PRINT"00B) SERIENSCHALTUNG ?"
530 PRINT"00C) AUFHOEREN ?"
540 GETA$: IFA$="A"THEN1000
550 IFA$="B"THEN5000
560 IFA$="C"THEN15000
570 GOT0540
1000 REM+++++
1005 PRINT"Q"

```



```

1010 PRINT"
1015 PRINT"
1020 PRINT"
1025 PRINT"
1030 PRINT"
1035 PRINT"
1038 PRINT"
1040 PRINT"
1045 PRINT"
1050 PRINT"
1055 PRINT"
1058 PRINT"
1060 PRINT"
1065 PRINT"
1070 FORX=1T01000:NEXT
1075 INPUT"WIEVIELE WIDERSTAEENDE 0####";Z:DIMA(Z)
1080 FORX=1TOZ
1084 IFR1<0ORR2<0THEN58000
1085 PRINT"NR"X"=""; INPUTA(X)
1086 IFA(X)<0THEN48000
1090 NEXT
1095 X=1
1096 IFA(X)=0THEN37000
1097 IFX=ZTHEN1100
1098 X=X+1:GOT01096
1100 T=0
1110 FORX=1TOZ:T=T+(1/A(X)):NEXT
1111 T=1/T
1120 PRINT"GESAMTWERSTAND="T" OHM
1900 GOT062000
4999 END
5000 REM=====

```

```

5001 PRINT"Q"
5002 PRINT"
5011 PRINT"
5012 PRINT"
5013 PRINT"
5022 PRINT"
5023 PRINT"
5032 PRINT"
5033 PRINT"

```



```

5042 PRINT"
5052 PRINT"
5062 PRINT"
5072 PRINT"
5082 PRINT"
5092 PRINT"
5100 INPUT"XXXWIEVIELE WIDERSTRENDE ●■■■";N
5105 FORX=1TON
5120 PRINT"XR";X;"=";:INPUTA(X)
5124 IFA(X)<0THEN48000
5125 NEXT
5130 FORX=1TON:ZZ=ZZ+A(X):NEXT
5140 PRINT"XXGESAMMTWIDERSTAND=";ZZ
6000 FORA=1TO5000:NEXT:RUN
11000 IFA(X)<0THEN
15000 REM*****
15001 END
48000 PRINT"XXSIDIDIOT■
48001 PRINT"X ES GIBT DOCH KEINE NEGATIVEN
48002 PRINT"X WIDERSTRENDE"
48005 PRINT"XX":FORZZ=1TO5000:NEXT:RUN
58000 PRINT"XXSIDIDIOT■
58001 PRINT"X ES GIBT DOCH KEINE NEGATIVEN
58002 PRINT"X WIDERSTRENDE"
58005 PRINT"XX":GOTO1075
59000 PRINT"XXR1■+R2■ MUESSEN UNGLEICH NULL■ SEIN":PRINT"XX":GOTO1075
62000 FORZ=1TO2000:NEXT
62001 PRINT"XX NOCHMAL ? (J/N)
62005 GETA$: IFA$="J"THENRUN
62010 IFA$="N"THENEND
62015 GOTO62005
READY.

```

# Rechnungsschreiben

## RECHNUNGSSCHREIBEN

Das nachfolgende Programm zum Schreiben von Rechnungen wurde auf einem CBM 3001 mit 16K entwickelt und getestet. Es arbeitet zusammen mit einem Drucker (# 4) und den neutralen Rechnungsformularen von Redysoft. Es ist wichtig, daß Sie einen Drucker angeschlossen haben. Ansonsten müssen die entsprechenden PRINT-Befehle abgeändert werden.

Die Produkte befinden sich in DATA-Statements. Dort stehen auch die Informationen über Preis und MwSt. Rabatt und Versandkosten werden in Abhängigkeit von der Bestellmenge ermittelt. Das Programm druckt automatisch einen Paketaufkleber mit aus. Das Rechnungs-

datum wird nur zu Beginn einmal eingegeben. Es wird bei den folgenden Rechnungen automatisch eingesetzt. Die Rechnungsnummer wird nach jedem Durchlauf um 1 erhöht.

Das Rechnungsschreibprogramm ist für das nachfolgend gezeigte Musterformular ausgelegt. Das Musterformular ist jedoch verkleinert. Es soll Ihnen als Anregung dienen. Die Formulare können vom Hofacker Verlag unter der Best.-Nr. 6206 bezogen werden. Es sind jeweils 2000 Blatt in einem Karton verpackt (Original plus die Kopien – NCR Papier). Sie dienen Ihnen in der Übergangsphase, bis Sie sich vom Drucker Ihre eigenen mit Firmenkopf versehenen Formulare besorgt haben.

```
100 REM C BY ING. HOFACKER GMBH 1980
110 DIM T<10,2>
130 INPUT"RECHNUNGSdatum EINGEBEN":RD#
140 INPUT"ERSTE RECHNUNGSNUMMER EINGEBEN":I1
150 INPUT"KUNDENNUMMER EINGEBEN":C1
160 IF C1=0 THEN 1400
170 INPUT"NAME DES KUNDEN EINGEBEN":S1#
180 INPUT"STRASSE UND NR.":S2#
190 INPUT"POSTLEITZAHL UND ORT":S3#
200 OPEN#4,CMD4
210 PRINT#4,TAB<12>:"ING. W. HOFACKER GMBH, TEGERNSEERSTR. 18, 8150
HOLZKIRCHEN
220 PRINT#4;PRINT#4;PRINT#4
230 PRINT#4,TAB<5>;C1
240 PRINT#4,TAB<5>;S1#
250 PRINT#4,TAB<5>;S2#
260 PRINT#4,TAB<5>;S3#
270 PRINT#4,TAB<42>;I1;TAB<18>;RD#
280 PRINT#4;PRINT#4
290 PRINT#4
300 PRINT#4;PRINT#4;PRINT#4
320 GOSUB 740
330 GOSUB 500
340 GOTO 320
350 S1=I
360 IFS1>0ANDS1<GTHEND=.25
370 IFS1>5ANDS1<11THEND=.33
380 IFS1>10THEND=.40
390 REM S1=MENGE U=KOSTEN FUER VERSAND UND VERPACKUNG
400 IFS1<15THENU=2
410 IFS1>14 ANDS1<30THENU=2.5
420 IFS1>29ANDS1<50 THENU=3
```

```

430 IFS1>49ANDS1<100THENU=4
440 IFS1>99THENU=5
450 REM NEXT LINE INCREMENTS INVOICE NUMBER
460 I1=I1+1
470 R=INT(D*100+.5)
480 GOTO 560
500 G=0/1.065;A=G;GOSUB1090;H#=D*;L2=L
510 A=0;GOSUB 1090
520 G#=D*;L1=L
525 C=S1*G
530 A=C;GOSUB 1090
535 J#=STR$(N1);K#=STR$(S1)
540 PRINT#4,TAB(5);N1;TAB(6-LEN(J*));S1;TAB(10-LEN(K*));E$.
545 PRINT#4,TAB(32-LEN(E*)-L1);G*;TAB(13-LEN(G*)-L2);H*;TAB(15-LEN(H*)-L);D*
548 C=S1*0;T=T+G
550 RETURN
560 D1=0*T
570 D1=INT(D1*100+.5)/100
580 PRINT#4
590 A=D1;GOSUB 1090
600 PRINT#4,TAB(33);R;TAB(2);"%RABATT= -";TAB(22-L);D$
610 FORP=1TO(24-ZS);PRINT#4;NEXT
620 A=T-D1;GOSUB 1090
630 G#=D*;L1=L
640 F=T-D1+U;M1=F*.065
650 A=M1;GOSUB 1090
660 PRINT#4,TAB(54-L);D$;" 6,5%";TAB(10-L1);G$
670 Y=U*.065;A=Y;GOSUB 1090;G#=D$
680 PRINT#4;PRINT#4;PRINT#4
690 A=U;GOSUB 1090
700 PRINT#4,TAB(50);G$;" 6,5%";TAB(10-L);D$
710 A=M1+F+Y;GOSUB 1090
720 PRINT#4,TAB(71-L);D$
730 GOSUB 950;GOTO 150
740 INPUT"WELCHE BESTELLNUMMER";N1
750 IF N1=0 THEN 350
760 RESTORE
770 READ N,E$,O
780 IFN=1THEN 810
790 IF N=N1 THEN 830
800 GOTO 770
810 PRINT"BESTELLNUMMER NICHT GEFUNDEN"
820 GOTO 740
830 INPUT"WIEVIEL STUECK VON DIESER BESTELLNUMMER?";S1
835 GOSUB 1300
840 I=I+S1;ZS=ZS+1
850 RETURN
860 REM HIER KOMMEN DIE PRODUKTE HINEIN
865 DATA 20,"OPERATIONSVERSTAERKER",19.80
868 DATA 113,"BASIC-PROGRAMMIERHANDBUCH",19.80
870 DATA 1000,"57 PROGRAMME IN BASIC",35.00
875 DATA 110,"PET PROGRAMMIERHANDBUCH",29.80
880 DATA 31,"57 PROGR. I. BASIC DEUTSCH",39.00
885 DATA 104,"1000 ELEKTRONIK SCHALTG",49.00
888 DATA 112,"PASCAL-PROGRAMMIERHANDBUCH",29.80
890 DATA 34,"TINY BASIC HANDBUCH",19.80
895 DATA 101,"CB-HANDBUCH F. HOBBYF.",19.80
900 DATA 109,"6502 PROGRAMMIERHANDBUCH",29.80
910 DATA 110,"PET PROGRAMMIERHANDBUCH",29.80
920 DATA 1,"1",0
930 RETURN

```

```

940 END
950 PRINT#4:PRINT#4
960 PRINT#4,"ING. W. HOFACKER GMBH"
970 PRINT#4:PRINT#4,"TEGERNSEERSTR. 18"
980 PRINT#4:PRINT#4,"8150 HOLZKIRCHEN"
990 FORZU=1TO9
1000 PRINT#4
1010 NEXTZU
1020 PRINT#4,TAB(30);S1$
1030 PRINT#4:PRINT#4,TAB(30);S2$
1040 PRINT#4:PRINT#4:PRINT#4
1050 PRINT#4,S3$:TAB(30);S3$
1060 PRINT#4:PRINT#4:PRINT#4:PRINT#4:PRINT#4
1070 ZS=0
1080 RETURN
1090 K1$="0"
1100 K2$="",00"
1120 B=INT(A*100+.5)/100
1130 D1$=STR$(B)
1140 D0=B*10
1150 D2=INT(D0)
1160 D3=D0-D2
1170 IF D3<0.09THEN 1200
1180 D$=D1$
1190 GOTO 1260
1200 D4=INT(B)
1210 D5=B-D4
1220 IF D5<0.09 THEN 1250
1230 D$=D1$+K1$
1240 GOTO1260
1250 D$=D1$+K2$
1260 L=LEN(D$)
1270 RETURN
1300 FORR1=1TO10
1310 IFT(R1,1)=N1ORT(R1,1)=0THEN1350
1320 NEXTR1
1350 T(R1,1)=N1
1360 T(R1,2)=T(R1,2)+S1
1370 RETURN
1400 PRINT#4:PRINT#4:PRINT#4
1410 PRINT#4,"BESTELNR.":TAB(12):"VERKAUFT"
1415 PRINT#4
1420 FORR1=1TO10
1425 IFT(R1,1)=0THEN1450
1427 J$=STR$(T(R1,1))
1430 PRINT#4,TAB(2);T(R1,1);TAB(20-LEN(J$));T(R1,2)
1440 NEXTR1
1450 CLOSE4,4
1460 END
READY.

```

**Auftragsbestätigung und Rechnung**

Rechnungs-Nr.		Rechnungs-Datum	
Bestellung vom		Bestellung-Nr.	Vertreter
Rebatt			

Best.-Nr.	Best.-Menge	gel. Menge	Titel/Bezeichnung		Brutto-Verkaufspreis	Nettopreis pro Stück	Nettopreis Gesamt	MW. St.
			Lieferung über	Mehrwertsteuer		Warenwert Netto		
				Mehrwertsteuer		Warenwert Netto		
				Änderungen		Versendkostenanteil		
			<b>Rechnungsbetrag</b>					

Eigentumsvorbehalt nach § 455 BGB.  
Es gelten unsere Lieferungs- u. Zahlungsbedingungen

1 = ermäßigte MWSL  
 2 = volle MWSL  
 0 = keine MWSL

Abseiner

---



---

(Postleitzahl) (Ort)

Für kurze Mitteilungen an den Empfänger

---



---

Bitte Doppel der Aufschrift in die Sendung legen!

Bitte frei lassen zum Aufkleben des Paketnummernzettels

Vermerke über besondere Versandverfahren und Voraussetzungen

(Straße und Hausnummer oder Paketausgabe)

(Bestimmungsort)

Postleitzahl bitte besonders groß und auffällig angeben!

# Poly-Opti

## POLY-OPTI      Polynomapproximation

In Physik und Technik ist es oft nötig, durch eine Schar von Meßpunkten eine durchgehende Kurve zu ziehen, um nämlich  $y$ -Werte für Stellen auf der  $x$ -Achse zu bekommen, an denen kein Meßwert vorliegt. Oder die Meßwerte sind so ungenau, daß sie selbst gar keine glatte Kurve ergeben würden.

Mathematisch betrachtet braucht man also eine Funktion, die diese approximierende (nähernde) Kurve darstellt. Da keine einzelne bekannte Funktion alle vorkommenden Kurvenformen darstellen kann, muß man immer mehrere kombinieren. Wenn man sie linear kombiniert, d. h. jede mit einem eigenen Faktor  $c_j$  versieht und diese Produkte alle zusammenzählt, kann man auch eine Berechnungsvorschrift für die Koeffizienten  $c_j$  herausbekommen. Das geht so, daß man mathematisch ausrechnet, mit welchem Satz von Koeffizienten die Näherungskurve am genauesten durch die vorgegebenen Meßpunkte geht. Als Maß für die Genauigkeit nimmt man die Fehlerquadratsumme: Man nehme die Differenz aus dem  $y$ -Wert eines Meßpunkts und des Punkts auf der Näherungskurve an derselben  $x$ -Stelle, quadriere diese Differenz und addiere alle diese Quadrate von allen Meßpunkten zusammen. Mit Hilfe der Differentialrechnung erhält man Bedingungsgleichungen dafür, bei welchen  $c$ -Werten die Fehlerquadratsumme am kleinsten wird. Die Bedingungsgleichungen bilden ein lineares Gleichungssystem ( $m$  Gleichungen mit  $m$  Unbekannten), aus dem die Koeffizienten  $c_j$  errechnet werden können.

Bisher war die Rede von praktisch beliebigen Funktionen, die kombiniert werden sollen. Für ganz spezielle Fälle kann man sich tatsächlich Gedanken über eine sinnvolle Wahl der Funktionen machen, in der Alltagspraxis kommt man aber mit zwei Grundtypen aus:

a) bei periodischen Abläufen wählt man Sinus-

und Cosinus-Funktionen und ihre Oberwellen (Fourieranalyse),

b) sonst Polynome ( $1, x, x^2, x^3, x^4, \dots$ )

Das vorliegende Programm führt eine solche Polynomapproximation aus, indem es zu vorgegebenen Meßpunkten ein Polynom

$$Y = c_m + c_{m-1} x + c_{m-2} x^2 + \dots + c_0 x^m$$

findet, das möglichst genau durch die Punkte geht.

Nun haben aber Polynome einen Nachteil: Sobald der Grad  $m$  eines Polynoms groß wird, passiert es leicht, daß die Kurve nicht mehr schön glatt ist, sondern stark wellig ist (oszilliert). Von Dr. K. Dettmer stammt dazu der Vorschlag, sich nicht von vornherein auf einen bestimmten Polynomgrad festzulegen, sondern die Approximation mehrmals mit jeweils steigendem Grad durchzuführen, bis die Fehlerquadratsumme nicht mehr kleiner wird.

So kann man also bei diesem Programm einen minimalen (MA, mindestens = 1) und einen maximalen (MM) Polynomgrad angeben, zwischen denen das Optimum gesucht wird.

Außer diesen beiden Zahlen müssen noch die Punktzahl  $N$  und die Punkte selbst (jeweils mit  $x$ - und  $y$ -Wert) eingegeben werden.

Zur Sicherheit gegen Vertippen bei größeren Datenmengen können diese Eingaben auch als DATA-Zeilen ins Programm eingebaut und weggespeichert werden.

Das Programm ist weitestgehend selbsterklärend. Zu Beginn wird eine Beschreibung auf dem Schirm gelistet, die auch angibt, welche Variablen was enthalten, damit der Benutzer

gegebenenfalls selbst Zusätze einbringen kann. Z. B. kann es gewünscht sein, sich die optimierte Kurve am Schluß in Form einer auf die Spezielle Anwendung zugeschnittenen Tabelle auf einem Printer abdrucken zu lassen.

Als erste Eingabe wird vom Programm ein M oder D für manuelle bzw. DATA-Eingabe erfragt. Wird die manuelle Eingabe gewählt, werden die jeweiligen Daten in Klartext angefordert.

Sind alle Punkte eingegeben, werden in Zeilen 116 – 119 immer wieder gebrauchte Werte S(I) berechnet, die später die Matrizen und die rechten Seiten der Gleichungssysteme füllen. Ein Gleichungssystem wird nämlich geschrieben als: Matrix A mal Unbekanntenvektor C gleich Vektor B der rechten Seite.

Für jeden Polynomgrad (angefangen mit MA) wird dann der Koeffizientensatz C bestimmt (Zeile 130: Füllen – und Zeilen 300 – 340: Lösen des Gleichungssystems). Als Ergebnis werden der Grad und die Fehlerquadratsumme ausgedruckt. Wird letztere nicht mehr kleiner, wird der als vorletzter behandelte Grad als Optimum angesehen und weiterverwendet (Zeilen 690 – 700).

Zum Schluß werden in Tabellenform die Koeffizienten C0(I) und die "korrigierten" Punkte Y0(I) ausgedruckt, das sind die y-Werte des optimalen Polynoms zu denselben X(I)-Werten wie die eingegebenen Meßpunkte.

Im Programm ist am Ende ein Beispieldatensatz eingebaut, dessen Punkte der Gleichung  $y = x^3$  gehorchen, was durch einen Programmablauf bestätigt werden kann (D für DATA-Eingabe eintippen).

Falls Sie ab Zeile 810 noch eigene Zusätze einbauen wollen, gebe ich hier an, wie Sie aus den Koeffizienten C0(I) und einem X-Wert einen Y-Wert nach dem Horner-Schema errechnen können:

Y=C0(0) : FOR I=1 TO M : Y=Y \*X + C0(I) : NEXT

Wegen der endlichen Größe des RAM-Speichers kann der PET nicht unendlich viele Meßpunkte verkraften, in der 8K-Version sind es jedoch

immerhin schon etwa 100 (Variable N), bei größerem RAM-Speicher entsprechend mehr. Das gilt allerdings nur, wenn der maximale Polynomgrad MM nicht größer als 15 ist, sonst werden es noch viel weniger.

Beim alten PET 2001 darf MM aber sowieso nie größer als 15 werden, weil eine Matrix A mit MMxMM Elementen gebraucht wird, und sowas beim alten PET nie mehr als 256 Elemente haben darf (Index 0 mitzählen). Daher darf beim alten PET auch die Punktzahl N (selbst bei eingebauter Speichererweiterung) den Wert 255 nicht übersteigen.

Als weitere Einschränkung ist nur noch anzuführen, daß der minimale Polynomgrad MA immer mindestens 1 sein muß (gibt Geradengleich, "lineare Regression") und daß der maximale Grad MM immer mindestens um 1 kleiner als die Punktzahl N sein muß.

#### Programm POLY-OPTI

Polynomausgleich mit Optimierung des Polynomgrades.

Eingeben sind der minimale Grad MA, der maximale Grad MM und die Anzahl N der Meßpunkte.

Anschließend werden die einzelnen Punktkoordinaten angefordert.

Ab Zeile 810 können weitere Rechnungen angefügt werden. Die eingegebenen Koordinaten stehen in X und Y und die korrigierten Ordinaten in Y0 (Indices jeweils 1, . . . ,N).

Die optimierten Polynomkoeffizienten stehen in C0 (Indices 0, . . . ,M, 0 bei höchster Potenz M).

Eingabe manuell (M) oder aus DATA-Zeilen (D) ? D

MA, MM, N: 1 4 5

Eingegebene Punkte (I,X(I), Y(I)):

1	0	0
2	1	1
3	2	8
4	-1	-1
5	3	27

weiter mit beliebigem Tastendruck

Grad, Fehlersumme: 1, 140,4

Grad, Fehlersumme: 1, 140,4

Grad, Fehlersumme: 3, 0

Grad, Fehlersumme: 4, 0

Optimaler Grad: 3  
 Fehlersumme: 0  
 Weiter mit beliebigem Tastendruck

Koeffizienten, (I, Potenz, C0(I)):  
 Nr. 0 X 3  $\Delta$  1  
 Nr. 1 X 2  $\Delta$  0  
 Nr. 2 X 1  $\Delta$  0  
 Nr. 3 X 0  $\Delta$  0

weiter mit beliebigem Tastendruck

Korrigierte Punkte, (I, X(I), Y0(I)):

1	0	0
2	1	1
3	2	8
4	-1	-1
5	3	27

```

10 PRINT"PROGRAMM # P O L Y - O P T I M I R U N G"
30 PRINT"POLYNOM AUSGLEICH MIT OPTIMIERUNG"
31 PRINT"DES POLYNOMGRADDES"
32 PRINT"EINZUGEBEN SIND DER MINIMALE GRAD MA, DER MAXIMALE GRAD MM UND DIE
33 PRINT"ANZAHL N DER MESSPUNKTE."
34 PRINT"ANSCHLIESSEND WERDEN DIE EINZELNEN"
35 PRINT"PUNKTKOORDINATEN ANGEFORDERT."
40 PRINT"AB ZEILE 810 KOENNEN WEITERE RECHNUNGEN"
41 PRINT"ANGEFUEGT WERDEN. DIE EINGEGEBENEN"
42 PRINT"KOORDINATEN STEHEN IN X UND Y"
43 PRINT"UND DIE KORRIGIERTEN ORDINATEN IN Y0"
44 PRINT"(INDICES JEWEILS 1,...,N)."
45 PRINT"DIE OPTIMISIERTEN POLYNOMKOEFFIZIENTEN"
46 PRINT"STEHEN IN C0 (INDICES 0,...,M ; 0 BEI"
47 PRINT"HOECHSTER POTENZ M)"
50 PRINT"EINGABE MANUELL (M) ODER AUS
51 INPUT"DATA-ZEILEN (D) ";M$
52 IFM$<"M"ANDM$<"D"THEN50
60 IFM$="D"THENREADMA,MM,N:PRINT"MA, MM, N: ";MA,MM,N:GOTO101
100 INPUT"MIN.GRAD, MAX.GRAD, PKTANZ.";MA,MM,N:PRINT
101 IFN<=MMTHENPRINT"GRAD ZU GROSS!" :END
102 DIMA(MM,MM),C(MM),C0(MM),B(MM),S(MM*3+1)
103 DIMY(N),Y0(N),Y1(N),X(N),XX(N)
104 W$="WEITER MIT BELIEBIGEM TASTENDRUCK"
105 IFM$="M"THEN110
106 PRINT"NEINGEGEBENE PUNKTE (I, X(I), Y(I)):"
107 FORI=1TO N:READX(I),Y(I):PRINTI;TAB(7)X(I);TAB(24)Y(I):NEXT:PRINTW$
108 GETM$:IFM$=""THEN108
109 GOTO116
110 FORI=1TO N:PRINT"PKT NR. ";I;TAB(13);"X,Y ";:INPUTX(I),Y(I):NEXT
116 FORI=1TO N:XX(I)=1:NEXT
117 FORI=0TO2*MM:S=0:FORJ=1TO N:S=S+XX(J):NEXT:S(I)=S:IFI>MMTHEN119
118 S=0:FORJ=1TO N:S=S+Y(J)*XX(J):NEXT:S(2*MM+1+I)=S
119 FORJ=1TO N:XX(J)=XX(J)*X(J):NEXT:NEXT:S(0)=N
120 M=MA-1:PRINT
125 M=M+1:IFM=MATHEN130
127 FORI=0TO M-1:C0(I)=C(I):NEXT
128 FORI=1TO N:Y0(I)=Y1(I):NEXT
129 IFM>MMTHENF0=F1:GOTO700
130 FORI=0TO M:FORJ=0TO M:A(I,J)=S(2*M-I-J):NEXT:B(I)=S(2*MM+1+M-I):NEXT
300 FORI=0TO M-1:FORJ=I+1TO M:F=A(J,I)/A(I,I)
330 FORK=I+1TO M:A(J,K)=A(J,K)-A(I,K)*F:NEXT:B(J)=B(J)-B(I)*F:NEXT:NEXT
400 C(M)=B(M)/A(M,M):FORI=1TO M:K=M-I:S=0
430 FORJ=K+1TO M:S=S+A(K,J)*C(J):NEXT:C(K)=(B(K)-S)/A(K,K):NEXT
600 IFM>MATHENF0=F1
610 F1=0:FORI=1TO N:S=C(0):FORJ=1TO M:S=S*X(I)+C(J):NEXT
650 Y1(I)=S:F1=F1+(S-Y(I))*(S-Y(I)):NEXT
680 PRINT"GRAD, FEHLERSUMME: ";M$," ";F1

```

