Lüers

MSX

Programm-sammlung

EIN DATA BECKER BUCH

Lüers

MSX

Programm-sammlung

EIN DATA BECKER BUCH

ISBN 3-89011-090-8

Wichtiger Hinweis:
Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technischen Angaben und Programme in diesem Buch wurden von dem Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler ist der Autor jederzeit dankbar.

INHALTSVERZEICHNIS
==========

Sehr geehrte, liebe MSX-Freunde und MSX-Interessenten!

Bisher schauten die Privatleute, die sich nach reiflicher Überlegung zum Kauf eines Home- oder Heimcomputers entschlossen hatten, neidisch auf die größeren Computer, denn unter diesen Gerätefamilien waren schon vor Jahren vorzügliche Standards geschaffen worden. Die Namen 'CP/M' und 'MS-DOS' sowie der Ausdruck 'IBM-Kompatibilität' gewährleisteten und tun dies gerade heute in immer stärkerem Maße, daß die Programmvielfalt von Computer X auch auf Computer Y ohne Probleme läuft und umgekehrt selbstverständlich ebenso. Da gibt es keine Eifersuchtsprobleme nach dem Motto 'Warum kann ich diese Datenbank ausgerechnet nicht auf meinem Computer laufen lassen?!'.

Wie sah und sieht es heute noch bei den meisten Homecomputern aus? Gerätehersteller X bringt einen Computer heraus, zu dem natürlich die Geräteanschlüsse und auch die Programmvielfalt nur dieses einen Herstellers passen. So zieht der Kauf des Computers X im Normalfall auch ein beachtliches Nachfolgegeschäft nur bei diesem Hersteller nach sich, woran sich manche bereits ein goldenes Näschen verdient haben. Ist der Markt mit Produkt X halbwegs gesättigt, bringt der gleiche Hersteller ein Produkt Y heraus. Damit auch hiernach wieder per Nachfolgegeschäft die Kassen klingeln, werden halt neue Anschlüsse (wieder keinem Standard genügend) am Computer angebracht, und - wo käme der arme Hersteller denn sonst auch hin - die Software von Produkt X läuft selbstverständlich nicht mehr auf dem Computer Y. Ob dieser Hersteller nicht irgendwann einmal mit dieser Politik auf die Nase fallen wird ... warten wir es ab ... die MSX-Computer können nicht nur, sie müssen Sieger am Ende der Wegstrecke sein, denn hier wird niemand mehr für 'dumm' verkauft.

Wie kann so etwas gelingen? Alle japanischen Firmen aus dem HIFI-VIDEO-Bereich haben sich vor einiger Zeit gemeinsam an einen großen Tisch gesetzt und beratschlagt, wie den Wirtschaftsmächten USA und Europa mal wieder ein Schnäppchen mit bombastischer Wirkung geschlagen werden könne (siehe auch die Entwicklung bei Kameras, dem HIFI-VIDEO-Sektor und der Automobilproduktion nicht zuletzt).

Da man den Homecomputermarkt nicht als erster betreten würde, mußte nun gemeinsam etwas besonderes ausgegoren werden. Schließlich einigte man sich darauf, Homecomputer auf den Markt zu bringen, die nicht nur Kompatibilität (Austauschbarkeit) bei der Software (Programme) sondern auch bei der Hardware (Anschlüsse zu weiteren Geräten) gewährleisten sollten.

Man stellte all' dieses unter den Tenor 'MSX', eine Abkürzung für das leistungsfähigste BASIC, das ... mal wieder von 'MICROSOFT', einer äußerst renommierten amerikanischen Softwarefirma, stammt: 'Microsoft Super Extended Basic'.

Doch der MSX-Standard umfaßt mehr: Nicht nur das BASIC ist gleich, auch die Computerchips für Sound und Grafik (zu diesem Zentralthema gibt es übrigens ein weiteres DATA BECKER-Buch: 'MSX Sound und Grafik') gleichen bei allen MSX-Computern wie ein Ei dem anderen. Außerdem hat jeder MSX-Computer, soweit zur Zeit zu sehen, eine Standard Centronics Schnittstelle, die Verwendung von kommerziellen CP/M-Programmen (s.o. - eigentlich kommt diese Software ja von den größeren Computern her!) ist zu 100% möglich und last not least: Das MSX-Diskettenbetriebssystem MSX-DOS ist datenkompatibel zum MS-DOS von IBM ... d.h. Sie können sich Ihre Daten aus dem Büro nach Hause mitnehmen (Vorsicht! Datenschutz!) und in aller Ruhe auf Ihrem MSX-Computer mit angeschlossener Diskettenstation weiterverarbeiten!!!

Sie merken vielleicht, eine Überlegung der Japaner, die Hand und Fuß hat. Diese Idee muß einfach erfolgreich sein, anders kann es gar nicht sein!

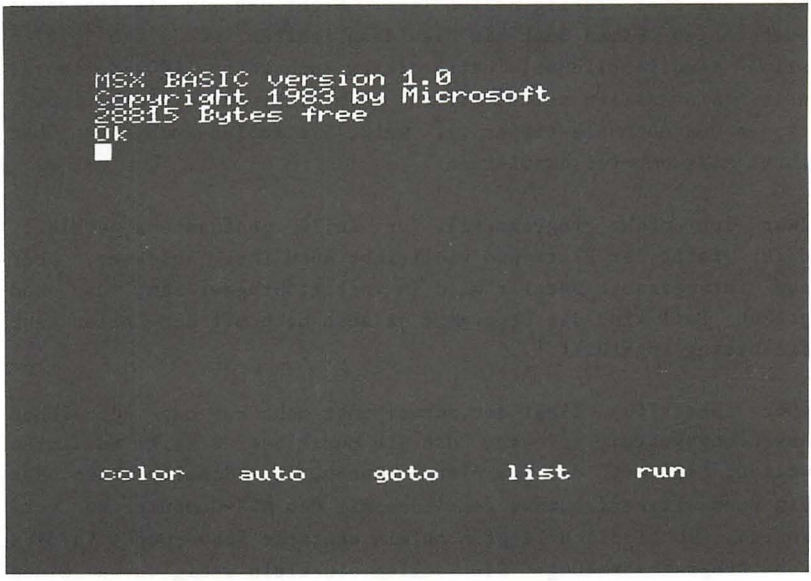
Zwei Punkte seien noch angefügt:

1) Weil die Hersteller von MSX-Computern fast allesamt aus der HIFI-VIDEO-Branche stammen, ist der Anschluß des MSX-Homecomputers z.B. an den Videorecorder, den Bildplattenspieler, die Tonbandmaschine usw. gar nicht mehr fern (Beispiele dafür gab es bereits 1984 auf der HIFIvideo in Düsseldorf en masse zu sehen). Also: ein Homecomputersystem, das wahrlich ins häusliche Innenleben hereingehört, ja, sich sogar in seine Umgebung schnellstens einpaßt.

2) Nicht nur die Anwendungsmöglichkeiten der MSX-Computer bestechen! Auch technische Daten wie Ausbaubarkeit auf 256 KB RAM, 32 Sprites, 16 Farben, hochauflösende Farbgrafik mit 256*192 Punkten und ein eingebauter dreistimmiger Soundprozessor sprechen für den Computerkenner geradezu Bände!

Warten wir ab, wie sich der Markt umkrempelt. Die Japaner (repräsentiert durch mehr als zwanzig Firmen mit renommierten Namen wie SONY, SPECTRAVIDEO, HITACHI, JVC usw.), die Koreaner (u.a. GOLDSTAR und SAMSUNG) und auch PHILIPS aus Deutschland erwarten, daß allein 1985 die Hälfte aller verkauften Homecomputer das Kürzel MSX tragen werden. Toi, Toi, Toi MSX!

P.S.: Die Begriffe 'Homecomputer' und 'Personalcomputer' sind inzwischen so nichtssagend, daß wir bei allen MSX-Computern in jedem Fall auch von Personalcomputern sprechen können.



Sehr geehrte, liebe Leserin!
Sehr geehrter, lieber Leser!
Sehr geehrte, liebe Programmiererin!
Sehr geehrter, lieber Programmierer!

Sie halten das Buch 'MSX Programmsammlung' in Händen und wollen sich vielleicht erst einmal ein schnelles Urteil über dieses Buch erlauben. Dazu haben Sie sicherlich einen Blick ins Inhaltsverzeichnis geworfen ... gut so ... und nun wollen Sie noch schnell die Anrede überfliegen ... und dann geht es ab ins stille Kämmerlein zum Programmieren bzw. zum Programm eintippen in Ihren MSX-Computer!?

Lassen Sie mich bitte trotzdem noch für ein paar Worte zur Sprache kommen, bevor es richtig losgeht.

Hinter dieser Programmsammlung steckt ein anderes Konzept, als man vielleicht auf den ersten Blick ahnen könnte. Diese Programmsammlung besteht nämlich nicht nur aus Programmen, nein auch aus über vierzig Seiten Text. Nicht daß mir die Programme ausgegangen wären oder mir die Ideen fehlten (würde ich gleich alles veröffentlichen, hätte dieses Buch knapp 400 Seiten!), nein, Sie sollen meiner Meinung nach nicht nur einfach tippen, tippen und nochmals tippen ... nein, Sie sollen auch etwas über Ihren MSX-Computer dazulernen.

Zwar ist nicht Programmzeile für Zeile genauestens erklärt, dafür fehlt der Platz und vielleicht auch Ihre Ausdauer. Aber auf interessante Details wird im Vortext hingewiesen, und zum großen Teil sind die Programme ja auch noch mit REM-Zeilen zur Erläuterng angefüllt.

Auch inhaltlich liegt der Schwerpunkt mehr auf der nützlichen Anwendungsseite, sei es, daß Sie Musik oder Grafik editieren wollen, sei es, daß Sie planen können, Ihre Schallplatten oder die Bundesligaergebnisse gemeinsam mit dem MSX-Computer zu verwalten. Schließlich liegt noch ein weiterer Schwerpunkt in dieser Programmsammlung: Sie sollen ein klein wenig davon einen Eindruck bekommen, wie Ihr MSX-Computer 'innerlich' arbeitet.

Programme wie 'Disassembler' und 'Variablenreferenzliste' sind nur zwei Beispiele dafür.

Auf jeden Fall, und lassen Sie mich damit schließen, sollen Sie Spaß daran verspüren, in die geheimnisvolle Welt der Computer und des MSX-Computersystems im besonderen immer tiefer einzudringen und vielleicht sogar durch die Programmsammlung selbst zum kleinen Programmierer aufzusteigen.

Noch einmal: Viel Freude und viel Spaß an der Arbeit mit diesem DATA BECKER-Buch wünscht Ihnen der Autor von Programmen und Text.

MSX und Spectravideo 318/328

Da die Computer SVI-318 und SVI-328 vor der Einigung über die genauen Bestimmungen, was MSX eigendlich bedeutet, auf dem Markt erschienen, die Modelle SVI-318 und SVI-328 aber als "Vorreiter" der MSX-Computer angesehen werden dürfen, gibt es zwei Möglichkeiten, die Programme in diesem Buch auch auf SVI-318 (soweit der Speicherplatz reicht) und SVI-328 laufen zu lassen:

1) Die Firma Spectravideo liefert ein Modul für die Geräte 318/328, das diese Computer 100% MSX-fähig macht

2) Im Anschluß an diese Erläuterung habe ich in 7 Punkten zusammengefasst, was Sie in den Programmen dieser MSX Programmsammlung verändern müssen, damit auch Sie als SVI-318/328-User in den Nutzen der Programme kommen

Ausserdem bietet die Firma Spectravideo mit dem SVI-728 einen Computer an, der Voll MSX-fähig ist.

Unterschiede zwischen MSX- und Spectravideo 318/328-BASIC

1) Ein- Ausstellen von Tastaturklick und Funktionstastenanzeige

MSX

Klick: 3.Parameter beim SCREEN-Befehl auf "1"(an) oder "0"(aus)
Funktionsanzeige: "KEY ON"(an) oder "KEY OFF"(aus)

SVI-318/328

Klick: Befehl "CLICK ON"(an) oder "CLICK OFF"(aus)
Funktionstastenanzeige: bei "SCREEN" 2. Parameter auf "1"(an) oder "0"(aus) - nur bei SCREEN 1"
"KEY ON" und "KEY OFF" bedeutet: Interruptabfrage der Funktionstasten an oder aus (bei "MSX" nicht an/abschaltbar)

2) 'SCREEN'-Anweisung

MSX

SCREEN 0: 40 Zeichen/Zeile; durch 'WIDTH' bis 1 Zeichen/Zeile
SCREEN 1: 32 Zeichen/Zeile; durch 'WIDTH' bis 1 Zeichen/Zeile
SCREEN 2: hochauflösende Grafik 256*192 Punkte
SCREEN 3: niedrigauflösende Grafik 64*48 Punkte

SVI-318/328

SCREEN 0: 40 Zeichen/Zeile; durch 'WIDTH' bis 39 Zeichen/Zeile
SCREEN 1: hochauflösende Grafik 256*192 Punkte
SCREEN 2: niedrigauflösende Grafik 64*48 Punkte

3) Textausgabe auf Grafikbildschirm

MSX

'SCREEN 2:'
'OPEN "GRP:" FOR OUTPUT AS #1:'
z.B. 'DRAW "BM 128,96":' oder 'PRESET (128,96):'
'PRINT #1, "MSX"'
pro Zeile sind bis zu 32 Zeichen in Folge darstellbar

SVI-318/328

'SCREEN 1:'
'LOCATE 128,96:'
'PRINT "SVI-318/328'
pro Zeile sind bis zu 42 Zeichen in Folge darstellbar

4) Inverse Textdarstellung

MSX

muß mit einem Unterprogramm erzeugt werden (siehe Programme)

SVI-318/328

invers anschalten: 'PRINT CHR$(27);"p";'
invers ausschalten: 'PRINT CHR$(27);"q";'

5) Umlaute im Zeichensatz

MSX

in den für den europäischen Markt bestimmten MSX-Computern sind deutsche Umlaute und 'ß' bereits im Zeichensatz enthalten, aber noch an der falschen Stelle, was den Ausdruck auf Nicht-MSX-Druckern betrifft

'[' im Video-RAM-Speicher befindet sich an Speicherstelle 2776

der Zeichensatz ist ohne Zerstörung der ASCII-Zeichen umdefinierbar ab Speicherstelle 3072 im Video-RAM-Speicher

SVI-318/328

im Zeichensatz sind keine deutschen Umlaute und kein 'ß' enthalten. Sie müssen in jedem Fall neu definiert werden

'[' im Video-RAM-Speicher befindet sich an Speicherstelle 2520

der Zeichensatz ist ohne Zerstörung der ASCII-Zeichen umdefinierbar ab Speicherstelle 3584 im Video-RAM-Speicher
Nebenbei: Buchstabe 'ö' = 'LEFT GRAPH' + 'Ä'

6) Token

bis auf folgende sind die Token von MSX und SVI-318/328 gleich

MSX

199	SPRITE	200	VDP	201	BASE	202	CALL
203	TIME	204	KEY	205	MAX	206	MOTOR
207	BLOAD	208	BSAVE	238	>	239	=
240	<	241	+	242	-	243	*
244	/	245	^	246	AND	247	OR
248	XOR	249	EQV	250	IMP	251	MOD

252 bis 254: keine sichtbaren Tokens

SVI-318/328

199	KEY	200	CLICK	201	SWITCH	202	MAX
203	MON	204	MOTOR	205	BLOAD	206	BSAVE
207	MDM	208	DIAL	238	SPRITE	239	TIME
240	>	241	=	242	<	243	+
244	-	245	*	246	/	247	^
248	AND	249	OR	250	XOR	251	EQV
252	IMP	253	MOD	254			

7) Errormeldungen

bis auf folgende sind die Errormeldungen von MSX und SVI-318/328 gleich

MSX

```
ERROR 58: 'Sequential I/O only'
ERROR 59: 'File not OPEN'
ERROR 60: 'Unprintable error'
```

SVI

```
ERROR 58: 'Sequential after PUT'
ERROR 59: 'Sequential I/O only'
ERROR 60: 'File not OPEN'
```

Speicher 1

Schauen wir uns das Resultat dieses Programms am Bildschirm an, so stellen wir fest, daß der Speicher, dessen Inhalt wir mit dem Befehl 'PEEK' abfragen, nicht überall leer = '0' ist.

Zwar erscheinen auf dem Bildschirm nur anscheinend wirre Zahlen, jedoch werden wir gleich feststellen, daß sich hinter diesen Zahlen zum Teil sinnvolle Buchstaben und Zeichen verbergen. Hierzu ein Experiment:

Geben Sie folgendes in Ihren MSX-Computer ein:

'PRINT ASC("!")' (ENTER)

Ergebnis: 33. Das heißt: Hinter dem Ausrufungszeichen verbirgt sich für den Computer die Zahl 33. Probieren wir es andersherum: Geben Sie ein:

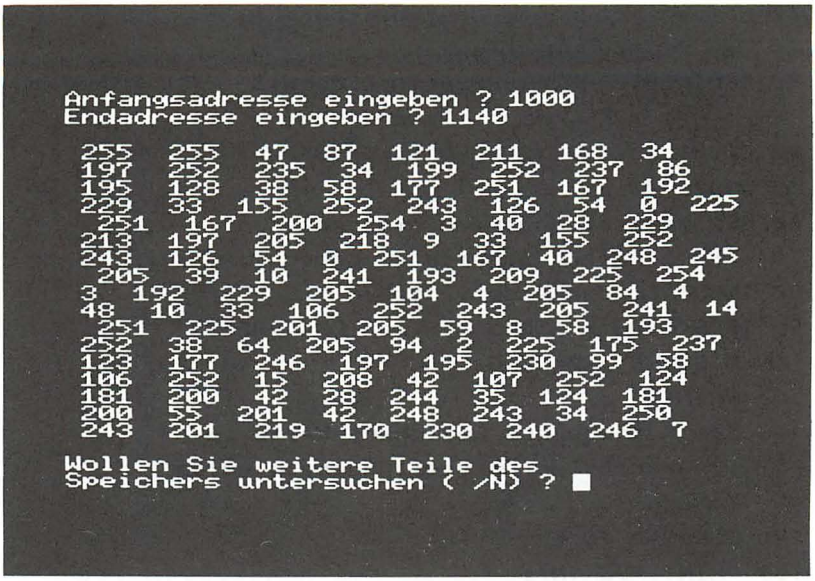
'PRINT CHR$(33)' (ENTER)

Ergebnis: '!'. Damit haben wir den Computer dazu aufgefordert, die Zahl 33 wieder in ein Zeichen umzuwandeln. Das Ergebnis von Zahl 33 ist unser Ausrufungszeichen!

So können wir nun auch unser Zahlenmeer aus Programm Speicher 1 zu deuten lernen. Geben Sie zur Überprüfung gleich anschließend bitte das Programm 'Speicher 2' ein.

Hinweis zum Gebrauch von Programm 'Speicher 1':

Anfangs- und Endspeicheradresse sollten nicht weiter als 140 Zeichen auseinanderliegen; so bekommen wir in der Darstellungsweise alle gewünschten Daten auf eine Bildschirmseite.


Anfangsadresse eingeben ? 1000
Endadresse eingeben ? 1140
255 255 47 87 121 211 168 34
197 252 235 34 199 252 237 86
195 128 38 58 177 251 167 192
229 33 155 252 243 126 54 0 225
251 167 200 254 3 40 28 229
213 197 205 218 9 33 155 252
243 126 54 0 251 167 40 248 245
205 39 10 241 193 209 225 254
3 192 229 205 104 4 205 84 4
48 10 33 106 252 243 205 241 14
251 225 201 205 59 8 58 193
252 38 64 205 94 2 225 175 237
123 177 246 197 195 230 99 58
106 252 15 208 42 107 252 124
181 200 42 28 244 35 124 181
200 55 201 42 248 243 34 250
243 201 219 170 230 240 246 7
Wollen Sie weitere Teile des
Speichers untersuchen (/N) ?

```
10 REM Untersuchung des Speichers 1
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 COLOR 15,1,15:SCREEN 0:KEY OFF
50 WIDTH 40
60 REM Darstellungsraum eingrenzen
70 INPUT "Anfangsadresse eingeben ";A
80 INPUT "Endadresse eingeben ";E
90 IF E<A OR E>65535! OR A<0 OR A>65535!
 THEN GOSUB 210:GOTO 50
100 PRINT
110 REM Ausgabe des Speicherinhalts im
    eingegrenzten Darstellungsraum durch
    Zahlen
120 FOR N=A TO E
130 PRINT PEEK(N);
140 NEXT N
150 PRINT
160 PRINT
170 INPUT "Wollen Sie weitere Teile des
           Speichers untersuchen ( /N) "
;F$
180 IF LEFT$(F$,1)="n" OR LEFT$(F$,1)="N
" THEN F$="N"
190 IF F$<>"N" THEN GOTO 10
200 END
210 PRINT:PRINT TAB(10) CHR$(200) "Falsc
he Eingabe!" CHR$(200)
220 GOSUB 230:RETURN
230 PRINT:PRINT TAB(6) "<Bitte eine Tast
e druecken>"
240 F$=INKEY$:IF F$="" THEN GOTO 240
250 RETURN
```

Speicher 2

Nachdem Sie dieses Programm abgetippt und es mit 'RUN' gestartet haben passiert folgendes: Was auch immer Sie tun, entweder passiert überhaupt nichts oder es passiert etwas Unvorhergesehenes (der Bildschirm wird gelöscht, Ihr MSX-Computer fängt wie wild zu 'beepen' an ...). Warum dies?

Wie schon in anderen Programmen dieser Sammlung aufgeführt: Es gibt sinnvoll direkt darstellbare Zeichen ('CHR$(32)' = Leerzeichen bis 'CHR$(126)' = Schlangenlinie bzw. bis 'CHR$(255)' = Grafikzeichen), es gibt aber auch andere CHR$-Zeichen, die sich nicht so eindeutig im Bild festhalten lassen ('CHR$(12)' = Bildschirm loeschen bzw. 'CHR$(7)' = 'BEEP' ertönen lassen - Einzelheiten siehe im Handbuch zu Ihrem MSX-Computer).

Also müssen wir noch ein drittes Programm schreiben, um nur die Zahlen als Zeichen auf dem Bildschirm entsprechend darzustellen, die in diesem Zusammenhang auch sinnvoll darstellbar sind. Dies geschieht nun mit einigen Programmerweiterungen in Programm 'Speicher 3'.

Hinweis zum Gebrauch von Programm 'Speicher 2':

Wenn auf dem Bildschirm nun nichts rechtes mehr zu erkennen ist, Sie aber das eingegebene Programm weiterbenutzen wollen, so starten Sie Programm 'Speicher 2' einfach wieder mit 'RUN'.

```
10 REM Untersuchung des Speichers 2
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 COLOR 15,1,15:SCREEN 0:KEY OFF
50 WIDTH 40
60 REM Darstellungsraum eingrenzen
70 INPUT "Anfangsadresse eingeben ";A
80 INPUT "Endadresse eingeben ";E
90 IF E<A OR E>65535! OR A<0 OR A>65535!
 THEN GOSUB 210:GOTO 50
100 PRINT
110 REM Ausgabe des Speicherinhalts im
    eingegrenzten Darstellungsraum durch
    Zahlen
120 FOR N=A TO E
130 PRINT CHR$(PEEK(N));
140 NEXT N
150 PRINT
160 PRINT
170 INPUT "Wollen Sie weitere Teile des
          Speichers untersuchen ( /N) "
;F$
180 IF LEFT$(F$,1)="n" OR LEFT$(F$,1)="N
" THEN F$="N"
190 IF F$<>"N" THEN GOTO 10
200 END
210 PRINT:PRINT TAB(10) CHR$(200) "Falsc
he Eingabe!" CHR$(200)
220 GOSUB 230:RETURN
230 PRINT:PRINT TAB(6) "<Bitte eine Tast
e druecken>"
240 F$=INKEY$:IF F$="" THEN GOTO 240
250 RETURN
```

Speicher 3

Mit Eingabe dieses Programms werden gleich zwei Fliegen mit einer Klappe geschlagen: Einerseits erscheint die Bildschirmwiedergabe des Speicherinhalts nun geordneter (jeweils Anfangsadresse, Inhalte der Anfangsadresse und der folgenden sieben Bytes sowie CHR$-Ausdruck der acht Bytes, alles zusammen in einer Bildschirmzeile), außerdem werden jetzt nur noch die sinnvollen 'CHR$(' dargestellt (zwischen 32 = Leerzeichen und 126 = Schlangenlinie). Für 'nicht ohne weiteres sichtbare CHR$' haben wir einfach 'CHR$(46)' = '.' zur Darstellung herangezogen.

Sie können wählen: entweder geben Sie die Anfangsadresse ein und drücken anschließend lediglich die (ENTER)-Taste; so werden nur die Anfangsadresse und die Inhalte der entsprechenden acht Bytes samt CHR$-Darstellung ausgedruckt, oder, um die Speicherinhalte auch über eine Zeile hinaus abbilden zu können, gilt es nach dem erstmaligen Drücken der (ENTER)-Taste dieselbige oder eine andere Taste fortlaufend niederzudrücken. So erscheinen zu je acht Bytes pro Zeile die sich anschließenden Speicherinhalte auf dem Monitor.

Schauen wir uns doch einmal an, wo unser BASIC-Programm 'Speicher 3' geblieben ist. Beim MSX-Computer wie auch bei jedem anderen Computer werden Befehle zum Teil nur als einzelne Zahlen = Token abgespeichert, hingegen bleiben Worte, die hinter REM-Anweisungen stehen, vollkommen erhalten. Geben Sie z.B. als Anfangsadresse '32906' ein ... rechts im CHR$-Bereich müßten Sie nun das Wörtchen 'MSX' entdecken ... schauen wir in das Programmlisting -) siehe Zeile 30:

```
'30 REM MSX Programmsammlung'
```

... das funktioniert allerdings nur, wenn Sie das Programm Zeile für Zeile genau so wie im Buch abgetippt haben, ohne auch nur ein Leerzeichen vergessen zu haben!

Geben wir '32906' als Anfangsadresse ein und drücken weiter irgendeine Taste, so werden uns noch weitere Bestandteile aus unserem Programm 'Speicher 3' begegnen.

Hinweis zum Gebrauch des Programms 'Speicher 3':

Wie Sie sicherlich bemerkt haben, wird hier als Zahlsystem nicht das uns vertraute Dezimalsystem (Zahlen aus 0 bis 9 zusammengesetzt) sondern das Hexadezimalsystem (Zahlen aus 0 bis F zusammengestzt) angewandt. Dies hat folgende Gründe: Einerseits ist das Hexadezimalsystem 'das' System der Hacker und Maschinensprachfreaks ... so wollen wir es hier als 'Speicherdurchforster', was gleichbedeutend mit 'Hacker' sein soll, auch anwenden! Andererseits läßt sich auf 40 Spalten mit dem Dezimalsystem nicht so viel Information unterbringen, wie mit Hilfe des Hexadezimalsystems ('255' Dez. = 'FF' Hex. bzw. '65535' Dez. = 'FFFF' Hex.).

Dezimalumwandlung in Hex läßt sich mit Ihrem MSX-Computer sehr leicht mit dem Befehl 'HEX$(' in die Tat umsetzen z.B. 'PRINT HEX$(65535)', Ergebnis = 'FFFF'.

```
Anfangsadresse eingeben ? 32906
808A 4D 53 58 20 50 72 6F 67 MSX Prog
Anfangsadresse eingeben ?
8092 72 61 6D 6D 73 61 6D 6D rammsamm
Anfangsadresse eingeben ?
809A 6C 75 6E 67 00 EA 80 28 lung...(
80A2 00 8F 20 43 6F 70 79 72 .. Copyr
80AA 69 67 68 74 20 31 39 38 ight 198
80B2 34 20 44 41 54 41 20 42 4 DATA B
80BA 45 43 4B 45 52 20 26 20 ECKER &
80C2 20 20 20 20 20 20 20 20
80CA 20 20 20 20 20 20 20 20
80D2 20 20 20 20 20 20 20 20
80DA 20 20 52 61 69 6E 65 72   Rainer
80E2 20 4C 75 65 65 72 73 00  Lueers.
80EA 00 81 32 00 C5 20 11 3A ..2.. .:
80F2 BD 20 0F 0F 2C 12 2C 0F . ..,.,.
80FA 0F 3A CC 20 EB 00 09 81 .:. ....
8102 3C 00 A0 20 0F 28 00 2B <.. .(.+
810A 81 46 00 8F 20 44 61 72 .F.. Dar
8112 73 74 65 6C 6C 75 6E 67 stellung
811A 73 72 61 75 6D 20 65 69 sraum ei
Anfangsadresse eingeben ?
```

```
10 REM Untersuchung des Speichers 3
20 REM Untersuchung des Speichers mit
       PEEK und CHR$( in uebersichtli-
       cher Form
30 REM MSX Programmsammlung
40 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
50 SCREEN 0:COLOR 15,1,15:KEY OFF
60 WIDTH 40
70 REM Darstellungsraum eingrenzen
80 INPUT "Anfangsadresse eingeben ";A
90 IF A<0 OR A>65535! THEN GOSUB 320:GOT
O 60
100 A$=STRING$(4-LEN(HEX$(A)),"0")+HEX$(
A)
110 REM Die naechsten Zeilen helfen,
        einen optimalen Bildschirmaus-
        druck mit 40 Zeichen/Zeile zu
        ermoeglichen
120 PRINT A$;" ";
130 REM Die Bildschirmzeile wird mit
        jeweils acht Bytes gefuellt
140 FOR Z=0 TO 7
150 ZZ=PEEK(A+Z):ZZ$=STRING$(2-LEN(HEX$(
ZZ)),"0")+HEX$(ZZ)
160 PRINT ZZ$;" ";
170 NEXT Z
180 REM Ausdruck der Charakterstrings
        bei besonderer Beachtung, wenn
        der PEEKwert <32 bzw. >126 ist
190 FOR Z=0 TO 7
200 ZZ=A+Z
210 ZZ=PEEK(ZZ)
220 IF ZZ<32 THEN ZZ=46
230 IF ZZ>126 THEN ZZ=46
240 PRINT CHR$(ZZ);
250 NEXT Z
260 PRINT
270 A=A+8
280 A$=INKEY$
```

```
290 IF A$="" THEN GOTO 80
300 GOTO 100
310 END
320 PRINT:PRINT TAB(10) CHR$(200) "Falsc
he Eingabe!" CHR$(200)
330 GOSUB 340:RETURN
340 PRINT:PRINT TAB(6) "<Bitte eine Tast
e druecken>"
350 F$=INKEY$:IF F$="" THEN GOTO 350
360 RETURN
```

Speicher 4

Bisher haben wir den Speicher durchsucht und dabei zum Teil nur Zahlen (Speicher 1), wirre Zeichen (Speicher 2) oder aber gar schließlich beides in geordneter Form auf dem Bildschirm erzeugt (Speicher 3). Beim Listing von 'Speicher 3' konnte im Vergleich der linken zur rechten Bildschirmseite deutlich werden, daß wahrlich unser BASIC-Programm nur aus Zahlen besteht, die erst im Nachhinein wieder von Ihrem MSX-Computer in Buchstaben und Zeichen umgewandelt wurden.

Hierzu muß man wissen, wie der Computer Daten und Programme abspeichert: Um Speicherplatz zu sparen, werden BASIC-Befehle nach der Eingabe durch (ENTER) in Abkürzungen umgewandelt, die sogenannten Token (siehe Anhang). So verbraucht der Befehl 'PRINT' nicht fünf Bytes (da fünf Buchstaben), sondern lediglich ein Byte.

Schreiben wir jedoch in eine REM-Zeile 'PRINT' oder geben in eine Programmzeile 'PRINT "PRINT"' ein, so wird das in diesem Fall nicht als Befehl zu interpretierende 'PRINT' Buchstabe für Buchstabe abgespeichert, d.h. hierzu werden wie zu erwarten fünf Speicherplätze = fünf Bytes in unserem MSX-Computer benötigt.

Suchen wir nach dem Vorhandensein irgendwelcher ausgeschriebenen Wörter im BASIC-Programm, so kann dies sehr lange dauern, bei 65535 Speicherplätzen ähnlich einer Suche nach der Stecknadel im Heuhaufen. Also müssen wir systematischer vorgehen!

Das Programm 'Speicher 4' hilft uns dabei erheblich in mehreren Stufen weiter. Sie geben ein Wort mit bis zu sechs Buchstaben ein und instruieren den Computer, wo er nach diesem Wort im Speicher suchen soll. Soll er im ganzen Speicher suchen, dauert es - bald - eine Ewigkeit. Soll er in einem begrenzten von Ihnen zu bestimmenden Speicherplatzrahmen suchen, dauert es dementsprechend lang oder kurz. Als Ergebnis bekommen Sie in jedem dieser beiden Fälle, vorausgesetzt der Suchbegriff läßt sich auffinden, die Speicherplatzstelle genannt. Spaßeshalber können

Sie dann durch 'POKE' die angegebene Stelle, (kleiner als 127, größer als 31) in Ihrem Programm direkt über die entsprechende Speicherstelle ändern. Ein Beispiel:

In Zeile 80 steht das Wörtchen 'Laenge'. Zuerst suchen wir uns die entsprechende Speicherstelle heraus. Wie? Programm starten mit 'RUN', Zeichen eingeben = 'Laenge' (ENTER), im ganzen Speicher suchen lassen = 'N' auf die nächste Frage hin eintippen (ENTER). Lange müssen wir warten, aber schließlich erscheint auf dem Bildschirm: 'Laenge an Speicherstelle 33054 gefunden'. Kommt bei Ihnen eine etwas andere Zahl heraus, so ist das gar nicht schlimm. Wir nennen die gefundene Zahl 'ZAHL' (in unserem Fall 'ZAHL' = 33054). Schauen wir mal nach, was bei 'ZAHL' im Speicher steht: 'PRINT PEEK(ZAHL)' (Bitte nicht 'ZAHL' eintippen, sondern die Zahl an dieser Stelle verwenden, die Ihr MSX-Computer vorhin ausgegeben hat - bei mir also: 'PRINT PEEK(33054)'). Ergebnis? '76'! Und was bedeutet in unserer Sprache die Zahl 76? 'PRINT CHR$(76)' = 'L'!

Also die Zahl 76 steht für den ersten Buchstaben unseres vorhin gesuchten Wortes 'Laenge'. Listen wir Zeile 80, so steht dort 'Laenge'. Mit 'POKE ZAHL,andere Zahl als 76' werden wir nun das 'L' von Laenge in unserem BASIC-Programm z.B. in ein 'G' verwandeln. Nur welche Zahl steht beim Computer für den Buchstaben 'G'? Ganz einfach rauszukriegen über den Befehl 'ASC('. Also: 'PRINT ASC("G")' = '71'. Nun einfach eingeben 'POKE ZAHL,71' und anschließend noch einmal Zeile 80 listen -) 'Laenge' wurde zu 'Gaenge'!

So kann man ein klein wenig mehr über die Abspeicherung von BASIC-Programmen dazulernen und vielleicht dadurch noch eingehender die Komplexität eines Computers wie Ihres MSX-Computers erahnen lernen (BASIC ist wohlgemerkt erst das Endprodukt zahlreicher Verknüpfungen).

Aber noch eine weitere Dimension wird mit dem Programm 'Speicher 4' erschlossen: Die Abspeicherung der Zeilennummer. Starten Sie hierfür noch einmal das Programm mit 'RUN', geben wieder den Suchbegriff 'Laenge' (bzw. nun 'Gaenge') ein und lassen im

'P=Programm' danach suchen. Nach kurzer Zeit steht nicht nur die entsprechende Speicherstelle wieder auf dem Bildschirm, nein auch die entsprechende Zeilennummer in unserem BASIC-Programm.

Dieses Errechnen geht verhältnismäßig einfach vor sich ... wenn man weiß wie. Jedes MSX-BASIC-Programm beginnt im Normalfall bei der Speicherstelle 32769. In Speicherstelle 32769 und 32770 steht die Speicherstelle des Beginns der nächsten Zeile unseres Programms ('PRINT PEEK(32769)' ='36; 'PRINT PEEK(32770)' ='128). Die Errechnung geht folgendermaßen vor sich: 'PRINT PEEK(32769) + (PEEK(32770)*256)' ='32804. Also: die zweite Prorammzeile beginnt bei 32804 (daraus kann man auch die Laenge der aktuellen Zeile errechnen, indem man 32769 von 32804 subtrahiert). In Speicherstelle 32771 und 32772 ist die Zeilennummer der ersten Programmzeile gespeichert: 'PRINT PEEK(32771)' = '10; 'PRINT PEEK(32772)' = '0. Errechnung s.o.: 'PRINT PEEK(32771) + (PEEK(32772)*256)' ='10. Also: die erste Zeile unseres Programms 'Speicher 4' ist Zeile Nummer 10 und die nächste Zeile beginnt bei Speicherstelle 32804.

