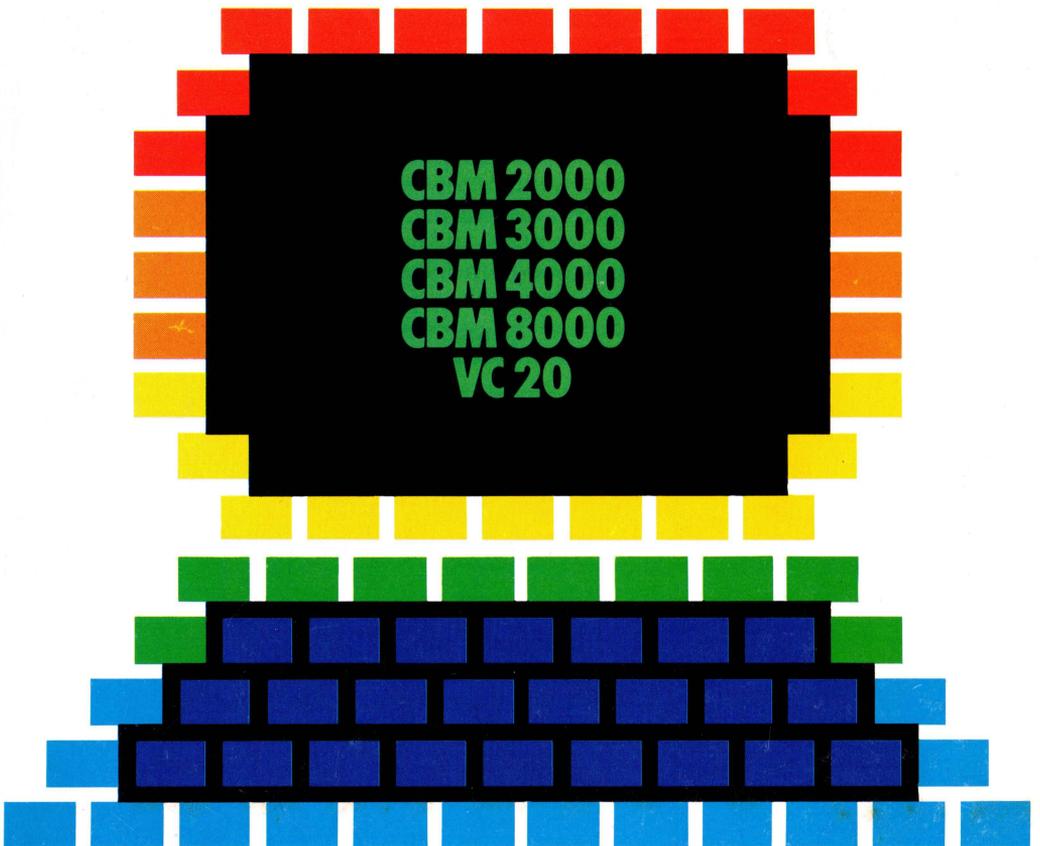


Karl-Heinz Heß

BASIC-Programme für CBM/VC 20-Computer



BASIC-Programme für CBM/VC 20-Computer

Karl-Heinz Heß

**BASIC-Programme
für CBM/
VC 20-Computer**

Verlag Markt & Technik

Hess, Karl-Heinz:
BASIC-Programme für CBM/VC 20-Computer /
Karl-Heinz Hess. — Haar bei München: Verlag
Markt und Technik, 1983.
(Computer persönlich)
ISBN 3-922120-28-8

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

ISBN 3-922120-28-8

© 1983 by Markt & Technik, 8013 Haar bei München
Alle Rechte vorbehalten
Einbandgestaltung: Grafikdesign Heinz Rauner
Druck: Color-Offset, 8000 München 70
Printed in Germany

Vorwort

Ein Personal Computer kann zur Lösung spezieller beruflicher Aufgabenstellungen, zur Bewältigung forschender und lehrender Tätigkeiten genauso eingesetzt werden wie zur Entspannung und Unterhaltung. Die schier unerschöpfliche Vielfalt breiter Anwendungsmöglichkeiten verursacht demnach eine große Nachfrage nach nützlichen und allgemein verständlichen Programmen.

Dieses Buch soll sowohl dem Computerneuling als auch dem Basic-Profi die programmtechnische Lösung bestimmter Probleme aufzeigen. Dabei beschränkt sich die Problemlösung nicht etwa auf die Darstellung des Programmlistings. Die verschiedenen Aufgabenstellungen werden genau analysiert, allgemeingültige Lösungswege erarbeitet und in CBM-Basic konvertiert. Alle Programme sind darüber hinaus ausführlich dokumentiert und anwendbar für die Serien CBM 2000, 3000, 4000 und 8000. Sie wurden im praktischen Einsatz sorgfältig getestet. Einige Programme laufen auch auf VC 20 und anderen basic-programmierbaren Rechnern, wobei etwaige Programmanpassungen näher beschrieben sind. Die Funktionsweise und Bedienung der einzelnen Programme wird durch eine umfassende Erläuterung des jeweils zugrundeliegenden Sachverhalts auch für den Laien verständlich. Dem Profi, der ähnliche Aufgabenstellungen, wie in diesem Buch beschrieben, zu bewältigen hat, kann eine Orientierung an den hier vorgegebenen Lösungswegen durchaus hilfreich sein.

So versteht sich dieses Buch nicht nur als Sammlung und Nachschlagewerk fertiger Basic-Programme, die einmal eingetippt, ständig zur Lösung eines bestimmten Problems dienen. Bei der Programmierung wurde auch besonderes Augenmerk darauf verwendet, dem Benutzer durch einfache Parametersteuerungen zu ermöglichen, die abgebildeten Programme seinen spezifischen Anforderungen und darüber hinaus der eigenen Hardware-Konfiguration bequem anzupassen. In einigen Artikeln werden nützliche Tips rund um das Programmieren erläutert.

Sinntal, im Januar 1983

Karl-Heinz Heß

Inhaltsverzeichnis

Etiketten-Beschriftung	11
Problembeschreibung	11
Problemlösung, Bedienungsanleitung	16
Programmbeschreibung	18
Lotto: 6 aus 49/7 aus 38	23
Programmbeschreibung	26
Wie kann eine kleine Programmroutine in ein bestehendes größeres Programm eingebaut werden	27
Hardcopy für alle CBM-Typen	31
Problembeschreibung	32
Problemlösung	32
Bedienungsanleitung	34
Programmbeschreibung	37
Kopfrechen-Training	41
Bedienungsanleitung	43
Programmbeschreibung	45
Bildschirmrahmen	49
Für 40- und 80-Zeichen-Bildschirm	49
Römische Zahlen	55
Bedienungsanleitung	57
Programmbeschreibung	60
Kugelabschnitt	63
Bedienungsanleitung	65
Programmbeschreibung	67
Der Zeichengenerator des CBM	71
»Jogging« für 40- und 80-Zeichen-Bildschirm	71
Entwurf von Sonderzeichen durch Nadelansteuerung	75
Für CBM-Printer 3022	75
Problembeschreibung	76

Problemlösung	77
Bedienungsanleitung	78
Programmbeschreibung	81
Code-Umwandlung	85
8-Kanal-BCD in dezimal	85
Problembeschreibung	86
Problemlösung	87
Bedienungsanleitung	88
Programmbeschreibung	93
Horizontale Balkendiagramme	97
Problembeschreibung	100
Problemlösung	101
Bedienungsanleitung	103
Programmbeschreibung	106
Laufbandanzeige	109
Bedienungsanleitung	121
Besonderheiten bei CBM-Rechnern mit Betriebssystem Basic 4.0 ...	122
Programmbeschreibung	123
Banküberweisungen ausfüllen	129
Problembeschreibung	131
Problemlösung	134
Bedienungsanleitung	136
Programmbeschreibung	142
Ohne Worte	147

Etiketten-Beschriftung

Etiketten-Beschriftung

Die Fülle und die Vielfalt der bereits auf dem Markt befindlichen Computer-Programme spiegelt die breiten Anwendungsmöglichkeiten eines Personal Computers wider. Seit der Markteinführung dieser kleinen Rechner kann erstmals jeder seinen eigenen Computer haben. Für den Anwender ist dabei entscheidend, was man alles mit einem solchen »Gerät« machen kann. Charakteristisch für ein Personal Computer-System sind die Unabhängigkeit und Praxisnähe, die dem Benutzer ein möglichst einfaches und eigenständiges Arbeiten erlauben. Denn nicht immer findet man unter den bereits verfügbaren Programmen seine spezielle Problemlösung. Oftmals aber kann der User mit minimalem Aufwand seinen Computer, wenn er schon einen besitzt, leichte Routinearbeiten erledigen lassen, deren Durchführung er ursprünglich gar nicht geplant hatte.

Wenn man also seine Adreß-Etiketten noch per Schreibmaschine und Lager-schildchen mit Schablonen schreibt, so kann das nachfolgende Programm rasche Abhilfe schaffen und dafür sorgen, daß auch in Zukunft gut lesbare Aufkleber mit einem Bruchteil des bisherigen Aufwandes zuverlässig und fehlerfrei per Computer beschriftet werden. Die dazu nötige Hardware-Konfiguration umfaßt mindestens einen Rechner mit Kassettenlaufwerk oder Floppy-Disk und einen angeschlossenen Drucker.

Problembeschreibung

Eine Etikettenbeschriftung wird einmal erfaßt und soll dann beliebig oft ausgedruckt werden können. Ein Programm, welches obigen Anforderungen gerecht werden soll, muß folgende Voraussetzungen erfüllen:

- Darstellung einer Erfassungsmaske, um die Etikettenbeschriftung einmalig per Tastatur in den Rechner zu tippen;
- Fehlerabfrageroutine mit Überprüfung der Eingaben und Korrekturmöglichkeit durch den Benutzer;
- Abfrage, wie viele Aufkleber mit gleichem Text beschriftet werden sollen;
- Druckeransteuerung;
- Bedrucken der Etiketten, dabei Protokollieren der bereits beschrifteten Schildchen auf dem Bildschirm;
- parametergesteuerte Angleichung des Programmes an die verschiedenen Etikettengrößen durch Festlegung der Anzahl beschrifteter Zeilen und Zeilen gesamt pro Etikett;
- Wiederholungs- und Abbruchmöglichkeit des Programmes durch einfachen Tastendruck.

◆◆◆ ETIKETTENFORMAT ◆◆◆

```
100 REM FESTSTELLUNG DER
110 REM DRUCKERINDIVIDUELLEN
120 REM FORMATEINTEILUNG
130 :
140 OPEN1,4
150 FORI=1TO8:REM ZEILEN PRO ETIKETT
160 FORJ=1TO35:REM ZEICHEN PRO ZEILE
170 PRINT#1,"X";
180 NEXTJ
190 PRINT#1
200 NEXTI
210 PRINT#1:CLOSE1
```

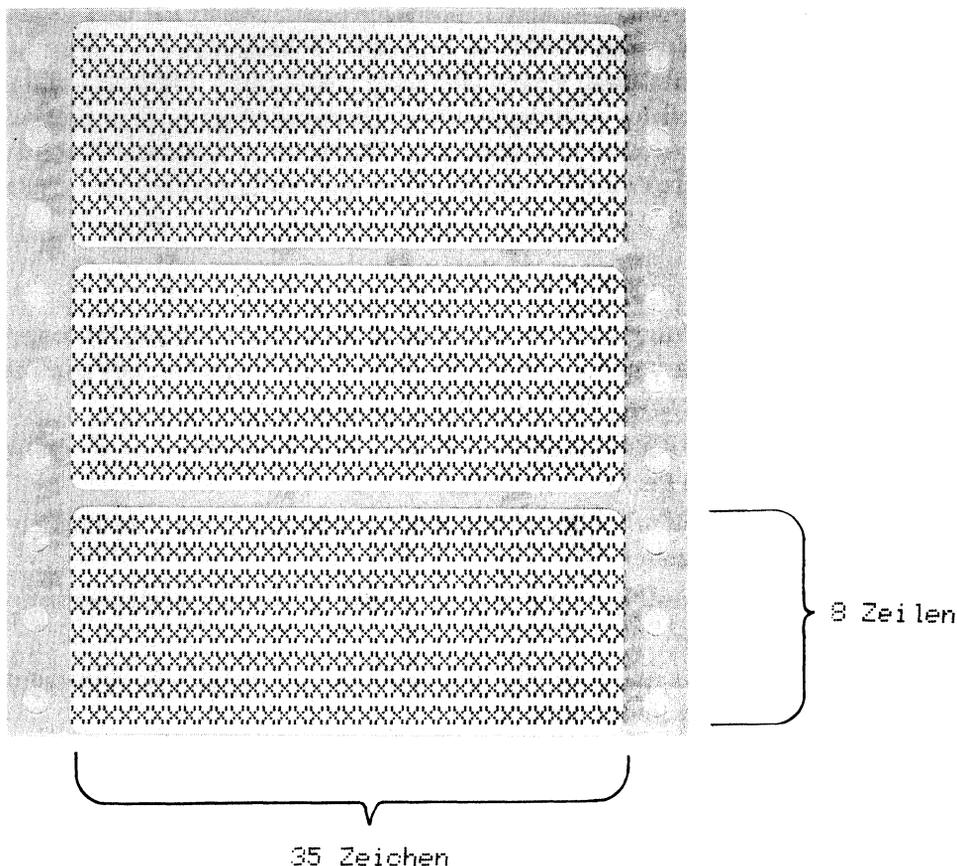


Bild 1: Etikettenformat

Wenn man kein EDV-Lineal besitzt, kann die Bestimmung der Formatgröße (Zeichen pro Zeile, Zeilen pro Etikett) einige Schwierigkeiten bereiten. Hier kann man sich ganz einfach weiterhelfen, indem man ein Muster-Etikett mit X-Zeichen bedruckt und die Anzahl der Zeichen und Zeilen abzählt.

Diese Methode kann auch zur druckerindividuellen Bestimmung der Spaltenbreite in Formularen erfolgreich eingesetzt werden, da in der Praxis viele Drucker von der genormten Zeileneinteilung (10, 12 oder 15 Pitch) abweichen.

◆ ETIKETTEN-BESCHRIFTUNG ◆

```

100 REM * ETIKETTENBESCHRIFTUNG *
110 REM *****
120 :
130 POKE59468,12:REM* GRAFIK/GROSS
135 REM ZEILE 130 GILT NICHT FUER VC20
140 :
150 REM* VARIABLENBELEGUNG *
160 CL#=CHR$(157):REM* CURSOR LINKS
170 MB=8:REM* MAX. ANZAHL BESCHRIFTETER ZEILEN
180 IS#=CHR$(32)+CHR$(32)+CHR$(160)+CL#+CL#+CL#
190 I=0:U:I:AE=I
200 ZE=12:REM* ZEILEN PRO ETIKETT (6LPI) *
210 DIMA$(MB):REM* TABELLE DER ZEILEN *
220 :
230 REM* ERFASSUNGSMASKE *
240 PRINT"ERFASSEN ETIKETT:■"
250 PRINT"0"
260 FORI=1TOMB
270 PRINTI;" "IS#;:INPUTA$(I)
280 NEXTI
290 :
300 REM* UEBERPRUEFUNG DER EINGABEN *
310 PRINT"UEBERPRUEFUNG DER EINGABEN OK ? (J/N)■"
320 GETX#:IFX#=""THEN320
330 IFX#="N"THEN:RUN
340 IFX#="J"THEN380
350 GOTO320
360 :
370 REM* ABFRAGE, WIEVIEL ETIKETTEN *
380 PRINT"0"
390 FORI=1TOMB
400 PRINTI;CL#". 0"A$(I)
410 NEXTI
420 PRINT"000"
430 INPUT"ANZAHL ETIKETTEN ";AE
440 IFAE=0THEN240
450 PRINT"ANZAHL GEDRUCKTER ETIKETTEN:"
460 :
470 REM*DRUCKER*
480 OPEN1,4
490 FORU=1TOAE
500 :
510 REM* LEERZEILEN *
520 FORUU=ZETO(MB-2)STEP-1:PRINT#1:NEXTUU
530 :
540 FORI=1TOMB
550 PRINT#1,CHR$(1)A$(I)
560 PRINTU:PRINT"0";
570 NEXTI:PRINT#1:PRINT#1:NEXTU

```

```
580 CLOSE1
590 :
600 PRINT"NOCHMAL GLEICHES ETIKETT ? (J/N)"
610 GETX$:IFX$=""THEN610
620 IFX$<>"N"THENPRINT"OK":GOTO380
630 :
640 PRINT"OK AUSDRUCK BEEENDET! ■":END
```

Problemlösung, Bedienungsanleitung

Das abgebildete Basic-Programm »Etiketten-Beschriftung« erfüllt alle genannten Voraussetzungen und läuft ohne Modifikation auf allen CBM 2000, 3000, 4000 und 8000 mit angeschlossenem Drucker. Durch Entfernen der Zeile 130 ist es auch auf dem VC 20 lauffähig.

Nach Einladen und Starten des Programmes mit »RUN« wird der Computer in Groß-/Grafikmodus geschaltet. Anschließend erfolgt das Anlegen der Variablen-tabelle und Dimensionieren der Etikettenzeilen. Jetzt kommt die Eingabemaske auf den Bildschirm. Das Programm zeigt die Durchnummerierung der zu beschriftenden Zeilen an, um dem Benutzer eine Orientierungshilfe bei der Erfassung zu geben. Die Erfassungsroutine erlaubt die Eingabe von »Carriage Return« (CHR\$(13)) und interpretiert diese als Leerzeile, d.h., der Bediener kann Leerzeilen an beliebigen Stellen erfassen. Nachdem die Eingabemaske ausgefüllt ist, fragt das System, ob alle Eingaben »ok« sind. Beantwortet man diese Frage durch Betätigen der Taste »N« für »Nein«, so startet das Programm erneut. Sind alle Eingaben richtig, dann wird dies durch Drücken der Taste »J« bestätigt, und die Etiketten-Beschriftung erscheint invertiert auf dem Bildschirm.

Gleichzeitig fragt der Computer nach der Anzahl der Etiketten, die mit dem erfaßten Text beschriftet werden sollen. Ratsam ist, am Anfang hier zunächst eine »1« einzutippen, um festzustellen, ob die vorher eingespannten Endlos-Etiketten auch an richtiger Stelle bedruckt werden. Gegebenenfalls müssen die Schildchen durch horizontales Verschieben und vertikalen Vorschub von Hand genau justiert werden. Das Programm eröffnet nun den Drucker und druckt den erfaßten Text in Sperrschrift (verdoppelte Matrix, CHR\$(1)) auf die Etiketten. Dabei stellt der Computer zur Fortschrittskontrolle durch den Benutzer die Anzahl der bereits bedruckten Schildchen auf dem Bildschirm dar. Nach Beendigung wird das Druckfile geschlossen und das System fragt: »Nochmal gleiches Etikett? (J/N)«. An dieser Stelle kann durch Tastendruck entschieden werden, ob noch mehr Etiketten mit dem gleichen Text bedruckt werden sollen oder Programmbeendigung erfolgen soll.

MARKT & TECHNIK
VERLAGSGES. MBH

HANS-PINSEL-STR. 2

8013 HAAR BEI MUENCHEN

MARKT & TECHNIK
VERLAGSGES. MBH

HANS-PINSEL-STR. 2

8013 HAAR BEI MUENCHEN

MARKT & TECHNIK
VERLAGSGES. MBH

HANS-PINSEL-STR. 2

8013 HAAR BEI MUENCHEN

MARKT & TECHNIK
VERLAGSGES. MBH

HANS-PINSEL-STR. 2

8013 HAAR BEI MUENCHEN

Bild 2: Adreß-Etiketten

Programmbeschreibung

Die Flexibilität der verwendeten Basic-Befehle macht es möglich, daß dieses Programm auf allen CBM- und VC 20-Computern ohne Änderungen läuft. Außerdem dürfte es nicht sehr schwierig sein, durch kleine Modifikationen diese »Etiketten-Beschriftung« auch auf anderen basicprogrammierbaren Kleincomputern zum Laufen zu bringen.

In Zeile 130 erfolgt die Umschaltung des Zeichengenerators. Die Zeilen 150 bis 210 zeigen die Deklaration und Belegung der benötigten Variablen und das Dimensionieren der Etikettenbeschriftung. Während sich die Erfassungsmaske in den Zeilen 230 bis 280 befindet, ist die Korrekturmöglichkeit in 300 bis 350 angegeben. Die Frage nach der »Anzahl Etiketten« stellt der Computer in den Zeilen 370 bis 450. Die Programmzeile 480 eröffnet den angeschlossenen Drucker mit der Geräteadresse 4. Nach Ausgabe der Leerzeilen zwischen den einzelnen Schildchen in 510, 520 erfolgt in der Schleife zwischen 540 und 570 die eigentliche Etikettenbeschriftung. Anschließend wird das Druckerfile wieder in Zeile 580 geschlossen und das Programm bietet die Möglichkeit, den Druckvorgang zu wiederholen oder zum Programmende zu gehen.

ERFASSEN ETIKETTEN

```
1 :? MARKT & TECHNIK
2 :? VERLAGSGES. MBH
3 :?
4 :? HANS-PINSEL-STR. 2
5 :?
6 :? 8013 HAAR BEI MUENCHEN
7 :?
8 :?
```

BEINGABEN OK ? (J/N)

```
1. MARKT & TECHNIK
2. VERLAGSGES. MBH
3.
4. HANS-PINSEL-STR. 2
5.
6. 8013 HAAR BEI MUENCHEN
7.
8.
```

```
ANZAHL ETIKETTEN ? 50
ANZAHL GEDRUCKTER ETIKETTEN: 21
NOCHMAL GLEICHES ETIKETT ? (J/N)
```

AUSDRUCK BEENDEN

Durch Verändern der Parameter »MB« in Zeile 170 (maximale Anzahl der zu beschriftenden Zeilen pro Etikett) und »ZE« in Zeile 200 (Zeilen pro Etikett, d.h. von Etikettenanfang bis Etikettenanfang bei 6 lines per inch) können Etiketten der verschiedensten Formate bedruckt werden. Entfernt man in Zeile 550 den Befehl CHR\$(1), so kann statt Breitschrift die Normalschrift angewendet werden.

Lotto: 6 aus 49/7 aus 38

Lotto: 6 aus 49 / 7 aus 38

Nach der Einführung des Mittwochs-Lotto »7 aus 38« haben Lottospieler die Möglichkeit, zweimal in der Woche zu gewinnen. Das folgende Basic-Programm unterstützt die Spieler bei Normal-Lotto »6 aus 49« und Volks-Lotto »7 aus 38« durch Findung von Zahlenkombinationen, die dann nur noch auf den Lottoscheinen anzukreuzen sind.

*** **L O T T O** ***

TASTE **F1** = NORMAL-LOTTO **6 AUS 49**

TASTE **F2** = VOLKS-LOTTO **7 AUS 38**

BITTE WAEHLN !!! **< OK >**

1 . ZIEHUNG !

LOTTO - 6 AUS 49

< 1 >

< 2 >

< 33 >

< 39 >

< 45 >

< 49 >

F3 = NOCHMAL **F4** = ANFANG **F5** = ENDE

1 . ZIEHUNG !

LOTTO - 7 AUS 38

< 2 >

< 8 >

< 11 >

< 25 >

< 38 >

< 35 >

< 38 >

N = NOCHMAL **A** = ANFANG **E** = ENDE

V I E L G L U E C K ! ! !

UND - GEBEN SIE IHREN LOTTO-
SCHEIN RECHTZEITIG AB.

Dieses, auf einem CBM 3001 geschriebene Programm läuft ohne Modifikationen auf allen CBM-Rechnern und VC 20. Nach Einladen und Starten des Programmes mit »RUN« erscheint die Auswahlmaske auf dem Bildschirm. Der Bediener kann sich durch Drücken der Taste »1« für Normal-Lotto »6 aus 49« oder durch Betätigen der Taste »2« für Mittwochs-Lotto entscheiden. In beiden Fällen erzeugt das Programm über einen Zufallsgenerator die jeweils erforderliche Anzahl Zahlen und gibt diese aufsteigend sortiert auf dem Bildschirm aus. Der Benutzer erhält Vorschläge, wie er seinen nächsten Lottoschein ausfüllen und gewinnen kann (ohne Gewähr!).

Durch Drücken der Taste »N« erzeugt das Programm die nächste Zahlenkombination; wird die Taste »A« gedrückt, dann erscheint die Auswahlmaske auf dem Bildschirm. Betätigt der Bediener die Taste »E«, erscheint ein sehr wichtiger Hinweis auf dem Bildschirm und das Programm wird beendet.

◆◆◆ L O T T O ◆◆◆

```

100 REM **** L O T T O ****
110 REM =====
120 REM <<< 6 AUS 49 >>>
130 REM UND
140 REM <<< 7 AUS 38 >>>
150 REM
160 REM =====
170 REM
180 :
190 DIMK(7), I, J, Z, A1, A2, M, KK, T, V$, X$
200 :
210 PRINT"□";TAB(8);"***  L O T T O  ***"
220 PRINT"□□□□":Z=0
230 PRINT"□□□TASTE 1'□ = NORMAL-LOTTO 6 AUS 49 □"
240 PRINT"□□□TASTE 2'□ = VOLKS-LOTTO 7 AUS 38 □"
250 PRINT"□□□□□□□□□□□□□□□□□□□□BITTE WAEHLN !!!";
260 :
270 GETX$: IFX$="1" THEN330
280 IFX$<"2" THEN270
290 PRINT" □ < OK > □"
300 REM 7 AUS 38
310 A1=7:A2=38
320 GOTO370
330 REM 6 AUS 49
340 PRINT" □ < OK > □"
350 A1=6:A2=49
360 :
370 REM *ERMITTLUNG*
380 FORI=1TOA1
390 K(I)=INT(1+(A2*RNDRND(0)))
400 NEXTI
410 :
420 GOSUB690
430 :
440 FORI=1TOA1:FORJ=1TOA1
450 IFI=JTHEN470
460 IFK(I)=K(J)THEN:GOTO370
470 NEXTJ,I
480 :
490 REM *BILDSCHIRM*
500 Z=Z+1:PRINT"□";TAB(10);Z;". ZIEHUNG !"
510 PRINT"□□";TAB(9);"LOTTO -";A1;" AUS";A2
520 PRINT"□□"
530 FORI=1TOA1
540 PRINT"<";K(I);">□□□";
550 NEXTI
560 PRINT
570 PRINT"□'N'□ = NOCHMAL □'A'□ = ANFANG □'E'□ = ENDE"
580 :

```

```

590 GETX$: IFX$="N" THEN 370
600 IFX$="A" THEN 210
610 IFX$<>"E" THEN 590
620 :
630 PRINT"DUO":V$=" V I E L   G L U E C K ! ! ! "
640 PRINTV$
650 PRINT"DUOJUND - GEBEN SIE IHREN LOTTO-"
660 PRINT"DUOISCHIN RECHTZEITIG AB."
670 END:REM *ENDE*
680 :
690 REM *UNTERPROGRAMM SORT*
700 M=A1
710 M=INT(M/2): IFM=0 THEN:RETURN
720 KK=A1-M:J=1
730 I=J
740 IFK(I)<=K(I+M) THEN:GOTO770
750 T=K(I):K(I)=K(I+M):K(I+M)=T:I=I-M
760 IFI>=1 THEN:GOTO740
770 J=J+1: IFJ>KK THEN:GOTO710
780 GOTO730

```

Programmbeschreibung

In Zeile 190 wird die Variablentabelle angelegt. Das Darstellen des Auswahlmenüs, die Tastenabfrage und die Verzweigung zum jeweilig gewünschten Programmteil geschieht in den Zeilen 200 bis 350. Der Zufallsgenerator wird in den Zeilen 370 bis 400 aktiviert. Anschließend werden die Zahlen aufsteigend sortiert (Zeilen 690 bis 780) und in den Zeilen 440 bis 470 auf Redundanz geprüft. Nach erfolgreicher Prüfung stellt das Programm die Zahlenkombination auf dem Bildschirm dar (Zeilen 490 bis 570) und erwartet eine neue Bedieneranweisung (Zeilen 590 bis 610).

Noch ein kurzer Tip: Das Unterprogramm »SORT« (ab Zeile 690) sortiert Zahlen in aufsteigender Reihenfolge und kann unverändert in andere Problemlösungen eingebaut werden. Variablenbelegung beachten!

Wie kann eine kleine Programmroutine in ein bestehendes größeres Programm eingebaut werden?

Bei Routinen, bestehend aus wenigen kurzen Programmzeilen, ist es wohl am einfachsten, die Befehle über die Tastatur einzutippen. Handelt es sich aber um komplizierte Programmteile, so kann man sich die Eigenschaften des Basic-Editors zunutze machen.

Hierzu kann der Bildschirm des CBM-Computers als Zwischenspeicher für kurze Programme genutzt werden. Zunächst lädt man ein Kurzprogramm in den Programmspeicher und listet es auf den Bildschirm. Jetzt lädt man das große Programm in den Speicher. Dabei muß darauf geachtet werden, daß der Bildschirm nicht zu weit nach oben abgerollt wird, da sonst die erste Zeile des gelisteten Programmes verschwindet. Anschließend bringt man den Cursor durch Betätigen der »HOME«-Taste an den Bildschirmanfang und übernimmt durch »RETURN« jede dargestellte Programmzeile in den Programmspeicher.