```

690 IFM=MAORF1<F0THEN125
700 M=M-1:PRINT"OPTIMALER GRAD:";M
720 PRINT"FEHLERSUMME  ";F0:PRINTW$
725 GETM$:IFM$=""THEN725
730 PRINT"KOEFFIZIENTEN, (I, POTENZ, C0(I)):|"
750 FORI=0TOM:PRINT"NR. ";I;TAB(8);"X ↑";M-I;TAB(20);C0(I):NEXT:PRINTW$
780 GETM$:IFM$=""THEN780
790 PRINT"KORRIGIERTE PUNKTE, (I, X(I), Y0(I)):|"
800 FORI=1TON:PRINTI;TAB(7);X(I);TAB(22);Y0(I):NEXT
810 END
990 REM
991 REM REIHENFOLGE BEI DATA-EINGABE:
992 REM MIN.-, MAX.-GRAD, PUNKTANZAHL,
994 REM PUNKTE (JEWEIFS X,Y).
996 REM
998 REM BEISPIELDATEN:
1000 DATA1,4,5,0,0,1,1,2,8,-1,-1,3,27
READY.

```

# Druckaufbereitung

## DRUCKAUFBEREITUNG

### 1. Problemstellung

Ursprünglich konzipierte man die BASIC-Programmiersprache für technisch-wissenschaftliche Anwendungen. Inzwischen wurde bei zahlreichen Systemen – so z. B. bei CBM (PET) – ihr Sprachumfang so erweitert, daß mit BASIC auch zahlreiche kommerzielle Probleme einfach zu lösen sind.

Etwas größeren Aufwand verursacht (u. a.) noch die Druckaufbereitung, wenn es die Aufgabenstellung erfordert, Zahlen auf zwei Dezimalen genau (häufigster Fall: Geldbeträge) dezimalpunktgerecht untereinanderzuschreiben. Die Schwierigkeiten entstehen vor allem dadurch, daß in BASIC sowohl führende als auch nachgezogene Nullen unterdrückt werden. Mit dem unten aufgeführten Programm wollen wir eine Lösungsmöglichkeit aufzeigen.

### 2. Erläuterungen zum Programmablaufplan

#### 2.1 Hauptprogramm

Aus Gründen der besseren Überschaubarkeit untergliederten wir unser Programm in das HAUPTPROGRAMM und in die hieraus aufgerufenen UNTERPROGRAMME

UP-VORLAUF,  
UP-LESEN,  
UP-RUNDEN,  
UP-ERGAENZEN und  
UP-AUSGABE.

Bei Übernahme des Programms für einen konkreten Anwendungsfall empfiehlt es sich, die obigen Unterprogramme zu einem einzigen Modul zusammenzufassen (Vgl. Abb. 1: HAUPTPROGRAMM).

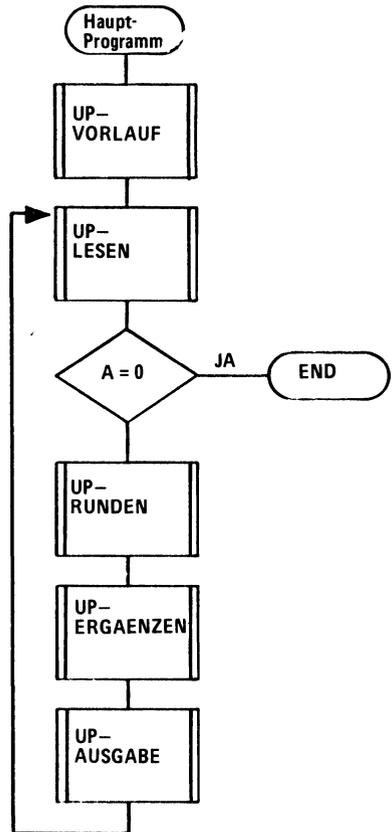


Abb. 1: Hauptprogramm

#### 2.2 UP-VORLAUF

Im noch zu behandelnden Unterprogramm UP-RUNDEN werden die auszugebenden Zahlen auf zwei Dezimalen genau gerundet. Da in BASIC –

wie wir oben schon ausführten – nachgezogene Nullen unterdrückt werden, können wir hinsichtlich des Erscheinungsbildes der Zahlen folgende drei Alternativen unterscheiden:

- 1) keine Ziffern nach dem Dezimalpunkt:  
Erscheinungsbild (z. B.) 100
- 2) 1 Ziffer nach dem Dezimalpunkt:  
Erscheinungsbild (z. B.) 100.7
- 3) 2 Ziffern nach dem Dezimalpunkt:  
Erscheinungsbild (z. B.) 100.73

Für die Druckausgabe muß bei Alternative (1) ".00" und bei Alternative (2) "0" ergänzt werden. Diese Zeichen definieren wir als Strings (K2\$ und K1\$) im Unterprogramm UP-VORLAUF. Bei Vorliegen der Alternative (3) erübrigt sich jede "Ergänzungsmaßnahme".

In Z legen wir fest, an welcher Schreibstelle die Zahl rechtsbündig ausgegeben werden soll (Vgl. Abb. 2: UP-VORLAUF).

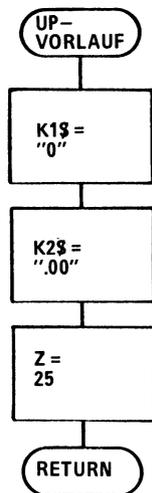


Abb. 2: UP-VORLAUF

### 2.3 UP-LESEN

Statt der in unserem Unterprogramm UP-LESEN aufgeführten Befehle (READ/DATA) wird man

bei kaufmännischen Anwendungen zweckmäßigerweise den INPUT-Befehl verwenden. Außerdem werden die für die Ausgabe aufzubereitenden Beträge meistens aufgrund von Rechenoperationen entstehen. Will man beim Einlesen keinen numerischen Wert als Endekriterium (in unserem Fall: A = 0; vgl. Abb. 1) verwenden, dann sind die Zahlen als String einzugeben und nach der EOF-Abfrage (z. B. A\$ = "/\*") in einen numerischen Ausdruck umzuwandeln (A = VAL(A\$)). (Vgl. Abb. 3: UP-LESEN).

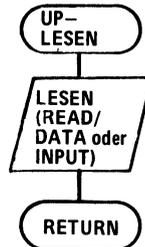


Abb. 3: UP-LESEN

### 2.4 UP-RUNDEN

Das "kaufmännische Runden" führen wir mit einem einzigen BASIC-Befehl (LET B = INT(A\*100+0.5/100) durch (vgl. Abb. 4: UP-RUNDEN). Die Wirkungsweise des Befehls sei an den folgenden zwei Beispielen erläutert.

1. Beispiel:

A (vor dem Runden)	=	17.9658	
A*100	→	1796.58	
+0.5	→	0.5	
		1797.08	
INT(A*100+0.5)	→	1797	
INT(A*100+0.5)/100	→	17.97	→ B

(B = kaufmännisch gerundete Zahl)

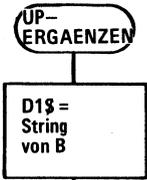
2. Beispiel:

A (vor dem Runden)	=	17.9648	
A*100	→	1796.48	
+0.5	→	0.5	
		1796.98	
INT(A*100+0.5)	→	1796	
INT(A*100+0.5)/100	→	17.96	→ B

(B = kaufmännisch gerundete Zahl)

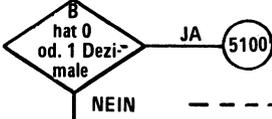
Befehle

5000  
5010  
5020

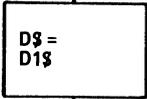


5030

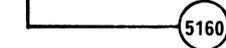
5040  
5050  
5060  
5070



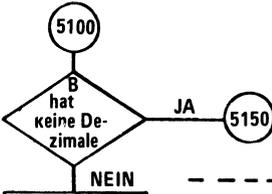
5080



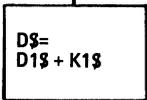
5090



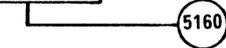
5100  
5110  
5120



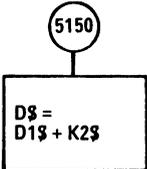
5130



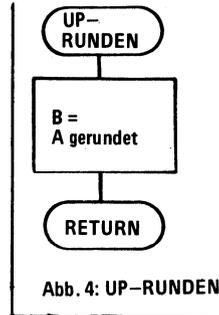
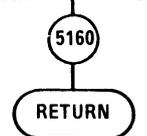
5140



5150



5160



[ d. h. B hat 2 Dezimale

[ d. h. für Zwecke der Ausgabe  
braucht keine Ergänzung vor-  
genommen zu werden.

[ d. h. B hat 1 Dezimale

[ Ergänzung einer "0"

[ Ergänzung von ".00"

Abb. 5: UP-ERGAENZEN

## 2.5 UP-ERGAENZEN

Im Unterprogramm UP-ERGAENZEN wird das "Aussehen" der auszugebenden Zahlen untersucht und ggf. ".00" oder "0" ergänzt. Die Vorgehensweise ergibt sich aus dem Programmablaufplan (vgl. Abb. 5: UP-ERGAENZEN).

Nur zu den Befehlen in den Zeilen 5070 und 5120 sei angemerkt, daß die Abfragen

"IF D3<0.09 ..." bzw.  
 "IF D5<0.09 ..."

verständlicher wären, wenn sie

"IF D3=0 ..." bzw.  
 "IF D5=0 ..."

lauten würden. Die logisch etwas weniger einschichtigen Abfragen mußten wegen systembedingter Ungenauigkeiten gewählt werden.

Statt der "arithmetischen Vorgehensweise" können für die Analyse der auszugebenden Zahlen auch ausschließlich String-Befehle verwendet werden. Das entsprechende Unterprogramm ist am Ende des Source-Listing aufgeführt. (Es ist aber zu beachten, das sich die in Abb. 5 aufgeführten Befehle/Zeilennummern auf die "arithmetische Version" beziehen.)

## 3. Source-Listing

```

10 REM PROGRAMM "DRUCKAUFBEREITUNG"
20 REM PROGRAMMIERSPRACHE: BASIC(CBM)
1000 REM ++++++
1010 REM +   HAUPTPROGRMM   +
1020 REM ++++++
1030 GOSUB 2000 : REM (UP-VORLAUF)
1040 GOSUB 3000 : REM (UP--LESEN)
1050 IF A = 0 THEN 1100
1060 GOSUB 4000 : REM (UP-RUNDEN)
1070 GOSUB 5000 : REM (UP-ERGAENZUNG)
1080 GOSUB 6000 : REM (UP-AUSGABE)
1090 GOTO 1040
1100 END
2000 REM ++++++
2010 REM +   UP-VORLAUF   +
2020 REM ++++++
2030 LET K1$ = "0"
```

## 2.6. UP-AUSGABE

Die Druckposition ergibt sich aus dem im Unterprogramm UP-VORLAUF festgelegten Wert Z abzüglich der in L gespeicherten Länge der Zahl (vgl. Abb. 6: UP-AUSGABE).



Abb. 6: UP-AUSGABE

## 2.7. Ergänzende Hinweise

Falls die Funktionsweise des Programms nicht durchschaut wird, empfiehlt es sich, mit vorher festgelegten Daten einen "Schreibtischtest" durchzuführen. Das Verständnis ergibt sich dann fast "von selbst". Hierbei wird natürlich angenommen, daß die Wirkungsweise der Befehle bekannt ist.

```

2040 LET K2$ = ".00"
2050 LET Z = 25
2060 RETURN
3000 REM ++++++
3010 REM +      UP-LESEN      +
3020 REM ++++++
3030 READ A
3040 RETURN
3050 DATA 1, 1., 2.3, 45.67, 7.891
3060 DATA 175.218, 300.005, 9.00123
3070 DATA 0
4000 REM ++++++
4010 REM +      UP-RUNDEN    +
4020 REM ++++++
4030 LET B = INT(A*100+0.5)/100
4040 RETURN
5000 REM ++++++
5010 REM +      UP-ERGAENZEN  +
5020 REM ++++++
5030 LET D1$ = STR$(B)
5040 LET D1 = B * 10
5050 LET D2 = INT(D1)
5060 LET D3 = D1 - D2
5070 IF D3 < 0.09 THEN 5100
5080 LET D$ = D1$
5080 GOTO 5160
5100 LET D4 = INT(B)
5110 LET D5 = B - D4
5120 IF D5 < 0.09 THEN 5150
5130 LET D$ = D1$ + K1$
5140 GOTO 5160
5150 LET D$ = D1$ + K2$
5160 RETURN
6000 REM ++++++
6010 REM +      UP-AUSGABE    +
6020 REM ++++++
6030 LET L = LEN(D$)
6040 PRINT TAB(Z-L)D$
6050 RETURN
READY.

```

#### Alternative für Unterprogramm

##### UP-ERGAENZEN

