

Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401, CH-6000 Luzern 15

3. Jahrgang  
Erscheint 6 mal jährlich  
ISSN 0251-2017

Commodore-Anwenderzeitschrift  
Jahresbezugspreis inkl. Versand und Porto  
SFr. 48.-/DM 55.-/öS 400

Lieber Commodore-Freund,

Mit grossem Wirbel rührt Commodore die Reklametrommel für ihre jüngsten Kreationen. Noch mehr gerührt wird allerdings in der Gerüchteküche, denn von den vielgepriesenen neuen Computertypen gibt es zur Zeit anscheinend nur wenige Prototypen oder Holzmodelle. Einziges Licht im Dunkel: der VC 64 soll tatsaechlich noch in diesem Jahr auf den Markt kommen - zumindest in Deutschland. Hier bei uns werden wir erfahrungsgemaess noch ein wenig laenger warten müssen. Eines steht aber schon heute fest: sobald die neuen Typen erhaeltlich sind, werden auch diese Geraete zu einem festen Bestandteil unserer Zeitschrift.

Rückblickend auf das zu Ende gehende 1982 sind wir der Meinung, dass wir den eingeschlagenen redaktionellen Kurs für die CBM/PET NEWS beibehalten und sogar da und dort ein wenig verbessert haben. Natürlich möchten wir auch Ihre Meinung vermehrt kennen lernen und sind Ihnen für Ihre Zuschriften dankbar. Für die CBM/PET NEWS beginnt nun schon das vierte Erscheinungsjahr. Damit hat sich ein Konzept durchgesetzt, das als echte und brauchbare Hilfestellung, Diskussionsplattform und Anregungslieferant für die zahlreichen Commodore-Benutzer gedacht ist. Das wir damit voll ins Schwarze getroffen haben, wird uns auch von vielen CBM/PET NEWS-Lesern aus Deutschland bestaetigt. Besonders geschaezt wird unsere Praezision und konsequente redaktionelle Haltung. Auch im naechsten Jahr wollen wir diese Strategie beibehalten.

In dieser Ausgabe haben wir, dem Wunsch vieler Leser folgend, unter der neuen Rubrik "VC 20 NEWS", diesem Geraetetyp einen grösseren Platz eingeräumt, den wir auch in Zukunft beibehalten wollen. Es darf schliesslich nicht vergessen werden, dass weltweit etwa 500.000 dieser Einsteigergeraete im Gebrauch sind, wir also mit dieser neuen Rubrik einen enorm grossen Leserkreis ansprechen. Allerdings hoffen wir auch, dass dieses neue Forum auch unseren Lesern ein Ansporn sein soll, uns vermehrt Beitraege zum VC 20 zuzusenden.

Wir wünschen Ihnen ein geruhsames Weihnachtsfest, einen fröhlichen Jahreswechsel und für das kommende Jahr weiterhin viel Freude mit Ihrem Commodore.

Die Redaktion



## Inhaltsverzeichnis

---

2 Sortierroutinen	19 Wussten Sie ...
10 Wussten Sie ...	20 Der schnelle Digiplot
11 Light - Pen	21 ROM Routinen
12 Cursorsteuerungen	22 Erscheinungsdaten 1983
20 Monitore	23 Zeichensatzaenderung des VC 20
16 Wussten Sie ...	24 Simon
17 Siedekurve mit HRG	25 Silvester Ueberraschungsprogramm

# Tricks und Tips

## Sortier Routinen

Beim indexsequentiellen Suchen von Daten in der Lagerkartei (CBM/PET NEWS 5/82) haben wir bereits erwahnt, dass diese Methode nur optimal funktioniert, wenn die Daten mit aufsteigender Artikelnummer sortiert abgespeichert werden.

Waehrend das manuelle Sortieren sehr grosse Schwierigkeiten bietet, da das menschliche Gehirn die Sortieroutine, bei der es sich prinzipiell um ein Vertauschen von Informationen handelt, oftmals durchlaufen muss, ist dies für einen Computer wesentlich einfacher. Die Problematik des Sortierens mittels eines Rechners liegt einzig im Zeitaufwand, den der Rechner benötigt. Dies sei an einfachen Beispielen gezeigt:

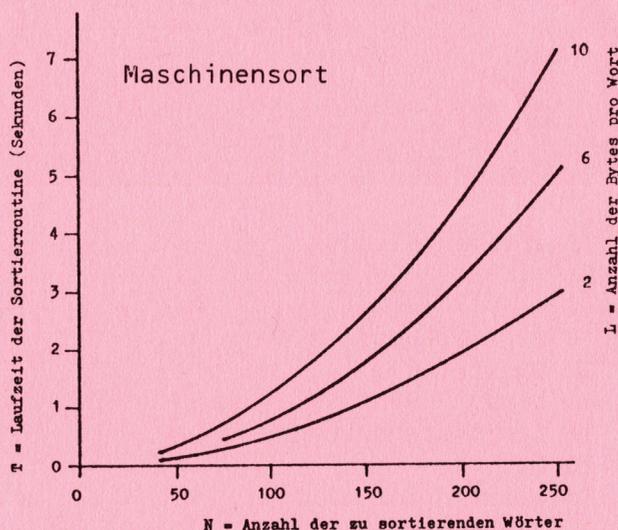
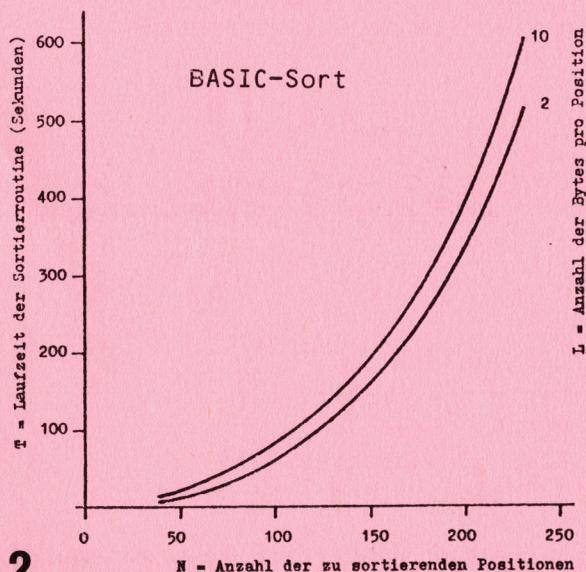
Die Zahlen 2, 4, 1, 5, 3, 6 sollen in aufsteigender Reihenfolge sortiert werden. Um dieses Zahlen in der vorgebenen Reihenfolge einzuordnen sind 27 Operationen erforderlich. Nehmen wir an die Zahlenreihe sei bereits vorsortiert und weise die Reihenfolge 1, 2, 4, 5, 3, 6 auf. Um die 3 an den richtigen Platz zu bringen, muss zuerst 3 mit 5 und anschliessend noch 3 mit 4 vertauscht werden.

Für den Rechner ist dieses Verfahren bei grossen Datenmengen nicht möglich, da nicht so viele freie Speicherplaetze zur Verfügung stehen, als Daten vorhanden sind.

Grundsätzlich ist ein Sortieren von numerischen Daten nach 3 Methoden möglich:

- a. durch Austauschen
- b. durch Auswahl
- c. durch Einfügen

Alle Sortierprogramme können in diese Methodiken eingeordnet werden. Von der Art und Weise, nach welcher Methode Daten sortiert werden, haengt schliesslich die benötigte Zeit ab. Eine ungefaehre Vorstellung der erforderlichen Zeit soll das nachstehende Diagramm vermitteln, eine genauere zeitliche Erfassung erfolgt dann innerhalb der Demoprogramme.



Das erste Programm zeigt die 3 verschiedenen Sortierarten, die alle ins gleiche Programm eingebaut sind. Das Sortieren von Daten ist prinzipiell ein Vergleichen und Austauschen der Daten. Die Programme gestatten einen genauen Vergleich der zum Sortieren benötigten Zeit, sowie die Anzahl Vergleiche und Dreiecksaustausche, die zum Sortieren der 10 Werte erforderlich sind. Hierdurch ist ein genauer Ueberblick möglich.

## Sortierroutinen

```

1000 rem sortierprogramme
1010 rem h. kastien 10.82
1020 ifpeek(50003)=160thenx=20:goto1100
1030 x=1:poke59468,14
1100 print"#####";tab(x+5)"*** SORTIERMETHODEN ***"
1110 printtab(x)"### 1. SORTIEREN DURCH AUSTAUSCH
1120 printtab(x)"# 2. SORTIEREN DURCH AUSWAHL"
1130 printtab(x)"# 3. SORTIEREN DURCH EINFUEGEN"
1140 getz$:ifz$=""then1140
1150 s=val(z$)
1160 ifs>3then1140
1170 printtab(x)"#### 1. AUSGABE AUF DEM BILDSCHIRM"
1180 printtab(x)"# 2. AUSGABE AUF DEM DRUCKER"
1190 getz$:ifz$=""then1190
1200 p=3;ifval(z$)=2thenp=4
1210 ifval(z$)>2then1190
1220 restore:readn
1230 dima(2*n)
1240 fori=1ton
1250 reada(i)
1260 next
1270 open1,p:cmd1
1280 onsgoto2000,3000,4000
2000 rem sortieren durch austausch
2010 t$="SORTIEREN DURCH AUSTAUSCH"
2020 print"#####";tab(x)t$
2030 print
2040 printtab(x);
2050 fori=1ton:printa(i);" ";:next:print:z=ti
2060 f=1
2070 fori=1ton-1
2080 rem paarweises vertauschen
2090 ifa(i)<=a(i+1)thenn2=n2+1:goto2130
2100 rem dreiecktausch
2110 h=a(i):a(i)=a(i+1):a(i+1)=h:n1=n1+1
2120 f=f+1
2130 next :
2140 z=ti-z:iff=1then2210
2150 printtab(x);
2160 fork=1ton
2170 printa(k);" ";
2180 next
2190 print
2200 goto2060
2210 printtab(x)"###ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;" SEK."
2220 printtab(x)"#PAARWEISES VERTAUSCHEN : ";n2
2230 printtab(x)"#DREIECK VERTAUSCHEN   : ";n1
2240 print#1:close1
2250 printtab(x+1)"#WEITER MIT < RETURN >"
2260 getz$:ifz$<>chr$(13)then2260
2270 run
3000 rem sortieren durch auswahl
3010 t$="SORTIEREN DURCH AUSWAHL"
3020 print"#####";tab(x)t$
3030 print