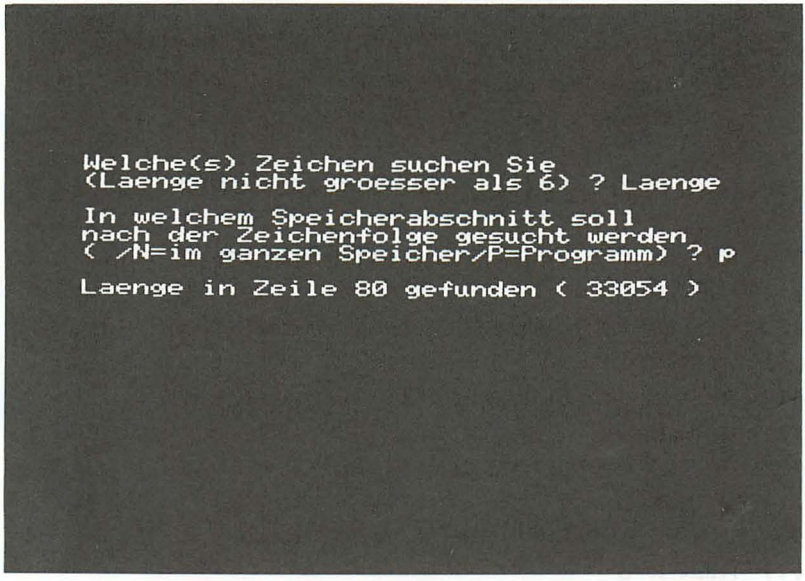
Genau diese Berechnungen führt Ihr MSX-Computer immer und immer wieder (ab Zeile 170) aus, um uns beim Auffinden des Suchbegriffs auch die Zeilennummer anzeigen zu können.

Nun aber genug der grauen Theorie! Gehen wir zu 'Speicher 5' über.

```
10 REM Untersuchung des Speichers 4
20 REM Zeichenfolge im Speicher suchen
30 REM MSX Programmsammlung
40 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
50 SCREEN 0:COLOR 15,1,15:KEY OFF
60 WIDTH 40
70 REM Eingabe des Suchbegriffs
80 INPUT "Welche(s) Zeichen suchen Sie
        (Laenge nicht groesser als 6)
";F$
90 IF LEN(F$)>6 OR LEN(F$)=0 THEN GOSUB
540:GOTO 60
100 PRINT
110 REM Speicherraum fuer Suche waehlen
120 INPUT "In welchem Speicherabschnitt
soll       nach der Zeichenfolge gesucht
 werden    ( /N=im ganzen Speicher/P=Pro
gramm) ";F1$
130 IF LEFT$(F1$,1)="p" OR LEFT$(F1$,1)=
"P" THEN F1$="P" ELSE IF LEFT$(F1$,1)="n
" OR LEFT$(F1$,1)="N" THEN F1$="N"
140 PRINT
150 IF F1$="N" THEN A=0:E=65535!:GOTO 37
0
160 IF F1$="P" THEN A=32769!:GOTO 180 EL
SE GOTO 300
170 REM Suche nach eingegebenem Such-
        begriff nur im Programmrahmen;
        wenn gefunden -> Anzeige der
        entsprechenden Zeilennummer
180 AA=PEEK(A)+(PEEK(A+1)*256):BB=PEEK(A
+2)+(PEEK(A+3)*256)
190 FOR N=A TO AA-1
200 IF CHR$(PEEK(N))=MID$(F$,1,1) THEN K
=1
210 IF K=1 AND CHR$(PEEK(N+1))=MID$(F$,2
,1) THEN K=2
220 IF K=2 AND CHR$(PEEK(N+2))=MID$(F$,3
,1) THEN K=3
```

```
230 IF K=3 AND CHR$(PEEK(N+3))=MID$(F$,4
,1) THEN K=4
240 IF K=4 AND CHR$(PEEK(N+4))=MID$(F$,5
,1) THEN K=5
250 IF K=5 AND CHR$(PEEK(N+5))=MID$(F$,6
,1) THEN K=6
260 IF K=LEN(F$) THEN PRINT F$;" in Zeil
e";BB;"gefunden (";N;")"
270 K=0
280 NEXT N
290 A=AA:IF AA=0 OR BB=0 THEN GOTO 470 E
LSE GOTO 180
300 PRINT
310 REM Darstellungsraum eingrenzen
320 INPUT "Anfangsadresse ";A
330 INPUT "Endadresse ";E
340 IF A>E OR A<0 OR E<0 OR A>65535! OR
E>65535! THEN GOSUB 540:GOTO 320
350 PRINT
360 REM Suche im Speicher von Adresse
        a bis e nach dem Suchbegriff f$
370 FOR Z=A TO E
380 IF CHR$(PEEK(Z))=MID$(F$,1,1) THEN K
=1
390 IF K=1 AND CHR$(PEEK(Z+1))=MID$(F$,2
,1) THEN K=2
400 IF K=2 AND CHR$(PEEK(Z+2))=MID$(F$,3
,1) THEN K=3
410 IF K=3 AND CHR$(PEEK(Z+3))=MID$(F$,4
,1) THEN K=4
420 IF K=4 AND CHR$(PEEK(Z+4))=MID$(F$,5
,1) THEN K=5
430 IF K=5 AND CHR$(PEEK(Z+5))=MID$(F$,6
,1) THEN K=6
440 IF K=LEN(F$) THEN PRINT F$;" an Spei
cherstelle";Z;"gefunden"
450 K=0
460 NEXT Z
470 PRINT
480 PRINT
```

```
490 REM Programmende oder Fortfahren
500 INPUT "Suchen Sie weitere Zeichen (
/N) ";F$
510 IF LEFT$(F$,1)="n" OR LEFT$(F$,1)="N
" THEN F$="N"
520 IF F$<>"N" THEN GOTO 60
530 END
540 PRINT:PRINT TAB(10) CHR$(200) "Falsc
he Eingabe!" CHR$(200)
550 GOSUB 560:RETURN
560 PRINT:PRINT TAB(6) "<Bitte eine Tast
e druecken>"
570 F$=INKEY$:IF F$="" THEN GOTO 570
580 PRINT:RETURN
```



Speicher 5

Während wir im Programm 'Speicher 3' jeweils in einer Zeile acht Bytes dargestellt haben (hexadezimal und als CHR$-Wert), wird in diesem Programm alles übersichtlicher = ein Byte pro Zeile und zugleich vielseitiger dargestellt.

Die Darstellungsformen dezimal und hexadezimal laufen jeweils nebeneinander:

1) Adresse dezimal
2) Adresse hexadezimal
3) Inhalt der Adresse dezimal
4) Inhalt der Adresse hexadezimal
5) Inhalt der Adresse (wenn darstellbar) als 'CHR$('

Zu Ihrer eigenen Speicherdurchforstung ist dieses Programm recht sinnvoll, denn Sie brauchen nicht immer alles umzurechnen; außerdem bietet dieses Programm eine ideale Möglichkeit, die Vielzahl der Token (siehe Anhang) kennenzulernen (erinnern Sie sich noch? Token = BASIC-Wort).

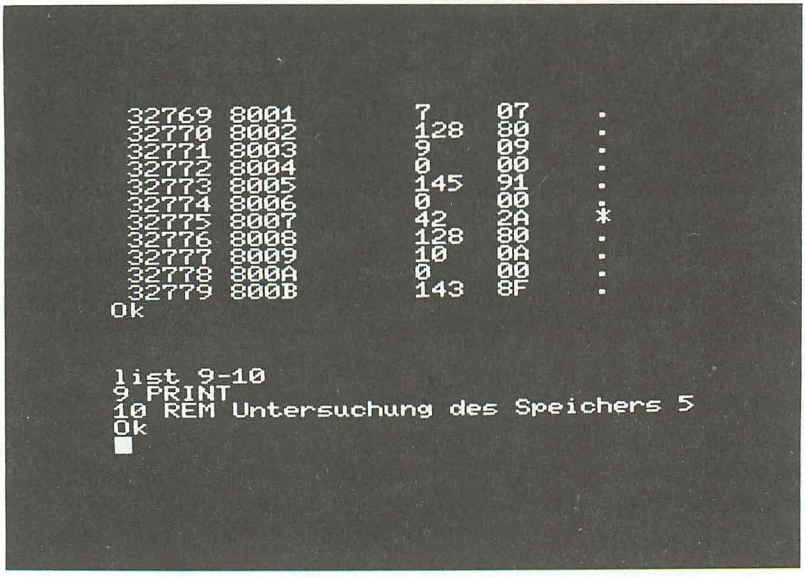
Machen wir einen Versuch: Ergänzen Sie das Programm um Zeile 9: '9 PRINT' (ENTER). Wie Sie wissen, beginnt der BASIC-Programmspeicher bei Adresse 32769. Geben wir also als Anfangsadresse '32769' ein; der Endwert soll uns egal sein, soll aber in jedem Fall größer als die Anfangsadresse sein (drücken wir nur (ENTER), so läuft die Speicherabbildung bis Anfangsadresse+2000 = 34769). Es reicht für unser Experiment, wenn wir uns nur die ersten 10 Speicherplätze anzeigen lassen, also Endadresse 32779! Listen Sie bitte zusätzlich zur Kontrolle Zeile 9 und Zeile 10.

Kurz zur Deutung des Speicherabbildes:

```
Adresse 32769:Inhalt 7   (32769 und 32770 geben die Adresse
Adresse 32770:Inhalt 128  der nächsten Zeile an:
                          Adr.32769+(256*Adr.32770)=Adr.32775)
```

```
Adresse 32771:Inhalt 9   (32771 und 32772 geben die
                          Zeilennr. an:
Adresse 32772:Inhalt 0    Adr.32771+(256*Adr.32772)=Zeilennr.9)
Adresse 32773:Inhalt 145 (Tokenwert für den Befehl 'PRINT')
Adresse 32774:Inhalt 0   (Zeilenende-Anzeige)
Adresse 32775:Inhalt 42  (32775 und 32776 geben die Adresse
Adresse 32776:Inhalt 128  der nächsten Zeile an:
                          Adr.32775+(256*Adr.32776)=Adr.32810)
Adresse 32777:Inhalt 10  (32777 und 32778 geben die
                          Zeilennummer an:
Adresse 32778:Inhalt 0    Adr.32777+(256*Adr.32778)=Zeilennr.10)
Adresse 32779:Inhalt 143 (Tokenwert für den Befehl 'REM')
```

Interessieren soll uns in diesem Zusammenhang nur Adresse 32773. Geben Sie bitte einmal 'POKE 32773,143' ('143' = Tokenzahl für 'REM') ein und listen anschließend Zeile 9. Nun steht dort nicht mehr wie ursprünglich 'PRINT', dafür aber viel besser 'REM'. Es gibt nicht nur Einbyte-('PRINT' = 145, 'REM' = 143)Token, es gibt auch Zweibytetoken. Aber davon hier nicht mehr. Schauen Sie zur Orientierung bitte im Anhang nach!



```
10 REM Untersuchung des Speichers 5
20 REM Speicherausgabe Byte fuer Byte
      pro Zeile
30 REM MSX Programmsammlung
40 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
50 SCREEN 0:COLOR 15,1,15:KEY OFF
60 WIDTH 40
70 REM Darstellungsrahmen eingrenzen
80 INPUT "Anfangsadresse ";A
90 PRINT "Endadresse (Zahl/<ENTER>->";A+
2000;")":INPUT E
100 IF A<0 OR E<0 OR A>65535! OR E>65535
! THEN GOSUB 260:GOTO 60
110 IF E=0 THEN E=A+2000
120 CLS
130 REM Reihenfolge im Ausdruck:
        Adresse dezimal
        Adresse hexadezimal
        Inhalt der Adresse dezimal
        Inhalt der Adresse hexadezimal
        Inhalt der Adresse als CHR$(
140 FOR Z=A TO E
150 PRINT Z;
160 PRINT TAB(7) STRING$(4-LEN(HEX$(Z)),
"0")+HEX$(Z);
170 PRINT TAB(17) PEEK(Z);
180 PRINT TAB(23) STRING$(2-LEN(HEX$(PEE
K(Z))),"0")+HEX$(PEEK(Z));
190 PRINT TAB(29);
200 N1=PEEK(Z)
210 IF N1<32 THEN N1=46
220 IF N1>126 THEN N1=46
230 PRINT CHR$(N1)
240 NEXT Z
250 END
260 PRINT:PRINT TAB(10) CHR$(200) "Falsc
he Eingabe!" CHR$(200)
270 GOSUB 280:RETURN
```

```
280 PRINT:PRINT TAB(6) "<Bitte eine Tast
e druecken>"
290 F$=INKEY$:IF F$="" THEN GOTO 290
300 RETURN
```

Grafikeditor

Beim MSX-BASIC wird uns eine Sound- sowie eine Grafikprogrammiersprache kostenlos dazugeliefert. Während wir zur Sounderstellung neben der Programmiersprache mit 'PLAY' (z.B. 'PLAY "CDE"') lediglich auf einen Befehl, nämlich die Anweisung 'SOUND' zurückgreifen können, sieht dies bei der Grafikprogrammierung sehr viel besser aus: Neben der Grafikprogrammiersprache mit 'DRAW' (z.B. 'DRAW "BM 128,96 R8 U8 R8"'=Zeichnen einer Treppe in der Mitte des Bildschirms) können wir auf eine Vielzahl von Grafikprogrammierbefehlen zurückgreifen, die uns das Zeichnen auf dem Bildschirm erheblich vereinfachen: Wir zeichnen Kreise, Kreisausschnitte, ja sogar Ellipsen mit 'CIRCLE', Linien und Rechtecke mit 'LINE' und können schließlich geschlossene Flächen mit 'PAINT' ausmalen. Sprites lassen sich mit 'SPRITE$' und 'PUT SPRITE' sehr einfach erstellen usw.

All' unsere Grafik wird mit einer Auflösung von 256*192 Punkten in bis zu sechzehn Farben auf dem Bildschirm wiedergegeben, wenn auch mit einer gewissen Einschränkung, was die Farbvielfalt betrifft (s.u.).

Die Grafikprogrammierung ist verhältnismäßig einfach mit dieser großen Befehlsvielzahl durchführbar, aber um ein Bild zu zeichnen bedarf es dann doch erst einmal eines Blattes Papier - möglichst eines Millimeterpapiers -, damit wir uns später im 'Koordinatendschungel' auch zurechtfinden.

Anders, d.h. einfacher, gelingt dies mit dem hier abgedruckten 'Grafikeditor'. Über den Befehl 'ON KEY GOSUB' werden die zehn Funktionstasten abgefragt. Diese Tasten arbeiten zeitweise mit dem Zehnertastenfeld, den Cursorsteuertasten, manchmal sogar mit der gesamten Tastatur zusammen und zwar folgendermaßen:

F1 drücken: mit den Cursortasten den Radius festlegen, (ENTER). Der Kreis wird entsprechend auf den Bildschirm gezeichnet.

F2 drücken: mit den Cursortasten den Zielpunkt festegen, (ENTER). Eine Linie wird zwischen Ursprungs- und Zielpunkt gezeichnet.

F3 drücken: zwei Ziffern eingeben (zwischen '00' und '15'). Ab nun wird mit der eingegebenen Farbziffer gezeichnet (z.B. '15'=weiß, '01'=schwarz).

F4 drücken: Text an der eingegebenen Cursorposition ins Grafikbild einfügen. Ende durch (ENTER).

F5 drücken: wenn zuvor mit dem Cursor auch gezeichnet wurde, so wird nun nicht mehr gezeichnet; wurde zuvor mit dem Cursor nicht gezeichnet, so wird jetzt gezeichnet.

F6 drücken: es erscheint in der Mitte der obersten Bildschirmzeile ein schwarzes Feld, in das nun mit sechs Buchstaben der Titel=der Name des anschließend zu speichernden Grafikbildes eingegeben wird; (ENTER) und das Abspeichern beginnt (dauert auf Kassettenrecorder länger als 10 Minuten).

F7 drücken: es erscheint in der Mitte der obersten Bildschirmzeile ein schwarzes Feld, in das nun mit sechs Buchstaben der Titel=der Name des anschließend zu ladenden Grafikbildes eingegeben wird; (ENTER) und das Laden beginnt (dauert von Kassettenrecorder über 10 Minuten).

F8 drücken: zwei Ziffern eingeben (zwischen '00' und '15'). Ab nun hat der Bildschirmrand die eingegebene Farbnummer (z.B. '15'=weiß, '01'=schwarz).

F9 drücken: mit der zur Zeit aktuellen Farbe (siehe 'F3') wird von der Cursorposition ausgehend gemalt, bis die Farbe auf einen gleichfarbigen Rand trifft.

F10 drücken: der aktuelle Cursor ist nun die linke untere Ecke eines Bildschirmausschnitts; ein zweiter Cursor wird zur rechten oberen Ecke des Bildschirmausschnitts mit den Cursortasten hingelenkt (ENTER); ein dritter Cursor wird irgendwohin auf den Bildschirm gesetzt, der als Ecke links unten der Bildschirmkopie fungieren soll.

Die Eingaben müssen so erfolgen, wie hier vorgegeben. Andererseits zeigen die Funktionstasten nicht immer die Wirkung, wie dies eigentlich beabsichtigt ist.

Leider erfolgt während des Zeichnens keine Bildschirmausgabe, (bis auf die Bewegung der Cursorfadenkreuze), die Ihnen anzeigen würde, welche Taste Sie gerade gedrückt haben. Dies ist deshalb so, damit Ihr Grafikfeld nicht zerstört wird.

Der Grafikcursor besteht aus zwei Sprites: das Fadenkreuz ist schwarz, der Fadenkreuzmittelpunkt weiß. So kann es nie passieren, daß der Grafikcursor in der Zeichnung verschwindet und Sie nicht mehr wissen, wo Sie sich im Augenblick befinden. Wird ein zweiter Cursor gesetzt (wie z.B. zum Festlegen des Radius beim Kreiszeichnen), teilt sich der Grafikcursor in zwei Teile: Der weiße Mittelpunkt bleibt zurück, während das Fadenkreuz zur Ansteuerung des zweiten Cursors dient. So merkt sich Ihr MSX-Computer jederzeit, wo er eigentlich im Augenblick seine Position hat.

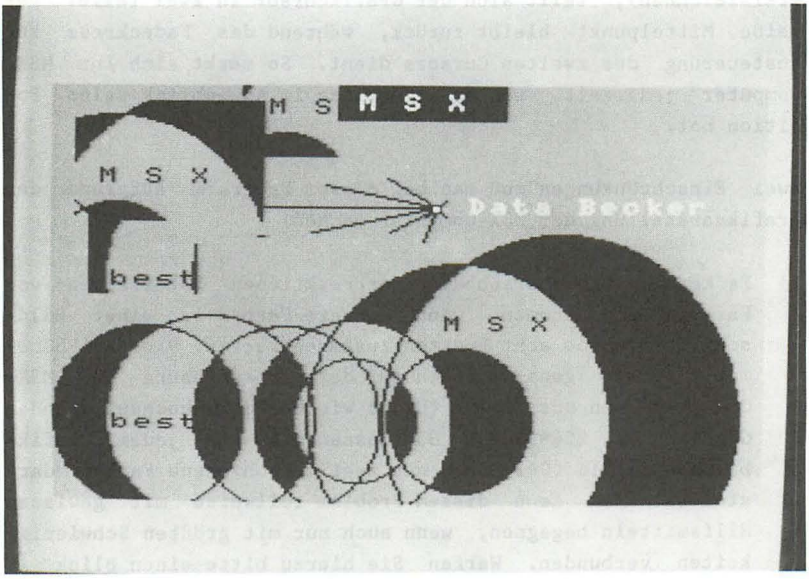
Zwei Einschränkungen muß man bei diesem Programm aufgrund der Grafikdarstellung der MSX-Computer machen:

1) Es kann zu unvorhersehbaren Farbreaktionen (Überlappen von Farbrändern) kommen, wenn mehrere Farben in einer Bildschirmzeile von acht Punkten zusammentreffen. Dieses 'Phänomen' beruht ganz einfach auf der Farbauflösung der MSX-Computer von nur 32*192 (nicht wie bei der hochauflösenden Grafik von 256*192!). So lassen sich in jedem Grafikbildschirmfeld (8*1) auch nur zwei verschiedene Farben darstellen. Man kann diesem Problem teilweise mit gewissen Hilfsmitteln begegnen, wenn auch nur mit größten Schwierigkeiten verbunden. Werfen Sie hierzu bitte einen Blick in

mein zweites MSX-Buch, das bei DATA BECKER erschienen ist: 'MSX Grafik und Sound'.

2) Alle MSX-Computer zeichnen bei normaler Anwendung des 'CIRCLE'-Befehls keinen Kreis, sondern eine Ellipse auf unsere PAL-Fernsehapparate. Um diesem MSX-Grafikproblem sinnvoll zu begegnen, habe ich den 'Kreisverformungsparameter' mit '4/3' belegt (z.B. 'CIRCLE (128,96),20,,,,4/3'); so haben wir nun eigentlich eine Ellipse, auf dem PAL-Fernsehbildschirm erscheint aber ein 'richtiger' Kreis. Aus diesem Grund kann es ab und zu vorkommen, daß der beim 'CIRCLE'-Befehl gebrauchte zweite Cursor (Radiusbestimmung) nun nicht mehr ganz da sitzt, wo Sie ihn hingesteuert haben. Gewinnen Sie Erfahrung und berechnen Sie den entsprechenden Korrekturfaktor beim Zeichnen mit ein.

Sonst aber ist Ihnen dieses Programm sicherlich eine große Hilfe. Haben Sie Vorstellungen, wie man es noch erweitern kann? Probieren Sie es selbst aus oder schauen Sie in 'MSX Sound und Grafik' von DATA BECKER nach!



```
10 REM Grafikeditor
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 REM Damit man nicht mit einer un-
       vorhersehbaren Farbkombination
       auf den Textbildschirm
       zurueckkehrt
50 ON STOP GOSUB 1970
60 STOP ON
70 REM F1 Kreis zeichnen, wenn der
          zweite Cursor gesetzt ist
       F2 Linie zeichnen, wenn der
          zweite Cursor gesetzt ist
       F3 Zeichenfarbe aendern
          Zweistellige Zahl eingeben
80 REM F4 Text an der Cursorposition
          schreiben; <ENTER> heisst
          Ende
90 REM F5 Zeichnen oder nicht zeichnen
          nicht zeichnen=Cursorbewegung
       F6 Abspeichern mit Name
          und Ueberschrift
          (dauert ca. 10 Minuten)
       F7 Einladen von Bildschirmen
100 REM    Namenseingabe
        F8 Umrandungsfarbe aendern
           immer mit zwei Ziffern
           eingeben (z.B. '08' aber
           ohne <ENTER>
110 REM F9 Ausmalen mit der
           aktuellen Farbe
           Vorsicht! Es wird bis zur
           gleichen Randfarbe aus-
           gemalt
120 REM F10 Kopieren von Bildschirm-
            teilen. Aktueller Cursor
            ist die Ecke unten links
            2.Cursor=oben rechts
            3.Cursor=neues Bild
              Ecke unten links
```

```
130 ON KEY GOSUB 530,660,790,830,910,940
,1170,1390,1420,1450
140 REM Vorbedingungen treffen
150 COLOR 15,4,5
160 SCREEN 2,2
170 FOR N=1 TO 10
180 KEY(N) ON
190 NEXT N
200 REM Einlesen des Fadenkreuzes,
        das als Grafikcursor=Sprite
        dient
210 FOR N=1 TO 8
220 READ A$
230 A$="&b"+A$
240 B$=B$+CHR$(VAL(A$))
250 NEXT  N
260 SPRITE$(1)=B$
270 REM Einlesen des Fadenkreuz-
        mittelpunkts, damit man auf
        jedwedem Untergrund den
        Cursor auch sieht
280 FOR N=1 TO 8
290 READ A$
300 A$="&b"+A$
310 C$=C$+CHR$(VAL(A$))
320 NEXT  N
330 SPRITE$(2)=C$
340 REM Setzen des Grafikcursors
        in den Bildschirmmittelpunkt
        Farbe schwarz
350 A=128
360 B=96
370 F=1
380 COLOR F
390 PUT SPRITE 1,(A,B),1
400 PUT SPRITE 2,(A,B),15
410 REM
420 IF ME=1 THEN PSET(A+4,B+4),F
```

```
430 REM Abfrage der Richtung ueber
        die Betaetigung der Pfeil-
        tasten
440 A$=INKEY$
450 IF A$="" THEN GOTO 440
460 IF ASC(A$)=28 THEN IF A<>250 THEN A=
A+1
470 IF ASC(A$)=29 THEN IF A<>-4 THEN A=A
-1
480 IF ASC(A$)=30 THEN IF B<>-4 THEN B=B
-1
490 IF ASC(A$)=31 THEN IF B<>186 THEN B=
B+1
500 PUT SPRITE 1,(A,B),1
510 PUT SPRITE 2,(A,B),15
520 GOTO 390
530 REM Kreis zeichnen
        ==============
540 C=A
550 D=B
560 REM Abfrage der Richtung ueber
        die Betaetigung der Pfeil-
        tasten
570 A$=INKEY$
580 IF A$="" THEN GOTO 570
590 IF ASC(A$)=28 THEN IF C<>250 THEN C=
C+1
600 IF ASC(A$)=29 THEN IF C<>-4 THEN C=C
-1
610 IF ASC(A$)=30 THEN IF D<>-4 THEN D=D
-1
620 IF ASC(A$)=31 THEN IF D<>186 THEN D=
D+1
630 IF (A$)=CHR$(13) THEN A1=ABS(A-C):A2
=ABS(B-D):IF A1>=A2 THEN CIRCLE (A+4,B+4
),A1+13,F,,,4/3:RETURN ELSE CIRCLE (A+4,
B+4),A2,F,,,4/3:RETURN
640 PUT SPRITE 1,(C,D),1
650 GOTO 570
```

```
660 REM Linie zeichnen
      ==============
670 C=A
680 D=B
690 REM Abfrage der Richtung ueber
      die Betaetigung der Pfeil-
      tasten
700 A$=INKEY$
710 IF A$="" THEN GOTO 700
720 IF ASC(A$)=28 THEN IF C<>250 THEN C=
C+1
730 IF ASC(A$)=29 THEN IF C<>-4 THEN C=C
-1
740 IF ASC(A$)=30 THEN IF D<>-4 THEN D=D
-1
750 IF ASC(A$)=31 THEN IF D<>186 THEN D=
D+1
760 IF (A$)=CHR$(13) THEN LINE (A+4,B+4)
-(C+4,D+4),F:RETURN
770 PUT SPRITE 1,(C,D),1
780 GOTO 700
790 REM Farbe aendern
      ==============
800 F$=INPUT$(2)
810 F=VAL(F$)
820 IF F<16 THEN COLOR F:RETURN ELSE RET
URN
830 REM Text eingeben
      ==============
840 OPEN "grp:"FOR OUTPUT AS #1
850 DRAW "bm =a;,=b;"
860 A$=INKEY$
870 IF A$="" THEN GOTO 860
880 IF A$=CHR$(13) THEN CLOSE:RETURN
890 PRINT #1,A$;
900 GOTO 860
910 REM Zeichnen - nicht zeichnen
      ==========================
920 IF ME=1 THEN ME=0 ELSE ME=1
930 RETURN
```

```
940 REM Abspeichern
      ===========
950 REM Damit es nicht zu unvorher-
      gesehenen Farbvermischungen
      kommen kann
960 LINE (100,15)-(160,0),1,BF
970 OPEN "grp:" FOR OUTPUT AS #1
980 NN$=""
990 COLOR 15
1000 DRAW "bm 108,4"
1010 REM Eingabe des Programmnamens=
       der Bildueberschrift mit bis
       zu sechs Buchstaben/Zeichen
1020 FOR N=1 TO 6
1030 NN$=NN$+N$
1040 N$=INPUT$(1)
1050 IF N$=CHR$(13) THEN GOTO 1060 ELSE
PRINT #1,N$;:NEXT N
1060 A$=INKEY$
1070 REM Sicherheitsfunktion
1080 IF A$="" THEN GOTO 1060 ELSE IF A$<
>CHR$(13) THEN RETURN
1090 CLOSE #1
1100 OPEN NN$ FOR OUTPUT AS #1
1110 REM Abspeicherung der 16K
       Video-RAM
1120 FOR N=0 TO 16383
1130 PRINT #1,VPEEK(N)
1140 NEXT N
1150 CLOSE
1160 RETURN
1170 REM Laden
       =====
1180 REM Damit es nicht zu unvorher-
       gesehenen Farbvermischungen
       kommen kann
1190 LINE (100,15)-(160,0),1,BF
1200 OPEN "grp:" FOR OUTPUT AS #1
1210 NN$=""
1220 COLOR 15
```

```
1230 DRAW "bm 108,4"
1240 REM Eingabe des Programmnamens=
        der Bildueberschrift mit bis
        zu sechs Buchstaben/Zeichen
1250 FOR N=1 TO 6
1260 NN$=NN$+N$
1270 N$=INPUT$(1)
1280 IF N$=CHR$(13) THEN GOTO 1060 ELSE
PRINT #1,N$;:NEXT N
1290 REM Sicherheitsfunktion
1300 A$=INKEY$
1310 IF A$="" THEN GOTO 1300 ELSE IF A$<
>CHR$(13) THEN RETURN
1320 CLOSE #1
1330 REM Einladen der 16K
        Video-RAM
1340 OPEN NN$ FOR INPUT AS #1
1350 FOR N=0 TO 16383
1360 INPUT #1,D
1370 VPOKE N,D
1380 NEXT N
1390 REM Border aendern
        ===============
1400 A$=INPUT$(2)
1410 IF VAL(A$)<16 THEN COLOR ,,VAL(A$):
RETURN ELSE RETURN
1420 REM Ausmalen
        ========
1430 PAINT (A,B),F
1440 RETURN
1450 REM Verdoppeln
        ==========
1460 C1=A
1470 D1=B
1480 REM Festlegen des Ursprung-
        bildschirmausschnitts
        Ecke rechts oben
1490 REM Abfrage der Richtung ueber
        die Betaetigung der Pfeil-
        tasten
```

```
1500 A$=INKEY$
1510 IF A$="" THEN GOTO 1500
1520 IF ASC(A$)=28 THEN IF C1<>250 THEN
C1=C1+1
1530 IF ASC(A$)=29 THEN IF C1<>-4 THEN C
1=C1-1
1540 IF ASC(A$)=30 THEN IF D1<>-4 THEN D
1=D1-1
1550 IF ASC(A$)=31 THEN IF D1<>186 THEN
D1=D1+1
1560 IF (A$)=CHR$(13) THEN GOTO 1600
1570 PUT SPRITE 1,(C1,D1),1
1580 GOTO 1500
1590 REM Der Cursor muss im Verhaelt-
        nis zum Ursprung rechts oben
        postiert werden!
1600 IF A>C1 OR B<D1 THEN BEEP:RETURN EL
SE C2=A:D2=B
1610 REM Festlegen der Kopie
        Bildschirmausschnitt
        Ecke links unten
1620 REM Abfrage der Richtung ueber
        die Betaetigung der Pfeil-
        tasten
1630 A$=INKEY$
1640 IF A$="" THEN GOTO 1630
1650 IF ASC(A$)=28 THEN IF C2<>250 THEN
C2=C2+1
1660 IF ASC(A$)=29 THEN IF C2<>-4 THEN C
2=C2-1
1670 IF ASC(A$)=30 THEN IF D2<>-4 THEN D
2=D2-1
1680 IF ASC(A$)=31 THEN IF D2<>186 THEN
D2=D2+1
1690 IF (A$)=CHR$(13) THEN GOTO 1730
1700 PUT SPRITE 1,(C2,D2),1
1710 GOTO 1630
1720 REM Eigentlicher Kopiervorgang
        vom Ursprungsbild ausgehend
1730 FOR N=A TO C1
```

```
1740 FOR M=B TO D1 STEP-1
1750 PSET(C2+(N-A),D2+(M-B)),POINT(N,M)
1760 NEXT M,N
1770 RETURN
1780 REM Sprite 1
        Schwarzes Fadenkreuz
1790 DATA 10000001
1800 DATA 01000010
1810 DATA 00111100
1820 DATA 00100100
1830 DATA 00100100
1840 DATA 00111100
1850 DATA 01000010
1860 DATA 10000001
1870 REM Sprite 2
        Weisser Fadenkreuzmittel-
        punkt
1880 DATA 00000000
1890 DATA 00000000
1900 DATA 00000000
1910 DATA 00011000
1920 DATA 00011000
1930 DATA 00000000
1940 DATA 00000000
1950 DATA 00000000
1960 REM Bei Programmunterbrechung
        wird wieder ein lesbares Farb-
        verhaeltnis erzeugt
1970 COLOR 15,1
```

Soundeditor

Im Gegensatz zu fast sämtlichen 'tonangebenden' Computersystemen, gibt uns die Firma MICROSOFT mit ihrem/unserem MSX-BASIC einfachste Möglichkeiten an die Hand, Musik mit dem Computer zu erstellen.

Es ist mit Hilfe des Befehls 'PLAY' sogar nicht nur möglich, die uns wohlbekannten Notennamen eintippen und somit die entsprechenden Töne erklingen lassen zu können (z.B. 'PLAY "CDE"' (ENTER)), nein, wir können gar Dreiklänge hervorzaubern, indem wir drei Notennamen durch Kommata abtrennen und mit 'PLAY' spielen lassen (z.B. 'PLAY "C","E","G"' (ENTER)). Den gewünschten Ton eine Oktave höher erklingen zu lassen ... kein Problem durch das Hinzufügen des Parameters "O" (z.B. 'PLAY "O5 CDE"' (ENTER)). Wird hingegen versucht - was ohne weiteres möglich ist -, Synthesizerklänge mit 'PLAY' zu erzeugen, mag es ja noch durch Hinzufügen des Parameters 'S' leicht ansteuerbar sein (z.B. 'PLAY "S8 CDE"' (ENTER)); aber jetzt auch noch die Feinheiten mit einzufügen, da hört die Lust auf und es fängt die Arbeit an: Immer wieder muß der Klang neu aktiviert werden, eine Zahl wird eingefügt, eine andere gelöscht ... und schließlich verliert man die Lust daran, weil der Überblick verloren gegangen ist.

Um all' die Musikuntermalung für unsere Programme platzsparend unterzubringen und gar die erstellten Strings noch zu mischen, ist die Anweisung 'PLAY' als Programmierbefehl ideal.

Für die Darstellung und Erarbeitung komplizierterer Klangmuster bedient man sich aber doch viel eher und viel besser des 'SOUND'-Befehls. Zwar heißt es hier nun, ins abstrakte Musikzahlenmeer einzutauchen, aber das bringt auch erhebliche Vorteile mit sich:

1) Bis auf die Zuweisung von Hüllkurvenparametern gilt: erklingt ein mit 'SOUND' erstellter Ton, klingt er so lange weiter, bis wir ihn abbrechen oder einen Abbruch durch eine Fehlermeldung hervorrufen

2) Da wir nur einmal das Grundtonmuster erstellen, können wir uns im Nachhinein nur auf wenige zusätzlich einzugebende Hüllkurvenparameter konzentrieren, es muß nicht immer wieder von vorn angefangen werden

Da sich ziemlich viele von Ihnen vor dem Zahlenmeer des 'SOUND'-Befehls sträuben, wird Ihnen der vorliegende 'Soundeditor' nun eine große Hilfe sein. Sie halten hier nicht nur übersichtlich Ausschau auf alle möglichen Tonveränderungskriterien (sechzehn an der Zahl), Sie arbeiten zudem nicht mehr mit Zahlentasten, sondern nur noch mit dem eingebauten Pfeileditor und zwar folgendermßen:

Pfeil nach unten: in den nächsten 'SOUND'parameter überspringen
Pfeil nach oben: in den vorherigen 'SOUND'parameter wechseln
Pfeil nach links: Parameterziffer in der zugehörigen Zeile um jeweils '1' verringern
Pfeil nach rechts: Parameterziffer in der zugehörigen Zeile um jeweils '1' erhöhen

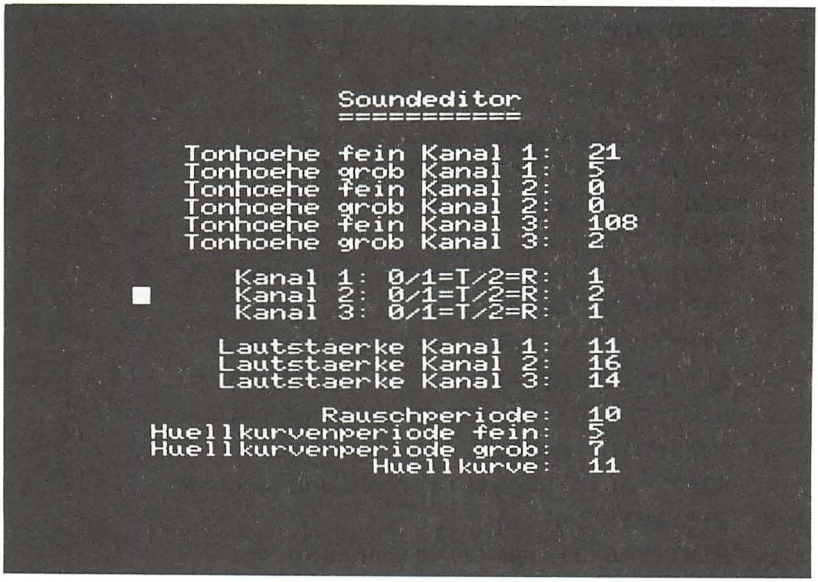
Dabei führt Sie der Editor nicht nur übersichtich und somit anwenderfreundlich zu jedem Tonparameter hin, er achtet auch darauf, daß keine falschen=ungültigen Zahlwerte angesteuert werden. Nicht zuletzt lernen Sie beim spielerisch einfachen Umgang mit dem 'Soundeditor' auch noch ein klein wenig besser die vielfältigen Register des Soundprozessors kennen und schätzen.