```

5000 REM ++++++
5010 REM +      UP-ERGAENZEN/ALTERN.  +
5020 REM ++++++
5030 LET D1$ = STR$(B)
5040 LET D2$ = RIGHT$(D1$,3)
5050 LET D3$ = LEFT$(D2$,1)
5060 IF D3$ = "." THEN 5110
5070 LET D4$ = RIGHT$(D1$,2)
5080 LET D5$ = LEFT$(D4$,1)
5090 IF D5$ = "." THEN 5120
5100 LET D$ = D1$ + K2$
5110 GOTO 5150
5120 LET D$ = D1$ + K1$
5130 GOTO 5150
5140 LET D$ = D1$
5150 RETURN
READY.

```

## NOTIZEN

# **ROM-Vergleichsliste PET/CBM und CBM/PET**

## **ROM-Vergleichsliste**

Die nachfolgende ROM-Vergleichsliste soll es Ihnen ermöglichen, Programme die für eine bestimmte ROM-Version geschrieben sind für eine andere ROM-Version umzuschreiben. Commodore hat bis heute vier verschiedene ROM-Versionen in seiner PET/CBM-Serie verwendet.

- 1. Version 1 (uralt) in den ersten PETs eingebaut**
- 2. Version 2 ROMs in der 2001 Serie bekannt unter alte ROMs**
- 3. Version 3 ROMs in der 3001 Serie bekannt als neue ROMs**
- 4. Version 4 Ende 1980/Anfang 1981**

**BASIC-Programme laufen grundsätzlich auf allen PETs/CBMs egal welches ROM die Maschinen enthalten.**

**Programme die jedoch irgend welche Maschinenprogramme oder PEEK, POKE oder SYS in die ZERO-Page enthalten, müssen geändert werden. Meist ist dies jedoch auch für den ungeübten Programmierer kein Problem.**

**Sie starten das Programm und verfolgen den Ablauf. Wo Fehler auftreten, sehen Sie nach. Wenn sich in dieser Zeile ein PEEK, POKE oder SYS in die ZERO-Page befindet schreiben Sie sich den Wert heraus, sehen in der Vergleichsliste nach und setzen den neuen Wert ein. Die ROM-Vergleichsliste bietet Ihnen die Möglichkeit von Version 2 auf Version 3 oder auch umgekehrt zu schreiben.**

## ROM-VERGLEICHSLISTE

## NEW - OLD PET CHARTS

As you probably already know the NEW PET computers are incompatible with the OLD PETs. Memory locations are used for different things in the NEW PETs. Since most decent PET programs took full advantage of the PET and used it's memory locations they now will NOT function correctly if at all on the NEW PET.

To convert OLD PET programs to function on the NEW PET, one needs a chart of the OLD memory location followed by the NEW memory location. Another chart would be needed to convert from the NEW locations into the OLD, so that OLD PET users would be able to use programs coming out for the NEW PETs. Realizing that Commodore will probably NOT care to help PET users by

0	OLDPET:NEW PET	-	POKE LOCATION CHANGE
1	-2	:1-2	USER FUNCTION ADDRESS LO, HI
3	.	:14	ACTIVE I/O CHANNEL (FOR PROMPT-SUPPRESS)(CURRENT INPUT DEVAJ
4	.	:"?	MULLS TO PRINT FOR CARRIAGE RETURN LINE FEED (UNUSED)
5	.	:"?	CURSOR COLUMN FOR NEXT INPUT OR PRINT
6	.	:15	TERMINAL WIDTH (UNUSED)
7	.	:16	LIMIT FOR SCANNING SOURCE COLUMNS (UNUSED)
8	-9	:17-18	LINE NUMBER BEFORE STORAGE (INTEGER ADDRESS FROM BASIC)
10	-89	:512-591	BASIC INPUT_BUFFER (80 BYTES) [NUMBER OF ARRAY SUBSCRIPTS]
90	.	:3	GENERAL COUNTER FOR BASIC (SEARCH CHAR- USUALLY 'I' OR ENDLIN
91	.	:4	00 USED AS DELIMITER (SCAN BETWEEN QUOTES FLAG)
92	.	:5	GENERAL COUNTER FOR BASIC [INPUT BUFFER PTR]E#OF SUBSCRIPTS]
93	.	:6	FLAG TO REMEMBER DIMENSIONED VARIABLES[1ST CHAR OF ARRAY NAME
94	.	:7	FLAG FOR VARIABLE TYPE 0= NUMERIC; 1= STRING [EFF=STRING]
95	.	:8	FLAG FOR INTEGER TAPE (80=INTEGER; 00=FLOATING POINT)
96	.	:9	FLAG TO CRUNCH RESERVED WORDS (DATA SCAN FLAG)\LIST QUOTE FLAG
97	.	:10	FLAG WHICH ALLOWS SUBSCRIPTS IN SYNTAX (FN X FLAG)
98	.	:11	FLAGS INPUT OR READ (0=INPUT; 64=GET; 152=READ)
99	.	:12	FLAG SIGN OF TAN (FLAG FOR TRIG SIGNS\COMPARN EVALUATN FLAG
100	.	:13	FLAG TO SUPPRESS OUTPUT (+=NORMAL; -=SUPPRESSED)
101	.	:19	INDEX TO NEXT AVAILABLE DESCRIPTOR (VARBL DESCRIPTR STACK PNTR
102	-103:20-21		POINTER TO LAST STRING TEMPORARY LO;HI (SECOND DESCRIPT POINTER
104	-111:22-29		TABLE OF DOUBLE BYTE DESCRIPTORS WHICH POINT TO VARIABLES
105	.....		[DESCRIPTOR STACK FOR TEMPORARY STRINGS]
112	-113:30-31		INDIRECT ADDRESS #1 LO;HI (POINTER FOR NUMBER TRANSFER)
114	-115:32-33		INDIRECT INDEX #2 LO;HI (NUMBER POINTERS)
116	-121:34-39		PSEUDO REGISTER FOR FUNCTION OPERANDS
117	-120:?"		[PRODUCT STAGING AREA FOR MULTIPLICATION]
122	-123:40-41		POINTER TO START OF BASIC TEXT AREA LO;HI
124	-125:42-43		POINTER TO START OF VARIABLES LO;HI (END BASIC\START VARBL
126	-127:44-45		POINTER TO ARRAY TABLE LO;HI (END VARIABLES\START ARRAYS]
128	-129:46-47		POINTER TO END OF VARIABLES LO;HI (START OF AVAILBL SPACE PTR
130	-131:48-49		START OF STRINGS POINTER LO;HI (BOTTOM OF STRINGS(MOVING DOWN

132 -133:50-51 TOP OF STRING SPACE POINTER LO;HI (MOVING DOWN)  
134 -135:52-53 HIGHEST RAM ADDRESS LO;HI (TOP OF BASIC MEMORY)  
136 -137:54-55 CURRENT LINE BEING EXECUTED (136=0 MEANS DIRECT) CURRENT LINE  
138 -139:56-57 LINE NUMBER FOR CONTINUE COMMAND LO;HI (LINE# SAVED BY END)  
140 -141:58-59 NEXT STATEMENT TO EXECUTE POINTER LO;HI (PREV LINE# FOR CONT)  
142 -143:60-61 DATA LINE# FOR ERRORS LO;HICLN# OF DATA LINE\CUR LNH READ PTR  
144 -145:62-63 DATA STATEMENT POINTER LO;HI (READ POINTER)  
145 ..... MEMORY ADDRESS OF DATA LINE  
146 -147:64-65 SOURCE OF INPUT LO;HI (INPUT VECTOR(DATA ETC)) DATA STMT PNTR  
148 -149:66-67 CURRENT VARIABLE NAME [CURRENT VARIABLE SYMBOLS]  
150 -151:68-69 POINTER TO VARIABLE IN MEMORY LO;HI [CURRENT VARBLE START ADR  
152 -153:70-71 POINTER TO VARIABLE REFERRED TO IN CURRENT FOR-NEXT  
153 ..... [152 END OPERAND FOR WAIT\153 AND OPERAND FOR WAIT]  
154 -155:72-73 POINTER TO CURRENT OPERATOR IN TABLE LO;HI  
155 ..... [154 Y SAVE REGISTER\ NEW OPERATOR SAVE]  
156 . :74 SPECIAL MASK FR CURRENT OPERATOR\COMPRES SML ACMLTR <1 =2 >4  
157 -158:75-76 FUNCTION DEFINITION POINTER LO;HI (NUMBER WORK AREA FOR SUB  
159 -160:77-78 POINTER TO A STRING DESCRIPTION LO;HI (MNR WRK AREA 157-161)  
161 . :79 LENGTH OF ABOVE STRING  
162 . :80 CONSTANT USED BY GARBAGE COLLECT ROUTINE\3 OR 7 FOR GRBG CLCT  
163 . :81 \$4C CONSTANT-6502 JMP INSTRUCTION \JUMP VECTOR FOR FUNCTIONSJ  
164 -165:82-83 VECTOR FOR FUNCTION DISPATCH LO;HI  
166 -171:84-89 FLOATING ACCUMULATOR #3 (NUMERIC STORE AREA)  
172 -173:90-91 BLOCK TRANSFER POINTER#1 LO;HI (NUMERIC STORE AREA)  
174 -175:92-93 BLOCK TRANSFER POINTER#2 LO;HI (NUMERIC STORE AREA)  
176 -181:94-99 FLOATING ACCUMULATOR\ (USER FUNCTION EVALUATED HERE)  
177 ..... [PRIMARY ACCUMULATR E,M,M,M,SJMSB PARAMETERS 181=FLPT SIGN  
182 . :100 DUPLICATE COPY OF HANTISSA OF FAC\TAYLOR SERIES CNST COUNTR  
183 . :101 COUNTER OF NUMBEK OF BITS TO SHIFT TO NORMALIZE FAC\ [SEE 185  
184 -189:102-107 FLOATING ACCUMULATOR#2 (SECONDARY ACUMULTR\DIADIC HLDNG AREA  
185 ..... [183 ACCUMULATOR HIGH ORDER PROPAGATION WORD]  
190 . :108 OVERFLOW BYTE FOR FLOATING ARGUMENT [SIGN COMPARSN PRIM/SCND  
191 . :109 DUPLICATE COPY HANTISSA SIGN\LOW ORDER ROUNDING BYTE-PRRY ACHL  
192 -193:110-111 POINTER TO ASCII REP OF FAC IN CONVERSION ROUTINE LO;HI  
193 ..... [CASSETTE BUFFER LENGTH\ TAYLOR CONSTANT POINTER]  
194 -199:112-117 CHARGET RAM CODE GETS NEXT CHARACTER FROM BASIC TEXT  
200 . :118 CHRGET RAM CODE GETS CURRENT CHARACTERS  
201 -202:119-120 POINTER TO SOURCE TEXT LO;HI  
203 -223:121-140 NEXT RANDOM NUMBER IN STORAGE  
224 -225:196-197 POINTER TO CURSOR LINE (SCREEN POSITION ON LINE)

publishing an official conversion chart, I decided to compile an unofficial conversion chart memory map combination. Please note: Extreme care was taken to compile these charts, but NO guarantee can be made as to their accuracy.

Rather than just list address changes I decided to also briefly summarize what each address is used for by the PET. This is referred to as a MEMORY MAP.

My sources for information included the following.

- 1) Commodore PET USER MANUAL
- 2) Barry Miles & Commodore Systems (in London, England) memory map
- 3) PET GAZETTE, Spring 79 issue, memory map by Jim Butterfield
- 4) BEST OF THE PET GAZETTE memory map compiled by Roy Busdicker
- 5) BEST OF THE PET GAZETTE memory map compiled by Commodore Systems in ENGLAND

226	:	:198	COLUMN POSITION OF CURSOR [POSITION OF CURSOR ON LINE]C0-79J
227	-	228:	199-200 GENERAL PURPOSE START ADDRESS INDIRECT LO;HI [UTILITY POINTER]
228	:	:	.....CTAPE BUFFER COUNTER, SCROLLINGINVERSE VIDEO CURSOR=1J
229	-	233:	201-204 GENERAL PURPOSE AND ADDRESS DIRECT LO;HI (ALSO 180 IN NEW)
230	:	:	.....LEND OF CURRENT PROGRAM\ TAPE END ADDRESSJ
231	-	232:	.....CTAPE TILING CONSTANTSJ
233	:	:	.....CTAPE BUFFER CHARACTERJ
234	:	:205	FLAG FOR QUOTE MODE ON/OFF [DIRECT/PROGRAMMED CURSOR\0=DIRECT
235	:	:"?	CTIMER 1 INTERRUPT STATUS \ 0=DISABLEDJ
236	:	:"?	LEOT CHARACTER RECEIVEDJ
237	:	:"?	ICCHARACTER ERROR RECEIVEDJ
238	:	:209	CURRENT FILE NAME LENGTH [NUMBER OF CHARACTERS IN FILE NAME]
239	:	:210	CURRENT LOGICAL FILE NUMBER [GPIB FILE #J
240	:	:211	CURRENT PRIMARY ADDRESS [FILE COMMAND(FROM OPEN)ICGPIB COMMAND
241	-	242:	212-213 CURRENT SECONDARY ADDRESS [241 DEVICE]IC242 MAX LINE LNH40/80
243	-	244:	214-215 START OF CURRENT TAPE BUFFER POINTER LO;HI
245	:	:216	CURRENT SCREEN LINE [LINE WHERE CURSOR LIVES]
246	:	:217	DATA TEMPORARY FOR I/O [LAST KEY HIT(ASCII)\BUFFER CHECKSUM]
247	-	248:	251-252 POINTER TO START LOC FOR O.S. LO;HI [TAPE START ADDR\TAPE PNT
248	:	:	.....[POINTER TO PROGRAM DURING VERIFY, LOAD, SAVE]
249	-	250:	218-219 POINTER TO CURRENT FILE NAME LO;HI [FILE NAME POINTER]
251	-	254:	:"? UNUSED [251 REMAINING COUNT FOR INSERT MODE]
252	:	:	.....[252 SERIAL WORD]
253	:	:	.....[253 NUMBER OF BLOCKS REMAINING TO WRITE]
254	:	:	.....[254 SERIAL WORD BUFFER]
255	:	:"?	OVERFLOW BYTE THAT BASIC USES WHEN DOING FAC TO ASCII CONVRSN
256	-	511:	:"? 62 BYTE ON BOTTOM ARE USED FOR ERROR CORRECTION IN TAPE READS
257	:	:	.....ALSO BUFFER FOR ASCII WHEN BASIC IS EXPANDING THE FAC INTO A
258	:	:	.....PRINTABLE NUMBER. THE REST OF PAGE 1 IS USED FOR STORAGE OF
259	:	:	.....BASIC GOSUB AND FOR NEXT CONTEXT AND HARDWARE STACK FOR THE
260	:	:	.....MACHINE.
261	:	:	.....[256-266 IS BINARY TO ASCII CONVERSION AREA]
267	:	:	.....[267-311 IS STACK AREA]
512	-	514:	141-143 24 HOUR CLOCK IN 1/40 SEC. [CLOCK THAT INCREMENTS 60 PER SEC
515	:	:151	KEYSTROKE VALUE [WHICH KEY DEPRSD\255=NO KEY]ICMATRIX ROW-COL
516	:	:152	SHIFT FLAG (0=NO SHIFT 1=SHIFT) [SHIFT KEY 1 IF DEPRESSED]
517	-	518:	153-154 CORRECTION FACTOR FOR CLOCK LSB; ASB [CLOCK INCREMENTS 30/SEC
519	-	520:	249-250 INTERRUPT DRIVER FLAG FOR CASSETTE\1 SWITCHES; #2 SWITCHES
520	:	:	.....[519 FOR CASSETTE #1 ON]IC520 FOR CASSETTE #2 ONJ

6) SPINX PET  
 Newsletter Vol 0 NO 4  
 memory map

If there was an official  
 Connode location  
 discription it is listed  
 first. Discriptions from  
 other sources are  
 printed inside square  
 brackets. To fit the  
 discriptions into one  
 line many words were  
 abbreviated by dropping  
 vowels and other  
 letters. If two lines  
 were needed to fully  
 list the discriptions  
 the second of the two  
 begins with a row of  
 dots .....

USES OF THE CHARTS

- \* If you have an OLD PET program that doesn't function correctly on your new PET you may be able to modify it using the OLDNEW chart. LOAD the program into your NEW PET. Carefully scan the LISTING of the program. Check every PEEK and PONE used.
- See my PEEK/PONE find and list program listing following these charts.
- Change the location

being PEEKed or POKEd to the location listed on the chart. (Find the OLD location on the charts first column. The NEW location is listed in the second column to its right.) In most cases this will allow the OLD program to function on the NEW PET. SAVE your corrected version on tape before you try a RUN.	521 . :155	DUPLICATE OF 59410, BOTTOM ROW KEYS [KEYSWITCH PIA FLAGS]
	522 . :??	[TIMING CONSTANT BUFFER]
	523 . :157	FLAG <> MEANS VERIFY NOT LOAD INTO MEMORY [LOAD=0\ VERIFY=1]
	524 . :150	I/O STATUS BYTE [STATUS (ST)]
	525 . :158	INDEX INTO KEYSTROKE BUFFER [N OF CHARACTERS IN KEYSTRK_BUFFER]
	526 . :159	FLAG TO INDICATE REVERSE FIELD ON INTERRUPT DRIVEN KEYSTROKE_BUFFER
	527 -536:623-632	I/O RAM VECTOR LO;HI [HARDWARE INTERRUPT VECTOR]
	537 -538:144-145	BRK INSTRUCTION RAM VECTOR LO;HI [BREAK INTERRUPT VECTOR]
	539 -540:146-147	[IEE MODE]
	541 . :??	END OF LINE FOR INPUT POINTR[IN# OF CHARACTERS ON SCREEN LINE]
	542 . :??	?
	543 . :??	[CURSOR LOG (ROW,COLUMN)] [USED IN INPUT ROUTINE]
	544 -545:??	[PBD IMAGE FOR TAPE I/O]
	546 . :??	[KEY IMAGE]
	547 . :??	
* If you have a NEW PET program that doesn't function correctly on your OLD PET use the NEW:OLD chart. Follow the procedure listed above.	548 . :??	LO=FLASHING CURSOR: ELSE NO CURSOR SHOWS [CURSOR_ENABLE\POKE=0]
	549 . :168	COUNTDOWN TO FLIP CURSOR [CURSOR_TIMING_COUNTDOWN][POKE=1]
	550 . :169	SCREEN VALUE OF CHARACTER UNDER CURSOR
	551 . :170	FLAG FOR CURSOR ON/OFF [CURSOR_BLINK_FLAG\1=BLNK STARTED\POKE=0]
	552 . :??	LEOT BIT RECEIVED [TAPE WRITE]
	553 -577:224-248	TABLE OF LSB OF START ADDRESSES OF VIDEO DISPLAY LINES(25)
	554 . :??	[SCREEN LINE WRAP TABLE]
	578 -587:593-602	TABLE OF LOGIC ADDRESSES [LOGICAL NUMBERS OF OPEN FILES]
* If you are writing a PET program you can make a version for each PET by using the charts. Or you can make a program that works on BOTH PETs. This takes a bit more effort and more memory. Your program can find out which type of PET it is running in. Simply start your program with this line: 1 PT=PEEK(50003)	588 -597:603-612	TABLE OF PRIMARY ADDRESSES [DEVICE #S OF OPEN FILES]
	598 -609:613-622	TABLE OF SECONDARY ADDRESSES [COMMAND/SECONDARY ADRES OPEN FILES]
	608 . :??	[608 INPUT FROM: 0=KEYBOARD; 1=SCREEN]
	610 . :174	INDEX INTO LA, FA, SA TABLES [NUMBER OF OPEN FILES]
	611 . :175	DEFAULT INPUT DEVICE # [NORMALLY=0 FOR KEYBOARD]
	612 . :176	DEFAULT OUTPUT DEVICE#[OUTPUT TO CMD DEVICE, NORMALLY=3 SCREEN]
	613 . :177	COMPUTATION OF PARITY ON CASSETTE [KLIECTAPE CHARACTER PARITY]
	614 . :??	[BYTE RECEIVED FLAG]
	615 . :??	?
	616 . :181	TAPE BUFFER ITEM COUNTER [POINTER IN FILE NAME TRANSFER]
	617 . :??	?
	618 . :??	UNUSED
	619 . :??	?
	620 . :??	SERIAL BIT COUNT
	621 . :184	[COUNT OF REDUNDANT TAPE BLOCKS]
	622 . :??	?
An OLD PET will set PT=0. A NEW PET will set	623 . :??	[CYCLE COUNTER][FLIP FOR EVERY BIT COMING OFF OF TAPE]

```

624 . :186 COUNTDOWN SYNCHRONIZATION ON CASSETTE WRITE
625 -626:187-188 INDEX NEXT CHARACTER IN/OUT TAPE BUFFER#1;#2 L625 FORM#626-#2
627 . :189 COUNTDOWN SYNCHRONIZATION ON CASSETTE READLEADR CNTR/PASS1/2
628 . :190 FLAG TO INDICATE BIT/BYTE TAPE ERROR [WRITE NEW BYTE]
629 . :191 FLAG TO INDICATE TAPE ROUTINE READING SHORTS [WRITE START BIT

630 -631:192-193 INDEX TO ADDRESSES TO CORRECT ON TAPE READ PASS 1; PASS 2
631 .....L631 TAPE DROUPT CNTRJL630 PASS1 ERR PNTR\631 PASS2 ERR PNTR
632 . :194 FLAG FOR CASSETTE READ-TELLS CURRENT FUNCTN-COUNTDOWN,READL=635
633 . :195 COUNT OF SECONDS OF SHORTS TO WRITE BEFORE DATA [CHECKSUM]
634 -825:634-825 BUFFER FOR CASSETTE #1 (192 BYTES)
635 (CONTINUED 632) [632 READ MODE: 0=SCAN 1-15=COUNTDOWN #40=LOAD #80=END]
636 .....L632 TAPE CORRECTION COUNT]
826 -1017:826-1017 BUFFER FOR CASSETTE #2 (192 BYTES)

```

## CHART #2 NEW to OLD

```

0 NEWPET/OLDPET- POKE LOCATION CHANGE
1 -2 :1-2 USER FUNCTION ADDRESS LO;HI
3 . :90 GENERAL COUNTER FOR BASIC [SEARCH CHRTR-USUALLY ',' OR ENDLIN
4 . :91 ;00 USED AS DELIMITER [ESCAP BETWEEN QUOTES FLAG]
5 . :92 GENERAL COUNTER FOR BASIC [INPUT BUFFER PNTR][#OF SUBSCRIPTS]
6 . :93 FLAG TO REMEMBER DIMENSIONED VARIABLES [1ST CHAR OF ARRAY NAM
7 . :94 FLAG FOR VARIABLE TYPE 0= NUMERIC; 1= STRING [$$$=STRING]
8 . :95 FLAG FOR INTEGER TAPE L80=INTEGER; 00=FLOATING POINT]
9 . :96 FLAG TO CRUNCH RESERVED WORDS [DATA SCAN FLAG\LIST QUOTE FLAG
10 . :97 FLAG WHICH ALLOWS SUBSCRIPTS IN SYNTAX [FN X FLAG]
11 . :98 FLAGS INPUT OR READ [0=INPUT; 64=BET; 152=READ]
12 . :99 FLAG SIGN OF TAN [FLAG FOR TRIG SIGNS\COMPARSN EVALUATN FLAG
13 . :100 FLAG TO SUPPRESS OUTPUT [*=NORMAL; -=SUPPRESSED]
14 . :3 ACTIVE I/O CHANNEL [FOR PROMPT-SUPPRESS]
15 . :6 TERMINAL WIDTH [UNUSED]
16 . :7 LIMIT FOR SCANNING SOURCE COLUMNS [UNUSED]
17 -18 :8-9 LINE NUMBER BEFORE STORAGE [INTEGER ADDRESS FROM BASIC]
19 . :101 INDEX TO NEXT AVAILABLE DESCRIPTOR[VARIABLE DESCRIPTR STACK PNTR
20 -21 :102-103 POINTER TO LAST STRING TEMPORARY LO;HI [SECOND DESCRIPTR POINTER
22 -29 :104-111 TABLE OF DOUBLE BYTE DESCRIPTORS WHICH POINT TO VARIABLES
23 .....[DESCRIPTOR STACK FOR TEMPORARY STRINGS]
30 -31 :112-113 INDIRECT ADDRESS #1 LO;HI [POINTER FOR NUMBER TRANSFER]
32 -33 :114-115 INDIRECT INDEX #2 LO;HI [NUMBER POINTER]
34 -39 :116-121 PSEUDO REGISTER FOR FUNCTION OPERANDS

```

PT=1.

more

Now, for example, if you wish to clear the keyboard buffer you could code this:

```

250 IF PT THEN POKE
158,0 : GOTO 270
260 POKE 525,0
270 program continues

```

This utilizes the variable PT we set in the first line of our program. There are two lines, one for each PET along with a conditional IF THEN which allows your PET to execute only the correct POKE. This works fine, but I prefer the following method which I have used to convert many of my programs already. It uses only one POKE or PEEK, but has a computed location, utilizing the variable PT as set in the first line of our program. I hope this method can become a "standard" method to allow programs to be used on either PET.

This is a bit more complex:

In this example, an OLD PET would POKE 525-0\*367 or POKE 525. A NEW PET would POKE 525-1\*367 or POKE 158.

To use this computed location method simply use the POKE location for the OLD PET. Then if the NEW PET location is higher, add PT times the difference. If the NEW PET location is lower, subtract PT times the difference. Could someone compile a chart with the OLD PET location followed by the difference to the NEW PET location.

A SMART PROGRAM

Using the method just described a program can decide for itself which version of PET it is running on. Just make sure you start your program with:

```
1 PT=PEEK(50003)
```

Why does this work? On the OLD PET PT=0. Anything times 0 equals 0 so the POKE location is not affected by the amount added or subtracted since it is multiplied by PT first. Since PT=1 on the NEW PET the addition or

40	-41	:122-123	POINTER TO START OF BASIC TEXT AREA LO;HI
42	-43	:124-125	POINTER TO START OF VARIABLES LO;HI LEAD OF BASIC\START VAR#L
44	-45	:126-127	POINTER TO ARRAY TABLE LO;HI LEAD VARIABLES\START ARRAYSJ
46	-47	:128-129	POINTER TO END OF VARIABLES LO;HI (START OF AVAILBL SPACE PTR
48	-49	:130-131	START OF STRINGS POINTER LO;HI (BOTTOM OF STRINGS\MOVING DOWN
50	-51	:132-133	TOP OF STRING SPACE POINTER LO;HI (MOVING DOWN)
52	-53	:134-135	HIGHEST RAM ADDRESS LO;HI (TOP OF BASIC MEMORY)
54	-55	:136-137	CURRENT LINE BEING EXECUTED (54=2 MEANS DIRECT) (CURRENT LNH
56	-57	:138-139	LINE NUMBER FOR CONTINUE COMMAND LO;HI (LINE# SAVED BY ENDJ
58	-59	:140-141	NEXT STATEMENT TO EXECUTE POINTER LO;HI (PREV LINE# FOR CONTJ
60	-61	:142-143	DATA LINE# FOR ERRORS LO;HI (LINE# OF DATA LINE)
62	-63	:144-145	DATA STATEMENT POINTER LO;HI (READ POINTER)
63	.....		MEMORY ADDRESS OF DATA LINE
64	-65	:146-147	SOURCE OF INPUT LO;HI (INPUT VECTOR (DATA ETC))\DATA STMT PTR
66	-67	:148-149	CURRENT VARIABLE NAME (CURRENT VARIABLE SYMBOLS)
68	-69	:150-151	POINTER TO VARIABLE IN MEMORY LO;HI (CURRENT VARIABLE START ADR
70	-71	:152-153	POINTER TO VARIABLE REFERRED TO IN CURRENT FOR-NEXT
72	-73	:154-155	POINTER TO CURRENT OPERATOR IN TABLE LO;HI
73	.....		(154 Y SAVE REGISTER)\NEW OPERATOR SAVEJ
74	.....		SPECIAL MASK FOR CURRENT OPERATOR (CMPR SYMBL ACHLTR <1 => >4
75	-76	:157-158	FUNCTION DEFINITION POINTER LO;HI (NUMBER WORK AREA FOR SOR...)
77	-78	:159-160	POINTER TO A STRING DESCRIPTION LO;HI (NUM#R WRK AREA 157-161J
79	.....		LENGTH OF ABOVE STRING
80	.....		CONSTANT USED BY GARBAGE COLLECT ROUTINE (3 OR 7 FOR GR8G CLT
81	.....		44C CONSTANT-6502 JMP INSTRUCTION (JUMP VECTOR FOR FUNCTIONS)
82	-83	:164-165	VECTOR FOR FUNCTION DISPATCH LO;HI
84	-89	:166-171	FLOATING ACCUMULATOR #3 (NUMERIC STORE AREA)
90	-91	:172-173	BLOCK TRANSFER POINTER#1 LO;HI (NUMERIC STORE AREA)
92	-93	:174-175	BLOCK TRANSFER POINTER#2 LO;HI (NUMERIC STORE AREA)
94	-99	:176-181	FLOATING ACCUMULATOR#1 (USER FUNCTION EVALUATED HERE)
95	.....		(PRIMARY ACCUMULATR E,M,M,M,SJ)\MSB PARAMETERS 181=FLPT SIGN
100	.....		DUPLICATE COPY OF WANTISSA OF FAC#1 (TAYLOR SERIES CONST CNTR
101	.....		COUNTER - NUMBER OF BITS TO SHIFT TO NORMALIZE FAC#1 (SEE 103
102	-107	:184-189	FLOATING ACCUMULATOR#2 (SECONDARY ACCUMULTR)\DIADIC HLDNG AREA
103	.....		(101 IS ACCUMULATOR HIGH ORDER PROPAGATION WORD)
108	.....		OVERFLOW BYTE FOR FLOATING ARGUMENT (SIGN COMPARISSN PRIM\SCND
109	.....		DUPLICATE COPY WANTISSA SIGN (LOW ORDER ROUNDING BYTE-PRR ACHL
110	-111	:192-193	POINTER TO ASCII REP OF FAC IN CONVERSION ROUTINE LO;HI
111	.....		(CASSETTE BUFFER LENGTH)\TAYLOR CONSTANT POINTERJ
112	-117	:194-199	CHRGRT RAM CODE GETS NEXT CHARACTER FROM BASIC TEXT
118	.....		CHRGRT RAM CODE GETS NEXT CHARACTER
119	-120	:201-202	POINTER TO SOURCE TEXT LO;HI
121	-140	:203-223	NEXT RANDOM NUMBER IN STORAGE \ OR
136	-140	.....	NEXT RANDOM NUMBER IN STORAGE
141	-143	:512-514	24 HOUR CLOCK IN 1/60 SEC (CLOCK THAT INCREMENTS 60 PER SEC

subtraction is performed  
yielding the NEW PET  
POKE location as  
planned. This sounds  
complicated, but is  
really quite simple.  
Here are some examples  
of common POKES that  
will work on BOTH PETS:

    CLEAR KEYBOARD BUFFER  
    & GET CHARACTER

200 POKES25-PT\*367,0:  
WAIT525-PT\*367,1: GET#

    FILL KEYBOARD BUFFER  
    WITH RETURNS

210 FOR N=0 TO 9:  
POKES27+N+PT\*96,13: NEXT

    DISABLE STOP KEY

2POKES37-PT\*393,PEEK(537-  
PT\*393)+3

\* If you would like to  
understand what someone  
elses program is doing  
with its PEEKs and POKES  
look up the location on  
the applicable chart.  
The description on the  
right may help your  
understanding of that  
POKE or PEEK function.

    FINAL NOTES

144 -145:537-538 IRQ RAM VECTOR LO;HI [HARDWARE INTERRUPT VECTOR]  
146 -147:539-540 BRK INSTRUCTION RAM VECTOR LO;HI [BREAK INTERRUPT VECTOR]  
148 . :? NMI RAM VECTOR  
150 . :524 I/O OPERATION STATUS BYTE [STATUS (ST)]  
151 . :515 LAST KEY INDEX [WHICH KEY DEPRSD;255=NO KEY;1=MATRIX ROW-COL  
SHIFT FLAG (0=NO SHIFT\ 1=SHIFT) [SHIFT KEY 1 IF DEPRESSED]  
152 . :516 CORRECTION FACTOR FOR CLOCK LSB; MSB [CLOCK INCREMENTS 30/SEC  
153 -154:517-518 DUPLICATE OF 59410.(NEW=?) [BOTTOM ROW KEYS [KEYSWITCH PIA FLAGS  
155 . :521 " FLAG <> MEANS VERIFY NOT LOAD INTO MEMORY [LOAD=0=VERIFY=1]  
157 . :523 INDEX INTO KEYSTROKE BUFFER [# OF CHARACTERS IN MYSTRK BUFFER  
158 . :525 FLAG TO INDICATE REVERSE FIELD ON  
159 . :526 UNUSED  
160 -166:?? CURSOR ON FLAG  
167 . :? COUNT OF JIFFIES TO BLINK CURSOR[CURSOR TIMING CNTDN][POKE=1]  
168 . :549 SCREEN VALUE OF CHARACTER UNDER CURSOR  
169 . :550 CHAR SAVED DURING BLNK\CURSOR ON/OFF FLOI[CURSOR BLNK FLAG][POKE=0  
170 . :551 UNUSED  
171 -173:?? POINTER INTO LOGICAL FILE TABLE [NUMBER OF OPEN FILES]  
174 . :610 DEFAULT INPUT DEVICE # [NORMALLY=0]  
175 . :611 DEFAULT OUTPUT DEVICE # [OUTPUT TO CMD DEVICE, NORMALLY=3]  
176 . :612 TAPE VERTICAL PARITY[COMPUTATION OF PARITY ON CASSETTE WRITE  
177 . :613 UNUSED [?]  
178 -185:?? GENERAL PURPOSE AND ADDRESS DIRECT LO;HI (ALSO LOCATN 201-204  
180 . :229-233 TAPE BUFFER ITEM COUNTER [POINTER IN FILE NAME TRANSFER]  
181 . :616 [COUNT OF REDUNDANT TAPE BLOCKS]  
184 . :621 SYNC ON TAPE HEADER COUNT\COUNTDOWN SYNC ON CASSETTE WRITE  
186 . :624 POINTER TO ACTIVE CASSETTE \ OR  
187 -188:625-626 INDEX NEXT CHARACTER IN/OUT TAPE BUFFER#1;#2[187 FORM1\188=#2  
188 ..... COUNTDOWN SYNCHRONIZATION - CASSETTE READER CTRN\PASS1/2  
189 . :627 FLAG TO INDICATE BIT/BYTE TAPE ERROR [WRITE NEW BYTE]  
190 . :628 FLAG TO INDICATE TAPE ROUTINE READING SHORTS [WRITE START BIT  
191 . :629 INDEX TO ADDRESSES TO CORRECT ON TAPE READ PASS 1: PASS 2  
192 -193:630-631 ..... [192 FOR PASS 1 ERROR LOG POINTR\193 FOR PASS 2 ERR LOG PTR  
193 ..... FLAG-CASSETTE REAR-TELLS CURRENT FUNCTN-COUNTDOWN READ  
194 . :632 COUNT OF SECONDS OF SHORTS TO WRITE BEFORE DATA [CHECKSUM]  
195 . :633 POINTER TO CURSOR POSITION [SCREEN POSITION ON LINE]  
196 -197:224-225 COLUMN POSITION OF CURSOR [POSITION OF CURSOR ON LINE]  
198 . :226 LOAD START ADDRESS LO;HI [UTILITY POINTER]  
199 -200:227-228 ..... [TAPE BUFFER,SCROLLING[INVERSE VIDEO CURSOR=1]  
200 ..... PRINT LOAD END ADDRESS LO;HI (INCLUDES LOCATION 180)  
201 -202:229-233

These charts were compiled within 2 weeks of the arrival of the NEW PETs. If you find any flaws or areas to be clarified please let me know. A corrected, more complete chart hopefully will be printed in next issue.

**COMMODORE NEWS**

The 32K NEW PET went up in price to \$1295. The PET 2021 electrostatic printer was cancelled & will not be produced. The PET 2040 dual floppy disk unit went up \$200 to \$1295. The PET 2041 single floppy disk unit has been cancelled due to lack of interest. It is replaced by the 2040A single floppy for \$895, which is \$300 more than the model cancelled. It is the dual floppy with only 1 drive installed. Also note: two different RELIABLE sources have told me they consider the PET Printer to be "junk" or "not commercial quality". That of course is strictly opinion. I have not seen them yet and thus can not say one way or the other.

202 .....[END OF CURRENT PROGRAM\ TAPE END ADDRESS]  
 205 . :234 FLAG FOR QUOTE MODE ON/OFF [DIRECT/PROGRAMMED CURSOR\*0=DIRECT  
 206 -208: "?" UNUSED  
 209 . :238 CURRENT FILE NAME LENGTH [NUMBER OF CHARACTERS IN FILE NAME]  
 210 . :239 CURRENT FILE LOGICAL ADDRESS [GPIB FILE #]  
 211 . :240 CURRENT PRIMARY ADDRESS [FILE COMMAND(FROM OPEN)] [GPIB COMND]  
 212 -213:241-242 CURRENT SECONDARY ADDRESS [212 DEVCEN][213 MAX LINE LNTH40/80  
 214 -215:243-244 START OF CURRENT TAPE BUFFER POINTER LO:HI  
 216 . :245 CURRENT SCREEN LINE# [LINE WHERE CURSOR LIVES]  
 217 . :246 DATA TEMPORARY FOR I/O [LAST KEY HIT(ASCII)\*BUFFER CHECKSUM]  
 218 -219:249-250 POINTER TO CURRENT FILE NAME LO:HI [FILE NAME POINTER]  
 220 -221: "?" UNUSED  
 222 . : "?" CASSETTE READ BLOCK COUNT  
 223 . : "?" UNUSED  
 224 -248:553-577 TABLE OF LSB OF START ADDRESSES OF VIDEO DISPLAY LINES(25)  
 225 .....[SCREEN LINE WRAP TABLE]  
 249 -250:519-520 INTERRUPT DRIVER FLAG FOR CASSETTE#1 SWITCHES; #2 SWITCHES  
 250 .....[249 FOR CASSETTE #1 ON][250 FOR CASSETTE #2 ON]  
 251 -252:247-248 POINTER TO START LOC FOR O.S. LO:HI [TAPE START ADDR\*TAPE PNT  
 252 .....[POINTER TO PROGRAM DURING VERIFY, LOAD]  
 512 -591:10-89 BASIC INPUT BUFFER (80 BYTES) [NUMBER OF ARRAY SUBSCRIPTS]  
 513 .....512-513 IS THE PROGRAM COUNTER  
 514 .....514 IS PROCESSOR STATUS  
 515 .....515 IS ACCUMULATOR  
 516 .....516 IS X INDEX  
 517 .....517 IS Y INDEX  
 518 .....518 IS STACK POINTER  
 519 -520 .....519-520 IS USER MODIFIABLE IRQ  
 593 -602:578-587 LOGICAL FILE NUMBERS [LOGICAL NUMBERS OF OPEN FILES]  
 603 -612:588-597 PRIMARY DEVICE NUMBERS [DEVICE #S OF OPEN FILES]  
 613 -622:598-609 TABLE OF SECONDARY ADDRESSES [COMMAND/SECONDARY ADRS OPEN FIL  
 623 -632:527-536 INTERRUPT DRIVEN KEYSTROKE BUFFER  
 634 -825 BUFFER FOR CASSETTE #1 (192 BYTES)  
 826 -1017:826-1017 BUFFER FOR CASSETTE #2 (192 BYTES)

## NOTIZEN

# Wichtige Routinen in PET-BASIC

A few routines from PET basic

C2AC-C2D9 peeks at the stack for an active FOR loop  
C2AD-C31C 'opens up' a space in Basic for insertion of a new line.  
C31D-C329 tests for stack-too-deep and aborts if found.  
C32A-C356 check available memory space  
C357-C388 sends a canned error message from C190 area, then drops into:  
C389-C391 Signals 'ready'  
C394-C3A9 gets a line of input, analyzes it, executes it  
C3AC-C42E handles a new line ~~fr~~ of Basic from keyboard; deletes old line, etc.  
C430-C460 corrects the chaining between Basic lines after insert/delete  
C462-C476 receives a line from the keyboard into the Basic buffer  
C479-C48C gets each character from keyboard  
C48D-C521 looks up the keywords in an input line and changes to "tokens"  
C522-C550 searches for the location of a Basic line from number in 8,9  
C551-C599 implements NEW command - clears everything  
C59A-C5A7 sets the Basic pointer to start-of-program  
C5A8-C647 performs LIST command  
C649-EC68F executes a FOR statement  
C692-C6B4 continues to build ~~the~~ FOR vectors  
C6B5-C6EF reads and executes the next Basic statement, finds next line, etc.  
C6F2-C70A executes the Basic Command as a subroutine  
C70D-C71B performs RESTORE  
C71C-C742 handles STOP, END, and BREAK procedures.  
C745-C75E performs CONT  
C75F-C76D set pause after carriage return (never called)  
C770-C772 performs CLR  
C775-C77D performs RUN  
C780-C79A performs GOSUB  
C79D-C7C9 performs GOTO  
C7CA-C7FD performs RETURN  
C7FE-C81E scans for start of next Basic Line  
C820-C840 performs IF  
C843-C862 performs ON  
C863-C89A gets a fixed-point number from Basic and stores in 8,9  
C89D-C91B performs LET  
C91C-C97E check numeric digit/move string pointer  
C97F-C982 performs PRINT#  
C985-C996 performs CMD  
C999-CA24 performs PRINT  
CA27-CA41 prints ~~the~~ string from address in Y,A  
CA44-CA76 prints a character  
CA77-CA9E handles bad input data  
CA9F-CAC5 performs GET  
CAC6-CADF performs INPUT#  
CAEO-CB14 performs INPUT  
CB17-CB21 prompts and receives the input  
CB24-CC11 performs READ

CC12-CC35 canned messages: EXTRA IGNORED; REDO FROM START  
 CC36-CC8F performs NEXT  
 CC92-CCB5 checks Basic format, data type, flags TYPE MISMATCH  
 CCB8-CD38 inputs and evaluates any expression (numeric or string)  
 CD3a-CD9C pushes a partially-evaluated argument to the stack  
 CD9D-CDB9 evaluates a numeric, variable, or pi, or identifies other symbol  
 CDEC-CDCC value of pi in floating binary  
 CD1-CDE7 checks for special characters (+,-,",.) at start of expression  
 CDE8-CDF6 performs NOT function  
 CDF7-CE04 checks for various functions  
 CE05 evaluates expression within parentheses ( )  
 CE0B checks for right parenthesis )  
 CE0E checks for left parenthesis (  
 CE11-CE1B checks for comma  
 CE1C-CE20 prints SYNTAX ERROR and exits  
 CE21-CE27 sets up function for future evaluation  
 CE28-CE39 set up a variable name search  
 CE3E-CE96 checks for special variables TI, TI\$, and ST  
 CE97-CEE5 identifies and sets up function references  
 CED6-CF05 perform the OR and AND functions  
 CFC6-CF6D performs comparisons  
 CF6E-CF7A sets up DIM execution  
 CF7B-DO0E searches for a Basic variable  
 DO0F-DO78 creates a new Basic variable  
 DO79-DO87 logs Basic variable location  
 DO88-DO98 is array pointer subroutine  
 DO99-DO9C is 32768 in floating binary  
 DO9D-DOB8 is floating point-to-fixed conversion for signed values  
 DOB9-D263 locates and/or creates arrays  
 D264-D277 performs FRE function  
 D278-D284 converts fixed point-to-floating  
 D285-D28A performs POS function  
 D28B-D294 checks direct/indirect command, gives 'ILLEGAL DIRECT'  
 D295-D348 executes DEF statements and evaluation FNK  
 D349-D36A performs STR\$ function  
 D36B-D3D1 scans and sets up string elements  
 D3D2-D403 builds string vectors  
 D404-D5C3 does 'garbage collection' - discards unwanted strings  
 D5C4-D5D7 performs CHR\$ function  
 D5D8-D653 performs LEFT\$, RIGHT\$, MID\$ functions  
 D654-D662 performs LEN, gets string length  
 D663-D672 performs ASC function  
 D673-D684 gets a single-byte value from Basic  
 D685-D6C3 evaluates VAL function  
 D6C4-D6CF gets two arguments (16-bit and 8-bit) from Basic  
 E6D0-E6E5 checks argument is in range 0-65535  
 E6E6-D701 performs PEEK and POKE  
 D702-D71D executes WAIT statement  
 D71E-D890 performs addition and subtraction  
 D891-D8EE contains floating-point constants  
 98BF-D8FC performs LOG function  
 D8FD-D95D performs multiplication  
 D95E-D988 loads secondary accumulator from memory (\$B8 to \$BD)  
 D989-D9B3 test and adjust primary/secondary accumulators  
 D9B4-D9E0 routines to multiply or divide by 10  
 D9E1-DA73 performs division  
 DA74-DA98 loads primary accumulator from memory (\$B0-\$B5)  
 EA99-DACD transfers primary accumulator to memory  
 DACE-DADD transfers secondary accumulator to primary  
 DAD3-DAEC transfers primary accumulator to secondary

DAED-DAFC rounds the primary accumulator  
 DAFD-DB29 extracts primary sign; performs SGN function  
 DB2A-DB2C performs ABS  
 DB2D-DB6C compares primary accumulator to memory  
 DB6D-DB9D Convert Floating point to fixed, unsigned  
 DB9E-DBC4 perform INT function  
 DBC5-DC4F convert ASCII string to floating point  
 DC50-DC84 get new ASCII digit  
 DC94-DCAE print Basic Line number  
 DCAF-DDE2 convert floating point to ASCII string (at 0100 up)  
 DDE3-DE23 conversion constants - decimal or clock  
 DE24-DE2D evaluation SQR function  
 DE2E-DE66 evaluation of power function  
 DE67-DE71 negate (monadic -)  
 DEAO-DEF2 perform EXP function  
 DEF3-DF3C perform function series evaluation  
 DF45-DF9D perform RND calculation  
 DF9E evaluate COS function  
 DFA5-DFED evaluate SIN function  
 DFEE-E019 evaluate TAN function  
 E048-E077 evaluate ATN function  
 E0B5-E0CC Basic scan program, transferred to OOC2-00D9  
 E0D2-E173 completion of power-on-reset; memory test, etc.  
 E19B-E1BB partial test for TI and TI\$  
 E1BC-E1E0 input/read/get director  
 E1E1-E27C initialize I/O registers. clear screen, reset subroutine  
 E27D-E3C3 receive input from keyboard/screen  
 E3C4-E3E9 set up new screen line  
 E3EA-E52F output character to screen  
 E530-E5DA check for and perform screen scrolling  
 E5DB-E66A start new screen line  
 E66B-E67D interrupt entry  
 E67E-E683 interrupt return  
 E685-E73E hardware interrupt routine: cursor flash, tape motor.  
 E73F-E7A2 convert keyboard matrix to ASCII keyboard  
 E7AC-E7B9 write-on-screen subroutine  
 E7DE-E7EE print canned monitor message  
 F0B6-F1CB IEEE-488 channel open, test, close  
 F1CC-F22F get input character from keyboard, screen cassette. IEEE  
 F230-F27C output character to screen, cassette. IEEE  
 F27D-F2A3 restore normal I/O, clear IEEE channels  
 F2A4-F2AA abort (not close!) all files  
 F2AB-F2B7 locate logical file table entry  
 F2B8-F2C7 transfer file table entries to Device, Command  
 F2C8-F329 perform file CLOSE  
 F32A-F33E test stop key  
 F33F-F345 test if direct/indirect command for suppressing file advice  
 F346-F3FE perform file LOAD  
 F3FF-F421 print "SEARCHING .. "  
 F422-F432 print "LOADING .. " or "VERIFYING"  
 F433-F461 get parameters for LOAD and SAVE  
 F462-F494 perform IEEE sequences for LOAD, SAVE, and OPEN  
 F495-F4BA search for specific tape header  
 F4BB-F4D3 perform VERIFY  
 F4D4-F529 get parameters for OPEN and CLOSE  
 F52A-F5AD perform OPEN  
 F5AE-F5E2 search for any tape header  
 F5E3-F5EC clear tape buffer  
 F5ED-F64C write tape header  
 F64D-F666 get start & end addresses from tape header

F667-F67C Set buffer start address  
 F67D-F694 set tape buffer start and end pointers  
 F695-F69D perform SYS command  
 F69E-F71E perform SAVE  
 F71C-F735 find unused secondary address  
 F736-F78A update clock  
 F78B-F7DB set input device  
 F7DC-F82C set output device  
 F82D-F83A bump tape buffer counter  
 F83B-F85D wait for cassette PLAY switch  
 F85E-F870 test cassette switch line  
 F871-F87E wait for cassette RECORD and PLAY switches  
 F87F-F8B8 read tape initiation routine  
 F8B9-F8D1 write tape initiation routine  
 F8D2-F912 complete tape read or write  
 F913-F91D wait for I/O completion  
 F91E-F92D test stop key and abort if necessary  
 F92E-F95E subroutine to set tape read timing  
 F95F-FBDB interrupt routine for tape read  
 FBDC-FBE4 save memory pointer  
 FBE5-FBEB set ST error flag  
 FBEC-FBFF subroutine to count 8 serial bits per byte  
 FC00-FC1B subroutine to write a bit to tape  
 FC1C-FCFA interrupt 1 for tape write - entry at FC21  
 FCFB-FD15 terminate I/O and restore normal vectors  
 FD16-FD37 subroutine to set interrupt vector  
 FD38-FD47 power-on reset entry; test for diagnostic  
 FD48-FD7B diagnostic routine  
 FD7C-FD8F checksum routine  
 FD90-FD9A pointer advance subroutine  
 FD9B-FFB1 diagnostic routines  
           JUMP TABLE:  
 FFC0          OPEN  
 FFC3          CLOSE  
 FFC6          set input device  
 FFC9          set output device  