```

```

3040 printtab(x);
3050 fori=1ton:printa(i);" ";:next:print:z=ti
3060 fori=1ton-1
3070 rem suchen des minimums
3080 l=i
3090 forj=i+1ton
3100 ifa(l)<=a(j)thenn2=n2+1:goto3120
3110 l=j:rem index des minimums
3120 nextj
3130 rem dreiecktausch
3140 h=a(i):a(i)=a(l):a(l)=h:n1=n1+1
3150 z=ti-z
3160 printtab(x);
3170 fork=1ton
3180 printa(k);" ";
3190 next
3200 print
3210 nexti
3220 printtab(x)"@ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;" SEK."
3230 printtab(x)"@SUCHEN DES MINIMUMS    : ";n2
3240 printtab(x)"@DREIECK VERTAUSCHEN   : ";n1
3250 print#1:close1
3260 printtab(x+1)"@WEITER MIT < RETURN >"
3270 getz$:ifz$<>chr$(13)then3270
3280 run
4000 rem sortieren durch einfuegen
4010 t$="SORTIEREN DURCH EINFUEGEN"
4020 print"???"tab(x)t$
4030 print
4040 printtab(x);
4050 fori=1ton:printa(i);" ";:next:print:z=ti
4060 fori=2ton
4070 rem von vorne einfuegen
4080 forl=itotstep-1
4090 ifa(l)>=a(l-1)thenn2=n2+1:goto4120
4100 rem dreiecktausch
4110 h=a(l-1):a(l-1)=a(l):a(l)=h:n1=n1+1
4120 nextl
4130 z=ti-z
4140 printtab(x);
4150 fork=1ton
4160 printa(k);" ";
4170 next
4180 print
4190 nexti
4200 printtab(x)"@ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;" SEK."
4210 printtab(x)"@SUCHEN DES MINIMUMS    : ";n2
4220 printtab(x)"@DREIECK VERTAUSCHEN   : ";n1
4230 print#1:close1
4240 printtab(x+1)"@WEITER MIT < RETURN >"
4250 getz$:ifz$<>chr$(13)then4250
4260 run
5000 rem datawerte
5010 data10,5,9,0,3,6,1,8,7,2,4

```

- 1000 - 1010 Remarks
- 1020 - 1030 Selection CBM 8000 oder CBM 3000
- 1100 - 1210 Menu mit Art des Ausdrucks (Bildschirm oder Drucker); der Bildschirm wird mit der Gerateadresse 3 angesprochen.
- 1220 - 1260 Dimensionieren des Datenfeldes und Einlesen der Daten
- 1270 Eröffnen des Schreibefile
- 1280 Bedingte Verzweigung zu den einzelnen Sortierprogrammen

2000 - 2270 Sortieren durch Austausch  
3000 - 3280 Sortieren durch Auswahl  
4000 - 4260 Sortieren durch Einfügen  
5000 - 5010 DATA Statements

Die drei Sortierprogramme werden im folgenden einzeln besprochen :

#### Sortieren durch Austausch

2000 - 2030 Titelausdruck  
2040 - 2050 Ausdruck der unsortierten Daten, die Zeit wird gestartet  
2060 - 2090 FOR ... NEXT Schleife, n2 zaehlt die Anzahl Vergleiche  
2090 Vergleich ob a(i) kleiner oder gleich a(i+1) ist, wenn ja wird der naechste Wert aufgerufen  
2100 - 2130 Dreieckeaustausch, n1 zaehlt die Anzahl Austausche, f wird gleich 0 gesetzt  
2140 Wenn f=1 ist, also noch ein Austausch vorgenommen wurde, folgt Rucksprung und erneuter Vergleich. Die Zeit wird gestoppt  
2150 - 2200 Die sortierten Daten werden ausgedruckt  
2210 - 2240 Ausgabe der Zeit, der Anzahl Vergleiche und Austausche  
2250 - 2270 GET Abfrage und Neubeginn

#### Sortieren durch Auswahl

3000 - 3030 Titelausdruck  
3040 - 3050 Ausdruck der unsortierten Daten und starten der Zeit.  
3060 - 3100 FOR ... NEXT Schleife und Suchen des Minimums. Vergleich ob a(l) kleiner oder gleich a(j) ist, zaehlen der Vergleiche.  
3110 - 3120 Indexieren des Minimums .  
3130 - 3140 Dreieckeaustausch, zaehlen der Austausche  
3150 Stoppen der Zeit  
3160 - 3210 Ausdruck der sortierten Daten  
3220 - 3280 Ausgabe der Zeit, der Anzahl Vergleiche und Austausche, GET Abfrage zum Neubeginn

#### Sortieren durch Einfügen

4000 - 4030 Titelausdruck  
4040 - 4050 Ausdruck der unsortierten Daten und starten der Zeit  
4060 FOR ... NEXT Schleife für alle Daten  
4070 - 4090 Einfügen durch Vergleich ob a(l) grösser oder gleich a(l-1) ist, zaehlen der Vergleiche.  
4100 - 4120 Dreieckeaustausch, zaehlen der Austausche  
4130 - 4190 Stoppen der Zeit und Ausdruck der sortierten Daten  
4200 - 4260 Ausdruck der Zeit, der Anzahl Vergleiche und Austausche, GET zum Neubeginn  
5000 - 5010 DATA Statements

In der Datenverarbeitung haben sich von all den bekannten Sortierrou-tinen drei Verfahren eingebürgert, dies sind

1. Ripple Sort
2. Bubble Sort
3. Shell Metzner Sort

Diese drei Verfahren beruhen alle auf dem Auswahl-sortierverfahren und führen diese Auswahl mittels Vergleichen und Dreieckeaustausch durch. Die drei genannten Sortierverfahren, die in einem Programm zusammengefasst sind, werden an Hand der Flowchart demonstriert. Als Besonderheit ist noch ein Blitz-Sortprogramm beschrieben, das sich durch eine ausserordentlich Schnelligkeit auszeichnet, aber nicht in das Schema der Sortier-routinen passt, da es nicht im bekannten Sinn sortiert, sondern lediglich Daten nach ihrer Grösse liest. Mit diesem Verfahren können nur Integerzahlen verarbeitet werden. Für viele kleinere Sortierarbeiten reicht aber dieses Verfahren vollkommen aus.

## Sortier-routinen

```

1000 rem sortierprogramme
1010 rem h. kastien 10.82
1020 ifpeek(50003)=160thenx=20:goto1040
1030 x=1:poke59468,14
1040 print"#####";tab(x+5)"*** SORTIERMETHODEN ***"
1050 printtab(x+4)"### 1. RIPPLE SORT"
1060 printtab(x+4)"# 2. BUBBLE SORT"
1070 printtab(x+4)"# 3. SHELL - METZNER SORT"
1080 printtab(x+4)"# 4. BLITZ SORT"
1090 getz$:ifz$=""then1090
1100 s=val(z$)
1110 ifs>4then1090
1120 printtab(x+4)"#### 1. AUSGABE AUF DEM BILDSCHIRM"
1130 printtab(x+4)"# 2. AUSGABE AUF DEM DRUCKER"
1140 getz$:ifz$=""then1140
1150 p=3;ifval(z$)=2thenp=4
1160 ifval(z$)>2then1140
1170 restore:readm
1180 dimd(2*m)
1190 fori=1tom
1200 readd(i)
1210 next
1220 open1,p:cmd1
1230 onsgoto1240,1490,1720,2020
1240 rem sortieren durch austausch
1250 t$="R I P P L E   S O R T"
1260 print"#####";tab(x)t$;"####"
1270 print:printtab(x);
1280 fori=1tom:printd(i);" ";:next:print:z=ti
1290 c=0
1300 fori=1tom-1:n2=n2+1
1310 ifd(i)<=d(i+1)then1330
1320 t=d(i);d(i)=d(i+1);d(i+1)=t:c=1:n1=n1+1
1330 next
1340 ifc=1then1290
1350 z=ti-z
1360 printtab(x);
1370 fork=1tom
1380 printd(k);" ";
1390 next
1400 ifc=1then1290
1410 print:print
1420 printtab(x)"###ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;" SEK."
1430 printtab(x)"#PAARWEISES VERTAUSCHEN : ";n2
1440 printtab(x)"#DREIECK VERTAUSCHEN    : ";n1
1450 print#1:close1
1460 printtab(x+1)"#WEITER MIT < RETURN >"
1470 getz$:ifz$<>chr$(13)then1470
1480 run
1490 rem sortieren durch auswahl