Sind Ihnen gute Klänge in drei Tonkanälen geglückt, notieren Sie sich die entsprechenden Zahlwerte und geben dieselbigen anschließend mit der gewünschten Variationsbreite (z.B. durch den Gebrauch der Befehle 'FOR', 'NEXT' und 'STEP') in Ihr Programm ein. In einem Programm meines DATA BECKER-Buchs 'MSX Grafik und Sound' wird Ihnen auch noch diese Aufgabe abgenommen sowie vieles andere mehr ermöglicht und auch ausführlich erläutert. Versuchen Sie es ähnlich, die Möglichkeiten sind 'fast' unbegrenzt!

Noch ein Tip zum ersten Gebrauch des 'Soundeditors': Nicht wie bei 'PLAY' reicht lediglich die Eingabe der Tonhöhe (z.B. 'PLAY "C"'). Um einen Ton erklingen zu lassen, müssen wenigstens

1) die 'Tonhöhe' des Kanals auf mindestens '1'
2) der 'Kanal' auf 'Ton'='1' - nicht auf 'Rauschen' -
3) die 'Lautstärke' des Kanals auf mindestens '1'

eingestellt werden. Danach ist es Ihrem Ideenereichtum überlassen, nach Belieben zu experimentieren.



```
10 REM Soundeditor
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                  Rainer Lueers
40 REM Vorbedingungen treffen
50 SCREEN 0:WIDTH 35
60 CLEAR 2000:DIM A$(21),A(21)
70 COLOR 15,1,1
80 ON STOP GOSUB 1130:STOP ON
90 REM Alle Soundparameter in die
       Ursprungslage versetzen
100 FOR N=0 TO 13
110 SOUND N,0
120 NEXT N
130 SOUND 7,63
140 REM Bildschirmaufbau
150 PRINT TAB(12) "Soundeditor"
160 PRINT TAB(12) "============"
170 PRINT
180 PRINT TAB(3);:A$(3)="Tonhoehe fein K
anal 1:":PRINT A$(3)
190 PRINT TAB(3);:A$(4)="Tonhoehe grob K
anal 1:":PRINT A$(4)
200 PRINT TAB(3);:A$(5)="Tonhoehe fein K
anal 2:":PRINT A$(5)
210 PRINT TAB(3);:A$(6)="Tonhoehe grob K
anal 2:":PRINT A$(6)
220 PRINT TAB(3);:A$(7)="Tonhoehe fein K
anal 3:":PRINT A$(7)
230 PRINT TAB(3);:A$(8)="Tonhoehe grob K
anal 3:":PRINT A$(8)
240 PRINT
250 PRINT TAB(6);:A$(10)="Kanal 1: 0/1=T
/2=R:":PRINT A$(10)
260 PRINT TAB(6);:A$(11)="Kanal 2: 0/1=T
/2=R:":PRINT A$(11)
270 PRINT TAB(6);:A$(12)="Kanal 3: 0/1=T
/2=R:":PRINT A$(12)
280 PRINT
```

```
290 PRINT TAB(5);:A$(14)="Lautstaerke Ka
nal 1:":PRINT A$(14)
300 PRINT TAB(5);:A$(15)="Lautstaerke Ka
nal 2:":PRINT A$(15)
310 PRINT TAB(5);:A$(16)="Lautstaerke Ka
nal 3:":PRINT A$(16)
320 PRINT
330 PRINT TAB(11);:A$(18)="Rauschperiode
:":PRINT A$(18)
340 PRINT TAB(1);:A$(19)="Huellkurvenper
iode fein:":PRINT A$(19)
350 PRINT TAB(1);:A$(20)="Huellkurvenper
iode grob:":PRINT A$(20)
360 PRINT TAB(14);:A$(21)="Huellkurve:":
PRINT A$(21)
370 REM Wertzuweisung der Sound-
        parameter auf dem Bildschirm
380 FOR N=3 TO 8
390 LOCATE 26,N
400 PRINT A
410 NEXT N
420 FOR N=10 TO 12
430 LOCATE 26,N
440 PRINT A
450 NEXT N
460 FOR N=14 TO 16
470 LOCATE 26,N
480 PRINT A
490 NEXT N
500 FOR N=18 TO 21
510 LOCATE 26,N
520 PRINT A
530 NEXT N
540 REM Einbringen des Zeilencursors
        in die entsprechende Bild-
        schirmzeile in Position 0
550 LOCATE 0,3:PRINT CHR$(200);
```

```
560 REM Warten auf Eingabe; die Ein-
        gabemoeglichkeit bezieht sich
        nur auf das Cursorsteuerfeld
        d.h. Pfeil nach oben bzw.
             Pfeil nach unten ->
             Cursorpositionierung
570 REM      Pfeil nach links bzw.
             Pfeil nach rechts ->
             Soundparameter +/-
580 A$=INKEY$:IF A$="" THEN GOTO 580
590 IF A$=" " THEN SOUND 13,A(21):GOTO 5
80
600 IF ASC(A$)=30 OR ASC(A$)=31 THEN GOT
O 610 ELSE IF ASC(A$)=28 OR ASC(A$)=29 T
HEN GOTO 690 ELSE GOTO 580
610 IF ASC(A$)=30 AND CSRLIN=3 THEN GOTO
 580
620 IF ASC(A$)=31 AND CSRLIN=21 THEN GOT
O 580
630 IF ASC(A$)=31 THEN IF CSRLIN=8 OR CS
RLIN=12 OR CSRLIN=16 THEN A=2 ELSE A=1 E
LSE GOTO 660
640 LOCATE 0,CSRLIN:PRINT " ";:LOCATE 0,
CSRLIN+A:PRINT CHR$(200);:LOCATE 25,CSRL
IN
650 GOTO 580
660 IF ASC(A$)=30 THEN IF CSRLIN=18 OR C
SRLIN=14 OR CSRLIN=10 THEN A=-2 ELSE A=-
1 ELSE GOTO 580
670 LOCATE 0,CSRLIN:PRINT " ";:LOCATE 0,
CSRLIN+A:PRINT CHR$(200);:LOCATE 25,CSRL
IN
680 GOTO 580
690 IF ASC(A$)=28 THEN Z=1 ELSE Z=-1
700 LOCATE 26,CSRLIN:PRINT "  ";:LOCATE
 26,CSRLIN
```

```
710 REM Eingeben der Soundparameter
        + oder - in den 'SOUND'-Be-
        fehl; ausserdem Ueberpruefung
        auf Erreichen der Grenzwerte
        sowie Kontrollieren der
        Bildschirmanzeige
720 REM Tonhoehe Kanal 1 fein
        ====================
730 IF CSRLIN=3 THEN IF (A(3)+Z<0 OR A(3
)+Z>255) THEN PRINT A(3);:GOTO 580 ELSE
A(3)=A(3)+Z:PRINT A(3);:SOUND 0,A(3):GOT
O 580
740 REM Tonhoehe Kanal 1 grob
        ====================
750 IF CSRLIN=4 THEN IF (A(4)+Z<0 OR A(4
)+Z>15) THEN PRINT A(4);:GOTO 580 ELSE A
(4)=A(4)+Z:PRINT A(4);:SOUND 1,A(4):GOTO
 580
760 REM Tonhoehe Kanal 2 fein
        ====================
770 IF CSRLIN=5 THEN IF (A(5)+Z<0 OR A(5
)+Z>255) THEN PRINT A(5);:GOTO 580 ELSE
A(5)=A(5)+Z:PRINT A(5);:SOUND 2,A(5):GOT
O 580
780 REM Tonhoehe Kanal 2 grob
        ====================
790 IF CSRLIN=6 THEN IF (A(6)+Z<0 OR A(6
)+Z>15) THEN PRINT A(6);:GOTO 580 ELSE A
(6)=A(6)+Z:PRINT A(6);:SOUND 3,A(6):GOTO
 580
800 REM Tonhoehe Kanal 3 fein
        ====================
810 IF CSRLIN=7 THEN IF (A(7)+Z<0 OR A(7
)+Z>255) THEN PRINT A(7);:GOTO 580 ELSE
A(7)=A(7)+Z:PRINT A(7);:SOUND 4,A(7):GOT
O 580
```

```
820 REM Tonhoehe Kanal 3 grob
        ======================
830 IF CSRLIN=8 THEN IF (A(8)+Z<0 OR A(8
)+Z>15) THEN PRINT A(8);:GOTO 580 ELSE A
(8)=A(8)+Z:PRINT A(8);:SOUND 5,A(8):GOTO
 580
840 REM Lautstaerke Kanal 1
        ====================
850 IF CSRLIN=14 THEN IF (A(14)+Z<0 OR A
(14)+Z>16) THEN PRINT A(14);:GOTO 580 EL
SE A(14)=A(14)+Z:PRINT A(14);:SOUND 8,A(
14):GOTO 580
860 REM Lautstaerke Kanal 2
        ====================
870 IF CSRLIN=15 THEN IF (A(15)+Z<0 OR A
(15)+Z>16) THEN PRINT A(15);:GOTO 580 EL
SE A(15)=A(15)+Z:PRINT A(15);:SOUND 9,A(
15):GOTO 580
880 REM Lautstaerke Kanal 3
        ====================
890 IF CSRLIN=16 THEN IF (A(16)+Z<0 OR A
(16)+Z>16) THEN PRINT A(16);:GOTO 580 EL
SE A(16)=A(16)+Z:PRINT A(16);:SOUND 10,A
(16):GOTO 580
900 REM Rauschperiode
        ==============
910 IF CSRLIN=18 THEN IF (A(18)+Z<0 OR A
(18)+Z>31) THEN PRINT A(18);:GOTO 580 EL
SE A(18)=A(18)+Z:PRINT A(18);:SOUND 6,A(
18):GOTO 580
920 REM Huellkurvenperiode fein
        ========================
930 IF CSRLIN=19 THEN IF (A(19)+Z<0 OR A
(19)+Z>255) THEN PRINT A(19);:GOTO 580 E
LSE A(19)=A(19)+Z:PRINT A(19);:SOUND 11,
A(19):GOTO 580
```

```
940 REM Huellkurvenperiode grob
        ========================
950 IF CSRLIN=20 THEN IF (A(20)+Z<0 OR A
(20)+Z>255) THEN PRINT A(20);:GOTO 580 E
LSE A(20)=A(20)+Z:PRINT A(20);:SOUND 12,
A(20):GOTO 580
960 REM Huellkurve
        ==========
970 IF CSRLIN=21 THEN IF (A(21)+Z<0 OR A
(21)+Z>15) THEN PRINT A(21);:GOTO 580 EL
SE A(21)=A(21)+Z:PRINT A(21);:SOUND 13,A
(21):GOTO 580
980 REM Kanal 1: aus/Ton/Rau/beides
        ============================
990 IF CSRLIN=10 THEN IF (A(10)+Z<0 OR A
(10)+Z>3) THEN PRINT A(10);:GOTO 580 ELS
E A(10)=A(10)+Z:PRINT A(10);
1000 REM Kanal 2: aus/Ton/Rau/beides
         ============================
1010 IF CSRLIN=11 THEN IF (A(11)+Z<0 OR
A(11)+Z>3) THEN PRINT A(11);:GOTO 580 EL
SE A(11)=A(11)+Z:PRINT A(11);
1020 REM Kanal 3: aus/Ton/Rau/beides
         ============================
1030 IF CSRLIN=12 THEN IF (A(12)+Z<0 OR
A(12)+Z>3) THEN PRINT A(12);:GOTO 580 EL
SE A(12)=A(12)+Z:PRINT A(12);
1040 IF CSRLIN<10 OR CSRLIN>12 THEN GOTO
 1120
1050 REM genaue Berechnung des 'SOUND'-
         Parameters 7
1060 A=0
1070 IF A(10)=1 THEN A=A+8 ELSE IF A(10)
=2 THEN A=A+1 ELSE IF A(10)=0 THEN A=A+1
+8
1080 IF A(11)=1 THEN A=A+16 ELSE IF A(11
)=2 THEN A=A+2 ELSE  IF A(11)=0 THEN A=A
+16+2
```

```
1090 IF A(12)=1 THEN A=A+32 ELSE IF A(12
)=2 THEN A=A+4 ELSE IF A(12)=0 THEN A=A+
32+4
1100 A=A+128
1110 SOUND 7,A
1120 GOTO 580
1130 WIDTH 40
```

Deutscher Zeichensatz

Alle schimpfen sie darüber, wenn sie ihren modernen MSX-Computer neben der alt-ehrwürdigen Schreibmaschine stehen sehen: Auf der einen Seite ein internationaler Computer ... natürlich auch mit internationalem Zeichensatz (also ohne deutsche Umlaute und 'sz'), auf der anderen Seite die Schreibmaschine, die über all' diesen nützlichen Komfort verfügt, dafür aber halt (normalerweise) kein Computer ist.

Mit diesem Programm machen wir aus Ihrem MSX-Computer ein Pendant zur Schreibmaschine, zumindest was die Wiedergabe auf dem Bildschirm angeht: das 'z' sitzt zwar nicht da, wo es sitzen soll, nämlich beim 'y' - und entsprechend umgekehrt. Auch die Umlaute 'ae', 'ue', 'oe' sowie das 'sz' finden sich noch nicht an der Stelle auf der Tastatur, wo sie eigentlich sein sollten, aber wir können diesen deutschen Zeichensatz jetzt sehr leicht ansprechen und denselbigen nicht nur auf den Bildschirm, sondern vor allen Dingen auch auf j e d e n Drucker bringen.

Wie Sie sicherlich bereits festgestellt haben, verfügen die MSX-Computer, die für den europäischen Markt bestimmt sind, bereits überwiegend über den entsprechend landestypischen Zeichensatz. Was Deutschland anbetrifft:

'ß' durch 'CODE' und '7'
'ä' durch 'CODE' und 'a'
'Ä' durch 'SHIFT', 'CODE' und 'a'
'ö' durch 'CODE' und 'f'
'Ö' durch 'SHIFT', 'CODE' und 'f'
'ü' durch 'CODE' und 'g'
'Ü' durch 'SHIFT', 'CODE' und 'g'

Jedoch ist neben der ungewöhnlichen Tastenbedienung damit lediglich sichergestellt, daß die Ausgabe auf dem Bildschirm entsprechend erfolgt; die Ausgabe auf dem Drucker funktioniert dann aber nur zu Ihrer Zufriedenheit auf speziellen MSX-Druckern, nicht wie bei dem abgebildeten Programm auch mit 'normalen' EPSON-Druckern.

Das Programm starten Sie mit 'RUN'. Sie werden anschließend gefragt, auf welchem Screen Sie zu arbeiten wünschen. Die Beantwortung dieser Frage ist wichtig, denn durch den Wechsel des Screens mit der Anweisung 'SCREEN ...' löschen Sie den umdefinierten Zeichensatz und das Programm 'Deutscher Zeichensatz' muß erneut geladen und gestartet werden.

Anschließend werden Sie gefragt, ob Ihr Computer bereits über den für den europäischen Markt bestimmten Zeichensatz verfügt (probieren Sie dies aus, indem Sie die Tasten wie oben angegeben niederdrücken - erscheinen die angegebenen Umlaute, können Sie die Frage mit 'j'(a) beantworten). Ist dies der Fall, belegt Ihr MSX-Computer nur die vorhandenen Umlaute aus dem Zeichensatz auf die entsprechenden Tasten um (s.u.), ist dies nicht der Fall (z.B. ein MSX-Computer, der für den japanischen Markt bestimmt war), werden Umlaute aus den DATA-Zeilen in den bestehenden Zeichensatz an den entsprechenden Positionen eingeladen.

Wie können Sie nun die 'deutsche Tastatur' ansprechen und damit auch den Ausdruck auf normalen Druckern mit Centronics-Anschluß erreichen?

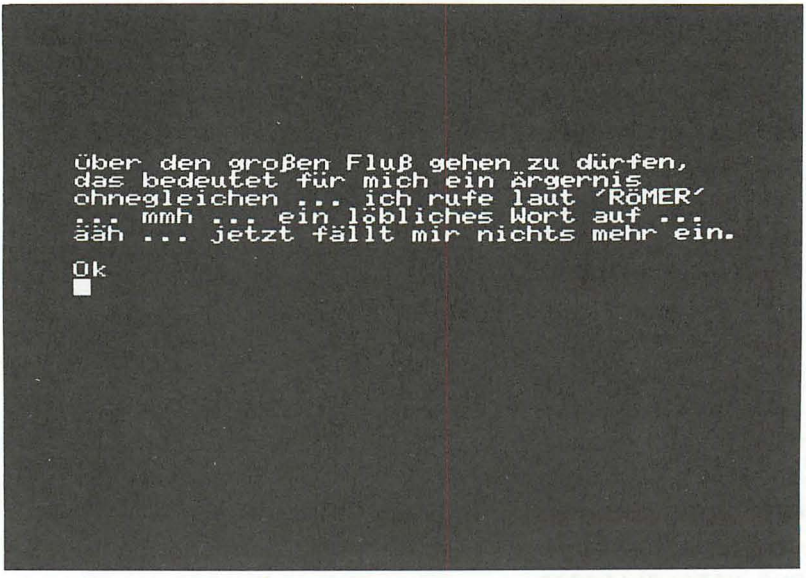
```
'[' CHR$(91)  = 'Ä'
'{' CHR$(123) = 'ä'
']' CHR$(93)  = 'Ü'
'}' CHR$(125) = 'ü'
'~' CHR$(126) = 'ß'
'\' CHR$(92)  = 'Ö'
'|' CHR$(124) = 'ö'
```

```
10 REM Deutscher Zeichensatz
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                       Rainer Lueers
40 REM Initialisieren von SCREEN 0 und 1
50 SCREEN 1:WIDTH 29:SCREEN 0:WIDTH 40:C
OLOR 15,1:KEY OFF
60 REM Vorbedingungen klaeren
70 CLS:INPUT "Screen (0/1) ";SC
80 REM Anwahl von SCREEN 1 bzw.
                    SCREEN 0
90 INPUT "Deutscher Zeichensatz vorhande
n (J/ ) ";F$:IF LEFT$(F$,1)="J" OR LEFT$
(F$,1)="j" THEN F$="J"
100 IF SC=0 THEN SCREEN 0:RESTORE 240 EL
SE SCREEN 1:RESTORE 320
110 REM Unterprogramm fuer MSX-Computer,
        deren Zeichensatz nicht der
        europaeischen Norm entspricht
        (europaeisch=Umlaute zwar
        vorhanden, aber an der falschen
        Stelle)
120 IF F$<>"J" THEN FOR N=1 TO 7:READ A(
N),A:NEXT N:RESTORE 430:FOR N=1 TO 7:FOR
 M=0 TO 7:READ A$:A$="&b"+A$:VPOKE A(N)+
M,VAL(A$):NEXT M,N:GOTO 210
130 REM Hauptprogramm fuer MSX-Computer,
        die der europaeischen Norm
        entsprechen (d.h. die Umlaute
        sind zwar im Zeichensatz ent-
        halten, aber die Bildschirmwie-
        dergabe entspricht keineswegs
140 REM der Ausgabe auf dem Drucker
150 FOR N=1 TO 7
160 READ A,B
170 FOR M=0 TO 7
180 VPOKE A+M,VPEEK(B+M)
190 NEXT M,N
```

```
200 REM Demonstrationstext
      ==================
210 IF SC=0 THEN PRINT "Über den großen
Fluß gehen zu dürfen,   das bedeutet für
 mich ein Ärgernis      ohnegleichen ...
 ich rufe laut 'RöMER'  ... mmh ... ein
löbliches Wort auf ...  ääh ... jetzt fä
llt mir nichts mehr ein."
220 IF SC=1 THEN PRINT "Über den großen
Fluß gehen zudürfen, das bedeutet für mi
chein Ärgernis ohnegleichen   ... ich r
ufe laut 'RöMER' ...mmh ... ein löbliche
s Wort   auf ... ääh ... jetzt fällt  mi
r nichts mehr ein."
230 REM Wenn Umlaute vorhanden->SCREEN 0
240 DATA 2776,3184:REM 'AE'
250 DATA 3032,3104:REM 'ae'
260 DATA 2792,3280:REM 'UE'
270 DATA 3048,3080:REM 'ue'
280 DATA 3056,3848:REM 'sz'
290 DATA 2784,3272:REM 'OE'
300 DATA 3040,3232:REM 'oe'
310 REM Wenn Umlaute vorhanden->SCREEN 1
320 DATA 728,1136:REM 'AE'
330 DATA 984,1056:REM 'ae'
340 DATA 744,1232:REM 'UE'
350 DATA 1000,1032:REM 'ue'
360 DATA 1008,1800:REM 'sz'
370 DATA 736,1224:REM 'OE'
380 DATA 992,1184:REM 'oe'
390 REM  DATAS der Umlaute und des 'sz'
400 REM  ==============================
410 REM  Der Buchstabe 'AE'
420 REM  CHR$(91)
430 DATA 01010000
440 DATA 00000000
450 DATA 00100000
460 DATA 01010000
470 DATA 10001000
480 DATA 11111000
490 DATA 10001000
500 DATA 00000000
```

```
510 REM  Der Buchstabe 'ae'
520 REM  CHR$(123)
530 DATA 01001000
540 DATA 00000000
550 DATA 01110000
560 DATA 00001000
570 DATA 01111000
580 DATA 10001000
590 DATA 01111000
600 DATA 00000000
610 REM  Der Buchstabe 'UE'
620 REM  CHR$(93)
630 DATA 01010000
640 DATA 00000000
650 DATA 10001000
660 DATA 10001000
670 DATA 10001000
680 DATA 10001000
690 DATA 01110000
700 DATA 00000000
710 REM  Der Buchstabe 'ue'
720 REM  CHR$(125)
730 DATA 10010000
740 DATA 00000000
750 DATA 00000000
760 DATA 10010000
770 DATA 10010000
780 DATA 10010000
790 DATA 01101000
800 DATA 00000000
810 REM  Der Buchstabe 'sz'
820 REM  CHR$(126)
830 DATA 00110000
840 DATA 01001000
850 DATA 01001000
860 DATA 01110000
870 DATA 01001000
880 DATA 01001000
890 DATA 01110000
900 DATA 11000000
```

```
910 REM   Der Buchstabe 'OE'
920 REM   CHR$(92)
930 DATA  01010000
940 DATA  00000000
950 DATA  01110000
960 DATA  10001000
970 DATA  10001000
980 DATA  10001000
990 DATA  01110000
1000 DATA 00000000
1010 REM  Der Buchstabe 'oe'
1020 REM  CHR$(124)
1030 DATA 10010000
1040 DATA 00000000
1050 DATA 00000000
1060 DATA 01100000
1070 DATA 10010000
1080 DATA 10010000
1090 DATA 01100000
1100 DATA 00000000
```



Computerzeichensatz

Ehrlich eingestanden kann man feststellen, daß der Zeichensatz Ihres MSX-Computers sehr umfangreich ist. So sind es nicht nur die Buchstaben von 'A' bis 'Z' in Groß- und Kleinschrift, sondern auch die Mathematikzeichen und manche Grafikelemente, die unser Gemüt hoch erfreuen.

Warum jetzt noch einen anderen Zeichensatz? Seit Jahren hält die Euphorie für Computerspiele an; warum soll für dieses futuristische Etwas der normale Zeichensatz herhalten - her mit dem entsprechenden Computerzeichensatz!

Das Programm gliedert sich in zwei Hauptteile:
1) Ein fast unübersehbares DATA-Zeichenmeer mit dem Computerzeichensatz
2) Die Verarbeitungs- und Laderoutine für die DATAS.

Ihr MSX-Computer kann auf dem Textbildschirm bis zu 256 verschiedene Zeichen ('CHR$(') gleichzeitig darstellen. Wird ein Zeichen im Aussehen verändert, so ändern sich gleichzeitig alle diesem Zeichen entsprechenden Zeichen auf dem Bildschirm ebenfalls.

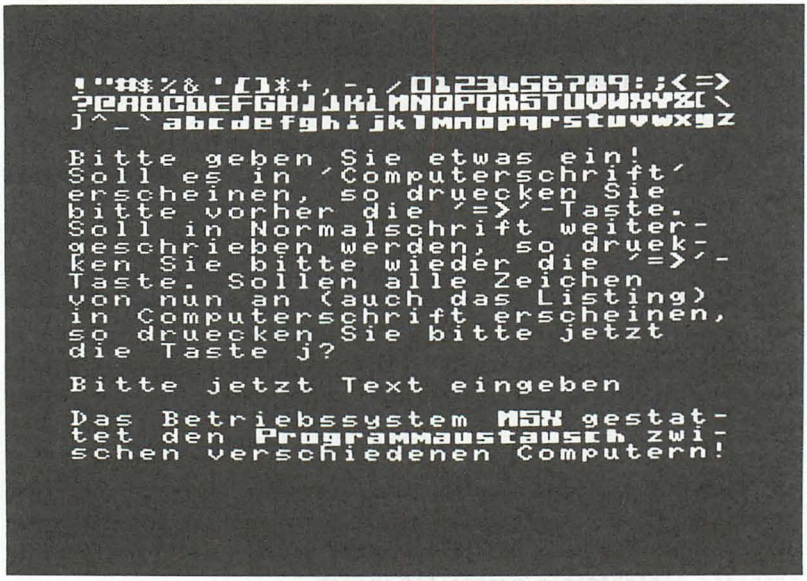
Mit diesem Programm können Sie folgendermaßen verfahren: Es kann einerseits der gesamte Zeichensatz in Computerschrift umgewandelt werden, es kann andererseits mit Normal- und Computerzeichensatz gleichzeitig gearbeitet werden.

Über ein 'UmPOKEn' des Video-RAMs erreichen wir die gesamte Neuerstellung des Zeichensatzes.

Wie werden aber zwei Zeichensätze gleichzeitig im Speicher gehalten? Hierfür wird der normale Zeichensatz (CHR$(0) bis CHR$(255)) ab 128 verändert, sprich: wir verlieren, während wir dieses Programm benutzen, einen Großteil der zusätzlichen Zeichen (Grafik, Mathematik ...) und gewinnen anstatt dessen Computerzeichen hinzu. Zum Umschalten zwischen den zwei Zeichensätzen haben wir nun bei laufendem Programm einen Schalter zwischen den

Zeichensätzen eingebaut: '=)' gedrückt -) Computerzeichensatz an, '=)' wieder gedrückt -) Normalzeichensatz an usw.