 FFCC          restore normal I/O devices  
 FFCE          input character (from screen)  
 FFD2          output character  
 FFD5          LOAD  
 FFD8          SAVE  
 FFDE          VERIFY  
 FFDE          SYS  
 FFE1          test stop key  
 FFE4          get character from keyboard buffer  
 FFE7          abort all I/O channels  
 FFEA          update clock  
 FFED-FFFA    turn off cassette motors  
 FFFA-FFFB    NMI vector (mangled)  
 FFFC-FFFD    reset vector  
 FFFE-FFFF    interrupt vector

## Quellenverzeichnis

<b>Roulette</b>	<b>Ing. F. Tilgner</b>
<b>Textverarbeitung</b>	<b>Achim Konz</b>
<b>Mäuse</b>	<b>H. Holighaus</b>
<b>Lotto</b>	<b>Dr. G. Bauer</b>
<b>PET als Adressbuch</b>	<b>Juergen Rued</b>
<b>Konto 99</b>	<b>Franz Tilgner</b>
<b>Musik mit CBM</b>	<b>Winfried Hofacker</b>
<b>Word Special</b>	<b>Juergen Rued</b>
<b>Laufschrift</b>	<b>Juergen Rued</b>
<b>Fakturierung</b>	<b>Winfried Hofacker</b>
<b>Auto-Jahresabrechnung</b>	<b>ELCOMP-Fachzeitschrift</b>
<b>Druckaufbereitung</b>	<b>Prof. Dr. G. O. Hamann /ELCOMP</b>
<b>IC-Kreisberechnung</b>	<b>P. Sterrenburg</b>
<b>Roulette-Auswertung</b>	<b>Josef Nuessgens</b>
<b>Serien-, Parallelschaltung</b>	<b>Peter Sparlinek</b>
<b>Poly-Opti</b>	<b>Dr. Brewig /ELCOMP</b>
<b>Super Drei und Vier gewinnt</b>	<b>Michael Eicke</b>
<b>CMC/Wortprozessor</b>	<b>Hofacker Verlag, Holzkirchen</b>
<b>Zusammengestellt von L. Oswald</b>	

**Best.Nr.**

**1**

## **TBB - Handbuch Band 1 , W. Hofacker** **Transistor Berechnungs- und Bauanleitungshandbuch Band 1**

Das Handbuch für jeden Elektroniker. Rechenbeispiele, Berechnungsgrundlagen, Bauanleitungen, Nomogramme, Tabellen und Vergleichslisten aus den wichtigsten Bereichen der Elektronik. Ein Buch zum Einarbeiten in die Elektronik. Ein Buch zum Nachschlagen. Grundlagen Digitaltechnik, Netzgeräte und Transformatorenberechnung, Berechnung von Multivibratoren, Schmitt Trigger u. v. a. Über 130 Seiten.

**DM 19,80**

**2**

## **TBB - Handbuch Band 2 , W. Hofacker** **Transistor Berechnungs- und Bauanleitungshandbuch Band 2**

Dieses Buch ist die Fortsetzung des erfolgreichen Handbuches TBB-Handbuch Band 1. Ein Buch, das sich in der Hand des Praktikers bestens bewährt hat. Weitere neueste Schaltbeispiele und Berechnungsgrundlagen. Experimentier- und Versuchsbeschreibungen. Integrierte Spannungsregler, Wärmeableitung, Operationsverstärker Einführung, RC-Zeitglieder, Transistortester u. v. a.

**DM 19,80**

**3**

## **Elektronik im Auto , H. Gebauer**

Ein Buch für jeden technisch interessierten Autofahrer. Es zeigt Ihnen die vielen Möglichkeiten zur Verbesserung von Sicherheit, Leistung und Fahrkomfort in Ihrem Auto. Thyristorzündung, Drehzahlmesser, Beschleunigungsmesser, Geschwindigkeitswarner, Batterieladegerät u. v. a. Tips, genaue Beschreibungen, Bauanleitungen.

**DM 9,80**

**4**

## **IC - Handbuch , C. Lorenz** **Handbuch für digitale und lineare integrierte Schaltungen.**

Sensationelle Neuheit. Ein Handbuch für digitale und lineare Schaltkreise. Daten- und Auswahllisten, Vergleichslisten, Gehäuseformen, Grundlagen, Einführungsbeschreibungen, viele Schaltbeispiele, Bauanleitungen, Printvorlagen, u. v. a.

Alles über TTL-Technik, C MOS, MOS-Schaltungen, Uhren ICs, lineare Schaltungen, ein Chip-Rechner, integrierte NF-Verstärker u. v. a.

**DM 19,80**

**5**

## **IC - Datenbuch , D. Steinbach**

Daten- und Auswahllisten der gebräuchlichsten integrierten Schaltkreise. Digital und analog. Gerade bei ICs ist es wichtig die Anschlußfolgen genau zu kennen. Auf über 55 Seiten finden Sie: Die wichtigsten TTL-Schaltkreise, C-Mos Serie, lineare Schaltungen wie Operationsverstärker, Komparatoren, NF-Verstärker, Spannungsregler, Triggerschaltungen, Impulsgeber u. v. a. Weiterhin finden Sie eine C Mos-Vergleichsliste sowie Kurzdaten und logisches Verhalten dieser C MOS Elemente. Das IC-Datenbuch wird auch Ihnen ein unentbehrlicher Begleiter bei allen Arbeiten mit integrierten Schaltungen sein.

**DM 9,80**

**6**

## **IC - Schaltungen , D. Steinbach**

Hier finden Sie eine gelungene Zusammenstellung der wichtigsten Anwendungsbeispiele aus dem Bereich der integrierten Schaltungen. TTL – C MOS – Linear. Alle Schaltungen sind übersichtlich und klar dargestellt und mit einer kurzen, jedoch sehr genauen Beschreibung versehen. Viele Schaltungen sind Grundsaltungen, die man beim Umgang mit integrierten Schaltungen immer wieder benötigt. Tastenentprellung, Zähler, Impulsgeber, Codierer, Dekodierer, Datenübertragung, Serien-Parallel-Wandler, Digitalvoltmeter u. v. a.

**DM 9,80**

**7**

## **Elektronik Schaltungen, 4. völlig neu überarbeitete Auflage, W. Hofacker**

Die ideale Schaltungssammlung zum Basteln und Experimentieren. Schaltungen mit Operationsverstärkern, Spannungsreglern, TTL, C-MOS Schaltkreisen. MOS Uhr mit Wecker, elektronischer Würfel, Musik Synthesizer, Timer 555 Anwendungen, Experimentieranleitungen und viele andere hochinteressante Schaltbeispiele tlw. mit Printvorlage. 64 Seiten Inhalt.

**DM 5,-****8**

## **IC - Bauanleitungs -Handbuch -IC -KIT , C. Lorenz**

Ein Bauanleitungsbuch mit vielen hochinteressanten Bauanleitungen aus dem Bereich der LSI Schaltungstechnik. Schaltbeispiele mit Printvorlagen zum Selbstherstellen der Leiterplatten mit genauesten Beschreibungen. Hochaktuell und brandneu: Funktionsgenerator XR 2206, MOS-Uhr mit Wecker, Schlummerautomatik und programmierbarem Wecktongenerator, Sensortastenwahl, IC-Netzteil, Funktionsgenerator 8038 neuartige Transistorzündung, 35 W NF-Verstärker, Experimentieranleitung und Grundkurs über Flip Flops, Experimente mit Digitalschaltungen u. v. a. Zu allen Schaltungen finden Sie Platinvorlagen oder Sie können die Experimentierschaltungen auf der Experimentierplatine WH-1 g durchführen. Über 125 Seiten.

**DM 19,80****41**

## **Experimentierplatine mit Sockel, Stecker und Füßen Typ WH-1g für 40,28,24,16 und 14 polige DIL -Gehäuse**

**DM 79,-****9**

## **Feldeffekttransistoren, C. Lorenz**

Der Feldeffekttransistor (FET) gehört heute zu den interessantesten Bauteilen überhaupt. Wie man damit experimentiert, wie man seine Funktion versteht und wie man damit brauchbare und hochinteressante Schaltungen aufbauen kann, zeigt Ihnen dieses Buch. Grundlagen, Kennlinienfelder, Tabellen, Berechnungsgrundlagen, Rechenbeispiele, Anschlußbilder und eine Vergleichsliste für Feldeffekttransistoren bilden den Kern dieser umfangreichen Darstellung. Alles in allem finden Sie hier eine praxisnahe und komplette Arbeitsunterlage, mit der Sie im Beruf und auch im Hobby erfolgreich arbeiten können. Über 45 Seiten.

**DM 5,-****10**

## **Elektronik und Radio, C. Lorenz 4. Auflage. Völlig neu bearbeitet und stark erweitert.**

Eine Einführung in die Radiotechnik, wie man sie nicht alle Tage findet. Eine sehr geschickte gemacht Einführung mit vielen Schaltungen, Bauanleitungen und genauesten Funktionsbeschreibungen. Vom einfachen Diodenempfänger (Detektor) bis zu interessanten Sender- und Empfängerschaltungen. (Minispione) Viele hundert Bilder zeigen Ihnen genau, wie Sie beim Experimentieren vorgehen müssen. IC-Radio, IC-Sender, Antennen, Berechnungsgrundlagen, Tabellen u. v. a. Über 150 Seiten

**DM 19,80****11**

## **IC -Niederfrequenzverstärker , C. Lorenz**

Grundlagen der integrierten NF-Verstärker, Berechnung von kompletten IC-NF-Verstärkerstufen. Anwendungsbeispiele mit den interessantesten und gebräuchlichsten Standard IC-NF-Verstärkern wie TBA 800, TBA 830, usw. Printvorlagen, Auswahltabellen, Experimentieranleitungen und Anschlußbilder machen dieses Buch zu einem unentbehrlichen Begleiter für alle, die sich mit NF-Verstärkern beschäftigen wollen. Über 65 Seiten.

**DM 9,80**

**12**

## **BIS BUCH, Beispiele integrierter Schaltungen H. Bernstein**

Auf über 130 Seiten Anwendungsbeispiele mit integrierten Schaltkreisen. Zeitgeber 555, Funktionsgenerator ICL 8038, Opto Elektronik, Operationsverstärker, Analogschalter, Digital-Analog-Wandler, Analoge Rechenbausteine, Schreib-Lese-Speicher (RAM), Festwertspeicher (ROM), Speicherschaltungen, Uhrenbausteine u. v. a.

**DM 19,80****13**

## **HEH, Hobby Elektronik Handbuch , C. Lorenz**

Das Schaltungsbuch für jeden Hobbyelektroniker. Schaltbeispiele und Bauanleitungen aus dem gesamten Hobbybereich. Lichtorgeln, Eiswarngerät fürs Auto, Alarmanlagen, Metallsuchgerät, PLL-Schaltungen, Logik-Tester, Funktionsgeneratoren u. v. a. Über 55 Seiten.

**DM 9,80****14**

## **IC - Vergleichsliste , C. Lorenz Vergleichsliste für digitale und lineare integrierte Schaltkreise.**

Standard TTL, Low Power Schottky TTL, C MOS, Triacs Thyristoren, Optoelektronik, Operationsverstärker, Spannungskomparatoren, Spannungsregler, NF-Verstärker u. v. a. Funktionsvergleichsliste CMOS zu TTL. Vergleichstabelle für Transistoren und Dioden sowie Darlingtonttransistoren. Eine Vergleichsliste, die man immer wieder braucht.

**DM 29,80****15**

## **Opto -Handbuch, Handbuch für Opto- elektronik , C. Lorenz**

Das Handbuch für die gesamte Optoelektronik. Eine Einführung und ein ideales Nachschlagwerk. Grundlagen, Definitionen aller Kenngrößen, Opto-Lexikon, Berechnungsgrundlagen, Rechenbeispiele, Schaltbeispiele: Lichtsender, Lichtempfänger, Anzeigen, Infrarot Detektoren, Lichtmeßgerät, Optokoppler, Pegelschalter, Opto-Vergleichsliste. Anschlußbilder wichtiger 7-Segment-Anzeigen u. v. a. Über 106 Seiten.

**DM 19,80****16**

## **C MOS Einführung, Entwurf, Schaltbeispiele, Teil 1 , H. Bernstein**

Vom C MOS Gatterbaustein über Schieberegister und Zähler bis hin zum C MOS Schreib-Lesespeicher. Insgesamt werden neunzehn interessante und bekannte C MOS Schaltkreise beschrieben. Zu jedem Bauelement sind genaue Daten, Schaltbild und Anwendungsbeispiele angegeben. Im großen Applikationsteil finden Sie: C MOS-Kippstufen, Addierwerke und Rechenschaltungen, Digital Analog Wandler, Schieberegister für analoge Spannungen, Multiplexsysteme für analoge Signale u. v. a. Eine komplette Einführung und gut geeignet für das Selbststudium der C MOS Technik. Über 140 Seiten.

**DM 19,80****17**

## **C MOS Entwurf und Schaltbeispiele, Teil 2 , H. Bernstein**

Fortsetzung von Teil 1. Anwendungsbeispiele mit genauen Schaltungsbeschreibungen und Bauelementunterlagen. Daten, Anschlußbelegungen weiterer wichtiger hochintegrierter C MOS Elemente. Ein komplettes Arbeits- und Experimentierbuch. C-MOS Uhrenschaltungen, Schieberegisterschaltungen, Parallel-Serien Umsetzung, statische und dynamische Speicherschaltungen, Zählschaltungen, Digital Analog Wandler, Analog Digital Wandler, Digital Voltmeter, I/O Registerschaltungen, Codier und Dekodierschaltungen. RAM und ROM Anwendungen. Über 140 Seiten.

**DM 19,80**

**18**

## **C MOS Entwurf und Schaltbeispiele, Teil 3 , H. Bernstein**

Fortsetzung von Teil 2. Eine sehr umfangreiche Applikationssammlung mit hochintegrierten C MOS Elementen. Rechnerschaltungen, Speicher- und Steuerschaltungen, Multiplex- und Datenbussysteme, Uhrenschaltungen, PLL-Schaltungen, Liquid Cristal Anzeigen und deren Treiberschaltungen, Optoelektronik in Verbindung mit C MOS. Grundlagen, Aufbau und Wirkungsweise der Prozeßrechentechnik, Arithmetische Logische Einheiten (ALU) und andere wichtige Funktionen aus der Prozeßrechentechnik. RAMs, ROMs und FIFO-Speicherschaltungen. Über 140 Seiten.

**DM 19,80****19**

## **IC Experimentier Handbuch - IC -EX, C. Lorenz C. Lorenz**

Eine sehr umfangreiche Schaltungssammlung und Bauanleitungssammlung mit neuesten integrierten Bausteinen. Neue, jedoch beim Fachhandel erhältliche Standard ICs. Rechner-schaltungen, Mikroprozessoren, I/O Schaltungen, druckende und anzeigende Rechner, Stoppuhren, Zählerschaltungen, Digitalvoltmeter, professioneller Synthesizer, Hilfsschaltungen für den Elektronik Experimentier, Analog Digital Wandler, Frequenzzähler u. v. a. hochinteressante Bauanleitungen. Viele Schaltungen können auf der IC KIT Experimentierplatine WH-1g aufgebaut werden. Über 120 Seiten.

**DM 19,80****20**

## **Operationsverstärker, Grundlagen und Schaltbeispiele , C. Lorenz**

Dieses Buch umfaßt das gesamte Gebiet der linearen Schaltungstechnik und stellt ein in dieser Preislage bisher noch nie dagewesen Nachschlagwerk und Einführungshandbuch dar. Bestens geeignet für das Selbststudium. Nach einer pädagogisch geschickt gemachten Einführung folgen theoretische Arbeitsunterlagen und die zugehörigen Schaltbeispiele mit Daten und Gehäuseanschlüssen. Dieses wertvolle Buch dürfte seinen Platz auch bei Ihren Arbeitsunterlagen finden, und wird dann immer von Nutzen sein, wenn es um die Lösung von nicht routinemäßigen Aufgaben geht.

**DM 19,80****21**

## **Digitaltechnik Grundkurs (TTL -C MOS - MOS und Software) , C. Lorenz**

Ein Einführungskurs in die Digitaltechnik für Anfänger und Fortgeschrittene. Ein Fachbuch für den programmierten Selbstunterricht. Der ideale Kurzlehrgang für das Selbststudium. Der Kurs vermittelt Ihnen alle wichtigen Grundkenntnisse vom TTL-Gatter bis zum Mikroprozessor und Lösung von Schaltungsaufgaben durch Software. Viele Versuchsaufbauten und Experimente aus diesem Kurs können auf der IC-KIT Platine WH-1g durchgeführt werden. Grundlagen, Gatter, Zähler, programmierbare Zähler, IC-Tester, Schieberegister, Speicher, Mikroprozessoren u. v. a. Über 130 Seiten.

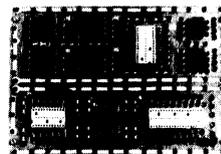
**DM 19,80****41**

## **Experimentierplatine WH-1g dazu, Best.Nr. 41**

**DM 79,-**

Experimentierplatine WH - 1 g  
Abmessungen: 210 mm x 150 mm

Experimentierplatine WH - 1 g  
fertig aufgebaut und mit Sockeln  
bestückt.



**22**

## **Mikroprozessoren, Eigenschaften und Aufbau Teil 1 , H. Bernstein**

Grundlagen, Eigenschaften und Aufbau von Mikroprozessoren. Organisation von Rechen-einheiten und Mikroprogrammen. Programmierung und Klassifizierung von Mikroprozessoren. Ablaufdiagramm, Flußdiagramm. Ein Chip-Technik und Multi Chip-Technik, Transfer- und Sprungfunktionen. Speichertechnik: RAMs, ROMs, FIFO, FILO. Programmierbare logische Arrays (PLA)

Anwendungsbeispiele und Anwendungsbereiche. Über 120 Seiten.

**DM 19,80**

**23**

## **Elektronik Grundkurs (Kurzlehrgang Elektronik) , C. Lorenz**

Eine leichtverständliche und pädagogisch geschickt gemachte Einführung in die Technik der elektronischen Schaltungen. Ein Kurzlehrgang und Schnellkurs zugleich. Aber auch ein recht brauchbares Nachschlagewerk für den fortgeschrittenen Elektroniker. Mit wenig Mühe können Sie sich hier die Grundkenntnisse der elektronischen Schaltungspraxis aneignen. Das Buch schafft die Voraussetzungen für ein erfolgreiches und sicheres Arbeiten mit interessanten Schaltkreisen modernster Technologien. Unentbehrlich für das Experimentieren mit den heutigen modernen hochintegrierten Schaltkreisen. Über 150 Seiten.

**DM 9,80**

**24**

## **Mikrocomputer Technik, Hans Peter Blomeyer-Bartenstein**

(völlig neu überarbeitete Auflage, Herbst 1979)

In diesem Buche finden Sie eine umfassende, einführende und weiterführende Hilfe zum Einstieg in die Mikrocomputertechnik mit vielen Schalt- und Programmierbeispielen. Als praktische Betrachtungsgrundlage dient das supermoderne Microcomputerkonzept Z80A von ZILOG. Das Buch geht auf alle wichtigen Zusammenhänge ein und erklärt diese dem Leser so ausführlich, daß kaum noch Fragen offen bleiben. Über 240 Seiten

**DM 29,80**

**25**

## **Hobby Computer Handbuch, C. Lorenz** **Eine leicht verständliche Einführung in die Microcomputertechn**

Diese sehr umfangreiche Einführung in die Microcomputertechnik dürfte zu diesem Preise einmalig sein. Auf über 450 Seiten finden Sie: Grundlagen der Computer- und Microcomputer-Technik, Was ist ein Microcomputer?, Microcomputer KITS, Einplatinen-computer, KIM, MIKIT, Z80 KIT, 6800 KIT, NEC 8080 KIT u. v. a. Genaue Beschreibungen der wichtigsten Mikroprozessortypen. Zusammenstellung und Beschreibung der modernen Personal Computer. (IMSAI, CROMEMCO, CAT, OSI, POLY 88 u. v. a.) Kompaktcomputer wie PET und TR-8. Interface-Techniken, Ein/Ausgabegeräte, ROMs, RAMs, Programmiergeräte für PROMs. Löscheräte für EPROMs u. v. a. mehr. Das ideale allumfassende Buch für den Microcomputertechniker. Für Industrieanwendung ebenso geeignet, wie für den Hobby-Computer Fan. Über 450 Seiten

**DM 29,80**

**26**

## **Mikroprozessor Teil 2, H. Bernstein**

Die Fortsetzung unseres ersten so erfolgreichen Buches über Mikroprozessoren. Technologie von Mikroprozessor- und Speicherbausteinen. Festwertspeicher, PROM, REPROM, FIFO, Schieberegister, MPR-Register, ARL-Register, SAR-Register. Aufbau eines Mikroprozessorsystems mit 8080, RAM- und ROM Schnittstellen. Befehlssatz 8080. Über 120 Seiten.

**DM 19,80**

**27**

## **Mikroprozessor Software Handbuch MSH, C. Lorenz**

Grundlagen und Einführung in die Mikroprogrammierung. Grundlagen und Einführung in die wichtigsten Programmiersprachen (BASIC, FORTRAN, ASSEMBLER-Sprachen) Zusammenstellung der wichtigsten Befehlslisten: 8080, Z80, M 6800, National, Fairchild, etc.

Ein Software Handbuch für jeden der mit Mikroprozessoren oder Mikrocomputern zu tun hat. Über 200 Seiten. **Preis DM 29,80**

**28**

## **Lexikon und Wörterbuch für Elektronik und Mikroprozessortechnik, LEM, C. Lorenz**

Ein Hilfs- und Arbeitsbuch für jeden der sich heute mit der modernsten Elektronik beschäftigt. Viele engl. Ausdrücke werden heute in der Elektronik, Computer- und Mikroprozessortechnik verwendet und oft fehlt uns eine genaueste und präzise Erklärung. Dieses Buch übersetzt Ihnen den englischen Fachausdruck und gibt Ihnen zusätzlich noch eine deutsche Erläuterung und Erklärung dieses Begriffes und was es damit auf sich hat. Ein Lexikon und Wörterbuch in einem einzigen Buch vereinigt. Das Buch, das Sie schon lange gesucht haben. Ca. 250 Seiten. **Preis DM 29,80**

**29**

## **Mikrocomputer Datenbuch, C. Lorenz Zusammenstellung der wichtigsten Mikroprozessordaten.**

Eine übersichtliche und sehr informative Zusammenstellung der wichtigsten Mikroprozessorbausteine auf dem Markt. 8080A, 8085, 8048, Z80, Z8, 6500, 6800, 2650, 1802, F8, 3870, SC/MP, PACE u. v. a. Daten, Anschlußbilder, wichtige technische und elektrische Daten, Architektur, grundlegende Eigenschaften. Zu jedem Mikroprozessor werden dann auch noch die peripheren Bausteine sowie RAM und ROM Elemente behandelt. Das ideale Handbuch für jeden modernen Elektroniker.