Zwei Punkte müssen jedoch besonders berücksichtigt werden. Erstens ist auf die richtige Numerierung des Kurzprogrammes zu achten, wenn gewährleistet sein soll, daß die Routine folgerichtig eingebaut oder angehängt wird, ohne den Inhalt des Hauptprogrammes unbeabsichtigt zu überschreiben. Wenn man bei der Erstellung von Programmen die Befehlsabkürzungen benutzt, kann es vorkommen, daß die Zeilen einschließlich der Numerierung im Listing länger als 80 Zeichen sind. Basic-Programmzeilen dürfen bei der Eingabe aber nicht länger als 80 Zeichen sein. Bei der Übernahme durch »RETURN« ist dieser Punkt sehr wichtig, da der Basic-Editor überhängende Zeichen einfach abschneidet.

Hardcopy für alle CBM-Typen

Hardcopy für alle CBM-Typen

Aus verschiedenen Gründen kann es oftmals notwendig werden, den Bildschirminhalt auf einem Drucker auszugeben. Dies ist der Fall, wenn Bildschirmmasken zur Dokumentation eines Programmablaufes archiviert werden müssen oder Eingabemasken zum besseren Verständnis von Software-Bedienungsanleitungen abgebildet werden sollen. Aber auch die Darstellung von Bildschirmgrafiken auf einem angeschlossenen Drucker während des Programmablaufes wird sehr oft gewünscht. Kurzum, immer wenn der Bildschirminhalt in kürzester Zeit unverändert zu Papier gebracht werden soll, egal aus welchen Gründen, dann stellt die nachfolgende »Hardcopy-Subroutine in Basic« eine wertvolle Hilfe dar. Voraussetzung dafür ist, daß Sie einen CBM 2000, 3000, 4000 oder 8000 besitzen, an dem ein Drucker CBM 3022 oder 4022 angeschlossen ist. Die Programmbeschreibung zeigt außerdem, wie durch kleine Modifikationen das Programm auch anderen Druckern angepaßt werden kann.

TEXT-HARDCOPY:

```
! " # % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G  
H I J K L M N O P Q R S T U V W X Y Z I \ ] ^ _ ` ! " # % & ' ( ) * + , - . /  
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
```

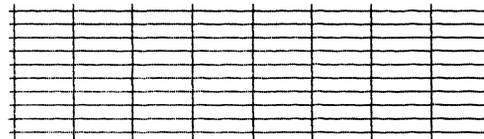
```
! " # % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G  
H I J K L M N O P Q R S T U V W X Y Z I \ ] ^ _ ` ! " # % & ' ( ) * + , - . /  
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
```

```
! " # % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G  
H I J K L M N O P Q R S T U V W X Y Z I \ ] ^ _ ` ! " # % & ' ( ) * + , - . /  
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
```

DIES IST EIN **■ ■ ■ ■ ■ ■ ■ ■ ■ ■** VON TEXT
< ZEILENABSTAND NORMAL >

GRAFIK-HARDCOPY:

■ ■ ■ ■ ■ ■ ■ ■ ■ ■



■ ■ ■ ■ ■ ■ ■ ■ ■ ■

Problembeschreibung

Die Mehrzahl der auf dem Markt angebotenen Utilities (Programmierhilfen) und Betriebssystemerweiterungen für Personal Computer enthalten bereits eine sogenannte »Hardcopy-Routine«, die es erlaubt, eine Ausgabe des Bildschirminhaltes per Programmbefehl zu erreichen. In der Regel haben die meisten dieser Befehlssatzerweiterungen drei wesentliche Nachteile:

1. Sie sind für den Hobby-Anwender meist zu teuer und
2. läßt sich der »Hardcopy«-Befehl bei vielen Toolkits nur im Direktmodus (per Tastatureingabe) aktivieren und ist nicht programmierbar.
3. Die Routine setzt oft einen ganz bestimmten Druckertyp voraus, die Hardcopy-Ausgabe kann herstellerfremden Druckern nicht angepaßt werden.

Wir benötigen also eine Routine, die zuverlässig den Bildschirminhalt auf einem Drucker darstellt und darüber hinaus per Programm angesteuert und durch kurze Programmänderung fast jedem beliebigem Drucker angepaßt werden kann.

Problemlösung

In den Zeilen 150 bis 200 erfolgt die Definition der Zufallsfunktionen und die Anlegung der Variablen-tabelle. Die Auswahlmaske mit der Bestimmung des Schwierigkeitsgrades und der Eingabe »Anzahl Aufgaben« (Rechenkettenglieder) verbirgt sich in den Programmzeilen 220 bis 360. Die Ermittlung des Anfangswertes und der »Countdown« stehen in 380 bis 490. Ab der Zeile 510 bis einschließlich 720 ist die Rechenroutine untergebracht mit der Möglichkeit, Zwischenergebnisse zu überprüfen. Bei erfolgreicher Prüfung werden Strafsekunden angerechnet, die dann zur tatsächlich benötigten Gesamtzeit hinzuaddiert werden. Die Kontrolle des Endergebnisses, Berechnung der Rechendauer und Ausgabe entsprechender Erfolgs- oder Fehlermeldungen erfolgt in 740 bis 820. Die Wiederholungsmöglichkeit mittels »Ja/Nein-Abfrage« ist in den Programmzeilen 840 bis 880 untergebracht.

In einem Satz ausgedrückt, besteht die Aufgabe einer Hardcopy-Routine darin, den Bildschirmspeicher Zelle für Zelle auszulesen und den jeweiligen Inhalt so an den angeschlossenen Drucker zu übergeben, daß dieser quasi eine Kopie der aktuellen Bildschirmdarstellung erzeugen kann. Speziell für CBM-Computer bedeutet dies, daß dazu der Bildschirmcode des Zeichensatzes umgewandelt werden muß. Im einzelnen sind folgende Teilprobleme zu lösen:

- Ermittlung und Speicherung des aktuellen Bildschirm-Darstellungsmodes (Groß/Grafik, Text);
- Druckereröffnung, Wahl des Zeilenabstandes über Kennzeichen;

- Lesen jeder einzelnen Bildschirmspeicherzelle, dabei Umwandlung des Bildschirmcodes in CBM-ASCII-Code, ggf. Reversstatus setzen und Übergabe an den Druckpuffer;
- Leeren des Druckpuffers, Reversstatus neutralisieren;
- Zeilenabstand (Vorschub) des Druckers normalisieren, Druckerfiles schließen und Rücksprung zur nächst höheren Programmebene.

Im allgemeinen ist der Bildschirmcode nur bei Verwendung der Maschinensprache interessant. Mit »PEEK« und »POKE« kann aber auch von Basic aus direkt auf den Bild-Wiederholpeicher zugegriffen werden. Den Befehl »PEEK« nutzen wir zum Auslesen des Bildschirminhaltes und zur Ermittlung der Darstellungsweise des Zeichengenerators. Ihr CBM-Computer kann 128 verschiedene Zeichen auf dem Bildschirm darstellen (Bildschirmcode 0 bis 127). Addiert man zu dem jeweiligen Wert 128 dazu, erhält man den entsprechenden Bildschirmcode des Zeichens für die Reversdarstellung. Hierbei ist der Zustand des Zeichengenerators zu beachten. Befindet er sich im Textmode, dann erscheinen u. a. Klein- und Großbuchstaben auf dem Bildschirm, während im Grafikmodus Großbuchstaben und grafische Zeichen dargestellt werden. Das folgende Kurzprogramm soll dies verdeutlichen:

```

10 REM -Bildschirmcode- (alle CBM)
20 PRINT CHR$(147):REM CLEAR HOME
30 FOR I=0TO255:POKE32768+I,I:NEXT I
40 FOR J=1TO1000:NEXT J:POKE 59468,12
50 FOR J=1TO1000:NEXT J:POKE 59468,14
60 GOTO40

```

Wie im obigen Beispiel klar zu erkennen ist, erfolgt die Umschaltung zwischen Grafik und Kleinschreibung normalerweise wie folgt:

```

Grafik — POKE 59468,12
Text   — POKE 59468,14

```

Bei CBM-Rechnern der Serie 8000 ist jedoch zusätzlich auf eine Besonderheit zu achten. Die beiden POKE-Befehle ändern hier nur den Zeichensatz, während mit PRINT CHR\$(14) der Zeichengenerator so auf Textmodus umgeschaltet wird, daß zwei Dunkelzeilen zwischen zwei Textzeilen eingefügt werden, um die Lesbarkeit zu erhöhen. Durch PRINT CHR\$(142) werden diese entfernt, so daß Grafik auf dem Bildschirm vertikal nicht unterbrochen wird.

Um nun Schaubilder und Diagramme unverzerrt auf dem Drucker wiederzugeben, muß auch dort der Zeilenabstand »Null« (8 lpi) eingestellt werden (normal = 6 lines per inch). Bei den CBM-Printern 3022 und 4022 kontrolliert die Se-

kundäradresse 6 den Zeilenabstand, während beim CBM 8024 beispielsweise die Umschaltung zwischen 6 und 8 lpi per Geräteschalter erfolgt. Da er aber nicht den gesamten Grafikzeichensatz beherrscht, kann auf diesem und anderen herstellerfremden Printern nur die Hardcopy von Text interessant sein. Eine Möglichkeit, diese Einschränkung zu umgehen, besteht darin, die Grafik- und Sonderzeichen bereits auf dem Bildschirm in druckbare Zeichen umzuwandeln. Ferner ist zu berücksichtigen, daß der Darstellungsmodus des Druckers (Text oder Grafik) über spezielle Kommandos unabhängig vom Bildschirm geschaltet werden muß. Für CBM 3022 und 4022 gilt:

```
Grafik — PRINT# 1,CHR$(145)
Text   — PRINT# 1,CHR$(17).
```

In fast allen CBM-Handbüchern sind Codetabellen abgebildet. Mit ihrer Hilfe ist es leicht, einen Algorithmus für die Wandlung Bildschirmcode in CBM-ASCII-Code zu finden.

Das abgebildete Unterprogramm »Hardcopy« (63000-) bietet alle beschriebenen Möglichkeiten und gibt ohne Einschränkung den gesamten CBM-Zeichensatz auf dem angeschlossenen Drucker aus.

Bedienungsanleitung

Wie Sie vielleicht schon bemerkt haben, ist dem eigentlichen Unterprogramm eine Demonstrationsroutine vorgeschaltet. Sie enthält verschiedene Beispiele und Hinweise für Programmierer. Im übrigen sollten alle Unterprogramme und Hilfsroutinen, die Sie unter eigenem Programmnamen archivieren, einen solchen Vorspann enthalten. Er kann dazu dienen, durch einfachen Programmstart den Sinn und Zweck des Programmteiles leicher und schneller zu erkennen.

Tippen Sie also das Listing wie abgebildet in ihren Computer ein und speichern es danach auf einen externen Datenträger (Kassette oder Diskette) weg. Jetzt starten Sie mit »RUN« das im Arbeitsspeicher befindliche Programm. Achten Sie darauf, daß der angeschlossene Drucker zu diesem Zeitpunkt bereits eingeschaltet ist. Auf dem Bildschirm erscheint nun die Auswahlmaske, die wahlweise Text- oder Grafikausgabe ermöglicht. Durch Drücken der Taste »1« kann sich der Bediener ein Beispiel für Text-Hardcopy vorführen lassen, während durch die Taste »2« die Grafikausgabe demonstriert wird. In beiden Fällen springt der Computer in die Hardcopy-Unterroutine (GOSUB 63000), arbeitet diese ab und startet das Programm erneut. Hier kann der Programmablauf mittels der STOP-Taste unterbrochen werden.


```

63016 GOTO63020
63017 IFRV%THEN63019
63018 RV%=1:PRINT#204,CHR$(18);
63019 U3=U3-128
63020 IFU3>31THENIFU3<64THEN63029
63021 IFU3>95THENU3=U3+64:GOTO63029
63022 IFPZ=14THEN63025
63023 IFU3<32THENU3=U3+64:GOTO63029
63024 IFU3>63THENU3=U3+128:GOTO63029
63025 IFU3<27THENU3=U3+64:GOTO63029
63026 IFU3<32THENU3=U3+192:GOTO63029
63027 IFU3>90THEN63029
63028 IFU3>63THENU3=U3+128
63029 PRINT#204,CHR$(U3);
63030 NEXTU0
63031 :
63032 RV%=0:PRINT#204:REM REVERSTATUS OFF
63033 NEXTUX
63034 :
63035 PRINT#206,CHR$(24):REM [ZEILENABST. = NORMAL]
63036 CLOSE206:CLOSE204
63037 RETURN

```

ÄNDERUNGEN FÜR 40-ZEICHEN-BS

```

10 REM *HARDCOPY-SUBROUTINE IN BASIC*
11 REM * 40-ZEICHEN-BILDSCHIRM *
12 REM
63009 FORUX=32768TO33767STEP40
63010 U1=UX+39

```

Programmbeschreibung

In den Zeilen 10 bis 42 befindet sich die Demo-Routine. Sie dient lediglich dazu, die Wirkungsweise des Unterprogramms »Hardcopy« (63000 bis 63037) zu verdeutlichen. Das heißt um dieses Programm in anderen Problemlösungen einzusetzen, brauchen Sie nur die Programmzeilen 63000 bis 63037 abzutippen. Damit die Routine dann auch einwandfrei funktioniert, müssen Sie folgendes beachten:

Das Unterprogramm arbeitet mit den Variablen ZA% (= Status für Zeilenabstand des Druckers), PZ (= Inhalt des Zeichengenerators), GK\$ (= Darstellungsmode des Druckers), UX, U1, U0 (= Laufvariablen), U3 (= Bildschirmcode der aktuellen Bildschirmzelle) und RV% (= Reversstatus). Die genannten Variablennamen sollten in ihrem Hauptprogramm nicht anderweitig verwendet werden — mit einer Ausnahme:

ZA% muß vor jedem Einsprung in die Subroutine mit GOSUB 63000 entweder auf 1 oder 2 gesetzt werden. Wenn Sie beispielsweise eine Grafik-Hardcopy anfertigen möchten, dann könnte die Ansteuerung des Unterprogramms wie folgt aussehen:

```
100 ZA% = 1:GOSUB63000:...
```

Zur Erläuterung wird auf die Programmzeilen 63004 und 63005 hingewiesen. Wenn Sie statt eines 80-Zeichen-Bildschirmes (CBM 8000, teilweise CBM 4000) mit Rechnern der Serie 2000, 3000 und 4000 arbeiten, dann müssen Sie in den Zeilen 63009 und 63010 die entsprechenden Befehle einsetzen.

Kopfrechen-Training

Kopfrechen-Training

Im Zeitalter der Mikro- und Taschencomputer fällt uns das Lösen der einfachsten mathematischen Aufgaben scheinbar immer schwerer. Denkbar einfach ist ja auch, beispielsweise »-9« und »+21« in den Taschenrechner einzutippen und per Ergebnistaste das Resultat sofort auf dem Display stehen zu haben. Hand aufs Herz: Welcher Besitzer eines Taschenrechners greift nicht automatisch, vielleicht sogar unbewußt, schon bei einfachen Additionen und Subtraktionen in die Tasten? Natürlich kann ein Computer hervorragend rechnen und ist bestens geeignet, auch komplizierteste mathematische Aufgaben zu lösen. Er befreit den Menschen von Routinearbeiten und kann dadurch zur besseren Kreativitätsentfaltung wesentlich beitragen.

Allerdings kann man darüber streiten, ob bereits Schüler den Taschenrechner zur Erlernung der Grundrechenarten und zur Lösung solcher Aufgaben benutzen sollten. Diese Bequemlichkeit kann nämlich u.U. die Lernentwicklung ganz erheblich einschränken, da sie dazu führt, sich nur die Bedienung des Taschencomputers einzuprägen, nicht aber die Lösungsmethoden einfachster Aufgabenstellungen zu erlernen.

Uns allen würde es wenig schaden, würden wir des öfteren unser Gehirn zur Bewältigung kleinerer mathematischer Probleme beanspruchen und trainieren. Das Lösen komplizierter Formeln mittels Taschenrechner setzt nämlich zumindest die Beherrschung der einfachen Lösungstechniken voraus. Kettenaufgaben waren schon immer ein beliebtes Hilfsmittel zum Kopfrechen-Training. Hierbei sind, ausgehend von einem Anfangswert, eine beliebige Anzahl Einzelaufgaben zu lösen, deren Zwischenergebnis jeweils automatisch Anfangswert der nächsten Aufgabe ist. Paradox ist nur, daß wiederum ein Computerprogramm ein ideales Hilfsmittel für unser Kopfrechen-Training darstellt. Ein Computer ist durch ein Programm in der Lage, Überwachungsfunktionen und die Fortschrittskontrolle der Lernentwicklung zu übernehmen — unbedingte Voraussetzungen für ein erfolgreiches Schulungstraining.

Das nachfolgende Basicprogramm »Kopfrechen-Training« läuft auf allen CBM 2000, 3000, 4000, 8000 und VC 20. Es kann auch auf ähnlichen Computersystemen installiert werden, da lediglich der Rechner mit Bildschirm benötigt wird.

KOPFRECHNEN-TRAINING

HIER KANNST DU DIE ADDITION UND
SUBTRAKTION GANZER ZAHLEN UEBEN.

SCHWIERIGKEITSGRAD:

1 = LEICHT

2 = MITTEL

3 = SCHWER

BITTE WAEHLN !

ANZAHL DER AUFGABEN: ? 5

ANFANGSWERT IST 14

ES GEHT LOS IN
6 SEKUNDEN

(PLUS) + 24

NOCH 2 AUFGABE(N) !

2 = ZWISCHENERGEBNIS PRUEFEN

R = WEITER

ZWISCHENERGEBNIS:
? 18

??? LEIDER **FALSCH** !
ZWISCHENERGEBNIS IST 61
UND NICHT 18 .

NOCHMAL ? (U/N)

ENDERGEBNIS: ? -17
??? FALSCH !

ERGEBNIS IST-49 UND NICHT-17 .
DU HAST BEI SCHWIERIGKEITSGRAD 2
28 SEKUNDEN FUER 5 AUFGABEN
BENOETIGT. DAS SIND DURCHSCHNITTLICH
4 SEKUNDEN PRO AUFGABE.

NOCHMAL ? (U/N)

(PLUS) + 15

NOCH 0 AUFGABE(N) !

2 = ZWISCHENERGEBNIS PRUEFEN

R = WEITER

ENDERGEBNIS: ? -20

BRAVO, RICHTIG !
DU HAST BEI SCHWIERIGKEITSGRAD 2
19 SEKUNDEN FUER 5 AUFGABEN
BENOETIGT. DAS SIND DURCHSCHNITTLICH
3.8 SEKUNDEN PRO AUFGABE.

NOCHMAL ? (U/N)

TSCHUESS !!!

Bedienungsanleitung

Nach Einladen und Starten des Programmes mit »RUN« erscheint auf dem Monitor des Rechners eine Auswahlmaske. Der Benutzer kann hier zwischen drei Schwierigkeitsgraden wählen und die gewünschte Anzahl Aufgaben (Glieder der Rechenkette) eingeben. Über die Zufallsfunktion ermittelt der Computer nun einen Anfangswert und teilt diesen dem Anwender durch Ausdruck auf dem Bildschirm mit. Ein kleiner »Countdown« bereitet den Benutzer durch Anzeige der noch verbleibenden Zeit auf den Beginn der Rechenkette vor. Die anschließende Rechenroutine enthält folgende Komponenten:

- Generieren einer Zufallszahl gemäß dem gewählten Schwierigkeitsgrad;
- Ermittlung der Rechenart (Addition oder Subtraktion) durch Erzeugung eines entsprechenden Vorzeichens;
- Darstellung der so ermittelten Rechenoperationen auf dem Bildschirm;
- Anzeige der Anzahl noch zu lösender Aufgaben;
- Überprüfungsmöglichkeit des Zwischenergebnisses mit der Vergabe von Strafsekunden, abhängig vom Schwierigkeitsgrad;
- Nach Darstellung der gewünschten Anzahl Rechenoperationen Eingabe und Überprüfung des vom Benutzer ermittelten Endergebnisses mit den entsprechenden Meldungen;
- Ermittlung der durchschnittlichen Lösungsdauer pro Teilaufgabe zur Feststellung des Lernfortschrittes.

Am Ende jeder Rechenkette bzw. bei falschem Zwischenergebnis besteht die Möglichkeit, durch Tastendruck die Programmausführung erneut zu starten oder zu beenden. Innerhalb der Rechenroutine kann durch Betätigen der Taste »W« (für »Weiter«) die nächste Rechenoperation angezeigt oder durch Drücken der »Z«-Taste (für »Zwischenergebnis«) das Zwischenergebnis kontrolliert werden. Eine positive Überprüfung der Zwischenergebnisse wird durch Strafsekunden belegt, während bei Nichtübereinstimmung die Rechenkette beendet und zur Wiederholungsroutine verzweigt wird.

Deutlich muß hier wiederholt werden, daß das Programm »Kopfrechen-Training« nicht zur Lösung mathematischer Aufgaben dient, sondern dem Anwender durch Überwachung und Kontrolle beim Training mathematischer Grundfertigkeiten behilflich sein soll.

◆◆◆ KOPFRECHNEN ◆◆◆

```

100 REM *** KOPFRECHNEN-TRAINING ***
110 REM      * RECHENKETTE *
120 REM * ADDITION & SUBTRAKTION *
130 REM +-----+
140 :
150 DEFFNRD(X)=INT(X*RND(1)+1):REM ZAHLENGENERATOR
160 DEFFNVZ(X)=INT(RND(1)+.5):REM VORZEICHENGENERATOR
170 :
180 REM VARIABLENLISTE
190 DIMVZ$(1):VZ$(0)="(MINUS) "":VZ$(1)="(PLUS) +"
200 DIME,I,J,V,W,AA,EE,ZZ,TA,TG,SG,ZE,X$
210 :
220 PRINT"■KOPFRECHNEN-TRAINING■"
230 PRINT"■HIER KANNST DU DIE ADDITION UND"
240 PRINT"SUBTRAKTION GANZER ZAHLEN UEBEN."
250 PRINT"■SCHWIERIGKEITSGRADE:"
260 PRINT"■ 1 ■ = LEICHT"
270 PRINT"■ 2 ■ = MITTEL"
280 PRINT"■ 3 ■ = SCHWER"
290 PRINT"■BITTE WAEHLEN !"
300 GETX$:IFX$="1"THENZZ=10:GOTO340
310 IFX$="2"THENZZ=25:GOTO340
320 IFX$="3"THEN300
330 ZZ=100
340 SG=VAL(X$):REM SCHWIERIGKEITSGRAD
350 INPUT"■ANZAHL DER AUFGABEN: ";AA
360 IFAA<=0THEN350
370 :
380 W=FNRD(ZZ):REM ANFANGSWERT
390 V=FNVZ(V):REM VORZEICHEN
400 IFV=0THEN:W=(-W):REM V=0,NEGATIV
410 PRINT"■ANFANGSWERT IST ";W
420 PRINT"■ES GEHT LOS IN"
430 :
440 REM COUNTDOWN
450 FORI=10TO0STEP-1
460 TA=VAL(RIGHT$(TI$,1))
470 IFVAL(RIGHT$(TI$,1))=TATHEN470
480 PRINT" ■■■";I;" SEKUNDEN  ⌏"
490 NEXTI
500 :
510 E=W:TA=TI
520 FORJ=AA-1TO0STEP-1
530 W=FNRD(ZZ):V=FNVZ(V):IFV=0THEN:W=(-W)
540 E=E+W:ZE=0:REM ERGEBNIS
550 PRINT"⌏"
560 PRINT"  _____,"
570 PRINTVZ$(V);W
580 PRINT"  _____"
590 PRINT"■NOCH ";J;"AUFGABE(N) !"

```

```

600 PRINT"Z = ZWISCHENERGEBNIS PRUEFEN"
610 PRINT"W = WEITER"
620 GETX$:IFX$="W"THEN720
630 IFX$<>"Z"THEN620
640 :
650 PRINT"ZWISCHENERGEBNIS:"
660 INPUTZE
670 IFZE=ETHEN:TA=TA-(((1/SG)*3)*60):GOTO720:REM STRAFSEK.
680 PRINT" ??? LEIDER FALSCH !"
690 PRINT"ZWISCHENERGEBNIS IST ";E
700 PRINT"UND NICHT ";ZE;". "
710 GOTO840:REM FALSCHES ZWISCHENERGEBNIS
720 NEXTJ
730 :
740 INPUT"ENDERGEBNIS: ";EE
750 TG=INT((TI-TA)/60):REM ZEITNAHME
760 IFEE=ETHEN:PRINT"BRAVO, RICHTIG !":GOTO790
770 IFEE<>ETHEN:PRINT" ??? FALSCH !"
780 PRINT"ERGEBNIS IST";E;" UND NICHT";EE;". "
790 PRINT"DU HAST BEI SCHWIERIGKEITSGRAD ";SG
800 PRINTTG;" SEKUNDEN FUER ";AA;"AUFGABEN"
810 PRINT"BENOETIGT. DAS SIND DURCHSCHNITTLICH"
820 PRINTTG/AA;" SEKUNDEN PRO AUFGABE."
830 :
840 REM WIEDERHOLUNG
850 PRINT"NOCHMAL ? (J/N)"
860 GETX$:IFX$="J"THEN220
870 IFX$<>"N"THEN860
880 PRINT"TSCHUESS !!!":END

```

Programmbeschreibung

In den Zeilen 150 bis 200 erfolgt die Definition der Zufallsfunktionen und die Anlegung der Variablen-tabelle. Die Auswahlmaske mit der Bestimmung des Schwierigkeitsgrades und der Eingabe »Anzahl Aufgaben« (Rechenkettenglieder) verbirgt sich in den Programmzeilen 220 bis 360. Die Ermittlung des Anfangswertes und der »Countdown« stehen in 380 bis 490. Ab der Zeile 510 bis einschließlich 720 ist die Rechenroutine untergebracht mit der Möglichkeit, Zwischenergebnisse zu überprüfen. Bei erfolgreicher Prüfung werden Strafsekunden angerechnet, die dann zur tatsächlich benötigten Gesamtzeit hinzuaddiert werden. Die Kontrolle des Endergebnisses, Berechnung der Rechendauer und Ausgabe entsprechender Erfolgs- oder Fehlermeldungen erfolgt in 740 bis 820. Die Wiederholungsmöglichkeit mittels »Ja/Nein-Abfrage« ist in den Programmzeilen 840 bis 880 untergebracht.

Bildschirmrahmen

Bildschirmrahmen

Für 40- und 80-Zeichen-Bildschirm

Wer seinen Computer selbst programmiert, der stellt oft fest, daß er das eine oder andere Teilproblem in irgendeinem Programm schon einmal gut gelöst hat. Es liegt nun nahe, den entsprechenden Programmteil zu suchen und auf die jetzigen Bedürfnisse zuzuschneiden.

Betrifft dies ein häufig wiederkehrendes Problem, wie z.B. die Umrahmung einer Bildschirmmaske, Fehlerabfrageroutinen oder komfortable Eingabeprogramme, so ist dem Benutzer zu raten, möglichst allgemeingültige Kurzprogramme zu schreiben und unter eigenem Namen zu archivieren. Dies erspart dem Programmierer in den meisten Fällen Zeit in Form von Such- und Modifikationsaufwand. Allerdings sind hierbei ein paar wichtige Regeln einzuhalten, die bei Nichtbeachtung die eventuelle Zeitersparnis u.U. wieder zunichte machen. Zunächst muß die jeweilige Routine »demofähig« sein, das bedeutet, sie muß für sich alleine einen Sinn ergeben, durch Programmstart selbständig ablaufen. Häufig ist dazu ein kurzer Vorspann in Form von Bemerkungen und Variablenbelegungen erforderlich, hat aber den entscheidenden Vorteil, daß man das Programm einfach nur zu starten braucht, um zu sehen, was es bewirkt. Weiterhin ist wichtig, die Variablennamen so zu wählen und zu dokumentieren, daß eine schädliche Mehrfachbenutzung möglichst ausgeschlossen ist. Die Unterroutinen sollten unter einem, der Bedeutung entsprechenden, Namen in der Programmbibliothek abgelegt werden, damit die Suche erleichtert wird.

Entwickelt man nun Programme, in denen die Bildschirmdarstellung als Bedienungsführung oder Eingabemaske funktioniert, so kann sich eine Umrahmung dessen zur optischen Gestaltung positiv auf die Benutzerfreundlichkeit auswirken. Das Listing zeigt drei Versionen für die Darstellung eines Bildschirmrahmens auf CBM-Computern. Dabei ist unwesentlich, ob man einen 40 Zeichen oder 80 Zeichen breiten Bildschirm benutzt.