```

```

1500 t$="B U B B L E   S O R T"
1510 print"☺☺☺"tab(x)t$;"☺☺☺"
1520 print:printtab(x);
1530 fori=1tom:printd(i);" ";next:print:z=ti
1540 fori=1tom-1
1550 forj=i+1tom:n2=n2+1
1560 ifd(i)<=d(j)then1580
1570 t=d(i):d(i)=d(j):d(j)=t:n1=n1+1
1580 next:next
1590 z=ti-z
1600 printtab(x);
1610 fork=1tom
1620 printd(k);" ";
1630 next
1640 print
1650 printtab(x)"☺☺ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;" SEK."
1660 printtab(x)"☺☺SUCHEN DES MINIMUMS    : ";n2
1670 printtab(x)"☺☺DREIECK VERTAUSCHEN    : ";n1
1680 print#1:close1
1690 printtab(x+1)"☺☺WEITER MIT < RETURN >"
1700 setz$:ifz$<>chr$(13)then1700
1710 run
1720 rem sortieren durch einfuesen
1730 t$="S H E L L - M E T Z N E R   S O R T "
1740 print"☺☺☺"tab(x)t$;"☺☺☺"
1750 print:printtab(x);
1760 fori=1tom:printd(i);" ";next:print:z=ti
1770 n=m
1780 m=int(m/2):ifm=0then1890
1790 j=1:k=n-m
1800 i=j
1810 l=i+m:n2=n2+1
1820 ifd(i)<=d(l)then1860
1830 t=d(i):d(i)=d(l):d(l)=t:n1=n1+1
1840 ifi<lthen1860
1850 goto1810
1860 j=j+1
1870 ifj>kthen1780
1880 goto1800
1890 z=ti-z
1900 printtab(x);
1910 fork=1ton
1920 printd(k);" ";
1930 next
1940 print
1950 printtab(x)"☺☺ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;" SEK."
1960 printtab(x)"☺☺SUCHEN DES MINIMUMS    : ";n2
1970 printtab(x)"☺☺DREIECK VERTAUSCHEN    : ";n1
1980 print#1:close1
1990 printtab(x+1)"☺☺WEITER MIT < RETURN >"
2000 setz$:ifz$<>chr$(13)then2000
2010 run
2020 rem blitzsort
2030 t$="B A S I C   -   B L I T Z   S O R T"
2040 print"☺☺☺"tab(x)t$;"☺"
2050 printtab(x)"☺☺☺Die Methode ist nur fuer Integerzahlen geeignet !☺☺☺"
2060 print:printtab(x);
2070 fori=1tom:printd(i);" ";next:print:restore:readm
2080 z=ti:fori=1tom:readd:d(d)=d:next:z=(ti-z)
2090 printtab(x);
2100 fork=0tom-1
2110 printd(k);" ";
2120 next
2130 print
2140 printtab(x)"☺☺ERFORDERLICHE ZEIT      : ";int(z/60*100)/100;"
2150 print#1:close1

```

SEKUNDEN"

```

2160 printtab(x+1)"WEITER MIT < RETURN >"
2170 getz$:ifz$<chr$(13)then2170
2180 run
2190 rem datawerte
2200 data10,5,9,0,3,6,1,8,7,2,4

```

- 1000 - 1010 Remarks
- 1020 - 1030 Selection CBM 8000 oder CBM 3000
- 1040 - 1160 Menu mit Art des Ausdrucks (Bildschirm oder Drucker); der Bildschirm wird mit der Geraeteadresse 3 angesprochen.
- 1170 - 1210 Dimensionieren des Datenfeldes und Einlesen der Daten
- 1220 Eröffnen des Schreibefile
- 1230 Bedingte Verzweigung zu den einzelnen Sortierprogrammen
- 1240 - 1480 Ripple Sort
- 1490 - 1710 Bubble Sort
- 1720 - 2010 Shell-Metzner Sort
- 2020 - 2180 Basic-Blitz Sort
- 2190 - 2200 DATA Statements

Die drei Sortierprogramme werden im folgenden einzeln besprochen :

#### Ripple Sort

- 1240 - 1270 Titelausdruck
- 1280 Ausdruck der unsortierten Daten, die Zeit wird gestartet
- 1290 Variable c wird gleich 0 gesetzt
- 1300 FOR ... NEXT Schleife, n2 zaehlt die Anzahl Vergleiche
- 1310 Vergleich ob d(i) kleiner oder gleich d(i+1) ist, wenn ja wird der naechste Wert aufgerufen
- 1320 Dreieckeaustausch, n1 zaehlt die Anzahl Austausche, c wird gleich 1 gesetzt
- 1340 Wenn c=1 ist, also noch ein Austausch vorgenommen wurde, folgt Rücksprung und erneuter Vergleich.
- 1350 Die Zeit wird gestoppt
- 1360 - 1390 Die sortierten Daten werden ausgedruckt
- 1410 - 1440 Ausgabe der Zeit, der Anzahl Vergleiche und Austausche
- 1450 - 1480 GET Abfrage und Neubeginn

#### Bubble Sort

- 1490 - 1530 Titelausdruck, Ausdruck der unsortierten Daten und starten der Zeit
- 1540 - 1560 Vergleich ob d(i) kleiner oder gleich d(j) ist. Es wird jedes d(j) mit einem d(i) verglichen, wenn ja dann wird der naechste Vergleich aufgerufen, zaehlen der Vergleiche
- 1570 Dreieckeaustausch, zaehlen der Austausche
- 1590 Stoppen der Zeit
- 1600 - 1640 Ausdruck der sortierten Daten
- 1650 - 1710 Ausgabe der Zeit, der Anzahl Vergleiche und Austausche, GET Abfrage zum Neubeginn

#### Shell - Metzner Sort

- 1720 - 1760 Titelausdruck, Ausdruck der unsortierten Daten und starten der Zeit

1770 n wird gleich der Anzahl Werte gesetzt  
 1780 Es wird der Integerwert der Haelfte der Daten ermittelt, wenn dieser Wert gleich 0 ist, erfolgt Abbruch des Sortiervorganges.  
 1790 - 1810 Setzen von Variablen  
 1820 Vergleichen, wenn Vergleich stimmt, Aufruf des naechsten Wertes  
 1830 Dreieckeaustausch, zaehlen der Austausch  
 1840 Wenn i kleiner 1 dann Aufruf des naechsten Wertes  
 1850 - 1880 Wenn j grösser k ist, wird weiter sortiert  
 1890 - 1940 Soppen der Zeit und Ausdruck der sortierten Daten  
 1950 - 2010 Ausdruck der Zeit, der Anzahl Vergleiche und Austausch, Get zum Neubeginn

#### Basic - Blitz Sort

2020 - 2070 Titelausdruck, Ausgabe der unsortierten Daten, Einlesen der Anzahl Daten  
 2080 Starten der Zeit, Einlesen nach Grösse, Stoppen der Zeit  
 2090 - 2130 Ausgabe der sortierten Daten  
 2140 - 2180 Ausgabe der Zeit und GET Abfrage für Neubeginn  
 2190 - 2200 DATA Statements

Mit dem Shell-Metzner Sort ist auch ein Sortieren alphanummerischer Daten möglich. Diese Art des Sortierens soll an einem separaten Beispiel gezeigt werden.

#### Alphasortprogramm

```

10 rem sortieren von alphanumerischen daten
20 ifpeek(50003)=160thenx=20:goto40
30 x=1:poke59468,14
40 print"#####";tab(x+5)"*** Sortierprogramm ***"
50 printtab(x+3)"##Eingabe von maximal 100 Namen"
60 printtab(x+3)"##Abbruch der Eingaben mit Ende#."
65 fori=1to2000:next
70 print"#####"
100 dimd$(100)
140 i=1
150 printtab(x)i;". Name";
170 inputd$(i)
180 ifd$(i)<>"ende"theni=i+1:goto150
190 n=i-1:print
210 m=n
220 m=int(m/2):ifm=0then340
230 j=1:k=n-m
240 i=j
250 l=m+i
260 ifd$(i)<=d$(l)then310
270 t#=d$(i):d$(i)=d$(l):d$(l)=t#
280 i=i-m
290 ifi<1then310
300 goto250
310 j=j+1
  
```

```

320 ifj>kthen220
330 goto240
340 print
350 print"###";tab(x)"@Ausgabe der sortierten Daten###"
400 fori=1ton
410 printtab(x)i". ";d$(i)
420 next
ready.

```

Es sind die verschiedensten Verfahren gezeigt worden, die alle zum Sortieren der anfallenden Daten im Programm Lagerbuchhaltung bzw. Adressverwaltung eingesetzt werden können. Allerdings gibt es hier eine weitere Schwierigkeit, da die zu sortierenden Daten nicht alle gleichzeitig zur Verfügung stehen. Dies ist vor allem dann der Fall, wenn Daten nachträglich angefügt werden, da dann bereits in den verschiedenen File Daten abgelegt sind, und die neu hinzukommenden Werte, in die verschiedenen File eingeordnet werden müssen. Dies ist jedoch eine Sache der Floppyprogrammierung und nicht der Sortierverfahren. Nachdem die Grundlagen des Sortierens erarbeitet worden sind, soll das Sortieren der Daten in die File im nächsten Heft zur Diskussion stehen.

## Wussten Sie schon...

dass eine Cursorsteuerung mittels eines kleinen Maschinenprogramms auch beim VC 20 möglich ist.

### Cursorsteuerung

```

10000 REM          SETCURSOR FUER VC 20
10010 :
10020 REM DER CURSOR WIRD AUF EINE BELIEBIGE SCHREIBSTELLE GESETZT
10030 :
10040 REM          AUFRUF SYSXXXX,ZEILE,SPALTE
10050 :
10060 REM          ZEILE: 1-23          SPALTE: 1-22
10070 :
10080 REM          M. BAUER          8000 MUENCHEN 21
10090 :
10100 REM          NOV 1982
10110 :
10120 A=1024:REM PROGRAMM VOLL VERSCHIEBBAR
10130 FORB=ATO A+31:READC:POKEB,C:NEXT
10140 :
10150 REM ***** DATA FUER VC 20
10160 DATA32,253,206,32,158,215,202,224,23,176,18,134
10170 DATA214,32,253,206,32,158,215,202,224,22,176,5,134
10180 DATA211,76,135,229,76,72,210
READY.