**DM 49,-**

**30**

## **Aktivtraining - Mikrocomputer**

Der ideale Einführungskurs in die Mikrocomputertechnik anhand des Mikroprozessors 8080 (8085). Viele interessante Anwendungsbeispiele. Ideal zum Selbststudium geeignet. Auch die wichtigsten peripheren Einheiten werden besprochen und aufgezeichnet. Erscheint Ende 1979/ Anfang 1980 als gebundenes Buch und nicht wie vorher angekündigt als Sammelordner. Umfang ungefähr 400 – 500 Seiten. **DM 49,-**

**31**

## **57 Programme in BASIC, C. Lorenz**

Ein Buch mit technisch-wissenschaftlichen Programmen und einer großen Anzahl von Spielprogrammen in BASIC. (Games) Ein Buch für jeden, der sich mit dem faszinierenden Hobby der Mikrocomputertechnik befassen will. Alle Listings sind in BASIC und können auf den meisten Personal Computer Systemen gefahren werden.

**DM 39,-**

**32****ATARI BASIC Handbuch**

Das komplette Einführungs- und Programmierhandbuch für die neuen ATARI-Computer ATARI 400 und ATARI 800. Anhand von vielen Beispielen wird die Leistungsfähigkeit des ATARI Computer-Systems gezeigt. Auch für den Anfänger ideal als Einstieg gedacht. Die ATARI-Computer sind mit Shepardon's BASIC ausgerüstet. Erscheint voraussichtlich  
Erscheint Ende 1980 DM 29,80

**33****Mikrocomputer Programmierbeispiele für 265  
Dr. J. Hatzenbichler**

Eine Einführung in die Programmierung von Mikrocomputern an Hand des Prozessors 2650 von Signetics. Viele Programmierbeispiele in Maschinensprache, die Sie auf einem preiswerten Mikroprozessorsystem MIKIT 2650-P2 ausführen können. Zeitschleifenprogramme, Blinkschaltung, Lauflicht, Stufenzähler, Elektronischer Würfel, Stoppuhr, Reaktionszeittester, Computer Musik Programm u. v. a. Jeder Befehl wird genau erläutert und an Hand eines ausführlichen Flußdiagrammes erklärt. Jedes Programm liegt in Form eines Computer-Listings vor. Jedes Programm ist genauestens beschrieben. Sie können so auf einfache Weise die Zusammenhänge erkennen und erwerben damit die Grundkenntnisse zur Erstellung Ihrer eigenen Programme. Sie erkennen, wie man mit nur einer Schaltung (Mikrocomputer) unendlich viele praktische Anwendungsschaltungen realisieren kann. Zu diesem Buche ist auch ein komplett aufgebautes und getestetes Mikrocomputersystem erhältlich, auf dem Sie alle beschriebenen Programme selbst ausführen können. Über 120 Seiten DM 19,80

**34****TINY BASIC Handbuch**

Eine Programmiersprache ist um so leistungsfähiger, je einfacher eine gewünschte Funktion dargestellt werden kann. TINY BASIC ist eine abgemagerte BASIC-Version, die mit ca. 2 K RAM auskommt. Dieses Buch zeigt Ihnen, wie Sie eine solche preiswerte BASIC-Version auf Ihrem KIM-1 oder einem anderen 6502-System implementieren können. Komplette Hard- und Softwarebeschreibungen, Programmiertricks, Programmierbeispiele und vieles andere mehr. DM 19,80

**35****Der freundliche Computer**

Ein Microcomputerbuch für jedermann. Dieses Buch sollte jeder lesen, der auch in Zukunft noch mitreden möchte. Was macht man mit einem Microcomputer? Die verschiedenen Computersprachen? Welche Computersprache ist die beste? Wie lerne ich eine Computersprache? Zusammenfassung verschiedener Computersprachen wie BASIC, MUMPS, PL/M, FOCAL, FORTRAN, ASSEMBLER. Vergleich dieser Computersprachen untereinander und Gegenüberstellung. Einkaufsführer für Microcomputer und Peripherie. Besprechung der einzelnen Teile des Microcomputers, ausführliches Stichwortverzeichnis der Fachausdrücke und v. a. m. Erscheint Mitte 1981 DM 29,80

**36****Microcomputer und Roboter**

Ein Buch für diejenigen, der sich externe Schaltungen für Microcomputer bauen möchte, die roboterartige Funktionen ausführen können. Spracherkennung, Analog-/Digital-Wandler, Digital-/Analog-Wandler, Ultraschallsensoren, Lichtschranken, Tonerzeugung u. v. a. Erscheint Ende 1981 DM 29,80

**104****1000 Elektronik Schaltungen**

Ein ideales Handbuch für jeden, der öfter eine Schaltung zur Lösung eines bestimmten Problems sucht. Tausend Schaltungen aus fast allen Bereichen der Elektronik. Industrielle Steuerschaltungen, Microcomputer, Peripherie, Hobby-Elektronik-Schaltungen u. v. a. mehr. Ein Buch, daß bei keinem Elektroniker fehlen sollte. Erscheint Mitte 1981 **DM 49,-**

**103****Oszillographen Handbuch**

Ein Buch für jeden, der seinen Oszillographen optimal nutzen will. Der Anfänger, der noch nie einen Oszillographen benutzt hat, wird auf einfache Weise mit der Technik und Handhabung vertraut gemacht. Auch die Anwendung in Zusammenhang mit den modernen Mikrocomputersystemen wird beschrieben. Oszillograph als alphanumerisches Darstellungsgerät u. v. mehr. Erscheint ca. Mitte 1981 **DM 19,80**

**105  
1****TTL - Experimentierbuch  
Eine kleine Einführung in die Digitaltechnik**

Grundlagen der Digitaltechnik kurz erklärt. Bauanleitung für einen praktischen Logik-Tester. Viele Experimente und Anwendungsschaltungen mit dem TTL Gatterbaustein 7400. Das ideale Einführungsbuch in die Digitaltechnik. **DM 5,-**

**106  
1****CMOS - Experimentierbuch  
Eine kleine Einführung in die CMOS Schaltungstechnik**

CMOS Grundlagen, Behandlungshinweise für CMOS Bausteine, Zusammenstellung der wichtigsten CMOS Bausteine und deren Anschlußbilder. Viele praktische CMOS Schaltbeispiele. Ideal für jeden Elektroniker. **DM 5,-**

**107****Praktische Antennentechnik, C. Lorenz**

Ein Buch für jeden Funkamateurliebhaber oder Hobbyfunker. Grundlagen, Einführung, praktische Beispiele, Berechnungsgrundlagen u. v. a. mehr. Erscheint Ende 1981 **DM 19,80**

**108****Handbuch für SC/MP – Das Handbuch für INS8060 und INS8070, C. Lorenz**

Eine Anleitung zum Aufbau eines eigenen Microcomputersystems. Komplett mit allen Unterlagen (Schaltbilder, Printvorlagen, Anleitungen). Systemdaten: 4/8K RAM, residentes BASIC, residenter Assembler/Editor/Debugger, TV-Interface, Floppy Disk Interface, Cassetten-Interface. **DM 29,80**

**109****6502 Microcomputer Programmierung, P. Heuer,**

Ein deutsches Anleitungsbuch zum Einstieg in die Microcomputertechnik. Als System kann ein KIM-1 zur Ausführung der Programmierbeispiele verwendet werden. Die Maschinenprogramme lassen sich jedoch auch auf jedes 6502 System umschreiben. Daher ist dieses Buch auch für alle anderen 6502-System-Besitzer interessant. (AIM, SYM, CBM, Challenger, APPLE, ATARI, PET und PC100 sowie PC1000. **DM 29,80**

**110****Programmierhandbuch für PET, C. Lorenz**

Ein Handbuch für jeden PET-Besitzer und solche die es werden wollen. Grundlagen, Einführung und viele Programmierbeispiele. Einführung in die Maschinensprache, Ein-/Ausgabeprogrammierung, Spracheingabe für PET, Graphik mit dem PET, Computermusik, Programmierung des IEC-Ports, viele Programmbeschreibungen, Datenblätter, Programmbeschreibungen von interessanten Geschäftsprogrammen. 325 Seiten **DM 29,80**

**111****Programmieren mit TRS-80, M. Stübs**

Ausführliche Beschreibung des TRS-80 Computers von Radio Shack. Betriebssystem, Level II BASIC, Graphik, Mathematik und Logik, Programmiertricks, TRS-80 Floppy, Verbindung zur Außenwelt, RAM-Erweiterung u. v. a. mehr. Viele Programmierbeispiele mit ausführlicher Beschreibung runden das Buch ab und machen es zu einer wichtigen Informationsquelle für jeden, der vor der Anschaffung eines TRS-80 steht. Dem TRS-80-Besitzer wird es stets ein wertvoller Begleiter sein. **DM 29,80**

**112****PASCAL Programmierhandbuch, C. Lorenz**

Eine leicht verständliche Einführung in die PASCAL-Programmiersprache mit vielen Beispielen. Auch auf eine TINY-PASCAL-Version wird eingegangen. Viele Programmierbeispiele. **DM 29,80**

**113****BASIC Programmierhandbuch, C. Lorenz**

Einführung und Nachschlagewerk. Speziell für die BASIC-Versionen der modernen Microcomputersysteme. Jeder Befehl wird ausführlich beschrieben und ein Beispielprogramm gezeigt. Sehr übersichtlich und praktisch. Am Schluß finden Sie ein komplettes BASIC-Programm, das Ihnen über einen Computer BASIC lehrt. Mit Begleittext. **DM 19,80**

**114****Der Microcomputer im Kleinbetrieb, L. Oswald**

Wie setzt man die modernen, preiswerten Microcomputer im Kleinbetrieb ein? Dieses Buch gibt Ihnen Aufschluß, wie Sie mit Microcomputersystemen in der Preisklasse zwischen DM 2.000,- und DM 20.000,- wichtige Aufgabe in Ihrem Klein- oder Mittelbetrieb bewältigen können. Inventuraufnahme, Inventurbewertung, Adresslisten, Rechnungen schreiben u. v. mehr. Erscheint Anfang 1981 **DM 39,80**

**115****FOCAL Programmierhandbuch, L. Oswald**

FOCAL ist eine Programmiersprache die von Digital Equipment entwickelt wurde. Sie ähnelt etwas der BASIC-Sprache, ist jedoch wesentlich leistungsfähiger. Hauptsächlich mathematische Probleme lassen sich mit ihr wesentlich einfacher lösen. Wie man diese Sprache auf einem 6502 Computersystem (KIM-1) mit Speichererweiterung implementiert, soll dieses Buch zeigen. Erscheint Mitte 1981 **DM 19,80**

**116****Einführung 16-Bit Microcomputer, C. Lorenz**

Eine kleine Einführung in die 16-Bit-Microcomputersysteme von Motorola, Zilog, Texas Instruments, Intel und Rockwell. Erscheint Mitte/Ende 1980 **DM 19,80**

**117****FORTRAN für Heimcomputer, C. Lorenz**

Einführung in die FORTRAN-Programmiersprache mit vielen Beispielen. Grundsätzliches über die verschiedenen Microcomputersysteme, die bereits mit FORTRAN-Compiler lieferbar sind. Allgemeine Übersicht, Tips und Hinweise. Mitte/Ende 1980 **DM 19,80**

**118****Programmieren in Maschinensprache mit dem 6502**, C. Lorenz und R. Lullus

Eine sehr ausführliche Beschreibung der 6502 Mnemonics mit kleinen Beispielen. Komplettes Assembler-Listing für CBM 8K alte ROMs und CBM 16, 32K neue ROMs. Text-Editor, Disassembler und Binder. BASIC-Listings mit genauer Beschreibung. **DM 98,-**

**119****Programmieren in Maschinensprache (Z80)**, C. Lorenz

Eine sehr ausführliche Einführung in die Z80 Maschinensprache. Die Beispiele können mit Hilfe des TRS-80 Level II sowie dem T-BUG von TANDY und den T-BUG-Erweiterungen (IN LOCO, T-STEP, T-LEGS) ausgeführt werden. Ein unentbehrliches Buch für jeden, dem die BASIC-Programmiersprache von der Geschwindigkeit her zur Lösung seiner Aufgaben nicht mehr ausreicht. **DM 49,-**

**120****Anwenderprogramme für TRS-80**, M. Stübs

Ein Buch, voll mit interessanten Anwenderprogrammen für TRS-80 Level II 16K (teilweise Diskette und/oder Cassette). Hauptsächlich Programme für den Manager, Geschäftsmann, Klein- und Mittelbetrieb. Auch einige interessante Spiele sind enthalten. Termin-kalender, Reservierungsprogramm für Omnibusunternehmen und Hotels usw. **DM 29,80**

**121****Microsoft BASIC-Handbuch**, Microsoft

Die deutsche Übersetzung des erfolgreichen Microsoft BASIC-Handbooks. Leicht verständliche Einführung mit vielen interessanten Programmbeispielen. **DM 29,80**

**122****BASIC für Fortgeschrittene**, C. Lorenz

Ein Buch für denjenigen, der tiefer in die BASIC-Programmiersprache einsteigen möchte. Sehr viele nützliche Unterroutrinen, Programmiertricks und Hinweise. **DM 39,-**

**123****IEC Bus-Handbuch**, von Max P. Gottlob

Ein Handbuch und Nachschlagwerk für alle Besitzer von Computern mit IEC (IEEE 488 Bus). Dazu gehört auch der PET sowie alle CBM-Computer. Grundlegen, das BUS-System, Meßdatenübertragung, Adressierung eines Instrumentes, kleines IEC-BUS-Lexikon u. v. a. ISBN 3 921 682-73-8 **nur 19,80 DM**

**124****Programmieren in Maschinesprache mit CBM**

An Hand eines praktischen Beispiels (Sortierroutine) wird der Unterschied zwischen BASIC und Maschinenprogrammen gezeigt. Das Maschinenprogramm kann mit dem leistungsfähigen MONJANA/1 Monitor In ROM erstellt werden. Am Schluß finden Sie weitere wichtige Informationen wie Dez/Hex- Umrechnungstabelle, Befehls-listen, ASCII-Tabelle sowie eine ROM-Vergleichsliste zwischen 8k PET und den neuen CBM-Maschinen. 60 Seiten. ISBN 3-921 682-70-3. **19,80 DM**

**125****ELCOMP, Fachzeitschrift für Microcomputertechnik**

ISSN-Nr. 0171-0958. Die kompetente Fachzeitschrift für das moderne Gebiet der Microcomputertechnik. Erscheint 10 x pro Jahr. Jahrespreis **DM 59,-** incl. MwSt. Porto und Verpackung. Wer die neuesten Informationen aus diesem Gebiet für sich nützen möchte, muß ELCOMP lesen. Technische Tips, Software, Bauan-leitungen, Programmiertricks, Systembeschreibungen. Jeden Monat brandneu. **4,50 DM**

**126****ELCOMP Doppelheft**

Von den zehn Heften pro Jahr erscheint im Juli/August und November/Dezember ein Doppelheft. **9,- DM**

**127****Einführung in die Microcomputer Programmierung mit 6800**

Eine sehr gute Einführung in die Microcomputertechnik mit Hilfe des Mikroprozessors 6800. Ausführliche Erklärungen mit vielen Beispielen und Anleitungen. Theoretische Grundlagen. CPU-Architektur, Befehlsatz, Systemaufbau. Hilfsmittel der Programmierung, Trainingsprogramme, Systemkomponenten, FIRMWARE. Ein komplettes Monitorprogramm (Betriebssystem) ist als Listing enthalten. Über 250 Seiten. ISBN-Nr. 3-921682-79-9 **49,- DM**

**128****Programmieren mit dem CBM**

Ein Hand- und Programmierbuch für alle CBM-Besitzer der 3000 und 8000er Serie. Viele Tricks und Programmierbeispiele, Anleitungen. Erscheint Ende 1980. **29,80 DM**

**129****ELCOMP Leser Programmierhandbuch**

Hier fassen wir die besten Programme unserer ELCOMP-Leser zusammen. Programme für PET, CBM, TRS-80, AIM, SUPERBOARD, C4P, EXIDY, SHARP, MZ80K, Apple II, NASCOM I u. II und TI 99/4 werden als Listing mit kurzer Beschreibung allen Lesern zugänglich gemacht. DIN A 4 ca. 300 Seiten. Erscheint Anfang 1981. **69,- DM**

**130****Programmierbeispiele für CBM**

Ein Buch mit vielen BASIC-Programmen für CBM und PET. Spiele, Geschäftsbereich, Erziehung und Wissenschaft, Utilities, trickreiche Programme, Hilfen für Maschinensprachenprogrammierung. Viele Programme für wenig Geld. Erscheint Ende 1980 **19,80 DM**

**131****Cobol für Anfänger**

Eine Einführung in die Cobol-Programmierung für den Microcomputer-Besitzer. Erscheint Anfang 1981. **19,80 DM**

**132****CP/M-Handbuch**

Grundlagen, Einführung, Hilfs- und Handbuch für jeden der mit dem "Software-Bus" arbeiten möchte. Ideal auch für Anfänger. Praktisches Handbuch für den Profi. Anfang 1980 **29,80 DM**

**133****Welches Betriebssystem brauche ich?**

Ein Leitfaden und Handbuch für den Anwender der gehobenen Microcomputer-Klasse. Anfang 1981. **19,80 DM**

**150**

**Care and Feeding of the Commodore Personal Electronic Transactor**, edited by Silver Spur

Ein Buch für den PET-Besitzer und Hardware-Fan. Viele wichtige Kniffe und Tricks, die Sie benötigen, wenn Sie selbst ihren PET-Computer erweitern wollen. **DM 19,80**

**151**

**8K Microsoft BASIC Reference Manual**

Eine Einführung in die 8K Microsoft BASIC-Version mit vielen Beispielen. Dieses Buch braucht jeder, der einen Microcomputer mit Microsoft-BASIC besitzt. **DM 19,80**

**152**

**Expansion Handbook for 6502 and 6800**, S. Roberts

Viele interessante periphere Schaltungen für 6502 und 6800 Microcomputersysteme. Ein Buch das jeder benötigt, der sein System selbst erweitern möchte und noch mehr Vorteile daraus ziehen will. RAM-Platine, ROM-Platine, CPU-Platine, Ein-/Ausgabeplatine u. v. a. **DM 19,80**

**153**

**Microcomputer Application Notes**, Intel

Eine Zusammenfassung der interessantesten Applikationsberichte von Intel. Programmierbarer Interfacebaustein 8255, Anwendung des USART 8251, entwickeln mit statischen RAMs, Benutzung der seriellen Ein-/Ausgabeleitungen des 8085, Anwendung von 5V EPROMs, Anwendung des 2708 EPROMs u. v. **DM 29,80**

**154**

**Complex Sound Generation using the SN76477**

Dieses Buch zeigt Ihnen, wie man mit dem SN76477 die interessantesten und schönsten Töne erzeugen kann. Vogelgezwitscher, Dampfeisenbahn, Autorennen, Raketenabschuß, Granateneinschlag u. v. a. mehr. Viele Schaltungen mit genauen Beschreibungen. Kurzdaten. **DM 19,80**

**155**

**The First Book of 80-US (TRS-80)**

In diesem Buch haben wir noch einmal einige wichtige und interessante Programme sowie Artikel aus der amerikanischen Spezialfachzeitschrift für TRS-80 (80-US, The TRS-80 Users Journal) zusammengestellt und als Buch herausgegeben. Dieses Buch sollte nur kaufen, wer die Einzelhefte dieser Zeitschrift von Vol. I Nr. 1 bis Vol. II Nr. 5 noch nicht besitzt. Das Buch enthält sehr wertvolle, nützliche Informationen und phantastische Software: Kleinbuchstaben für TRS-80, Biorythmus für TRS-80, Bowlingspiel, Adressliste, Bildzeichnungsroutine, komplettes Geschäftsprogramm für Friseurläden, Telefon und Namensverzeichnis. Texteditor, Persönlichkeitsprofil u. v. a. **DM 19,80**

**156**

**Small Business Programs**, S. Roberts

Ein Buch für denjenigen, der die modernen Microcomputer (speziell TRS-80, APPLE, PET, North Star, Challenger) zur Rationalisierung in seinem Klein- oder Mittelbetrieb einsetzen möchte. Viele nützliche Tips, Hinweise und Programmierbeispiele. Dieses Buch sollte jeder Geschäftsmann und Microcomputerfreund besitzen. **DM 29,80**

**157**

**The first Book of Ohio Scientific Vol. I**

Das erste weltweit produzierte Buch für die erfolgreiche Ohio Scientific Challenger Computerserie. Grundlagen, viele Programmiertricks, Hardwaretips, Umbauanleitungen, Programmierbeispiele u. v. a. 186 Seiten. Glanzumschlag **19,80 DM**

**158****The second Book of Ohio Scientific**

Eingehende Beschreibungen über praktische und geschäftsorientierte Software. Speicher Test Programm, Tricks und Tips für Disketten-Anwender. Mini-Floppy-Expansion u. v. a. 159 Seiten. **19,80 DM**

**159****The third Book of Ohio Scientific**

Fortsetzung der oben beschriebenen Reihe. In diesem Buch findet der Hardware-Spezialist das was er schon länger gesucht hat. Wichtige Informationen für die Systemerweiterung, Schaltpläne u. v. a. 180 Seiten. Herbst 1980 **19,80 DM**

**160****The fourth Book of Ohio Scientific**

Ein Buch voll mit Programme für das Superboard, C4P, C4PMF und C28P. Die Softwarequelle für jeden Challenger Fan. Alle Programme sind getestet und auch auf Cassette verfügbar. 170 Seiten Listings u. Beschreibungen. Herbst 1980.

Buch **29,80 DM**Cassette (extra) **29,80 DM****161****The fifth Book of Ohio Scientific**Frühjahr 1981. **19,80 DM****162****ATARI GAMES IN BASIC**

Ein Büchlein mit vielen Programmen für den ATARI-Computer (400 + 800). 64 Seiten. Ende 1980 **19,80 DM**

**163****The peripheral Handbook**

Ein Handbuch über Drucker, Floppys, Hard-Disks, und alles was Sie um den Micro-Computer herum benötigen. Tricks und Kniffe. Technische Beschreibung. Anfang 1981. **29,80 DM**

# ELCOMP

## -INHALTSVERZEICHNISSE

Inhaltsverzeichnis der zurückliegenden  
ELCOMP - Hefte

**Nr. 1 September 1978 3,50 DM**

Symbole  
Microcomputer Grundkurs  