◆ BILDSCHIRMRAHMEN-VERSION 1 ◆

```

100 REM *BILDSCHIRMRAHMEN FUER ALLE*
110 REM *CBM-COMPUTER MIT*
120 REM *40- UND 80-ZEICHEN-BILDSCHIRM*
130 :
140 REM *****
150 :
160 POKE59468,12
170 INPUT"BILDSCHIRMBREITE (40 ODER 80):";BB
180 IFBB<>40&NB&BB<>80THEN:FFX=1
190 IFFFXTHEN:FFX=0:PRINT"❌ FALSCH EINGABE! ■ WIEDERHOLUNG"
    :GOTO170
200 AZ=25:REM ANZAHL ZEILEN
210 A=24:REM X-ZEICHEN FUER BILDSCHIRMRAND
220 REM 24 IST BSC FUER X
230 BA=32767:REM BILDSCHIRMANFANGSADRESSE - 1
240 :
250 PRINT"J":REM *** VERSION 1 ***
260 A1=BA:A2=BA+((AZ-1)*BB)
270 FORI=1TOBB:POKEA1+I,A:POKEA2+I,A:NEXTI
280 A1=BA:A2=BA+BB
290 FORI=1TOAZ:POKEA1+1,A:A1=A1+BB:POKEA1,A:NEXTI
300 FORUU=1TO500:NEXT:GOTO250

```

◆ BILDSCHIRMRAHMEN-VERSION 2 ◆

```

250 PRINT"J":REM *** VERSION 2 ***
260 A1=BA:A2=BA+(AZ*BB)+1
270 FORI=1TOBB:POKEA1+I,A:POKEA2-I,A:NEXTI
280 A1=BA+BB:A2=BA+((AZ-1)*BB)+1
290 FORI=1TOAZ:POKEA1+(I*BB),A:POKEA2-(I*BB),A:NEXTI
300 FORUU=1TO500:NEXT:GOTO250

```

◆ BILDSCHIRMRAHMEN-VERSION 3 ◆

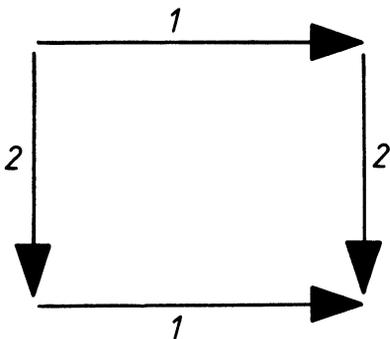
```

250 PRINT"J":REM *** VERSION 3 ***
260 A1=BA+BB/2:A2=BA+((AZ-1)*BB)+BB/2
270 FORI=1TOBB/2:POKEA1+(1-I),A:POKEA1+I,A:POKEA2+(1-I),A
    :POKEA2+I,A:NEXTI
280 A1=BA:A2=BA+((AZ-1)*BB):FORI=1TOAZ/2+1:POKEA1+1,A
290 A1=A1+BB:POKEA1,A:POKEA2,A:POKEA2+1,A:A2=A2-BB
    :NEXTI
300 FORUU=1TO500:NEXT:GOTO250

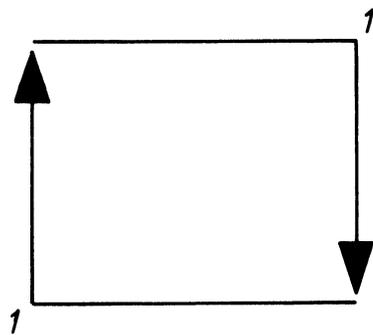
```

Auch auf den sichtbaren Bildschirmbereich kann man, wie auf den sonstigen RAM-Bereich, mit »POKE« zugreifen. Man braucht die entsprechende Zeilenlänge nur in kurze Formeln einzusetzen, und der Rahmen erscheint für jeden Bildschirm an der richtigen Stelle. Die Zeilen 100 bis 240 enthalten im sogenannten Vorspann die Variablenbelegung und erforderliche Bemerkungen. Die eigentlichen portablen Unterprogramme befinden sich jeweils in den Zeilen 250 bis 300 und sind untereinander austauschbar. Die einzelnen Versionen unterscheiden sich lediglich in der Ablauffolge der Darstellung des Rahmens.

Version 1



Version 2



Version 3

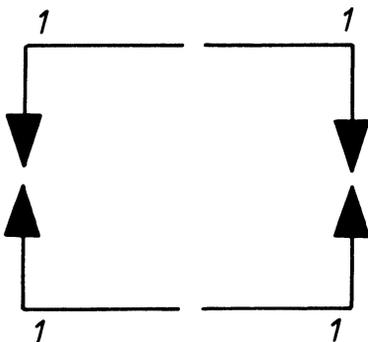


Bild 1: Aufbau von Bildschirmrahmen

Römische Zahlen

R O E M I S C H E Z A H L E N ! ! !

U M W A N D L U N G

[R] = ROEMISCH —> ARABISCH

[A] = ARABISCH —> ROEMISCH

.....

BITTE, WAEHLEN SIE !!!

ROEMISCHE ZAHL: ? MCMLXXXII
ROEMISCH **[MCMLXXXII]** = ARABISCH **[1992]**
ROEMISCHE ZAHL: ? IC
ROEMISCH **[IC]** = ARABISCH **[99]**
ROEMISCHE ZAHL: ? MCLM
——> FALSCH: MCLM
EINGABE WIEDERHOLEN!
ROEMISCHE ZAHL: ? MCLI
ROEMISCH **[MCLI]** = ARABISCH **[1151]**
ROEMISCHE ZAHL: ? *

ARABISCHE ZAHL: ? 189
ARABISCH **[189]** = ROEMISCH **[CLXXXIX]**
ARABISCHE ZAHL: ? 444
ARABISCH **[444]** = ROEMISCH **[CDXLIV]**
ARABISCHE ZAHL: ? 123
ARABISCH **[123]** = ROEMISCH **[CXXIII]**
ARABISCHE ZAHL: ? 998
ARABISCH **[998]** = ROEMISCH **[CMXCVIII]**
ARABISCHE ZAHL: ? 99999

Bedienungsanleitung

Nach Starten des Programmes mit »RUN« kann der Benutzer durch Drücken der Taste »R« dem Computer mitteilen, daß er römische Zahlen eingeben möchte, die dann per Programm in arabische Zahlen umgewandelt werden sollen. Durch Drücken der Taste »A« entscheidet der Anwender, daß er die Umwandlung von arabischen in römische Zahlen wünscht.

Wird nun die »R«-Taste gedrückt, verlangt der Computer die Eingabe einer römischen Zahl. Nach Beenden der Eingabe mit »RETURN« wird diese in die Variable RZ\$ übernommen und zunächst plausibilliert. Anschließend prüft das Programm die Einhaltung aller Regeln, gibt die arabische Zahl auf dem Bildschirm aus und springt zurück in die Eingaberoutine. Bei etwaigen Fehlern (unplausibel, Regelverstöße, Eingabefehler) gibt das System eine Fehlermeldung aus mit dem Hinweis, an welcher Stelle die Eingabe nicht korrekt ist und verlangt eine erneute Eingabe. Bei Eintippen von »ENDE« kann das Programm hier beendet werden.

Hat der Benutzer nun vorher die »A«-Taste gedrückt, dann verlangt das System die Eingabe einer arabischen Zahl. Durch anschließendes Betätigen der »RETURN«-Taste wird diese Eingabe in die Variable »AZ« übernommen und auf Eingabefehler überprüft. Anschließend wandelt das Programm unter Beachtung aller Regeln die arabische in die entsprechende römische Zahl um und gibt diese auf dem Bildschirm aus. Bei etwaigen Eingabefehlern gibt das System eine Fehlermeldung aus und wartet erneut auf eine Bedieneringabe. Durch Eintippen von »99999« kann hier das Programm beendet werden.


```

580 IFH1=H0THENM0%=1:GOTO650
590 IFH1>H0THENIFM0%THEN730
600 IFH1>H0THENIFIE>1THENIFH0/2=INT(H0/2)THEN730
610 IFH1>H0THENIFIE>1THENM%=M%+1:M1%=1:H3=H2:GOTO650
620 IFH1=>H2THENIFIE>1THEN730
630 H2=H1:M%=0:M0%=0
640 :
650 H0=H1:IFM%>1THEN730
660 E0=RZ(H1):IFM%=1THENE2=E2-(2*E1)
670 E1=E0:E2=E2+E1
680 GOTO480
690 :
700 PRINT"ARABISCH ";RZ%;" = ARABISCH ";INT(E2)
710 GOTO770
720 :
730 REM* FEHLERMELDUNG *
740 PRINT" —> FALSCH: ";LEFT$(RZ$,IE-1);
750 PRINT" "MID$(RZ$,IE,1)" "MID$(RZ$,IE+1)
760 PRINT" EINGABE WIEDERHOLEN!"
770 GOTO310
780 REM* UMWANDLUNG ARABISCH --> ROEMISCH *
790 :
800 :
810 PRINT" " : REM* ARABISCH -> ROEMISCH
820 PRINT
830 INPUT"ARABISCHE ZAHL:   0000";AZ
840 AZ=INT(AZ):IFAZ=0THENPRINT" " : GOTO830
850 IFAZ<99999THENEND
860 IFAZ<0ORAZ>10000THENPRINT" FALSCH EINGABE!"
   : GOTO820
870 :
880 RZ=AZ
890 RZ$=""
900 H1=0
910 H1=H1+1:IFH1>7THENRZ$=RZ$+"M":RZ=RZ-RZ(7):GOTO900
920 SG=SGN(RZ-RZ(H1)):IFSG>0THEN910
930 :
940 IFSG=0THENRZ$=RZ$+RZ$(H1):GOTO1030
950 :
960 FORH2=1TO(H1-1)STEP2:VG=RZ(H1)-RZ:VG%=0:VH%=0
970 IFVG=RZ(H2)THENRZ$=RZ$+RZ$(H2)+RZ$(H1):GOTO1030
980 IFRZ>RZ(H2)ANDRZ<RZ(H2+2)THEN:IFVG<RZ(H2)THENVH%=1
990 IFVH%THEN:RZ$=RZ$+RZ$(H2)+RZ$(H1):VG%=1
1000 IFVG%THEN:RZ=RZ+RZ(H2)-RZ(H1):RZ=ABS(RZ):GOTO900
1010 NEXT
1020 RZ$=RZ$+RZ$(H1-1):RZ=RZ-RZ(H1-1):GOTO900
1030 PRINT"ARABISCH ";AZ%;" = ROEMISCH ";RZ$
1040 GOTO820

```

Programmbeschreibung

In den Zeilen 140 bis 180 werden die Hilfsvariablen dimensioniert und belegt. In 200 bis 250 wird das Auswahlmenü auf dem Bildschirm dargestellt. Die Tastenabfrage und der Sprung zum gewählten Programmteil erfolgen in den Zeilen 260 bis 290. Die Eingaberoutine für die Umwandlung römisch-arabisch steht in 300 bis 350. Anschließend erfolgt von 370 bis 440 die Plausibilierung, d.h. die Eingaben werden auf syntaktische Fehler geprüft. Die Zeilen 460 bis 710 enthalten die Regeln der Schreibweise römischer Zahlen und das Umwandlungsprogramm einschließlich der Ausgabe der arabischen Zahl auf dem Bildschirm. Bei etwaigen Fehlern springt das Programm in die Fehlermelde-Routine, welche in den Zeilen 730 bis 770 aufgeführt ist. Die Eingaberoutine arabisch-römisch steht in 780 bis 860. In den Zeilen 880 bis 1040 wird die arabische Zahl in die entsprechende römische Zahl umgewandelt und auf dem Bildschirm ausgegeben.

Kugelabschnitt

Kugelabschnitt

Bei der Entwicklung und Konstruktion von Baugruppen, in denen Kugeln, Kugelkalotten oder ähnliche Einzelteile als Dichtungs-, Durchflußregulierungs- oder Bewegungselemente eingesetzt werden, kann es oftmals notwendig werden, die Abhängigkeit der Pfeilhöhe (h) vom Radius (r) zu errechnen, um durch simultane Veränderung der Komponenten Pfeilhöhe (h), Radius (r) und Sehnenlänge (s) eine größtmögliche Funktionssicherheit zu erreichen.

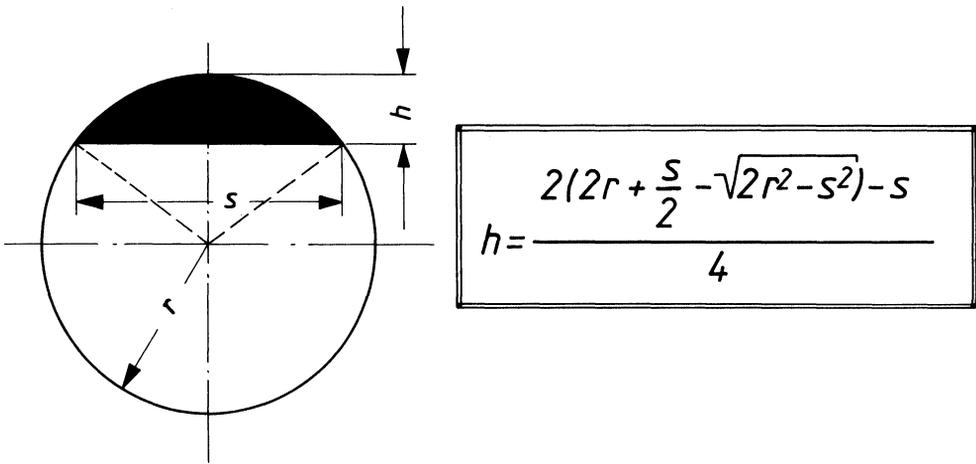


Bild 1: Formel zur Berechnung der Kalottenhöhe

Das Programm »Kugelabschnitt« stellt hierbei ein wesentliches Hilfsmittel dar, da es den Konstrukteur in die Lage versetzt, simultan veränderte Daten in komplizierte Formeln (siehe Bild 1) einzusetzen und in kürzester Zeit die erforderlichen Ergebnisse auf dem Bildschirm oder Papier stehen zu haben. Dieses mathematisch-wissenschaftliche Basic-Programm läuft auf allen PET/CBM-Rechnern (wahlweise mit Drucker) und dürfte ohne Modifikationen auch die Portabilität mit ähnlichen basicprogrammierbaren Computersystemen gewährleisten.

***** PFEILHOEHENBERECHNUNG *****

(EINGABEWERTE IN MM)

ANFANGS-DURCHMESSER : ? 5
 END-DURCHMESSER : ? 10
 SEHNENLAENGE : ? 4
 SCHRITTWEITE DER
 DURCHMESSERAENDERUNG : ? 0.5
 NACHKOMMASTELLEN : ? 4
 VERGROESSG.-FAKTOR : ? 10

B = BILDSCHIRM **D** = DRUCKER

***** ERGEBNISSTABELLE *****

DURCHMESS.	PFEILHOEHE	DIFFERENZ
5.0000	0.0000	0
5.5000	0.0000	1
6.0000	0.0000	4
6.5000	0.0000	9
7.0000	0.0000	16
7.5000	0.0000	25
8.0000	0.0000	36
8.5000	0.0000	49
9.0000	0.0000	64
9.5000	0.0000	81
10.0000	0.0000	100

VERGROESSG.-FAKTOR : 10

SEHNENLAENGE : 4

(**R** = WIEDERHOLUNG) (**F** = RUN) (**E** = END)

Bedienungsanleitung

Nach Einladen und Starten des Programmes mit »RUN« erscheint die Eingabemaske auf dem Bildschirm. Es ist darauf zu achten, daß alle Eingaben zu einer Längenmaßeinheit gehören (mm, cm, dm ...). Alle Eingaben werden plausibilisiert, ggf. wird eine Fehlermeldung auf dem Bildschirm ausgegeben und der Computer verlangt eine erneute Dateneingabe.

Zunächst erfolgt die Eingabe des Anfangs- und Enddurchmessers (Variablen »AD« und »ED«), d.h. der Benutzer teilt dem Computer den Durchmesserbereich mit, der simultan geändert werden soll. Anschließend fragt der Rechner nach der Sehnenlänge (s). Sie bleibt im Programmablauf konstant. Als nächstes verlangt das Programm die Eingabe der Schrittweite der Durchmesseränderung (»SW«). Diese kennzeichnet die Intervalle der Änderung des Durchmesserbereiches. Der Benutzer kann nachfolgend die Anzahl der Nachkommastellen eingeben, um so festzulegen, mit welcher Genauigkeit die Ergebnisse ausgegeben werden sollen. Durch die Eingabe eines Vergrößerungs-Faktors kann der Anwender die Differenz der Pfeilhöhe von Intervall zu Intervall beliebig vergrößern, um so zu möglichst brauchbaren Ergebnissen zu kommen.

Nachdem alle erforderlichen Daten erfaßt sind, kann der Bediener durch Tastendruck entscheiden, ob die Ergebnistabelle auf dem Bildschirm oder Drucker ausgegeben werden soll.

Beispiel:

Anfangsdurchmesser:	5
Enddurchmesser:	10
Sehnenlänge:	4
Schrittweite der Durchmesseränderung:	0,5
Nachkommastellen:	4
Vergrößerungsfaktor:	10

Diese Eingabewerte bedeuten, daß bei einer Sehnenlänge 4 die Kugeln vom Durchmesser 5 bis 10 auf ihre Pfeilhöhe hin untersucht werden sollen. Das Programm erhöht also den Durchmesser um die Schrittweite 0,5, errechnet mit diesen Werten die Pfeilhöhe und gibt sie nebst dem Durchmesser in Tabellenform aus. Dieser Vorgang wiederholt sich so lange, bis der Enddurchmesser erreicht ist. In obigem Beispiel werden die Ergebnisse mit vier Nachkommastellen ausgegeben. Pro Berechnungsvorgang wird neben dem jeweiligen Durchmesser und der Pfeilhöhe auch die Differenz von »h« zur vorigen Höhe ausgegeben. Die Eingabe des Faktors 10 in dem Beispiel bedeutet, daß diese Differenz mit 10 multipliziert auf dem Bildschirm oder Drucker angezeigt wird.

Nach der Ausgabe der Ergebnistabelle kann durch Betätigen der Taste »W« der Vorgang mit den gleichen Werten wiederholt, durch Drücken der Taste »R« das Programm erneut gestartet oder mit »E« beendet werden.

◆◆◆ KUGELABSCHNITT ◆◆◆

```

100 REM      **** KUGELABSCHNITT ****
110 REM      -----
120 REM * SIMULATION EINER DURCHMESSER- *
130 REM * ÄNDERUNG UND IHRE AUSWIRKUNG *
140 REM * AUF DIE PFEILHOEHE DER KALOTTE *
150 REM
160 REM      *****
170 REM
180 :
190 DIM I,R,H1,H2,H3,AD,ED,S,SW,NK,VF,X#
200 :
210 PRINTCHR$(142)CHR$(147):POKE59468,12
220 PRINT"  *** PFEILHOEHENBERECHNUNG ***  "
230 PRINT"    (EINGABEWERTE IN MM)"
240 INPUT"ANFANGS-DURCHMESSER: ";AD
250 IFAD<=0THENGOSUB700:GOTO240
260 INPUT"END-DURCHMESSER: ";ED
270 IFED<ADTHENGOSUB700:PRINT"END-DURCHM. < ANF.-DURCHM."
    :GOTO260
280 INPUT"SEHNENLAENGE: ";S
290 IFS>ADTHENGOSUB700:PRINT"SEHNENLAENGE > ANF.-DURCHM."
    :GOTO280
300 IFS>EDTHENGOSUB700:PRINT"SEHNENLAENGE > END-DURCHM."
    :GOTO280
310 PRINT"SCHRIITWEITE DER"
320 INPUT"DURCHMESSERAENDERUNG: ";SW
330 INPUT"ANACHKOMMASTELLEN: ";NK
340 NK=INT(NK):IFNK>9THENNK=9
350 INPUT"VERGROESSG.-FAKTOR: ";VF:VF=INT(VF)
360 :
370 DEFFNKA(KR)=INT(KR*10↑NK)/INT(10↑NK)
380 :
390 PRINT"  = BILDSCHIRM      = DRUCKER"
400 :
410 GETX$:IFX$="D"THEN:CLOSE1:OPEN1,4:CMD1
420 IFX$<>"B"THEN410
430 :
440 PRINTCHR$(147)
450 PRINT"  *** ERGEBNISTABELLE ***  "
460 PRINT"-----"
470 PRINT" DURCHMESS. | PFEILHOEHE | DIFFERENZ

```

```

480 PRINT"-----"
490 :
500 FOR I=ADTOEDSTEPSW:R=I/2:REM R = RADIUS
510 H3=((2*((2*R)+(S/2))-(SQR(((2*R)^2)-(S^2))))-S)/4
520 H1=H2-(H3*VF):IF H1<0 THEN H1=0
530 H2=(H3*VF)
540 R=FNKA(R*2):H3=FNKA(H3):H1=FNKA(H1)
550 PRINTR;TAB(11)"I";H3;TAB(24)"I";H1
560 NEXT I
570 :
580 PRINT"-----"
590 PRINT:PRINT"VERGROESS.-FAKTOR: ";VF
600 PRINT:PRINT"SEHNENLAENGE: ";S
610 PRINT:PRINT
620 PRINT"(W = WIEDERHOLUNG) (R = RUN) (E = END)"
630 :
640 GETX$:IFX$="R" THEN RUN
650 IFX$="W" THEN 380
660 IFX$<>"E" THEN 640
670 :
680 END
690 :
700 REM* FEHLER-ROUTINE *
710 PRINT"NE FALSCH! ";:RETURN

```

Programmbeschreibung

Dimensionierung der Variablen in Zeile 190. In den Zeilen 210 bis 350 befindet sich die Dialogmaske: Hier fordert das Programm den Benutzer auf, die erforderlichen Daten einzugeben. Die Eingabewerte werden gleichzeitig plausibilliert, d.h. auf Gültigkeit geprüft. Im Fehlerfall wird eine detaillierte Meldung ausgegeben (Sprung zum Unterprogramm 700 bis 710), und das Programm verzweigt erneut zur Eingabe. In Zeile 370 wird die Nachkommastellenberechnung definiert. Ob die Ergebnistabelle auf dem Bildschirm oder Drucker ausgegeben werden soll, entscheidet sich in den Zeilen 390 bis 420. In den Zeilen 440 bis 610 erfolgt die Berechnung und formatierte Ausgabe der Ergebnisse. Die Entscheidung, ob das Programm wiederholt, neu gestartet oder beendet werden soll, fällt in den Zeilen 620 bis 660.

Der Zeichengenerator des CBM

Der Zeichengenerator des CBM

»Jogging« für 40- und 80-Zeichen-Bildschirm

Mit »POKE« und »PEEK« ist es möglich, gezielt einzelne Speicherzellen des Computers zu verändern bzw. auszulesen. Erfahrene Programmierer nutzen dies z.B. zur Programmierung des User-Ports und zur Beeinflussung der Betriebssystem-Pointer. Aber auch Basic-Anfänger können diese Befehle effektiv einsetzen.

Die für diesen Zweck wohl bekannteste Speicherzelle ist die Adresse des Peripherie-Kontrollregisters \$E84C (59468 dez.). Der Inhalt dieser Speicherzelle ist u. a. für den Darstellungsmodus der Zeichen auf dem Bildschirm verantwortlich, und zwar kann durch »POKE 59468,14« auf Textmodus (Groß-/Kleinschreibung) und mit »POKE 59468,12« auf Grafikmodus umgeschaltet werden. Durch die Tatsache, daß der Bildschirmspeicherinhalt 60mal pro Sekunde an die Video-Hardware übergeben wird, bewirkt der POKE-Befehl ein schlagartiges Umschalten aller auf dem Bildschirm dargestellten Zeichen. Dieser interessante Effekt kann in verschiedenen Problemlösungen zur Bildschirmgestaltung oder zur Darstellung von Bewegungsabläufen usw. verwendet werden.

◆ JOGGING FUER 40-ZEICHEN-BS ◆

```
100 REM * JOGGING FUER CBM MIT 40-ZEICHEN-BILDSCHIRM *
110 REM      *** CBM 2000, 3000, 4000 ***
120 :
130 ZG=59468:BA=32767:G=12:K=14:X=3:W=20:VZ=60
140 T1=59467:T2=T1-1:T3=T1-3:REM FUER TON
150 PRINT"J":POKEZG,G:POKET1,16:POKET2,60:POKET3,0
160 :
170 PRINT"ALLE AUFSTELLEN ZUM JOGGING !!!"
180 FORUU=1TO2000:NEXT:PRINT"J"
190 FORI=1TO1000STEP2:POKEBA+I,X:NEXT
200 FORJ=1TO50:POKEZG,K:GOSUB280:POKEZG,G:GOSUB280
   :NEXT:POKET3,0
210 FORI=1TO(1000+VZ)STEP2:IFI>1000THEN230
220 POKEBA+I,85
230 IFI>VZTHEN:POKEBA+I-VZ,82
240 NEXT
250 PRINT"UEBEN - UEBEN - UEBEN"
260 FORI=1TO5:GOSUB280:NEXT:POKET3,0:POKET1,0
270 END
280 POKET3,223:GOSUB300:POKET3,250:GOSUB300:POKET3,0
290 W=W+2:FORUU=1TOW:NEXT:RETURN
300 FORUU=1TO30:NEXT:RETURN
```

◆ JOGGING FUER 80-ZEICHEN-BS ◆

```

100 REM * JOGGING FUER CBM MIT 80-ZEICHEN-BILDSCHIRM *
110 REM          *** CBM 8000 ***
120 :
130 ZG=59468:BA=32767:G=12:K=14:X=3:W=1:VZ=60
140 PRINT"J":POKEZG,G
150 :
160 PRINT"ALLE AUFSTELLEN ZUM JOGGING !!!"
170 FORUU=1TO2000:NEXT:PRINT"J"
180 FORI=1TO2000STEP2:POKEBA+I,X:NEXT
190 FORJ=1TO70:POKEZG,K:GOSUB270:POKEZG,G:GOSUB270
    :NEXT
200 FORI=1TO(2000+VZ)STEP2:IFI>2000THEN220
210 POKEBA+I,85
220 IFI>VZTHEN:POKEBA+I-VZ,82
230 NEXT
240 PRINT"UEBEN - UEBEN - UEBEN"
250 FORUU=1TO20:PRINTCHR$(7):NEXT
260 END
270 PRINTCHR$(7):W=W+1:FORUU=1TOW:NEXT:RETURN

```

Die kurzen Listings verdeutlichen anhand eines lustigen Beispiels die beeindruckende Wirkung der beiden POKE-Befehle. Das Programm »Jogging für CBM mit 40-Zeichen-Bildschirm« läuft, wie der Name schon sagt, auf allen CBM 2000, 3000 und 4000 mit einer Bildschirmbreite von 40 Zeichen/Zeile und ist wahlweise mit Tonausgabe an CB2 des User-Ports. Das zweite Programm läuft auf CBM 8000 und benutzt zur Tonerzeugung die eingebaute Klingel.

Beide Programme sind so konstruiert, daß durch Ändern weniger Parameter in Zeile 130 eine völlig andere Wirkung erzielt werden kann. Die Variable »X« beinhaltet den Bildschirmcode für das dargestellte aktuelle Zeichen, während sich in »W« der Startwert für die Warteschleife befindet. Die Variable »VZ« enthält einen Verzögerungsfaktor und wird dazu benötigt, den Bildschirminhalt durch »POKE« zu verändern.

Viel Spaß beim Experimentieren!

Entwurf von Sonderzeichen durch Nadelansteuerung

Entwurf von Sonderzeichen durch Nadelansteuerung

Für CBM-Printer 3022

Der Einsatz von Mikrocomputern in mathematisch-wissenschaftlichen Bereichen erfordert oft die Darstellung von Sonderzeichen, die nicht im normalen Zeichensatz vorhanden sind. Das Generieren von solchen Zeichen auf dem Bildschirm ist sehr schwierig, da in aller Regel ein Eingriff in das Betriebssystem des Rechners vorgenommen werden müsste. Wer sich aber damit begnügt, mathematische Formelzeichen, griechische Buchstaben usw. wenigstens mit einem Drucker zu Papier bringen zu können, wird mittels nachfolgendem Programm in die Lage versetzt, jedes beliebige Symbol in einer 7x6-Matrix zu erzeugen. Vorausgesetzt wird eine Hardware-Konfiguration, bestehend aus Rechnern der Serie CBM 2000, 3000, 4000 oder 8000 mit angeschlossenem Printer CBM 3022. Natürlich läuft dieses Programm auch auf Printern, die statt über Traktorführung das Papier über Friktionsantrieb vorschieben (CBM 3023). Da sich bei diesen Druckern jeder einzelne Druckpunkt in der 7x6-Matrix separat ansteuern lässt, ist es möglich, ein Drucksymbol frei zusammenzustellen. Dies kann ausgenutzt werden, um eigene Sonderzeichen zu entwerfen oder mit hoher Auflösung Kurven in einer Richtung zu plotten.

SONDERZEICHEN ENTWERFEN:

STEUERTASTEN:

- = NADEL SETZEN
- = NADEL LÖSCHEN
- = ENDE ERFASSUNG
- = WEITER
- + = ZURUECK

< ZEICHEN OK ? (J/N) >

SONDERZEICHEN ENTWERFEN:

STEUERTASTEN:

- = NADEL SETZEN
- = NADEL LÖSCHEN
- = ENDE ERFASSUNG
- = WEITER
- + = ZURUECK

< ZEICHEN OK ? (J/N) >

Problembeschreibung

In den Handbüchern der CBM-Printer kann man nachlesen, daß die Sekundäradresse 5 für die Ansteuerung einzelner Punkte in der Druckmatrix zuständig ist. Weiterhin wird die Empfehlung gegeben, eine Skizze zu erstellen, die gewünschten Druckpunkte einzuzeichnen und die Binärwerte der Matrix per Hand zu errechnen. Wenn diese Arbeit der Computer per Programm wiederum selbst durchführt, dann kann sich der Benutzer voll auf die Zeichengenerierung konzentrieren und seine Kreativität besser entfalten.

Das dazu erforderliche Programm muß den Bediener in die Lage versetzen, jedes beliebige Zeichen auf den Bildschirm zu malen und per Drucker wieder auszugeben.