Programmtechnisch spielt sich folgendes ab: Der ASCII-Code des gewünschten Zeichens wird interpretiert (z.B. '!' = 33); dazu wird das entsprechende Computerzeichensatzbild gesucht und ... das entsprechende Computerzeichensatzschriftbild wird auf dem Bildschirm ausgegeben.



```
10 REM Computerschrift
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 SCREEN 1:COLOR 15,1,1:KEY OFF
50 WIDTH 30
60 FOR M=129 TO 218:PRINT CHR$(M);:NEXT:
PRINT
70 REM Einlesen der Computerschrift in
       die Charakter groesser als 126
80 Z=31:FOR N=1 TO 73
90 READ A:Z=Z+1
100 IF Z=A THEN FOR M=1 TO 8 ELSE FOR M=
1 TO 8:VPOKE 1023+M+((Z-32)*8),VPEEK(255
+M+((Z-32)*8)):NEXT M:Z=Z+1:GOTO 100
110 FOR M=1 TO 8
120 READ A$
130 REM Umwandlung der Stringdar-
        stellung in eine Binaerzahl und
        anschliessend in eine Dezimal-
        zahl
140 A$="&b"+A$
150 VPOKE 1031+M+((Z-33)*8),VAL(A$)
160 NEXT M
170 NEXT N
180 REM Auswahlmoeglichkeiten:
        1) bei laufendem Programm
           Text in Computerschrift oder
           Normalschrift einzugeben
        2) den gesamten normalen Zei-
           chensatz (CHR$(32) bis
190 REM    CHR$(126)) in Computerschrift
           umzuwandeln
200 PRINT "Bitte geben Sie etwas ein!
 Soll es in 'Computerschrift' erscheine
n, so druecken Sie  bitte vorher die '=
>'-Taste.  Soll in Normalschrift weiter-
 geschrieben werden, so druek- ken Sie b
itte wieder die '=>'-";
```

```
210 PRINT "Taste. Sollen alle Zeichen
 von nun an (auch das Listing) in Comput
erschrift erscheinen,so druecken Sie bit
te jetzt   die Taste j";:A$="":INPUT A$
220 IF LEFT$(A$,1)="J" OR LEFT$(A$,1)="j
" THEN A$="J"
230 IF A$="J" THEN GOTO 510
240 REM Der Normalzeichensatz bleibt
        erhalten, nur einzelne Buchsta-
        ben, Zahlen und Zeichen werden
        auf Wunsch in Computerzeichen
        umgewandelt
250 PRINT
260 PRINT "Bitte jetzt Text eingeben"
270 PRINT
280 A$=INKEY$
290 IF A$="" THEN GOTO 280
300 REM Kleiner Texteditor: Die Tasten
        <ENTER>=13,<=>>=9 und <<=>=8
        werden interpretiert
310 IF ASC(A$)=9 THEN GOTO 370
320 IF ASC(A$)=13 THEN PRINT:GOTO 280
330 IF ASC(A$)=8 THEN A$="":IF POS(0)>0
THEN LOCATE POS(0)-1,CSRLIN:PRINT " ";:L
OCATE POS(0)-1,CSRLIN:GOTO 280
340 PRINT A$;
350 GOTO 280
360 END
370 A$=INKEY$
380 IF A$="" THEN GOTO 370
390 REM Kleiner Texteditor: Die Tasten
        <ENTER>=13, <=>>=9 und <<=>=8
        werden interpretiert
400 IF ASC(A$)=9 THEN GOTO 280
410 IF ASC(A$)=13 THEN PRINT:GOTO 370
420 IF ASC(A$)=8 THEN A$="":IF POS(0)>0
THEN LOCATE POS(0)-1,CSRLIN:PRINT " ";:L
OCATE POS(0)-1,CSRLIN:GOTO 370 ELSE GOTO
 370
```

```
430 IF ASC(A$)>122 THEN PRINT A$;:GOTO 3
70
440 REM Jedes Zeichen wird aus den zur
        Verfuegung stehenden Charakter-
        strings herausgelesen, so dass
        der normale Zeichensatz nicht
        zerstoert wird
450 Z=ASC(A$)
460 REM Auswahl der Charakter
470 PRINT CHR$(Z+96);
480 GOTO 370
490 END
500 REM Einlesen der Computerschrift und
        direkte Veraenderung des
        normalen Zeichensatzes
510 FOR N=0 TO 719
520 VPOKE 264+N,VPEEK(1032+N)
530 NEXT N
540 END
550 REM !
560 DATA 33
570 DATA 00111000
580 DATA 00111000
590 DATA 00111000
600 DATA 00111000
610 DATA 00011000
620 DATA 00000000
630 DATA 00011000
640 DATA 00000000
650 REM "
660 DATA 34
670 DATA 01100110
680 DATA 01100110
690 DATA 01100110
700 DATA 00000000
710 DATA 00000000
720 DATA 00000000
730 DATA 00000000
740 DATA 00000000
```

```
750 REM #
760 DATA 35
770 DATA 01100110
780 DATA 11111111
790 DATA 01100110
800 DATA 01100110
810 DATA 11111111
820 DATA 01100110
830 DATA 00000000
840 DATA 00000000
850 REM '
860 DATA 39
870 DATA 00011000
880 DATA 00011000
890 DATA 00011000
900 DATA 00000000
910 DATA 00000000
920 DATA 00000000
930 DATA 00000000
940 DATA 00000000
950 REM (
960 DATA  40
970 DATA  00011110
980 DATA  00011000
990 DATA  00011000
1000 DATA 00111000
1010 DATA 00111000
1020 DATA 00111000
1030 DATA 00111110
1040 DATA 00000000
1050 REM )
1060 DATA 41
1070 DATA 01111000
1080 DATA 00011000
1090 DATA 00011000
1100 DATA 00011100
1110 DATA 00011100
1120 DATA 00011100
1130 DATA 01111100
1140 DATA 00000000
```

```
1150 REM Ø
1160 DATA 48
1170 DATA Ø1111111
1180 DATA Ø11ØØØ11
1190 DATA Ø11ØØØ11
1200 DATA Ø11ØØØ11
1210 DATA Ø11ØØØ11
1220 DATA Ø11ØØØ11
1230 DATA Ø1111111
1240 DATA ØØØØØØØØ
1250 REM 1
1260 DATA 49
1270 DATA ØØ1110ØØ
1280 DATA ØØØ11ØØØ
1290 DATA ØØØ11ØØØ
1300 DATA ØØØ11ØØØ
1310 DATA ØØ111110
1320 DATA ØØ111110
1330 DATA ØØ111110
1340 DATA ØØØØØØØØ
1350 REM 2
1360 DATA 5Ø
1370 DATA Ø1111111
1380 DATA ØØØØØØ11
1390 DATA ØØØØØØ11
1400 DATA Ø1111111
1410 DATA Ø11ØØØØØ
1420 DATA Ø11ØØØØØ
1430 DATA Ø1111111
1440 DATA ØØØØØØØØ
1450 REM 3
1460 DATA 51
1470 DATA Ø1111110
1480 DATA ØØØØØ11Ø
1490 DATA ØØØØØ11Ø
1500 DATA Ø1111111
1510 DATA ØØØØØ111
1520 DATA ØØØØØ111
1530 DATA 11111111
1540 DATA ØØØØØØØØ
```

```
1550 REM 4
1560 DATA 52
1570 DATA Ø1110ØØØ
1580 DATA Ø1110ØØØ
1590 DATA Ø1110ØØØ
1600 DATA Ø1110111
1610 DATA Ø1110111
1620 DATA Ø1111111
1630 DATA ØØØØØ111
1640 DATA ØØØØØØØØ
1650 REM 5
1660 DATA 53
1670 DATA Ø1111111
1680 DATA Ø11ØØØØØ
1690 DATA Ø11ØØØØØ
1700 DATA Ø1111111
1710 DATA ØØØØØ111
1720 DATA ØØØØØ111
1730 DATA Ø1111111
1740 DATA ØØØØØØØØ
1750 REM 6
1760 DATA 54
1770 DATA Ø1111ØØ
1780 DATA Ø11Ø11ØØ
1790 DATA Ø11ØØØØØ
1800 DATA Ø1111111
1810 DATA Ø11ØØØ11
1820 DATA Ø11ØØØ11
1830 DATA Ø1111111
1840 DATA ØØØØØØØØ
1850 REM 7
1860 DATA 55
1870 DATA Ø1111111
1880 DATA ØØØØØØ11
1890 DATA ØØØØØØ11
1900 DATA ØØØ11111
1910 DATA ØØØ11ØØØ
1920 DATA ØØØ11ØØØ
1930 DATA ØØØ11ØØØ
1940 DATA ØØØØØØØØ
```

```
1950 REM 8
1960 DATA 56
1970 DATA 00111110
1980 DATA 00110110
1990 DATA 00110110
2000 DATA 01111111
2010 DATA 01110111
2020 DATA 01110111
2030 DATA 01111111
2040 DATA 00000000
2050 REM 9
2060 DATA 57
2070 DATA 01111111
2080 DATA 01100011
2090 DATA 01100011
2100 DATA 01111111
2110 DATA 00000111
2120 DATA 00000111
2130 DATA 00000111
2140 DATA 00000000
2150 REM :
2160 DATA 58
2170 DATA 00000000
2180 DATA 00011000
2190 DATA 00011000
2200 DATA 00000000
2210 DATA 00011000
2220 DATA 00011000
2230 DATA 00000000
2240 DATA 00000000
2250 REM ;
2260 DATA 59
2270 DATA 00000000
2280 DATA 00011000
2290 DATA 00011000
2300 DATA 00000000
2310 DATA 00011000
2320 DATA 00011000
2330 DATA 00110000
2340 DATA 00000000
```

```
2350 REM =
2360 DATA 61
2370 DATA 00000000
2380 DATA 01111110
2390 DATA 00000000
2400 DATA 00000000
2410 DATA 01111110
2420 DATA 00000000
2430 DATA 00000000
2440 DATA 00000000
2450 REM ?
2460 DATA 63
2470 DATA 01111111
2480 DATA 01100011
2490 DATA 00000011
2500 DATA 00011111
2510 DATA 00011100
2520 DATA 00000000
2530 DATA 00011100
2540 DATA 00000000
2550 REM §
2560 DATA 64
2570 DATA 01111111
2580 DATA 01100011
2590 DATA 01101111
2600 DATA 01101111
2610 DATA 01101111
2620 DATA 01100000
2630 DATA 01111111
2640 DATA 00000000
2650 REM A
2660 DATA 65
2670 DATA 00111111
2680 DATA 00110011
2690 DATA 00110011
2700 DATA 01111111
2710 DATA 01110011
2720 DATA 01110011
2730 DATA 01110011
2740 DATA 00000000
```

```
2750 REM B
2760 DATA 66
2770 DATA 01111110
2780 DATA 01100110
2790 DATA 01100110
2800 DATA 01111111
2810 DATA 01100111
2820 DATA 01100111
2830 DATA 01111111
2840 DATA 00000000
2850 REM C
2860 DATA 67
2870 DATA 01111111
2880 DATA 01100111
2890 DATA 01100111
2900 DATA 01100000
2910 DATA 01100011
2920 DATA 01100011
2930 DATA 01111111
2940 DATA 00000000
2950 REM D
2960 DATA 68
2970 DATA 01111110
2980 DATA 01100110
2990 DATA 01100110
3000 DATA 01110111
3010 DATA 01110111
3020 DATA 01110111
3030 DATA 01111111
3040 DATA 00000000
3050 REM E
3060 DATA 69
3070 DATA 01111111
3080 DATA 01100000
3090 DATA 01100000
3100 DATA 01111111
3110 DATA 01110000
3120 DATA 01110000
3130 DATA 01111111
3140 DATA 00000000
```

```
3150 REM F
3160 DATA 70
3170 DATA Ø1111111
3180 DATA Ø1100000
3190 DATA Ø1100000
3200 DATA Ø1111111
3210 DATA Ø1110000
3220 DATA Ø1110000
3230 DATA Ø1110000
3240 DATA ØØØØØØØØ
3250 REM G
3260 DATA 71
3270 DATA Ø1111111
3280 DATA Ø1100011
3290 DATA Ø1100000
3300 DATA Ø1101111
3310 DATA Ø1100111
3320 DATA Ø1100111
3330 DATA Ø1111111
3340 DATA ØØØØØØØØ
3350 REM H
3360 DATA 72
3370 DATA Ø1110011
3380 DATA Ø1110011
3390 DATA Ø1110011
3400 DATA Ø1111111
3410 DATA Ø1110011
3420 DATA Ø1110011
3430 DATA Ø1110011
3440 DATA ØØØØØØØØ
3450 REM I
3460 DATA 73
3470 DATA ØØØØ1100
3480 DATA ØØØØ1100
3490 DATA ØØØØ1100
3500 DATA ØØØØ1100
3510 DATA ØØ111100
3520 DATA ØØ111100
3530 DATA ØØ111100
3540 DATA ØØØØØØØØ
```

```
3550 REM J
3560 DATA 74
3570 DATA 00001100
3580 DATA 00001100
3590 DATA 00001100
3600 DATA 00001110
3610 DATA 00001110
3620 DATA 01101110
3630 DATA 01111110
3640 DATA 00000000
3650 REM K
3660 DATA 75
3670 DATA 01100110
3680 DATA 01100110
3690 DATA 01101100
3700 DATA 01111111
3710 DATA 01100111
3720 DATA 01100111
3730 DATA 01100111
3740 DATA 00000000
3750 REM L
3760 DATA 76
3770 DATA 00110000
3780 DATA 00110000
3790 DATA 00110000
3800 DATA 01110000
3810 DATA 01110000
3820 DATA 01110000
3830 DATA 01111110
3840 DATA 00000000
3850 REM M
3860 DATA 77
3870 DATA 01100111
3880 DATA 01111111
3890 DATA 01111111
3900 DATA 01110111
3910 DATA 01100111
3920 DATA 01100111
3930 DATA 01100111
3940 DATA 00000000
```

```
3950 REM N
3960 DATA 78
3970 DATA 01100111
3980 DATA 01110111
3990 DATA 01111111
4000 DATA 01101111
4010 DATA 01100111
4020 DATA 01100111
4030 DATA 01100111
4040 DATA 00000000
4050 REM O
4060 DATA 79
4070 DATA 01111111
4080 DATA 01100011
4090 DATA 01100011
4100 DATA 01100111
4110 DATA 01100111
4120 DATA 01100111
4130 DATA 01111111
4140 DATA 00000000
4150 REM P
4160 DATA 80
4170 DATA 01111111
4180 DATA 01100011
4190 DATA 01100011
4200 DATA 01111111
4210 DATA 01110000
4220 DATA 01110000
4230 DATA 01110000
4240 DATA 00000000
4250 REM Q
4260 DATA 81
4270 DATA 01111111
4280 DATA 01100011
4290 DATA 01100011
4300 DATA 01100111
4310 DATA 01100111
4320 DATA 01100111
4330 DATA 01111111
4340 DATA 00000111
```

```
4350 REM R
4360 DATA 82
4370 DATA 01111110
4380 DATA 01100110
4390 DATA 01100110
4400 DATA 01111111
4410 DATA 01110111
4420 DATA 01110111
4430 DATA 01110111
4440 DATA 00000000
4450 REM S
4460 DATA 83
4470 DATA 01111111
4480 DATA 01100000
4490 DATA 01111111
4500 DATA 00000011
4510 DATA 01110011
4520 DATA 01110011
4530 DATA 01111111
4540 DATA 00000000
4550 REM T
4560 DATA 84
4570 DATA 01111111
4580 DATA 00011100
4590 DATA 00011100
4600 DATA 00011100
4610 DATA 00011100
4620 DATA 00011100
4630 DATA 00011100
4640 DATA 00000000
4650 REM U
4660 DATA 85
4670 DATA 01100111
4680 DATA 01100111
4690 DATA 01100111
4700 DATA 01100111
4710 DATA 01100111
4720 DATA 01100111
4730 DATA 01111111
4740 DATA 00000000
```

```
4750 REM V
4760 DATA 86
4770 DATA 01100111
4780 DATA 01100111
4790 DATA 01100111
4800 DATA 01100111
4810 DATA 01101111
4820 DATA 00111110
4830 DATA 00011100
4840 DATA 00000000
4850 REM W
4860 DATA 87
4870 DATA 01100111
4880 DATA 01100111
4890 DATA 01100111
4900 DATA 01101111
4910 DATA 01111111
4920 DATA 01111111
4930 DATA 01100111
4940 DATA 00000000
4950 REM X
4960 DATA 88
4970 DATA 01110011
4980 DATA 01110011
4990 DATA 01110011
5000 DATA 00111110
5010 DATA 01100111
5020 DATA 01100111
5030 DATA 01100111
5040 DATA 00000000
5050 REM Y
5060 DATA 89
5070 DATA 01100111
5080 DATA 01100111
5090 DATA 01100111
5100 DATA 01111111
5110 DATA 00011100
5120 DATA 00011100
5130 DATA 00011100
5140 DATA 00000000
```

```
5150 REM Z
5160 DATA 90
5170 DATA 01111111
5180 DATA 01100110
5190 DATA 01101100
5200 DATA 00011000
5210 DATA 00110111
5220 DATA 01100111
5230 DATA 01111111
5240 DATA 00000000
5250 REM a
5260 DATA 97
5270 DATA 00000000
5280 DATA 00000000
5290 DATA 00111110
5300 DATA 00000110
5310 DATA 01111110
5320 DATA 01100110
5330 DATA 01111110
5340 DATA 00000000
5350 REM b
5360 DATA 98
5370 DATA 00000000
5380 DATA 01110000
5390 DATA 01110000
5400 DATA 01111110
5410 DATA 01110110
5420 DATA 01110110
5430 DATA 01111110
5440 DATA 00000000
5450 REM c
5460 DATA 99
5470 DATA 00000000
5480 DATA 00000000
5490 DATA 01111100
5500 DATA 01110000
5510 DATA 01110000
5520 DATA 01110000
5530 DATA 01111100
5540 DATA 00000000
```

```
5550 REM d
5560 DATA 100
5570 DATA 00000000
5580 DATA 00000110
5590 DATA 00000110
5600 DATA 01111110
5610 DATA 01101110
5620 DATA 01101110
5630 DATA 01111110
5640 DATA 00000000
5650 REM e
5660 DATA 101
5670 DATA 00000000
5680 DATA 00000000
5690 DATA 01111110
5700 DATA 01100110
5710 DATA 01111110
5720 DATA 01100000
5730 DATA 01111110
5740 DATA 00000000
5750 REM f
5760 DATA 102
5770 DATA 00000000
5780 DATA 00011110
5790 DATA 00011000
5800 DATA 00111110
5810 DATA 00011000
5820 DATA 00011000
5830 DATA 00011000
5840 DATA 00000000
5850 REM g
5860 DATA 103
5870 DATA 00000000
5880 DATA 00000000
5890 DATA 01111110
5900 DATA 01101110
5910 DATA 01101110
5920 DATA 01111110
5930 DATA 00001110
5940 DATA 01111110
```

```
5950 REM h
5960 DATA 104
5970 DATA 00000000
5980 DATA 01110000
5990 DATA 01110000
6000 DATA 01111100
6010 DATA 01110110
6020 DATA 01110110
6030 DATA 01110110
6040 DATA 00000000
6050 REM i
6060 DATA 105
6070 DATA 00000000
6080 DATA 00011000
6090 DATA 00000000
6100 DATA 00011000
6110 DATA 00011000
6120 DATA 00111100
6130 DATA 00111100
6140 DATA 00000000
6150 REM j
6160 DATA 106
6170 DATA 00000000
6180 DATA 00001110
6190 DATA 00000000
6200 DATA 00001110
6210 DATA 00001110
6220 DATA 00001110
6230 DATA 00001110
6240 DATA 00111110
6250 REM k
6260 DATA 107
6270 DATA 00000000
6280 DATA 01100000
6290 DATA 01100000
6300 DATA 01101100
6310 DATA 01111000
6320 DATA 01101100
6330 DATA 01100110
6340 DATA 00000000
```

```
6350 REM l
6360 DATA 108
6370 DATA 00000000
6380 DATA 00111100
6390 DATA 00011100
6400 DATA 00011100
6410 DATA 00011100
6420 DATA 00011100
6430 DATA 00011100
6440 DATA 00000000
6450 REM m
6460 DATA 109
6470 DATA 00000000
6480 DATA 00000000
6490 DATA 01100111
6500 DATA 01111111
6510 DATA 01111111
6520 DATA 01101011
6530 DATA 01100011
6540 DATA 00000000
6550 REM n
6560 DATA 110
6570 DATA 00000000
6580 DATA 00000000
6590 DATA 01111110
6600 DATA 01111110
6610 DATA 01100110
6620 DATA 01100110
6630 DATA 01100110
6640 DATA 00000000
6650 REM o
6660 DATA 111
6670 DATA 00000000
6680 DATA 00000000
6690 DATA 01111110
6700 DATA 01101110
6710 DATA 01101110
6720 DATA 01101110
6730 DATA 01111110
6740 DATA 00000000
```

```
6750 REM p
6760 DATA 112
6770 DATA 00000000
6780 DATA 00000000
6790 DATA 01111110
6800 DATA 01110110
6810 DATA 01110110
6820 DATA 01111110
6830 DATA 01110000
6840 DATA 01110000
6850 REM q
6860 DATA 113
6870 DATA 00000000
6880 DATA 00000000
6890 DATA 01111110
6900 DATA 01101110
6910 DATA 01101110
6920 DATA 01111110
6930 DATA 00001110
6940 DATA 00001110
6950 REM r
6960 DATA 114
6970 DATA 00000000
6980 DATA 00000000
6990 DATA 01111110
7000 DATA 01110110
7010 DATA 01110000
7020 DATA 01110000
7030 DATA 01110000
7040 DATA 00000000
7050 REM s
7060 DATA 115
7070 DATA 00000000
7080 DATA 00000000
7090 DATA 01111110
7100 DATA 01100000
7110 DATA 01111110
7120 DATA 00001110
7130 DATA 01111110
7140 DATA 00000000
```

```
7150 REM t
7160 DATA 116
7170 DATA 00000000
7180 DATA 00111000
7190 DATA 01111110
7200 DATA 00111000
7210 DATA 00111000
7220 DATA 00111000
7230 DATA 00111110
7240 DATA 00000000
7250 REM u
7260 DATA 117
7270 DATA 00000000
7280 DATA 00000000
7290 DATA 01101110
7300 DATA 01101110
7310 DATA 01101110
7320 DATA 01101110
7330 DATA 01111110
7340 DATA 00000000
7350 REM v
7360 DATA 118
7370 DATA 00000000
7380 DATA 00000000
7390 DATA 01101110
7400 DATA 01101110
7410 DATA 01101110
7420 DATA 00111100
7430 DATA 00011000
7440 DATA 00000000
7450 REM w
7460 DATA 119
7470 DATA 00000000
7480 DATA 00000000
7490 DATA 01100011
7500 DATA 01101011
7510 DATA 01111111
7520 DATA 01111111
7530 DATA 00110110
7540 DATA 00000000
```

```
7550 REM x
7560 DATA 120
7570 DATA 00000000
7580 DATA 00000000
7590 DATA 01100110
7600 DATA 00111100
7610 DATA 00011000
7620 DATA 00111100
7630 DATA 01100110
7640 DATA 00000000
7650 REM y
7660 DATA 121
7670 DATA 00000000
7680 DATA 00000000
7690 DATA 01101110
7700 DATA 01101110
7710 DATA 01101110
7720 DATA 01111110
7730 DATA 00001110
7740 DATA 01111110
7750 REM z
7760 DATA 122
7770 DATA 00000000
7780 DATA 00000000
7790 DATA 01111110
7800 DATA 00001100
7810 DATA 00011000
7820 DATA 00110000
7830 DATA 01111110
7840 DATA 00000000
```

Errormeldungen

Manchmal ist es wie verhext: Man gibt sein lange Zeit ausgearbeitetes BASIC-Programm in den Computer ein und trotz größter Bemühungen, hier und da treten laufend Fehler auf.

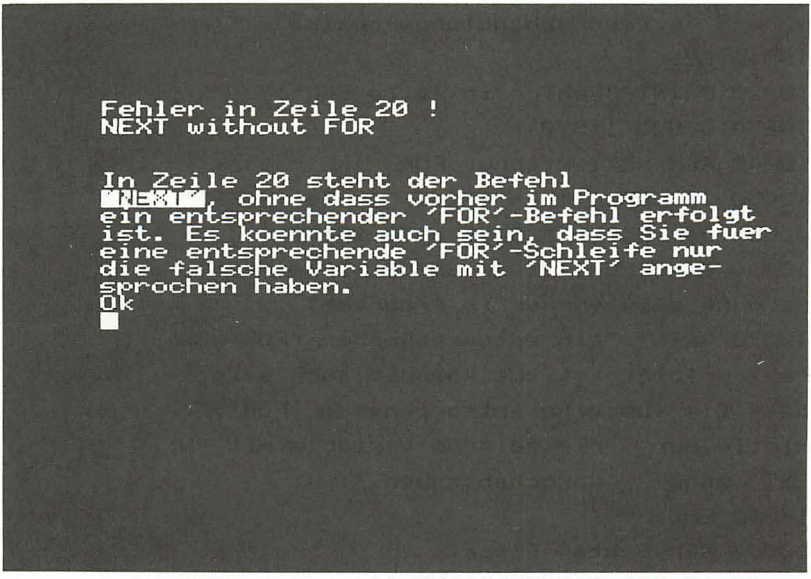
Da Ihr MSX-Computer ursprünglich nicht aus Deutschland stammt, werden die Fehlermeldungen selbstverständlich in englischer Sprache ausgeworfen. Da Ihr MSX-Computer zudem nicht über unbegrenzten Speicherraum verfügt, sind diese Fehlermeldungen oft nur ein allzu kurzer Hinweis ('Syntax error', 'Type mismatch' ...).

Diesen Hinweis lernt der eifrige Computerprogrammierer mit der Zeit richtig kennen und zu verstehen, aber bis dahin ist es oft noch ein weiter Weg. Aus diesem Grund haben wir ein ausführliches Errormeldungsprogramm in diese Programmsammlung mit aufgenommen. Da Ihr MSX-Computer den Vorgabebefehl 'ON ERROR GOTO' kennt, kann er bei einem auftretenden Fehler so zu reagieren lernen, wie wir das wollen (er kann sogar selbst die Fehler teilweise korrigieren; bestes Beispiel dazu: Ihr MSX-Computer lädt Daten von Kassette; schließlich sind alle vorhandenen Daten eingelesen; normalerweise wird nun bei weiteren Leseversuchen die Fehlermeldung ausgegeben: 'Input past end'; viel einfacher ist es da doch, in einer Errorbehandlungsroutine nach Auftreten dieses Fehlers ('ON ERROR GOTO') den noch offenen Datenfile zu schließen und im Programm dort fortzufahren, wo dies im Augenblick sinnvoll erscheint).

Zwar merzt unser Errormeldungsprogramm keine vorhandenen Errors aus (das kann pauschal so gehandhabt leicht zu unvorhersehbaren Folgen führen), dafür hilft es aber bei der Suche nach dem Fehler um so effektiver. Am besten Sie probieren es einmal aus! Geben Sie in Zeile 20 nur 'NEXT N' ein. Starten wir das Programm, so wird gleich der Fehler (Error 1 = 'NEXT without FOR') angezeigt und eine Lösungsstrategie zur Programmverbesserung angeboten.

Wie verfahren Sie mit Ihren bereits bestehenden Programmen? Zeilen so umnummerieren (Befehl 'RENUM'), daß sich keine Zeile mit dem Errorbehandlungsprogramm überschneidet (die Zeilennummern müssen größer als 20 und kleiner als 10000 sein). Nun gebrauchen Sie den Befehl 'MERGE' (siehe Handbuch), um die beiden Programme miteinander zu verbinden; anschließend mit 'RUN' starten und versuchen, die Fehlermeldungen mit den vorgegebenen Korrekturangeboten positiv anzuwenden.

Neben dem BASIC-Befehl 'ON ERROR GOTO' hilft Ihr MSX-Computer am besten weiter durch die Abfrage der Error-Zeilenvariable 'ERL' und die Fehlermeldevariable 'ERR'. Fehler selbst erzeugen können wir schließlich mit dem Befehl 'ERROR'.



```
10 REM Errormeldungen
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 KEY OFF:SCREEN 0:WIDTH 40:PRINT "Bitt
e einen Augenblick Geduld":GOSUB 11940:C
OLOR 15,1:PRINT:PRINT "Es ist ratsam, Ze
ile 40 nun zu loeschen,falls der SCREEN
nicht geaendert wird."
50 ON ERROR GOTO 10000
60 REM Bitte Programm zwischen Zeile
       20 und Zeile 9999 eintippen
       oder einMERGEN
9999 STOP
10000 REM Beginn der ausfuehrlichen
          Errorbehandlungsroutine
10010 CLS
10020 PRINT "Fehler in Zeile"ERL"!"
10030 GOSUB 11570
10040 REM NEXT without FOR
10050 IF ERR=1 THEN PRINT "In Zeile"ERL"
steht der Befehl" ELSE GOTO 10100
10060 A$="'NEXT'":A=0:GOSUB 12060:PRINT
", ohne dass vorher im Programm"
10070 PRINT "ein entsprechender 'FOR'-Be
fehl erfolgt ist. Es koennte auch sein,
dass Sie fuereine entsprechende 'FOR'-Sc
hleife nur   die falsche Variable mit 'N
EXT' ange-   sprochen haben."
10080 END
10090 REM Syntax Error
10100 IF ERR=2 THEN PRINT "Irgendetwas i
st":PRINT "im Satzbau von Zeile"ERL ELSE
 GOTO 10150
10110 PRINT "falsch bzw. irgendein Wort
ist falsch   geschrieben. Man spricht vo
n einem     ":A$="SYNTAX ERROR":A=0:GOSU
B 12060:PRINT"."
10120 PRINT
10130 END
```

```
10140 REM RETURN without GOSUB
10150 IF ERR=3 THEN PRINT "In Zeile"ERL"
steht der Befehl":A$="RETURN":A=0:GOSUB
12060:PRINT ", ohne dass vorher im"ELSE
GOTO 10190
10160 PRINT "Programm ein entsprechendes
 Unterpro-   gramm mit 'GOSUB' aufgerufe
n wurde."
10170 END
10180 REM Out of DATA
10190 IF ERR=4 THEN PRINT "Es sollen nac
h Ihren Wuenschen noch     weitere ";:A$
="DATAS":A=0:GOSUB 12060:PRINT " eingele
sen werden." ELSE GOTO 10230
10200 PRINT "Dafuer reicht die angegeben
e Menge der  DATAS aber nicht aus. Viell
eicht wurde  auch zum nochmaligen Lesen
der DATAS    mit dem Befehl 'READ' nur e
in entspre-  chender DATAruecksetzbefehl
 wie         'RESTORE' vergessen."
10210 END
10220 REM Illegal function call
10230 IF ERR=5 THEN PRINT "Ein Parameter
 oder ein ";:A$="Argument":A=0:GOSUB 120
60:PRINT " wur-" ELSE GOTO 10270
10240 PRINT "de falsch angegeben. Sie ko
ennen        z.B. nicht beliebige Zahlwe
rte fuer den 'SOUND' oder den 'DRAW'-Bef
ehl verwen-  den.Bitte schauen Sie im Ha
ndbuch nach!"
10250 END
10260 REM Overflow
10270 IF ERR=6 THEN PRINT "Ihr MSX-Compu
ter registriert einen":A$="OVERFLOW":A=0
:GOSUB 12060:PRINT "; das bedeutet einen
" ELSE GOTO 10320
```

```
10280 PRINT "Ueberlauf, da in Zeile"ERL"
versucht":PRINT "wurde, eine Zahl darzus
tellen, die      ueber die Darstellungsm
oeglichkeit von  Zahlen hinausging (groe
sser             als 1.7 E+63). Vielleic
ht haben Sie auchdie Umwandlung von Zahl
en in andere"
10290 PRINT "Zahldarstellungen bei unerl
aubter       Groesse angestrebt."
10300 END
10310 REM Out of memory
10320 IF ERR=7 THEN PRINT "Speicher oder
 englisch ";:A$="MEMORY":A=0:GOSUB 12060
:PRINT " voll." ELSE GOTO 10380
10330 PRINT "Das ist ein starkes Stueck,
 wenn Sie denSpeicher mit einem reinen B
ASIC-Programmvollgeschrieben haben. Am b
esten, Sie  schauen das Programm noch e
inmal durch, wo vielleicht Kuerzungsmoeg
lichkeiten  bestehen z.B. sind 'REM'-Ze
ilen nur zur"
10340 PRINT "Dokumentation des Programms
 noetig.     Uebrigens ... Sie kommen sc
hnell zu ei- nem vollen Speicher, wenn S
ie die Dimen-sionen von Variablen zu hoc
h auslegen  oder nacheinander in zu vie
le Unterpro- gramme oder 'FOR...NEXT'-Sc
hleifen ver-"
10350 PRINT "zweigen."
10360 END
10370 REM Undefined line number
10380 IF ERR=8 THEN PRINT "Sie wollen ei
ne Zeile im Programm an-  springen, die
 es ueberhaupt nicht gibt  oder in engli
sch: eine ":A$="Undefined line number":A
=0:GOSUB 12060:PRINT "." ELSE GOTO 10420
10390 PRINT "Vielleicht haben Sie da ein
en Schritt  zu weit gedacht oder gar vo
rhin das Pro-gramm umnummeriert und die
gleiche Feh- lermeldung (s.o.) ignoriert
."
```

```
10400 END
10410 REM Subscript out of range
10420 IF ERR=9 THEN PRINT "Das ";:A$="Su
bscript":A=0:GOSUB 12060:PRINT " (die Gr
oesse der ge-" ELSE GOTO 10470
10430 PRINT "waehlten Dimension einer Va
riablen) ist in Zeile";ERL;"zu gross.":P
RINT "Entweder haben Sie vorhin vergesse
n, dieVariablen mit 'DIM' zu dimensionie
ren   oder aber Sie haben gedacht, Ihr M
SX-"
10440 PRINT "Computer koenne mehr als 11
 Variablen,  die einer Variable zugeordn
et sind, ohneDimensionierung verkraften.
"
10450 END
10460 REM Redimensioned Array
10470 IF ERR=10 THEN PRINT "Die Dimensio
nierung mit ";:A$="DIM":A=0:GOSUB 12060:
PRINT " wurde" ELSE GOTO 10510
10480 PRINT "schon einmal vorher im Prog
ramm fest-   gelegt. Es ist nicht moegli
ch, in Zei-  le";ERL;"noch einmal, ohne
vorher":PRINT "'CLEAR' gesagt zu haben,
neu zu dimen-  sionieren."
10490 END
10500 REM Division by zero
10510 IF ERR=11 THEN A$="DIVISION":A=0:G
OSUB 12060:PRINT " durch 0 ist nicht erl
aubt.":PRINT "Hier ist es dennoch gesche
hen! Bitte im Programm nachsehen, wie es
 dazu kommen  konnte (auch Variablen in
der Fehlerzei-le ueberpruefen)." ELSE GO
TO 10540
10520 END
10530 REM Illegal direct
10540 IF ERR=12 THEN PRINT "Als direkte
Befehlseingabe (";:A$="Illegal":A=0:GOSU
B 12060:PRINT:A$="direct":A=0:GOSUB 1206
0:PRINT ") ist dies so nicht statthaft."
 ELSE GOTO 10580
```

```
10550 PRINT "Bitte diesen Befehl in ein
Programm ein-binden und dieses mit 'RUN'
 starten."
10560 END
10570 REM Type mismatch
10580 IF ERR=13 THEN PRINT "Sie haben de
n falschen Variablentyp":PRINT "(";:A$="
TYPE MISMATCH":A=0:GOSUB 12060:PRINT ")
ausgewaehlt.":PRINT "Entweder fragt man
mit einer Variablen  nur nach einer Zahl
 (Variable A,B,C...)" ELSE GOTO 10630
10590 PRINT "und liest danach auch nur e
ine Zahl ein (durch 'READ' oder 'INPUT')
, oder man   erwartet ein alphanumerisch
es Zeichen   bzw. eine Zahl und liest di
eses mit     einer Stringvariablen (A$,B
$,C$...) ein."
10600 PRINT "Will man eine Zahl als Stri
ng einlesen, so kann der String durch An
wendung des  Befehls 'VAL(...)' wieder i
n eine Zahl- variable zurueckverwandelt
werden."
10610 END
10620 REM Out of string space
10630 IF ERR=14 THEN A$="STRINGS":A=0:GO
SUB 12060:PRINT " finden keinen freien P
latz mehr":PRINT "im Speicher. Es gibt z
wei Moeglichkei-  ten zur weiteren Strin
geingabe:" ELSE GOTO 10680
10640 PRINT:PRINT "1) Mit 'CLEAR' mehr S
peicherplatz fuer     Strings schaffen
                  2) Die vorhandenen St
rings auf Kassette    abspeichern und mi
t 'CLEAR' den         Stringspeicher loe
schen. Danach kann    die Neueingabe der
 Variablen"
10650 PRINT "   erfolgen."
10660 END
```

```
10670 REM String too long
10680 IF ERR=15 THEN PRINT "Die in der Z
eile";ERL;"verwendete":PRINT "Zeichenket
te (";:A$="STRING":A=0:GOSUB 12060:PRINT
 ") ist laenger als" ELSE GOTO 10720
10690 PRINT "255 Zeichen. Dies ist unbed
ingt zu ver- meiden! Deshalb nicht belie
bige Strings erweitern (durch '+') und a
uch nicht zu oft die Stringmanipulations
funktionen   wie 'INSTR' usw. als Vergro
esserungs-   faktor einsetzen."
10700 END
10710 REM String formula too complex
10720 IF ERR=16 THEN PRINT "Die Zeichenk
ette in Zeile";ERL:PRINT "ist ";:A$="zu
komplex":A=0:GOSUB 12060:PRINT ". Bitte
nicht zu sehr" ELSE GOTO 10760
10730 PRINT "den String kuenstlich verko
mplizieren   durch Eingaben von Stringma
nipulations- funktionen wie 'LEFT$', 'RI
GHT$', 'MID$'oder 'INSTR'.
10740 END
10750 REM Can't continue
10760 IF ERR=17 THEN A$="CONT":A=0:GOSUB
 12060:PRINT "inue ist zwar eine tolle F
unktion," ELSE GOTO 10810
10770 PRINT "die eine Fortsetzung nach '
END', 'STOP' oder Programmunterbrechung
durch Druek- ken der <CTRL STOP>-Tasten
fast immer   ermoeglicht, aber bitte VOR
SICHT!"
10780 PRINT "In diesem Fall ist seit der
 Programmun- terbrechung zu viel gescheh
en, als dass der Computer sich noch dara
n erinnert,  in welcher Zeile er mit dem
 Programmab- lauf fortfahren soll."
10790 END
```

```
10800 REM Undefined user function
10810 IF ERR=18 THEN A$="FN":A=0:GOSUB 1
2060:PRINT " ist zwar ein toller Befehl,
 um mathe-"; ELSE GOTO 10870
10820 PRINT "matische Funktionen, die du
rch 'DEF FN' an anderer Stelle im Progra
mm definiert wurden, mit beliebigen Vari
ablen auszu- rechnen. Aber in diesem Fal
l wurde in   Zeile";ERL;"mit FN eine Fun
ktion"
10830 PRINT "aufgerufen, die vorher uebe
rhaupt       nicht so definiert worden i
st.":PRINT:PRINT "Ein Hinweis der Uebers
ichtlichkeit      halber: Definitionen s
ollten immer zu   Beginn des Programms i
m sogenannten De-"
10840 PRINT "klarationsteil festgelegt w
erden."
10850 END
10860 REM No RESUME
10870 IF ERR=21 THEN A$="RESUME":A=0:GOS
UB 12060:PRINT " sollte bei einer Interr
uptsteu-" ELSE GOTO 10920
10880 PRINT "erung mit 'ON ERROR GOTO' v
erwendet wer-den, um in den aktuellen Pr
ogrammablauf an der richtigen Stelle wie
der ein-      schwenken zu koennen. Sie k
oennen ein-   fach RESUME im Programm ein
geben, so er-folgt die Programmweiterfue
hrung an der"
10890 PRINT "Stelle, wo der Fehler aufge
treten ist.  Sie koennen auch RESUME Zei
lennummer     eingeben, um an einer gewue
nschten Pro- grammzeile fortzufahren."
10900 END
10910 REM RESUME without error
10920 IF ERR=22 THEN PRINT "Der Befehl "
;:A$="RESUME":A=0:GOSUB 12060:PRINT " is
t in Zeile" ELSE GOTO 10970
```

```
10930 PRINT ERL;"aufgetreten, ohne dass
eine Feh-":PRINT "lerroutine (ausgeloest
 durch den Befehl 'ON ERROR GOTO') im Pr
ogramm aufgerufen wurde. Die Loesung des
 anscheinend      ueberfluessigen 'RESUM
E' koennte daraus resultieren, dass das
eigentliche"
10940 PRINT "Programm nicht mit 'END' ab
geschlossen  wurde und somit nun in die
Fehlerbe-    handlungsroutine reinrutsch
t."
10950 END
10960 REM Direct statement in file
10970 IF ERR=57 THEN PRINT "Waehrend der
 Kassettenladung des Pro-   gramms wurde
 ein Befehl gefunden (";:A$="DIRECTSTATE
MENT":A=0:GOSUB 12060:PRINT "), der ohne
 Zeilennummer" ELSE GOTO 11010
10980 PRINT "im Programm auftaucht. Das
darf  nicht  so sein!"
10990 END
11000 REM Missing operand
11010 IF ERR=24 THEN PRINT "Manche BASIC
-Befehle ergeben ohne Ver-  wendung eine
r Mindestanzahl von ":A$="OPERANDEN":A=0
:GOSUB 12060:PRINT " keinen Sinn. Z.B. d
er Befehl" ELSE GOTO 11030
11020 PRINT "'SOUND' muss mindestens aus
 der Angabe  des gewaehlten Tonkanals un
d der Ton-   frequenz bestehen.":PRINT "
Z.B. 'SOUND 1,100' und nicht 'SOUND'.":E
ND
11030 REM Line buffer overflow
11040 IF ERR=25 THEN PRINT "Die Programm
zeile ist zu lang (in eng-  lisch: ";:A$
="buffer overflow":GOSUB 12060:PRINT ");
 sie darf aus" ELSE GOTO 11080
```

```
11050 PRINT "hoechstens bis zu 255 Zeich
en bestehen.":PRINT "Bitte korrigieren z
.B. durch eine Auf-  teilung dieser Zeil
e in mehrere kleine  Zeilen.":END
11060 END
11070 REM Input past end
11080 IF ERR=55 THEN GOTO 11090 ELSE GOT
O 11160
11090 PRINT "Es wurde irrtuemlicherweise
 in Zei-     le";ERL;"versucht, eine Dat
ei laenger"
11100 PRINT "in den Computer einzuladen,
 als sie     wahrlich lang ist.":PRINT "
Um diesen Fehler zu umgehen, gibt es
folgende Ratschlaege:"
11110 PRINT "Vor dem Einlesen von Daten
von der Kas- sette in einer Zeile nachfr
agen:        'IF EOF(...) THEN CLOSE:GOT
O ...'"
11120 PRINT "d.h. wenn EOF erreicht wird
, wird die   Datei ordnungsgemaess gesch
lossen und   der Programmablauf an ander
er Stelle    fortgesetzt. Andere Moeglic
hkeit: Die   Anzahl der speichernden Fil
es zuerst in der Datei ablegen. Beim Lad
en zuerst"
11130 PRINT "diese Zahl einlesen und die
selbige als  Zaehler fuer eine 'FOR...NE
XT'-Schleife verwenden."
11140 END
11150 REM Bad file name
11160 IF ERR=56 THEN PRINT "Die Art des
gewaehlten Files (";:A$="FILE":A=0:GOSUB
 12060:PRINT:A$="NAME":A=0:GOSUB 12060:P
RINT ") ist nicht korrekt." ELSE GOTO 11
200
```

```
11170 PRINT "Das Einlesen von Dateien mi
t dem Befehl 'OPEN...INPUT' ist nur bei
ASCII-Dateienmoeglich. 'CLOAD' hingegen
laedt nur    Programme, die mit 'CSAVE'
direkt abge- speichert wurden (hier wurd
e naemlich   nicht Buchstabe fuer Buchst
abe,"
11180 PRINT "sondern BASICwort(=Token) f
uer BASIC-   wort verschluesselt und som
it auch ver- kuerzt abgespeichert)."
11190 END
11200 REM File already open
11210 IF ERR=54 THEN PRINT "Eine Datei,
die mit dem Befehl ":A$="OPEN":A=0:GOSUB
 12060 ELSE GOTO 11260
11220 PRINT " bereits geoeffnet war, dar
f in":PRINT "Zeile";ERL; "nicht noch ein
mal mit OPEN"
11230 PRINT "FOR INPUT oder OPEN FOR OUT
PUT geoeff-  net werden. Das ist zudem s
innlos. Viel-leicht liegt der offensicht
liche Fehler darin begruendet, dass die
Datei in ei- nem anderen Programmabschni
tt oder nach dem Neustart des Programms
zwar geoeff-"
11240 PRINT "net, jedoch nicht wieder ge
schlossen    ('CLOSE') wurde."
11250 END
11260 REM Device I/O error
11270 IF ERR=19 THEN PRINT "Sie haben di
e Ein-Ausgabe (engl.:";:A$="I/O":A=0:GOS
UB 12060:PRINT ")":PRINT "eines angeschl
ossenen Geraets unter-    brochen. Das s
ollte nicht sein, denn    die durchgefue
hrte Operation laesst sich"; ELSE GOTO 1
1290
```

```
11280 PRINT "nicht fortsetzen.":PRINT "G
efaehrlich wird solch eine Unterbre-  c
hung vor allem bei der Datenabspeiche- r
ung. Schliessen Sie in jedem Fall of-  f
engebliebene Datenfiles mit 'CLOSE'.":EN
D
11290 REM Verify error
11300 IF ERR=20 THEN PRINT "Sie haben Ih
ren MSX-Computer aufgefor-  dert, ein ab
gespeichertes Programm mit  dem Speicher
inhalt zu vergleichen. Sie  haben zu die
sem Ueberpruefen (";:A$="VERIFY":A=0:GOS
UB 12060:PRINT ")" ELSE GOTO 11320
11310 PRINT "den Befehl 'CLOAD?' verwend
et. Nun ist  ein 'Error' aufgetreten. Sp
eichern Sie  den Speicherinhalt noch ein
mal ab, viel-leicht auf einem anderen To
nband?!":END
11320 REM FIELD overflow
11330 IF ERR=50 THEN PRINT "Sie haben be
i der Diskettenabspeiche-  rung eines r
elativen Files nicht daran  gedacht, das
s das vorher festgelgte For-mat (";:A$="
FIELD":A=0:GOSUB 12060:PRINT ") mit dem
Datenformat" ELSE GOTO 11360
11340 PRINT "uebereinstimmen muss."
11350 END
11360 REM Internal error
11370 IF ERR=51 THEN PRINT "Dieser 'Erro
r' beruht darauf, dass      im Inneren d
es Computers etwas nicht    so funktioni
ert hat, wie es eigentlich  vorgesehen w
ar. Dies kann Ihnen z.B.    beim unueber
legten Gebrauch des Befelhs" ELSE GOTO 1
14000
11380 PRINT "'OUT' sehr leicht passieren
.           Ein interner Fehler - engli
sch: ":A$="INTERNAL ERROR":A=0:GOSUB 120
60:PRINT ". Schauen Sie mal im Handbuch
nach."
```

```
11390 END
11400 REM Bad file number
11410 IF ERR=52 THEN PRINT "Sie duerfen
nur Dateien benutzen, die   Sie ueber ei
nen geoeffneten Datenfile   auch empfang
en kennen. Also: erst       'OPEN...FOR
INPUT/OUTPUT' und am Ende   der Bearbeit
ung nicht 'CLOSE' vergessen." ELSE GOTO
11440
11420 PRINT "Ausserdem muessen Sie achtg
eben, dass   Sie nicht mehr Dateien oeff
nen koennen, wie Sie zuvor mit dem Befeh
l 'MAXFILES' festgelegt haben."
11430 END
11440 REM File not found
11450 IF ERR=53 THEN PRINT "Dieser 'Erro
r' tritt nur bei Disketten- bearbeitung
auf. Sie wollen eine Datei  oder ein Pro
gramm in den Computerspei-  cher laden,
das sich ueberhaupt nicht   auf der eing
elegten Diskette befindet." ELSE GOTO 11
480
11460 PRINT "Vielleicht schauen Sie sich
 zuerst das  Inhaltsverzeichnis auf der
Diskette mit 'FILES' bzw. bei einer zwei
ten Disket-  tenstation mit 'FILES2' an.
"
11470 END
11480 REM Sequential I/O only
11490 IF ERR=58 THEN PRINT "Sie muessen
zwischen ";:A$="sequentiellen":A=0:GOSUB
 12060:PRINT:PRINT "und relativen Dateie
n unterscheiden.    Sequentiell heisst:
alle Daten hinerein-ander abspeichern."
ELSE GOTO 11520
11500 PRINT "Relativ heisst: Direkter Da
tenzugriff   mit einem festgelgten Daten
format!"
11510 END
```

```
11520 REM File not OPEN
11530 IF ERR=59 THEN PRINT "Sie muessen
eine Datei erst mit dem    Befehl ";:A$
="OPEN":A=0:GOSUB 12060:PRINT " oeffnen,
 bevor Sie mit" ELSE GOTO 11560
11540 PRINT "den Befehlen wie 'PRINT #'
und 'INPUT #'darauf zugreifen koennen."
11550 END
11560 PRINT "Unbekannte Fehlermeldung":E
ND
11570 REM Die Errormeldungen mit ihrer
          englischen Kurzerklaerung
11580 IF ERR=1 THEN PRINT "NEXT without
FOR"
11590 IF ERR=2 THEN PRINT "Syntax error"
11600 IF ERR=3 THEN PRINT "RETURN withou
t GOSUB"
11610 IF ERR=4 THEN PRINT "Out of DATA"
11620 IF ERR=5 THEN PRINT "Illegal funct
ion call"
11630 IF ERR=6 THEN PRINT "Overflow"
11640 IF ERR=7 THEN PRINT "Out of memory
"
11650 IF ERR=8 THEN PRINT "Undefined lin
e number"
11660 IF ERR=9 THEN PRINT "Subscript out
 of range"
11670 IF ERR=10 THEN PRINT "Redimensione
d array"
11680 IF ERR=11 THEN PRINT "Division by
zero"
11690 IF ERR=12 THEN PRINT "Illegal dire
ct"
11700 IF ERR=13 THEN PRINT "Type mismatc
h"
11710 IF ERR=14 THEN PRINT "Out of strin
g space"
11720 IF ERR=15 THEN PRINT "String too l
ong"
```

```
11730 IF ERR=16 THEN PRINT "String formu
la too complex"
11740 IF ERR=17 THEN PRINT "Can't contin
ue"
11750 IF ERR=18 THEN PRINT "Undefined us
er function"
11760 IF ERR=19 THEN PRINT "Device I/O e
rror"
11770 IF ERR=20 THEN PRINT "Verify error
"
11780 IF ERR=21 THEN PRINT "No RESUME"
11790 IF ERR=22 THEN PRINT "RESUME witho
ut error"
11800 IF ERR=24 THEN PRINT "Missing oper
and"
11810 IF ERR=25 THEN PRINT "Line buffer
overflow"
11820 IF ERR=50 THEN PRINT "Field overfl
ow"
11830 IF ERR=51 THEN PRINT "Internal err
or"
11840 IF ERR=52 THEN PRINT "Bad file num
ber"
11850 IF ERR=53 THEN PRINT "File not fou
nd"
11860 IF ERR=54 THEN PRINT "File already
 open"
11870 IF ERR=55 THEN PRINT "Input past e
nd"
11880 IF ERR=56 THEN PRINT "Bad file nam
e"
11890 IF ERR=57 THEN PRINT "Direct state
ment in file"
11900 IF ERR=58 THEN PRINT "Sequential I
/O only"
11910 IF ERR=59 THEN PRINT "File not OPE
N"
11920 PRINT:PRINT
11930 RETURN
```

```
11940 REM Umwandlung >CHR$(127)->invers
11950 FOR M=2304 TO 3055
11960 A$=BIN$(VPEEK(M))
11970 A$=STRING$(8-LEN(A$),"0")+A$
11980 FOR N=1 TO 8
11990 IF MID$(A$,N,1)="1" THEN MID$(A$,N
,1)="0" ELSE MID$(A$,N,1)="1"
12000 NEXT N
12010 A$="&B"+A$
12020 VPOKE M+768,VAL(A$)
12030 A$=""
12040 NEXT M
12050 RETURN
12060 REM Umwandlung in Inversschrift
12070 REM a=0 -> alles
12080 REM a=1 -> bis auf Leerzeichen
12090 REM a=2 -> nur Kleinbuchstaben
12100 REM a=3 -> nur Grossbuchstaben
12110 REM a=4 -> nur Zahlen
12120 FOR PQ=1 TO LEN(A$)
12130 IF A=0 THEN GOTO 12180
12140 IF A=1 AND MID$(A$,PQ,1)=" " THEN
GOTO 12190
12150 IF A=2 AND (MID$(A$,PQ,1)<CHR$(97)
 OR MID$(A$,PQ,1)>CHR$(125)) THEN GOTO 1
2190
12160 IF A=3 AND (MID$(A$,PQ,1)<CHR$(65)
 OR MID$(A$,PQ,1)>CHR$(93)) THEN GOTO 12
190
12170 IF A=4 AND (MID$(A$,PQ,1)>CHR$(57)
 OR MID$(A$,PQ,1)<CHR$(48)) THEN GOTO 12
190
12180 MID$(A$,PQ,1)=CHR$(ASC(MID$(A$,PQ,
1))+96)
12190 NEXT PQ
12200 PRINT A$;
12210 RETURN
```