North-Star Horizon Computer  
Software für Z 80 Systeme  
Selbstbaucomputer mit Z 80 Bauan-  
leitung für ein eigenes Microcomputer-  
System, Teil I  
8085 Microcomputer-KIT  
Microcomputer-Anwendung im Kleinbe-  
trieb  
S-100 Bus Adapter für PET 2001  
Exidy - Ein neues Microcomputer-  
System mit Z-80  
Hauskauf mit BASIC  
Heimcomputer-Vergleichsliste  
BASIC-Programm FACTORS  
Der S-100 Bus (Beschreibung)  
Viele Programme für den PET  
Software für 8080/Z-80/6800 Systeme  
AIM-Computer von Rockwell  
Erweitern Sie Ihren KIM-1  
Der VIM-1 ist da

**Nr. 2 Oktober 1978 3,50 DM**

Microcomputer Grundkurs  
Wie lernt man BASIC  
TINY BASIC, klein aber oho  
Selbstbaucomputer Z 80, S-100 RAM-  
Karte  
Supermonitor für den PET  
Die Silicon Valley Story  
Programmierung von Bewegungsabläufen  
TRS-80 Microcomputer des Monats  
Ein IK-Monitor für Z-80 (Betriebssystem)  
Musik für den PET  
Microcomputer Lexikon  
Ein Monitor für 6502  
Breakpoint-Routine  
Joystick Programmierung  
Microchess  
Ein leistungsfähiger Monitor für den PET  
BASIC Plus  
Messeberichte  
Personal Computing

**Nr. 3/4 Nov./Dez. 1978 8,00 DM**

Graphik mit dem PET  
Wie lernt man BASIC, Teil II  
Preiswertes Datenerfassungssystem mit  
Scampi  
IPS - Die Programmiersprache der Nach-  
richtensatelliten f. Funkamateure  
Apple II - Microcomputer d. Monats  
PSI - Ein komfortables Programmsystem  
Ein linearer Joystick für PET  
NASCOM 1  
KIM-1 wird noch leistungsfähiger

Fädelttechnik für Elektronik  
Neues SC/MP Microcomputersystem  
Microcomputer-Lexikon  
Silicon-Valley-Story, Teil II

Biorythmus  
Z-80 Selbstbaucomputer Ein-/Ausgabe-  
platine  
Drucker für SC/MP  
S-100 Bus Mutterplatine  
Messebericht WESCON  
Industrie und Personal Computer EXPO  
**Nr. 1 Januar 1979 3,50 DM**  
Bubble-Speicher von Rockwell  
Programmiertricks für PET  
Computerspiele  
Programmierexperimente für PET  
Wie lernt man BASIC, Teil III  
ASCII - Selbstbautastatur  
PASCAL  
Biorythmus für PET  
Microstar - Microcomputer d. Monats  
Z-80 Selbstbaucomputer - Bauanleitung  
für ein eigenes Microcomputersystem,  
Teil 4 - EPROM-Karte  
APL für Z-80  
TINY-BASIC f. NASCOM-1  
Floppy Disk für Apple II  
PROTEUS III  
4k C MOS RAM

**Nr. 2 Februar 1979 4,50 DM**

Computer-Musik  
Wie lernt man BASIC, Teil IV  
Kraftstoffverbrauch und Computer  
Löschen des Speichers beim 8080/8085  
Dual-Joystick für den PET  
Was der Elektronik-Fachhändler über  
Microcomputer wissen muß  
Spielen Sie Lotto ?  
Einfache Programmorganisation für PET  
Ein-/Ausgabeprogrammierung mit PET  
16-Bit-Microprozessoren  
Schaltregler für Microcomputer  
Einfaches Statistikprogramm für PET

**Bereits vergriffen !**

**Nr. 3 März 1979 4,50 DM**

Challenger IP und Superboard II  
ATARI-Computer  
Suchlauf  
SC/MP-Programm BINBCD  
Sprungentfernung bei relativer Adres-  
sierung  
Laufzeitreduzierung b. BASIC-Interpreter  
Heimcomputer und Bildungswesen  
Lernen für die Anwendung von Micro-  
prozessoren  
Umwandlung von BCD-Code in 7-Seg-  
ment-Code  
Microcomputer des Monats: MSI 6800  
von Midwest Scientific Instruments

Wie lernt man BASIC, Teil V  
S-100 Adapter für TRS-80  
Computer-Musik

**Nr. 4 April 1979 4,50 DM**

KIM-Software - Es leber der KIM-1  
Disassembler und Editor für KIM-1  
CAI Computer Assisted Instruction  
Optimieren Sie Ihre Programmiertech-  
nik  
Jetzt kommen die Roboter  
Zeilenveränderungsprogramm f. PET  
Was ist eine Computersprache  
Messebericht Consumer Show in Las Vegas  
Dreidimensionale Graphik mit dem PET  
Wie lernt man BASIC, Teil VI

**Nr. 5 Mai 1979 4,50 DM**

Adressenverwaltungssystem KARTEI  
JANA - ein neuer Monitor für den PET  
Experimente für Anfänger mit KIM-1  
TANDY TRS-80 Level II  
4k RAM-Erweiterung für KIM-1, SYM-1  
oder AIM  
Z-80 System auf Europakarten  
Microcomputer für den Geschäftsbereich  
Zufallsgraphik  
Wie lernt man BASIC, Teil VII

**Nr. 6 Juni 1979 4,50**

ELCOMP-Leserumfrage  
Z-80-KIT Tips  
Wie lernt man BASIC, Teil VIII  
Mindiskette für TRS-80  
Preiswerter Drucker für den Microcom-  
puteranwender  
Computer-System 79  
Schrittmotoransteuerung mit SYM-1  
Wie baue und programmiere ich einen  
Joystick?  
Assembler für SC/MP  
Starten mit SYM-1  
Displayanzeige mit dem SYM-1

**Nr. 7/8 Juli/August 1979 9,00 DM**

Die neue TRS-80 Überraschung  
6502 hat Zukunft  
Grenzüberwachung  
NASCOM-1 mit TINY BASIC  
BASIC-Vergleichstabelle  
Der SORCERER Computer  
Interview mit dem geistigen Vater der  
TRS-80 Familie  
Der neue Heathkit-Computer  
Microchat  
Wie lernt man BASIC, Teil IX  
Die Hardwarestruktur des ELZET 80  
Was ist ein Wortverarbeitungssystem?  
Computersystem CONDOR  
Computerspiele in BASIC  
PILOT

Ein einfaches PROM-Programmiergerät  
Ein-/Ausgabeprogrammierung mit PET  
Erweiterung für NASCOM-1  
Lernen für die Anwendung von Micro-  
prozessoren

**Nr. 9 September 4,50 DM**

Tips für den BASIC-Programmierer  
Wie lernt man BASIC, Teil X  
EUROCOM-1  
Microchatter  
Die neuen ROMs im CBM  
Neue Publikationen  
Speichererweiterung für den PET  
Musikprogramm für den NASCOM-1  
Computersystem 79  
Umschalten des Bildschirms auf Breit-  
schrift beim TRS-80 Level I  
Vergleich dreier Zahlen  
ELCOM-Leserbriefe  
Diskettenprogramm APPLE II  
Neuer Einplatinencomputer von Motorola  
Einstellung der Tonspur beim PET

**Nr. 10 Oktober 1979 4,50 DM**

Ein-/Ausgabeprogrammierung mit dem  
TRS-80  
Texas Instruments stellt seinen Heim-  
computer vor  
Microchatter  
Wie lernt man BASIC, Teil XI  
Der MZ-80 von Sharp  
Microcomputer als Hobby  
Ein komplettes Textverarbeitungssystem  
mit CBM-Computer u. Heathkit-Drucker  
Wer ist OHIO SCIENTIFIC?  
KURDIS - Ein Programm zur Kurven-  
diskussion  
BASIC im Unterricht  
Monolithischer 8 Bit Analog/Digital-  
Wandler  
Band-Test

**Nr. 11/12 Nov./Dez. 1979 9,00 DM**

Jetzt kommt PASCAL  
Aus der PET-Modulkiste  
HISTORGRAMM  
32k RAM Zusatzspeicher f. PET 2001  
Exidy Sorcerer  
Kleinschrift und Verbesserung des Video-  
RAMs beim TRS-80  
Filmkamerasteuerung mit PET  
Anschluß eines Floppy-Disk-Controllers  
an 6500 Microprozessoren  
Anschluß eines Baudot-Fernschreibers an  
einen Microcomputer  
Ein Besuch bei ATARI  
Kalender  
Ein wenig Trigonometrie  
Berechnen der Kreiszahl  
Der NASCOM-2 ist da  
PET als IC-Tester  
Das NIM-Spiel und was es damit auf sich  
hat  
Linking Loader für PET 2001 (8k)  
MEMDMP-Hex Dump- Programm für  
6502 Systeme

KIM-1 schreibt auf Skop  
IRRGARTEN  
KIM-Orgel  
Lichtgriffel  
Melde-ton für PET  
Preiswerter Drucker f. NASCOM-1  
Wie lernt man BASIC, Teil XII

**Bereits vergiffen I**

**Nr. 1 Januar 1980 4,50 DM**  
Microcomputer-Software und Vernunft  
TRS-80 als "elektronisches Kurvenlineal"  
Schrottknäppel  
Programmbeschr.: FLAG OPERATION  
Aus der PET-Modul-Kiste  
Alternatives TINY-BASIC für den TRS-80  
unter Level 1  
Programmtricks APPLE II  
NEU!!! Einführungskurs: Programmieren  
in Maschinensprache auf Z-80  
Programmiertrick für TRS-80  
Eine professionelle Analogschnittstelle  
GRAPHYSYS  
Wie lernt man BASIC, Teil XIII  
Programmierhilfen f. NASCOM-1  
SUMA 85 (Selbstbausystem)  
Automatische Zeilennummerierung für  
PET  
ULTRA-MON-CBM Bedienungsanleitung  
Entwicklungsboard f. AmZ8000  
ELTRO-HOBBY 79, der große Renner

**Nr. 2 Februar 1980 4,50 DM**

"Echte" Großschrift beim TRS-80  
BASIC-Programme - kürzer und schneller  
Datenlogger BASIC - Programmierbar  
EPROM-Programmiergerät für PET  
HIGH-Resolution-Graphic auf dem  
APPLE II  
Wie lernt man BASIC, Teil XIV  
Der TI-Programmer  
SUMA 85  
Programmieren in Maschinensprache mit  
Z-80, Teil 3  
PET-Petits

**Nr. 3 März 1980 4,50 DM**

Der Personalcomputer von HP ist da  
Lernen mit dem AIM, Teil 1  
Die neuen CBM-Modelle  
SUMA 85, Teil 3  
Programmieren in Maschinensprache mit  
Z-80, Teil 3  
Maschinencodemonitor in TINY-BASIC  
PET-Petits  
Was ist Datenschutz?  
Monitorprogramm für SC/MP  
Wie lernt man BASIC, Teil XV  
Quadratische Gleichungen  
Wer ist Jim Butterfield?  
Messebericht: Consumer Electronics  
Show  
Der PC-100

**Nr. 4 April 1980 4,50 DM**

Grundlagen der Finanzbuchhaltung für  
den Microcomputeranwender

SUMA 85, Teil 4  
Messebericht: Hobbytronic Dortmund  
Berechnung ganzzahliger Funktionen  
Aus der PET-Modulkiste  
SDS-100 MicroMini  
Erstes Home-Computer-System von TI  
Programmieren in Maschinensprache mit  
Z-80, Teil 4  
PRINT USING für X80 und PET  
Microchatter  
Wie lernt man BASIC, Teil XVI  
PET Petits  
Grafik mit dem Superboard  
Untersuchen von Texten auf die Häufig-  
keit des Auftretens einzelner Buch-  
staben  
Zum lernen ideal. . .  
Hochinteressante Interviews

**Nr. 5 Mai 1980 4,50 DM**

Aus der PET-Modul-Kiste  
Z80-Disassembler für den NASCOM  
Implementierung einer REAL-TIME-  
CLOCK  
SUMA 85, Teil 5  
Programmieren in Maschinensprache mit  
Z-80, Teil 5  
Lernen mit AIM, Teil II  
AIM 65 hilft im Büro  
Protokollieren von TRS-80 Bildschirm-  
ausgaben  
Wie lernt man BASIC, Teil XVII  
West Coast Computer Faire  
Der HY Q-1000  
Der BASE2 Drucker

**Nr. 6 Juni 1980 4,50 DM**

Die einfache Verkettungstechnik  
Programmieren in Maschinensprache mit  
Z-80, Teil 6  
Lernen mit dem AIM, Teil 3  
Microcomputer Fools Corner  
Challenger C4PMF Microcomputer  
SUMA 85, Teil 6  
Wie lernt man BASIC, Teil XVIII  
TINY-FORTH für TRS-80 Level II, 16k  
CBM-Disk BASIC Version 4.0  
Heathkit - Matrixdrucker / Erfahrungs-  
bericht  
Monostabile Kippstufen  
5. West Coast Computer Faire  
Hex-DUMP in BASIC für Superboard  
Microchatter

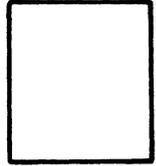
**ELCOMP ist in der Micro-  
computer-Berichterstattung  
immer vorne.**

**Keiner, der mitreden will,  
kann es sich noch leisten,  
ELCOMP nicht zu lesen!  
Bestellen Sie deshalb heute  
alle noch verfügbaren Hefte  
und sichern Sie sich ELCOMP  
im Abonnement für die Zu-  
kunft.**



# ELCOMP

POSTKARTE



Absender  
Bitte deutlich ausfüllen

\_\_\_\_\_  
Vorname/Name

\_\_\_\_\_  
Beruf

\_\_\_\_\_  
Straße/Nr.

\_\_\_\_\_  
Plz Ort

# ELCOMP

MIKROCOMPUTER BOOK STORE

Tegernseerstr. 18

D-8150 Holzkirchen /Obb.

ABSENDER:

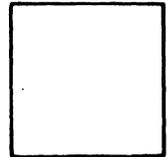
.....  
Name, Vorname

.....  
Straße

( ) .....

PLZ Ort

.....  
Telefon



# ELCOMP

Ing. W. Hofacker GmbH  
Tegernseer Straße 18

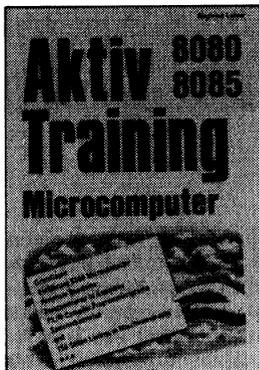
D-8150 Holzkirchen

Das Abonnement wird automatisch verlängert  
(Kündigung 8 Wochen z. Abonnement-Ablauf)

# Mikrocomputer Fachbücher



deutsch

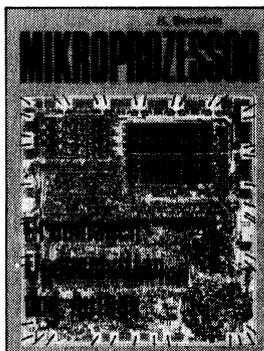


## ◀ Aktiv Training Microcomputer 8080 8085

Dieses Werk mit über 360 Seiten beschäftigt sich ausschließlich mit den Microcomputerbausteinen und Peripherieelementen der 8080A und 8085 Microprozessoren. Grundlagen, Einführung 8080 Microcomputer, Programmieren in Assembler, PL/M Compiler, Speicherbausteine u. v. a. mehr. Ideal für jeden, der ein System mit 8080, 8085 CPU besitzt. Auch der Z-80 Systembesitzer kann von diesem Buch viel profitieren. Am Schluß des Buches finden Sie noch ca. 150 Programmlistings in Maschinensprache (Nützliche Utilities und Spielprogramme).

Best.-Nr. 30

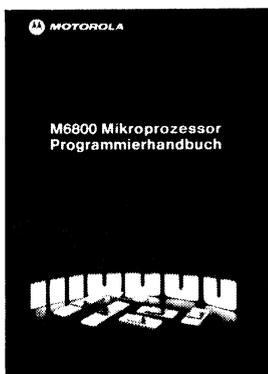
DM 49,80



◀ **Mikroprozessoren, Eigenschaften und Aufbau, Teil 1**, H. Bernstein  
Grundlagen, Eigenschaften und Aufbau von Mikroprozessoren, Organisation von Recheneinheiten und Mikroprogrammen. Programmierung und Klassifizierung von Mikroprozessoren. Ablaufdiagramm, Flußdiagramm. Ein-Chip-Technik und Multi-Chip-Technik, Transfer- und Sprungfunktionen. Speichertechnik: RAMs, ROMs, FIFO, FILO. Programmierbare logische Arrays (PLA). Anwendungsbeispiele und Anwendungsbereiche. Über 120 Seiten.

Best.-Nr. 22

DM 19,80



## ◀ M6800 Mikroprozessor Programmierhandbuch

Ein deutschsprachiges Handbuch für den 6800 Microcomputer. Grundlagen, Einführung und genaue Befehlsklärungen.

Best.-Nr. 8063

DM 19,80





### Programmieren in Maschinensprache mit 6502

Ein deutsches Buch für jeden, der mit dem 6502 Mikroprozessor arbeitet. Grundlagen Maschinensprache, Beschreibung des Extended Monitors für Superboard von Challenger. Ausführliche Beschreibung jedes einzelnen Maschinenbefehls mit Beispiel. Übersichtliche Beschriftung. Sie finden jeden Befehl schnellstens und können nachlesen, was im Computer selbst geschieht. Am Schluß finden Sie Listings und Beschreibung für einen kompletten 2-Pass-Assembler, Editor, Binder und Disassembler. Für PET 8K, alte ROMs und für die neue CBM-Serie.

Best.-Nr. 118

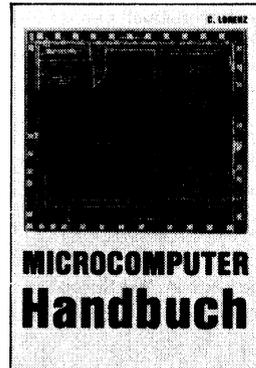
DM 98,-

### Handbuch für SC/MP

Ein echtes Handbuch für SC/MP (INS8060) Microcomputerbesitzer und solche, die es werden sollen. Komplette Einführung mit vielen Schaltbeispielen, Schaltbildern und Programmlistings in Maschinensprache und TINY BASIC. Komplettes ELBUG-Listing, CPU-Karte, RAM-Karte, ROM-Karte, ROM-Programmierer, Cassetten-Interface, Fernsehinterface, Netzteil, Hex Ein-/Ausgabe, SC/MP Einkartenmicrocomputer u. v. a. mehr.

Best.-Nr. 108

DM 29,80

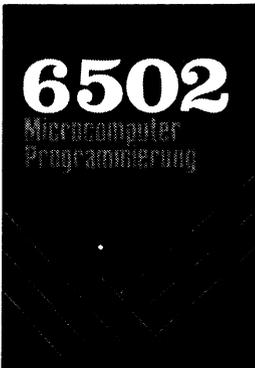


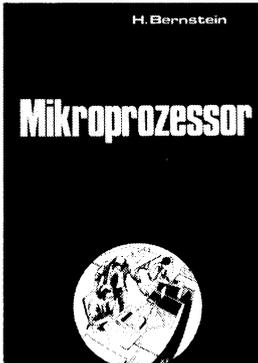
### 6502 Microcomputer-Programmierung, Peter Heuer

Eine deutschsprachige Einführung in die Maschinensprachenprogrammierung anhand des 6502 Microcomputers. Ein echtes Anleitungsbuch zum Einstieg in die Microcomputertechnik mit Hilfe des KIM-1. Viele Programmierbeispiele, die von einem Pädagogen speziell für Anfänger entwickelt wurden. Auch PET, AIM, SYM und ATARI-Besitzer brauchen dieses Buch.

Best.-Nr. 109

DM 29,80





### ◀ Mikroprozessor, Teil 2

von H. Bernstein. Die Fortsetzung unseres ersten, so erfolgreichen Buches über Mikroprozessoren. Technologie von Mikroprozessor- und Speicherbausteinen. Festwertspeicher, PROM, REPROM, FIFO, Schieberegister, MPR-Register, ARL-Register, SAR-Register. Aufbau eines Mikroprozessorsystems mit 8080, RAM- und ROM-Schnittstellen. Befehlssatz 8080. Über 120 Seiten. Best.-Nr. 26 **DM 19,80**

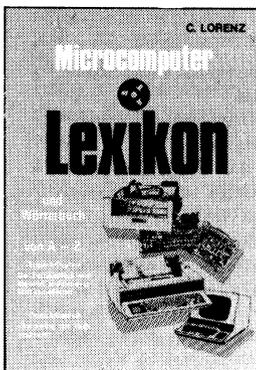
### ▶ Programmierhandbuch für PET

C. Lorenz

Ein vom Hofacker-Verlag für den PET produziertes Buch. Es beginnt da, wo Ihr mitgeliefertes Handbuch aufhört. Einführung Maschinenprogrammierung, Assembler, Ein-/Ausgabeprogrammierung, Programmiertricks, Analog/Digital-Wandler, Graphik, Spracherkennung u. v. a. mehr. Viele Listings, die Sie selbst eintippen können. 324 Seiten.

Best.-Nr. 110

**DM 29,80**



### ◀ Microcomputer Lexikon und Wörterbuch von A-Z, C. Lorenz

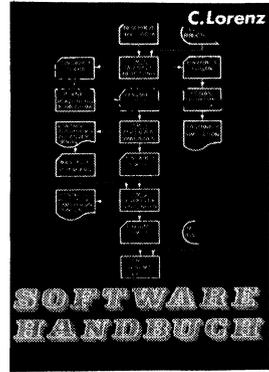
Englisch/Deutsch — Der Fachausdruck wird übersetzt, ausführlich erklärt und erläutert.

Deutsch/Englisch — Übersetzung des Fachausdrucks. Ein Hilfs- und Arbeitsbuch für jeden, der sich heute mit der modernsten Elektronik beschäftigt. Viele engl. Ausdrücke werden heute in der Elektronik, Computer- und Mikroprozessortechnik verwendet und oft fehlt uns eine genaueste und präzise Erläuterung. Ein Lexikon und Wörterbuch in einem einzigen Buch vereint.

Best.-Nr. 28

**DM 29,80**

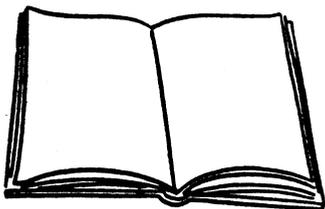
**Mikroprozessor Software Handbuch**  
**C. Lorenz**  
 Grundlagen und Einführung in die  
 Mikrocomputerprogrammierung,  
 Grundlagen und Einführung in die  
 wichtigsten Programmiersprachen  
 (BASIC, FORTRAN, ASSEMBLER-  
 Sprachen). Zusammenstellung der  
 wichtigsten Befehlslisten: 8080, Z80,  
 M6800, National, Fairchild etc. Ein  
 Software Handbuch für jeden, der  
 mit Mikroprozessoren oder Mikro-  
 Computern zu tun hat. Über 200 S.  
**Best.-Nr. 27** **DM 29,80**



Best. Nr.	Stck.	BUCHTITEL BEZEICHNUNG	a DM	ges. DM
1		TBB - Handbuch Band 1	19.80	
2		TBB - Handbuch Band 2	19.80	
3		Elektronik im Auto	9.80	
4		IC - Handbuch, TTL, CMOS, Lin.	19.80	
5		IC - Datenbuch, TTL, CMOS, Lin.	9.80	
6		IC - Schaltungen, TTL, CMOS	9.80	
7		Elektronik Schaltungen	5.-	
8		IC - Bauleitungsbandbuch	19.80	
9		Feldeffekttransistoren	5.-	
10		Elektronik und Radio 4. Auflage	19.80	
11		IC - NF - Verstärker	9.80	
12		Beispiele Integrierter Schaltungen	19.80	
13		HEH, Hobby Elektronik Handbuch	9.80	
14		IC - Vergleichsliste	9.80	
15		Optoelektronik Handbuch	19.80	
16		CMOS Teil 1, Einführung	19.80	
17		CMOS Teil 2, Schaltbeispiele	19.80	
18		CMOS Teil 3, Schaltbeispiele	19.80	
19		IC - Experimentier Handbuch	19.80	
20		Operationsverstärker	19.80	
21		Digitaltechnik Grundkurs	19.80	
22		Mikroprozessoren 2. Auflage	19.80	
23		Elektronik Grundkurs	9.80	
24		Mikrocomputer Technik	29.80	
25		Hobby Computer Handbuch	29.80	
26		Mikroprozessor Teil 2	19.80	
27		Mikroprozessor Software Handbuch	29.80	
28		Lexikon + Wörterbuch	29.80	
29		Mikrocomputer Datenbuch	49.80	
30		Aktivtraining Microcomputer	49.80	
31		57 Programme in BASIC	39.-	
32		Circuits Digital ET Pratique	19.80	
33		Microcomputer Programmierbeispiele	19.80	
41		IC-KIT Experimentierplatte WH-1 g	79.-	
106/1		TTL-Experimentierbuch	5.-	
106/1		CMOS-Experimentierbuch	5.-	

# Mikrocomputer Fachbücher

englisch



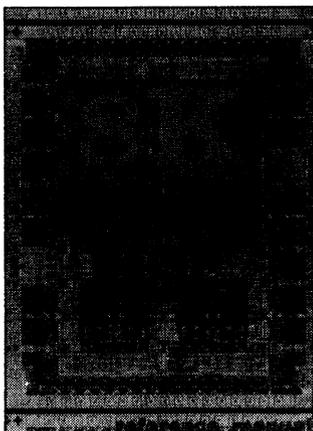
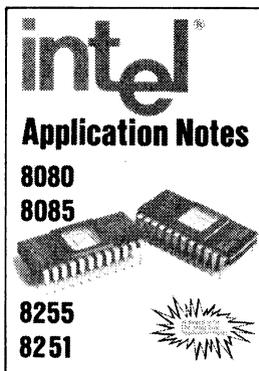
**Intel Application Notes (8080, 8085, 8255, 8251)**

Dieses Buch braucht jeder, der mit 8080, 8085 oder Z-80 Mikroprozessoren arbeitet.

Wir haben die interessantesten Applikationsberichte in diesem Buch zusammengefaßt.

Aus dem Inhalt: Designing with Intel's Static RAMs 2102, Memory Design with the Intel 2107B.

8255 Programmable Peripheral Interface Applications, Using the 8202 Dynamic RAM Controller u. v. a.  
Best.-Nr. 153 DM 29,80



**Care and Feeding of the Commodore PET**

Das ideale Buch für den Hardware-Bastler. Viele Tricks, Schaltbilder, Hinweise und Erläuterungen für den, der gerne selbst Erweiterungen bauen möchte. Memory Map für 8K PET und CBM, Bauanleitung für eine serielle Schnittstelle u. v. a. mehr.

Best.-Nr. 150 DM 19,80

**Microsoft 8K BASIC Reference Manual**

Eine sehr gute BASIC-Einführung. Auch als Handbuch zum Nachschlagen bestens geeignet. Ideal für jeden PET, CBM, TRS-80, KIM-BASIC, SYM-BASIC, AIM- und APPLE-Besitzer. 73 Seiten DIN A4 mit vielen Beispielen. Eine Produktion des Hofacker-Verlages in englischer Sprache.

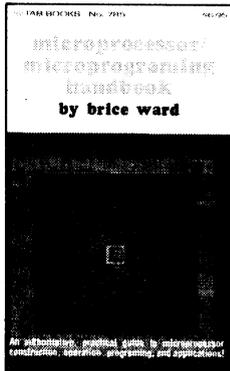
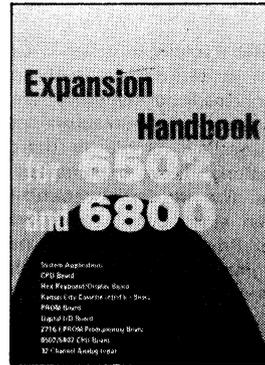
Best.-Nr. 151 DM 19,80



### Expansion Handbook for 6502 and 6800

Das ideale Handbuch für alle KIM, SYM, AIM, PET und Challenger Computer-Freunde. Das Buch beschäftigt sich ausschließlich mit dem S-44-Bus. Dies ist exakt der Bus von SYM, AIM und KIM. Sehr viele Schaltbilder: CPU-Platine, Hex-Tastatur Eingabe, Kansas City Interface, RAM u. ROM-Karte, Analog-Eingabe Board u. v. a. Das Buch ist für jeden 6502 Systembesitzer unentbehrlich. Ca. 150 Seiten