GRIECHISCHE BUCHSTABEN ALS SONDERZEICHEN:

ALPHA:

α α α α α α α α α α α α α α α α
BINAERWERTE: 8 20 20 24 36 72

BETA:

β β β β β β β β β β β β β β β β
BINAERWERTE: 0 63 84 84 40 0

GAMMA:

γ γ γ γ γ γ γ γ γ γ γ γ γ γ γ γ
BINAERWERTE: 32 96 22 9 118 0

DELTA:

δ δ δ δ δ δ δ δ δ δ δ δ δ δ δ δ
BINAERWERTE: 0 6 41 81 73 38

EPSILON:

ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε ε
BINAERWERTE: 0 44 82 82 66 36

LAMBDA:

λ λ λ λ λ λ λ λ λ λ λ λ λ λ λ λ
BINAERWERTE: 34 68 72 94 34 4

MY:

μ μ μ μ μ μ μ μ μ μ μ μ μ μ μ μ
BINAERWERTE: 126 8 8 120 8 16

PHI:

φ φ φ φ φ φ φ φ φ φ φ φ φ φ φ φ
BINAERWERTE: 48 72 126 72 48 0

Eine klare Bedienungsführung und die Betätigung weniger Steuertasten zur Programm Benutzung garantieren Komfort und Flexibilität seitens des Anwenders und schaffen damit die Voraussetzung hoher Akzeptanz und Gestaltungsfreiheit. Dies kann wohl generell gesagt werden und trifft in verstärktem Maße auf alle Problemlösungen zu, in denen der Computer nicht »nur lästige Routinearbeiten« übernimmt. Um nun unser spezielles Problem zu lösen, muß das System folgende Arbeiten durchführen:

- Darstellung einer Eingabenmaske auf dem Bildschirm mit Erklärung der verfügbaren Steuertasten;
- Programmtechnische Verwaltung der Steuerfunktionen mit Plausibilisierung und Verhinderung von Bedienungsfehlern;
- Abfrage, ob eingegebenes Zeichen ausgedruckt werden soll, mit der Möglichkeit, zur Erfassung zurückzuspringen;
- Berechnung der Binärwerte der Matrix;
- Druckeransteuerung, Ausgabe des entworfenen Zeichens in Breit- und Normalschrift;
- Ausgabe der errechneten Binärwerte, die zur Einzelnadelansteuerung des Printers notwendig sind, um das Zeichen drucken zu können;
- Rücksprung in die Erfassungsroutine, ohne die vorhandene Matrix zu löschen.

Der oben angeführte Problemaufriß ist allgemeingültig und stellt nur einen Vorschlag zur generellen Vorgehensweise dar. Um das Programm nun auf einem bestimmten Computertyp, beispielsweise einem CBM-System, zu realisieren, ist eine genauere Erläuterung der einzelnen Punkte erforderlich.

Problemlösung

Per Programm muß zunächst eine Skizze auf dem Monitor dargestellt werden, die der 7 x 6-Druckmatrix entspricht. Diese Tabelle (7 Felder vertikal, 6 Felder horizontal) hat gleichzeitig die Funktion einer Eingabenmaske. Lediglich durch die Betätigung einiger Steuertasten wird der Benutzer in die Lage versetzt, sich in der Matrix beliebig zu bewegen und so eigene Zeichen zu kreieren. Nach Darstellung der Eingabenmaske auf dem Bildschirm blinkt der Cursor in der Matrix links oben. Folgende Steuertasten stehen dem Anwender jetzt primär zur Verfügung:

1. »RETURN«-Taste = Der Cursor wandert in der Tabelle um ein Feld vorwärts. Er durchquert dabei die Tabelle zeilenweise von links nach rechts. Am Ende

der Matrix angekommen, springt der Cursor durch »RETURN« wieder in das erste Feld links oben.

2. »SHIFT« + »RETURN«-Taste = Der blinkende Cursor bewegt sich rückwärts von rechts nach links. Im ersten Feld angekommen, wird er durch »SHIFT« + »RETURN« ins untere rechte Feld dirigiert.
3. »S«-Taste = Drückt man statt »RETURN« beispielsweise die Taste »S«, so wird an der aktuellen Cursor-Position ein Druckpunkt gesetzt. Gleichzeitig wandert der Blinker wie bei »RETURN« um eine Stelle vorwärts.
4. »L«-Taste = Mittels der »L«-Taste können Nadeln gelöscht werden. Befindet sich an der aktuellen Cursorposition ein Druckpunkt, dann wird dieser durch »L« gelöscht, der Cursor wandert eine Stelle vorwärts. Wenn kein Punkt vorhanden ist, wirkt diese Taste ebenso wie »RETURN«.
5. »E«-Taste = Nachdem der Benutzer ein Symbol entworfen hat, betätigt er die »E«-Taste. Diese Taste bedeutet nichts anderes als »Ende Erfassung«.

Hat der Anwender die zuletzt beschriebene Taste gedrückt, erscheint folgende Abfrage auf dem Bildschirm: »ZEICHEN OK? (J/N)«. Wie Sie vielleicht bereits bemerkt haben, stehen nun sekundär noch die Steuertasten »J« für »Ja« und »N« für »Nein« zur Verfügung. Bei negativer Beantwortung der Frage verzweigt das Programm erneut zur Erfassungsroutine. Ist die Antwort positiv, wird das Zeichen auf verschiedene Arten auf dem Drucker ausgegeben. Außer den Binärwerten und der Darstellung des Sonderzeichens in Normalschrift erkennt der Benutzer auch die Wirkung der verdoppelten Matrix. Anhand der Binärwerte ist es nun kein Problem mehr, das Zeichen in jeder beliebigen Druckroutine zu erzeugen. Nachdem der Ausdruck beendet ist, verzweigt das Programm erneut in die Erfassungsroutine, ohne die bereits vorhandene Matrix zu löschen.

Bedienungsanleitung

Nach Eintippen des Listings sollte das Programm zunächst zur Datensicherung auf einen externen Datenträger weggespeichert werden. Eingabemaske und Bedienerführung erscheinen nach Starten des im Arbeitsspeicher befindlichen Programmes mit »RUN«. Die Darstellung aller erforderlichen Eingaben und Anweisungen in einer Bildschirmmaske entspricht der eingangs gestellten Forderung nach Klarheit bei gleichzeitiger Knappheit der Bedienungsanleitung. Bitte entnehmen Sie die Wirkungsweise der Steuerfunktionen der obigen Beschreibung. Vergewissern Sie sich, daß spätestens vor Betätigen der Taste »E« der angeschlossene Drucker auch eingeschaltet ist.

Wer dieses Programm auf einem CBM 8000 »fahren« möchte, findet im Anschluß an das Listing die entsprechenden Änderungen. Es müssen lediglich die Programmzeilen 75, 80, 550 und 680 korrigiert werden.


```

490 IFA=83THENPOKEBA,81:GOTO530
500 IFA=76THENPOKEBA,32:GOTO530
510 IFA=69THENS=0:Z=0:GOTO610
520 GOTO430
530 S=S+1:IFS>5THENS=0:Z=Z+1
540 IFZ>6THENZ=0
550 BA=AB+(S*2)+(Z*80)
560 REM* FUER 8032 -> ...+(Z*160)
570 GOTO430
580 S=S-1:IFS<0THENS=5:Z=Z-1
590 IFZ<0THENZ=6
600 GOTO550
610 PRINTSP#P#("< ZEICHEN OK ? (J/N) >")
620 GETX#:IFX#="N"THEN400
630 IFX#<"J"THEN620
640 :
550 PRINTCH#
660 BW#="":PS=0:REM* PS=PRUEFSUMME
670 FORS=0TO5:BW(S)=0
680 FORZ=6TO0STEP-1:BA=AB+(S*2)+(Z*80)
690 REM* FUER 8032 -> ...+(Z*160)
700 BI=PEEK(BA):IFBI<081THEN720
710 BW(S)=BW(S)+INT(2↑(ABS(Z-6))) : PS=PS+BW(S)
720 NEXTZ
730 BW#=BW#+CHR#(BW(S))
740 NEXTS:S=0:Z=0
750 IFPS=0THEN400
760 :
770 REM *DRUCKER*
780 OPEN5,4,5:PRINT#5,BW#
790 OPEN1,4
800 FOR00=1TO6:PRINT#1,SZ#" "VD#;:NEXT:PRINT#1
810 FOR00=1TO18:PRINT#1,SZ#" " : :NEXT:PRINT#1
820 PRINT#1,"BINNWERTE:" : FOR00=0TO5:PRINT#1,BW(00) :
: NEXT:PRINT#1
830 CLOSE1:CLOSE5
840 GOTO400

```

◆ AENDERUNGEN FUER 80-ZEICHEN-BS ◆

```

10 REM* ENTWURF VON SONDERZEICHEN *
20 REM* DURCH NADEL-ANSTEUERUNG *
30 :
40 REM ** 80-ZEICHEN-BILDSCHIRM **
50 :
75 PRINTCHR$(142):REM GRAFIK
80 AB=32893+120:REM* BILDSCH.-ADRESSE
550 BA=AB+(S*2)+(Z*160)
680 FORZ=6TO0STEP-1:BA=AB+(S*2)+(Z*160)

```

Programmbeschreibung

Die Zeilen bis 220 führen zur Anlegung der Variablen-tabelle, d.h. alle erforderlichen Variablen werden deklariert und im Arbeitsspeicher unter dem entsprechenden Namen archiviert. Die Anweisungen in den Programmzeilen 240 bis 300 bewirken die Darstellung der Eingabemaske auf dem Monitor des Computers, während durch die Befehle in 320 bis 380 die Erklärung der verfügbaren Steuertasten erreicht wird. Der wichtigste Programmteil befindet sich zwischen 390 und 640. Hier werden sämtliche Eingaben plausibilliert und die entsprechenden Steuerbefehle des Bedieners ausgeführt. In den Programmzeilen 650 bis 750 werden die Binärwerte der Druckmatrix errechnet. Dabei wird eine Prüfsumme durch Addition aller Werte ermittelt. Ist die Prüfsumme gleich Null, dann wurde kein Druckpunkt gesetzt und das Programm verzweigt in 750 zur Eingabemaske, ohne den Drucker anzusteuern. Sobald mindestens eine Nadel gesetzt war, geht das Programm zur Druckroutine, die sich in den Zeilen 770 bis 830 befindet. Der Rücksprung zur Eingabemaske wird durch den Befehl in Zeile 840 veranlaßt. Programmunterbrechung ist nur durch die »STOP«-Taste möglich.

Die Vielfalt möglicher Sonderzeichen und Symbole ist jetzt nur noch begrenzt durch Fantasie und Einfallsreichtum des jeweiligen Programmbenutzers.

Viel Erfolg beim Kreieren Ihrer eigenen Zeichen!

Code-Umwandlung

Code-Umwandlung

8-Kanal-BCD in dezimal

Computer sind bekanntlich in der Lage, sehr komplexe Probleme zu lösen und dabei die verschiedensten Zeichen zu »verarbeiten«. Intern werden aber alle Informationen in Form von Bitgruppen gespeichert, d.h. alle Daten liegen im Speicherbereich in verschlüsselter (codierter) Form vor. In der Elektronik benutzt man zur Informationsdarstellung die Wiedergabe der Zustände »0« und »1«. Dabei werden die Schaltungszustände »Ein« und »Aus« als die logischen Symbole »1« und »0« interpretiert.

Die Speicherung von Informationen auf Lochstreifen bedient sich ebenfalls dieser Darstellungsform. Hier werden die beiden Zustände »1« und »0« durch »Lochung« und »keine Lochung« wiedergegeben. Die meisten Computer arbeiten mit einer Bytelänge von 8 Bit, das bedeutet, alle Zeichen werden dort durch Aneinanderreihung und Kombination von acht Schaltungszuständen dargestellt. Manchmal ist es erwünscht, diese unmittelbar im Binärformat (Nullen oder Einsen, Lochung oder keine Lochung) vorhandenen Informationen in ihre leichter verständliche, dezimale Darstellungsform zu konvertieren. Eine »17« ist viel leichter zu erkennen als das Bitmuster »00010001«. Aus diesem Grund sind noch andere leichter verständliche Darstellungsformen entwickelt worden, die die Zusammenarbeit Mensch-Maschine verbessern helfen. Erwähnt seien hier die Oktal- und Hexadezimalsymbole sowie der ASCII-Code. Wir beschränken uns hier auf die dezimale Darstellungsweise, da sie in vielen Fällen bereits ein ausreichendes Hilfsmittel ist.

Man denke hierbei beispielsweise an das Lesen oder Korrigieren von Daten auf einem Lochstreifen, das Testen von Ein- oder Ausgabeports und überall dort, wo Informationen in Form von Bytes, also in Folgen von acht Bit (oder Kanälen), gespeichert werden.

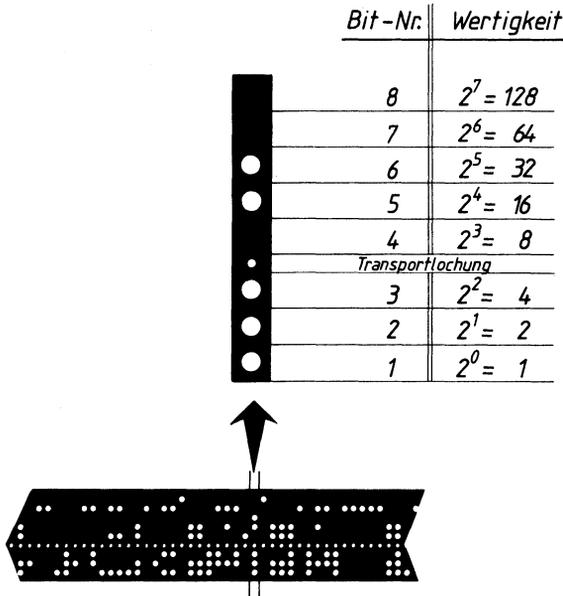


Bild 1: Informationen auf Lochstreifen

Problembeschreibung

Das Hauptproblem besteht darin, den vorhandenen 8-Kanal-BCD-(binary coded decimal)Code in Dezimalziffern umzuwandeln. Damit erreicht man aber lediglich eine Zwischenstufe zur benutzerfreundlichen Informationsdarstellung. Um nun zu erfahren, welches Zeichen, welche Zahl oder welcher Buchstabe sich hinter einzelnen Lochkombinationen verbirgt, ist eine weitere Umwandlung erforderlich. Hierzu benötigt der Benutzer in der Regel eine maschinenspezifische Umwandlungstabelle, in der jeder Dezimalzahl ein entsprechendes Zeichen zugeordnet ist. Dies liegt daran, daß die verschiedenen Lochstreifenmaschinen mit den unterschiedlichsten Codes arbeiten. In vielen Fällen reicht der Dezimalcode jedoch als aussagefähige Information aus.

Im Falle einer Korrektur von Daten auf einem Lochstreifen sind normal lediglich einzelne Bytes manuell abzuändern. Für den ungeübten Anwender ist es nicht einfach, genau die Stellen auf dem Streifen zu finden, an denen die Information durch Stanzen oder Überkleben einzelner Lochungen abgeändert werden muß. Um die eingestanzten Informationen ohne Lochstreifenleser »lesen« zu können, ist eine ständige Umwandlung in den Dezimalcode erforderlich.

Wie schon erwähnt, ist dies für den routinierten Anwender problemlos — dem Laien aber kann ein Computerprogramm die Arbeit wesentlich erleichtern. Einfache Handhabung und klare Bedienerführung sind die hauptsächlichsten Eigenschaften eines solchen Programmes, wenn es ein unkompliziertes Hilfsmittel darstellen soll. Dazu sind umfangreiche Fehlerabfrageroutinen und -meldungen in das Programm einzubauen, die logische Fehler verhindern und den Benutzer auf etwaige Eingabefehler aufmerksam machen. Nach Eingabe der jeweiligen Lochkombination soll sofort die Dezimalzahl errechnet und dem Anwender mitgeteilt werden. Dabei muß das Programm universell gestaltet werden, damit es auch für ähnliche Probleme wie die manuelle Lochstreifenkorrektur eingesetzt werden kann.

Problemlösung

Die oben beschriebene Problematik kann beispielsweise auf einem CBM-Computer rationell gelöst werden. Wie die meisten Tischrechner verfügt der CBM auch über ein numerisches Tastenfeld zur schnellen Eingabe von Zahlen und einen Bildschirm, auf dem die Ergebnisse zur Anzeige gebracht werden können. Damit sind die Voraussetzungen der Hardware zur einfachen, unkomplizierten Handhabung gegeben. Die Umwandlung des BCD-Codes in Dezimalcode kann mittels eines Basic-Programmes erfolgen, dessen Ablauf sich in die folgenden Teilprobleme untergliedert:

- Programmbeschreibung, Bedienungsanleitung und Beispiele, die jeweils zu Programmbeginn auf dem Monitor des Computers erscheinen;
- Bedieneringaben mit der Möglichkeit, das Programm zu beenden;
- Fehlererkennung durch Überprüfung der Eingabengänge sowie Kontrolle, ob Zahlen in aufsteigender Reihenfolge erfaßt sind;
- ggf. entsprechende Fehlermeldung mit dem genauen Hinweis, an welcher Stelle die Erfassung falsch ist, Rücksprung zur Erfassungsroutine;
- Errechnung und Anzeigen der jeweiligen Dezimalzahl auf dem Bildschirm, erneute Anwahl der Erfassungsroutine mit Beendigungsmöglichkeit des Programmes durch Eingabe von »ende«.

Der eigentliche Kern des gesamten Programmes, nämlich die Handhabung der Erfassungsroutine, muß hier näher erläutert werden. Betrachten wir hierzu folgendes Beispiel:

Bit-Nr./Kanal:	8	7	6	5	4	3	2	1
Wertigkeit:	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bitmuster:	0	1	1	0	0	1	0	1

Bedienereingabe: 1367
Bildschirm-Ausgabe: Dezimalziffer 101

Im obigen Beispiel soll die Umwandlung der Bitkombination 01100101 besprochen werden. Die Darstellung dieses Bytes (bestehend aus 8 Bit) auf einem 8-Kanal-Lochstreifen ist dadurch gekennzeichnet, daß anstelle der »Einsersbit« jeweils eine Lochung im entsprechenden Kanal (1, 3, 6 und 7) vorhanden sein muß. Um den Erfassungsaufwand für die Umwandlung so gering wie möglich zu halten, werden dem Computer lediglich die Bitnummern der Einsersbit in aufsteigender Reihenfolge eingegeben (Eingabe: 1367). Der Rechner addiert nun die Wertigkeiten der einzelnen Bit und gibt die errechnete Summe als Dezimalzahl aus ($2^0 + 2^2 + 2^5 + 2^6 = 1 + 4 + 32 + 64 = 101$).

Deutlich erkennt man die Allgemeingültigkeit des Umwandlungsprogrammes, wenn es darum geht, einzelne Byteinhalte (Bitkombinationen) in die entsprechende dezimale Darstellungsform zu konvertieren. Dabei beschränkt sich diese Problemlösung nicht ausschließlich auf 8-Kanal-Lochstreifen. Mit Hilfe dieses Programmes können beispielsweise auch Informationen auf einem, in der Praxis häufigen, 5-Kanal-Lochstreifen dezimal entschlüsselt werden. Wenn der Benutzer jedoch außer der Dezimalzahl auch den Zeichen- oder Befehlscode benötigt, so sind dazu speziell für die Lochstreifenkorrektur typenspezifische Umwandlungstabellen erforderlich. Anhand der Dezimalzahl kann aus solchen Tabellen die genaue Bedeutung einzelner Lochkombinationen ersehen werden. Der Anwender erfährt dadurch, welches Zeichen (Buchstabe, Zahl, Sonderzeichen) oder welcher Befehl verschlüsselt durch Bitkombination auf dem Lochstreifen dargestellt ist.



Bild 2: 5-Kanal-Lochstreifen

Bedienungsanleitung

Nach Einladen und Starten des Programmes mit »RUN« erscheint auf dem Monitor des CBM-Computers eine kurze Bedienungsanleitung. Das Programm übernimmt sofort die Bedienungsführung, indem es dem Benutzer die Bedeutung der jeweils erforderlichen Steuertasten über den Bildschirm mitteilt. Durch Drücken der »RETURN«-Taste gelangt man zum Unterprogramm »Beispiele«.

Dieser Programmteil erzeugt über den eingebauten Zufallsgenerator die verschiedensten 8-Bit-Kombinationen, simuliert deren Umwandlung und stellt die

dazugehörige Dezimalzahl dar. Auch ungeübte Anwender werden so in die Lage versetzt, ohne Bedienungshandbuch das Programm zu verstehen und mit ihm arbeiten zu können. Ist die Antwort auf die Ja/Nein-Abfrage »Weitere Beispiele?« negativ (Drücken der »N«-Taste), wird zur Eingaberoutine verzweigt. Der eingangs gestellten Forderung nach einfacher, unkomplizierter Handhabung wird das Programm gerecht, indem es hier lediglich die Eingabe der Bitnummern der Einserbit in aufsteigender Reihenfolge (Wertigkeit) verlangt. Wird an dieser Stelle »ende« eingegeben, erfolgt die Beendigung des Programmes.

Umwandlung: 8-Kanal-BCD in Dezimal

Dieses Programm ermöglicht auf elegante Weise die Umwandlung von 8-Kanal-BCD-Code in Dezimalziffern!

Bei Lochung im Kanal ? *? wird die Lochkombination pro Lochung in aufsteigender Reihenfolge eingegeben!

Bsp.: Bei Lochungen im Kanal 1,5,2,7 müesste eingegeben werden: **10001**
(Eingabe **ende** bedeutet Ende Programm)

Der Computer plausibilisiert die Eingaben (ggf. mit Fehlermeldung!) und errechnet die Dezimalziffer.

Weiter mit **Return**-Taste

Umwandlung: BCD in Dezimal

Beispiel:

Lochung in Kanal								Dezimal
1	2	3	4	5	6	7	8	
■								00000000
■								00000001
■								00000010
■								00000011
■								00000100
■								00000101
■								00000110
■								00000111
■								00001000
■								00001001
■								00001010
■								00001011
■								00001100
■								00001101
■								00001110
■								00001111
■	■							00010000
■	■							00010001
■	■							00010010
■	■							00010011
■	■							00010100
■	■							00010101
■	■							00010110
■	■							00010111
■	■	■						00011000
■	■	■						00011001
■	■	■						00011010
■	■	■						00011011
■	■	■						00011100
■	■	■						00011101
■	■	■						00011110
■	■	■						00011111
■	■	■	■					00010000
■	■	■	■					00010001
■	■	■	■					00010010
■	■	■	■					00010011
■	■	■	■					00010100
■	■	■	■					00010101
■	■	■	■					00010110
■	■	■	■					00010111
■	■	■	■	■				00010000
■	■	■	■	■				00010001
■	■	■	■	■				00010010
■	■	■	■	■				00010011
■	■	■	■	■				00010100
■	■	■	■	■				00010101
■	■	■	■	■				00010110
■	■	■	■	■				00010111
■	■	■	■	■	■			00010000
■	■	■	■	■	■			00010001
■	■	■	■	■	■			00010010
■	■	■	■	■	■			00010011
■	■	■	■	■	■			00010100
■	■	■	■	■	■			00010101
■	■	■	■	■	■			00010110
■	■	■	■	■	■			00010111
■	■	■	■	■	■	■		00010000
■	■	■	■	■	■	■		00010001
■	■	■	■	■	■	■		00010010
■	■	■	■	■	■	■		00010011
■	■	■	■	■	■	■		00010100
■	■	■	■	■	■	■		00010101
■	■	■	■	■	■	■		00010110
■	■	■	■	■	■	■		00010111
■	■	■	■	■	■	■	■	00010000
■	■	■	■	■	■	■	■	00010001
■	■	■	■	■	■	■	■	00010010
■	■	■	■	■	■	■	■	00010011
■	■	■	■	■	■	■	■	00010100
■	■	■	■	■	■	■	■	00010101
■	■	■	■	■	■	■	■	00010110
■	■	■	■	■	■	■	■	00010111
■	■	■	■	■	■	■	■	00010000
■	■	■	■	■	■	■	■	00010001
■	■	■	■	■	■	■	■	00010010
■	■	■	■	■	■	■	■	00010011
■	■	■	■	■	■	■	■	00010100
■	■	■	■	■	■	■	■	00010101
■	■	■	■	■	■	■	■	00010110
■	■	■	■	■	■	■	■	00010111
■	■	■	■	■	■	■	■	00010000
■	■	■	■	■	■	■	■	00010001
■	■	■	■	■	■	■	■	00010010
■	■	■	■	■	■	■	■	00010011
■	■	■	■	■	■	■	■	00010100
■	■	■	■	■	■	■	■	00010101
■	■	■	■	■	■	■	■	00010110
■	■	■	■	■	■	■	■	00010111
■	■	■	■	■	■	■	■	00010000
■	■	■	■	■	■	■	■	00010001
■	■	■	■	■	■	■	■	00010010
■	■	■	■	■	■	■	■	00010011
■	■	■	■	■	■	■	■	00010100
■	■	■	■	■	■	■	■	00010101
■	■	■	■	■	■	■	■	00010110
■	■	■	■	■	■	■	■	00010111

Weitere Beispiele ? (J/N)

Benutzereingaben werden durch Betätigen der »RETURN«-Taste in die Variable »L\$« übernommen. Falls die maximale Länge der Eingabe, nämlich acht Stellen, überschritten ist, meldet der Computer: »Zu lang — neue Eingabe« und verzweigt erneut zur Erfassungsroutine. Andernfalls erfolgt die Umwandlung in Dezimalzahlen. Das Programm überprüft dabei die aufsteigende Reihenfolge der eingegebenen Bitnummern. Im Fehlerfall wird die Umwandlung sofort unterbrochen und auf dem Monitor erscheint die Meldung: »Ungültige Eingabe!« Ein blinkender Pfeil markiert die fehlerhafte Bitnummer und fordert den Bediener auf, die Erfassung zu wiederholen. Bei gültiger Eingabe wird die Dezimalzahl errechnet und auf dem Bildschirm ausgedruckt.

◆◆◆ Code — Umwandlung ◆◆◆

```

100 rem * 8-Kanal-bcd in dezimal *
105 rem *****
110 :
115 diml(8)
120 foruu=1to8: l(uu)=int(2*(uu-1)):nextuu
125 rv$="" : rem* reverse on
130 ro$="R" : rem* reverse off
135 sl$="|" : rem* shift und eckige Klammer
140 ct$="▮" : rem* cursor tief
145 ch$="▯" : rem* cursor hoch
150 cl$="┌" : rem* cursor links
155 cr$="┐" : rem* cursor rechts
160 dimr$(2): r$(1)=rv$+chr$(32)+ro$
165 r$(2)=ro$+chr$(183)+rv$
170 :
175 printchr$(142): poke59468,14
180 :
185 gosub465
190 gosub360
195 printchr$(147)
200 print"Umwandlung BCD -> Dezimal !":print:print
205 print"Lochuns in Kanal: *"|cl$cl$cl$|":inputl$
210 ifl$="ende"thenend
215 ifl$="*"thenerintchr$::soto205
220 ll=len(l$)
225 ifll>8thenerprintot$rv$" zu lang -> neue Einsabe "ro$ct$::soto205
230 bc=0: fori=1to ll: cb=bc
235 bc=val(mid$(l$,i,1)): ifbc=0orbc>8then255
240 ifi>1thenifbc<=cbthen255
245 su=su+l(bc)
250 nexti: soto230
255 printot$ct$"Unsaeltise Einsabe: "rv$l$
260 print"#####": rem* 19mal cursor rechts

```

```

265 foruu=1to1:printchr$(32)chr$(94)c1$:next
270 printot$" --- Fehler!";gosub320:print:print
275 i=11:su=0:goto250
280 ifsu=0then300
285 print"_____":rem* 16mal chr$(185)
290 printrv$" Dezimalziffer: "ro$:su
295 print"_____":rem* 16mal chr$(184)
300 su=0
305 foruu=1to39:print"-":next:print
310 goto205
315 :
320 rem *blink*
325 foryz=1to8
330 foryy=1to100:next
335 foruu=1to8:printo1$:next:printrv$" Fehler!"ro$:
340 foryy=1to100:next
345 foruu=1to8:printo1$:next:print" Fehler!";
350 nextyz:return
355 :
360 rem * beispiele *
365 printchr$(147)cr$:
370 print"Umwandlung:   BCD   in   Dezimal  "
375 printot$
380 print"Beispiel:":print"_____ "
385 print"Lochuns in Kanal 1   Dezimal"
390 print" |-----|
395 print" |1|2|3|4|5|6|7|8| |"
400 foryy=1to12:print" r$(1):su=0
405 foryz=1to8
410 bc=int(1+(2*rnd(1)))
415 printr$(bc)r$(1):
420 ifbc=2thensu=su+1(yz)
425 nextyz:printsl$:spc(7-len(str$(su))):su
430 nextyz
435 printchr$(32)rv$"          "ro$:sl$:rem17 space
440 printot$"Weitere Beispiele ?   (J/N)"
445 getx$:ifx$=""then445
450 ifx$="j"then360
455 ifx$<>"n"then445
460 su=0:return
465 rem* beschreibung *
470 printchr$(147)ct$rv$" Umwandlung:R 8-Kanal-BCD in Dezimal"
475 printot$ct$
480 print"Dieses Programm ermoeslicht auf elesen-"
485 print"te Weise die Umwandlung von 8-Kanal-"
490 print"BCD-Code in Dezimalziffern!"ct$
495 print"Bei 'Lochuns im Kanal ? *' wird die"
500 print"Lochkombination pro Lochuns in aufstei-"
505 print"sender Reihenfolge eingegeben!"
510 printot$ct$"Esp.: Bei Lochunsen im Kanal 1,3,2,7"
515 print"muesste eingegeben werden 01257R"
520 print"(Einsabe 0ender bedeutet Ende Programm)"

```

```

525 printot$"Der Computer klausuliert die Einsaben"
530 print"(gef. mit Fehlermeldung!) und errechnet"
535 print"die Dezimalziffer."
540 printot$ot$" # Weiter mit 3'Return'R-Taste #"
545 betx$:ifx$=""then545
550 ifx$Cchr$(13)then545
555 return

```

Programmbeschreibung

Das Basicprogramm »Code-Umwandlung« läuft ohne Änderungen auf allen CBM-2000-, -3000-, -4000- und -8000-Systemen. Es ist durch Anpassung der Bildschirmformatierung auch auf VC 20 lauffähig. Beachten Sie, daß für VC die Zeile 175 entfernt werden muß. Aber auch auf ähnlichen basicprogrammierbaren Rechnern müßte die Code-Umwandlung mühelos zu installieren sein. Abzuändern wäre lediglich Bildschirmausgabe sowie die Textformatierung.