```

# Hardware

## Light - Pen

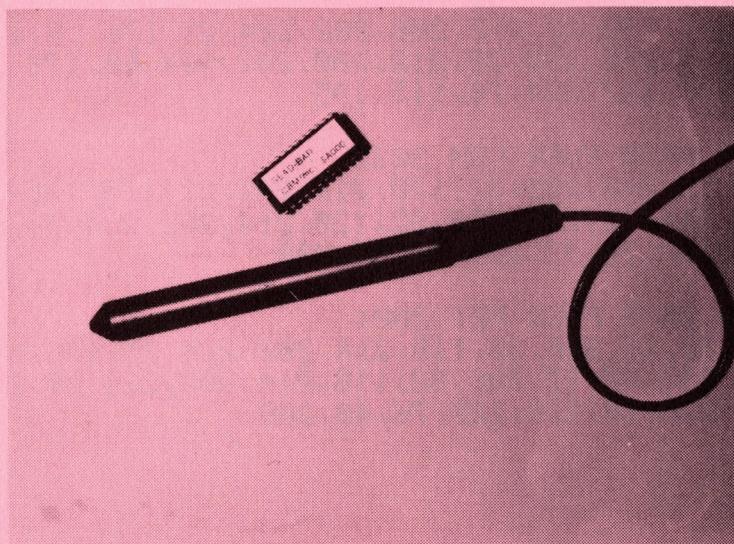
Das Barcodepack hat laenger auf sich warten lassen, als wir ursprünglich vermutet hatten. Hieran waren drei Faktoren schuld, die wir Ihnen nicht vorenthalten wollen. Dies war einmal die grosse Nachfrage und die damit verbundene Schwierigkeit, die erforderlichen Komponenten innert nützlicher Frist in ausreichender Stückzahl zu besorgen, aber auch der Wunsch nach einem Lesestift, der möglichst vielen Anwendungen gerecht wird. So ist es uns heute unter anderem möglich, den Lesestift auch als Light-Pen zu benutzen. Hierdurch wird naturgemaess die Einsatzbreite des Stiftes wesentlich vergrössert, ohne das zusaetzlich Komponenten erforderlich werden. Wir sind sicher, dass wir hier einem echten Bedürfniss nachgekommen sind, da es der Wunsch jedes Geraetebenutzers ist, mit wenigen Optionen, ein grosses Anwendungsgebiet zu überstreichen.

Als logische Schlussfolgerung des grossen Bestellungseinganges und der erweiterten Anwendungsmöglichkeiten, denen wir uns in den naechsten Ausgaben der Hefte auch von der softwaremaessigen Seite widmen werden, haben wir uns entschlossen, auch weiterhin das Barcodepack in unserem Programm zu führen. Lesestift, ROM und Software sind also weiterhin als Pack zum Preis von Fr. 345.-- gegen Voreinzahlung auf Postcheckkonto Luzern 60-27181, Stuttgart 3786-709 oder Wien PSK 7975.035 durch den

Mikro + Kleincomputer  
Informa Verlag AG  
Postfach 1401  
CH 6000 Luzern 15

erhaeltlich. Das Foto soll Ihnen vor allem die schlanke, handlich Form des Lesestiftes noch einmal vor Augen führen. Der Lesestift ist mit dem optoelektronischen Leser HEDS 3000 der Hewlett Packard ausgerüstet, verfügt also über ein professionelles Optosystem.

Von der Software ist noch zu sagen, dass neben dem bereits erwachten Einsatz als Light-Pen auch erweiterte Druckprogramme für die Drucker der Commodore Serie und Einleseprogramme für andere Barcode in Vorbereitung sind, alle diese Optionen erscheinen in Verlaufe des Jahres 1983.



# Leserbriefe

## Cursorsteuerung

Unser Beitrag Cursorsteuerung mit Maschinenprogrammen ist auf ein reges Interesse gestossen, (siehe hierzu auch Maschinensprache 6/82) vor allem haben sich viele unserer Leser Gedanken darüber gemacht, wie man das Programm auch auf den Rechnern der Serie 2000, 4000 und 8000 zum Laufen bringen kann. Wir drucken Ihnen dieses Programme in der Folge ab, wobei sich prinzipiell bei den verschiedenen Systemen nur die einzelnen DATA-Statements ändern.

## Cursorsteuerung

```
10000 REM          SETCURSOR FUER CBM 8023
10010 :
10020 REM DER CURSOR WIRD AUF EINE BELIEBIGE SCHREIBSTELLE GESETZT
10030 :
10040 REM          AUFRUF SYS826,ZEILE,SPALTE
10050 :
10060 REM          ZEILE: 1-25          SPALTE: 1-80 (1-40)
10070 :
10080 REM          M. BAUER                      8000 MUENCHEN 21
10090 :
10100 REM          NOV 1982
10110 :
10120 A=826:REM PROGRAMM VOLL VERSCHIEBBAR
10130 FORB=ATO A+31:READ C:POKE B,C:NEXT
10140 :
10150 REM ***** DATA FUER CBM 8032
10160 DATA 32,245,190,32,212,200,202,224,25,176,18,134
10170 DATA 216,32,245,190,32,212,200,202,224,80,176,5,134
10180 DATA 198,76,103,224,76,115,195
10190 :
10200 REM ***** DATA FUER CBM 4032
10210 DATA 32,245,190,32,212,200,202,224,25,176,18,134
10220 DATA 216,32,245,190,32,212,200,202,224,40,176,5,134
10230 DATA 198,76,127,224,76,115,195
10240 :
10250 REM ***** DATA FUER CBM 3032
10260 REM DATA 32,248,205,32,120,214,202,224,25,176,18,134
10270 REM DATA 216,32,248,205,32,120,214,202,224,40,176,5,134
10280 REM DATA 198,76,93,226,76,35,209
10290 :
10300 REM ***** DATA FUER PET 2001
10310 REM DATA 32,17,206,32,118,214,202,224,25,176,18,134
10320 REM DATA 245,32,17,206,32,118,214,202,224,40,176,5,134
10330 REM DATA 226,76,219,229,76,48,209
READY.
```

# Programm des Monats BASIC-BASIC

## Monitore

Haben Sie sich auch schon oft darüber geärgert, wenn Sie in einer Zeitschrift ein interessantes Maschinensprachprogramm fanden und dieses in Ihre Anlage eintippen mussten, um schlussendlich feststellen zu müssen, dass Ihnen bei der Eingabe der vielen Hexadezimalzahlen oder DATA-Statements eine Reihe von Fehlern unterlaufen sind. Wer hat dann voll Zorn nicht schon sein Heft in die Ecke geworfen. Vielen anderen Lesern ist es sicher auch schon so ergangen, denn die Programme die wir Ihnen heute als Programm des Monats abdrucken, stammen aus Leserbriefen. Zuerst aber einmal etwas Theorie über die Eingabe von DATEN eines Maschinenprogramms.

Zur Eingabe von Maschinenprogrammen gibt es eine Reihe von Möglichkeiten, von denen wir Ihnen hier die wichtigsten aufzeigen wollen.

1. Eingabe des Programms mit dem TIM Monitor als Hexadezimalzahlen.
2. Eingabe als DATA-Statements und Einpoken in den Speicher über ein BASIC-Programm
3. Eingabe über Monitore

### Eingabe über den TIM-Monitor

Der TIM-Monitor (Terminal Interface Monitor) wird bei allen CBM-Rechnern mit SYS 1024 aufgerufen. Er erlaubt folgende Befehle:

M 033a,034f	Anzeige des Speicherinhaltes von 033a bis 034f
R	Anzeige der Registerinhalte
G 033a	Ausführung des Programms ab Adresse
X	Rückkehr ins BASIC-Programm
L"Name",08	Laden eines Maschinenprogramms ab Disk
S"Name",08,033a,034f	Abspeichern des Programms mit der Startadresse 033a und der Endadresse 034e auf Disk

Um nun eine Reihe von Befehlen mittels TIM-Monitor einzugeben, wird der TIM mit SYS 1024 aufgerufen und M Startadresse,Endadresse eingetippt. Auf dem Bildschirm erscheint nun der Inhalt dieser Speicherstellen. Man überschreibt nun den Speicher einfach mit den neuen Zahlen, die natürlich in hexadezimaler Form vorliegen müssen, andernfalls müssen Sie umgerechnet werden. Nach jeweils 1 Zeile, also 8 zweistelligen HEX-Zahlen muss die RETURN-Taste gedrückt werden. Ein kleines Maschinenprogramm sieht dann auf dem Bildschirm folgendermassen aus:

```
.m 0384,03bd
.: 0384 a2 00 ad e8 03 9d 00 80
.: 038c 9d 80 87 e8 e0 50 d0 f5
.: 0394 a2 50 86 54 a2 80 86 55
.: 039c a2 00 a0 00 91 54 a5 54
.: 03a4 18 69 4f 90 02 e6 55 85
.: 03ac 54 ad e8 03 91 54 e6 54
.: 03b4 d0 02 e6 55 e8 e0 18 d0
.: 03bc e3 60 ff 00 ff 00 ff 00
.x
ready.
```

Diese Methode ist sehr einfach, jedoch für den Anfaenger nicht sonderlich gut geeignet, da sie schnell zu Fehlern führt. Nach erfolgter Eingabe wird das Programm mit G Startadresse gestartet oder wie oben beschrieben, abgespeichert, wobei die Endadresse um 1 höher sein muss, als das letzte Statement effektiv hat. Es kann nun auch mit X ins BASIC zurückgesprungen werden und das Maschinenprogramm mit SYS Startadresse (dezimal) aufgerufen werden.

#### Eingabe über DATA-Statements

Vor allem in reinen BASIC-Programmen, die aber Maschinensprachroutinen enthalten, werden diese vornehmlich über DATA-Statements eingegeben. Man muss hierzu allerdings die Adressen und die HEX-Befehle in Dezimalzahlen umrechnen. Diese können nun mittels POKE in die Speicherstellen geschrieben werden. Das obige kleine Maschinenprogramm sieht als BASIC Programm mit DATA-Statements dann so aus.

```
10 for a=900 to 957: read b: poke a, b: next
20 data 162,0,173,232,3,157,0,128,157,128,135,232,224,80,208
30 data 245,162,80,134,84,162,128,134,85,162,0,160,0,145,84
40 data 165,84,24,105,79,144,2,230,85,133,84,173,232,3,145
50 data 84,230,84,208,2,230,85,232,224,24,208,227,96
reads.
```

Der Vorteil dieser Methode ist klar, der Programmierer, der nur reines BASIC gewohnt ist, kommt mit diesem Verfahren besser zurecht, allerdings muss alles in Dezimalzahlen umgerechnet werden, was auch nicht gerade wenig Arbeit ist.