Best.-Nr. 152 DM 19,80



### Microprocessor/Microprogramming Handbook, Brice Ward

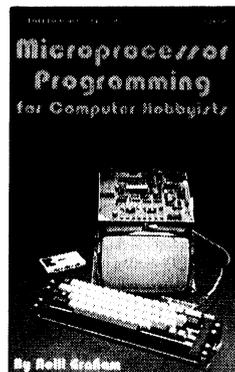
Ein praktisches Handbuch für jeden, der sich mit Mikroprozessoren beschäftigen möchte. Auf über 290 Seiten finden Sie Grundlagen, alles über Programmierung und Anwendungsbeispiele der interessantesten integrierten Schaltungen. Inhalt: Einführung in die Mikroprozessortechnik. Der Mikroprozessor von innen. MCS4, MCS40, MCS80, Speichersysteme, Mikroprogrammierung in Maschinen- und Assemblersprache.

Best.-Nr. 785 DM 35..

### Microprocessor Programming for the Computer Hobbyist

Dieses buch ist speziell für den Computer-Hobbyisten geschrieben, der sich bereits mit weiterführenden Programmieretechniken und Datenstrukturen beschäftigen möchte. Inhalt: Höhere Programmiersprachen (PL/M, PL/1), arithmetische Funktionen, Suchprogramme werden im Zusammenhang mit dem Schachspielproblem behandelt. Über 380 Seiten in englischer Sprache

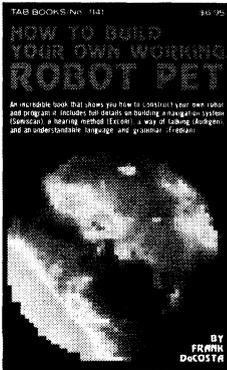
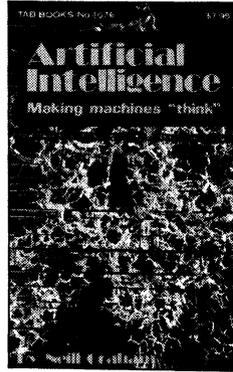
Best.-Nr. 952 DM 39,-



### Artificial Intelligence, Neil Graham

Hier ist endlich ein Buch speziell über künstliche Intelligenz. Der Autor benutzt Computerspiele und Robotertechnik für die illustrative Behandlung dieses brandheißen Themas. Was ist der Unterschied zwischen einer Lösung einer komplexen Programmieraufgabe und der Programmerstellung für intelligente Computerentscheidungen.

Best.-Nr. 1076 DM 29,80



### How to build your own working ROBOT PET

An incredible book that shows you how to construct your own robot and program it. Includes full details on building a navigation system (Soniscan), a hearing method (Excom), a way of talking (Audigen) and an understandable language and grammar (Fredian).

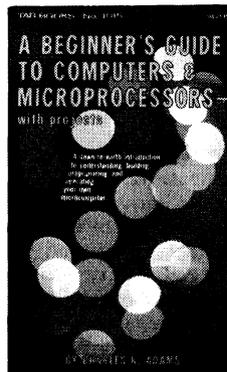
Eine wertvolle Hilfe für alle, die sich einen Roboter bauen wollen. (8085) CPU. Viele wertvolle Schaltungen, die Sie im Zusammenhang mit TRS-80 oder 8085-Computern verwenden können. 238 Seiten.

Best.-Nr. 1141 DM 29,80

### Beginner's Guide to Computer Programming, Brice Ward

Eine ideale Einführung in die Programmierung von Computern (Microprozessoren). Das Buch beginnt mit der Entwicklung einer einfachen Programmiersprache für den eigenen Bereich und zum Selbststudium, sodann wird auf andere Sprachen übergegangen. Inhalt: Grundlegende Programmkonzepte, I/O-Schaltungen, Flußdiagramme, Programmtest, Schleifen, Indexregister, versch. Programmiersprachen, Compiler, Cobol u. v. a., 480 Seiten, 364 Bilder, in englischer Sprache.

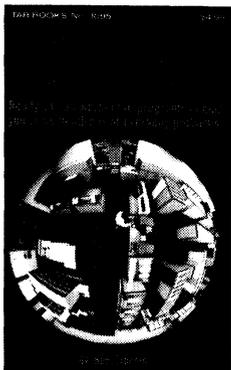
Best.-Nr. 1015 DM 29,80



## PROGRAMMING MICROPROCESSORS, M.W.McMurrin

Dieses Buch reicht von der grundlegenden Microprozessororganisation über Zahlensystem, Flußdiagramme, Adressierung, Assemblierung über Subroutinenbeschreibungen, Programmierhilfen, Datenaustausch, Compilern bis hin zu speziellen Programmiertechniken. **Schwergewicht Software. 270 Seiten, in englischer Sprache**  
Best.-Nr. 985

DM 35.--



◀ **25 Games in BASIC (Listings).** Die Programme laufen mit kleinen Änderungen auf den meisten Microcomputern. Inhalt: Number, Letter, Stars, Trap, Bagels, Mugwump, Hurkle, Snark, Reverse, Button Chomp, Taxman u. v. a.

Best.-Nr. 8057

DM 9,80

### Digital Interfacing with an Analog World, Joseph J. Carr ▶

Wie entwirft und baut man Interfaceschaltungen für Microcomputer? Der Schwerpunkt liegt bei der Verbindung mit der analogen Welt. Viele Tabellen, Schaltungen, Berechnungen und Hinweise. Über 400 Seiten.

Best.-Nr. 1070

DM 39.--



### Complex Sound Generation with SN 76477

Ein Applikationsheft für einen der interessantesten integrierten Bausteine unserer Zeit. Ein LSI-Baustein zur Tonerzeugung. Je nach äußerer Beschaltung können Sie mit diesem Baustein die verrücktesten Töne erzeugen.

Dampfisenbahngeräusch mit Dampfpeife, Vogelgezwitscher, Hundegebell, elektronische Orgel, Schuß mit Explosion u. v. a. mehr.

Best.-Nr. 154

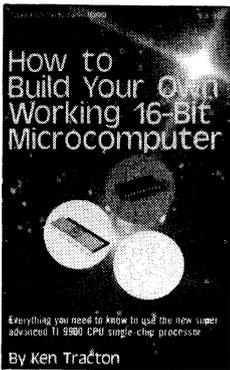
DM 19,80

### DR.DOBBS' JOURNAL OF COMPUTER CALISTHENICS AND ORTHODONTIA, VOLUME ONE

Die komplett gebundene Sammlung der ersten sehr interessanten Ausgaben von Dr.Dobb's Journal. Inhalt: Entwickeln Sie Ihre eigene BASIC-Sprache, Entwicklungshilfen für Tiny Basic, Wie programmiere ich den ALTAIR 8800 für Computer-Musikstücke, kompl.Listing des Denver Tiny-BASIC ' Computer für Musik und Komposition, Hardware und Software für Sprachsynthese, MINOL - ein Tiny Basic mit String, in 1,75 k Byte, System Monitor für 8080, Palo Alto Tiny Basic, kompl.Listing, 8080 Text Editor, Tiny Trek-Game, Fließkommaroutinen für 6502, Comp.Music, Z 80 KIT, 6800 Assembler, u.v.a., ca. 260 Seiten A 4

Best.-Nr. 80/20

DM 59.--



◀ **How to Build your Own Working 16-Bit-Microcomputer** von Ken Tracton

Alles, was Sie über den neuen Supermicroprozessor T19900 wissen müssen. Beschreibung des Schaltkreises, Peripherie-Bausteine für TMS9900. Tricks, Systemaufbau etc. Sehr interessant, da der TMS9900 das Herz des neuen Texas Instruments Personal Computers ist.

**Best.-Nr. 1099**

**DM 14,80**

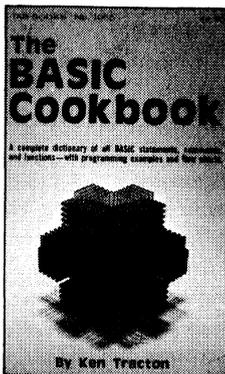
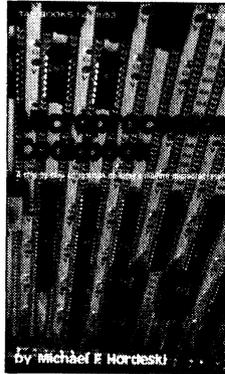
**Microprocessor Cookbook**, von ▶ Michael Hordeski

A chip-by-chip-comparison of todays modern microprocessors.

Die wichtigsten Prozessortypen werden genau beschrieben (Schaltung, Blockdiagramm, Befehlslisten, Programmbeispiele) und verglichen. 8080, 6800, F8, Z80, TMS 9900, SC/MP, Bit Slices, R6500, 264 S.

**Best.-Nr. 1053**

**DM 24,80**



◀ **The BASIC Cookbook**, von Ken Tracton

Ein komplettes Dictionary mit BASIC-Befehlen. Alle Befehle sind in alphabetischer Reihenfolge geordnet und jeder Befehl genau beschrieben. Zu jedem Befehl ist ein Demonstrationsbeispiel beigefügt.

Dieses Buch braucht jeder BASIC-Programmierer!

**Best.-Nr. 1055**

**DM 24,80**

# BASIC

Bücher für Microcomputer

deutsch



◀ **BASIC Programmierhandbuch** von C. Lorenz

Einführung und Nachschlagewerk. Speziell für die BASIC-Versionen der modernen Microcomputersysteme. Jeder Befehl wird ausführlich beschrieben und ein Beispielprogramm gezeigt. Sehr übersichtlich und praktisch. Am Schluß finden Sie ein komplettes BASIC-Programm, das Ihnen über einen Computer BASIC lehrt.

Best.-Nr. 113

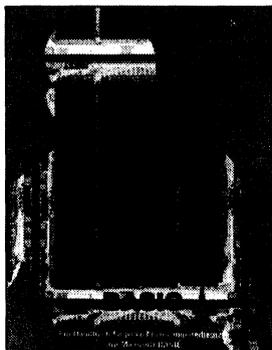
DM 19,80

**Microsoft BASIC Handbuch**

Ein Buch für alle, die das moderne Microsoft BASIC (BASIC80 und Standalone BASIC) kennenlernen möchten. Ideal als Einstieg und Nachschlagewerk. Mit vielen Beispielen (Programmlistings) und Hinweisen. Ideal auch für die Schule, Ausbildung und innerbetriebliche Schulung. Ca. 160 Seiten.

Best.-Nr. 121

DM 29,80

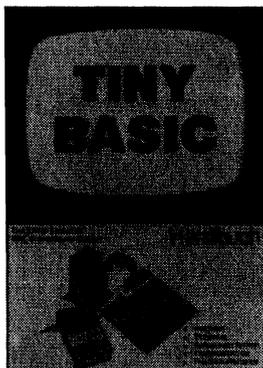


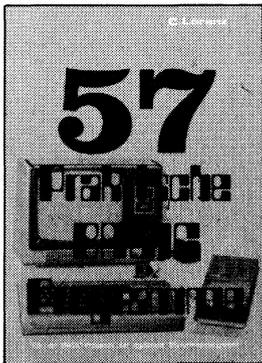
◀ **TINY BASIC Handbuch**, H. Hermann

Das erste deutschsprachige Handbuch über Tom Pittman's TINY BASIC. Eine Einführung in die TINY BASIC-Programmiersprache. Wie kann ich meinen Computer (KIM-1) erweitern und BASIC programmieren. Systemvorschläge. Viele Programmierbeispiele, Tricks und Kniffe.

Best.-Nr. 34

DM 19,80





## 57 praktische BASIC-Programme

C. Lorenz, Ken Tracton

Ein Buch mit technisch-wissenschaftlichen Programmen und einer großen Anzahl von Spielprogrammen in BASIC. (Games). Ein Buch für jeden, der sich mit dem faszinierenden Hobby der Microcomputertechnik befassen will. Alle Listings sind in BASIC und können auf den meisten Personal Computer Systemen gefahren werden. Alle Programme wurden sorgfältig getestet. Zum Beweis ist für jedes Programm ein Protokoll des Probelaufs abgedruckt.  
**Best.-Nr. 31** **DM 39,-**

## The A to Z Book of Computer Games

26 aufregende und instruktive Spielprogramme fix und fertig zum Eingeben in jeden BASIC-Computer. Nach einer kleinen Einführung in die Programmierung folgen interessante Spielprogramme, die jeder Microcomputerbesitzer gesehen haben muß.

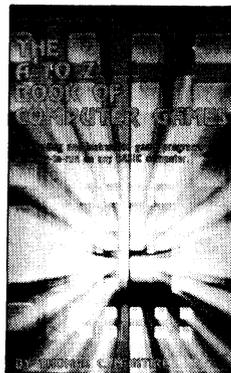
(Bandit, Cokes, Dice, Elevate, Firecard, Gunners, Hobbit, Invert, Justluck, Knights, Lapides, Match, Naughts + Crosses, Otello, Par 2, Quantal, Roulette, States Twenty 1, Ultranim, Verboten, Wumpus, X-Change, YATC, END.

Am Schluß werden noch einige Tips zur Umwandlung in andere BASIC-Versionen gegeben.

Über 300 Seiten.

**Best.-Nr. 1062**

**DM 29,80**



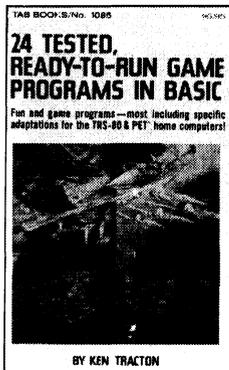
## englisch

### 24 Tested Ready to RUN Game Programs in BASIC, Ken Tracton

Spaß- und Spielprogramme in BASIC. Viele Programme enthalten spezielle Anpassungshinweise an die Homecomputertypen TRS-80 und PET. Viele Graphik- und Zeichenprogramme. Ideal für jeden BASIC-Computer-Besitzer.

**Best.-Nr. 1085**

**DM 24,80**



# Bücher für Microcomputer

# TRS-80



## ◀ Programmieren mit TRS-80 von Martin Stübs

Das erste in einem deutschen Verlag produzierte Buch über den erfolgreichen Personal Computer von TANDY. Ein Buch für jeden, der einen TRS-80 bereits besitzt oder vor der Entscheidung steht, welchen Computer er sich anschaffen soll. Einführung, Programmiertricks, Erweiterungen, Maschinenprogrammierung und viele Programme (Listing mit Beschreibung)

Best.-Nr. 111

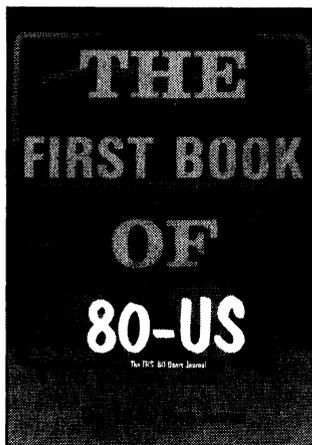
DM 29,80

## The First Book of 80-US

Für den TRS-80 Freund eine echte Preissensation. Die ersten fünf Hefte aus 80-US Journals in einem Sammelband zusammengefaßt. Voll mit vielen sehr interessanten Hard- und Softwareideen, Tricks. Viele komplette Programmbeispiele (Listings) in BASIC und Z-80 Maschinensprache. Über 250 Seiten DIN A4. Farbiger Umschlag. Dieses Buch sollte jeder TRS-80-Besitzer oder der es werden will, im Schrank haben.

Best.-Nr. 155

DM 19,80



# BÜCHER



- 25 Games in BASIC (Listing)** Die Programme laufen auf dem PET. Inhalt: Number, Letter, Stars, Trap, Bagels, Mugwump, Hurkle, Snark, Reverse, Button, Chomp, Taxman u. v. a.  
Best.-Nr. 80/57 9,80
- 57 Practical Programs and Games in BASIC**, ca. 200 Seiten Listings.  
Best.-Nr. 1000 DM 35,-
- BASIC Software Library Vol. I**, Personal Bookkeeping, Games and Pictures, 49 BASIC-Programme.  
Best.-Nr. 80/50 DM 99,-
- BASIC Software Library Vol. IV**, General Purpose und viele Spiele, 21 BASIC Programme.  
Best.-Nr. 80/53 DM 39 --
- Dr. DOBBs COMPUTERZEITSCHRIFT**. Die komplett gebundene Sammlung der ersten, sehr interessanten Ausgaben. Sehr viele Programmlistings für den Hobbycomputerfan.  
Best.-Nr. 80/20, ca. 260 Seiten DIN A 4 DM 59,-
- Microprocessor Software Handbuch**, C. Lorenz  
Microcomputerprogrammierung für Anfänger und Fortgeschrittene. BASIC-Einführung in Deutsch.  
Best.-Nr. 27, über 280 Seiten DM 29,80
- PET 6502 Programmierhandbuch**, das Handbuch für jeden PET-Besitzer und solche, die es werden wollen. Einführung, Programmbeispiele, PET-Erweiterungsmöglichkeiten.  
Best.-Nr. 110 DM 29,80
- The BASIC-Cookbook**, Ken Tracton  
Über 138 Seiten BASIC-Einführung mit vielen praktischen Programmierbeispielen. Alphabetisch geordnet, ideal auch als Lexikon.(Diagnose, Erhöhung der Rechengeschwindigkeit u. v. a.)  
Best.-Nr. 1055 DM 24,80
- A Beginner's Guide to Computers & Microprocessors – with projects**  
Eine Einführung von Anfang an in die Mikroprozessortechnik. Bauen Sie Ihren eigenen Computer, programmieren Sie ihn und bedienen Sie ihn. Viele praktische Schaltungen. Über 300 Seiten.  
Best.-Nr. 1015 DM 29,80
- 650X Software Manual (Programming Manual)**, sämtliche wichtige Daten und Programmierhinweise für die 6502 Programmierung. Ca. 200 Seiten.  
Best.-Nr. 80/42 DM 19,80
- 650X Hardware Manual**, 6502 Systemorganisation, Grundlagen, Applikationshiften u. v. a. mehr.  
Best.-Nr. 80/43 DM 19,80
- 6500 Seminarheft (Briefing Notebook)**, eine kleine Einführung in die 6500 Systemfamilie.  
Best.-Nr. 80/45 DM 9,80
- 6500 Datenblätter**, eine Sammlung aller wichtigen Datenblätter aus der 6500-Familie.  
Best.-Nr. 80/46 DM 9,80
- 6502 Microcomputer, Aufbau und Programmierung**. Ein deutsches Anleitungsbuch zum Einstieg in die Microcomputertechnik mit Hilfe des KIM-1. Viele Programmierbeispiele von einem Pädagogen speziell für den Anfänger entwickelt. (PET, VIM und AIM)  
Best.-Nr. 109 DM 29,80

# NOTIZEN

**NOTIZEN**

# NOTIZEN

# NOTIZEN

Auch Sie brauchen ELCOMP!



Jahresabonnement DM 59,-  
incl. MwSt. und Versand.  
Zurückliegende Hefte zu  
Originalpreisen noch verfü-  
gbar.

# ELCOMP



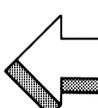
# ELCOMP

FACHZEITSCHRIFT FÜR MICROCOMPUTER

---

1. Jahrgang    Volume 1    September 1978    DM 3,50    SFR 4.-    OS 30.-    Nr. 1

### AUS DEM INHALT:



01	Symbole
03	Microcomputer Grundkurs
10	ELCOMP Software Shop
11	North-Star Horizon Computer
12	Software für Z 80 Systeme
21	Selbstbaucomputer mit Z 80
	Bauanleitung für ein eigenes Microcomputersystem, Teil 1
36	8085 Microcomputer-KIT
37	Microcomputer-Anwendung im Klein- betrieb
40	S-100 Bus Adapter für PET 2001
41	Exidy - Ein neues Microcomputer- System mit Z 80
42	Hauskauf mit BASIC
44	Heimcomputer-Vergleichsliste
45	BASIC-Programm FACTORS
46	Der S-100 Bus (Beschreibung)

SOFTWARE - NEUHEITEN

51	Viele Programme für den PET
52	Software für 8080/Z 80/6800 Systeme
53	Hardware-Neuheiten
53	AIM Computer von Rockwell
54	Erweitern Sie Ihren KIM-1
56	Termine - Neue Bücher
57	Der VIM-1 ist da

## HOFACKER-VERLAG

Ing. W. Hofacker GmbH  
Tegernseer Straße 18

### D-8150 Holzkirchen/Obb.

Die Fachzeitschrift für MICROCOMPUTER  
Eine unentbehrliche Informationsquelle für alle Elektroniker

Microcomputer-Anwendungsbeispiele  
Künstliche Intelligenz  
Block-Strukturierte Programme  
Datenverarbeitung im Kleinbetrieb  
Club-Neuheiten  
Computer und Kunst  
Musik mit dem Computer  
Monitore für 8080, 6800, 6502, Z 80,  
SC/MP, 2650, 1802  
Eigenbau-Computersysteme  
Interface-Techniken  
Microcomputer KITS

Neue Produkte  
Betriebssysteme für Floppys  
Programmiertechniken  
Software-Quellen  
Programmierbeispiele  
Soziale Aspekte der Microcomputer-  
technik  
Technologische Neuheiten  
Anwendungen in der Meß- und Regel-  
technik  
Anwendungen bei Funk-Amateuren

# Verlagsprogramm

Best.-Nr.	Titel	Preis DM	Best.-Nr.	Titel	Preis DM
<b>Bücher in deutscher Sprache</b>					
1	Transistor Berechnungs- u. Bauanleitungsbuch – 1	19,80	121	Microsoft BASIC-Handbuch	29,80
2	Transistor Berechnungs- u. Bauanleitungsbuch – 2	19,80	122	BASIC für Fortgeschrittene	39,00
3	Elektronik im Auto	9,80	123	IEC-Bus Handbuch	19,80
4	IC-Handbuch, TTL, CMOS, Linear	19,80	124	Programmieren in Maschinsprache mit CBM	19,80
5	IC-Datenbuch, TTL, CMOS, Linear	9,80	127	Einf. in die Microcomp.-Progr. mit 6800	49,00
6	IC-Schaltungen, TTL, CMOS, Linear	9,80	128	Programmieren mit dem CBM	29,80
7	Elektronik Schaltungen	5,00	129	ELCOMP-Leser Programmierhandbuch	69,00
8	IC-Bauanleitungsbuch	19,80	130	Programmierbeispiele für CBM	19,80
9	Feldeffekttransistoren	5,00	131	Cobol für Anfänger	19,80
10	Elektronik und Radio	19,80	132	CP/M-Handbuch	29,80
11	IC-NF Verstärker	9,80	133	Welches Betriebssystem brauche ich?	19,80
12	Beispiele Integrierter Schaltungen (BIS)	19,80	<b>Bücher in englischer Sprache</b>		
13	HEH, Hobby Elektronik Handbuch	9,80	150	Care and Feeding of the Commodore PET	19,80
14	IC-Vergleichsliste	29,80	151	8k Microsoft BASIC Reference Manual	19,80
15	Optoelektronik Handbuch	19,80	152	Expansion Handbook for 6502 and 6800	19,80
16	CMOS Teil 1, Einführung, Entwurf, Schaltbeispiele	19,80	153	Microcomputer Application Notes	29,80
17	CMOS Teil 2, Entwurf und Schaltbeispiele	19,80	154	Complex Sound Generation using the SN76477	19,80
18	CMOS Teil 3, Entwurf und Schaltbeispiele	19,80	155	The First Book of 80-US (TRS-80)	19,80
19	IC-Experimentier Handbuch	19,80	156	Small Business Programs	29,80
20	Operationsverstärker	19,80	157	The First Book of Ohio Scientific	19,80
21	Digitaltechnik Grundkurs	19,80	158	The Second Book of Ohio Scientific	19,80
22	Mikroprozessoren, Eigenschaften und Aufbau	19,80	159	The Third Book of Ohio Scientific	19,80
23	Elektronik Grundkurs, Kurzlehrgang Elektronik	9,80	160	The Fourth Book of Ohio Scientific	29,80
24	Microcomputer Technik	29,80	161	The Fifth Book of Ohio Scientific	19,80
25	Hobby Computer Handbuch	29,80	162	ATARI Games in BASIC	19,80
26	Mikroprozessor, Teil 2	19,80	163	The Peripheral Handbook	29,80
27	Microcomputer Software Handbuch	29,80	255	Programmierbeispiele mit TRS-80 Pocket	19,80
28	Lexikon + Wörterb. f. Elektr. u. Mikroprozessor LEM	29,80	1045	The Programers Guide to LISP	24,80
29	Mikrocomputer Datenbuch	49,80	1050	The Most Popular Subroutines in BASIC	24,80
30	Aktivtraining-Mikrocomputer	49,80	1053	Microprocessor Cookbook	24,80
31	57 Programme in BASIC	39,00	1055	The BASIC Cookbook	24,80
32	ATARI BASIC Handbuch	29,80	1062	The A to Z Book of Computer Games	29,80
33	Microcomputer Programmierbeispiele	19,80	1070	Digital Interfacing with an Analog World	39,00
34	TINY-BASIC Handbuch	19,80	1071	The Complete Handbook of Robotics	29,80
35	Der freundliche Computer	29,80	1076	Artificial Intelligence	29,80
101	CB-Handbuch	19,80	1085	24 Tested Ready to RUN Game Programs in BASIC	24,80
103	Oszillographen-Handbuch	19,80	1088	Illustrated Dictionary of Microcomputer Terminology	35,00
104	1000 Elektronik Schaltungen	49,00	1095	Programs in BASIC for Electronic Engineers	19,80
107	Praktische Antennentechnik	19,80	1099	How to Build Your own Working 16-Bit Microc.	14,80
108	SC/MP Mikrocomputer-Handbuch	29,80	1141	How to Build Your own Working ROBOT PET	29,80
109	6502 Microcomputer Programmierung	29,80	1160	1001 Things to do with Y. P. C.	29,80
110	Programmierhandbuch für PET	29,80	1169	The Giant Book of Computers	39,00
111	Programmieren mit TRS-80	29,80	8029	Z-80 Assemblerhandbuch	29,80
112	PASCAL-Programmier-Handbuch	29,80	8042	6500 Software Manual	19,80
113	BASIC-Programmier-Handbuch	19,80	8043	6500 Hardware Manual	19,80
114	Der Microcomputer im Kleinbetrieb	39,80	8048	BASIC Software Vol. VI	199,00
115	FOCAL Programmier-Handbuch	19,80	8049	BASIC Software Vol. VII	159,00
116	Einführung 16-Bit Microcomputer	29,80	8050	BASIC Software Vol. I	99,00
117	FORTRAN für Heimcomputer	19,80	8051	BASIC Software Vol. II	99,00
118	Programmieren in Maschinsprache mit dem 6502	98,00	8052	BASIC Software Vol. III	149,00
119	Programmieren in Maschinsprache (Z80)	49,00	8053	BASIC Software Vol. IV	39,00
120	Anwenderprogramme für TRS-80 u. Video Genie	29,80	8054	BASIC Software Vol. V	39,00
			8063	M6800 Programmierhandb.	19,80