Die Programmzeilen 115 und 120 dienen zur Anlegung der Wertigkeitstabelle. In 125 und 165 erfolgt die Definition wichtiger Variablen, die zur Cursorsteuerung und Textformatierung benötigt werden. Anschließend geschieht die Umschaltung des Zeichengenerators in Textmodus (Zeile 175). Der »GOSUB«-Befehl in 185 bewirkt die Abarbeitung des Unterprogrammes »Beschreibung« (465 bis 555). Dieser Programmteil dient lediglich zur Darstellung einer Kurz-Bedienungsanleitung auf dem Monitor. In Zeile 190 wird zum Unterprogramm »Beispiele« verzweigt (360 bis 460). Dieser Teil des Umwandlungsprogrammes generiert immer wieder neue Bitmuster per Zufall; daher kann die Beispiel-Routine stets zu Lernzwecken herangezogen werden.

Aber nun zur Umwandlungsroutine, die sich in den Programmzeilen 195 bis 310 befindet. Hier werden alle Eingaben plausibiliert, ggf. mit Fehlermeldung, und in Dezimalcode umgewandelt. Während die Zeile 195 lediglich die Bildschirmlöschung bewirkt, fordert der Computer in 205 durch die Meldung: »Lochung in Kanal: ★« zur Benutzereingabe auf. Die Abfrage in Zeile 215 verhindert dabei, daß der Rechner bei »INPUT« durch Betätigen der »RETURN«-Taste aus dem Programm springt. Wie Sie unschwer erkennen können, geschieht die Fehlerabfrage, ob Eingabe (»L\$«) länger als acht Stellen in 220 und 225. Die »FOR ... NEXT«-Schleife zwischen 230 und 250 kontrolliert, ob die Bit- oder Kanalnummern in aufsteigender Reihenfolge erfaßt sind. Im Fehlerfall wird zur Zeile 255 verzweigt. Die Befehle in den folgenden Programmzeilen geben eine präzise Fehlermeldung auf dem Bildschirm aus, wobei ein Pfeil auf die fehlerhafte Bitnummer zeigt und den Bediener zur Korrektur auffordert. Die kleine Routine in 320 und 350 verursacht ein auffälliges Blinken der Fehlermeldung. Der Anwender ist in der Lage, etwaige Eingabefehler schnell zu erkennen und dementsprechend zu reagieren.

Horizontale Balkendiagramme

Horizontale Balkendiagramme

Die Statistik ist heute in nahezu allen Gebieten der Wissenschaft und Praxis ein unentbehrliches Hilfsmittel geworden. Mit ihrer Hilfe ist es möglich, Massenerscheinungen größenmäßig zu bestimmen, zu vergleichen und mittels Auszählung, Tabellierung und Darstellung einer analytischen Beobachtung zugänglich zu machen.

Mit der Verbreitung der Personal Computer gewinnt die Statistik auch in der betrieblichen Praxis immer mehr an Bedeutung. Dient sie doch einerseits zur Kontrolle vergangener und andererseits der Erleichterung zukünftiger Entscheidungen. Auch zum Zwecke der Werbung, also für Mitteilungen an die Presse beispielsweise, findet die betriebliche Statistik Verwendung. In vielen Fällen genügt es, betriebliches Zahlenmaterial, das schon vorhanden ist, zusammenzutragen und in Statistik umzuformen. Oft erfordert die Anfertigung neuer Statistiken die eigene Erfassung bestimmter Daten und deren Auswertung. Personal Computer-Systeme bieten vielfach eine kostengünstige Lösung, wenn es darum geht, bestimmte Teilgebiete im Betrieb statistisch zu durchleuchten oder eine individuelle Analyse bestimmter Daten vor Ort, also am Arbeitsplatz, durchzuführen. Diese Aufgabenstellung beschränkt sich nicht nur auf mathematisch-wissenschaftliche Problemlösungen, es gibt viele kommerzielle Einsatzgebiete, in denen die statistische Datenanalyse zur individuellen Kontroll- und Planungshilfe herangezogen werden kann. Es ist unmöglich, alle Anwendungsgebiete erschöpfend aufzuzählen. So sind beispielsweise allein zum Zwecke der Verkaufsförderung und Materialdisposition die unterschiedlichsten Auswertungen möglich: Umsatz- und Absatzstatistiken, gegliedert nach Ländern, Kunden oder Branchen, aufgeschlüsselt und sortiert nach Produktgruppen, Deckungsbeitrag oder prozentualem Umsatzanteil.

* UMSATZ-STATISTIK *

DM-UMSATZ, MONATSWEISE
01/82 -- 12/83

PRODUKT: EDV-ZUBEHOER

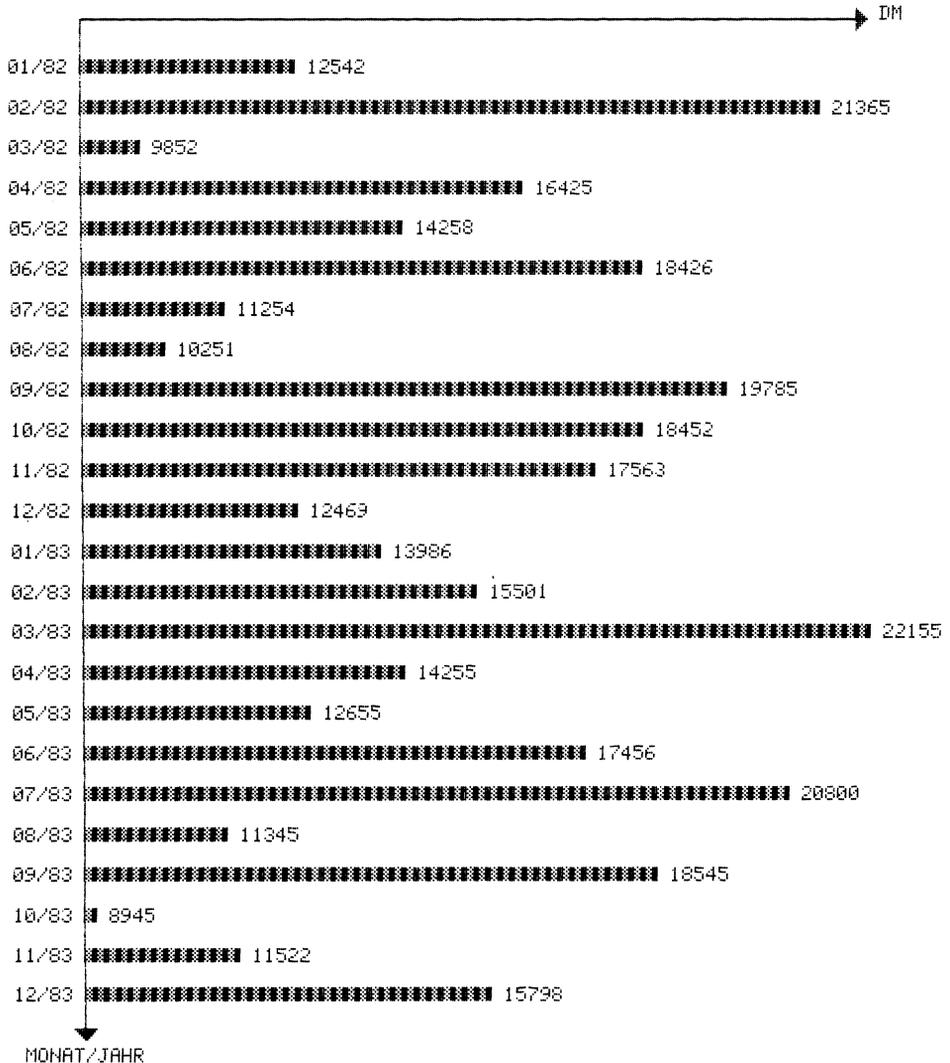


Absolute Balkenlänge

* UMSATZ-STATISTIK *

DM-UMSATZ, MONATSWEISE
01/82 -- 12/83

PRODUKT: EDV-ZUBEHOER



Relative Balkenlänge

Die Darstellung der aufbereiteten Ergebnisse statistischer Erhebungen dient dazu, den Benutzer in Form von Tabellen und Grafiken in übersichtlicher und prägnanter Form über die beobachteten Merkmale zu informieren. Dabei ist die grafische Darstellung des Zahlenmaterials zum Teil sehr viel anschaulicher und informiert den Anwender manchmal besser, als dies eine tabellarische Auflistung der Ergebnisse vermag. Eine mögliche grafische Darstellung statistischer Merkmale ist das Balkendiagramm. Zum Erstellen von Balkendiagrammen auf einem Personal Computer ist außer einem Programm eine Hardware-Konfiguration, bestehend aus Rechner, Kassettenrecorder oder Floppy-Disk und Drucker, nötig.

Problembeschreibung

Oft ist es notwendig, die grafische Darstellung statistischer Merkmale nicht nur auf dem Computermonitor zu sehen, sondern auch zu Papier zu bringen. Bei Systemen, die über einen Hardcopy-Befehl verfügen, kann man dies ohne Programmieraufwand schnell erreichen; per Tastendruck kann der Bildschirminhalt auf den angeschlossenen Drucker ausgegeben werden. Die begrenzte Zeilenbreite vieler Rechnerbildschirme läßt jedoch eine vernünftig aufgebaute Grafik nicht zu. In solchen Fällen kommt man nicht umhin, die Druckerausgabe zu programmieren, will man eine plastische Illustration des statistischen Zahlenmaterials zu Papier bringen, um sie anschließend eingehender studieren zu können.

Balken- oder Säulendiagramme werden bevorzugt vertikal dargestellt, für die programmgesteuerte Erstellung grafischer Statistiken aber bieten sich waagrechte Balken geradezu an. Übersichtliche Darstellung, beliebig große Anzahl statistischer Werte und relativ einfache Programmierung sind die wesentlichsten Vorteile der horizontalen gegenüber der vertikalen Darstellung von Balkendiagrammen.

Welche grundlegenden Anforderungen werden an ein Computerprogramm gestellt, um beispielsweise die monatlichen DM-Umsätze eines bestimmten Produktes durch horizontale Balken grafisch zu veranschaulichen?

- Eingabe der statistischen Merkmalswerte (DM-Umsätze);
- Individuelle Festlegung des Betrachtungszeitraumes muß möglich sein;
- Maßstab für die darzustellende Balkenlänge soll per Programm errechnet werden;
- Programmgesteuerter Druckvorgang unter Ausnutzung der grafischen Möglichkeiten des angeschlossenen Printers;

— Parametergesteuerte Programmanpassung für jeden angeschlossenen Nadel-
drucker der verschiedensten Breiten.

BALKEN - DIAGRAMME

DIESES PROGRAMM ERSTELLT BALKENDIA-
GRAMME AUF DEM DRUCKER CBM 3022.

BITTE GEBEN SIE HIER DIE ANZAHL DER MO-
NATE DES BETRACHTUNGSZEITRAUMES EIN:

EINGABE: ? 24

NUN MUESSEN SIE DEN LETZTEN MONAT EIN-
GEBEN (FORMAT: **XXXX**, BSP.: **0282**):

EINGABE: ? 1282

FORMULAR AUF DRUCKER AUSGABEN

WEITER MIT **RETURN**

ABSOLUTE
BALKENLAENGE
 RELATIVE

DRUCKWEGANG

Die oben angeführten Punkte gelten allgemein für alle Computer-Systeme und bilden lediglich das Gerüst der gesamten Aufgabenstellung. Es folgt eine detaillierte Beschreibung der Teilaufgaben, doch wollen wir uns zuvor auf einen bestimmten Computertyp festlegen: Ein gutes Preis-/Leistungsverhältnis, die leicht erlernbare Programmiersprache »BASIC« und der bereits standardmäßig druckbare Grafikzeichensatz prädestinieren Commodore-Anlagen geradezu für solche Einsatzgebiete.

Problemlösung

Grundvoraussetzung ist, daß die Software auf allen CBM-Anlagen ohne große Modifikationen läuft. Deshalb sollten »PEEK«- und »POKE«-Befehle möglichst vermieden werden. Auch die, ausschließlich im Betriebssystem »Basic 4.0« ver-

wendeten, Disketten-Statusvariablen »DS« und »DS\$« sollten vom Programmierer nie anderweitig benutzt werden. Bei Verwendung der in Basic 4 vorhandenen Floppy-Kurzbefehle meldet die Diskettenstation nach jedem Diskettenzugriff einen Fehlerstatus an den Rechner. Dieser kann dann aus den Variablen »DS« (Fehlernummer) und »DS\$« (Fehlertext) gelesen werden. Beim Versuch, »DS« oder »DS\$« unter Basic 4 einen Wert zuzuweisen, läuft das System auf »SYNTAX ERROR«. Da Basic 3 die Floppy-Kurzbefehle nicht kennt, darf man dort Wertzuweisungen mit »DS« und »DS\$« vornehmen. Mit anderen Worten, die einzelnen Betriebssysteme sind leider nicht 100%ig aufwärtskompatibel. Wir wollen an dieser Stelle nicht näher auf die verschiedenen Rechnerbetriebs- und Diskettenoperations-Systeme eingehen, deshalb nur ein kurzer Tip: Basicprogramme für PET/CBM-Anlagen können unter Einhaltung der obigen Regeln so entwickelt werden, daß sie für alle Betriebssystem-Versionen zumindest aufwärtskompatibel sind.

Da das Programm »Horizontale Balkendiagramme« keine Floppy-Kommandos enthält, war hier die Voraussetzung nach Kompatibilität unschwer zu erfüllen. Die abgebildete Software läßt keine programmgesteuerte Datenerfassung über die Tastatur zu. Zur besseren Demonstration wurden die statistischen Merkmalswerte in DATA-Statements im Programm verankert. Ferner bieten diese Programme die

- freie Wahl von Lage und Länge des Betrachtungszeitraumes (eingeteilt in Monate);
- eine spezielle Routine zur Errechnung und Zuordnung von Monat und Jahr zu jedem Merkmalswert;
- automatische Maßstabsberechnung der Balkenlänge durch Ermitteln von Minimum und Maximum innerhalb der statistischen Wertreihe;
- Druckersteuerung für CBM-Printer der Serien 302X und 402X;
- Ausdruck eines, von Betrachtungszeitraum und Druckerbreite abhängigen, Koordinationskreuzes einschließlich der dazugehörigen Beschriftung;
- Einzeichnen der Balken in das Diagramm mit Angabe der Ganzzahlwerte der statistischen Merkmale;
- parametergesteuerte Programmanpassung an angeschlossene Nadeldrucker der verschiedensten Zeilenbreite;
- durch geringfügige Programmänderung erreichbare Anpassung an nahezu alle Drucker mit den verschiedensten Zeichensätzen.

Darüber hinaus bietet das Programm noch die Wahl zwischen »relativer« und »absoluter« Balkenlänge. Interessiert sich der Benutzer in erster Linie für die einzelnen Merkmalswerte und deren Größenverhältnisse, dann wählt er die abso-

lute Darstellungform. Die relative Darstellung dagegen veranschaulicht die Größenordnung der Differenzen zwischen den einzelnen statistischen Werten. Für die Maßstabberechnung bedeutet dies: Um die Absolutlänge der Balken zu erreichen, muß die jeweilige Druckerzeilenbreite mit dem größten statistischen Wert ins Verhältnis gesetzt werden. Das führt bei einer weitgehend homogenen Wertereihe zu einer unbefriedigenden Abbildung des Diagramms, da vorhandene Schwankungen nur sehr schwach optisch sichtbar werden. Die relative Darstellung kann solche Schwankungen optimal zum Ausdruck bringen, da hier die Differenz zwischen Größt- und Kleinstwert als Maßstab für das Balkendiagramm herangezogen wird. So lassen sich Unterschiede und Tendenzen unter Ausnutzung der gesamten Druckerbreite genauer erkennen und deuten.

Die vorhandene Druckerbreite kann dem Computer durch Veränderung eines Parameters im Programm mitgeteilt werden. Bei Printern, die keine grafischen Zeichen ausdrucken, müssen die verwendeten Commodore-Grafiksymbole durch geeignete druckbare Zeichen ersetzt werden. Die Erstellung von Balkendiagrammen muß also nicht daran scheitern, daß Sie keinen CBM-Nadelprinter der Serien 302X oder 402X an Ihr System angeschlossen haben.

Bedienungsanleitung

Nach dem Starten des Programmes mit »RUN« wird der Bediener aufgefordert, die Anzahl der Monate des Betrachtungszeitraumes einzugeben (Beispiel: 24). Anschließend wird der letzte zu betrachtende Monat im Format »MMJJ« eingegeben (Beispiel: letzter Monat ist Dezember 1982, dann Eingabe: 1282). Jetzt liest der Rechner die Monatswerte (DM-Umsatz) aus den DATA-Statements ein und errechnet über eine spezielle Formel die einzelnen Monate (12/82, 11/82, 10/82 ...). Vorher muß darauf geachtet werden, daß die statistische Wertereihe folgerichtig in den DATA-Statements abgelegt wird. Der erste Wert innerhalb der DATAs entspricht gleichzeitig dem ersten Balken innerhalb des Diagramms. Das Programm überprüft nicht die Zugehörigkeit der Monate zu den DM-Umsätzen, sondern entnimmt lediglich die zeitliche Reihenfolge aus der Anordnung der im Programm verankerten Werte, ausgehend vom ersten Wert.

Durch Betätigen der Taste »A« kann die absolute Balkenlänge, durch Drücken der »R«-Taste die relative Balkendarstellung gewählt werden. Bei eingeschaltetem Drucker beginnt das Programm nach der Maßstabberechnung mit dem Ausdruck des horizontalen Balkendiagramms. Dabei werden die statistischen Werte mit dem Maßstab ins Verhältnis gesetzt und deren Balkenlänge errechnet.

◆◆◆ BALKEN-DIAGRAMME ◆◆◆

```

10 REM* S T A T I S T I K*
20 REM* BALKEN-DIAGRAMME *
30 REM* - HORIZONTAL - *
40 .
50 REM *****
60 .
70 DEFFNKB(KS)=INT(KS*10+.5)/INT(10)
80 POKE59468,12:REM* GROSS/GRAFIK
90 .
100 REM* VARIABLENBELEGUNG *
110 ZZ=80:REM* DRUCKERBREITE (ZEICHEN/ZEILE)
120 MA=25:DIMM(MA):REM* DIM F. MONATSWERTE
130 AM=0:REM* ANZAHL MONATE
140 DIMLM#:REM* LETZTER MONAT
150 CR$=CHR$(13):LE$=CHR$(1)
160 .
170 PRINT"■■■■BALKEN - DIAGRAMME■■"
180 PRINT"■■■■DIESES PROGRAMM ERSTELLT BALKENDIA-
190 PRINT"■■■■GRAMME AUF DEM DRUCKER CBM 3022.■■"
200 PRINT"■■■■BITTE GEBEN SIE HIER DIE ANZAHL DER MO-
210 PRINT"■■■■NATE DES BETRACHTUNGSZEITRAUMES EIN.■■"
220 INPUT"■■■■EINGABE: 0 ■■■■■■":AM
230 IFAM=0THENPRINT"■■■■":GOTO220
240 IFAM>MATHENPRINT"■■■■":GOTO220
250 DIMAM$(AM):REM* MONATE BSP.: 12/81
260 PRINT"■■■■NUN MUESSEN SIE DEN LETZTEN MONAT EIN-
270 PRINT"■■■■GEBEN (FORMAT: ■■■■MJJ■, BSP.: 30282■■):"
280 INPUT"■■■■EINGABE: * ■■■■■■■■":LM$
290 IFLM$="*"THEN340
300 IFLEN(LM$)<>4ORVAL(LM$)=0THEN340
310 L0=VAL(LEFT$(LM$,2)):L1=VAL(MID$(LM$,3))
320 IFL0<1ORL0<12ORL1<60THEN340
330 GOTO350
340 PRINT"■■■■":GOTO280
350 PRINT"■■■■FORMULAR AUF DRUCKER AUSRICHTEN! ■■"
360 .
370 PRINT"■■■■■■■■■■ WEITER MIT ^RETURN■■"
380 GETX$:IFX$<>CHR$(13)THEN380
382 PRINT"■■■■A■■■ABSOLUTE■■■BALKENLAENGE"
384 PRINT"■■■■R■■■RELATIVE"
385 GETX$:IFX$="R"THEN390
386 IFX$<>"A"THEN385
387 AD%=1:REM *FLAG FUER "ABSOLUTE"*
390 PRINT"■■■■■■ DRUCKVORGANG! ■■"
400 REM*READ DATA UND ERRECHNEN DER MONATE*
410 FORL=1TOAM:READM(L):NEXTL
420 .
430 MN$=LM$:MN=INT(VAL(MN$)+(10*2))
440 .
450 REM*ERRECHNEN DER MONATE*
460 FORL=AMTO1STEP-1

```

```

470 MN=MN-(10↑2):IFMNC100THENMN=MN+1199
480 LM#=RIGHT$(STR$(MN),4):IFLEFT$(LM#,1)=" "THEN
:LM#="0"+MID$(LM#,2)
490 AM$(L)=LEFT$(LM#,2)+"/"+MID$(LM#,3)
500 NEXTL
510 :
520 REM*SUCHEN GROESSTE UND KLEINSTE ZAHL*
530 FORL=1TOAM:IFL=1THENQ1=M(L):Q2=Q1
540 IFM(L)>Q1THENQ1=M(L)
550 IFM(L)<Q2THENQ2=M(L)
560 NEXTL:IFAD%THEN:AD%=0:Q2=0
570 Q3=Q1-Q2:IFQ3=0THENQ3=1E-10
580 :
590 REM*DRUCKER-EROEFFNUNG*
600 OPEN1,4:OPEN14,4,4:PRINT#14:REM*ERRORFILE
610 OPEN6,4,6:PRINT#6,CHR$(24):REM* 6LPI
620 :
630 PRINT#1,LE$LE$" * UMSATZ-STATISTIK *"
640 GOSUB650:GOTO660
650 FORQ0=1TO79:PRINT#1," "):NEXT:PRINT#1:RETURN
660 PRINT#1,CR$LE$" DM-UMSATZ, MONATSWEISE"
670 PRINT#1,LE$" "AM$(1)" -- "AM$(AM)
680 GOSUB650
690 :
700 PRINT#1,CR$"PRODUKT:"LE$LE$" EDV-ZUBEHOER"
710 PRINT#1,CR$:PRINT#6,CHR$(18):REM* 8LPI
720 PRINT#1,CR$" "
730 FORQ0=1TO22-15:PRINT#1,"_"):NEXT
740 PRINT#1," "DM"
750 PRINT#1," "
760 FORQ0=1TO22-15:PRINT#1,"?"):NEXT
770 PRINT#1," "
780 PRINT#1," "
790 PRINT#1," "
800 :
810 FORL=1TOAM
820 Q4=((M(L)-Q2)*(22-7-LEN(STR$(Q1)))/Q3)+1:Q4=FNKB(Q4)
830 Q8#="":FORQ0=1TOQ4:Q8#=Q8#+":":NEXT:Q8#=Q8#+":#"
840 PRINT#1," "AM$(L)" "Q8$:INT(M(L))
850 PRINT#1," "
860 PRINT#1," "
870 NEXTL
880 :
890 PRINT#1," "▼":PRINT#6,CHR$(24):REM* 6LPI
900 PRINT#1," MONAT/JAHR"
910 PRINT#1:CLOSE1:CLOSE6:CLOSE14:END
920 :
930 REM* MONATSWERTE
940 DATA12542.20,21365.41,9852.32,16425.17,14258.14
950 DATA18426.85,11254.46,10251.63,13785.23,18452.84
960 DATA17563.89,12469.85,13986.89,15501.4,22155.10
970 DATA14255.25,12655.55,17456.95,20800.45,11345
980 DATA18545.14,8945.1,11522.41,15798.33,16825.1,,,,,

```

Programmbeschreibung

Das abgebildete Basicprogramm arbeitet ohne Änderungen auf allen CBM 2000, 3000, 4000 und 8000 mit angeschlossenem Printer CBM 302X oder 402X. Bei der Erstellung des Programms wurde darauf geachtet, daß es ohne größere Modifikationen auch auf nahezu allen angeschlossenen Nadeldruckern der verschiedenen Breiten läuft. Durch Veränderung des Parameters »ZZ« in Zeile 110 kann das Diagramm der Druckerbreite angeglichen werden. Dieser Wert darf die maximale Anzahl der Zeichen pro Zeile (bei 10 Pitch) nicht überschreiten. In Zeile 830 kann durch Änderung des Strings »Q8\$« (Beispiel: ...:Q8\$ = Q8\$ + ''X'' :...) erreicht werden, daß die Balken mit jedem beliebigen Zeichen gebildet werden. Bei Druckern, die keine grafischen Zeichen ausdrucken, müssen diese in den Zeilen 650, 730 bis 790 und 830 bis 890 durch geeignete druckbare Zeichen ersetzt werden (Beispiel: Zeile 780 PRINT# 1, ''I'').

Nach der Umschaltung des Zeichengenerators in Zeile 80 folgt die Definition der wichtigsten Variablen (100 bis 150). In 170 bis 240 wird die Monatsanzahl des Betrachtungszeitraumes eingegeben. Der letzte Monat kann in den Programmzeilen 250 bis 340 festgelegt werden. Ob die absolute oder relative Balkenlänge gewählt ist, entscheidet sich zwischen 380 und 390. Nach Einlesen der DATA-Statements (Schleife in Zeile 410) erfolgt die Berechnung der Monate (430 bis 500). Nun wird der größte und kleinste Wert innerhalb der statistischen Wertereihe gesucht (520 bis 570). Druckereröffnung und Ausdrucken des Diagrammkopfes erfolgt in 590 bis 790. Der wohl wichtigste Programmteil befindet sich in den Zeilen 810 bis 870. Hier geschieht Berechnung und Ausdruck der einzelnen Balken unter Berücksichtigung des Maßstabes, der Druckerbreite und der Höhe des jeweiligen DM-Umsatzes. Ab der Programmzeile 940 stehen hinter DATA die einzelnen statistischen Merkmalswerte.

Laufbandanzeige

Laufbandanzeige

Ähnlich den konventionellen Werbemitteln wie Prospekten, Anzeigen, Plakaten usw. kann auch der Bildschirm eines Mikrocomputers äußerst kostengünstig dazu eingesetzt werden, werbespezifische Informationen über ein Produkt, eine Sache oder eine Dienstleistung einer möglichst breiten Ziel- oder Interessengruppe optisch näherzubringen.

Ein Mikrocomputer-System kann also als Träger von Werbebotschaften im kommunikativen Sinne dazu dienen, z. B. sich selbst, die dazugehörige Software und Peripherie einer bestimmten Personengruppe verkaufsfördernd vorzustellen. Dies kann aber nicht die Hauptaufgabe eines Computers sein. Vielmehr sollten damit eventuelle Standzeiten und längere Benutzerpausen des Systems, hauptsächlich in den Bereichen Verkauf und Werbung sowie auf Messeständen sinnvoll und akquisitiv überbrückt werden. Mit Hilfe eines Programmes ist es dem Benutzer innerhalb kürzester Zeit möglich, längere Werbetexte in großen Buchstaben horizontal über den Bildschirm des Computers laufen zu lassen. Der Bediener ist dadurch in der Lage, durch optimale Gestaltung und Kombination der werbewirksamen Bestandteile eine möglichst günstige Kontaktsituation herzustellen.

Durch die effektive Ausnutzung der Zeit, in der die Computeranlage sowieso eingeschaltet, aber ungenutzt wäre, entstehen praktisch keinerlei Zusatzkosten. Außerdem kann ein solches Computerprogramm natürlich überall dort eingesetzt werden, wo die Darstellung von Texten (bestehend aus Buchstaben, Zahlen und Sonderzeichen) über eine größere Distanz (mehrere Meter) zum Bildschirm erkennbar und gut lesbar sein muß. Dies ist der Fall, wenn der Computer z.B. Kontroll- und Überwachungsfunktionen ausübt und seine Resultate visuell ausgibt, um dem Benutzer so zu ermöglichen, dementsprechend zu reagieren. Dabei ist Bedingung, daß, wie schon oben erwähnt, die Großschrift nicht vertikal (wie in vielen herkömmlichen Programmen), sondern horizontal von rechts nach links über den Bildschirm wandert, um eine optimale Lesbarkeit zu gewährleisten.

Da die normale Zeichengröße des Computers nur auf kurze Distanz erkennbar ist, muß ein solches Programm die Buchstaben, Zahlen und Sonderzeichen um ein Vielfaches (im Beispiel 8fach) vergrößert darstellen können. Eine Zeichenvergrößerung ist aber nicht einfach per Befehl zu erreichen. Es muß vielmehr ein entsprechender Zeichensatz definiert werden, der sich dann nach dem Starten des Programmes im Arbeitsspeicher zur Verfügung hält. Als Generator und zum Zweck der Abspeicherung der Daten wird hier eine 8 x 7-Matrix benutzt.