Variablenreferenzliste

Im Programm 'Speicher 4' haben wir bereits kennengelernt, wie die jeweilige Zeilennummer und der Speicherplatz der nächsten Programmzeile im Speicher abgelegt werden (Speicherstelle 32769 und 32770 weisen auf den Speicherplatz der zweiten Programmzeile hin, Speicherstelle 32771 und 32772 teilen uns mit, wie die erste Zeilennummer heißt usw.). Mit Programm 'Speicher 4' haben wir zudem eine einfache Möglichkeit an die Hand bekommen, den Speicher nach einem bestimmten Wort mit bis zu sechs Buchstaben zu durchsuchen. Jedoch so mancher unter Ihnen wird sich fragen: 'was kann ich damit sinnvoll anfangen?'. Nun ja, im Endprodukt haben wir dadurch einerseits den Speicher etwas näher kennengelernt, andererseits kann uns Programm 'Speicher 4' aber außerdem behilflich dabei sein, wenn wir gerade nach solch einer alphanumerischen Zeichenfolge suchen.

Im Programm 'Variablenreferenzliste' gehen wir mit ähnlichem Fundament an den Start, aber wieder erfolgt eine entscheidende Erweiterung: diesmal wird der Speicher nicht nur nach bestimmten Zeichenfolgen durchsucht, nein, es wird zudem eine Ordnung hergestellt.

Die gesuchten Zeichenfolgen ergeben sich aus dem Vorhandensein von Großbuchstaben oder Zahlen, die unter bestimmten Bedingungen nur als Variable und als nichts anderes gedeutet werden können:

Bedingung 1: es wurde vorher in der Zeile kein Anführungszeichen gesetzt (dann würde es sich bei dem Großbuchstaben um einen Teilstring handeln)
Bedingung 2: wir befinden uns nicht in einer 'REM'-Zeile
Bedingung 3: vor dem gefundenen Großbuchstabenrepräsentanten steht keine '255' (das würde Token bedeuten).

Da Ihr MSX-Computer nur zwei Zeichen bei Variablennamen unterscheidet, er aber die Möglichkeit bietet, auch längere Variablennamen (ohne Token) zu akzeptieren, müssen wir selbstverständlich den ganzen Variablennamen im Speicher aufsuchen. Da es verschiedene Variablentypen gibt (String, ganze Zahl ...) muß

auch ein dementsprechendes Zeichen ('!', '$' ...) mitgelesen und bei unserer Anzeige mit aufgelistet werden.

Was kann unser 'Variablenreferenzlistenprogramm' nun alles von dem und noch darüberhinaus?

1) Es werden Variablen im Programmspeicher gesucht (Start des Programms 'Variablenrefernzliste' mit 'RUN 10000') - bis Zeile 9999 kann Ihr Programm lang sein.

2) Steht der gefundene Variablenname alphabetisch an erster Stelle (von allem in Ihrem Programm vorkommenden Variablen), so wird im Speicher = Programm weitergesucht, in welchen Programmzeilen dieser gleiche Name zudem auftaucht. Auch dies wird am Bildschirm angezeigt. So wird mit der Zeit eine alphabetische Variablenliste auf dem Bildschirm erzeugt, die uns bei unserem 'BASIC-Programm-Debugging' (Fehlersuche) erheblich weiterhelfen kann.

Probieren Sie das Programm doch gleich aus, indem Sie in Zeile 10 bis 60 folgende Variablenzuweisungen eintragen:

```
'10 a$="MSX-Computer"
 20 b$="Computer"
 30 MS=64
 40 BE!=100
 50 BA%=200
 60 D#=4000'
```

Nun starten Sie das 'Variablenreferenzlistenprogramm' mit 'RUN 10000' und nach kurzer Zeit werden nicht nur die verwendeten Variablen sowie ihre Anzahl und die Häufigkeit Ihres Auftretens, sondern auch die entsprechenden Zeilennummern am Bildschirm aufgeführt.

Ist das Programm durchforstet, so schreibt Ihr MSX-Computer den verbrauchten Speicherplatz auf den Bildschirm (Angabe für Programm = ohne Referenzlistenprogramm) sowie die Anzahl und Häufigkeit der verschiedenen Variablennamen.

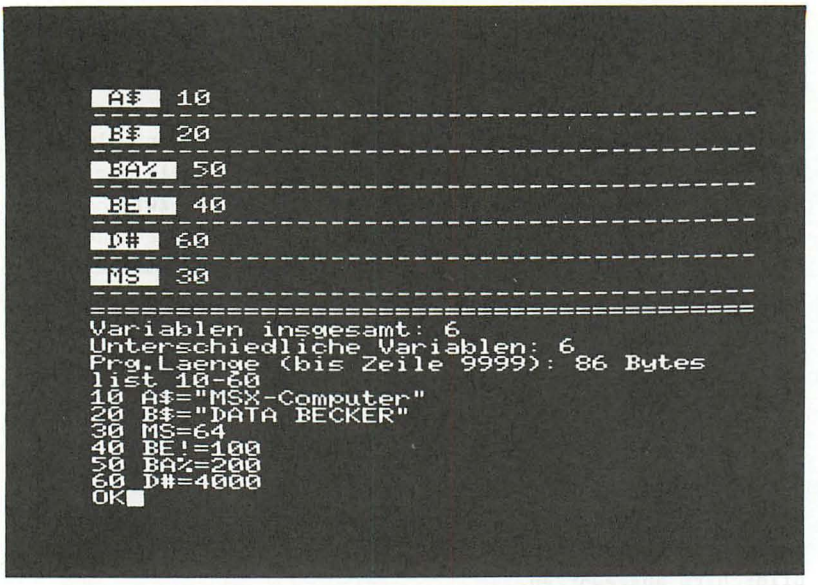
Noch ein paar Hinweise

Am Programmbeginn sehen Sie das suksessive Verändern eines Teils des vorliegenden Zeichensatzes in ein entsprechend inverses Schriftbild. Diese langwierige Prozedur brauchen Sie nur einmal über sich ergehen lassen, es sei denn, Sie ändern zwischenzeitlich den Screen. Wird der 'SCREEN'-Befehl nicht angewendet, so können Sie laut Anweisung nach dem ersten Programmstart die ersten Zeilen löschen.

Je länger Ihr Programm ist, desto ausführlicher und damit aufwendiger muß Ihr MSX-Computer seine Speicherdurchsuchung durchführen. Also: Bitte ein bißchen Geduld.

Die Variablennamen werden erst einmal alphabetisch sortiert, dann ihrer Zeilennummer - bzw. auch mehreren Zeilennummern - zugeordnet und schließlich im Rahmen einer Referenzliste auf den Bildschirm ausgeworfen.

Die angezeigte 'Programmlänge' bezieht sich nur auf das Programm in den Zeilen bis 10000, denn die 'Referenzliste' gehört ja schließlich nicht zum eigentlichen Hauptprogramm dazu; sie fungiert lediglich als Utility.


A$ 10
B$ 20
BA% 50
BE! 40
D# 60
MS 30
Variablen insgesamt: 6
Unterschiedliche Variablen: 6
Prg.Laenge (bis Zeile 9999): 86 Bytes
list 10-60
10 A$="MSX-Computer"
20 B$="DATA BECKER"
30 MS=64
40 BE!=100
50 BA%=200
60 D#=4000
OK

```
10000 REM Variablenreferenzliste
10010 REM MSX Programmsammlung
10020 REM Copyright 1984 DATA BECKER &
                         Rainer Lueers
10030 REM Vorbedingungen treffen
10040 SCREEN 0
10050 WIDTH 40
10060 COLOR 15,1
10070 KEY OFF
10080 REM Einladen des inversen
          Zeichensatzes
10090 PRINT "Einen Augenblick bitte"
10100 FOR N=32 TO 255
10110 PRINT CHR$(N);
10120 NEXT N
10130 GOSUB 10640
10140 PRINT
10150 PRINT "Wenn der SCREEN nicht geaen
dert wird,   bitte Zeile 10040 bis 10160
 loeschen."
10160 STOP
10170 REM Beginn des eigentlichen
          ========================
          Hauptprogramms
          ==============
10180 CLS
10190 DEFINT A-Z
10200 CLEAR 5000
10210 REM Dimensionierung fuer bis zu
          1000 Variablennamen und
          die dazugehoerende Zeilen-
          nummer
10220 DIM A$(1000),A(1000)
10230 REM BASICprogrammstart
10240 Z=32769!
10250 V=1
10260 REM Verweis auf den Beginn der
          naechsten Programmzeile im
          Speicher
10270 A=PEEK(Z)+256*PEEK(Z+1)
```

```
10280 REM Zeilennummer
10290 B=PEEK(Z+2)+256*PEEK(Z+3)
10300 REM Es wird nur bis Zeile
          9999 im Speicher nach
          Variablen gesucht
10310 IF B=10000 THEN Q1=Z-32769!:GOSUB
10490:GOSUB 10920:PRINT "Variablen insge
samt:";V-1:PRINT "Unterschiedliche Varia
blen:";QQ:PRINT "Prg.Laenge (bis Zeile 9
999):";Q1;"Bytes":END
10320 FOR N=Z+4 TO A-1
10330 REM Kriterien zur Suche im Spei-
          ============================
          cher Variable ja/nein:
          ======================
10340 REM 1) zwischen 65 und 90 ->
             Grossbuchstabe
          2) im vorherigen Speicher-
             platz keine 255 -> Token
10350 REM 3) nicht der Inhalt einer
             PRINT "-Zeile (kein An-
             fuehrunszeichen=34 gesetzt
             Kennzeichnung Varible 'ME'
10360 REM 4) keine Remark oder REM-
             Zeile (REM oder ')
10370 IF PEEK(N)<91 AND PEEK(N)>64 AND P
EEK(N-1)<>255 AND ME<>1 AND M1<>1 THEN G
OTO 10460
10380 IF PEEK(N)=34 THEN IF ME=0 THEN ME
=1 ELSE ME=0
10390 IF PEEK(N)=230 OR PEEK(N)=143 THEN
 M1=1
10400 NEXT N
10410 ME=0
10420 M1=0
10430 Z=A
10440 GOTO 10270
```

```
10450 REM Zusammensetzen des
          Variablennamens in A$
10460 A$=CHR$(PEEK(N))
10470 IF (PEEK(N+1)<91 AND PEEK(N+1)>64)
 OR (PEEK(N+1)=33) OR (PEEK(N+1)>34 AND
PEEK(N+1)<38) OR (PEEK(N+1)<58 AND PEEK(
N+1)>47) OR (PEEK(N+1)=40) THEN A$=A$+CH
R$(PEEK(N+1)):N=N+1:GOTO 10470 ELSE A(V)
=B:A$(V)=A$:V=V+1:GOTO 10400
10480 REM Ordnen der Variablennamen
          ==========================
10490 A$="z"
10500 FOR M=1 TO V-1
10510 IF A$(M)<A$ THEN A$=A$(M):A=M
10520 NEXT M
10530 A$(A)="z"
10540 IF A$="z" THEN PRINTSTRING$(40,"="
);:RETURN ELSE Q$=" "+A$+" ":GOSUB 10760
:PRINT A(A);
10550 REM Zuweisen von einer oder
          ========================
          mehreren Zeilennummern, wo
          ==========================
          die Variable auftritt
          =====================
10560 FOR O=1 TO V-1
10570 IF A$(O)=A$ THEN PRINT A(O);:A$(O)
="z"
10580 NEXT O
10590 REM Beendigung einer Variablen-
          ausgabe auf dem Bildschirm
10600 PRINT
10610 PRINT STRING$(40,"-");
10620 QQ=QQ+1
10630 GOTO 10490
10640 REM Umwandlung >CHR$(127)->invers
          ============================
10650 FOR M=2304 TO 2775
10660 A$=BIN$(VPEEK(M))
10670 A$=STRING$(8-LEN(A$),"0")+A$
```

```
10680 FOR N=1 TO 8
10690 IF MID$(A$,N,1)="1" THEN MID$(A$,N
,1)="0" ELSE MID$(A$,N,1)="1"
10700 NEXT N
10710 A$="&B"+A$
10720 VPOKE M+768,VAL(A$)
10730 A$=""
10740 NEXT M
10750 RETURN
10760 REM Umwandlung in Inversschrift
10770 REM k=0 -> alles
10780 REM k=1 -> bis auf Leerzeichen
10790 REM k=2 -> nur Kleinbuchstaben
10800 REM k=3 -> nur Grossbuchstaben
10810 REM k=4 -> nur Zahlen
10820 FOR PQ=1 TO LEN(Q$)
10830 IF K=0 THEN GOTO 10880
10840 IF K=1 AND MID$(Q$,PQ,1)=" " THEN
GOTO 10890
10850 IF K=2 AND (MID$(Q$,PQ,1)<CHR$(97)
 OR MID$(Q$,PQ,1)>CHR$(125)) THEN GOTO 1
0890
10860 IF K=3 AND (MID$(Q$,PQ,1)<CHR$(65)
 OR MID$(Q$,PQ,1)>CHR$(93)) THEN GOTO 10
890
10870 IF K=4 AND (MID$(Q$,PQ,1)>CHR$(57)
 OR MID$(Q$,PQ,1)<CHR$(48)) THEN GOTO 10
890
10880 MID$(Q$,PQ,1)=CHR$(ASC(MID$(Q$,PQ,
1))+96)
10890 NEXT PQ
10900 PRINT Q$;
10910 RETURN
10920 IF V-1=0 THEN QQ=0
10930 RETURN
```

Kartei für Schallplatten

Dieses Programm trägt seinen Namen nur dann zu recht, wenn Sie es Wort für Wort bzw. Befehl für Befehl so eintippen, wie Sie es in unserer Programmsammlung vorfinden.

Da es sich aber im Endprodukt um eine gar nicht so unluxuriöse kleine Datenbank handelt, ist es auch möglich, die entsprechenden PRINT-Befehle z.B. 'PRINT "Platten eingeben"' in gewünschte Befehle mit nachfolgendem Text umzuwandeln z.B. 'PRINT "Adressen eingeben"'.

So hätten wir nach ein paar Abänderungen im Programm eine Adreßdatei, die wir natürlich auch später so abspeichern können bzw. so abspeichern sollten.

Jedoch zeigt das Programm 'Kartei für Schallplatten' einige Beschränkungen, die sich erst dann sinnvoll beseitigen ließen, wenn wir nicht mit Kassette sondern mit Diskette - und dort mit wahlfreiem Zugriff - arbeiten könnten.

Die Grenzen des Programms zeigen sich nämlich deutlich bei dem zur freien Verfügung stehenden Speicherplatz. Wollen wir eine Karteikarte vollständig nutzen (drei mal 40 Zeichen), so werden dazu bereits 120 Speicherplätze benötigt. Bei 100 Karteikarten sind dies schon 12000 Speicherplätze ... und daraus resultiert, daß wir als Obergrenze vorerst die Zahl 300 für die Menge der unterschiedlichen Karteikarten vorgegben haben.

Speichern Sie pro Karteikarte (120 Speicherplätze) nicht so viele Zeichen ab, können Sie selbstverständlich hierzu auch das Programm entsprechend ändern. In diesem Fall wird die Höchstzahlgrenze einfach größer umschrieben z.B. Zeile 100:

'IF z<1 OR z>400 ...'

Im einzelnen stellt Ihnen das Programm 'Kartei für Schallplatten' folgende Elemente zur Verfügung:

1. Sie können Karteikarten mit je drei Feldern anlegen (Plattenname, Sänger, Nummer)

2. Sie können Dateien von Kassette laden und
3. Sie können Dateien auf Kassette abspeichern

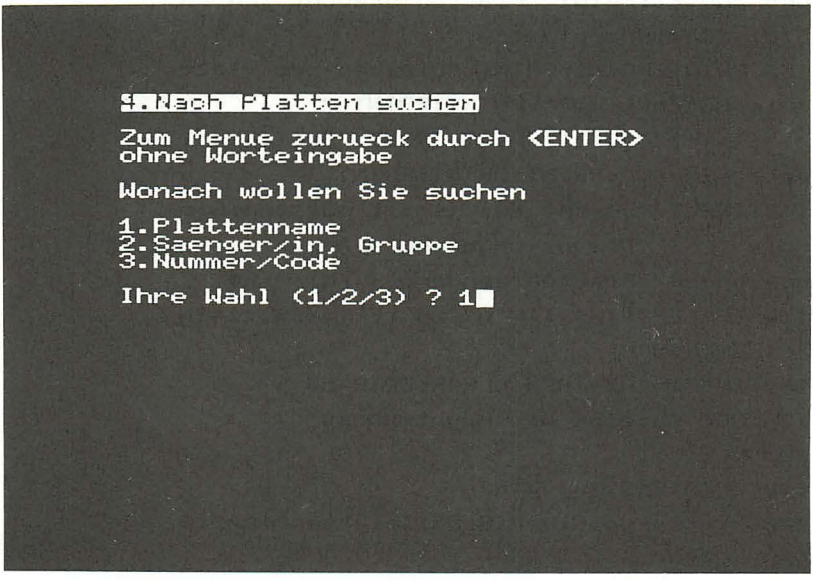
4. Sie können nach einer Platte im Speicher suchen lassen (nach Plattenname, Sänger oder Nummer); bei dieser Funktion wird Ihnen der Bildschirm mit gefundenen Daten angefüllt, bis er voll ist. Danach drücken Sie die (ENTER)-Taste und die Auflistung geht weiter. Mehrmaliges Auffinden kann möglich sein, wenn Sie mehrere Langspielplatten eines Sängers, z.B. von Herbert Grönemeyer, besitzen.

5. Sie können Ihre gesamte Plattenliste auf den Bildschirm - oder falls vorhanden auf einen angeschlossenen Drucker - ausgeben lassen (bei der Bildschirmausgabe immer nur fünf Datensätze, dann (ENTER) drücken).

6. Sie können eingegebene Daten ändern. Da bereits bei der Eigabe (1.) die Möglichkeit der nachträglichen Korrektur eingeräumt wird ('Richtig?'), wird Funktion 6. wohl kaum angewendet werden müssen, um Eingabefehler nachträglich zu beseitigen. Vielmehr ist bei dieser Funktion daran gedacht, daß Sie das Dateiprogramm z.B. - wie oben bereits vorgeschlagen - zur Verwaltung von Adressen verwenden. Wie oft kommt es da doch vor, daß jemand seinen Wohnsitz gewechselt hat ... Ganz einfach: Adreßdatei aufrufen, Punkt 6 = 'Änderungen vornehmen' aufrufen. Da sich sicherlich nicht alle Daten Ihres Bekannten von heute auf morgen verändert haben (zumindest sein Name nicht), können Sie diese Programmteile durch Drücken der (ENTER)-Taste einfach überspringen. Erst an der Stelle, wo wahrlich eine Änderung aufgetreten ist, können Sie eingeben.

Ob Sie dieses Programm nun wie ursprünglich gedacht für die Speicherung Ihres Schallplattenarchivs verwenden oder ob Sie das Programm in eine Adreßverwaltung ummodeln, auf jeden Fall wünschen wir Ihnen viel Freude dabei, der Unordung der Ordnung halber mit Ihrem MSX-Computer und unserem Programm zu Leibe zu rücken.


Kartei fuer Schallplatten
Menue
1.Platten eingeben
2.Platten laden
3.Platten speichern
4.Nach Platten suchen
5.Platten ausgeben
6.Platten aendern
Ihre Wahl (1/2/3/4/5/6) ?


4.Nach Platten suchen
Zum Menue zurueck durch <ENTER>
ohne Worteingabe
Wonach wollen Sie suchen
1.Plattenname
2.Saenger/in, Gruppe
3.Nummer/Code
Ihre Wahl (1/2/3) ? 1

```
10 REM Kartei fuer Schallplatten
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 SCREEN 0:COLOR 15,1,15:KEY OFF:CLEAR
15000
50 PRINT "Bitte ein bisschen Geduld!":GO
SUB 2370:CLS:M=1:WIDTH 40
60 REM Vorbedingungen treffen zum
       Anlegen von dimensionierten
       Variablen
70 A$="Kartei fuer Schallplatten":A=0:GO
SUB 2490
80 PRINT
90 INPUT "Wie viele LPs sollen denn gesp
eichert   werden (hoechstens 300) ";Z
100 IF Z<1 OR Z>300 THEN GOSUB 2290:GOTO
 50
110 FR=FRE(0):DIM P$(Z),S$(Z),N$(Z)
120 PRINT:PRINT "Allein die Dimensionier
ung von";Z:PRINT "Platten braucht";FR-FR
E(0);"Speicherplaetze."
130 PRINT:PRINT:PRINT "Bleiben pro Platt
e im Hoechstfall":PRINT:PRINT "
  ";INT(17000/Z):PRINT:PRINT "Speicherpl
aetze frei.":PRINT:PRINT:PRINT
140 INPUT "Reicht das (sonst muessen Sie
 die       Dimensionierung = Anzahl einz
ugebender  Platten aendern ( /N) ";F$
150 IF LEFT$(F$,1)="n" OR LEFT$(F$,1)="N
" THEN F$="N"
160 IF F$="N" THEN CLEAR:GOTO 50
170 REM Anzeige des Hauptmenues
180 CLS
190 LOCATE 5,7
200 A$="Kartei fuer Schallplatten":A=0:G
OSUB 2490
210 LOCATE 15,9
220 PRINT "Menue"
```

```
230 LOCATE 7,12
240 PRINT "1.Platten eingeben"
250 LOCATE 7,13:PRINT "2.Platten laden"
260 LOCATE 7,14:PRINT "3.Platten speiche
rn"
270 LOCATE 7,15:PRINT "4.Nach Platten su
chen"
280 LOCATE 7,16:PRINT "5.Platten ausgebe
n"
290 LOCATE 7,17:PRINT "6.Platten aendern
"
300 LOCATE 5,20
310 INPUT "Ihre Wahl (1/2/3/4/5/6) ";F$
320 IF F$<"1" OR F$>"6" THEN GOSUB 2290:
GOTO 180
330 ON VAL(F$) GOSUB 370,680,930,1210,17
40,2010
340 F$="":F1$="":F2$="":N1=0
350 GOTO 180
360 REM Unterprogramm zur Eingabe
        von Daten
370 FOR N=M TO Z
380 CLS
390 A$="1.Platten eingeben":A=0:GOSUB 24
90
400 PRINT
410 GOSUB 2350
420 PRINT
430 F1$="":PRINT "Plattenname"
440 INPUT P$(N):IF LEN(P$(N))>40 THEN P$
(N)=LEFT$(P$(N),40)
450 IF P$(N)="" THEN M=N:RETURN
460 PRINT "Saenger/in, Gruppe"
470 INPUT S$(N):IF LEN(S$(N))>40 THEN S$
(N)=LEFT$(S$(N),40)
480 PRINT "Nummer/Code"
490 INPUT N$(N):IF LEN(N$(N))>40 THEN N$
(N)=LEFT$(N$(N),40)
```

```
500 REM Erste Anzeige der eingegebenen
        Daten mit
        Korrekturmoeglichkeit
510 PRINT
520 PRINT "Noch einmal die eingegebenen
Daten:"
530 PRINT "Satznummer";N;"von bis zu";Z
540 PRINT
550 A$="Plattenname":A=0:GOSUB 2490
560 PRINT P$(N)
570 A$="Saenger/in, Gruppe:":A=0:GOSUB 2
490
580 PRINT S$(N)
590 A$="Nummer/Code":A=0:GOSUB 2490
600 PRINT N$(N)
610 PRINT
620 INPUT "Richtig ( /N) ";F1$
630 IF LEFT$(F1$,1)="N" OR LEFT$(F1$,1)=
"n" THEN F1$="N"
640 IF F1$="N" THEN PRINT:GOTO 430
650 NEXT N
660 RETURN
670 REM Unterprogramm zum Laden
        von Dateien von Kassette
680 CLS
690 A$="Platten laden":A=0:GOSUB 2490
700 PRINT
710 GOSUB 2350
720 PRINT
730 PRINT "Sie loeschen durch die Dateie
ingabe      alle bereits vorhandenen Date
n           im Speicher!"
740 PRINT
750 INPUT "Wollen Sie Daten laden (J/ )
";F1$
760 IF LEFT$(F1$,1)="j" OR LEFT$(F1$,1)=
"J" THEN F1$="J"
770 IF F1$="J" THEN GOTO 780 ELSE RETURN
```

```
780 PRINT
790 PRINT "Dateiname"
800 INPUT DN$
810 OPEN DN$ FOR INPUT AS#1
820 INPUT #1,M
830 PRINT
840 PRINT "Die einzuladende Datei ";DN$
850 PRINT "besteht aus";M-1;"Saetzen"
860 FOR N=1 TO M-1
870 INPUT #1,P$(N),S$(N),N$(N)
880 PRINT N:PRINT P$(N):PRINT S$(N):PRIN
T N$(N)
890 NEXT N
900 CLOSE #1
910 RETURN
920 REM Unterprogramm zum Speichern
        von Dateien auf Kassette
930 CLS
940 A$="3.Platten speichern":A=0:GOSUB 2
490
950 PRINT
960 GOSUB 2350
970 PRINT
980 INPUT "Wollen Sie Daten speichern (J
/ ) ";F1$
990 IF LEFT$(F1$,1)="J" OR LEFT$(F1$,1)=
"j" THEN F1$="J"
1000 IF F1$="J" THEN GOTO 1010 ELSE RETU
RN
1010 PRINT
1020 IF DN$<>"" THEN PRINT "Soll unter d
em Dateinamen ";DN$:INPUT "abgespeichert
 werden (J/ ) ";F2$:IF LEFT$(F2$,1)="J"
OR LEFT$(F2$,1)="j" THEN F2$="J"
1030 IF F2$="J" THEN GOTO 1060
1040 PRINT "Dateiname ";
1050 INPUT DN$
1060 OPEN DN$ FOR OUTPUT AS#1
1070 PRINT #1,M
1080 PRINT
```

```
1090 PRINT "Die zu speichernde Datei ";D
N$:PRINT "besteht aus";M-1;"Saetzen"
1100 FOR N=1 TO M-1
1110 PRINT #1,P$(N),S$(N),N$(N)
1120 PRINT N
1130 PRINT P$(N)
1140 PRINT S$(N)
1150 PRINT N$(N)
1160 PRINT
1170 NEXT N
1180 CLOSE #1
1190 RETURN
1200 REM Unterprogramm zum Suchen nach
        Platten, Saengern oder
        Nummern
1210 CLS
1220 A$="4.Nach Platten suchen":A=0:GOSU
B 2490
1230 PRINT
1240 GOSUB 2350
1250 PRINT
1260 REM Vorgabe eines Untermenues:
        Suche nach:
        1) Plattenname
        2) Saenger/in, Gruppe
        3) Nummer/Code
1270 PRINT "Wonach wollen Sie suchen"
1280 PRINT
1290 PRINT "1.Plattenname"
1300 PRINT "2.Saenger/in, Gruppe"
1310 PRINT "3.Nummer/Code"
1320 PRINT
1330 INPUT "Ihre Wahl (1/2/3) ";F1$
1340 IF F1$<"1" OR F1$>"3" THEN RETURN
1350 ON VAL(F1$) GOSUB 1380,1500,1610
1360 F1$="":F2$="":N1=0:GOTO 1210
1370 REM Suche nach Plattenname
1380 CLS
1390 PRINT "Plattenname"
```

```
1400 INPUT P$:IF LEN(P$)>40 THEN P$=LEFT
$(P$,40)
1410 IF P$="" THEN RETURN
1420 FOR N=1 TO M-1
1430 IF P$=P$(N) THEN PRINT N:PRINT P$(N
):PRINT S$(N):PRINT N$(N):N1=N1+1
1440 IF N1=5 THEN N1=0:GOSUB 2320
1450 NEXT N
1460 PRINT
1470 GOSUB 2320
1480 RETURN
1490 REM Suche nach Saenger/in, Gruppe
1500 CLS
1510 PRINT "Saenger/in, Gruppe"
1520 INPUT S$:IF LEN(S$)>40 THEN S$=LEFT
$(S$,40)
1530 IF S$="" THEN RETURN
1540 FOR N=1 TO M-1
1550 IF S$=S$(N) THEN PRINT N:PRINT P$(N
):PRINT S$(N):PRINT N$(N):N1=N1+1
1560 IF N1=5 THEN N1=0:GOSUB 2320
1570 NEXT N
1580 PRINT
1590 GOSUB 2320
1600 RETURN
1610 CLS
1620 REM Suche nach Nummer/Code
1630 PRINT "Nummer/Code"
1640 INPUT N$
1650 IF N$="" THEN RETURN
1660 FOR N=1 TO M-1
1670 IF N$=N$(N) THEN PRINT N:PRINT P$(N
):PRINT S$(N):PRINT N$(N):N1=N1+1
1680 IF N1=5 THEN N1=0:GOSUB 2320
1690 NEXT N
1700 PRINT
1710 GOSUB 2320
1720 RETURN
```

```
1730 REM Unterprogramm zur Ausgabe der
         gespeicherten Daten auf
         Bildschirm oder Drucker
1740 CLS
1750 A$="5.Platten ausgeben":A=0:GOSUB 2
490
1760 PRINT
1770 GOSUB 2350
1780 PRINT
1790 PRINT "Auf Bildschirm oder Drucker
ausgeben?"
1800 INPUT "(B/D) ";F2$
1810 IF F2$="" THEN RETURN
1820 IF LEFT$(F2$,1)="b" OR LEFT$(F2$,1)
="B" THEN F2$="B" ELSE IF LEFT$(F2$,1)="
d" OR LEFT$(F2$,1)="D" THEN F2$="D"
1830 IF F2$="D" THEN GOTO 1940
1840 REM Ausgabe der Daten auf Monitor
1850 CLS:FOR N=1 TO M-1
1860 N1=N1+1
1870 IF N1=5 THEN N1=0:GOSUB 2320:CLS
1880 PRINT N:PRINT P$(N):PRINT S$(N):PRI
NT N$(N)
1890 NEXT N
1900 PRINT
1910 GOSUB 2320
1920 RETURN
1930 REM Ausgabe der Daten auf Drucker
1940 FOR N=1 TO M-1
1950 LPRINT N:LPRINT P$(N):LPRINT S$(N):
LPRINT N$(N):LPRINT
1960 NEXT N
1970 PRINT
1980 GOSUB 2320
1990 RETURN
2000 REM Unterprogramm zur Aenderung
         von Daten im Speicher
2010 CLS
2020 A$="6.Platten aendern":A=0:GOSUB 24
90
```

```
2030 PRINT
2040 GOSUB 2350
2050 PRINT
2060 INPUT "Welche Plattennummer ";N1
2070 IF N1>M-1 OR N1<0 THEN GOSUB 2290:G
OTO 2060
2080 IF N1=0 THEN RETURN
2090 PRINT
2100 PRINT N1:PRINT P$(N1):PRINT S$(N1):
PRINT N$(N1)
2110 PRINT
2120 PRINT "Was hat sich geaendert (Nein
=<ENTER>)?"
2130 PRINT P$(N1)
2140 INPUT D$:IF D$<>"" THEN P$(N1)=D$
2150 PRINT S$(N1)
2160 INPUT D$:IF D$<>"" THEN S$(N1)=D$
2170 PRINT N$(N1)
2180 INPUT D$:IF D$<>"" THEN N$(N1)=D$
2190 CLS
2200 PRINT "Die Eingabe zu Plattennummer
";N1;":"
2210 PRINT P$(N1)
2220 PRINT S$(N1)
2230 PRINT N$(N1)
2240 PRINT
2250 GOSUB 2320
2260 N1=0
2270 GOTO 2010
2280 REM Unterprogramm zur Anzeige bei
         falscher Eingabe
2290 PRINT:PRINT TAB(11);:A$="Falsche Ei
ngabe!":A=0:GOSUB 2490:GOSUB 2320:RETURN
2300 A$=INKEY$:IF A$="" THEN GOTO 2300 E
LSE RETURN
```

```
2310 REM Unterprogramm zum Einfrieren
         der Bildschirmanzeige
2320 PRINT:PRINT TAB(6) "<Bitte eine Tas
te druecken>"
2330 A$=INKEY$:IF A$="" THEN GOTO 2330 E
LSE RETURN
2340 REM Allgemeines Unterprogramm
2350 PRINT "Zum Menue zurueck durch <ENT
ER>          ohne Worteingabe"
2360 RETURN
2370 REM Umwandlung >CHR$(127)->invers
2380 FOR M=2304 TO 3055
2390 A$=BIN$(VPEEK(M))
2400 A$=STRING$(8-LEN(A$),"0")+A$
2410 FOR N=1 TO 8
2420 IF MID$(A$,N,1)="1" THEN MID$(A$,N,
1)="0" ELSE MID$(A$,N,1)="1"
2430 NEXT N
2440 A$="&B"+A$
2450 VPOKE M+768,VAL(A$)
2460 A$=""
2470 NEXT M
2480 RETURN
2490 REM Umwandlung in Inversschrift
2500 REM a=0 -> alles
2510 REM a=1 -> bis auf Leerzeichen
2520 REM a=2 -> nur Kleinbuchstaben
2530 REM a=3 -> nur Grossbuchstaben
2540 REM a=4 -> nur Zahlen
2550 FOR TQ=1 TO LEN(A$)
2560 IF A=0 THEN GOTO 2610
2570 IF A=1 AND MID$(A$,TQ,1)=" " THEN G
OTO 2620
2580 IF A=2 AND (MID$(A$,TQ,1)<CHR$(97)
OR MID$(A$,TQ,1)>CHR$(125)) THEN GOTO 26
20
2590 IF A=3 AND (MID$(A$,TQ,1)<CHR$(65)
OR MID$(A$,TQ,1)>CHR$(93)) THEN GOTO 262
0
```

```
2600 IF A=4 AND (MID$(A$,TQ,1)>CHR$(57)
OR MID$(A$,TQ,1)<CHR$(48)) THEN GOTO 262
0
2610 MID$(A$,TQ,1)=CHR$(ASC(MID$(A$,TQ,1
))+96)
2620 NEXT TQ
2630 PRINT A$
2640 RETURN
```