#### Eingabe mittels Monitoren

Es hat sich gezeigt, dass die beschriebenen Verfahren sowohl positive als auch negative Seiten haben, es hat daher nicht an Versuchen gefehlt, mittels Hilfsprogrammen die Daten einzugeben, die es ermöglichen, sowohl hexadezimal als auch dezimal zu arbeiten. Zwei solcher Monitorprogramme stellen wir Ihnen heute vor.

Das Programm "Speicherpoke'r" ist ein kleines BASIC-Programm, bei dem Start- und Endadresse und die Daten in hexadezimaler Form eingegeben werden. Die Daten werden nach jeder Zeile mittels POKE direkt im Speicher abgelegt. Nach Beendigung der Eingaben, wird der gesamte Memoryinhalt auf dem Bildschirm angezeigt und kann dann durch Drücken der "S"-Taste auf Disk abgespeichert werden.

### Speicherpoke'r

```
100 REM* S P E I C H E R P O K E ' R *
110 REM* E.MEIER 3052 ZOLLIKOFEN *
120 REM* CBM3032/8032 15.8.82 *
130 REM* HEX-ADR. 4-STELLIG EINGEBEN *
140 REM* DANN SPEICHERZELLEN IN HEX *
150 REM* EINGEBEN. *
160 REM* IMMER GANZE ZEILE FUELLEN *
170 REM* EV.MIT 00 AUFFUELLEN,DANN *
180 REM* MIT RETURN ABSCHLIESSEN *
190 GOTO 200:"␣=CRSAB ␣=CRSLINKS ␣=CLRHOME ␣=HOME ␣=CRSRECHTS
```

```

200 POKE158,0:PRINT"##### VON ADRESSE 1 BIS ADRESSE 2 :@
210 PRINT" BEREICHE:#027A-03F9 + 0000- 7FFF"
220 PRINT" DEZIMAL : 634-1017 + 3328-32767"
230 IFE<2THENA=0:E=E+1:PRINT"#####E"||. HEX-ZAHL: ";:GOTO250
240 C=0:GOTO330
250 GETA#:IFLEN(A1#)<>4THEN290
260 IFA>633AND<1018THEN280
270 IFA<3328ORA>32767THENRUN
280 PRINT" ="A"DEZ":J=K:K=A:A1#="":FORI=1TO5+4:NEXT:GOTO200
290 IFA#<"0"ORA#>"9"AND<"A"ORA#>"F"THEN250
300 B=ASC(A#)-55:IFB<10THENB=VAL(A#)
310 A=A*16+B:A1#=#A#
320 PRINTA#;:C=1:GOTO250
330 IFJ>KTHENRUN
340 PRINT"##### P E I C H E R P O K E ' R":POKE53,13
350 PRINT"#####@
360 PRINT"ADRESSE/DEZ/HEX KORREKTUREN MIT @":PRINT"EINGABE IN HEX@"
370 FORI=JTOK:A1=I:I#=RIGHT#(" "+STR#(I)+ " ",6):IFC>0THEN420
380 B#="":FORP=3TO0STEP-1
390 B=INT(A1/16+P):IFB<10THENB#=B#+CHR#(B+48):GOTO410
400 B#=B#+CHR#(B+55)
410 A1=A1-(B*16+P):NEXT:IFS=0THENB1#=B#:S=1
420 IFC=0THENPRINTI#B# " . . . . .#####";
430 GETA#:IFA#=#CHR#(13)ANDC=0THEN600
440 IFA#<"0"ORA#>"9"AND<"@"ORA#>"F"THEN430
450 IFA#<"@"THEN490
460 IFD=0ANDC=0THEN430
470 IFC=0THENC=7:D=D-1:I=I-1:POKEI,0:PRINTTAB(32)"|.#####";:GOTO430
480 PRINT"#####.#####";:I=I-1:C=C-1:POKEI,0:GOTO430
490 A=ASC(A#):IFA>64THENA=A-55:GOTO510
500 A=VAL(A#)
510 PRINTA#;:B=A*16
520 GETA#:IFA#<"0"ORA#>"9"AND<"@"ORA#>"F"THEN520
530 IFA#<"@"THEN550
540 PRINT"#####";:GOTO430
550 A=ASC(A#):IFA>64THENA=A-55:GOTO570
560 A=VAL(A#)
570 PRINTA#+" ";:B=A+B:POKEI,B:C=C+1:IFC=8THENPRINT:C=0:D=D+1
580 IFC=0ANDD=16THEND=0:PRINT"#####";
590 NEXT
600 PRINT"##### M "B1#" "B#"
610 PRINT"#####WENN DIE EINGABE ABGESPEICHERT WERDEN
620 PRINT"#####SOLL, MIT CURSOR AUF ##### GEHEN + RETURN
630 PRINT"#####MIT ##### AUS DEM MONITOR AUSSTIEGEN, UND
640 PRINT"#####VOR DEM LADEN EINES ANDEREN PROGRAMMS
650 PRINTTAB(8)"##### SYS64721 EINGEBEN! #####:REM BEI 8050 SYS64790
660 PRINT"#####CHR#(34)"@@"B1#/"B#CHR#(34)",08,"B1#","B#":SYS1024
670 END
READY.

```

Beim Programm "Eingabe-Monitor" wird nur die Startadresse eingeben und danach der entsprechende Wert. Bei diesem Programm können alle Werte sowohl dezimal als auch hexadezimal (Adresse bis 5 stellig dezimal oder 4 stellig hexadezimal, Daten 2 stellig hexadezimal oder 3 stellig dezimal) eingeben werden. Das Programm wird mit "\*" abgebrochen und dann auf dem Bildschirm gesamthaft dargestellt. Eine direkt Abspeicherung ist bei diesem Programm nicht vorgehen. Da die Programme in Ihrem Aufbau nicht sehr schwierig sind, haben wir hier auf eine Programmbe-schreibung verzichtet.

**Eingabemonitor**

```

10 print"#####Eingabemonitor#####
11 rem *copyright by ins. f.tilgner **          **          wien 1982          **
12 print
15 print"einsabe in @h#ex , in @d#ezimal
20 setk#:ifk#<"h"andk#<"d"then20
22 ifk#="h"thenclr:print"#####:goto25
24 ifk#="d"thenclr:print"#####:goto300
25 print"beginadresse nicht: $ 0400 - $ 0a00 !"
26 print"

```

```

30 input"beginnadresse (hex)   #####";a$:ifa$=" "thenerprint"##":G0T030
35 x$=a$
40 gosub1400;a=g:g=0;bg=a;aa=bg
50 print"$ @";:gosub1500:input"#.byte(end='*')", (hex xx)   #####;b$
52 x$=b$
53 ifb$="*"thenb$="**"
54 iflen(b$)<>2thenprint"@wiederholen!";a=bg:goto50
55 ifb$="**"then200
60 gosub1400;b=g:g=0
100 pokebg,b:ed=a
110 a=a+1;bg=bg+1;a=bg
120 goto50
200 print"@@m#onitor 1024 (54386)
210 getk$:ifk$<>"m"then210
220 ifk$="m"thenprint"?? $ @";a=aa:gosub1500:print" - ";a=bg-1:gosub1500:sys1024
300 print"beginnadresse nicht:1024 - 2560 !"
310 print"
320 input"beginnadresse (dez)   #####";a
330 print:bg=a;aa=bg
340 print"$ @";:gosub1500:input"#.byte (end='*'),dez   #####;b$
342 ifb$=" "thenprint"@wiederholen!";a=bg:goto340
345 b=val(b$)
350 ifb>255thenprint"@wiederholen !":print" ":A=BG:G0T0340
355 ifb$="*"then400
360 pokebg,b:ed=a
370 a=a+1;bg=bg+1;a=bg:goto340
400 print"@@m#onitor 1024 (54386)
410 getk$:ifk$<>"m"then410
420 ifk$="m"thenprint"?? $ @";a=aa:gosub1500:print" - ";a=bg-1:gosub1500:sys1024
1400 hl=0:g=0:hj=len(x$)
1410 fori=1tohj
1415 xx$=mid$(x$,i,1)
1420 h=asc(xx$):hl=hj-i
1430 ifh<=57then1445
1435 b(i)=(h-55)*(16^hl)
1440 goto1450
1445 b(i)=(h-48)*(16^hl)
1450 next
1455 fori=1tohj:g=g+b(i):next:return
1500 goto1550:rem**input"Zumzuwandelnde zahl";a:goto1550
1510 n=0:ad$=""
1520 n=n+1;b=int(a/z):c=a-(b*z):b(n)=int(c+.5):a=b
1530 if b>0 then 1520
1540 return
1550 z=16
1560 gosub 1510
1570 goto 1600
1580 n=n-1
1590 if n=0 thenreturn
1600 if b(n)=>10 then 1640
1610 a1=b(n)+48
1620 print chr$(a1);
1630 goto 1580
1640 a1=b(n)+55:goto1620
1650 end
ready.

```

Wir wollen hoffen, dass durch diese Eingabeprogramme auch die Maschinensprachneulinge auch ein wenig mehr Freude an dieser interessanten Anwendung finden werden.

## Wussten Sie schon...

dass Sie beim VC 20 die RUN/STOP Taste mit POKE 788,194 ausser Funktion setzen und die interne Uhr ausschalten. Die Umkehrung dieses Befehls lautet POKE 788,191.

## Siedekurve mit HRG

Im Heft 5/82 wurde ein Programm zur Berechnung einer Siedekurve veröffentlicht und gleichzeitig eine Erweiterung des Programms auf hochauflösende Graphik angekündigt.