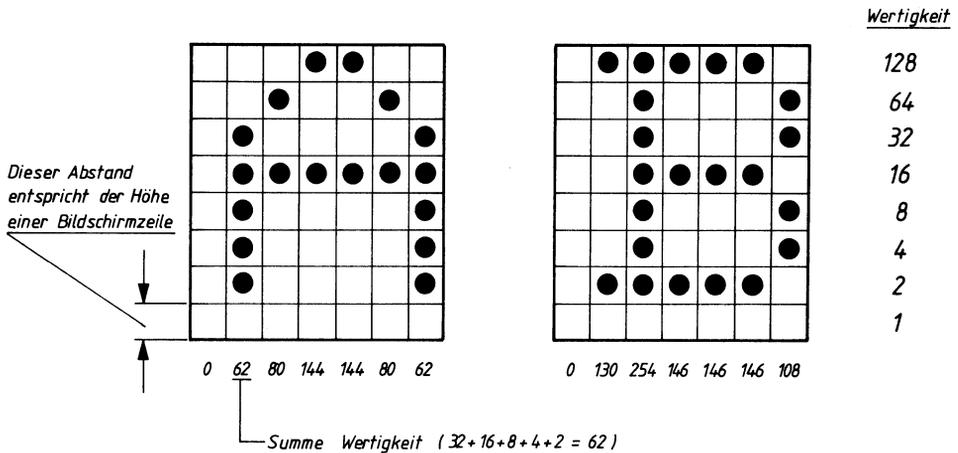


Bild 1: 8 x 7-Matrix

Bild 1 zeigt anhand der Buchstaben »A« und »B«, wie der Zeichensatz in der 8 x 7-Matrix dargestellt und durch vertikale Addition der Wertigkeiten dezimal verschlüsselt wird. »A« wird also durch folgende sieben Zahlen im Arbeitsspeicher dargestellt: 0, 62, 80, 144, 144, 80, 62. Eine solche Vorgehensweise erlaubt das Entwerfen von eigenen Sonderzeichen und Schriftarten. Das vorliegende Beispiel beschränkt sich jedoch auf die Buchstaben des Alphabets, die Zahlen 0 bis 9, ein paar mathematische Kurzzeichen sowie einige Sonderzeichen. Bezogen auf den Zeichenvorrat des CBM betrifft dies den Bereich von CHR\$(32) bis CHR\$(90).

Des weiteren muß für den Benutzer eine Eingabemöglichkeit geschaffen werden, um möglichst komfortabel individuelle Textteile zu erfassen, die dann, zumindest für den Bereich Werbung, zu einem »Slogan« komponiert werden können. Hierzu ist zunächst eine Input-Routine notwendig, die nur solche Eingaben akzeptiert, die innerhalb des definierten Zeichensatzbereiches liegen (in unserem Falle alle Zeichen zwischen CHR\$(32) und CHR\$(90)).

Für die Texterfassung wird folgender Ablauf festgelegt:

- Benutzer-Mitteilung an den Computer, wie viele Sätze erfaßt werden sollen;
- programmkontrollierte Eingabe der Sätze, dabei Überprüfung jedes Zeichens auf Gültigkeit;
- durch den Bediener kann für jeden Satz individuell die Wiederholungsrate festgelegt werden;
- Fehlerabfrage mit Korrekturmöglichkeit.

Ferner sind folgende Steuerungsfunktionen vorhanden:

- Eingabe eines Faktors, der den zeitlichen Ablauf der Darstellung auf dem Bildschirm steuert (Geschwindigkeitsfaktor);
- Abfrage, ob Laufband negativ (invertiert) abgebildet werden soll;
- Unterbrechungsmöglichkeit des »laufenden Bandes«, um oben genannte Parameter entweder zu ändern oder das Programm zu beenden.

Außerdem kann durch Abändern von zwei weiteren Parametern (Bildschirmcodes) im Programm jedes gewünschte Zeichen für die Schriftbildung und den Hintergrund des Laufbandes benutzt werden.

Wir besitzen nun ausreichend Möglichkeiten, um einerseits dem Benutzer die notwendige Gestaltungsfreiheit zu gewährleisten. Andererseits erlaubt eine durchdachte Programmorganisation Teilroutinen zu isolieren und ohne Modifikation für andere Problemlösungen einzusetzen oder durch kleine Abänderungen speziellen Anforderungen anzupassen.

Wie lösen wir jetzt das Hauptproblem, nämlich die Darstellung der Großbuchstaben und die Aktivierung des Laufbandes durch Verschieben der Zeichen nach links?

Betrachten wir hierzu die einzelnen Teilschritte der oben gestellten Problematik:

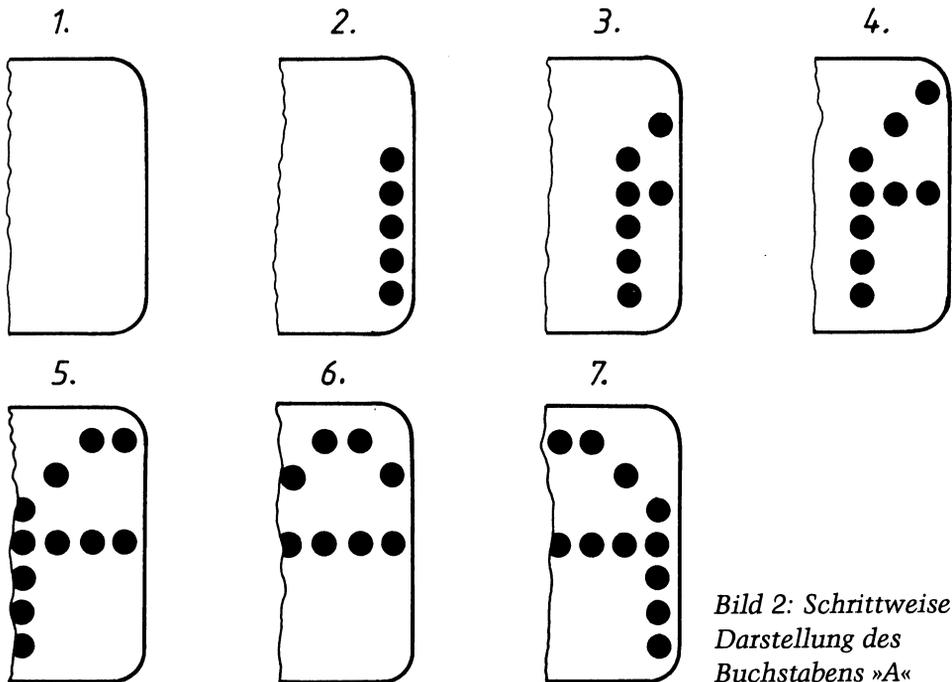


Bild 2: Schrittweise Darstellung des Buchstabens »A«

Bild 2 zeigt schrittweise die Darstellung des Buchstabens »A«. Nun wird klar, warum der Zeichensatz in der 8 x 7-Matrix durch die vertikale Addition der Wertigkeit verschlüsselt wurde. Deutlich erkennt man die vertikale Spaltenaufteilung der Großbuchstaben und die Verschiebung des Bildschirminhaltes nach links.

Zunächst einmal muß der Bildschirminhalt im Bereich des Laufbandes (Bildschirmzeilen 9 bis 16) um eine Spalte nach links verschoben werden. Hierbei wird die äußerst linke Spalte überschrieben. Anschließend erfolgt die Übergabe der ersten Schlüsselzahl des Buchstabens »A«. Da dies eine Null ist, füllt der Computer die rechte Bildschirmspalte mit dem Code für den Hintergrund aus. Erneut verschiebt das Programm den Bildschirminhalt des Laufbandes nach links und holt sich nun die zweite Schlüsselzahl (62). Dies veranlaßt den Computer, in der rechten Spalte die Laufbandzeilen 3, 4, 5, 6 und 7 durch Einpoken des Schriftcodes zu aktivieren. Die Zeilen 1, 2 und 8 werden mit dem Bildschirmcode für den Laufbandhintergrund gefüllt.

Wenn man sich vorstellt, daß der oben beschriebene Algorithmus zur Darstellung jedes einzelnen Zeichens siebenmal abgearbeitet werden muß, dann wird schnell deutlich, daß dieser Programmteil sehr zeitkritisch ist. Da ein Basic-Interpreter verhältnismäßig langsam arbeitet, ist für dieses Problem ein Unterprogramm in Maschinensprache erforderlich. Wir erfüllen jetzt alle theoretischen Voraussetzungen, um die Laufbandanzeige auf einem Personal Computer mit Bildschirm zu realisieren.

Das abgebildete Basic-Programm »Laufband« gilt für alle CBM-Computer. »LB-BASIC-LOADER« beinhaltet das Unterprogramm in Maschinensprache und läuft nur auf CBM-Rechnern mit 40-Zeichen-Bildschirm. Das Listing »LB-BASIC-LOADER/80XX« enthält das Maschinenprogramm für Rechner mit 80-Zeichen-Bildschirm. Um die Laufbandanzeige auf einem CBM 8000 aktivieren zu können, benötigen Sie also statt »LB-BASIC-LOADER« das Programm »LB-BASIC-LOADER/80XX«.

◆◆◆ LB-BASIC-LOADER ◆◆◆

```

100 REM      *LB-BASIC-LOADER*
110 REM      *LAUFBAND*
120 REM *ALLE CBM MIT 40 ZEICHEN-BILDSCHIRM*
130 :
140 REM =====
150 :
160 PS=0:POKE59468,12:REM* GROSS/GRAFIK
170 :
180 FDIRI=826T0960
190 READB:POKEI,B:PS=PS+B
200 NEXTI
210 :
220 IFPS<17018THEN:PRINT"PRUEFSUMMENFEHLER":STOP
230 :
240 PRINT"MASCHINENPROGRAMM BEFINDET SICH 2.KASS-"
250 PRINT"PUFFER - EINSPRUNG MIT  SYS826"
260 PRINT"00"
270 PRINT"DAS PROGRAMM LAUFBAND KANN NUN"
280 PRINT"GESTARTET WERDEN."
290 END
300 :
310 DATA216,162,7,173,244,3,157,232,3,202,16,250
   .173,242,3,141,243,3
320 DATA162,8,202,48,44,173,243,3,221,222,3,240
   .30,48,19,56,173,243
330 DATA3,253,222,3,141,243,3,173,245,3,157,232
   .3,76,78,3,173,244
340 DATA3,157,232,3,76,78,3,173,245,3,157,232,3
   .162,1,160,3,169
350 DATA0,141,231,3,189,64,129,202,157,64,129,232
   .189,224,129,202,157,224
360 DATA129,232,232,224,160,208,4,32,176,3,96,238
   .231,3,173,231,3,201
370 DATA39,208,221,32,176,3,232,76,129,3,185,236
   .3,157,63,129,185,232
380 DATA3,157,223,129,136,96,0,0,0

```

MASCHINEN-UNTERPROGRAMM
 ◆◆◆ LB-BASIC-LOADER ◆◆◆

```

033A D8      CLD
033B A2 07    LDX #07
033D AD F4 03 LDA 03F4
0346 9D E8 03 STA 03E8,X
0343 CA      DEX
0344 19 FA    BPL 0340
0346 AD F2 03 LDA 03F2
0349 8D F3 03 STA 03F3
034C A2 08    LDX #08
034E CA      DEX
034F 38 2C    BMI 037D
0351 AD F3 03 LDA 03F3
0354 DD DE 03 CMP 03DE,X
0357 F6 1E    BEQ 0377
0359 30 13    BMI 036E
035B 38      SEC
035C AD F3 03 LDA 03F3
035F FD DE 03 SBC 03DE,X
0362 8D F3 03 STA 03F3
0365 AD F5 03 LDA 03F5
0368 9D E9 03 STA 03E9,X
036B 4C 4E 03 JMP 034E
036E AD F4 03 LDA 03F4
0371 9D E8 03 STA 03E8,X
0374 4C 4E 03 JMP 034E
0377 AD F5 03 LDA 03F5
037A 9D E8 03 STA 03E8,X
037D A2 01    LDX #01
037F A0 03    LDY #03
0381 A9 00    LDA #00
0383 8D E7 03 STA 03E7
0386 BD 40 81 LDA 8140,X
0389 CA      DEX
038A 9D 40 81 STA 8140,X
038D E8      INX
038E BD E0 81 LDA 81E0,X
0391 CA      DEX
0392 9D E0 81 STA 81E0,X
0395 E8      INX
0396 E8      INX
0397 E0 A0    CFX #A0
0399 D0 04    BNE 039F
039B 20 B0 03 JSR 03B0
039E 60      RTS
039F EE E7 03 INC 03E7
03A2 AD E7 03 LDA 03E7
03A5 C9 27    CMP #27
03A7 D0 DD    BNE 0386
03A9 20 B0 03 JSR 03B0
03AC E8      INX
03AD 4C 81 03 JMP 0381
03B0 B9 EC 03 LDA 03EC,Y
03B3 9D 3F 81 STA 813F,X
03B6 B9 E8 03 LDA 03E8,Y
03B9 9D DF 81 STA 81DF,X
03BC 88      DEY
03BD 60      RTS
03BE 00      BRK
03BF 00      BRK
03C0 00      BRK
03C1 FF      ***
  
```

Fuer ASSEMBLER-Programmierer:

Disassembliertes Listing des durch
 'LB-BASIC-LOADER' in den Pufferspeicher
 der 2.Cassette (826-1017 dez.) ge-
 loadeten Maschinenprogrammes !

◆◆◆ LB-LOADER/80XX ◆◆◆

```

100 REM      *LB-BASIC-LOADER / 80XX *
110 REM      *LAUFBAND*
120 REM *ALLE CBM MIT 80 ZEICHEN-BILDSCHIRM*
130 .
140 REM=====
150 .
160 PS=0:POKE59468,12:REM* GROSS/GRAFIK
170 .
180 FORI=826T0986
190 READB:POKEI,B:PS=PS+B
200 NEXTI
210 .
220 IFFSK>21070THEN:PRINT"FRUEFSUMMENFEHLER":STOP
230 .
240 PRINT"MASCHINENPROGRAMM BEFINDET SICH 2.KASS-"
250 PRINT"■■■■PUFFER - EINSPRUNG MIT 259826■"
260 PRINT"■"
270 PRINT"DAS PROGRAMM ■LAUFBAND■ KANN NUN"
280 PRINT"GESTARTET WERDEN."
290 END
300 .
310 DATA16,162,7,173,244,3,157,232,3,262,16,250
    .173,242,3,141,243,3
320 DATA162,8,202,48,44,173,243,3,221,222,3,240
    .30,48,19,56,173,243
330 DATA3,253,222,3,141,243,3,173,245,3,157,232
    .3,75,78,3,173,244
340 DATA3,157,232,3,76,78,3,173,245,3,157,232,3
    .162,1,160,1,169
350 DATA0,141,231,3,189,128,130,202,157,128,130
    .232,189,32,131,262,157,32
360 DATA131,232,189,192,131,202,157,192,131,232
    .189,96,132,202,157,96,132
370 DATA232,232,224,160,208,4,32,192,3,96,238,231
    .3,173,231,3,201,79,208
380 DATA205,32,192,3,232,76,129,3,185,238,3,157
    .127,130,185,236,3,157
390 DATA31,131,185,234,3,157,191,131,185,232,3,157
    .95,132,136,96,0,0,0

```

MASCHINEN-UNTERPROGRAMM
 ◆◆◆ LB-LOADER/80XX ◆◆◆

```

033A D8      CLD
033B A2 07    LDX #07
033D AD F4 03 LDA 03F4
0340 9D E8 03 STA 03E8.X
0343 CA      DEX
0344 10 FA    BPL 0340
0346 AD F2 03 LDA 03F2
0349 9D F3 03 STA 03F3
034C A2 08    LDX #08
034E CA      DEX
034F 30 2C    BMI 037D
0351 AD F3 03 LDA 03F3
0354 DD DE 03 CMP 03DE.X
0357 F0 1E    BEQ 0377
0359 30 13    BMI 036E
035B 38      SEC
035C AD F3 03 LDA 03F3
035F FD DE 03 SBC 03DE.X
0362 8D F3 03 STA 03F3
0365 AD F5 03 LDA 03F5
0368 9D E8 03 STA 03E8.X
036B 4C 4E 03 JMP 034E
036E AD F4 03 LDA 03F4
0371 9D E8 03 STA 03E8.X
0374 4C 4E 03 JMP 034E
0377 AD F5 03 LDA 03F5
037A 9D E8 03 STA 03E8.X
037D A2 01    LDX #01
037F A0 01    LDY #01
0381 A9 00    LDA #00
0383 8D E7 03 STA 03E7
0386 BD 80 82 LDA 0280.X
0389 CA      DEX
038A 9D 80 82 STA 0280.X
038D E8      INX
038E BD 20 83 LDA 0320.X
0391 CA      DEX
0392 9D 20 83 STA 0320.X
0395 E8      INX
0396 BD C0 83 LDA 03C0.X
0399 CA      DEX
039A 9D C0 83 STA 03C0.X
039D E8      INX
039E BD 60 84 LDA 0460.X
03A1 CA      DEX
03A2 9D 60 84 STA 0460.X
03A5 E8      INX
03A6 E8      INX
03A7 E0 A0    CPX #A0
03A9 D0 04    BNE 03AF
03AB 20 C0 03 JSR 03C0
03AE 60      RTS
03AF EE E7 03 INC 03E7
03B2 AD E7 03 LDA 03E7
03B5 C9 4F    CMP #4F
03B7 D0 CD    BNE 03B6
03B9 20 C0 03 JSR 03C0
03BC E8      INX
03BD 4C 81 03 JMP 0381
03C0 B9 EE 03 LDA 03EE.Y
03C3 9D 7F 82 STA 027F.X
03C6 B9 EC 03 LDA 03EC.Y
03C9 9D 1F 83 STA 031F.X
03CC B9 EA 03 LDA 03EA.Y
03CF 9D BF 83 STA 03BF.X
03D2 B9 E8 03 LDA 03E8.Y
03D5 9D 5F 84 STA 045F.X
03D8 88      DEY
03D9 60      RTS
  
```

Fuer ASSEMBLER-Programmierer:

Disassembliertes Listing des durch
 'LB-LOADER/80XX' in den Pufferspeicher
 der 2.Cassette (026-1017 dez.) ge-
 speicherten Maschinenprogrammes !

◆◆◆ LAUFBANDANZEIGE ◆◆◆

```

100 REM      * LAUFBAND *
110 REM      * -HORIZONTAL- *
120 :
130 REM *****
140 :
150 POKE59468,12:PRINTCHR$(142):REM* GRAFIK
160 RO#=CHR$(18):REM* RVS ON
170 RV#=CHR$(146):REM* RVS OFF
180 BS#=CHR$(157):REM* CURSOR LINKS
190 QS#=CHR$(175):REM* QUERSTRICH
200 SP#=CHR$(32):REM* SPACE
210 :
220 DIMI,J,K,AS,GF,UU,NS%
230 DIMA1,L2,L1#,L2#
240 :
250 SL=255:REM MAX. SATZLAENGE
260 REM  X() = TABELLE DER MATRIZENWERTE
270 REM  I,J,K = LAUFVARIABLEN
280 REM  AS = ANZAHL SAETZE
290 REM  WS() = TABELLE SATZWIEDERHOLUNG
300 REM  WS#() = TABELLE SATZINHALT
310 REM  GF = GESCHWINDIGKEITSAKTOR (1...9)
320 REM  UU = HILFSVARIABLE
330 REM  NS% = NEGATIV-STATUS
340 REM A1,L2,L1#,L2# = VAR. INPUTROUTINE
350 :
360 PRINT"□":REM CLEAR SCHIRM
370 PRINT"■ - L A U F B A N D A N Z E I G E -"
380 PRINT"■BITTE BEACHTEN SIE, DASS ZU DIESEM"
390 PRINT"ZEITPUNKT DAS PROGRAMM 2LB-BASIC-LOADER"■
400 PRINT"BEREITS EINMAL GESTARTET WERDEN MUSSTE,"
410 PRINT"SONST BLEIBT DAS PROGRAMM IM 2.KASS.-"
420 PRINT"PUFFER HAENGEN ODER STEIGT AUS!"
430 PRINT"■■■■■■ EINLESEN DER MATRITZENWERTE !!! ■"
440 :
450 DIMX(59,6):REM TABELLE MATRIZEN
460 FORI=1TO59:FORJ=0TO6
470 READX(I,J):REM READ DATA MATRIZEN
480 NEXTJ,I
490 :
500 PRINT"■ANZAHL SAETZE (MAX. SATZLAENGE = 255"
510 PRINT"ZEICHEN): ";INPUTAS
520 IFAS>255THEN PRINT"■ZU GROSS!■TTT":GOTO510
530 REM ASC=255 FUER FET 2001
540 DIMWS#(AS),WS(AS):REM TABELLEN SATZINHALT, -WIEDERHOLUNG
550 PRINT"□":REM CLEAR SCHIRM
560 PRINT"GEBEN SIE NUN ";AS;"SA(E)TZ(E) EIN !"
570 PRINT"■BITTE BEACHTEN SIE, DAS ALLE EINGEGE-"
580 PRINT"BEHEN ZEICHEN, DIE ■NICHT■ ZWISCHEN"

```

```

590 PRINT"ASC(32) UND ASC(90) LIEGEN. VON DER IN-"
600 PRINT"PUT-ROUTINE DES PROGRAMMS IGNORIERT"
610 PRINT"WERDEN."
620 FORUU=1TO39:PRINT"=":NEXT:PRINT
630 :
640 PRINT"### EINGABEN: █".PRINT"###"
650 FORK=1TOAS
660 PRINTRO#;K;"### SATZ:"RV#SP#;
670 GOSUB1660:WS#(K)=L1#
680 WS#(K)=WS#(K)+SP#
690 PRINT"### WIE OFT SOLL OBIGER SATZ HINTEREINANDER"
700 PRINT"ANGEZEIGT WERDEN ":INPUTWS#(K)
710 PRINT"### █ - EINGABEN IN ORDNUNG ? (J/N) - █"
720 GETX#;IFX#="J"THEN750
730 IFX#<"N"THEN720
740 K=K-1:REM WIEDERHOLTE EINGABE
750 PRINT:NEXTK
760 :
770 PRINT"### GESCHWINDIGKEITSFAKTOR (1...9)":
780 INPUTGF
790 IFGF<1ORGF>9THEN:PRINT"###":GOTO770
800 PRINT:PRINT
810 PRINT"LAUFBAND INVERTIERT (3NEGATIV█) ? (J/N)"
820 GETX#;IFX#="N"THEN850
830 IFX#<"J"THEN820
840 NS%=128:GOTO880:REM NEGATIVSTATUS SETZEN !
850 NS%=0:REM NEGATIVSTATUS LOESCHEN !
860 :
870 REM EINPOKEN DER WERTIGKEIT
880 FORI=0TO7:POKE990+I,INT(2^I):NEXTI
890 POKE1012,32+NS%:REM BILDSCHIRMCODE FUER HINTERGRUND
900 POKE1013,108+NS%:REM BILDSCHIRMCODE FUER SCHRIFT
910 :
920 PRINT"█":REM CLEAR SCHIRM
930 PRINT"### UNTERBRECHUNG NACH JEDEM SATZ MOEGLICH"
940 PRINT"MIT ^3SHIFT+CLEAR HOME█ !"
950 PRINT:PRINT
960 PRINT" A C H T U N G - A C H T U N G"
970 PRINT"### IN WENIGEN AUGENBLICKEN BEGINNT DIE"
980 PRINT"#####LAUFANZEIGE."
990 FORUU=1TO3000:NEXT
1000 PRINT"█":REM CLEAR SCHIRM
1010 :
1020 REM LAUFBANDANZEIGE
1030 FORK=1TOAS:REM SCHLEIFE VERSCH. SAETZE
1040 REM TASTE ^3SHIFT+CLEAR^ GEDRUECKT ?
1050 GETX#;IFX#=CHR$(147)THEN:K=AS:GOTO1010
1060 FORL=1TOWS(K):REM SCHLEIFE PRO SATZ
1070 FORI=1TOLEN(WS#(K)):REM SCHLEIFE SATZLAENGE
1080 EZ=ASC(MID$(WS#(K),I,1))
1090 FORJ=0TO6:REM SCHLEIFE ZEICHENLAENGE
1100 POKE1010,X(EZ-31,J)

```

```

1110 SYS826:REM SPRUNG MASCHINEN-UP
1120 GOSUB1280:REM GESCHWINDIGKEIT
1130 NEXTJ
1140 GOSUB1310:REM ZWISCHENRAUM
1150 NEXTI
1160 GOSUB1310:REM ZWISCHENRAUM
1170 NEXTL
1180 NEXTK
1190 GOTO1030:REM ENDLOSSCHLEIFE
1200 :
1210 PRINT"SENDE ? (J/N)"
1220 GETX#:IFX#="J"THEN:PRINT"SENDE":END
1230 :
1240 IFX#<>"N"THEN1220
1250 PRINT"J":REM CLEAR SCHIRM
1260 PRINT"BISHERIGE GESCHWINDIGKEIT: ",GF
1270 GOTO770
1280 REM * GESCHWINDIGKEITSAKTOR
1290 FORUU=1TO100-(GF*11):NEXT:RETURN
1300 :
1310 REM FUER ZEICHENZWISCHENRAUM
1320 POKE1010,0:SYS826:GOSUB1280:RETURN
1330 :
1340 REM MATRIZENWERTE FUER ASC(32) - ASC(90)
1350 DATA0,0,0,0,0,0,0,0,0,0,242,0,0,0:REM SPACE, !
1360 DATA0,224,0,0,224,0,0,0,40,254,40,40,254,40
:REM ", #
1370 DATA0,40,84,254,84,72,0,0,96,130,8,16,38,70
:REM $, %
1380 DATA0,108,146,146,106,4,10,0,0,32,64,128,0,0
:REM &, ^
1390 DATA0,0,56,68,130,0,0,0,0,130,68,56,0,0:REM (, )
1400 DATA0,84,56,254,56,84,0,0,16,16,124,16,16,0
:REM *, +
1410 DATA0,0,2,12,0,0,0,0,16,16,16,16,16,16:REM KOMMA, -
1420 DATA0,0,6,6,0,0,0,0,2,4,8,16,32,64:REM ., /
1430 DATA0,124,138,146,146,162,124,0,0,34,66,254,2,2
:REM 0, 1
1440 DATA0,70,138,138,146,146,98,0,68,130,146,146,146,108
:REM 2, 3
1450 DATA0,8,24,40,72,254,8,0,228,162,162,162,148,136
:REM 4, 5
1460 DATA0,60,82,146,146,146,12,0,192,128,142,144,160,192
:REM 6, 7
1470 DATA0,108,146,146,146,146,108,0,96,146,146,146,148,120
:REM 8, 9
1480 DATA0,0,0,36,0,0,0,0,0,2,44,0,0,0:REM :, ;
1490 DATA0,16,56,108,198,130,130,0,40,40,40,40,40,40
:REM <, =
1500 DATA0,130,130,198,108,56,16,0,64,128,138,144,144,96
:REM >, ?
1510 DATA0,56,68,146,170,154,114,0,62,80,144,144,80,62

```

```

      REM @,A
1520 DATA0,130,254,146,146,146,108,0,56,68,130,130,130,68
      :REM B,C
1530 DATA0,130,254,130,130,68,56,0,254,146,146,146,130,130
      :REM D,E
1540 DATA0,254,144,144,144,128,128,0,56,68,130,146,146,92
      :REM F,G
1550 DATA0,254,16,16,16,16,254,0,0,0,130,254,130,0
      :REM H,I
1560 DATA0,4,2,2,130,252,128,0,254,16,16,40,68,130
      :REM J,K
1570 DATA0,254,2,2,2,2,2,0,254,64,48,48,64,254:REM L,M
1580 DATA0,254,64,32,16,8,254,0,56,68,130,130,68,56
      :REM N,O
1590 DATA0,254,144,144,144,144,96,0,56,68,130,138,68,58
      :REM P,Q
1600 DATA0,254,144,144,152,148,98,0,100,146,146,146,146,76
      :REM R,S
1610 DATA0,0,128,128,254,128,128,0,252,2,2,2,2,252
      :REM T,U
1620 DATA0,224,24,6,6,24,224,0,254,4,24,24,4,254
      :REM V,W
1630 DATA0,198,40,16,16,40,198,0,0,224,16,30,16,224
      :REM X,Y
1640 DATA0,134,138,146,146,162,194:REM Z
1650 :
1660 REM* INPUT-ROUTINE MIT GET
1670 L1#="":A1=0:PRINTQS#BS#:
1680 GETL2#:IFL2#=""THEN1680
1690 L2=ASC(L2#):IFL2=20THEN1760
1700 IFA1>SLTHEN1720
1710 IFL2=13THEN1780
1720 IFL2<32ORL2>90THEN1680
1730 IFA1>SL-1THEN1680:REM MAX SATZLAENGE = SL-1
1740 L1#=L1#+L2#:A1=A1+1:IFA1>SLTHEN:PRINTR0#L2#RV#QS#BS#:
      :GOTO1680
1750 PRINTL2#QS#BS#: :GOTO1680
1760 IFA1<1THEN1680
1770 A1=A1-1:L1#=LEFT$(L1#,A1):PRINTBS#QS#SP#BS#BS#:
      :GOTO1680
1780 PRINTSP#
1790 RETURN

```

Bedienungsanleitung

Zuerst muß das Programm »LB-BASIC-LOADER« gestartet werden. Der Computer liest nun das in DATA-Statements befindliche Maschinenprogramm und pokt es in den Pufferbereich des zweiten Kassettenrecorders (ab Speicherstelle 826 dez. oder 033A hex.). Dort ist es so lange vor Zerstörung gesichert, bis der zweite Kassettenrecorder benutzt, die Speicherstellen mit »POKE« anderweitig beschrieben oder das Gerät abgeschaltet wird. Während des Programmablaufs wird eine Prüfsumme gebildet, die gleich der Addition aller gültigen DATA-Statements sein muß. Erscheint also auf dem Bildschirm die Meldung »Prüfsummenfehler«, dann haben Sie beim Eintippen der DATAs etwas falsch gemacht. In diesem Fall sind alle Dezimalzahlen zu überprüfen und ggf. zu korrigieren.