Kalender

Wer sich an den Umgang mit Computern gewöhnt hat, wird dem Programm 'Kalender' in dieser oder einer ähnlichen Form häufiger begegnen. Sei es nun das Programm 'Kalender' allein wie in diesem Fall, oder aber als Einbindung in ein größeres Programm (z.B. Terminverwaltung).

In jedem Fall geht es um die Aufaddition von Jahren, Monaten und Tagen, wobei als Endresultat der gewünschte Wochentag errechnet wird.

Das vorliegende Programm teilt sich in drei Bereiche, die aus der INPUT-Abfrage zum gewünschten Jahr, Monat und Kalendertag resultieren.

Die Jahreseingabe darf zwischen 1901 und dem Jahr 2000 liegen, wobei Schaltjahre automatisch registriert werden (Zeile 140).

Bei der Monatsrechnung wird der eingegebenen Monatszahl entsprechend der Name zugeordnet und die Höchsttageszahl festgestellt. Liegt ein Merker von der Jahreseingabe in der Form Z=1 = Schaltjahr und M=2 = Februar vor, so wird der 29.Februar = Schaltjahr mit hinzugezählt (Zeile 590).

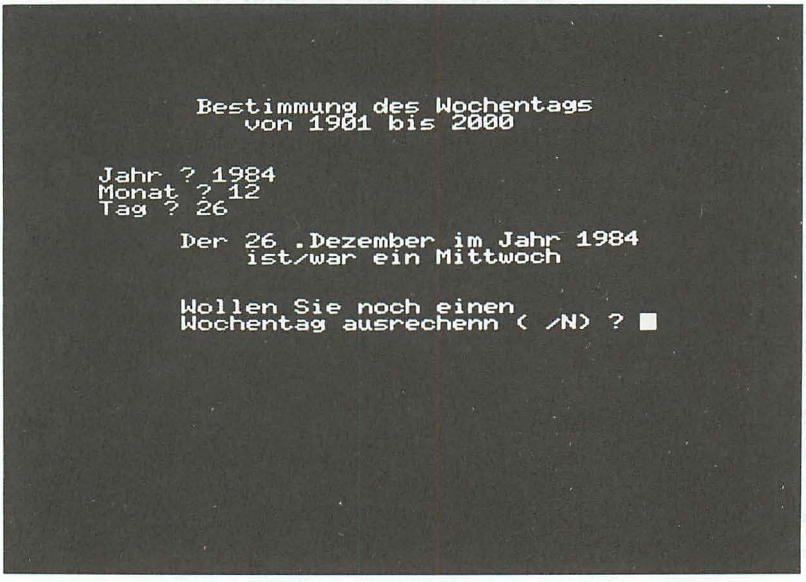
Schließlich erfolgt die Verzweigung zu den einzelnen Wochentagen (Zeile 750) und deren entsprechende Anzeige ('Der ... im Jahr ... ist/war ein ...'), und falls erwünscht, so kann wieder von vorne begonnen werden, sprich ein weiterer Wochentag ausgerechnet werden.

```
10 REM Kalender
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 SCREEN 0:COLOR 15,1,15:KEY OFF
50 WIDTH 40
60 PRINT
70 PRINT
80 PRINT
90 PRINT "      Bestimmung des Wochentag
s                   von 1901 bis 2000"
100 PRINT:PRINT
110 REM Eingabe der Jahreszahl und
        Berechnung (u.a. Schaltjahr)
120 INPUT "Jahr ";J
130 IF J<1901 OR J>2000 THEN GOSUB 990:G
OTO 50
140 IF INT(J/4)=J/4 THEN Z=1 ELSE Z=0
150 J1=2
160 IF J=1901 THEN GOTO 230
170 FOR N=1902 TO J
180 J1=J1+1
190 IF INT((N-1)/4)=(N-1)/4 THEN J1=J1+1
200 IF J1>6 THEN J1=J1-7
210 NEXT N
220 REM Eingabe des Monats und
        Berechnung (Name und Anzahl der
        Tage)
230 INPUT "Monat ";M
240 IF M<1 OR M>12 THEN GOSUB 990:GOTO 5
0
250 ON M GOSUB 270,290,310,330,350,370,3
90,410,430,450,470,490
260 GOTO 510
270 M$="Januar"
280 RETURN
290 M$="Februar"
300 RETURN
310 M$="Maerz"
320 RETURN
```

```
330 M$="April"
340 RETURN
350 M$="Mai"
360 RETURN
370 M$="Juni"
380 RETURN
390 M$="Juli"
400 RETURN
410 M$="August"
420 RETURN
430 M$="September"
440 RETURN
450 M$="Oktober"
460 RETURN
470 M$="November"
480 RETURN
490 M$="Dezember"
500 RETURN
510 RESTORE:FOR N=1 TO M
520 READ M1
530 NEXT N
540 DATA 31,28,31,30,31,30
550 DATA 31,31,30,31,30,31
560 IF Z=1 AND M=2 THEN M1=29
570 REM Eingabe des Tages (Zahl) und
        Berechnung des Wochentags
580 INPUT "Tag ";T
590 IF T=29 AND M=2 AND Z<>1 THEN GOSUB
990:GOTO 50
600 IF T<1 OR T>M1 THEN GOSUB 990:GOTO 5
0
610 T1=0
620 RESTORE
630 IF M=1 THEN GOTO 690
640 FOR N=2 TO M
650 READ M1
660 T1=T1+M1
670 NEXT N
680 IF Z=1 AND M>2 THEN T1=T1+1
690 T1=T1+T+J1
```

```
700 T1=T1-INT(T1/7)*7
710 IF T1=0 THEN T1=7
720 PRINT
730 PRINT "      Der";T;".";M$;" im Jahr"
;J
740 PRINT "          ist/war ein ";
750 ON T1 GOSUB 840,860,880,900,920,940,
960
760 PRINT
770 PRINT
780 INPUT "      Wollen Sie noch einen
                 Wochentag ausrechenn ( /
N) ";F$
790 IF LEFT$(F$,1)="N" OR LEFT$(F$,1)="n
" THEN F$="N"
800 IF F$="N" THEN END
810 RESTORE
820 CLS
830 GOTO 60
840 PRINT "Sonntag"
850 RETURN
860 PRINT "Montag"
870 RETURN
880 PRINT "Dienstag"
890 RETURN
900 PRINT "Mittwoch"
910 RETURN
920 PRINT "Donnerstag"
930 RETURN
940 PRINT "Freitag"
950 RETURN
960 PRINT "Sonnabend"
970 RETURN
980 END
```

```
990 PRINT:PRINT TAB(10) CHR$(200) "Falsc
he Eingabe!" CHR$(200)
1000 GOSUB 1010:RETURN
1010 PRINT:PRINT TAB(6) "<Bitte eine Tas
te druecken>"
1020 F$=INKEY$:IF F$="" THEN GOTO 1020
1030 RETURN
```



Balkendiagramm

Arne bekam vor einiger Zeit eine spezielle Schulaufgabe, die seine Programmierneigung und seine Begeisterung für grafische Darstellungen besonders fordern/fördern sollte: Von einer soziologischen Studie lagen verschiedene Daten vor und dieselbigen sollten nun - bitteschön - wie bei Wahlsendungen im Fernsehen in Form von Balkendiagrammen dargestellt werden.

Eine Aufgabe, über deren Realisierung Arne alsbald ins Schwitzen kam ... aber, wie Sie sehen, es hat schließlich doch mit Bravour geklappt!

Bei den vielen Programmierfreaks landauf/landab vielleicht auf den ersten Blick gar nicht so ungewöhnlich, daß ein Schüler auch mal solche Schulaufgaben lösen muß. Bedenkt man aber, daß Arne ein 'erst' 12-jähriger Gymnasiast aus Münster ist, muß man ihm zur Lösung dieses Problems besonders gratulieren!

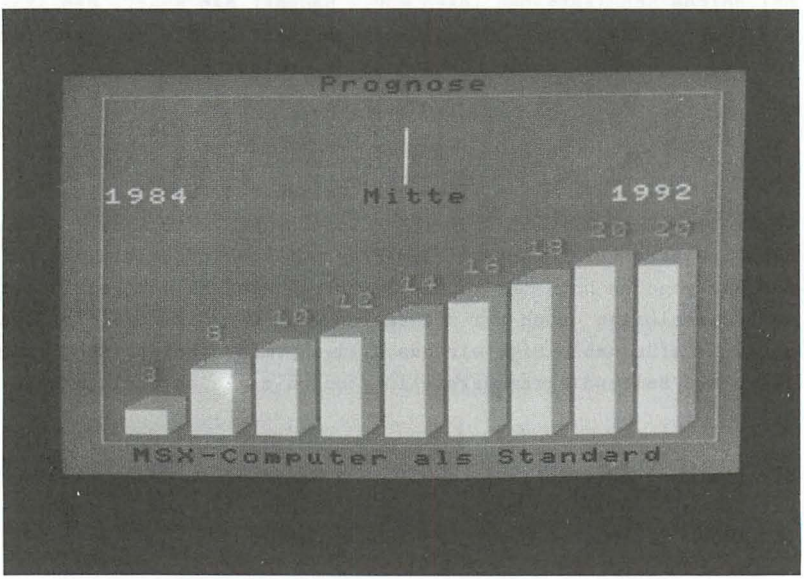
Da Sie, liebe Leser, wenig mit den soziologischen Daten von Arne anfangen können, habe ich das Programm ein klein wenig umgestrickt und auf einen allgemein anwendbaren Level gebracht. Geblieben sind allerdings die Einschränkungen, die Arne - sein Schulprojekt vor Augen - diesem Programm von vornherein mit auf den Weg gegeben hat: Es können lediglich neun Balken gezeichnet und dieselbigen auch nur mit Zahlen von '0' bis '20' belegt werden. Dafür haben Sie, wie das Bildschirmfoto zeigt, aber eine glasklare Balkendiagrammdarstellung und die gleich dreidimensional.

Sie können 'Ihren Zahlen' sowohl eine Überschrift als auch einen Untertitel geben, wobei die Positionierung des Textes automatisch in Mittenkoordinaten umgerechnet wird.

Zur textlichen Darstellung der Bedeutung des links und rechts angezeigten Anfangs- und Endwertes gibt es ebenfalls eine entsprechende Eingabemöglichkeit.

Eine Variation in der Anzahl der darzustellenden Werte ist nicht leicht durchführbar, hier aber sei dazu ein Tip angeführt, bevor Sie das Programm nach Ihren eigenen Ansprüchen umzumodeln beginnen: Die Darstellungsfolgen (zur Balkenzeichnung) laufen im Programm innerhalb von 'FOR ... NEXT'-Schleifen ab. Hierbei wird jeder Wert einmal durchlaufen ('STEP 1'). Geben Sie aber stattdessen 'STEP 2' ein, wird nun lediglich jeder zweite eingegebene Wert zur Bildschirmdarstellung herangezogen, also Wert 1, 3, 5, 7 und 9 - somit werden auch nur fünf Balken gezeichnet.

Nun aber viel Freude bei der Umsetzung Ihrer Zahlwerte in die dreidimensionale Balkengrafikdarstellung von Arne und Herrn Lüers.



```
10 REM Balkendiagramme
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
       Rainer Lueers & Arne Stoffregen
40 SCREEN 0:COLOR 15,1,1:WIDTH 40
50 REM Eingeben der neun Werte, der
       Ueberschriften und der
       Klassifikationen
60 INPUT "Ueberschrift (bis 30 Zeichen)
          ";C$:IF LEN(C$)>30 THEN GOTO 6
0
70 INPUT "Untertitel (bis 30 Zeichen)
          ";D$:IF LEN(D$)>30 THEN GOTO 7
0
80 INPUT "Positives Extrem (z.B.pos.) ";
A$:IF LEN(A$)>6 THEN A$=LEFT$(A$,6)
90 INPUT "Negatives Extrem (z.B.neg.) ";
B$:IF LEN(B$)>6 THEN B$=LEFT$(B$,6)
100 PRINT:PRINT "Eingabe von 9 Werten:":
PRINT
110 FOR N=1 TO 9:PRINT "Wert";N;:INPUT "
 (0 bis 20) ";WT(N)
120 IF WT(N)<0 OR WT(N)>20 THEN GOTO 110
 ELSE NEXT N
130 COLOR 1,13:OPEN "GRP:"FOR OUTPUT AS
#1
140 ON STOP GOSUB 570:STOP ON
150 FOR T=1 TO 9 STEP 1
160 C=C+WT(T)
170 IF WT(T)=0 THEN H(T)=.8*4:GOTO 190
180 H(T)=WT(T)*4
190 NEXT T
200 FH=170
210 REM Grafische Darstellung der
        Texte und des entsprechenden
        Bildrahmens
220 SCREEN 2
230 LINE(15,10)-(246,175),14,B
240 LINE(127,25)-(127,50),15
250 DRAW "bm 112,53"
```

```
260 PRINT #1,"Mitte"
270 DRAW "bm 42,1"
280 COLOR 1
290 A=(255-(LEN(C$)*8))/2:DRAW "bm =a;,1
"
300 REM Ausgabe der Ueberschrift und
        des Untertitels - jeweils
        'mittig' angeordnet
310 PRINT #1,C$
320 A=(255-(LEN(D$)*8))/2:DRAW "bm =a;,1
78":PRINT #1,D$:COLOR 15
330 DRAW "bm 17,53"
340 PRINT #1,A$
350 A=230-LEN(B$)*6:DRAW "bm =a;,53;"
360 PRINT #1,B$
370 REM Grafische Darstellung der
        eingegebenen Werte in Form
        von Balkendiagrammen
380 FOR I=1 TO 9 STEP 1
390 LINE (I*24,FH-H(I))-(I*24+15,FH),15,
BF
400 LINE -(I*24+21,FH-6),14
410 LINE -(I*24+21,FH-6-H(I)),14
420 LINE -(I*24+15,FH-H(I)),14
430 LINE -(I*24+15,FH),14
440 PAINT (I*24+17,FH-H(I)+2),14
450 LINE (I*24+21,FH-6-H(I))-(I*24+9,FH-
6-H(I)),14
460 LINE -(I*24,FH-H(I)),14
470 LINE (I*24,FH-H(I))-(I*24+15,FH-H(I)
),14
480 PAINT (I*24+8,FH-H(I)-4),14
490 NEXT I
500 FOR T=1 TO 9 STEP 1
510 A=T*24-2:B=FH-H(T)-20:DRAW "bm =a;,=
b;"
520 COLOR 3
530 PRINT #1,WT(T)
540 NEXT T
550 FOR T=1 TO 5000:NEXT T
560 A$=INKEY$:IF A$="" THEN GOTO 560
570 COLOR 15,1
```

Sporttabelle

Dieses Programm kann Ihnen in Zukunft sehr viel Freude bereiten und noch mehr Zeit ersparen.

Hier wird genau das vom Computer verwaltet, was sich innerhalb eines Jahres in Sportvereinen so oft und so regelmäßig abspielt. Als Paradebeispiel sei der Fußball genannt, der auch als Vorbild zu diesem Programm diente.

Zu Beginn der Eingabe haben Sie die Wahl, bis zu zwanzig Vereine gleichzeitig zu verwalten. Anschließend fragt Sie Ihr MSX-Computer, ob die Vereinsnamen, die in DATA-Zeilen bereits gespeichert sind (1. und 2. Bundesliga), für die Saison zutreffen, oder ob Sie selbst die Namen Ihrer bis zu zwanzig Vereine eingeben wollen.

Auch für die Begegnungen der Vereine in der Hin- und Rückrunde sind bereits alle möglichen Paarungen gespeichert. Wollen Sie diese Daten vom Computerspeicher übernehmen, so beantworten Sie auch diese Frage mit 'j'(a).

Hierzu ein Hinweis: Die in DATA-Zeilen gespeicherten Begegnungen der Clubs resultieren aus einer allgemeinen Aufstellung, wobei schließlich jede Mannschaft gegen jede Mannschaft nach einer Saison zweimal gespielt hat, einmal zu Hause, ein anderes Mal auswärts. Wenn Sie die Daten aus den DATA-Zeilen nicht übernehmen wollen, so geben Sie bitte, der Bildschirmanzeige entsprechend, alle Spieltage nacheinander in den Computerspeicher ein. Jedoch brauchen Sie nur die Hälfte aller Spielpaarungen einzugeben, denn die Rückrunde errechnet Ihr MSX-Computer selbstverständlich ohne Ihr Zutun.

Anschließend werden Sie mit einem Menü konfrontiert, daß Ihnen drei Möglichkeiten eröffnet: '1. Spieltag eingeben 2. aktuelle Tabelle zeigen 3. Spieltag ansehen!

Geben wir also zuerst die Ergebnisse des 1. Spieltags ein. Da es vorkommen kann, daß das eine oder andere Spiel ausfällt, gibt es auch hierfür eine Hilfe: Das Ergebnis '99,99' bedeutet: Spiel ausgefallen.

Wenn nicht bereits der zweite Spieltag erfolgt ist und diese Eintragungen Vorrang genießen, könen wir uns nun unsere Eingaben in übersichtlicher Form auf den Monitor holen. Wie funktioniert das nun: '3. Spieltag ansehen'? Uns wird nach der Auswahl dieses Menüpunktes die Möglichkeit eröffnet, nacheinander die ganze Saison auf Tastendruck vor unseren Augen abspielen zu lassen. Wir können aber selbstverständlich auch gezielt einige Spieltage zur Anzeige bringen.

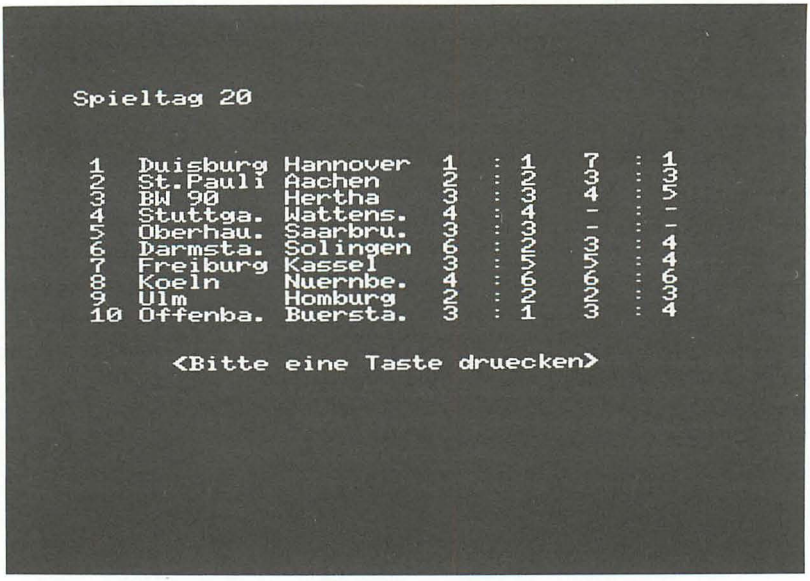
Springen wir in die Rückrunde (bei 20 Vereinen ab Spiel 20), so wird uns das - falls vorhanden - Spiel von der Hinrunde ebenfalls auf dem Bildschirm angezeigt.

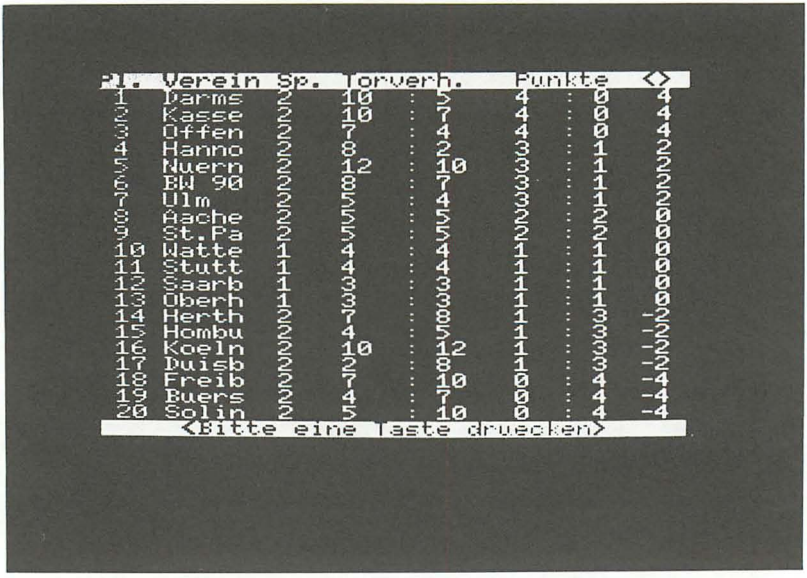
Ist eine Begegnung vor ein paar Spieltagen z.B. durch schlechtes Wetter ausgefallen (Eingabe '99, 99'), so können wir die Eintragung dieses Ergebnisses selbstverständlich nachholen. Hierzu bitte wieder '1. Spieltag eingeben' anwählen und den entsprechenden Spieltag aufrufen, wo die eine oder andere Partie noch aussteht. Ihr MSX-Computer wird mit einem kurzen Piepsen die bereits fertigen Spiele übergehen und genau da anhalten, wo noch Eintragungen vorzunehmen sind! Waren gleich mehrere Partien an einem Spieltag ausgefallen, so müssen die noch nicht nachgespielten Begegnungen auch bei dieser Wiederholungseingabe erneut mit '99, 99' (ausgefallen) quittiert werden.

Nun aber zum Kernstück des Programms, hin zur Tabellenerrechnung. Es dauert ein klein wenig, bis die Tabelle steht, dafür ist sie aber nicht nur geordnet, sondern auch noch recht ausführlich auf 40 Zeichen dargestellt: Hinter der erreichten Platzziffer und dem Vereinsnamenkürzel wird die Anzahl der bereits erfolgten Spiele des jeweiligen Vereins angezeigt. Schließlich wird auch noch das aktuelle Tor- und Punkteverhältnis zur Anzeige gebracht, das letztgenannte ergänzt durch die positive oder negative Differenz aus Plus- und Minuspunkten.

So bleibt die Übersicht jederzeit gewahr, wie und wo mit welchen Ergebnissen wann gespielt wurde und wie nun die aktuelle Platzziffer der jeweiligen Vereine ist.

Ein aufwendiges Programm, das allerdings eines erfordert: Lassen Sie Ihren MSX-Computer unter Strom! Eine Datenspeicherung ist absichtlich noch nicht eingefügt, denn mit Kassette würde dies bei solch einer Datenfülle einfach zu lange dauern. Warten wir geduldig auf die Diskettenstation! Die Anpassung dürfte Ihnen nicht allzu schwerfallen, da wir zu Beginn die dimensionierten Variablen in einer REM-Zeile erklärt haben. Hingegen können Sie einen Printer-Ausdruck der Tabelle ohne weiteres vornehmen, wenn Sie die dort stehenden PRINT-Befehle (Zeile 900 bis 960) in LPRINT-Befehle umwandeln.


Spieltag 20
1 Duisburg Hannover 1 : 1 7 : 1
2 St.Pauli Aachen 5 : 5 3 : 3
3 BW 90 Hertha 3 : 3 4 : 5
4 Stuttga. Wattens. 4 : 4 - : -
5 Oberhau. Saarbru. 3 : 3
6 Darmsta. Solingen 6 : 2 3 : 4
7 Freiburg Kassel 3 : 5 5 : 4
8 Koeln Nuernbe. 4 : 6 6 : 6
9 Ulm Homburg 2 : 2 2 : 3
10 Offenba. Buersta. 3 : 1 3 : 4
<Bitte eine Taste druecken>