Die Programmerweiterung läuft nur mit der Original Commodore 8032 HRG, diese besteht aus einer Platine mit der erforderlichen Hardware und einem ROM mit einem speziellen Befehlsatz. Die Auflösung des Bildschirms betragen 512\*256 Punkte. Da die Graphik aber nicht den ganzen Bildschirm beansprucht, reduziert sich die Auflösung der Graphik auf 188 \* 300 Punkte.

Die Commodore HRG hat einen eigenen Befehlsatz, obwohl zu HRG anderer Hersteller Unterschiede bestehen, ist trotzdem eine Umschreibung auf andere Systeme möglich, ebenso lehnt sich der Befehlsatz in gewissen Grenzen auch an den Befehlsatz des Plotters an. Die in diesem Programm gebrauchten Befehle sollen kurz erklärt werden.

#clear	löscht den HRG-Bildschirm
#grmode	gestattet die Benutzung der HRG mit normalem PRINT
#pen	bezeichnet die Art der Darstellung
#mowex,y	bewegt den Zeichenstift ohne zu Zeichnen auf den Punkt x,y
#imovex,y	bewegt den Zeichstift ohne zu zeichnen von der alten Position um den Betrag +x und +y
#drawx,y	zeichnet eine Linie von x nach y
#langle	ermöglicht den Ausdruck eines Textes im Winkel von 90, 180, 270 o zur X-Achse
#label	schreibt den Text auf den Bildschirm
#exg	Rückkehr ins BASIC

## Siedekurve mit HRG

```
2000 rem kurve mit hrs
2010 sys36864
2020 print"␣";tab((57-len(s$))/2)"*** Siedekurve von ";s$;" ***"
2030 #clear 3:#grmode3,2:#pen 1
2040 #move135,35:#draw435,35
2050 #move135,35
2060 #draw135,225
2070 #move135,35
2080 for i=1to11
2090 u=i*30
2100 #drawu+105,30
2110 #imove30,5
2120 next
2130 #move135,35
2140 for i=0to19
2150 u=i*10+25
2160 #draw130,u+10
2170 #imove5,10
2180 next
2190 #move100,222
2200 for i=1to20
```

```

2210 x$=str$(800-i*40)
2220 if len(x$)<4thenx$=" "+x$:goto2220
2230 #label x$
2240 #imove-28,-10
2250 next
2260 #move50,80
2270 #langle1
2280 #label"druck in mm hg"
2290 #move320,10:#langle0
2300 #label"siedepunkt in o c"
2310 #move110,22
2320 fori=0to10
2330 k$=str$(int((kp-273)/10*i))
2340 iflen(k$)<4thenk$=" "+k$:goto2340
2350 #imove2,0
2360 #label k$
2370 next
2380 print"#####Druck      Siedepunkt"
2390 print" -----"
2400 #move135,35
2410 fori=1to188
2420 p1=i*.40426:gosub10000
2430 p=(k(i)*300/(kp-273))
2440 ifp<0thenp=0
2450 gosub2530:#drawp+135,i+35
2460 next
2470 print"#####Zurueck ins Menu ?"
2480 printtab(7)"@ (j/n) "
2490 getz$:ifz$=""then2490
2500 ifz$="j"then#exg:run
2510 ifz$="n"thenprint"n":#exg:end
2520 goto2490
2530 pp$=str$(int(p1*10)):kk$=str$(k(i))
2540 iflen(pp$)<4thenpp$=" "+pp$
2550 iflen(kk$)<4thenkk$=" "+kk$
2560 print"#####";pp$;" mm Hg "kk$;" o C"
2570 return

```

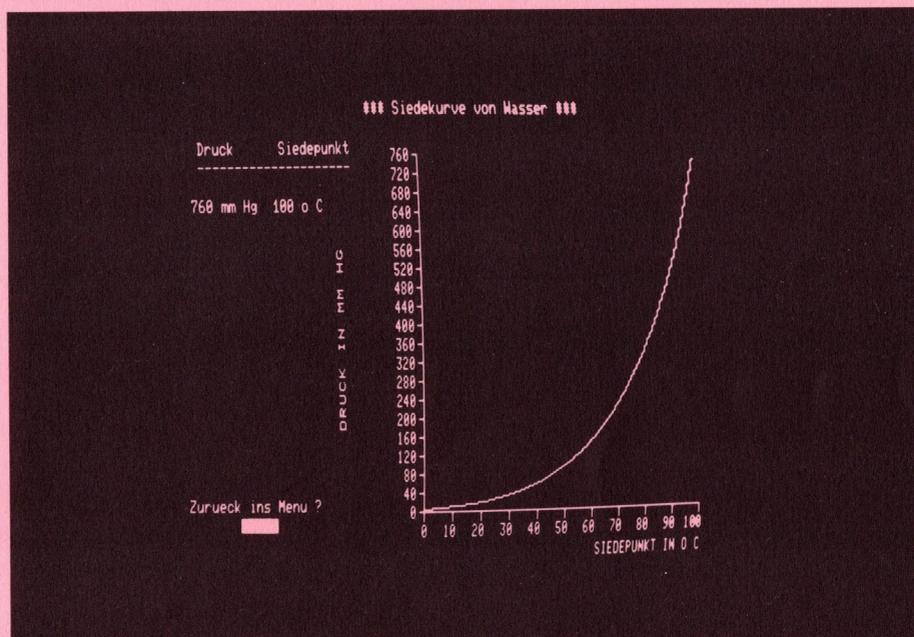
ready.

#### Programmbeschreibung

---

- 2000 - 2010 Remark und Aufruf der HRG mit SYS 36864 (es kann nicht gleichzeitig ein Toolkit benutzt werden)
- 2020 Ausdruck des Titels mit PRINT in der Mitte des Bildschirms
- 2030 Löschen des HRG-Bildschirms, es wird PAGE 1 der HRG und der normale Bildschirm nebeneinander benutzt, alle Linien werden als durchgehende Linien gezeichnet
- 2040 Ab Position X=135 und Y=35 wird eine Linie zum Punkt X=435 und Y=35 (x-Achse) gezeichnet.
- 2050 - 2060 Zeichnen der Y-Achse wie bei der X-Achse beschrieben
- 2070 - 2120 Zeichnen der Unterteilung der Y-Achse
- 2130 - 2180 Zeichnen der Unterteilung der X-Achse
- 2190 - 2250 Bezeichnung der Y-Achse
- 2260 - 2280 Der Text "Druck in mm HG" wird im Winkel von 90o zur X-Achseneben die Y-Achse geschrieben

- 2290 - 2300 Der Text "Siedepunkt in o C" wird unter die X-Achse geschrieben.
- 2310 - 2370 Die X-Achse wird mit den Siedepunkten bezeichnet, siehe hierzu Ausdruck auf dem Printer
- 2380 - 2390 Text zur Ausgabe der Einzelwerte auf dem Bildschirm
- 2400 - 2460 Zeichnen der 188 Siedepunktswerte
- 2470 - 2520 Rückkehr ins Menu und Abschalten der HRG mit |exg
- 2530 - 2570 Ausgabe der Einzelwerte auf dem Bildschirm



Siedekurve von Wasser mit HRG

## Wussten Sie schon...

dass sich der Arbeitsspeicher des VC 20 mit den Speichererweiterungen verschiebt (dies macht sich vor allem beim Bildschirmspeicher bemerkbar), wenn nicht dann beachten Sie bitte die nachfolgenden Programmzeilen.

```
1 bs=4*(peek(36869)and128)+64*(peek(36869)and120)
2 fs=37888+4*(peek(36866)and128)
```

Diese beiden Programmzeilen werden an den Anfang eines Programms gesetzt. Es erfolgt dann eine automatische Anpassung der Programme, bei Verwendung der 8 k oder 16 k Erweiterung. Zu beachten ist, dass innerhalb des Programms alle POKE Befehle umgeschrieben werden. So wird z.B. aus POKE 7684,32 POKE BS+4,32 oder aus POKE 38525,3 wird POKE FA+125,3 BS berücksichtigt alle Adressen des Bildschirmspeichers, FA die Adressen der Farbauswahl.

## Der schnelle Digiplot

Im Heft 5/82 haben wir das Programm "Siedepunkt" vorgestellt, das unter anderem eine Plotsroutine für den WATANABE PLOTTER DIGILOT WX 4671 enthält. Sofern Sie schon einmal mit diesem Plotter gearbeitet haben, werden Sie feststellen, dass dieses an und für sich gute Geräet mit einer ausreichenden Reproduzierbarkeit bei vernünftigem Preis nur einen grossen Nachteil hat, dies ist seine geringe Geschwindigkeit. Dies liegt einerseits in der Hardware, andererseits aber auch in der Plotsubroutine, die vom Hersteller des Geräetes fertig in BASIC angeboten wird. Die Befehlsfolge für einen Plotbefehl ist im String X8\$ enthalten, hier setzt nun das beschriebene Programm ein. Es sucht mit der ROM-Routine JSR C187 diesen String. Ist der String gefunden, sendet der Rechner die Zeichen an den USER-PORT und von dort zum Plotter. Durch das Maschinenprogramm erfolgt der hier beschriebene Vorgang mit grosser Geschwindigkeit, die bis zu 7 mal grösser ist, als bei dem entsprechenden BASIC-Programm.

Die Subroutine wird anstelle der bisherigen Plotsubroutine verwendet. Sie laeuft aber nur auf dem CBM 8032; für den CBM 3032 müssen die DATA-Statements geaendert werden. Beim eigentlichen Zeichnen reduziert sich die Geschwindigkeit nicht, dies ist aber auch nicht unbedingt notwendig, da sich hier für das Auge ja etwas tut.