Schließt das Programm mit dem Hinweis »Das Programm Laufband kann nun gestartet werden« ab, dann steht das Maschinenprogramm korrekt im Speicher. Mit »NEW« wird es nicht zerstört. Das Basic-Loader-Programm jedoch kann entweder im Arbeitsspeicher gelöscht oder überschrieben werden. Es wird für den weiteren Programmablauf nicht mehr benötigt.

Jetzt muß das Basic-Programm »Laufband« geladen und mit »RUN« gestartet werden. Nach Umschaltung des Computers auf Groß/Grafikmodus wird die Variablen-tabelle angelegt und der Bildschirm gelöscht.

Der Bediener wird darauf hingewiesen, daß das Programm »LB-BASIC-LOADER« bereits einmal gestartet werden mußte, da sonst der Computer aussteigt. Nach dem Dimensionieren werden die Zeichenschlüssel aus den DATA-Statements in die Matrizentabelle eingelesen. Im Anschluß daran fragt der Computer, wie viele Sätze eingegeben werden sollen (Beispiel: 3). Gibt man hier eine Zahl größer als 255 ein, so springt das Programm nach einer Fehlermeldung zurück zur erneuten Eingabe. Diese Fehlerabfrage bewirkt, daß das Basic-Programm auch auf CBM-Systemen mit altem Betriebssystem (2000) läuft. Der Rechner bringt anschließend die Meldung: »Geben Sie nun drei Sätze ein!« und weist den Benutzer darauf hin, daß alle eingegebenen Zeichen innerhalb CHR\$(32) bis CHR\$(90) liegen müssen, da sie sonst vom Programm ignoriert werden. Anschließend erfolgt die programmkontrollierte Erfassung der Sätze, dabei wird jedes Zeichen auf Gültigkeit getestet. Schließt der Bediener eine Eingabe mit »RETURN« (CHR\$(13)) ab, so wird dieser Satz in die Satz-tabelle übernommen und der Computer fragt, wie oft hintereinander der eben eingegebene Satz am Laufband angezeigt werden soll (Wiederholungsrate).

Nach einer Ja/Nein-Abfrage, ob »Eingaben in Ordnung«, wartet der Rechner auf die nächste Eingabe. Sind alle Sätze eingegeben und ist die Schleife vollständig abgearbeitet, dann kann der Benutzer mit der Festlegung der Steuerungsparame-

ter im Programmablauf fortfahren. Der Rechner wartet nun auf die Eingabe eines Geschwindigkeitsfaktors zwischen 1 (langsam) und 9 (schnelles Laufband). Anschließend kann durch Ja/Nein-Abfrage die Darstellungsart (normal oder invertiert) gewählt werden. Nach dem Einpoken der Wertigkeitstabelle und der Bildschirmcodes für Hintergrund und Schrift weist der Computer darauf hin, daß nach jedem Satz innerhalb der Laufbandanzeige eine Unterbrechung durch Drücken der Tasten »SHIFT + CLEAR HOME« möglich ist.

Jetzt beginnt die Laufbandanzeige. Hierzu werden, eingebettet in eine Endlosschleife, ineinander folgende Schleifen abgearbeitet:

- Schleife für Anzahl der verschiedenen Sätze;
- für die Wiederholungsrate pro Satz;
- Schleife zur Abarbeitung der Satzlänge und
- zur Darstellung der Zeichenlänge.

Innerhalb der zuletzt genannten Schleife werden die Codes jeder einzelnen vertikalen Wertigkeitsaddition an das Maschinenprogramm übergeben. Anschließend erfolgt mit »SYS 826« der Sprung in die Unterroutine. Dort wird der Zeichencode entschlüsselt und die entsprechenden Laufbandstellen aktiviert. Ebenso wird innerhalb des Maschinenunterprogramms der Bildschirminhalt im Bereich des Laufbandes nach links verschoben. Nach dem Rücksprung beginnt das Programm mit der Abarbeitung der Geschwindigkeitsschleife, welche praktisch eine Verzögerung des Laufbandes darstellt. Die Großbuchstaben würden sonst so schnell über den Bildschirm wandern, daß man die Schrift nicht deutlich lesen könnte.

Das Programm befindet sich nun in einer Endlosschleife, aus der man durch Betätigen der Tasten »SHIFT + CLEAR HOME« herauspringen kann. Der Computer fragt dann, ob das Programm beendet werden soll. Verneint man diese Frage durch Betätigen der Taste »N« (für »Nein«), so zeigt der Rechner den bisherigen Geschwindigkeitsfaktor des Laufbandes an. Es ist nun möglich, die Steuerungsparameter zu ändern, um so eine individuelle Laufbandgestaltung vorzunehmen, sei es durch Änderung der Geschwindigkeit oder invertierte Darstellungsform. Wird bei der Frage »Ende (J/N)?« die »J«-Taste gedrückt, erfolgt Bildschirmlöschung und Beendigung des Programmes.

Besonderheiten bei CBM-Rechnern mit Betriebssystem Basic 4.0

Kurze Maschinenprogramme werden im allgemeinen in einem vor BASIC geschützten Speicherbereich aufbewahrt. Der Befehl »NEW« löscht zwar den gesamten Arbeitsspeicher ab 1024 dez., greift aber nicht auf den Speicherbereich

des ersten und zweiten Kassettenpuffers zu. Da der Puffer des zweiten Recorders (826 bis 1017 dez.) nur selten benutzt wird, bietet er sich für Maschinenprogramme < 192 Bytes geradezu an. Doch Vorsicht — bei Rechnern, die mit dem Betriebssystem Basic 4.0 arbeiten, wird dieser Speicherbereich noch für andere Zwecke benutzt:

1. Die TAB-Funktion und
2. die Floppy-Kurzbefehle, z.B. DLOAD, DSAVE usw.

bewirken die Änderung einiger Speicherzellen des zweiten Kassettenpuffers. Während bei Verwendung des TAB-Befehles die einzelnen TAB-Positionen in diesem Bereich vermerkt werden, benötigt Basic 4.0 einige Speicherzellen zur Stringmanipulation der Kurzbefehle. Diese müssen nämlich erst floppygerecht aufbereitet werden, bevor sie über den IEC-Bus an das Peripheriegerät gesendet werden können.

Vorausgesetzt, Ihr CBM arbeitet mit Basic 4.0 und Sie haben die Programme zur Laufbandanzeige unter den angegebenen Namen auf der Floppy abgespeichert, so ist zur einwandfreien Funktion folgende Anweisung genau einzuhalten:

I. 40-Zeichen-Bildschirm, alle CBM-Basic-Versionen

- a) LOAD" LW:LB-BASIC-LOADER",GA
- b) RUN
- c) LOAD" LW:LAUFBANDANZEIGE",GA
- d) RUN

II. 80-Zeichen-Bildschirm, alle CBM-Basic-Versionen

- a) LOAD" LW:LB-LOADER/80XX",GA
- b) RUN
- c) LOAD" LW:LAUFBANDANZEIGE",GA
- d) RUN

(LW = Laufwerks-Nummer)

(GA = Geräteadresse der Floppy)

Programmbeschreibung

Das Basicprogramm »LB-BASIC-LOADER« hat lediglich den Zweck, das in DATA-Statements befindliche Maschinenprogramm in den Pufferbereich des zweiten Kassettenrecorders zu poken. In Zeile 160 wird die Prüfsummenvariable »PS« auf Null gesetzt und auf Groß-/Grafikmodus umgeschaltet. Die »FOR...NEXT«-Schleife in den Zeilen 180 bis 200 liest jedes einzelne DATA in

die Variable »B«, pokt es in die entsprechende Speicherstelle und bildet die Prüfsumme. Die Prüfsumme ist nichts anderes als die Addition aller DATAs. Stimmt sie nicht mit der in Zeile 220 stehenden Vergleichssumme überein, wird das Programm mit der Meldung »Prüfsummenfehler« abgebrochen. Bei Übereinstimmung kommt auf dem Bildschirm die Bestätigung für korrekten Programmablauf »Maschinenprogramm befindet sich im zweiten Kassettenpuffer...« (Zeilen 240 bis 280) und Beendigung des Programms (Zeile 290). In den Programmzeilen 310 bis 380 befindet sich, codiert in Dezimalzahlen, das Maschinenunterprogramm.

Im Hauptprogramm »Laufband« wird in Zeile 160 der Zeichengenerator auf Groß/Grafik geschaltet. Anschließend erfolgt das Anlegen der Variablen-tabelle in den Programmzeilen 160 bis 250. Von 260 bis 340 sind hinter REMarks die Bedeutungen der Variablen aufgeführt. Die Zeilen 360 bis 430 bewirken das Darstellen der ersten Bildschirmmaske. Im Anschluß daran werden die Zeichenschlüssel in eine zweidimensionale Tabelle (Variable »X(59,6)«) eingelesen (450 bis 480). In 500 bis 530 wird die gewünschte Satzanzahl in die Variable »AS« eingegeben und plausibiliert. Während die Zeilen 550 bis 620 lediglich Bedienerhinweise bewirken, erfolgt in der Schleife von 640 bis 750 die eigentliche programmkontrollierte Texterfassung. Hier wird außerdem zu jedem Satz die Wiederholungsrate festgelegt und in die Satz-wiederholungstabelle (»WS(K)«) weggeschrieben. In den Zeilen 710 bis 740 ist eine Fehlerkorrekturmöglichkeit gegeben. Die Steuerungsfunktionen »Geschwindigkeit« und »Negativstatus« werden in den Programmzeilen 770 bis 850 per Bediener-eingabe festgelegt. Der Programmteil 870 bis 900 greift mit »POKE« in den Pufferbereich des zweiten Kassettenrecorders. Zunächst werden die einzelnen Matrizenwertigkeiten in die Speicherzellen 990 bis 997 dez. geschrieben (Wert 1 in 990, 2 in 991, 3 in 992 usw.). In Speicherzelle 1012 dez. befindet sich der Bildschirmcode für den Laufbandhintergrund und in 1013 der Code für die Schriftbildung. Dem Benutzer ist es möglich, durch Änderung der Zahl 32 (in Zeile 890) bzw. 108 (in Zeile 900) jedes beliebige Zeichen für die Darstellung des Laufbandes zu benutzen. Es muß allerdings darauf geachtet werden, daß der eingegebene Code nie größer als 127 ist, sonst läuft das Programm auf »ILLEGAL QUANTITY ERROR«.

Trotzdem wird der Bediener angehalten, hier zu experimentieren, um seine individuellen Vorstellungen und Wünsche zu realisieren. Nach einer weiteren Bildschirmmaske in den Zeilen 920 bis 1000 beginnt die Endlosschleife zur Aktivierung und Darstellung des Laufbandes. BASIC übernimmt hierbei die Aufgabe, die Zeichen und Sätze aneinanderzuhängen und die einzelnen Zeichencodes an das Maschinenprogramm zu übergeben. Nach jeder Eingabe erfolgt mit »SYS 826« ein Sprung in die Unter-routine, die dann die Verschiebung und Generierung der Laufbandschrift übernimmt. In den Programmzeilen 1340 bis 1640 sind

in DATA-Statements die verschlüsselten Zeichencodes untergebracht. In 1660 bis 1790 befindet sich die komfortable Input-Routine, die es erlaubt, Satzlängen zu definieren und nur zugelassene Zeichen zu akzeptieren. Diese Subroutine kann in andere Problemlösungen eingebaut werden. Voraussetzung dafür ist, daß die verwendeten Variablennamen angelegt und nicht anderweitig benutzt werden.

Banküberweisungen ausfüllen

Banküberweisungen ausfüllen

Viele Kontoinhaber, die in großem Umfange Zahlungseingänge zu erwarten haben, fügen ihren Rechnungen, Mahnschreiben oder sonstigen Zahlungsanforderungsschreiben Überweisungs-Zahlvordrucke bei, auf denen ihr Name und ihre Anschrift sowie ihre Kontonummer und der Name des Kreditinstitutes bereits eingedruckt oder mittels EDV-Anlage vorbeschriftet sind. So soll einerseits den zur Zahlung verpflichteten Personen die Erledigung der Bezahlung erleichtert werden. Andererseits fördert die Vordruckbeilage einen beschleunigten Zahlungseingang zugunsten des Kontoinhabers selbst und erleichtert zudem noch die innerbetriebliche Weiterbearbeitung der Überweisungs-Zahlschein-Gutschriften, weil die zur Buchung benötigten Angaben vorbeschriftet sein können.

Hat man als Inhaber eines Bankkontos nun selbst häufig Überweisungen zu tätigen, die sich nicht durch einfaches Ergänzen von bereits vorbedruckten Belegen erledigen lassen (Lohn- und Gehaltsabrechnung usw.), so gibt es mehrere Möglichkeiten. Als Besitzer eines EDV-Systems beispielsweise kann man die Überweisungsformulare entweder per Programm mittels eines Druckers beschriften lassen oder man bedient sich der ständig wachsenden Möglichkeiten des Datenträger austausches.

Im Falle der EDV-mäßigen Beschriftung kann die Verwendung von neutralen Überweisungsvordrucken sehr von Vorteil sein, wenn man die Bezahlung über eigene Konten bei verschiedenen Kreditinstituten abwickeln möchte. Als neutral werden solche Zahlungsverkehrsvordrucke bezeichnet, bei denen die Bankleitzahl und die Bezeichnung des zu beauftragenden Kreditinstitutes nicht bereits bei der Vordruckerstellung eingedruckt sind, sondern erst bei der Vordruckbeschriftung eingesetzt werden. Demnach werden neutrale Überweisungsvordrucke von den Kreditinstituten nur dann akzeptiert, wenn den betreffenden Kontoinhabern daraus wesentliche Rationalisierungsvorteile entstehen, und wenn sie gegenüber den Banken eine schriftliche Haftungsfreistellung erklären. Wenn dagegen der Vordruckkopf des unbeschrifteten Überweisungsträgers bereits Name, Sitz und Bankleitzahl des jeweiligen Kreditinstitutes beinhaltet, kann von einer Haftungsfreistellungserklärung abgesehen werden.

PROGRAMM-AUSWAHL:

- 1** = LISTE EMPFAENGER
- 2** = LISTE VERWENDUNGSZWECK
- 3** = LISTE AUFTRAGGEBER
- 4** = DRUCK UEBERWEISUNG
- 5** = ENDE PROGRAMM

BITTE WAEHLN !!!

CODENUMMERN FUER

EMPFAENGERNAME,
VERWENDUNGSZWECK,
AUFTRAGGEBER

EINGEBEN !

(**0,0,0** ODER **?** BEDEUTET FORMULARTEST)

EINGABEN: ? 0,0,0

GEWAEHLTE UEBERWEISUNGSDATEN:

XX-EMPFAENGERNAME-XXXXXXXXXXXXXXXXXXXXX
PLZX XX-ORT-XXXXXXXXXX X-STRASSE-XX
XX-BLZ-XXXXXX
XX-KTONR.-XXXXXX
XX-BANKENNAME-XXXXXXXXXXXXXXXXXXXXX

XX-TEXT 1-XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX-TEXT 2-XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX-TEXT 3-XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX-DM-BETRAG-XX

XX-KTONR.-XXXXXX
XX-NAME-XX PLZX XXX-ORT-XXX XX-STR-XX

- D** = DRUCK **K** = KORREKTUR
- N** = NEUE EINGABEN **E** = ENDE PROG

BITTE WAEHLN !

DRUCK BEENDE-PROGRAMM

DRUCK FUER SAMMELUEBERWEISUNG

- G** = GUELTIG **U** = UNGUELTIG

BITTE WAEHLN !

Problembeschreibung

Wer seine Überweisungsträger per EDV-Drucker beschriftet, hat nicht nur den Vorteil der Vereinfachung; er kann darüber hinaus auch Kontoführungsgebühren sparen, da in der Regel bei fünf und mehr Einzelüberweisungen vom gleichen Konto mit dazugehöriger Sammelüberweisung extrem niedrigere Buchungspostengebühren berechnet werden.

Das Ausfüllen von Überweisungsträgern gliedert sich nun zunächst in die beiden Teilaufgaben »Einzelüberweisungen beschriften« und anschließend »Erstellen der Sammelüberweisung«.



Gutschrift

<small>Empfänger (Name und Anschrift)</small> OLGA MUSTERFRAU 7000 STUTTGART GASSE 888	<small>Bankleitzahl</small> 654 321 0
<small>Konto-Nr. des Empfängers</small> 222 222 2	<small>bei - oder ein anderes Konto des Empfängers</small> MUSTERBANK
<small>Verwendungszweck (nur für Empfänger)</small>	
MITGLIEDSNR 5689 MONATSBEITRAG JAN. 1983	DM XXXXXXXXXX62.00
<small>Konto-Nr. des Auftraggebers</small> 345 678 9	<small>Auftraggeber</small> ZAHLEMANN & SOEHNE , BERLIN
<small>Mehrzweckfeld</small>	<small>Konto-Nr.</small>
<small>Betrag</small>	<small>Bankleitzahl</small>
<small>Text</small>	

51d

Bitte dieses Feld nicht beschriften und nicht bestempeln



Gutschrift

<small>Empfänger (Name und Anschrift)</small> FRITZ MUSTERMANN 6000 FRANKFURT ZEIL 666	<small>Bankleitzahl</small> 123 456 7
<small>Konto-Nr. des Empfängers</small> 111 111 1	<small>bei - oder ein anderes Konto des Empfängers</small> MUSTERBANK
<small>Verwendungszweck (nur für Empfänger)</small>	
RECHNUNGSNR 54219 VOM 12.01.1983	DM XXXXXXXXXX125.60
<small>Konto-Nr. des Auftraggebers</small> 345 678 9	<small>Auftraggeber</small> ZAHLEMANN & SOEHNE , BERLIN
<small>Mehrzweckfeld</small>	<small>Konto-Nr.</small>
<small>Betrag</small>	<small>Bankleitzahl</small>
<small>Text</small>	

51d

Bitte dieses Feld nicht beschriften und nicht bestempeln

Gutschrift

Empfänger (Name und Anschrift)		Bankleitzahl
-SUMMENBELEG-		
Konto-Nr. des Empfängers — bei - oder ein anderes Konto des Empfängers		
Verwendungszweck (nur für Empfänger)		DM
2 UEBERWEISUNGEN		XXXXXXXX187.60
Konto-Nr. des Auftraggebers	Auftraggeber <i>15.01.83 Fahlmann</i>	
345 678 9	ZAHLEMANN & SOEHNE , BERLIN	
Mehrzweckfeld	Konto-Nr.	Betrag
		Bankleitzahl
		Text

51d

Bitte dieses Feld nicht beschriften und nicht bestempeln

Anwendungsbeispiel: Zwei Einzelüberweisungen einschließlich Sammelüberweisung

Gutschrift

AUFTRAGGEBERBANKENNAME		1234567
Empfänger (Name und Anschrift)		Bankleitzahl
XX-EMPFÄNGERNAME-XXXXXXXXXXXXXXXXXXXX PLZX XX-ORT-XXXXXXXXXX X-STRASSE-XX		XX-BLZ-XXXXX
Konto-Nr. des Empfängers — bei - oder ein anderes Konto des Empfängers		
XX-KTONR. -XXXXXX		XX-BANKENNAME-XXXXXXXXXXXXXXXXXXXX
Verwendungszweck (nur für Empfänger)		DM
XX-TEXT 1-XXXXXXXXXXXXXXXXXXXX XX-TEXT 2-XXXXXXXXXXXXXXXXXXXX XX-TEXT 3-XXXXXXXXXXXXXXXXXXXX		XX-DM-BETRAG-XX
Konto-Nr. des Auftraggebers	Auftraggeber	
XX-KTONR. -XXXXXX	XX-NAME-XX PLZX XXX-ORT-XXX XX-STR-XX	
Mehrzweckfeld	Konto-Nr.	Betrag
		Bankleitzahl
		Text

51d

Bitte dieses Feld nicht beschriften und nicht bestempeln

Formulartest für neutrale Überweisungsformulare (Name, Sitz und Bankleitzahl des Auftraggeber-Kreditinstitutes werden per EDV-Drucker beschriftet)



Gutschrift

Empfänger (Name und Anschrift)		Bankleitzahl
XX-EMPFÄNGERNAME-XXXXXXXXXXXXXXXXXXXX PLZX XX-ORT-XXXXXXXXXX X-STRASSE-XX		XX-BLZ-XXXXX
Konto-Nr. des Empfängers —, bei - oder ein anderes Konto des Empfängers —		
XX-KTONR. -XXXXX XX-BANKENNAME-XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
Verwendungszweck (nur für Empfänger)		DM
XX-TEXT 1-XXXXXXXXXXXXXXXXXXXXXXXXXXXX XX-TEXT 2-XXXXXXXXXXXXXXXXXXXXXXXXXXXX XX-TEXT 3-XXXXXXXXXXXXXXXXXXXXXXXXXXXX		XX-DM-BETRAG-XX
Konto-Nr. des Auftraggebers — Auftraggeber —		
XX-KTONR. -XXXXX XX-NAME-XX PLZX XXX-ORT-XXX XX-STR-XX		
Mehrzweckfeld	✗	Konto-Nr.
	✗	Betrag
	✗	Bankleitzahl
	✗	Text

51d

Bitte dieses Feld nicht beschriften und nicht bestempeln

Formulartest für nicht neutrale Überweisungsformulare (Name, Sitz und Bankleitzahl der Auftraggeberbank sind bereits vorbeschriftet)

Auf Einzelüberweisungen sind folgende Angaben zu machen:

- Bei neutralen Formularen ist der Vordruckkopf mit Name, Sitz und Bankleitzahl des beauftragten Kreditinstituts auszufüllen.
- Das Mittelfeld enthält Empfängername und Anschrift, Kontonummer, Bankname und Bankleitzahl der Empfängerbank sowie Angaben über den Verwendungszweck der Überweisung (Rechnungsnummer, Kundennummer usw.).
- Der Überweisungsbetrag sollte klar und deutlich lesbar, sowie durch spezielle Formatierung unmanipulierbar auf der Überweisung angedruckt werden.
- Kontonummer und Name des Auftraggebers sind ebenfalls anzugeben.
- Im Vordruck können bei Beschriftung per geeignetem EDV-Drucker in der Codierzeile die Empfängerkontonummer, der Betrag, die Bankleitzahl des Empfängerkreditinstitutes und die zugehörigen Sonderzeichen in OCR-A1-Schrift codiert werden.

(OCR von englisch Optical Character Recogniton — maschinell und visuell erkennbare Schriftzeichen, wobei mit dem Zusatz »A1« die normengerechte Vorrangreihe einer bestimmten Schriftgröße bezeichnet wird.)

Für Sammelüberweisungen (Summenbelege) gelten im allgemeinen folgende Bestimmungen:

- In der Regel kann für die Sammelüberweisung ein Einzelüberweisungsformular verwendet werden.
- Es können nur mehrere Überweisungsaufträge vom gleichen Konto zu einem Summenbeleg zusammengefaßt werden.

Die wichtigsten Angaben auf einer Sammelüberweisung sind:

- Anzahl der Einzelüberweisungen;
- Gesamtüberweisungsbetrag, resultierend aus der Addition der einzelnen Einzelüberweisungsbeträge von einem Auftraggeberkonto;
- Kontonummer und Name des Auftraggebers
- sowie Datum und Unterschrift des Auftraggebers.

Ein Computerprogramm »Druck Überweisungen« sollte den Anwender in die Lage versetzen, neutrale sowie nicht neutrale Einzel- und Sammelüberweisungen zu beschriften und häufig wiederkehrende Daten zu speichern. Darüber hinaus sollten etwaige Bedienereingaben plausibilisiert und durch geeignete Fehlermeldungen angezeigt werden. Die gespeicherten Daten müssen zur Information des Benutzers auch ausgedruckt werden können.

Problemlösung

Ein Anwendungsgebiet unter vielen für Personal Computer ist das exakte und schnelle Ausfüllen von Formularen allgemein. Endlosformulare mit Transportlochung haben hier den entscheidenden Vorteil, daß sie sich auf einem Drucker mit Traktorführung genau positionieren lassen. Die CBM-Printer 3022 und 8024 beispielsweise eignen sich demnach unter anderem auch hervorragend zur Formularbeschriftung. Das dazugehörige Programm, speziell für Überweisungsträger, ist in Commodore-Basic geschrieben und genügt allen oben genannten Anforderungen. Es läuft auf allen CBM 2000-, 3000-, 4000- und 8000-Anlagen sowie VC 20 mit angeschlossenem Drucker.

Es gibt mehrere Methoden, die zum Ausfüllen der Überweisungen erforderlichen Daten zu generieren. Denkbar wäre z.B. die Erfassung und Abspeicherung der Überweisungsdaten auf Diskette oder Kassette oder einfache Datenerfassung über Tastatur mit anschließendem Ausdrucken. Die reine Tastatureingabe wäre sicherlich zu zeitaufwendig und die Datenspeicherung in Files erfordert Dateiverwaltungsprogramme (Erfassen, Löschen, Ändern, Listen usw.) Im vorliegenden Programm befinden sich die Daten in DATA-Statements. Dadurch ge-

staltet sich die Datenverwaltung programmtechnisch recht einfach und effizient. Immer wiederkehrende Überweisungsdaten können so im Programm fest gespeichert und bei Bedarf abgeändert, ergänzt und ausgedruckt werden. Darüber hinaus erlaubt das Programm temporäre Datenspeicherung, was bedeutet, daß einmalig benötigte Angaben nicht fest ins Programm genommen werden müssen, sondern über Tastatur erfaßt werden können.

Zum Zwecke einer optimalen Speicherung sind die Überweisungsdaten in drei Teile untergliedert:

- Empfängerdaten (Empfängername, Anschrift und Kontonummer, Bankleitzahl und Bankname des Empfängers);
- Verwendungszweckdaten (Verwendungszweck, Überweisungsbetrag);
- Auftraggeberdaten (Kontonummer und Name des Auftraggebers).

Diese Vorgehensweise ermöglicht die Erstellung beliebiger Überweisungen, kombiniert aus den oben genannten Einzeldaten. Das Programm darf jedoch nur solche Einzelüberweisungen zu einem Summenbeleg zusammenfassen, deren Auftraggeberdaten identisch sind.

Die Möglichkeiten des Programms »Druck Überweisungen« im einzelnen sind:

- Auswahlmü mit der Möglichkeit, Listen selektiert nach Empfänger-, Verwendungszweck- und Auftraggeberdaten auszudrucken;
- Kombination der Einzeldaten durch Eingabe von Codenummern (lfd. Nr.) möglich;
- Arbeitswahl, bestehend aus Druck, Einzelüberweisung, Datenkorrektur, Rücksprung zur Eingabe oder Programmbeendigung;
- Druckersteuerung;
- Formulartest zum Justieren der Endlosformulare im Drucker möglich;
- nach Druck der Überweisung, Gültigkeitsprüfung für Sammelüberweisung, ggf. mit Fehlermeldung;
- Druck Sammelüberweisung;
- Überweisungsdatenerfassung über Tastatur und unabhängig davon;
- Datenspeicherung durch Programmergänzung (DATA-Zeilen) möglich;
- wahlweise Bedrucken von neutralen und nicht neutralen Überweisungen.

Bedienungsanleitung

Nach dem Einladen und Starten des Programms mit »RUN« erscheint das Auswahlmenü auf dem Bildschirm. Der Benutzer wählt hier eine bestimmte Funktion durch Betätigen der jeweils zugeordneten Taste an. Die Funktionen »1«, »2« und »3« beinhalten den Ausdruck von Listen auf dem angeschlossenen Drucker (Liste Empfänger-, Verwendungszweck- und Auftraggeberdaten). Wenn die Taste »5« gedrückt wird, erfolgt Programmbeendigung. Durch Betätigen der Taste »4« gelangt der Bediener in den Programmteil »Druck Überweisungen«. Dazu erscheint das Eingabemenü auf dem Monitor. Der Anwender wird aufgefordert, durch Eingabe der entsprechenden Codenummern aus den gespeicherten Einzeldaten die gewünschten Überweisungsdaten auszuwählen. Gibt er hier beispielsweise die Codenummern »0« (Null) ein, so erscheinen die unter dieser laufenden Nummer gespeicherten Daten auf dem Bildschirm. Der Benutzer hat nun wiederum vier Auswahlmöglichkeiten: Betätigt er die Taste »D«, so werden die auf dem Bildschirm befindlichen Einzelüberweisungsdaten auf den bereits im Drucker eingespannten Endlosformularen ausgedruckt. Die »K«-Taste ermöglicht die Korrektur oder Abänderung der Überweisungsdaten. Durch Drücken der Taste »N« springt das Programm zurück zur Codennummereingabe, d.h. es wird so die erneute Auswahl aus den bereits im Speicher befindlichen Datenbausteinen ermöglicht. Taste »E« an dieser Stelle bewirkt die Beendigung des Programms.