Pl. Verein Sp. Torverh. Punkte <>
1 Darms 2 10 : 5 4 : 0 4
2 Kasse 2 10 : 7 4 : 0 4
3 Offen 2 7 : 4 4 : 0 4
4 Hanno 2 8 : 2 3 : 1 2
5 Nuern 2 12 : 10 3 : 1 2
6 BW 90 2 8 : 7 3 : 1 2
7 Ulm 2 5 : 4 3 : 1 2
8 Aache 2 5 : 5 2 : 2 0
9 St.Pa 2 5 : 5 2 : 2 0
10 Watte 1 4 : 4 1 : 1 0
11 Stutt 1 4 : 4 1 : 1 0
12 Saarb 1 3 : 3 1 : 1 0
13 Oberh 1 3 : 3 1 : 1 0
14 Herth 2 7 : 8 1 : 3 -1
15 Hombu 2 4 : 5 1 : 3 -2
16 Koeln 2 10 : 12 1 : 3 -2
17 Duisb 2 2 : 8 1 : 3 -2
18 Freib 2 7 : 10 0 : 4 -4
19 Buers 2 4 : 7 0 : 4 -4
20 Solin 2 5 : 10 0 : 4 -4
<Bitte eine Taste druecken>

```
10 REM Sporttabelle
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 SCREEN 0:KEY OFF:COLOR 15,1,1:PRINT "
Bitte einen Augenblick Geduld":GOSUB 610
00:CLS
50 WIDTH 40:DEFINT A-Z
60 REM Auswahl zwischen 4 bis
       20 Vereinen
70 INPUT "Vereine (4/6/8/10/12/14/16/18/
20) ";F$
80 REM Liste der Variablen, die
       dimensioniert werden mit
       ihren Bedeutungen
90 REM a$( )=Vereinsnamen
       b( , )=Spiel stattgefunden
       c( , , )=Ergebnis
       e( , )=Tore insg.
       f( , )=reingelassen insg.
100 REM g( , )=gewonnen insg.
        h( , )=unentschieden insg.
        i( , )=verloren insg.
        k( , , )=Spielpaarung
        z1 bis z5=fuer Sortierroutine
110 F=VAL(F$):IF F/2<>INT(F/2) THEN PRIN
T:GOSUB 1290:GOTO 50
120 IF F<4 OR F>20 THEN GOSUB 1290:GOTO
50
130 DIM A$(F),B(F*2,F/2),C(F,F*2-2,2),K(
F-1,F/2,2),E(F),F(F),G(F),H(F),I(F),Z1(F
+1),Z2(F+1),Z3(F+1),Z4(F+1),Z5(F+1)
140 REM Eingabe eigener Vereinsnamen
        oder Einlesen der Vereinsnamen
        aus DATA-Zeilen
150 INPUT "Namen aus Programmzeilen (J/
)";F$
160 IF LEFT$(F$,1)="j" OR LEFT$(F$,1)="J
" THEN F$="J"
161 IF F$="J" THEN GOTO 230
```

```
170 FOR N=1 TO F
171 FOR N=1 TO F
180 PRINT "Verein";N;:INPUT ": ";F$
190 IF F$="" THEN GOTO 180
200 IF LEN(F$)>10 THEN A$(N)=LEFT$(F$,10
) ELSE A$(N)=F$
210 NEXT N
220 GOTO 350
230 ON F GOSUB 250,250,250,250,250,260,2
50,270,250,280,250,290,250,300,250,310,2
50,320,250,330
240 GOTO 350
250 RESTORE 6350:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
260 RESTORE 6320:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
270 RESTORE 6290:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
280 RESTORE 6260:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
290 RESTORE 6230:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
300 RESTORE 6200:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
310 RESTORE 6170:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
320 RESTORE 6130:FOR N=1 TO F:READ A$(N)
:NEXT N:RETURN
330 RESTORE 6080:FOR N=1 TO F:READ A$(N)
:NEXT N
340 REM Spielfolge in DATA-Zeilen
        (uebliches System) kann heran-
        gezogen werden; es ist aber
        auch moeglich, alle Spielzu-
        sammenstellungen selbst
        einzugeben
350 INPUT "Spielfolge wie in DATA-Zeilen
 (J/ ) ";F$:IF LEFT$(F$,1)="J" OR LEFT$(
F$,1)="j" THEN F$="J"
351 IF F$="J" THEN GOTO 420
```

```
360 PRINT:PRINT:PRINT "Spielfolge bitte
mit Zahlen eingeben          (z.B. 1 <Kom
ma> 2 <ENTER>)          vorige Eingabe wi
ederholen ->0 eingeben"
370 FOR N=1 TO F-1:PRINT:PRINT:PRINT "Sp
ieltag";N:PRINT
380 FOR M=1 TO F/2
390 PRINT:PRINT "Spiel";M:INPUT A,B:IF A
>F OR B>F OR A=B THEN GOTO 390 ELSE IF A
=0 OR B=0 THEN M=M-1:GOTO 390 ELSE K(N,M
,1)=A:K(N,M,2)=B:A=0:B=0:PRINT A$(K(N,M,
1));" - ";A$(K(N,M,2))
400 NEXT M,N
410 GOTO 530
420 ON F GOSUB 440,440,440,440,440,450,4
40,460,440,470,440,480,440,490,440,500,4
40,510,440,520
430 GOTO 530
440 RESTORE 5000:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
450 RESTORE 5040:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
460 RESTORE 5100:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
470 RESTORE 5180:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
480 RESTORE 5280:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
490 RESTORE 5400:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
500 RESTORE 5540:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
```

```
510 RESTORE 5700:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
520 RESTORE 5880:FOR N=1 TO F-1:FOR M=1
TO F/2:READ K(N,M,1),K(N,M,2):NEXT M,N:R
ETURN
530 CLS:A$="Sportverein-Tabellen":A=0:LO
CATE 6,6:GOSUB 61120
540 LOCATE 6,9
550 PRINT "1.Spieltag eingeben"
560 LOCATE 6,10
570 PRINT "2.Aktuelle Tabelle zeigen"
580 LOCATE 6,11
590 PRINT "3.Spieltag ansehen"
600 LOCATE 6,13
610 INPUT "Ihre Wahl (1 bis 3) ";F$
620 IF VAL(F$)<1 OR VAL(F$)>4 THEN GOSUB
 1270:GOSUB 1290:GOTO 420
630 ON VAL(F$) GOSUB 660,900,1010
640 GOTO 530
650 REM Spieltag eingeben
660 CLS
670 INPUT "Welcher Spieltag ";F$:IF VAL(
F$)<1 OR VAL(F$)>F*2-2 THEN GOSUB 1290:G
OTO 670
680 IF VAL(F$)>F-1 THEN GOTO 740
690 FOR N=1 TO F/2:PRINT "Spiel";N
700 IF B(VAL(F$),N)<>0 THEN PRINT CHR$(7
);"Fehler! Spielergebnis bekannt:":PRINT
 A$(K(VAL(F$),N,1));"-";A$(K(VAL(F$),N,2
));":";C(K(VAL(F$),N,1),VAL(F$),1);"-";C
(K(VAL(F$),N,1),VAL(F$),2):GOTO 730
710 PRINT A$(K(VAL(F$),N,1));" - ";A$(K(
VAL(F$),N,2));:INPUT A,B:IF A=99 THEN PR
INT "ausgefallen":B(VAL(F$),N)=0:GOTO 73
0 ELSE GOSUB 790
720 B(VAL(F$),N)=1:C(K(VAL(F$),N,1),VAL(
F$),1)=A:C(K(VAL(F$),N,1),VAL(F$),2)=B:C
(K(VAL(F$),N,2),VAL(F$),1)=B:C(K(VAL(F$)
,N,2),VAL(F$),2)=A
```

```
730 NEXT N:RETURN
740 FOR N=1 TO F/2:PRINT "Spiel";N
750 IF B(VAL(F$),N)<>0 THEN PRINT CHR$(7
);"Fehler! Spielergebnis bekannt:":PRINT
 A$(K(VAL(F$)-F+1,N,2));"-";A$(K(VAL(F$)
-F+1,N,1));":";C(K(VAL(F$)-F+1,N,1),VAL(
F$),1);"-";C(K(VAL(F$)-F+1,N,1),VAL(F$),
2):GOTO 780
760 PRINT A$(K(VAL(F$)-F+1,N,2));" - ";A
$(K(VAL(F$)-F+1,N,1));:INPUT A,B:IF A=99
 THEN PRINT "ausgefallen":B(VAL(F$),N)=0
:GOTO 780 ELSE GOSUB 840
770 B(VAL(F$),N)=1:C(K(VAL(F$)-F+1,N,1),
VAL(F$),1)=A:C(K(VAL(F$)-F+1,N,1),VAL(F$
),2)=B:C(K(VAL(F$)-F+1,N,2),VAL(F$),1)=B
:C(K(VAL(F$)-F+1,N,2),VAL(F$),2)=A
780 NEXT N:RETURN
790 E(K(VAL(F$),N,1))=E(K(VAL(F$),N,1))+
A:F(K(VAL(F$),N,1))=F(K(VAL(F$),N,1))+B:
E(K(VAL(F$),N,2))=E(K(VAL(F$),N,2))+B:F(
K(VAL(F$),N,2))=F(K(VAL(F$),N,2))+A
800 IF A=B THEN H(K(VAL(F$),N,1))=H(K(VA
L(F$),N,1))+1:H(K(VAL(F$),N,2))=H(K(VAL(
F$),N,2))+1
810 IF A>B THEN G(K(VAL(F$),N,1))=G(K(VA
L(F$),N,1))+2:I(K(VAL(F$),N,2))=I(K(VAL(
F$),N,2))+2
820 IF A<B THEN I(K(VAL(F$),N,1))=I(K(VA
L(F$),N,1))+2:G(K(VAL(F$),N,2))=G(K(VAL(
F$),N,2))+2
830 RETURN
840 E(K(VAL(F$)-F+1,N,1))=E(K(VAL(F$)-F+
1,N,1))+A:F(K(VAL(F$)-F+1,N,1))=F(K(VAL(
F$)-F+1,N,1))+B:E(K(VAL(F$)-F+1,N,2))=E(
K(VAL(F$)-F+1,N,2))+B:F(K(VAL(F$)-F+1,N,
2))=F(K(VAL(F$)-F+1,N,2))+A
850 IF A=B THEN H(K(VAL(F$)-F+1,N,1))=H(
K(VAL(F$)-F+1,N,1))+1:H(K(VAL(F$)-F+1,N,
2))=H(K(VAL(F$)-F+1,N,2))+1
```

```
860 IF A>B THEN G(K(VAL(F$)-F+1,N,1))=G(
K(VAL(F$)-F+1,N,1))+2:I(K(VAL(F$)-F+1,N,
2))=I(K(VAL(F$)-F+1,N,2))+2
870 IF A<B THEN I(K(VAL(F$)-F+1,N,1))=I(
K(VAL(F$)-F+1,N,1))+2:G(K(VAL(F$)-F+1,N,
2))=G(K(VAL(F$)-F+1,N,2))+2
880 RETURN
890 REM Aktuelle Tabelle zeigen
900 CLS:GOSUB 1180
910 A$="Pl. Verein Sp. Torverh.  Punkte
  <> ":A=0:GOSUB 61120
920 FOR N=1 TO F
930 PRINT N;TAB(4);LEFT$(A$(Z5(N)),5);TA
B(10);(G(Z5(N))+(H(Z5(N))*2)+I(Z5(N)))/2
;TAB(14);E(Z5(N));TAB(19);":";F(Z5(N));T
AB(25);G(Z5(N))+H(Z5(N));TAB(29);":"H(Z5
(N))+I(Z5(N));TAB(34);G(Z5(N))-I(Z5(N))
950 NEXT N
970 A$="     <Bitte eine Taste druecken>
     ":A=0:GOSUB 61120
980 A$=INKEY$:IF A$="" THEN GOTO 980
990 RETURN
1000 REM Spieltag ansehen
1010 CLS:PRINT:INPUT "Die ganze Saison (
J/ ) ";F$:IF LEFT$(F$,1)="j" OR LEFT$(F$
,1)="J" THEN F$="J"
1011 IF F$="J" THEN GOTO 1100
1020 PRINT "Spieltag ( 1 bis";F*2-2;:INP
UT ") ";F$:IF VAL(F$)<1 OR VAL(F$)>F*2-2
 THEN GOSUB 1290:GOTO 1020 ELSE IF VAL(F
$)>F-1 THEN GOTO 1060
1030 N=VAL(F$):CLS:PRINT "Spieltag";N:PR
INT:PRINT:PRINT:FOR M=1 TO F/2:PRINT M;T
AB(4);A$(K(N,M,1));TAB(14);A$(K(N,M,2));
1040 IF B(N,M)=0 THEN PRINT TAB(32);"-
: -" ELSE PRINT TAB(31);C(K(N,M,1),N,1);
TAB(35);":";TAB(36);C(K(N,M,1),N,2);
1050 NEXT M:PRINT:PRINT:GOSUB 1270:GOTO
1170
```

```
1060 N=VAL(F$):CLS:PRINT "Spieltag";N:N=
N-(F-1):PRINT:PRINT:PRINT:FOR M=1 TO F/2
:PRINT M;TAB(4);A$(K(N,M,2));TAB(13);A$(
K(N,M,1));TAB(22);
1070 IF B(N,M)=1 THEN PRINT C(K(N,M,1),N
,2);TAB(26);":";TAB(27);C(K(N,M,1),N,1);
 ELSE PRINT " -  : -";
1080 IF B(N+(F-1),M)=0 THEN PRINT TAB(32
);"-  : -" ELSE PRINT TAB(31);C(K(N,M,1)
,N+F-1,1);TAB(35);":";TAB(36);C(K(N,M,1)
,N+F-1,2);
1090 NEXT M:PRINT:PRINT:GOSUB 1270:GOTO
1170
1100 FOR N=1 TO F-1:CLS:PRINT "Spieltag"
;N:PRINT:PRINT:PRINT:FOR M=1 TO F/2:PRIN
T M;TAB(4);A$(K(N,M,1));TAB(14);A$(K(N,M
,2));
1110 IF B(N,M)=0 THEN PRINT TAB(32);"-
: -"; ELSE PRINT TAB(32);C(K(N,M,1),N,1)
;TAB(35);":";TAB(36);C(K(N,M,1),N,2);
1120 NEXT M:PRINT:PRINT:GOSUB 1270:NEXT
N
1130 FOR N=1 TO F-1:CLS:PRINT "Spieltag"
;N+F-1:PRINT:PRINT:PRINT:FOR M=1 TO F/2:
PRINT M;TAB(4);A$(K(N,M,2));TAB(13);A$(K
(N,M,1));TAB(22);
1140 IF B(N,M)=0 THEN PRINT " -  : -"; E
LSE PRINT C(K(N,M,1),N,2);TAB(26);":";TA
B(28);C(K(N,M,1),N,1);
1150 IF B(N+F-1,M)=0 THEN PRINT TAB(32);
"-  : -"; ELSE PRINT TAB(31);C(K(N,M,1),
N+F-1,1);TAB(35);":";TAB(36);C(K(N,M,1),
N+F-1,2);
1160 NEXT M:PRINT:PRINT:GOSUB 1270:NEXT
1170 RETURN
```

```
1180 REM Sortierunterroutine
1190 FOR N=1 TO F:Z1(N)=G(N)+H(N):Z2(N)=
E(N)-F(N):Z3(N)=E(N):Z4(N)=-1:NEXT N
1200 FOR N=1 TO F
1210 FOR M=1 TO F
1220 IF Z1(M)>Z4(N) THEN Z4(N)=Z1(M):A=M
:GOTO 1250
1230 IF Z1(M)=Z4(N) AND Z2(M)>Z2(A) THEN
 Z4(N)=Z1(M):A=M:GOTO 1250
1240 IF Z1(M)=Z4(N) AND Z2(M)=Z2(A) AND
Z3(M)>Z3(A) THEN Z4(N)=Z1(M):A=M
1250 NEXT M:Z1(A)=-1:Z5(N)=A:A=F+1:NEXT
N
1260 RETURN
1270 PRINT:PRINT TAB(6) "<Bitte eine Tas
te druecken>"
1280 A$=INKEY$:IF A$="" THEN GOTO 1280 E
LSE RETURN
1290 PRINT TAB(10) CHR$(200) "Falsche Ei
ngabe!" CHR$(200):GOSUB 1270:RETURN
5000 REM Datas fuer 4 Vereine
5010 DATA 1,4,2,3
5020 DATA 4,3,1,2
5030 DATA 2,4,3,1
5040 REM Datas fuer 6 Vereine
5050 DATA 1,6,2,5,3,4
5060 DATA 6,4,5,3,1,2
5070 DATA 2,6,3,1,4,5
5080 DATA 6,5,1,4,2,3
5090 DATA 3,6,4,2,5,1
5100 REM Datas fuer 8 Vereine
5110 DATA 1,8,2,7,3,6,4,5
5120 DATA 8,5,6,4,7,3,1,2
5130 DATA 2,8,3,1,4,7,5,6
5140 DATA 8,6,7,5,1,4,2,3
5150 DATA 3,8,4,2,5,1,6,7
5160 DATA 8,7,1,6,2,5,3,4
5170 DATA 4,8,5,3,6,2,7,1
```

```
5180 REM Datas fuer 10 Vereine
5190 DATA 1,10,2,9,3,8,4,7,5,6
5200 DATA 10,6,7,5,8,4,9,3,1,2
5210 DATA 2,10,3,1,4,9,5,8,6,7
5220 DATA 10,7,8,6,9,5,1,4,2,3
5230 DATA 3,10,4,2,5,1,6,9,7,8
5240 DATA 10,8,9,7,1,6,2,5,3,4
5250 DATA 4,10,5,3,6,2,7,1,8,9
5260 DATA 10,9,1,8,2,7,3,6,4,5
5270 DATA 5,10,6,4,7,3,8,2,9,1
5280 REM Datas fuer 12 Vereine
5290 DATA 1,12,2,11,3,10,4,9,5,8,6,7
5300 DATA 12,7,8,6,9,5,10,4,11,3,1,2
5310 DATA 2,12,3,1,4,11,5,10,6,9,7,8
5320 DATA 12,8,9,7,10,6,11,5,1,4,2,3
5330 DATA 3,12,4,2,5,1,6,11,7,10,8,9
5340 DATA 12,9,10,8,11,7,1,6,2,5,3,4
5350 DATA 4,12,5,3,6,2,7,1,8,11,9,10
5360 DATA 12,10,11,9,1,8,2,7,3,6,4,5
5370 DATA 5,12,6,4,7,3,8,2,9,1,10,11
5380 DATA 12,11,1,10,2,9,3,8,4,7,5,6
5390 DATA 6,12,7,5,8,4,9,3,10,2,11,1
5400 REM Datas fuer 14 Vereine
5410 DATA 1,14,2,13,3,12,4,11,5,10,6,9,7
,8
5420 DATA 14,8,9,7,10,6,11,5,12,4,13,3,1
,2
5430 DATA 2,14,3,1,4,13,5,12,6,11,7,10,8
,9
5440 DATA 14,9,10,8,11,7,12,6,13,5,1,4,2
,3
5450 DATA 3,14,4,2,5,1,6,13,7,12,8,11,9,
10
5460 DATA 14,10,11,9,12,8,13,7,1,6,2,5,3
,4
5470 DATA 4,14,5,3,6,2,7,1,8,13,9,12,10,
11
5480 DATA 14,11,12,10,13,9,1,8,2,7,3,6,4
,5
```

```
5490 DATA 5,14,6,4,7,3,8,2,9,1,10,13,11,
12
5500 DATA 14,12,13,11,1,10,2,9,3,8,4,7,5
,6
5510 DATA 6,14,7,5,8,4,9,3,10,2,11,1,12,
13
5520 DATA 14,13,1,12,2,11,3,10,4,9,5,8,6
,7
5530 DATA 7,14,8,6,9,5,10,4,11,3,12,2,13
,1
5540 REM Datas fuer 16 Vereine
5550 DATA 1,16,2,15,3,14,4,13,5,12,6,11,
7,10,8,9
5560 DATA 16,9,10,8,11,7,12,6,13,5,14,4,
15,3,1,2
5570 DATA 2,16,3,1,4,15,5,14,6,13,7,12,8
,11,9,10
5580 DATA 16,10,11,9,12,8,13,7,14,6,15,5
,1,4,2,5
5590 DATA 3,16,4,2,5,1,6,15,7,14,8,13,9,
12,10,11
5600 DATA 16,11,12,10,13,9,14,8,15,7,1,6
,2,5,3,4
5610 DATA 4,16,5,3,6,2,7,1,8,15,9,14,10,
13,11,12
5620 DATA 16,12,13,11,14,10,15,9,1,8,2,7
,3,6,4,5
5630 DATA 5,16,6,4,7,3,8,2,9,1,10,15,11,
14,12,13
5640 DATA 16,13,14,12,15,11,1,10,2,9,3,8
,4,7,5,6
5650 DATA 6,16,7,5,8,4,9,3,10,2,11,1,12,
15,13,14
5660 DATA 16,14,15,13,1,12,2,11,3,10,4,9
,5,8,6,7
5670 DATA 7,16,8,6,9,5,10,4,11,3,12,2,13
,1,14,15
5680 DATA 16,15,1,14,2,13,3,12,4,11,5,10
,6,9,7,8
```

```
5690 DATA 8,16,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1
5700 REM Datas fuer 18 Vereine
5710 DATA 1,18,2,17,3,16,4,15,5,14,6,13,
7,12,8,11,9,10
5720 DATA 18,10,11,9,12,8,13,7,14,6,15,5
,16,4,17,3,1,2
5730 DATA 2,18,3,1,4,17,5,16,6,15,7,14,8
,13,9,12,10,11
5740 DATA 18,11,12,10,13,9,14,8,15,7,16,
6,17,5,1,4,2,3
5750 DATA 3,18,4,2,5,1,6,17,7,16,8,15,9,
14,10,13,11,12
5760 DATA 18,12,13,11,14,10,15,9,16,8,17
,7,1,6,2,5,3,4
5770 DATA 4,18,5,3,6,2,7,1,8,17,9,16,10,
15,11,14,12,13
5780 DATA 18,13,14,12,15,11,16,10,17,9,1
,8,2,7,3,6,4,5
5790 DATA 5,18,6,4,7,3,8,2,9,1,10,17,11,
16,12,15,13,14
5800 DATA 18,14,15,13,16,12,17,11,1,10,2
,9,3,8,4,7,5,6
5810 DATA 6,18,7,5,8,4,9,3,10,2,11,1,12,
17,13,16,14,15
5820 DATA 18,15,16,14,17,13,1,12,2,11,3,
10,4,9,5,8,6,7
5830 DATA 7,18,8,6,9,5,10,4,11,3,12,2,13
,1,14,17,15,16
5840 DATA 18,16,17,15,1,14,2,13,3,12,4,1
1,5,10,6,9,7,8
5850 DATA 8,18,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1,16,17
5860 DATA 18,17,1,16,2,15,3,14,4,13,5,12
,6,11,7,10,8,9
5870 DATA 9,18,10,8,11,7,12,6,13,5,14,4,
15,3,16,2,17,1
```

```
5880 REM Datas fuer 20 Vereine
5890 DATA 1,20,2,19,3,18,4,17,5,16,6,15,
7,14,8,13,9,12,10,11
5900 DATA 20,11,12,10,13,9,14,8,15,7,16,
6,17,5,18,4,19,3,1,2
5910 DATA 2,20,3,1,4,19,5,18,6,17,7,16,8
,15,9,14,10,13,11,12
5920 DATA 20,12,13,11,14,10,15,9,16,8,17
,7,18,6,19,5,1,4,2,3
5930 DATA 3,20,4,2,5,1,6,19,7,18,8,17,9,
16,10,15,11,14,12,13
5940 DATA 20,13,14,12,15,11,16,10,17,9,1
8,8,19,7,1,6,2,5,3,4
5950 DATA 4,20,5,3,6,2,7,1,8,19,9,18,10,
17,11,16,12,15,13,14
5960 DATA 20,14,15,13,16,12,17,11,18,10,
19,9,1,8,2,7,3,6,4,5
5970 DATA 5,20,6,4,7,3,8,2,9,1,10,19,11,
18,12,17,13,16,14,15
5980 DATA 20,15,16,14,17,13,18,12,19,11,
1,10,2,9,3,8,4,7,5,6
5990 DATA 6,20,7,5,8,4,9,3,10,2,11,1,12,
19,13,18,14,17,15,16
6000 DATA 20,16,17,15,18,14,19,13,1,12,2
,11,3,10,4,9,5,8,6,7
6010 DATA 7,20,8,6,9,5,10,4,11,3,12,2,13
,1,14,19,15,18,16,17
6020 DATA 20,17,18,16,19,15,1,14,2,13,3,
12,4,11,5,10,6,9,7,8
6030 DATA 8,20,9,7,10,6,11,5,12,4,13,3,1
4,2,15,1,16,19,17,18
6040 DATA 20,18,19,17,1,16,2,15,3,14,4,1
3,5,12,6,11,7,10,8,9
6050 DATA 9,20,10,8,11,7,12,6,13,5,14,4,
15,3,16,2,17,1,18,19
6060 DATA 20,19,1,18,2,17,3,16,4,15,5,14
,6,13,7,12,8,11,9,10
6070 DATA 10,20,11,9,12,8,13,7,14,6,15,5
,16,4,17,3,18,2,19,1
```

```
6080 REM Datas fuer 2.Liga 1984/85
         Beispiel fuer 20 Vereine
6090 DATA Hannover,Aachen,Hertha,Wattens
.,Saarbru.
6100 DATA Solingen,Kassel,Nuernbe.,Hombu
rg,Buersta.
6110 DATA Offenba.,Ulm,Koeln,Freiburg,Da
rmsta.
6120 DATA Oberhau.,Stuttga.,BW 90,St.Pau
li,Duisburg
6130 REM Datas fuer Bundesliga 1984/85
         Beispiel fuer 18 Vereine
6140 DATA Muenchen,Gladbach,Werder,Bochu
m,Hamburg,Kaisers.
6150 DATA Koeln,Stuttga.,Uerding.,Leverk
u.,Karlsru.,Frankfu.
6160 DATA Waldhof,Schalke,Duessel.,Biele
fe.,Dortmund,Braunsc.
6170 REM Datas fuer 16 Vereine
6180 DATA A,B,C,D,E,F,G,H
6190 DATA I,J,K,L,M,N,O,P
6200 REM Datas fuer 14 Vereine
6210 DATA A,B,C,D,E,F,G
6220 DATA H,I,J,K,L,M,N
6230 REM Datas fuer 12 Vereine
6240 DATA A,B,C,D,E,F
6250 DATA G,H,I,J,K,L
6260 REM Datas fuer 10 Vereine
6270 DATA A,B,C,D,E
6280 DATA F,G,H,I,J
6290 REM Datas fuer 8 Vereine
6300 DATA A,B,C,D
6310 DATA E,F,G,H
6320 REM Datas fuer 6 Vereine
6330 DATA A,B,C
6340 DATA D,E,F
6350 REM Datas fuer 4 Vereine
6360 DATA A,B
6370 DATA C,D
```

```
61000 REM Umwandlung >CHR$(127)->invers
61010 FOR M=2304 TO 3055
61020 A$=BIN$(VPEEK(M))
61030 A$=STRING$(8-LEN(A$),"0")+A$
61040 FOR N=1 TO 8
61050 IF MID$(A$,N,1)="1" THEN MID$(A$,N
,1)="0" ELSE MID$(A$,N,1)="1"
61060 NEXT N
61070 A$="&B"+A$
61080 VPOKE M+768,VAL(A$)
61090 A$=""
61100 NEXT M
61110 RETURN
61120 REM Umwandlung in Inversschrift
61130 REM a=0 -> alles
61140 REM a=1 -> bis auf Leerzeichen
61150 REM a=2 -> nur Kleinbuchstaben
61160 REM a=3 -> nur Grossbuchstaben
61170 REM a=4 -> nur Zahlen
61180 FOR PQ=1 TO LEN(A$)
61200 IF A=0 THEN GOTO 61250
61210 IF A=1 AND MID$(A$,PQ,1)=" " THEN
GOTO 61260
61220 IF A=2 AND (MID$(A$,PQ,1)<CHR$(97)
 OR MID$(A$,PQ,1)>CHR$(125)) THEN GOTO 6
1260
61230 IF A=3 AND (MID$(A$,PQ,1)<CHR$(65)
 OR MID$(A$,PQ,1)>CHR$(93)) THEN GOTO 61
260
61240 IF A=4 AND (MID$(A$,PQ,1)>CHR$(57)
 OR MID$(A$,PQ,1)<CHR$(48)) THEN GOTO 61
260
61250 MID$(A$,PQ,1)=CHR$(ASC(MID$(A$,PQ,
1))+96)
61260 NEXT PQ
61270 PRINT A$
61280 RETURN
```

Hollow - das Kirschenspiel

Ein Spiel wie in der Spielhalle, das alle Grafikmöglichkeiten Ihres MSX-Computers voll und ganz nutzt.

Worum geht es dabei? Lassen wir Lars selbst zu Wort kommen, einen 14-jährigen Gymnasiasten aus Münster, der die Idee zu diesem Spiel hatte und den größten Teil dieses Programms im stillen Kämmerlein selbst programmierte:

HOLLOW

Los geht sie, die fröhliche Jagd nach Kirschen. Sie müssen mit einem Vogel möglichst viele davon fressen. Sie dürfen dabei allerdings nicht mit den anderen Vögeln zusammenstoßen. Immer wenn Sie 20 Kirschen gefressen haben, verwandeln sich die Vögel, als Bonus, in Kirschen, die Sie fressen können. Wenn Sie 500 Punkte haben, wird das Spiel schwieriger. Wenn Sie 700 Punkte und nicht mehr alle Leben haben, bekommen Sie ein Zusatzleben. Gesteuert wird ganz normal mit Joystick oder Tastatur, nur 'rauf' geht's mit dem Feuerknopf bzw. der Leertaste. Runter geht's aufgrund der Schwerkraft von selbst!'

Es ist wahrlich nicht leicht, selbst für Lars - unseren Jungprogrammierer - heute noch nicht, mehr als 500 Punkte zu erzielen. Woran liegt das? Ab 500 Punkten stürzen die Vögel immer unkontrollierter in steilem Bogen auf und nieder ... möglichst immer an den Stellen, wo Sie Ihre Kirschen zu fressen wünschen. Aber lassen Sie es mit Ruhe angehen, Lars hat kein Zeitlimit in das Programm 'Hollow' eingebaut!

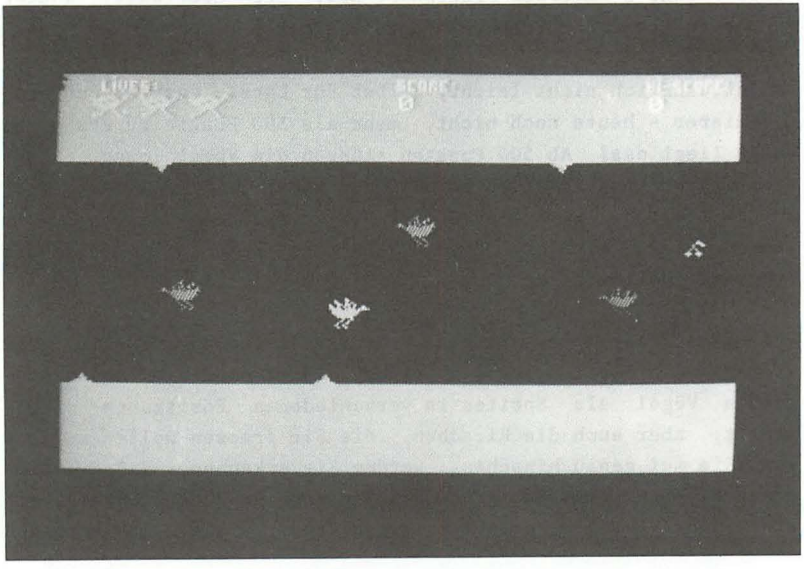
Was das Programmieren angeht, so werden Ihnen vor allem die vielen 'DATA'-Zeilen ins Auge fallen. Hier sind die herumfliegenden Vögel als Sprites in verschiedenen Positionen dargestellt, aber auch die Kirschen, die Sie fressen wollen/sollen. Wenn Sie mal genau hinsehen, werden Sie erkennen, daß die Kirschen nicht nur mit Sprites, sondern auch an anderer Stelle mit einem 'DRAW'-Befehl erstellt worden sind. Vielleicht erinnern

Sie sich: einmal verwandeln sich die anderen Vögel in Kirschen (Sprites), ein anderes Mal brauchen Sie als Vogel 'nur' zur feststehenden Kirsche hinzufliegen ('fest' als 'DRAW'-Befehl).

Stoßen Sie mit den Vögeln (Sprites) zusammen, tritt der Befehl 'ON SPRITE GOSUB' in Aktion und Sie streben einem raschen Absturz=Spieltod entgegen.

Der Sound ist noch ein klein wenig mickrig, aber laut Lars würde sonst die Bildschirmaktion zu langsam werden!? Hätte Lars zuvor den 'Soundeditor' gehabt, so wären zumindest Flattergeräusche ohne Unterbrechung während des Spielablaufs möglich gewesen (einmal mit 'SOUND' programmiert erklingt der selbst fabrizierte Klang andauernd ohne Programmpause). So wie es jetzt ist, gibt es halt nur eine Anfangs- und eine Endmelodie sowie besondere Geräusche bei besonderen Aktionen (Kirsche fressen, Spieltod usw.).