### Digiplot

```
8000 rem *** schnelle plotsubroutine      (w.venetz) ***
8010 rem --- fuer befehle grossbuchstaben verwenden ---
8020 :
8030 x8$=x9$+str$(int(y))+chr$(10):rem  *l,b,s,q,n-befehle
8040 goto8210
8050 x8$="x"+str$(int(x))+","+str$(int(y))
8060 x8$=x8$+","+str$(int(r))+chr$(10):rem axis
8070 goto8210
8080 x8$="D":goto8120:rem                * draw
8090 x8$="M":goto8120:rem                * move
8100 x8$="I":goto8120:rem                * rel draw
8110 x8$="R":rem                          * rel move
8120 x8$=x8$+str$(int(x))+","+str$(int(y))+chr$(10)
8130 goto8210
8140 x8$="P"+x$+chr$(10):goto8210:rem  * print
8150 ifpeek(634)=169andpeek(703)=96then8170:rem  * home
8160 fori=634to634+69:readx8:pokei,x8:next
8170 poke59459,255
8180 poke59457,0
8190 poke59457,128
8200 x8$="H"+chr$(10)
8210 sys634:return
8220 :
8230 data 169, 88,133, 66,169,184,133, 67
8240 data 32,135,193,160, 0,177, 68, 72
8250 data 200,177, 68,133, 66,200,177, 68
8260 data 133, 67,104,133, 68,162, 0,160
8270 data 0,196, 68,240, 32,177, 66,201
8280 data 65,144, 6,201, 91,176, 2, 9
8290 data 32, 9,128, 72,173, 77,232, 41
8300 data 2,240,249,142, 65,232,104,141
8310 data 65,232,200,208,220, 96
8320 end
```

# Maschinensprache

## ROM - Routinen

Wir wollen unter der Rubrik "Maschinensprache" wieder einmal etwas leichtere Kost bringen, damit auch dem Beginner Gelegenheit gegeben wird, in diesen interessanten Teil der Computerprogrammierung einzusteigen. Diese Reihe wird dann im naechsten Jahr in zwangloser Reihenfolge fortgesetzt.

Die meisten BASIC-Befehle bestehen aus dem eigentlichen Befehl und nachfolgenden Variablen, dies können natürlich auch Zeilennummern sein, Zahlen oder Zeichenketten, den sogenannten Parametern. Bei POKE 59468,12 lautet der eigentliche Befehl POKE, die Parameter 59468 und 12, geben an, was der Computer wohin "poken" soll. Die internen Routinen des BASIC - Interpreters können dazu benutzt werden, um auch eigenen Programmen die Fähigkeit zu verleihen, solche Parameter zu übernehmen.

Soll beispielsweise ein Programm erstellt werden, das den Cursor in eine definierte Zeile und Spalte bringt, so kann dies mit dem folgenden einfachen Maschinenprogramm erreicht werden. Um das Programm möglichst einfach zu halten, wurde auf eine Kontrolle der Eingaben verzichtet.

Versuchen Sie einmal das Programm mit folgender Eingabe ! `sys(B26)e,b`

Adresse Hexadezimal	OP-Code und Daten			Kommentar
	2001	3001	4001	
033A	JSR D676	JSR D678	JSR C8D4	Dezimalzahl nach X holen.
033D	STX F5	STX D8	STX D8	Abspeichern im Zeilenregister
033F	JSR CE11	JSR CDF8	JSR BEF5	Testen ob Komma folgt
0342	JSR D676	JSR D678	JSR C8D4	Zweite Dezimalzahl nach X holen
0345	STX E2	STX C6	STX C6	Abspeichern im Spaltenregister
0347	JMP E5DB	JMP E25D	JMP E07F	Setzen des Cursors nach den Registern

Der Text wird in die 10. Zeile und die 15. Spalte des sonst leeren Bildschirms geschrieben. Leider ist dieses kleine Programm beim CBM 8000 nicht so leicht zu realisieren.

Das Programm arbeitet mit insgesamt 4 ROM-Routinen des BASIC-Interpreters, die als Subroutinen mit dem Befehl JSR aufgerufen werden.

JSR D676  
JSR CE11  
JSR D676  
JSR E5DB

Nachfolgend nennen wir Ihnen einige weitere nützlich ROM-Routinen, die Sie einmal in Ihre Programme einbauen sollten.

2000	Adresse			Funktion
	3000	4000	8000	
D676	D678	C8D4	C8D4	Eine Zahl zwischen 0 und 255 wird in den FAC+4 und ins X-Register geholt.
DOA0	D090	C2E0	C2E0	Eine Intergerzahl zwischen -32767 und + 32767 wird in den FAC+4 (low) und FAC+3 (high) geholt.
B0-B6	5E-64	5E-64	5E-64	FAC (Fließkommaaccumulator), er entspricht dem ACC in der Maschinsprache. FAC+3 heisst beim PET B0+3=B3
CCA4	CC8B	BD84	BD84	Eine Fließkommazahl wird in den FAC geholt.
CCB8	CC9F	BD98	BD98	Ein beliebiger Parameter wird geholt.
D57B	D57D	C7B5	C7B5	Kontrolliert, ob der geholte Parameter ein String war, die Anfangsadresse des String steht in der Stringadresse und im X(low)-Y(high)-Register. Die Stringlaenge steht im ACC.
71-72	1F-20	1F-20	1F-20	Stringadresse (low/high)
DCA9	DCE3	CF8D	CF8D	Ausgabe des FAC als Fließkommazahl
DC9F	DCD9	DF83	CF83	Ausgabe von X(low) und ACC(high) als Integerzahl
CA2E	CA23	BB24	BB24	Stringausgabe (Adresse in der Stringadresse, Laenge im X-Register)

---

## Versanddaten der CBM/PET NEWS im Jahr 1983

Auch im kommenden Jahr erscheinen die "CBM/PET NEWS" im gewohnten zwei-monatlichen Intervall, erweitert um die neue Rubrik "VC 20 NEWS". Der Versand der Zeitschrift erfolgt an folgenden Tagen.

1/83	18.02.1983
2/83	15.04.1983
3/83	17.06.1983
4/83	13.08.1983
5/83	21.10.1983
6/83	09.12.1983

Haben Sie Ihr Abonnement schon erneuert? Nein? Dann aber nichts wie hin zu Ihrem naechsten Postamt und einzahlen. Es waere doch schade, wenn Sie die CBM/PET NEWS nicht mehr erhalten wuerden.

# VC 20-NEWS

## Zeichensatzänderung

Im Heft 3 und 4/82 haben wir Ihnen ein Programm zur Änderung des Zeichensatzes der Rechner CBM 3000 und 8000 vorgestellt. Heute zeigen wir Ihnen ein Programm, das eine ähnliche Funktion auch auf dem VC 20 bewirkt. Die Funktionsweise ist jedoch von den früheren Programmen ein wenig verschieden. Bei der hier gezeigten Version wird der Inhalt des Zeichen ROM's direkt in den Arbeitsspeicher geladen. Hier können nun einzelne Zeichen über DATA-Statements geändert werden. Im vorliegenden Programm sind dies die Umlaute und das "ss".

Das Programm arbeitet nur mit dem VC 20 ohne Speichererweiterung. Da sich beim VC 20 der Arbeitsspeicher mit den Speichererweiterungen verschiebt, muss dies im Programm berücksichtigt werden. Beachten Sie hierzu bitte die Rubrik "Wussten Sie schon ...".

```
2 REM DEUTSCHE ZEICHEN
5 POKE36879,25:V=1:GOTO410
9 PRINT"BITTE WARTEN"
10 POKE56,PEEK(56)-10:CLR:C=256*PEEK(52)+PEEK(51)
20 FORI=0TO2047:POKEC+I,PEEK(32768+2048+I):NEXTI
23 FORQ=0TO6:READH:F(Q)=H-0:NEXTQ
24 DATA29,122,58,59,0,45,27
30 FORI=0TO2047:READA:IFA=-1THEN200
33 POKEC+8*(I/8)+I,A:NEXTI
160 DATA36,24,36,66,126,66,66,0,36,0,66,66,66,66,60,0
162 DATA36,0,60,66,66,66,60,0,40,0,56,4,60,68,62,0
164 DATA36,0,66,66,66,70,58,0,28,36,68,72,68,82,76,64
166 DATA36,24,36,66,66,36,24,0,-1
200 FORX=1TO6:READA,B:FORY=0TO7
220 POKEC+Y+A*8,PEEK(32768+2048+Y+B*8)
230 NEXTY:NEXTX
250 DATA25,26,26,25,89,90,90,89,60,59,62,58
300 POKE36869,253:POKE36866,PEEK(36866)OR128:CLR
410 PRINT"#####"
420 PRINT"#+1234567890+-#/"
430 PRINT"#####QWERTYUIOP3#1#"
440 PRINT"#####ASDFGHJKL:;=#/"
450 PRINT"#####ZXCVBNM,./#/"
460 PRINT"#####"
520 PRINT"#+!"CHR$(34)"#%&'()@+|/#####"
530 PRINT"#####0_[]|/[]-#####"
540 PRINT"#####+*[]|/[]=#####"
550 PRINT"#####+*-X!^O?#####"
560 PRINT"#####":IFVTHEN9
570 IFVTHEN9
580 NEW
```

READY.





```

220 IFUZ>10THEN120
230 FC=8:FORGS=1T012
240 POKE36879,FC:FC=FC+17
250 FORLS=15T00STEP-1
260 POKE36878,LS:POKE36875,200
270 FORN=1T0100:NEXT
280 NEXT:NEXT
290 POKE36879,236
300 RESTORE
310 GOTO120
500 DATA 163,4,187,4,163,4,187,4,201,4,175
501 DATA 4,167,4,163,6,0,2,201,6,201,2,195
502 DATA 4,187,4,183,10,0,2,195,4,175,4,167
503 DATA 4,175,4,187,4,183,4,179,4,183,6,0
504 DATA 2,195,6,195,2,187,4,175,4,163,10,0
505 DATA 2,163,2,187,4,183,4,187,4,201,4
506 DATA 175,4,167,4,163,6,0,2,201,6,201,2
507 DATA 203,4,201,4,175,10,0,2,175,4,195,4
508 DATA 187,4,183,4,175,4,175,4,167,4,163
509 DATA 6,163,2,203,6,195,2,175,4,183,4
510 DATA 187,2,163,1,163,1,163,2,163,2,163
511 DATA 2,0,2
512 DATA 163,4,187,4,183,4,187,4,201,4,175
513 DATA 4,167,4,163,6,0,2,201,6,201,2,195
514 DATA 4,187,4,183,10,0,2,195,4,175,4,167
515 DATA 4,175,4,187,4,183,4,179,4,183,6,0
516 DATA 2,195,6,195,2,187,4,175,4,163,10,0
517 DATA 2,163,2,187,4,183,4,187,4,201,4
518 DATA 175,4,167,4,163,6,0,2,201,6,201,2
519 DATA 203,4,201,4,175,10,0,2,175,4,195,4
520 DATA 187,4,183,4,175,4,175,4,167,4,163
521 DATA 6,163,2,203,6,195,2,175,4,183,4
522 DATA 187,2,0,2,163,2,175,.5,183,.5,187
523 DATA 2,187,1,187,1,187,2,0,0

```

READY.