Nach Druck Einzelüberweisung (»D«-Taste) erfolgt die Gültigkeitsprüfung und Aktualisierung der Summenspeicher. Zunächst entscheidet der Anwender durch Tastendruck, ob die soeben gedruckte Einzelüberweisung für eine Sammelüberweisung berücksichtigt werden soll (Taste »G« = Gültig, Taste »U« = Ungültig). Nach der Bedienerentscheidung »Gültig« werden die Überweisungsdaten plausibilisiert. Dabei prüft der Computer, ob die Auftraggeberdaten der zuletzt gedruckten Einzelüberweisungen identisch sind. Im Fehlerfall erscheint eine entsprechende Meldung auf dem Monitor des Computers. Nach einer fehlerfreien Prüfung erfolgt die Fortschreibung der Summenbelegspeicher und die Ja/Nein-Abfrage, ob die Sammelüberweisung gedruckt werden soll. Die »N«-Taste ermöglicht das Erfassen und Bedrucken weiterer Einzelüberweisungen, »J« aktiviert den Programmteil »Druck Summenbeleg«. Die Summenspeicher werden anschließend geleert, das Programm verzweigt erneut zur Arbeitswahl.

Der Computer übernimmt weitestgehend die Bedienerführung, da das vorliegende Programm »Druck Überweisungen« für jede Benutzerentscheidung Auswahlmasken zur Verfügung stellt. Mittels weniger definierter Tastendrucke kann jeder ungeübte Anwender das Programm in der Praxis einsetzen. Falsche Angaben werden ignoriert oder angezeigt.

◆ DRUCK BANKUEBERWEISUNGEN ◆

```

10 REM          DRUCK UEBERWEISUNGEN
20 REM          *****
30 :
40 :
50 REM  UNIVERSELL EINSETZBARES PROGRAMM ZUM
60 REM  AUSFUELLEN VON BANKUEBERWEISUNGS-FORMULAREN
70 :
80 POKE59468,12:PRINTCHR$(142):REM GROSS/GRAFIK
85 REM ZEILE 80 GILT NICHT FUER VC20 !
90 :
100 REM  VARIABLENTABELLE
110 REM  A,B,C,A$,B$,X$ = HILFSVARIABLEN
120 REM  I = LAUFVARIABLE
130 REM  EL = SELECTION EMPF.DATEN
140 REM  VZ = SELECTION VERWENDGSZWECK
150 REM  AL = SELECTION AUFTRAGGEBER
160 REM  D$ = PRUEFVARIABLE AUFTRAGGEBER
170 REM  FUER SAMMELUEBERWEISUNG:
180 REM  S,S$ = FORMATIERUNG DM-BETRAG
190 REM  AU = ANZAHL UEBERWEISUNGEN
200 :
210 I=0:EL=I:VZ=I:AL=I:A=I:B=I
220 B$="":X$=B$
230 :
240 REM  AUSWAHLMENUE
250 PRINT"☐ PROGRAMM-AUSWAHL: ■"
260 PRINT:PRINT
270 PRINT"☐ 1 ■ = LISTE EMPFAENGER"
280 PRINT"☐ 2 ■ = LISTE VERWENDUNGSZWECK"
290 PRINT"☐ 3 ■ = LISTE AUFTRAGGEBER"
300 PRINT"☐ 4 ■ = DRUCK UEBERWEISUNG"
310 PRINT"☐ 5 ■ = ENDE PROGRAMM"
320 PRINT"☐ BITTE WAEHLN !!!"
330 :
340 GETX$:IFX$<"1"ORX$>"5"THEN340
350 ONVAL(X$)GOTO380,410,440,470,1800
360 GOTO340
370 :
380 REM  LISTE EMPFAENGER
390 GOSUB1830:LIST2000-2999
400 :
410 REM  LISTE VERWENDUNGSZWECK
420 GOSUB1830:LIST3000-3999
430 :
440 REM  LISTE AUFTRAGGEBER
450 GOSUB1830:LIST4000-4999
460 :
470 REM  DRUCK UEBERWEISUNG, DIMENSIONIERUNG
480 DIMEL$(20,5),VZ$(20,4),AL$(5,2)
490 :

```

```

500 REM READ DATA'S
510 RESTORE:A=-1
520 A=A+1:READA$:IFA$="*"THEN:A=-1:GOTO550
530 FORI=1TO5:READEL$(A,I):NEXTI:GOTO520
540 :
550 A=A+1:READA$:IFA$="*"THEN:A=-1:GOTO580
560 FORI=1TO4:READVZ$(A,I):NEXTI:GOTO550
570 :
580 A=A+1:READA$:IFA$="*"THEN:A=-1:GOTO610
590 FORI=1TO2:READAL$(A,I):NEXTI:GOTO580
600 :
610 PRINT"CODENUMMERN FUERN"
620 PRINTTAB(10)"EMPFANERNAME,"
630 PRINTTAB(10)"VERWENDUNGSZWECK,"
640 PRINTTAB(10)"AUFTRAGGEBER"
650 PRINT"NEINGEBEN !"
660 PRINT"X(0,0,0) ODER X, X BEDEUTET FORMULARTEST)"
670 :
680 PRINT"X"
690 :
700 INPUT"EINGABEN: ";EL,VZ,AL
710 :
720 PRINT"GEWAHLTE UEBERWEISUNGSDATEN:"
730 PRINT"X-----"
740 FORI=1TO5:IFEL$(EL,I)=""THEN:EL$(EL,I)=CHR$(160)
750 PRINTEL$(EL,I):NEXTI
760 PRINT"-----"
770 B#=VZ$(VZ,4):B=LEN(B#)
780 IFB<14THEN:VZ$(VZ,4)=LEFT$("XXXXXXXXXXXX",14-B)+B#
790 FORI=1TO4:IFVZ$(VZ,I)=""THEN:VZ$(VZ,I)=CHR$(160)
800 PRINTVZ$(VZ,I):NEXTI
810 PRINT"-----"
820 FORI=1TO2:IFAL$(AL,I)=""THEN:AL$(AL,I)=CHR$(160)
830 PRINTAL$(AL,I):NEXTI
840 PRINT"-----"
850 :
860 PRINT"X D X = DRUCK      X K X = KORREKTUR"
870 PRINT"X N X = NEUE EINGABEN  X E X = ENDE PROG"
880 PRINT"XXXXXXXXXXXXBITTE WAHLEN !"
890 :
900 GETX$:IFX$="D"THEN1090
910 IFX$="K"THEN960
920 IFX$="N"THEN610
930 IFX$="E"THEN1800
940 GOTO900
950 :
960 PRINT"XKORREKTUR UEBERWEISUNGSDATEN:"
970 PRINT"X-----"
980 FORI=1TO5:PRINT"  EL$(EL,I):PRINT"X";
990 INPUTEL$(EL,I):NEXTI
1000 PRINT"-----"
1010 FORI=1TO4:PRINT"  VZ$(VZ,I):PRINT"X";

```

```

1020 INPUTVZ$(VZ,I):NEXTI
1030 PRINT"-----"
1040 FORI=1TO2:PRINT"  AL$(AL,I):PRINT"J";
1050 INPUTAL$(AL,I):NEXTI
1060 PRINT"-----"
1070 GOTO720
1080 :
1090 REM DRUCKROUTINE
1100 CLOSE1:OPEN1,4:REM DRUCKER 4 EROEFFNEN
1110 PRINT#1,EL$(EL,1);SPC(43-LEN(EL$(EL,1)))EL$(EL,3)
1120 PRINT#1,EL$(EL,2)
1130 PRINT#1
1140 PRINT#1,EL$(EL,4);SPC(16-LEN(EL$(EL,4)))EL$(EL,5)
1150 PRINT#1:PRINT#1:PRINT#1:PRINT#1
1160 PRINT#1,VZ$(VZ,1)
1170 PRINT#1,VZ$(VZ,2)
1180 PRINT#1,VZ$(VZ,3);SPC(41-LEN(VZ$(VZ,3)))VZ$(VZ,4)
1190 PRINT#1:PRINT#1:PRINT#1
1200 PRINT#1,AL$(AL,1);SPC(16-LEN(AL$(AL,1)))AL$(AL,2)
1210 FORI=1TO10:PRINT#1:NEXTI
1220 CLOSE1
1230 :
1240 REM GUELTIGKEITSPRUEFUNG
1250 PRINT"DRUCK BEENDET !!!-■"
1260 PRINT"DRUCK FUER SAMMELUEBERWEISUNG"
1270 PRINT"■ G ■ = GUELTIG      ■ U ■ = UNGUELTIG"
1280 PRINT"■■■■■■■■■■BITTE WAEHLN !"
1290 :
1300 GETX$:IFX$="U"THEN1640
1310 IFX$<"G"THEN1300
1320 :
1330 REM PLAUSIBILIERUNG
1340 B$=VZ$(VZ,4):B=LEN(B$)
1350 FORI=1TOB:IFMID$(B$,I,1)="X"THEN:NEXTI:GOTO1660
1360 C=VAL(MID$(B$,I)):IFC<=0THEN1660
1370 :
1380 IFD$<" "THEN:IFAL$(AL,1)<D$THEN1730
1390 D$=AL$(AL,1):REM PRUEFEN OB GLEICHER AUFTRAGGEBER
1400 :
1410 REM SUMMENSPEICHER
1420 S=S+C:REM DM-BETRAG
1430 AU=AU+1:REM ANZAHL UEBERWEISUNGEN
1440 :
1450 PRINT"DRUCK SAMMELUEBERWEISUNG (J/N)"
1460 :
1470 GETX$:IFX$="N"THEN1640
1480 IFX$<"J"THEN1470
1490 S$=MID$(STR$(S-INT(S)),3,2)
1500 S$="."+S$+LEFT$("00",2-LEN(S$))
1510 S$=MID$(STR$(INT(S)),2)+S$
1520 S$=LEFT$("XXXXXXXXXXXXXXXX",14-LEN(S$))+S$

```

```

1530 :
1540 CLOSE1:OPEN1,4:REM DRUCKER 4 EROEFFNEN
1550 PRINT#1,CHR$(1)"-SUMMENBELEG-"
1560 FORI=1TO9:PRINT#1:NEXTI
1570 PRINT#1,AU:"UEBERWEISUNGEN";SPC(24);S$
1580 PRINT#1:PRINT#1:PRINT#1
1590 PRINT#1,AL$(AL,1);SPC(16-LEN(AL$(AL,1)))AL$(AL,2)
1600 FORI=1TO10:PRINT#1:NEXTI
1610 S=0:AU=0:D$="":REM SUMMENSPEICHER LEEREN
1620 CLOSE1
1630 :
1640 GOTO720
1650 :
1660 REM UNGUELTIG
1670 PRINT"XODIE SOEBEN AUSGEDRUCKTE UEBERWEISUNG"
1680 PRINT"IST UNGUELTIG UND WIRD AUF DER SAM-"
1690 PRINT"MELUEBERWEISUNG NICHT BERUECKSICHTIGT!"
1700 FORI=1TO5000:NEXT
1710 GOTO720
1720 :
1730 REM AUFTRAGGEBERKONTONUMMER FALSCH
1740 PRINT"XOAUFTRAGGEBERKONTO FALSCH, KEIN"
1750 PRINT"SAMMELUEBERWEISUNGDRUCK ,MOEGLICH !"
1760 PRINT"XPROGRAMM NEU MIT 'RUN' STARTEN!"
1770 END
1780 :
1790 :
1800 REM ENDE PROGRAMM
1810 PRINT"XODIE E N D E " :END
1820 :
1830 PRINT"XDRUCKER EINGESCHALTEN, DANN"
1840 PRINT"X 'RETURN' - TASTE DRUECKEN !"
1850 GETX$:IFX$<>CHR$(13)THEN1850
1860 PRINT"XNACH DRUCKBEENDIGUNG PROGRAMM"
1870 PRINT"XMIT 'PRINT#1:CLOSE1:RUN' ERNEUT"
1880 PRINT"XSTARTEN."
1890 CLOSE1:OPEN1,4:CMD1:RETURN
1900 :
1910 REM DATENSPEICHERUNG IN DATA-ANWEISUNGEN
1920 REM (MAX. FELDLAENGE IN KLAMMERN)
1930 REM DIE REIHENFOLGE DER FELDER IST UNBEDINGT
1940 REM EINZUHALTEN !
1950 REM -----
1960 :
2000 REM LISTE EMPFAENGER
2010 REM FELDNAMEN (FELDLAENGEN)
2020 :
2030 REM CODENUMMER (2)
2040 REM EMPFAENGERNAME (37)
2050 REM PLZ ORT STRASSE (37)
2060 REM BANKLEITZAHL (12)
2070 REM KONTONR (15)

```

```

2080 REM BANKENNAME (37)
2090 :
2100 DATA 0
2110 DATA XX-EMPFANERNAME-XXXXXXXXXXXXXXXXXXXXX
2120 DATA PLZX XX-ORT-XXXXXXXXXX X-STRASSE-XX
2130 DATA XX-BLZ-XXXXX
2140 DATA XX-KTONR.-XXXXX
2150 DATA XX-BANKENNAME-XXXXXXXXXXXXXXXXXXXXX
2160 DATA 1
2170 DATA FRITZ MUSTERMANN
2180 DATA 6000 FRANKFURT ZEIL 999
2190 DATA 123 456 7
2200 DATA 111 111 1
2210 DATA MUSTERBANK
2980 DATA *:REM READ DATA TRENNZEICHEN
2990 REM -----
3000 REM LISTE VERWENDUNGSZWECK
3010 REM FELDNAMEN (FELDLAENGEN)
3020 :
3030 REM CODENUMMER (2)
3040 REM TEXT 1 (37)
3050 REM TEXT 2 (37)
3060 REM TEXT 3 (37)
3070 REM DM-BETRAG (14)
3080 :
3090 DATA 0
3100 DATA XX-TEXT 1-XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3110 DATA XX-TEXT 2-XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3120 DATA XX-TEXT 3-XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3130 DATA XX-DM-BETRAG-XX
3140 DATA 1
3150 DATA , RECHNUNGSNR. XXX
3160 DATA VOM XX.XX.19XX
3170 DATA 125.60
3980 DATA *:REM READ DATA TRENNZEICHEN
3990 REM -----
4000 REM LISTE AUFTRAGGEBER
4010 REM FELDNAMEN (FELDLAENGEN)
4020 :
4030 REM CODENUMMER (2)
4040 REM KONTONR (15)
4050 REM NAME PLZ ORT STRASSE (37)
4060 :
4070 DATA 0
4080 DATA XX-KTONR.-XXXXX
4090 DATA XX-NAME-XX PLZX XXX-ORT-XXX XX-STR-XX
4100 DATA 1
4110 DATA 345 678 9
4120 DATA "ZAHLEMANN & SOEHNE , BERLIN"
4980 DATA *:REM READ DATA TRENNZEICHEN
4990 REM -----

```

◆ RENDERUNG FUER NEUTR. UEBERN. ◆

```
1104 REM 1105, 1106 GILT NUR FUER NEUTRALE FORMULARE
1105 PRINT#1,"AUFTRAGGEBERBANKENNAME           1234567"
1106 PRINT#1:PRINT#1
1210 FORI=1TO7:PRINT#1:NEXTI
1544 REM 1545, 1546 GILT NUR FUER NEUTRALE FORMULARE
1545 PRINT#1,"AUFTRAGGEBERBANKENNAME           1234567"
1546 PRINT#1:PRINT#1
1600 FORI=1TO7:PRINT#1:NEXTI
```

Programmbeschreibung

Die Programmzeilen 10 bis 220 enthalten die Beschreibung der Variablen, Zeile 80 schaltet den Zeichengenerator auf Groß-/Grafikzeichensatz um. Diese Zeile gilt nicht für VC 20. Das Hauptmenü befindet sich in 240 bis 360. In 380 bis 450 sind die Befehle für Listenausgabe auf dem angeschlossenen Drucker untergebracht. Die Dimensionierung der Variablentabelle in Zeile 480 ist derzeit auf maximal 20 Empfänger, 20 Verwendungszwecke und 5 Auftraggeber festgelegt und kann durch entsprechende Änderungen bis zur maximalen Arbeitsspeicherkapazität erhöht werden. Aus den DATA-Statements werden anschließend die Überweisungsdaten in die Variablentabellen übernommen (500 bis 590). Bei der Datenerfassung in DATA-Zeilen sind deshalb die vorgeschriebene Reihenfolge, Anzahl und Länge der Datenfelder genau einzuhalten. Außerdem benötigt das Programm zwischen den einzelnen Datensätzen ein Multiplikationszeichen (★) als Datentrennzeichen, um die Variablentabelle richtig zu füllen. Wie Sie wissen, können in einer DATA-Zeile mehrere Variablen hintereinander, durch Komma getrennt, aufgeführt sein. Beachten Sie aber, daß Komma am Anfang oder am Ende eines DATA-Statements für »READ« eine zusätzliche Variable bereitstellt (siehe Zeile 3150). Sie können Kommata aber auch innerhalb eines alphanumerischen Feldes als Trennzeichen benutzen, wenn Sie den betreffenden String mit Gänsefüßchen (CHR\$(34)) versehen (siehe Zeile 4120). Allerdings ist dann keine Datenkorrektur über Tastatur möglich, da alle nach dem Komma folgenden Zeichen bei »INPUT« normalerweise ignoriert werden. Das Feld »Codenummer« sollte möglichst die laufende Datensatznummer innerhalb der jeweiligen Liste beinhalten.

Nachdem Sie jetzt wissen, was bei der Datenspeicherung alles zu beachten ist, zurück zum eigentlichen Programmablauf. Die Programmzeilen 610 bis 940 beinhalten die Codenummerneingabe, Darstellung der gewählten Überweisungsdaten auf dem Bildschirm und die Arbeitswahl mit der Verzweigung zu jeweiligen Programmteil. Eine Korrektur der zuvor ausgewählten Überweisungsdaten wird durch die Zeilen 960 bis 1070 ermöglicht. Die Druckersteuerung zum Bedrucken von Einzelüberweisungsformularen befindet sich in den Programmzeilen 1090 bis 1220. In 1240 bis 1430 ist die Gültigkeitsprüfung mit anschließender Fehlerkontrolle und Aktualisierung der Summenspeicher untergebracht. Nach der Abfrage »Druck Sammelüberweisungen Ja/Nein« in 1450 wird in 1490 bis 1520 der Gesamtüberweisungsbetrag formatiert. Anschließend folgt der Sammelüberweisungsdruck (1540 bis 1640). Die Zeilen 1660 bis 1760 bewirken den Ausdruck von Fehlermeldungen auf dem Bildschirm. In 1830 bis 1890 befindet sich ein Unterprogramm zur Druckersteuerung bei Listenausdrucken. Ab der Programmzeile 1910 beginnen die Datensatzbeschreibungen einschließlich der Überweisungsdaten, abgespeichert in DATA-Statements.

Wie abgebildet, eignet sich das vorliegende Basicprogramm zum Ausfüllen von nicht neutralen Formularen. Um neutrale Überweisungsträger zu beschriften, sind die dem Programmlisting folgenden Ergänzungen bzw. Programmänderungen vorzunehmen.

Ohne Worte

◆◆◆ SIRENE ◆◆◆

```
100 REM *** SIRENE ***
110 :
120 REM * VARIABLEN: I,J,U
130 :
140 POKE59467,16:POKE59466,156
150 :
160 FORI=1T03:FORJ=1T02
170 FORU=60T0255:POKE59464,U:NEXTU
180 FORU=1T0150:NEXTU
190 FORU=255T060STEP-1:POKE59464,U:NEXTU
200 NEXTJ:NEXTI
210 :
220 POKE59467,0
```

◆◆◆ FEUERWEHR ◆◆◆

```
100 REM *** FEUERWEHR ***
110 :
120 REM * VARIABLEN: I,U,X1,X2
130 :
140 POKE59467,16:POKE59466,156:X1=4
150 :
160 FORI=1T0X1:X2=4
170 IFI>1THENX2=1
180 POKE59464,223:GOSUB270
190 POKE59464,149:GOSUB270
200 IFI=X1THENX2=4:GOSUB270
210 POKE59464,0
220 NEXTI
230 :
240 POKE59467,0
250 END
260 :
270 FORU=1T0100*X2:NEXTU:RETURN
```

◆◆◆ TELEFON ◆◆◆

```
100 REM *** TELEFON ***
110 :
120 REM * VARIABLEN: J,I,X1,U
130 :
140 POKE59467,16:POKE59466,85:X1=59464
150 :
160 FORJ=1T05
170 :
180 FORI=1T050
190 POKEX1,130:FORU=1T03:NEXT
200 POKEX1,250:FORU=1T05:NEXT
210 NEXTI
220 :
230 POKEX1,0:FORU=1T01000:NEXTU,J
240 :
250 POKE59467,0
```

◆◆◆ AMERIK. SIRENE ◆◆◆

```
100 REM *** AMERIK. SIRENE ***
110 :
120 REM * VARIABLEN: U,V
130 :
140 POKE59467,16:POKE59466,150
150 :
160 FORV=1T05
170 FORU=10T0255STEP9
180 POKE59464,U
190 NEXTU,V
200 :
210 POKE59467,0
```

◆◆◆ VOGELGEZWITSCHER ◆◆◆

```

100 REM *** VOGELGEZWITSCHER ***
110 :
120 REM * VARIABLEN: I,U,X1
130 :
140 POKE59467,16:POKE59466,85:X1=59464
150 :
160 FORU=1TO40
170 FORI=180TO40STEP-6:POKEX1,I:NEXTI
180 POKEX1,0
190 FORI=1TOINT(300*NRND(1)):NEXTI
200 NEXTU
210 :
220 POKE59467,0

```

◆◆◆ HR III ◆◆◆

```

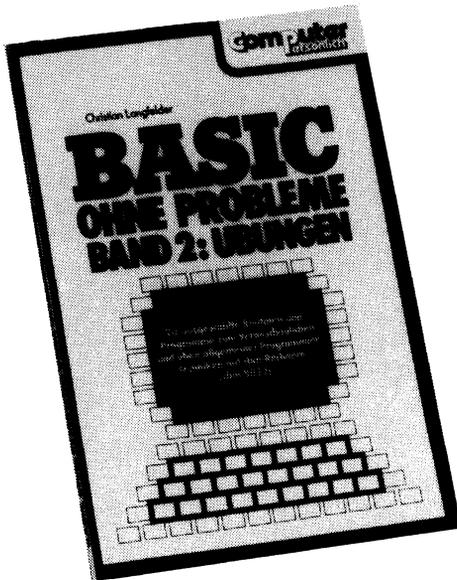
100 REM *** HR III ***
110 :
120 REM * VARIABLEN: A0,A1,A2,A3,X1,X2,U
130 :
140 A0=150:A1=250:A2=210:A3=190:X1=59464
150 POKE59467,16:POKE59466,A0
160 POKEX1,A1:GOSUB270:POKEX1,0:X2=1:GOSUB260
170 POKEX1,A2
180 GOSUB270:X2=2:GOSUB260:POKEX1,0:X2=1:GOSUB260
190 POKEX1,A1:X2=1:GOSUB260:POKEX1,0:GOSUB270
200 POKEX1,A2
210 X2=1:GOSUB260:POKEX1,0:X2=2:GOSUB260
220 POKEX1,A3:X2=4:GOSUB260
230 POKE59467,0
240 END
250 :
260 FORU=1TO110*X2:NEXT:RETURN
270 FORU=1TO60:NEXT:RETURN

```

◆◆◆ BR III ◆◆◆

```
100 REM *** BR III ***
110 :
120 REM * VARIABLEN: A0,A1,A2,A3,A4,X1,U
130 :
140 A0=150:A1=200:A2=150:A3=120:A4=100:X1=59464
150 POKE59467,16:POKE59466,A0
160 POKEX1,A1:GOSUB280
170 POKEX1,A2
180 GOSUB280:POKEX1,0:GOSUB280
190 POKEX1,A2:GOSUB280
200 POKEX1,A3
210 GOSUB280:POKEX1,0:GOSUB280
220 POKEX1,A3:GOSUB280
230 POKEX1,A4:GOSUB280:GOSUB280:GOSUB280
240 POKEX1,A3:GOSUB280
250 POKE59467,0
260 END
270 :
280 FORU=1TO110:NEXT:RETURN
```

M&T-Neuerscheinungen 1983!



Ch. Langfelder

BASIC ohne Probleme Band 2: Übungen

20 ausgewählte Routinen und Programme zum Veranschaulichen und Üben allgemeiner Programmierertechniken auf CBM-Rechnern (CBM 8032)

Die Programme setzen sich aus einer Beschreibung, Bedienungsanweisungen, Beispielen, Anmerkungen und reichlich mit Kommentaren versehenen Programmprotokollen zusammen. Die Information wird, sofern sinnvoll und erforderlich, durch Tips, Erläuterungen und Flußdiagramme ergänzt. Die Programme haben in erster Linie die Aufgabe, allgemeine Programmierertechniken zu veranschaulichen. Die Betonung liegt auf den alltäglichen und daher besonders wichtigen Problemen in der Programmierung: die Behandlung und Verarbeitung von Eingabedaten, die Steuerung des Cursors, die Verwendung von Unterprogrammen und die Programmierung von Fehlerausgängen.

1983, 119 Seiten

Bestellnummer MT 490

DM 24,—*

BASIC ohne Probleme Band 1: Unterweisung

Vom selben Autor erscheint eine Einführung in BASIC mit CBM-Rechnern (ca. 170 Seiten).

In 12 Kapiteln wird der Leser Schritt für Schritt mit BASIC, dem CBM-Rechner und seiner Bedienung vertraut gemacht. Jedes Kapitel schließt mit Übungen und Aufgaben ab — als Kontrolle für den jeweiligen Wissensstand. Im Anhang befinden sich dann unter anderem die Lösungen, ein Glossar, ein Stichwortregister usw.

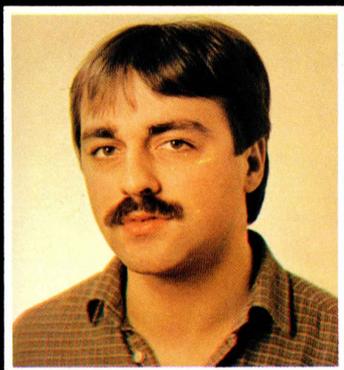
Bestellnummer MT 480

DM 29,80*

*Die Preise verstehen sich inkl. MwSt. zuzügl. Versandkosten.

Markt&Technik

Hans-Pinsel-Str. 2, 8013 Haar bei München, Telefon 089/4613-225



KARL-HEINZ HESS

wurde am 10. März 1957 in Schlüchtern/Hessen geboren. Nach dem Besuch der zweijährigen kaufmännischen Berufsfachschule wechselte er zur Fachoberschule über, wo er die Fachhochschulreife für Ingenieurwesen erwarb. Während seiner Berufsausbildung zum Technischen Zeichner in einem feinmechanischen Produktionsbetrieb verlagerte sich sein Interesse zunehmend in den Bereich der elektronischen Datenverarbeitung.

Durch eigene Initiative und unterstützt durch innerbetriebliche Fortbildungsmaßnahmen, eignete er sich grundlegende EDV-Kenntnisse an. Umfangreiches Fachwissen erlangte er innerhalb einer mehrjährigen beruflichen Praxis im Umgang und in der Programmierung von Computern. Dabei lernte er u. a. die speziellen Eigenschaften und Leistungsmerkmale der CBM-Computer genau kennen. Heute ist er Leiter der EDV/Organisation und hat das Programmieren längst zu seinem bevorzugten Hobby erklärt.

BASIC-Programme für CBM/VC 20-Computer

Ein Personal Computer kann zur Lösung spezieller beruflicher Aufgabenstellungen, zur Bewältigung forschender und lehrender Tätigkeiten genauso eingesetzt werden wie zur Entspannung und Unterhaltung. Die schier unerschöpfliche Vielfalt breiter Anwendungsmöglichkeiten verursacht demnach eine große Nachfrage nach nützlichen und allgemein verständlichen Programmen.

Dieses Buch soll sowohl dem Computerneuling als auch dem Basic-Profi die programmtechnische Lösung bestimmter Probleme aufzeigen. Dabei beschränkt sich die Problemlösung nicht etwa auf die Darstellung des Programmlistings. Die verschiedenen Aufgabenstellungen werden genau analysiert, allgemeingültige Lösungswege erarbeitet und in CBM-Basic konvertiert. Alle Programme sind darüber hinaus ausführlich dokumentiert und anwendbar für die Serien CBM 2000, 3000, 4000 und 8000. Sie wurden im praktischen Einsatz sorgfältig getestet. Einige Programme laufen auch auf VC 20 und anderen basicprogrammierbaren Rechnern, wobei etwaige Programmanpassungen näher beschrieben sind. Die Funktionsweise und Bedienungen der einzelnen Programme wird durch eine umfassende Erläuterung des jeweils zugrundeliegenden Sachverhalts auch für den Laien verständlich. Dem Profi, der ähnliche Aufgabenstellungen, wie in diesem Buch beschrieben, zu bewältigen hat, kann eine Orientierung an den hier vorgegebenen Lösungswegen durchaus hilfreich sein.

So versteht sich dieses Buch nicht nur als Sammlung und Nachschlagewerk fertiger Basic-Programme, die einmal eingetippt, ständig zur Lösung eines bestimmten Problems dienen. Bei der Programmierung wurde auch besonderes Augenmerk darauf verwendet, dem Benutzer durch einfache Parametersteuerungen zu ermöglichen, die abgebildeten Programme seinen spezifischen Anforderungen und darüber hinaus der eigenen Hardware-Konfiguration bequem anzupassen. In einigen Abschnitten werden nützliche Tips rund um das Programmieren erläutert.

Programme: Etiketten-Beschriftung · Lotto: 6 aus 49/7 aus 38 · Hardcopy für alle CBM-Typen · Kopfrechen-Training · Bildschirmrahmen · Römische Zahlen · Kugelabschnitt · Der Zeichengenerator des CBM · Entwurf von Sonderzeichen durch Nadelansteuerung · Code-Umwandlung · Horizontale Balkendiagramme · Laufbandanzeige · Banküberweisungen ausfüllen