## Wussten Sie schon...

dass Sie beim VC 20 mit POKE 650,128 alle Tasten mit der REPEAT-Funktion ausstatten. Ausgeschaltet wird REPEAT mit POKE 650,0.

### IMPRESSUM

**Verlag, Redaktion, Inserate**  
Mikro + Kleincomputer Informa Verlag AG  
Seeburgstrasse 12, 6006 Luzern

**Postanschrift:**  
**Postfach 1401, CH-6000 Luzern 15**

Telefon 041 - 31 18 46, Tx 72 227 (dcl ch)  
Postcheck-Konten:  
Luzern 60 - 27181  
Stuttgart 3786-709 (BLZ 600 100 70)  
Wien PSK 7975.035

**Verlagsleitung**  
Hans-Jürgen Ottenbacher

**Redaktion**  
Heinz Kastien

© 1982 by Mikro + Kleincomputer Informa Verlag AG, Luzern, aber Speicherung in Datenverarbeitungsanlagen für den eigenen Gebrauch erlaubt. Nachdruck, auch auszugsweise, sowie Vervielfältigungen jedwelcher Art nur mit schriftlicher Genehmigung des Verlages und unter voller Quellenangabe.

**Bezug:** Jahresabonnement Fr. 48.-, Ausland DM 55.-/öS 400 für sechs Hefte pro Jahr. Abonnenten von Mikro + Kleincomputer bezahlen für das CBM/PET NEWS-Abo nur Fr. 24.-, Ausland DM 29.-/öS 200. In den genannten Abonnementspreisen sind sämtliche Nebenkosten, inkl. Porto, enthalten. Die Kündigung ist jeweils 8 Wochen vor Ablauf des laufenden Bezuges möglich, die Abonnementsgebühr ist nach Erhalt der Rechnung fällig.

### Manuskripte

Mit der Zustellung von Manuskripten anerkennt Der Autor die Copyrightbestimmungen des Verlages. Mit der Annahme von Manuskripten durch die Redaktion und der Autor-Honorierung durch den Verlag hat dieser das Recht zur Veröffentlichung der entsprechenden Beiträge in anderen verlagseigenen Publikationen und zur Übersetzung in andere Sprachen erworben. Für die Veröffentlichung wird keine Gewähr oder Garantie übernommen, auch nicht dafür, dass die verwendeten Schaltungen, Firmennamen und Warenzeichnungen usw. frei von Schutzrechten Dritter sind, die Verwendung der Informationen erfolgt auf eigenes Risiko.

Printed in Switzerland

**Jetzt  
erhältlich!**

**Das unentbehrliche**

**CBM-Handbuch**



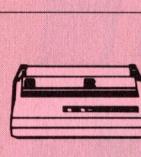
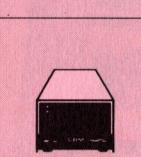
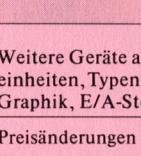
...für jeden ernsthaften Commodore-Benutzer, der seinen CBM-Rechner noch besser kennenlernen möchte und/oder auch in Maschinensprache damit arbeiten will ist dieses Buch eine wahre Fundgrube. Darin werden sämtliche CBM-"Spezialitäten", neue Adressen, Funktionen und Möglichkeiten der CBM-Betriebssysteme 3000 und 4000/8000 umfassend und eingehend erklärt.

Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401  
CH-6000 Luzern 15

Gegen Voreinzahlung von SFr./DM 49.-- auf eines unserer Postkonten Luzern 60-27181, Stuttgart 3786-709 oder Wien 7975.035, Stichwort "Commodore-Analyse" senden wir Ihnen das 224 Seiten starke Buch umgehend zu.

# Weil Sie trotz Rotstiftpolitik Ihr Computer Beschaffungsprogramm voll durchziehen können, kommen Sie gleich zu Commodore.



	Modell	Leistungs-Merkmale	Preis: Fr.
	<b>CBM 8032</b>	Zentraleinheit inkl. Bildschirm und Tastatur, 32 kByte RAM (frei verfügbar), 18 kByte ROM, Basic 4, Maschinensprache	3475.-
	<b>CBM 8096</b>	Zentraleinheit - wie 8032, jedoch weitere 64 kByte RAM zuschaltbar. Betriebssystem LOS 96	4675.-
	<b>CBM 8050</b>	Floppy Disk Speichereinheit - Speichervolumen 1 Megabyte (500 kByte pro Diskette), 2 Laufwerke, DOS 2	3975.-
	<b>CBM 8250</b>	Floppy Disk Speichereinheit - Speichervolumen über 2 Megabyte pro Diskette, 2 Laufwerke, DOS 2,7	5250.-
	<b>CBM 8023 P</b>	Matrixdrucker - Druckbreite 132 Zeichen/Zeile, 150 Zeichen/Sek.	2450.-
	<b>CBM 9060</b> <b>9090</b>	Hard-Disk Speichereinheit - 5 bzw. 7,5 Megabyte Speichervolumen, Übertragungsgeschwindigkeit 5 Megabyte/Sek.	7550.- 8950.-

Zunehmender Kostendruck zwingt immer mehr Unternehmen, den Rotstift anzusetzen. Auf die Vorteile der Datenverarbeitung wird verzichtet. Muss das sein? Dank seines unerreicht günstigen Preis/Leistungs-Verhältnisses kann sich jedes Unternehmen den Commodore-Tischcomputer leisten. Die Übersicht beweist es. Kein Wunder also, dass Commodore mit 7000 der meistgekauften Tischcomputer in der Schweiz ist. Ausschlaggebend für diesen Erfolg sind allerdings auch das reichhaltige Softwareangebot und das dichte Fachhändler-Netz.

Ihr Fachhändler: **Baar** Logon AG, 042/31 44 80 **Basel** Computer-Shop, 061/35 31 14 · **BD-Electronic**, 061/35 36 37 · **Kubli + Eicher AG**, 061/35 05 17 · **SYSAG AG**, 061/39 25 25 · **Geiger-Microcomputer**, 061/44 13 13 **Bern** **Radio TV Steiner AG**, 031/22 20 61/62 · **Computerland AG**, 031/24 25 54 **Biel** **EIM Computer AG**, 032/23 15 88 **Clarens** **Mafjoly SA**, 021/62 12 12 **Fontainemelon** **Urs Meyer Electronic**, 038/53 43 43 **Fribourg** **Jean-Claude Labastrou**, 037/22 12 22 · **Sovitel SA**, 037/24 22 82 **Thônex-Genève** **Gesmarco SA**, 022/49 88 44 **Gossau** **Pius Schäfler**, 071/85 45 66 **Huttwil** **Compu-Life Rüfenacht AG**, 063/72 11 12/13 **Interlaken** **H.U. Gurtner**, 036/22 10 21 **Lausanne** **Radio TV Steiner SA**, 021/23 11 77 · **Computer Shop**, 021/37 60 37 **Lugano** **Luigi Chiodoni**, 091/23 23 09 · **Datanel SA**, 091/23 45 44 **Luzern** **Dialog Computer Treuhand AG**, 041/31 53 33 · **Schweizer Computer-Club**, 041/31 45 45 · **Helfenstein + Bucher AG**, 041/22 13 43 **Mellingen** **Instant-Soft AG**, 056/91 20 21 **Neuchâtel** **Bolomey Organisation**, 038/25 97 38/39 **Rorschach** **Bruno Müller**, 071/41 00 31 **Schaffhausen** **PIM Systems**, 053/4 54 50 **Sion** **Sphere Corporation** **Clausen SA**, 027/22 68 14 **Thun** **HMB electronic**, 033/22 66 88 **Visp** **Peter Nellen**, 028/46 41 21 **Wettingen** **Elbatex AG**, 056/27 01 27 **Wetzikon** **CT Computer-Team AG**, 01/932 31 31 **Winterthur** **Nowak AG**, 052/22 08 03 **Wohlen** **SYSAG AG**, 057/22 36 50 **Zürich** **Erhard Wipf AG**, 01/221 21 00 · **Microspot AG**, 01/241 20 30 · **Fürrer-Büro-computer AG**, 01/42 05 05 · **Logon AG**, 01/62 59 22

Weitere Geräte aus dem Commodore Lieferprogramm: weitere Zentraleinheiten, Typenrad- und Matrix-Drucker sowie Plotter, hochauflösende Graphik, E/A-Steuerung sowie ein reichhaltiges Softwareangebot.

Preisänderungen vorbehalten.

**Ihr freundlicher  
Computer-Kontakt-Coupon**

Ich interessiere mich für  
CBM Tischcomputer

Name: \_\_\_\_\_

Adresse: \_\_\_\_\_

Einsenden an: Commodore AG  
Aeschenvorstadt 57  
4010 Basel