Nun aber genug der Vorerzählung. Viel Spaß beim Programmieren und Spielen wünschen Ihnen Lars und Herr Lüers.



```
10 REM Spiel 'HOLLOW'
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
       Rainer Lueers & Lars Lewejohann
40 REM Kirsche fest
50 B$="f211h111g1d1f312f1b14e112e2"
60 STOP ON
70 ON STOP GOSUB 4460
80 COLOR 10,1,1
90 SCREEN 2,2,0
100 OPEN "GRP:" FOR OUTPUT AS #1
110 DRAW "BM 100,85"
120 PRINT #1,"HOLLOW"
130 GOSUB 4910
140 DEFINT A-Z
150 REM Spritedefinition 8*8
160 FOR A=9 TO 11
170 S$=""
180 FOR B=1 TO 8
190 READ A$
200 S$=S$+CHR$(VAL("&B"+A$))
210 NEXT B
220 SPRITE$(A)=S$
230 NEXT A
240 REM Felsvorsprung unten
250 DATA 00000000
260 DATA 00000000
270 DATA 00000000
280 DATA 00001000
290 DATA 00011000
300 DATA 00011100
310 DATA 01111100
320 DATA 11111111
330 '
```

```
340 REM Felsvorsprung oben
350 DATA 11111111
360 DATA 01111100
370 DATA 00011100
380 DATA 00011000
390 DATA 00001000
400 DATA 00000000
410 DATA 00000000
420 DATA 00000000
430 '
440 REM Kirsche beweglich
450 DATA 00000100
460 DATA 00001110
470 DATA 00010011
480 DATA 00010000
490 DATA 00101000
500 DATA 01000100
510 DATA 11101110
520 DATA 01000100
530 REM Spritedefinition 32*32
540 FOR A=0 TO 8
550 S$=SPACE$(64)
560 FOR I%=1 TO 16
570 READ A$
580 MID$(S$,I%,1)=CHR$(VAL("&B"+LEFT$(A$
,8)))
590 MID$(S$,I%+16,1)=CHR$(VAL("&B"+RIGHT
$(A$,8)))
600 NEXT I%
610 SPRITE$(A)=S$
620 NEXT A
630 '
```

```
640 REM Vogel laufend rechts
650 DATA 0000000000000000
660 DATA 0000000000000000
670 DATA 0000000000000000
680 DATA 0000000000000000
690 DATA 1000000000001100
700 DATA 1101111110001110
710 DATA 1111111111011000
720 DATA 0111110011110000
730 DATA 0111001111100000
740 DATA 0011111111000000
750 DATA 0001111110000000
760 DATA 0000010010000000
770 DATA 0000100001000000
780 DATA 0111000000100000
790 DATA 0100000000110000
800 DATA 0000000000000000
810 '
820 DATA 0000000000000000
830 DATA 0000000000000000
840 DATA 0000000000000000
850 DATA 0000000000000110
860 DATA 1000000000000111
870 DATA 1101111110001100
880 DATA 1111111111011000
890 DATA 0111110011110000
900 DATA 0111001111100000
910 DATA 0011111111000000
920 DATA 0001111110000000
930 DATA 0000100001000000
940 DATA 0000010100000000
950 DATA 0000001000000000
960 DATA 0000110100000000
970 DATA 0000100011000000
980 '
```

```
990  REM Vogel fliegend rechts
1000 DATA 0000000000000000
1010 DATA 0000001000000000
1020 DATA 0010001100000000
1030 DATA 0011001010000000
1040 DATA 1001101101000000
1050 DATA 1101010101001100
1060 DATA 1111101111011110
1070 DATA 0111111111110000
1080 DATA 0111111111100000
1090 DATA 0011111111000000
1100 DATA 0001111110000000
1110 DATA 0000010010000000
1120 DATA 0000100100000000
1130 DATA 0001001000000000
1140 DATA 0010010000000000
1150 DATA 0001001000000000
1160 '
1170 DATA 0000000000000000
1180 DATA 0000000000000000
1190 DATA 0000000000000000
1200 DATA 0000000000000110
1210 DATA 1000000000000111
1220 DATA 1101111110001100
1230 DATA 1111111111011000
1240 DATA 0111110111110000
1250 DATA 0111101111100000
1260 DATA 0011011111000000
1270 DATA 0001111110000000
1280 DATA 0000010010000000
1290 DATA 0000100100000000
1300 DATA 0001001000000000
1310 DATA 0010010000000000
1320 DATA 0001001000000000
1330 '
```

```
1340 REM Vogel laufend links
1350 DATA 0000000000000000
1360 DATA 0000000000000000
1370 DATA 0000000000000000
1380 DATA 0000000000000000
1390 DATA 0000000000000001
1400 DATA 0011000111111011
1410 DATA 0111101111111111
1420 DATA 0000111110011110
1430 DATA 0000011111001110
1440 DATA 0000001111111100
1450 DATA 0000000111111000
1460 DATA 0000000100100000
1470 DATA 0000001000010000
1480 DATA 0000010000001110
1490 DATA 0001100000000010
1500 DATA 0000000000000000
1510 '
1520 DATA 0000000000000000
1530 DATA 0000000000000000
1540 DATA 0000000000000000
1550 DATA 0110000000000000
1560 DATA 1110000000000001
1570 DATA 0011000111111011
1580 DATA 0001101111111111
1590 DATA 0000111110011110
1600 DATA 0000011111001110
1610 DATA 0000001111111100
1620 DATA 0000000111111000
1630 DATA 0000000100010000
1640 DATA 0000000010100000
1650 DATA 0000000001000000
1660 DATA 0000000010110000
1670 DATA 0000001100010000
1680 '
```

```
1690 REM Vogel fliegend links
1700 DATA 0000000000000000
1710 DATA 0000000001000000
1720 DATA 0000000011000100
1730 DATA 0000000101001100
1740 DATA 0000001011011001
1750 DATA 0011001010101011
1760 DATA 0111101111011111
1770 DATA 0000111111111110
1780 DATA 0000011111111110
1790 DATA 0000001111111100
1800 DATA 0000000111111000
1810 DATA 0000000100100000
1820 DATA 0000000001001000
1830 DATA 0000000000100100
1840 DATA 0000000000100100
1850 DATA 0000000001001000
1860 '
1870 DATA 0000000000000000
1880 DATA 0000000000000000
1890 DATA 0000000000000000
1900 DATA 0110000000000000
1910 DATA 1110000000000001
1920 DATA 0011000111111011
1930 DATA 0001101111111111
1940 DATA 0000111101111110
1950 DATA 0000011110111110
1960 DATA 0000001111011100
1970 DATA 0000000111111000
1980 DATA 0000000100100000
1990 DATA 0000000001001000
2000 DATA 0000000000100100
2010 DATA 0000000000100100
2020 DATA 0000000001001000
2030 '
```

```
2040 REM Vogel tot
2050 DATA 0000000000000000
2060 DATA 0000000000000000
2070 DATA 0000000000000000
2080 DATA 0000000000000000
2090 DATA 0000000000000000
2100 DATA 0000000000000000
2110 DATA 1000000000001100
2120 DATA 1101111100010010
2130 DATA 1111100111001100
2140 DATA 0110011111000000
2150 DATA 0011111110100000
2160 DATA 0100000010011100
2170 DATA 1011111100001110
2180 DATA 0100000000000000
2190 DATA 0000000000000000
2200 DATA 0000000000000000
2210 '
2220 CLS
2230 DRAW "BM 0,0"
2240 COLOR 15
2250 PRINT #1,TAB(12);"HOLLOW"
2260 COLOR 3
2270 PRINT #1,TAB(12);"======"
2280 COLOR 10
2290 PRINT #1,
2300 PRINT #1,
2310 PRINT #1,
2320 PRINT #1,"        Wollen Sie die":P
RINT #1,"       Anleitung lesen?"
2330 PRINT #1,
2340 PRINT #1,
2350 PRINT #1,"             (J/N)"
2360 C$=INKEY$
2370 IF C$="J" OR C$="j" THEN GOTO 4960
ELSE IF C$="N" OR C$="n" THEN GOTO 2380
ELSE GOTO 2360
2380 CLS
2390 DRAW "bm 0,0"
2400 COLOR 15
```

```
2410 PRINT #1,TAB(10);"HOLLOW"
2420 COLOR 3
2430 PRINT #1,TAB(10);"======"
2440 COLOR 10
2450 PRINT #1,
2460 PRINT #1,
2470 PRINT #1,"      Wollen Sie mit "
2480 COLOR 15
2490 PRINT #1,"         Joystick"
2500 COLOR 10
2510 PRINT #1,"       oder";
2520 PRINT #1," mit der "
2530 COLOR 15
2540 PRINT #1,"         Tastatur "
2550 COLOR 10
2560 PRINT #1,"         spielen?"
2570 PRINT #1,
2580 PRINT #1,TAB(10);"(J/T)"
2590 C$=INKEY$
2600 IF C$="T" OR C$="t" THEN SZ=0 ELSE
IF C$="J" OR C$="j" THEN SZ=1 ELSE GOTO
2590
2610 IF MU=1 THEN GOSUB 4910
2620 REM Festlegung der Variablen
        von Leben usw., die nicht
        wieder hochgesetzt werden
        duerfen
2630 MS=0
2640 SM=0
2650 S=SZ
2660 PU=0
2670 PR(1)=0
2680 PR(2)=0
2690 PR(3)=0
2700 EN=4
2710 TI=0
2720 CLS
```

```
2730 REM Hoehlenwaende
2740 LINE (0,148)-(256,192),2,BF
2750 LINE (0,40)-(256,0),2,BF
2760 COLOR 15
2770 REM Anzeige Leben
2780 DRAW "c15bm16,3d4r2br2u4bm22,3d3f1e
1u3br2d4r212u2r111u2r2br412d2r2d212"
2790 PSET (20,1),15
2800 PSET (35,4),15
2810 PSET (35,6),15
2820 REM Anzeige Score
2830 DRAW "c15bm125,7r2u212u2r2br2r212d4
r2br2r2u412d4r2br2u4r2d1g1f1d1br2u4r212d
2r111d2r2"
2840 PSET (146,4),15
2850 PSET (146,6),15
2860 DRAW "bm 118,10":PRINT #1,PU
2870 REM Anzeige High-Score
2880 DRAW "c15bm220,7u4d2r2u2d4br2u4d2br
2r2bd2br2r2u212u2r2br2r212d4r2br2r2u412d
4r2br2u4r2d1g1f1d1br2u4r212d2r111d2r2"
2890 PSET (250,4),15
2900 PSET (250,6),15
2910 PSET (224,1),15
2920 DRAW "bm 213,10":PRINT #1,HC
2930 REM Variablen, die fuer das
        Hauptprogramm verwendet
        werden
2940 FR=110
2950 Z=182
2960 ZR=110
2970 G=0
2980 F=82
2990 H=0
3000 K=0
3010 RZ=3
3020 RF=-4
3030 RG=2
3040 R=12
3050 W=1
```

```
3060 X=100
3070 Y=133
3080 L=0
3090 GS=5
3100 FS=5
3110 ZS=5
3120 GR=110
3130 I=0
3140 J=0
3150 CO=10
3160 KI=0
3170 KK=0
3180 TR=0
3190 UT=122
3200 UH=48
3210 IF MS=1 THEN PLAY "l64cr64er64dr64f
r64ar64f":MS=2
3220 IF SM=1 THEN PLAY "l64cr64er64dr64f
r64ar64f":SM=2:GOSUB 4860
3230 IF PU>490 THEN RZ=8:RF=11:RG=10:UT=
115:UH=55
3240 REM Setzen der Sprites an die
        Anfangsposition
3250 PUT SPRITE 0,(X,Y),10,0
3260 PUT SPRITE 1,(K,139),2,9
3270 PUT SPRITE 2,(H+90,139),2,9
3280 PUT SPRITE 3,(G,110+GR),5,GS
3290 PUT SPRITE 4,(F+82,110+FR),4,FS
3300 PUT SPRITE 7,(Z+164,110+ZR),3,ZS
3310 PUT SPRITE 5,(I+30,40),2,10
3320 PUT SPRITE 6,(J+180,40),2,10
3330 PUT SPRITE 8,(12,4),11,PR(1)
3340 PUT SPRITE 9,(30,4),11,PR(2)
3350 PUT SPRITE 10,(48,4),11,PR(3)
3360 REM Loeschen der Kirsche (fest),
        falls vorhanden
3370 SPRITE ON
3380 ON SPRITE GOSUB 4290
3390 PSET (KX,KY),1
3400 DRAW B$
```

```
3410 REM Neues Setzen der Kirsche (fest)
3420 TIME=RND(-TIME)*65535!
3430 KX=20+RND(-TIME)*225
3440 KY=44+RND(-TIME)*90
3450 PSET (KX,KY),0
3460 DRAW "c8"+B$
3470 REM Hauptprogramm
3480 A%=STICK(S)
3490 REM Essen der Kirschen
3500 IF TI<>-1 AND KX-2>X AND KX-2<X+16
AND KY+4>Y AND KY+4<Y+16 THEN GOSUB 4580
3510 IF TI=-1 THEN PSET (KX,KY),1:DRAW B
$
3520 IF TI=20 THEN GOSUB 4510
3530 REM Essen der Bonus-Kirschen
3540 IF GR=209 AND FR=209 AND ZR=209 THE
N PLAY "som2000o4l16br7br5g":SPRITE ON:T
I=0:ET=0:GOTO 2940
3550 IF KI=1 AND GR+4>=Y AND GR+4<=Y+16
AND G+4>=X AND G+4<=X+16 THEN RG=0:GR=20
9:KK=1:GOSUB 4580
3560 IF KI=1 AND FR+4>=Y AND FR+4<=Y+16
AND F+4>=X AND F+4<=X+16 THEN RF=0:FR=20
9:KK=1:GOSUB 4580
3570 IF KI=1 AND ZR+4>=Y AND ZR+4<=Y+16
AND Z+4>=X AND Z+4<=X+16 THEN RZ=0:ZR=20
9:KK=1:GOSUB 4580
3580 REM Setzen der Felsvorspruenge
3590 IF Y=131 AND OT=1 THEN GOTO 4290
3600 IF KI<>1 THEN K=K-7:PUT SPRITE 1,(K
,139),2,9:IF K<0 THEN K=256
3610 IF KI=1 THEN PUT SPRITE 1,(0,209),2
,9
3620 IF KI<>1 THEN H=H-7:PUT SPRITE 2,(H
+90,139),2,9:IF H<0 THEN H=256
3630 IF KI=1 THEN PUT SPRITE 2,(0,209),2
,9
3640 IF KI<>1 THEN CO(1)=5:GS=GS+1 ELSE
GS=11:CO(1)=6
```

```
3650 REM Bewegen der Voegel
3660 GR=GR+RG
3670 G=G-11
3680 PUT SPRITE 3,(G,GR),CO(1),GS
3690 IF G<0 THEN G=256
3700 IF KI<>1 THEN IF GS>6 THEN GS=5
3710 IF GR>UT THEN RG=-RG ELSE IF GR<UH
THEN RG=-RG
3720 IF KI<>1 THEN CO(2)=4:FS=FS+1 ELSE
FS=11:CO(2)=9
3730 FR=FR-RF
3740 F=F-11
3750 PUT SPRITE 4,(F,FR),CO(2),FS
3760 IF F<0 THEN F=256
3770 IF KI<>1 THEN IF FS>6 THEN FS=5
3780 IF FR>UT THEN RF=-RF ELSE IF FR<UH
THEN RF=-RF
3790 IF KI<>1 THEN ZS=ZS+1 ELSE ZS=11
3800 ZR=ZR-RZ
3810 Z=Z-11
3820 PUT SPRITE 7,(Z,ZR),CO(1),ZS
3830 IF Z<0 THEN Z=256
3840 IF KI<>1 THEN IF ZS>6 THEN ZS=5
3850 IF ZR>UT THEN RZ=-RZ ELSE IF ZR<UH
THEN RZ=-RZ
3860 IF KI<>1 THEN I=I-7:PUT SPRITE 5,(I
+30,40),2,10:IF I<0 THEN I=256
3870 IF KI=1 THEN PUT SPRITE 5,(0,209),2
,10
3880 IF KI<>1 THEN J=J-7:PUT SPRITE 6,(J
+180,40),2,10:IF J<0 THEN J=256
3890 IF KI=1 THEN PUTSPRITE 6,(0,209),2,
10
3900 REM Steuerung der Hoehe des Vogels
3910 IF STRIG(S) THEN RU=-6 ELSE IF S=2
THEN RU=16 ELSE RU=9
3920 IF STRIG(S) AND A%=0 THEN Y=Y-2:GOS
UB 4180
3930 IF Y<178 AND A%=0 AND STRIG(S)=0 TH
EN Y=Y+2:GOSUB 4180
```

```
3940 REM Vogel Kopf nach rechts
3950 IF A%=0 THEN W=1
3960 REM Bewegung des Vogels (Joystick-
        abfrage) und Laufrhythmus
3970 ON A% GOSUB 3980,3990,4020,4050,408
0,4090,4120,4150:GOTO 3480 ELSE GOTO 393
0
3980 GOTO 4180
3990 W=1
4000 X=X+R
4010 GOTO 4180
4020 W=1
4030 X=X+R
4040 GOTO 4180
4050 W=1
4060 X=X+R
4070 GOTO 4180
4080 GOTO 4180
4090 W=2
4100 X=X-R
4110 GOTO 4180
4120 W=2
4130 X=X-R
4140 GOTO 4180
4150 W=2
4160 X=X-R
4170 GOTO 4180
4180 IF W=1 AND Y>=131 THEN SP=SP+1:IF S
P>1 THEN SP=0
4190 IF W=2 AND Y>=131 THEN SP=SP+1:IF S
P>5 THEN SP=4
4200 IF W=1 AND Y<131 THEN SP=SP+1:IF SP
>3 THEN SP=2
4210 IF W=2 AND Y<131 THEN SP=SP+1:IF SP
>7 THEN SP=6
4220 Y=Y+RU
4230 IF Y>131 THEN Y=131
4240 IF X>240 THEN X=240
4250 IF Y<40 THEN Y=40
4260 IF X<8 THEN X=8
```

```
4270 PUT SPRITE 0,(X,Y),CO,SP
4280 RETURN
4290 SPRITE OFF
4300 REM Lebensverlust
        Abfrage, ob tot oder nicht
4310 S=2
4320 IF Y=131 THEN PUT SPRITE 0,(X,134),
14,8:PLAY "124som9000o1c15o5som2000":FOR
 A=1 TO 2000:NEXT:S=SZ:OT=0:GOTO 4330 EL
SE S=2:OT=1:CO=14:PLAY "124som9000o2a":G
OTO 3480
4330 EN=EN-1
4340 GOSUB 4840
4350 IF EN<1 THEN DRAW "bm 100,96":PRINT
 #1,"Game over":IF PU>HC THEN HC=PU:COLO
R 2:DRAW "bm 213,10":PRINT #1,STRING$(6,
CHR$(200)):DRAW "bm 213,10":COLOR 15:PRI
NT #1,HC:GOTO 4370
4360 IF EN>=1 THEN GOTO 2940
4370 DRAW "bm 30,170"
4380 PRINT #1,"Bitte eine Taste druecken
"
4390 A$=INKEY$
4400 IF A$<>"" THEN GOTO 4390
4410 A$=INKEY$
4420 IF A$="" THEN GOTO 4410
4430 MU=1
4440 GOTO 2610
4450 REM Spielbeendigung und Rueckkehr
        auf einen 'sauberen' SCREEN 0
4460 SCREEN 0,,1
4470 COLOR 15,4,5
4480 KEY OFF
4490 END
4500 REM Umwandlung der Voegel in
        Kirschen
4510 SPRITE OFF
4520 ET=1
4530 TI=-1
4540 KI=1
```

```
4550 PLAY "15o5fco4bag"
4560 RETURN
4570 REM Punkte
4580 IF KK=1 THEN TI=TI-1:TR=1
4590 PLAY "15som2000o5go4"
4600 TI=TI+1
4610 PU=PU+10
4620 DRAW "bm 118,10"
4630 COLOR 2
4640 PRINT #1,STRING$(7,CHR$(200))
4650 COLOR 15
4660 DRAW "bm 118,10"
4670 PRINT #1,PU
4680 REM Abfrage, ob Bonus
4690 IF PU=500 AND MS=0 THEN MS=1:GOTO 2
940
4700 REM Abfrage, ob Bonus. Wenn Bonus,
        so wird das Spiel schwieriger.
4710 IF PU=700 AND SM=0 THEN SM=1:GOTO 2
940
4720 IF TR=1 THEN RETURN
4730 REM Setzen der Kirsche fest
4740 LINE (KX-8,KY-1)-(KX+5,KY+11),1,BF
4750 TIME=RND(-TIME)*65535!
4760 KX=20+RND(-TIME)*225
4770 KY=44+RND(-TIME)*90
4780 PSET (KX,KY),1
4790 DRAW "c8"+B$
4800 KK=0
4810 TR=0
4820 RETURN
4830 REM Anzeige der noch vorhandenen
        Leben
4840 IF EN=3 THEN PR(1)=8 ELSE IF EN=2 T
HEN PR(2)=8 ELSE IF EN=1 THEN PR(3)=8
4850 RETURN
4860 IF EN=4 THEN RETURN
4870 IF EN=3 THEN PR(1)=0 ELSE IF EN=2 T
HEN PR(2)=0 ELSE IF EN=1 THEN PR(3)=0
4880 EN=EN+1
```

```
4890 RETURN
4900 REM Anfangsmelodie
4910 PLAY "t255som2000o4l16br7br5gr3br7b
r5gr3"
4920 PLAY "15o5fco4bagr50"
4930 PLAY "164ar7ar5fr3ar7ar5fr3"
4940 RETURN
4950 REM Erklaerung
4960 SCREEN 0:WIDTH 40
4970 LOCATE 0,0
4980 PRINT TAB(13) "HOLLOW"
4990 PRINT TAB(13) "======"
5000 PRINT
5010 PRINT "Los geht sie, die froehliche
 Jagd nach  Kirschen. Sie muessen mit ei
nem Vogel   moeglichst viele davon fress
en."
5020 PRINT "Sie duerfen dabei allerdings
 nicht mit  den anderen Voegeln zusammen
stossen."
5030 PRINT "Immer wenn Sie 20 Kirschen g
efressen    haben, verwandeln sich die V
oegel, als  Bonus, in Kirschen, die Sie
fressen      koennen. Wenn Sie 500 Punkte
 haben,     wird das Spiel schwieriger."
5040 PRINT "Wenn Sie 700 Punkte und nich
t mehr      alle Leben haben, bekommen S
ie ein      Zusatzleben."
5050 PRINT "Gesteuert wird ganz normal m
it Joystick oder Tastatur, nur 'rauf' ge
ht's mit demFeuerknopf bzw. der Leertast
e. Runter"
5060 PRINT "geht's aufgrund der Schwerkr
aft von     selbst."
5070 PRINT:PRINT "Bitte eine Taste druec
ken ..."
5080 C$=INKEY$:IF C$="" THEN GOTO 5080 E
LSE RUN
```

Reaktion

Um Leute zu überraschen, vielleicht auch, um Leute, die bisher der ganzen Computerei eher negativ gegenüberstehen, positiver auf diese Wundermaschinen umzustimmen, sei dieses Programm zur Vorführung dringendst empfohlen.

Der ganze Inhalt, was die Programmierung an und für sich betrifft: Es muß eine Taste so schnell wie nur irgend möglich gedrückt werden, wenn der Computer dazu auffordert. Im Nachhinein erfolgt eine Wertung und es wird ein weiteres Mal zu einem neuen Spiel mit bis zu vier Teilnehmern aufgefordert.

Der Clou dieses Programms liegt vielmehr darin, wie mit dem nichts ahnenden Computerbediener umgegangen wird. Wie bereits an anderer Stelle in diesem Buch gesagt: Die Computer sind alle sehr dumm. Sie werden erst durch die Programmierung von uns Menschen schlau gemacht. Gerade dies wird dem laienhaften Benutzer beim Ablauf dieses Programms klar gemacht:

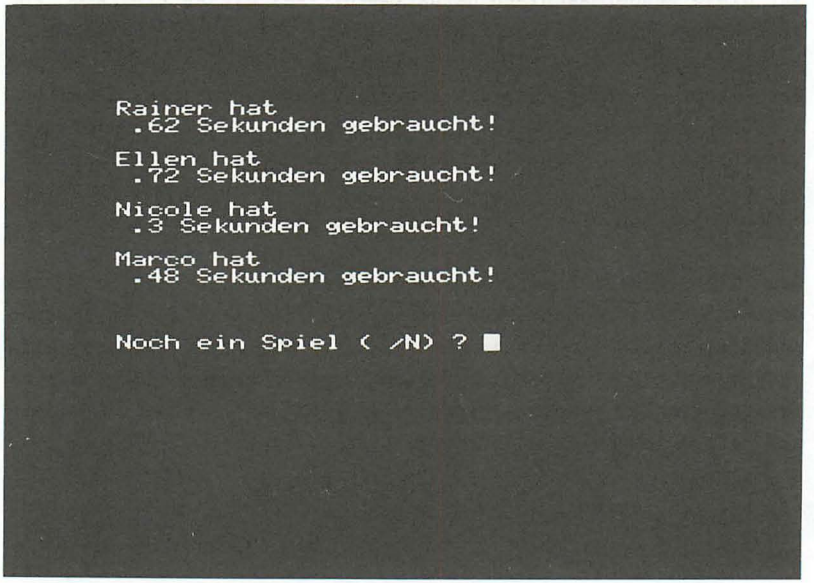
Erfolgt zu Beginn keine Namenseingabe, wird Ihr MSX-Computer bzw. dieses Programm langsam aber sicher richtig knartschig ... bis er gar droht, man mache den Computer durch Fehler kapputt! Hat jedoch alles seine Ordnung, antwortet Ihr MSX-Computer artig mit 'Danke'.

Beim Spielablauf achtet Ihr MSX-Computer ordentlich darauf, ob nicht vielleicht geschummelt wird. Ist dies der Fall, erfolgt eine Ermahnung und eine Zeitstrafe.

Die gemessene Zeit wird nicht nur bis zum hundertsten Bruchteil einer Sekunde genau angezeigt, nein Ihr MSX-Computer gibt auch seinen Kommentar dazu ('S U P E R' ... 'sehr schwach').

Programmtechnisch gesehen ein Klacks! Dafür aber vielleicht als 'Entschuldigung' für den sehr vernünftigen (!) Kauf Ihres MSX-Computers für all die lieben Mitmenschen gedacht. Vielleicht machen Sie so auch noch Computerfreaks aus ihnen, dann ist die Freude und Begeisterung halt mindestens doppelt so groß wie

zuvor. Noch ein Verwendungszweck dieses Programms: Messen Sie mal die Reaktionszeiten in verschiedenen Gemütslagen und Zuständen!



```
10 REM Reaktion
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 REM Vorbedingungen treffen
50 COLOR 15,1,1:SCREEN 0:WIDTH 40:KEY OF
F
60 INPUT "Wie viele Teilnehmer (1 bis 4)
 ";F$
70 IF VAL(F$)<1 OR VAL(F$)>4 THEN GOTO 6
0
80 FOR N=1 TO VAL(F$)
90 PRINT:PRINT "Ihr Name, bitte, Spieler
";N;:INPUT F1$(N)
100 REM Reaktion auf falsche Eingabe
110 IF F1$(N)="" AND ME=0 THEN PRINT:PRI
NT "Seien Sie doch nicht feige!
    Jede Sache hat auch einen Namen. Sow
eit ich dummer Computer weiss, gilt das
    auch fuer Menschen und Tiere!":ME=1:
GOTO 90
120 IF F1$(N)="" AND ME=1 THEN PRINT:PRI
NT "Diese Menschen, diese Menschen!
    Gehen wir also mal behutsam ans Werk
... vielleicht klappt es so! Ich heisse
    Standard. Nein, ich bin nicht vom Ma
rs. Mein Vorname ist MSX (komisch, wie?)
."
130 IF F1$(N)="" AND ME=1 THEN PRINT "Du
 darfst mich ruhig 'Homemicro' nennen ..
. aber nur wenn Du mir sagst, wie Du  mi
t Vornamen heisst. Also bitte!":ME=2:GOT
O 90
140 IF F1$(N)="" AND ME=2 THEN PRINT:PRI
NT "Noch eine Fehleingabe und der Comput
er  ist kapputt ... ich darf doch bitten
 ...";:ME=3:GOTO 90
150 IF F1$(N)="" AND ME=3 THEN FOR N=1 T
O 10000:PRINT CHR$(PEEK(N));:NEXT
```

```
160 ME=0
170 NEXT N
180 CLS:LOCATE 16,11:PRINT "Danke"
190 FOR N=1 TO 2000:NEXT N
200 CLS:FOR N=1 TO VAL(F$)
210 PRINT F1$(N);"!!!!":PRINT:PRINT:PRIN
T "Du bist dran!":PRINT "Druecke eine me
iner Tasten, wenn Du     bemerkt hast, d
ass sich die             Bildschirmfarbe
 geaendert hat."
220 REM Berechnung der Zeitspanne
230 FOR M=1 TO 2000:NEXT M:A=1000*RND(TI
ME):FOR M=1 TO A:A$=INKEY$:IF A$="" THEN
 NEXT M:GOSUB 300 ELSE CLS:PRINT "Schumm
ler!!!!":PRINT "Du hast 5 Sekunden":PRIN
T "gebraucht.":F(N)=5:FOR M=1 TO 2000:NE
XT M:CLS
240 REM Endergebnis ausgeben
250 NEXT N:CLS:FOR N=1 TO VAL(F$):PRINT
F1$(N);" hat":PRINT F(N);"Sekunden gebra
ucht!":PRINT:NEXT N:PRINT:PRINT:INPUT "N
och ein Spiel ( /N) ";Z$:IF LEFT$(Z$,1)=
"N" OR LEFT$(Z$,1)="n" THEN Z$="N"
260 IF Z$="N" THEN END
270 PRINT:INPUT "Mit den gleichen Spiele
rn (J/ ) ";Z$:IF LEFT$(Z$,1)="j" OR LEFT
$(Z$,1)="J" THEN Z$="J"
280 IF Z$="J" THEN GOTO 200 ELSE RUN
290 REM Unterprogramm zur Zeitkontrolle
300 CLS:COLOR 1,15:B=TIME
310 A$=INKEY$:IF A$="" THEN GOTO 310
320 C=TIME:COLOR 15,1:CLS:F(N)=(C-B)/50:
PRINT F1$(N):PRINT "Du hast";F(N);"Sekun
den gebraucht!"
330 PRINT
```

```
340 REM Wertung
350 IF F(N)<.3 THEN PRINT "S U P E R":GO
TO 390
360 IF F(N)<.35 THEN PRINT "Ganz gut":GO
TO 390
370 IF F(N)<.4 THEN PRINT "schwach":GOTO
 390
380 PRINT "Sehr, sehr schwach!"
390 FOR M=1 TO 2000:NEXT M:CLS:RETURN
```

Codeknacker

Dieses Spiel gibt es in mannigfaltigen Versionen z.B. unter den Namen 'Superhirn' und 'Mastermind'.

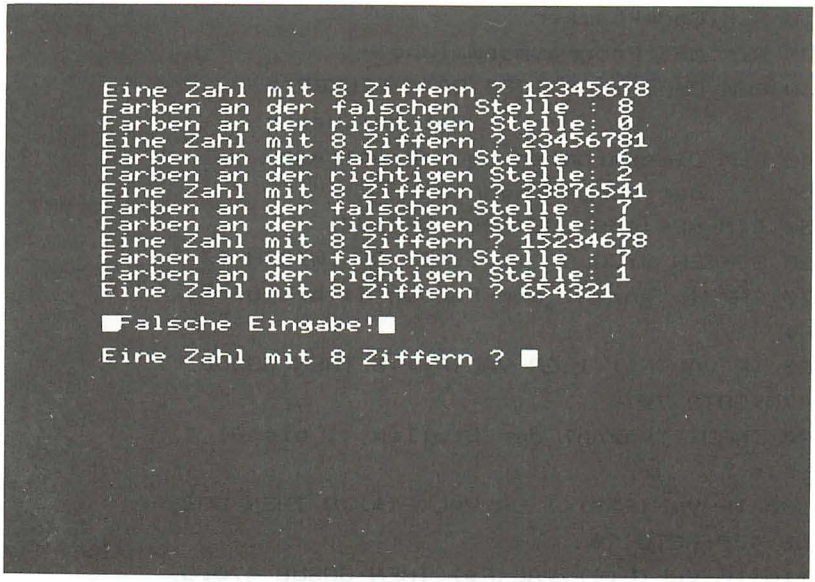
Wir haben das Computerprogramm jedoch umbenannt in 'Codeknakker', weil es sich von den Vorbildern doch in gewisser Weise stark unterscheidet.

Zu Beginn können Sie wählen, aus wieviel Farben=Zahlen die zu erratende Folge zusammengesetzt sein soll. Es stehen bis zu acht Farben=Zahlen zur Verfügung.

Anschließend erfolgt eine zweite Wahl: Aus wieviel Teilen soll die Folge bestehen? Es ist zwar möglich mehr Farben als Teile zu bestimmen, allerdings ist das umgekehrte Verhältnis nicht vorgesehen und wird somit ignoriert. Dies hat folgenden Grund: Der Spieler soll zu Beginn wissen, mit welcher Planung er es zu tun hat; so ist es klar, daß nach der Auswahl von jeweils acht Farben und acht Stellen auch wahrlich acht Farben in der Folge auftreten; Doppel oder gar Dreier sind somit ausgeschlossen!

Aus diesem Grund wird während des Spielablaufs ähnlich weiterverfahren: es nützt nicht, durch Vorgabe z.B. der Zahlenkombination '11112222' herausfinden zu wollen, ob die '1' bzw. '2' gleich mehrfach und an den entsprechenden Stellen in der Folge vorkommt.

Es wird sowohl auf dem Bildschirm angezeigt, wie viele Farben richtig aber auch wie viele gar genau an der richtigen Stelle positioniert wurden. Für den 'Codeknacker'-Anfänger ein recht schwieriges Unterfangen. Aber Übung macht den Meister. Um den Überblick nicht zu verlieren, wird der Bildschirm nach jeder Eingabe hochgescrollt (die letzten vier bis fünf Eintragungen bleiben oben sichtbar). Zudem werden alle Ergebnisse und Zahleingaben festgehalten. Man kann die gesamte Eingabefolge durch Drücken der (ENTER)-Taste wieder auf den Bildschirm bringen. Nun aber genug der Vorstellung. Viel Spaß!


Eine Zahl mit 8 Ziffern ? 12345678
Farben an der falschen Stelle : 8
Farben an der richtigen Stelle: 0
Eine Zahl mit 8 Ziffern ? 23456781
Farben an der falschen Stelle : 6
Farben an der richtigen Stelle: 2
Eine Zahl mit 8 Ziffern ? 23876541
Farben an der falschen Stelle : 7
Farben an der richtigen Stelle: 1
Eine Zahl mit 8 Ziffern ? 15234678
Farben an der falschen Stelle : 7
Farben an der richtigen Stelle: 1
Eine Zahl mit 8 Ziffern ? 654321
Falsche Eingabe!
Eine Zahl mit 8 Ziffern ?

```
10 REM Codeknacker
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 REM Dimensionierung und Abfrage
       der Vorbedingungen
50 DIM ME$(100),SC(100),WE(100)
60 SCREEN 0:COLOR 15,1:KEY OFF:WIDTH 40
70 INPUT "Anzahl der Farben (1 bis 8) ";
F$
80 IF VAL(F$)<1 OR VAL(F$)>8 THEN GOSUB
470:GOTO 70
90 INPUT "Anzahl der Stellen (1 bis 8) "
;F1$
100 IF VAL(F1$)<1 OR VAL(F1$)>8 THEN GOS
UB 470:GOTO 90
110 IF VAL(F1$)>VAL(F$) THEN GOSUB 470:G
OTO 90
120 REM Auswahl der Zahlen durch die
        RND=Zufallsfunktion
130 FOR N=1 TO VAL(F1$)
140 A=INT(10*RND(TIME))
150 IF A<1 OR A>VAL(F$) THEN GOTO 140
160 REM Alle ausgewaehlten Zahlen
        sollen unterschiedlich sein
170 FOR M=1 TO N-1
180 A(M)=VAL(MID$(A$,M,1))
190 NEXT M
200 FOR M=1 TO N-1
210 IF A=A(M) THEN GOTO 140 ELSE NEXT M
220 A$=A$+RIGHT$(STR$(A),1)
230 NEXT N
240 FOR N=1 TO VAL(F1$)
250 A(N)=VAL(MID$(A$,N,1))
260 NEXT N
270 CLS
```

```
280 REM Eingaberoutine
290 F3$="":PRINT "Eine Zahl mit";VAL(F1$
);"Ziffern ";:INPUT F3$
300 REM Auflistung aller Zahlen, die
        bisher eingegeben wurden,
        wenn keine Eingabe erfolgt
310 IF F3$="" THEN CLS:FOR Z1=1 TO Z:PRI
NT "Zug"Z1;TAB(7);ME$(Z1);"    OK:";SC(Z
1);"  not OK:";WE(Z1):NEXT Z1:PRINT:GOTO
 290 ELSE IF LEN(F3$)<>VAL(F1$) THEN GOS
UB 470:GOTO 290
320 FOR N=1 TO VAL(F1$)
330 B(N)=VAL(MID$(F3$,N,1))
340 IF A(N)=B(N) THEN SC=SC+1:C(N)=-1
350 NEXT N
360 FOR M=1 TO VAL(F1$)
370 FOR N=1 TO VAL(F1$)
380 IF A(N)=B(M) AND N<>M THEN WE=WE+1:G
OTO 410
390 NEXT N
400 REM Ausgaberoutine und
        anschliessende Ueberpruefung
        auf Richtigkeit
410 NEXT M:PRINT "Farben an der falschen
 Stelle :";WE
420 Z=Z+1:ME$(Z)=F3$:WE(Z)=WE:SC(Z)=SC:W
E=0
430 PRINT "Farben an der richtigen Stell
e:";SC
440 IF SC=VAL(F1$) THEN PRINT:PRINT CHR$
(200) "Genau richtig in";Z;"Zuegen" CHR$
(200):PRINT:INPUT "Nochmal (J/ ) ";F$:IF
 LEFT$(F$,1)="J" OR LEFT$(F$,1)="j" THEN
 F$="J" ELSE GOTO 450 ELSE SC=0:GOTO 290
450 IF F$="J" THEN RUN ELSE END
460 GOTO 290
470 PRINT:PRINT CHR$(200) "Falsche Einga
be!" CHR$(200):PRINT:RETURN
```

Zahlsystemumrechner

Ihr MSX-Computer kann nicht nur Zahlen in dem uns vertrauten Dezimalsystem abbilden, er hat auch spezielle Befehle zum Umrechnen in andere Zahlsysteme so z.B. bedeutet '&H' einer Zahl vorangestellt: Hexadezimalzahl; '&B' bedeutet Binär- bzw. Dualzahl.

Auf diese Art und Weise können wir 'PRINT &HFF' eingeben und bekommen als Ergebnis dieser Hexadezimalzahl umgerechnet in eine Dezimalzahl '255' auf dem Bildschirm ausgegeben.

Geben wir 'PRINT &B111' ein, so erhalten wir gar als Ergebnis dieser Binärzahl umgerechnet in eine Dezimalzahl: '7'.

Schwierig wird es schon, wenn man direkt von einer Hexadezimalzahl in eine Binärzahl umrechnen will. Unmöglich wird es gar, wenn wir mit einem anderen geläufigen Zahlsystem außer hexadezimal, dezimal oder binär (z.B. oktal) arbeiten wollen.

Um die Zahlsysteme besser kennenzulernen ist nun dieses Programm entstanden. Sie können durch Voranstellung des entsprechenden Buchstabens ('H' für Hexadezimal, 'B' für Binaer, 'D' für Dezimal) jede eingegebene Zahl in jedem dieser drei Zahlsysteme darstellen lassen.

Außerdem können Sie durch Drücken der (ENTER)-Taste Ihre Dezimaleingaben in ein Zahlsystem zwischen 2 und 9 umrechnen. Hierbei geschieht etwas interessantes und gleichzeitig lehrreiches: Die Zahl in dem gewünschten Zahlsystem wird vor Ihren Augen ausgerechnet - Schritt für Schritt.

Wie geht das vor sich? Wir wählen ein Beispiel ... rechnen wir im Fünfer-Zahlensystem. Drücken Sie also (ENTER) auf die Frage nach 'Dez.,Hex. o. Bin.'. Auf die nächste Frage antworten Sie mit '5' und geben nun z.B. die Zahl '230' ein. Die ausgerechnete Zahl im Fünfersystem ist die '1410'.

Nun zur Erklärung des Vorgangs: Im Fünfer-Zahlsystem haben wir die Ziffern '0, 1, 2, 3 und 4'. Ihr MSX-Computer teilt unsere eingegebene Zahl '230' durch 5. Ergebnis: '46' Rest '0'. Anschließend wird die '46' geteilt: '9' Rest '1'. Nun wird die '9' geteilt: '1' Rest '4' und so bleibt schließlich ein Rest '4' übrig. Schreiben wir uns die Restwerte noch einmal auf:

1. 0 2. 1 3. 4.

Beim letzten Wert nehmen wir nicht nur den Rest '4', sondern auch die Anzahl = '1'.

So erhalten wir für Dezimal '230' die Zahl im Fünfersystem: '1410'. Probe gefällig? Hierzu bilden wir Fünfer-Potenzen von rechts nach links:

```
1) 0 * 5↑0 = 0    2) 1 * 5↑1 = 5    3) 4 * 5↑2 = 100
4) 1 * 5↑3 = 125  -) 0 + 5 + 100 + 125 = 230 !
```

Nur der Vollständigkeit halber erwähnt: Auch ohne das Programm 'Zahlsystemumrechner' kann man mit dem MSX-Basic immerhin Dezimal-Hexadezimal und Dezimal-Binär umrechnen und zwar mit den Befehlen: 'HEX$(' und 'BIN$('.

```
Eingabe (Dez.,Hex.o.Bin.) ? d255
dezimal   binaer oder dual    hexadezimal
-------------------------------------------
255       00000000 11111111      00FF

Eingabe (Dez.,Hex.o.Bin.) ? hffff
dezimal   binaer oder dual    hexadezimal
-------------------------------------------
 65535    11111111 11111111      FFFF

Eingabe (Dez.,Hex.o.Bin.) ? b111
dezimal   binaer oder dual    hexadezimal
-------------------------------------------
 7        00000000 00000111      0007

Eingabe (Dez.,Hex.o.Bin.) ? ■
```

```
Umrechnung DEZ. - 5 ersystem
============================

Ihre Eingabe (DEZ.) ? 230

Anzahl: 46     MOD: 0
Anzahl: 9      MOD: 1
Anzahl: 1      MOD: 4

 1410


Welches Zahlsystem (2 bis 9) ? ■
```

```
10 REM Zahlsystemumrechner
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 COLOR 15,1:SCREEN 0:KEY OFF
50 ON ERROR GOTO 410
60 WIDTH 40
70 REM Auswahl zwischen den Zahlsystemen
80 A$="":INPUT "Eingabe (Dez.,Hex.o.Bin.
) ";A$:IF VAL(A$)>65535! THEN GOTO 80 EL
SE IF A$="" THEN GOTO 160 ELSE B$=LEFT$(
A$,1):IF B$="H" OR B$="h" THEN GOSUB 310
 ELSE IF B$="B" OR B$="b" THEN GOSUB 360
 ELSE IF B$="D" OR B$="d" THEN A$=RIGHT$
(A$,LEN(A$)-1)
90 REM Berechnung der Dezimal-, Binaer-
       und Hexadezimalzahlen fuer den
       Bildschirmausdruck
100 PRINT:PRINT "dezimal  binaer oder du
al   hexadezimal"
110 PRINT STRING$(39,"-")
120 PRINT A$;TAB(8);" ";
130 B$=STRING$(16-LEN(BIN$(VAL(A$))),"0"
)+BIN$(VAL(A$)):PRINT LEFT$(B$,8);" ";RI
GHT$(B$,8);"  ";
140 PRINT TAB(31);STRING$(4-LEN(HEX$(VAL
(A$))),"0")+HEX$(VAL(A$))
150 PRINT:PRINT:GOTO 80
160 CLS
170 REM Auswahl zwischen den
        Zahlsystemen 2 bis 9
180 A=0:INPUT "Welches Zahlsystem (2 bis
 9) ";A
190 IF A=0 THEN RUN ELSE IF A<2 OR A>9 T
HEN GOSUB 440:GOTO 160
200 CLS:PRINT "Umrechnung DEZ. -";A;"ers
ystem":PRINT STRING$(28,"="):PRINT
```

```
210 REM Berechnung der Zahlen im
        gewaehlten Zahlsystem (2 bis 9)
220 B=0:INPUT "Ihre Eingabe (DEZ.) ";B:I
F B>32767 THEN GOSUB 440:GOTO 160 ELSE P
RINT
230 C=B\A
240 PRINT "Anzahl:";C;
250 D=B MOD A
260 PRINT TAB(14);"MOD:";D
270 A$=RIGHT$(STR$(D),1)+A$
280 IF C<A THEN A$=STR$(C)+A$:PRINT:PRIN
T A$:A$="":PRINT:PRINT:PRINT:GOTO 180
290 B=C:GOTO 230
300 REM Eingabe der Hexadezimalzahl
        ueberpruefen und umrechnen
310 IF LEN(A$)>5 THEN RETURN
320 B$="&h"+MID$(A$,2,LEN(A$)-1)
330 IF VAL(B$)<0 THEN A$=STR$(VAL(B$)+65
536!) ELSE A$=STR$(VAL(B$))
340 RETURN
350 REM Eingabe der Binaerzahl
        ueberpruefen und umrechnen
360 IF LEN(A$)>17 THEN RETURN
370 B$="&b"+MID$(A$,2,LEN(A$)-1)
380 IF VAL(B$)<0 THEN A$=STR$(VAL(B$)+65
536!) ELSE A$=STR$(VAL(B$))
390 RETURN
400 REM Errorbehandlungsroutine
410 RESUME 420
420 RUN
430 END
440 PRINT TAB(10) CHR$(200) "Falsche Ein
gabe!" CHR$(200)
450 GOSUB 460:RETURN
460 PRINT:PRINT TAB(6) "<Bitte eine Tast
e druecken>"
470 F$=INKEY$:IF F$="" THEN GOTO 470
480 RETURN
```

Disassembler

In den Programmen 'Speicher 1' bis 'Speicher 5' sowie im Variablen-Referenzlistenprogramm haben wir uns bereits ein klein wenig mit den 'Innereien' unseres MSX-Computers beschäftigt. Unter 'Innereien' sei hier verstanden: Wie geht es überhaupt vor sich, daß ein unmenschliches Wesen wie der Computer zu denken fähig ist (merken Sie sich dazu folgendes: Ein Computer ist immer nur so schlau, wie die Menschen, die ihn programmiert haben).

In den Programmen 'Speicher 1' bis 'Speicher 5' sind wir aber noch nicht tief genug in unseren MSX-Computer eingedrungen. Wir haben uns seinerzeit nämlich nur angeschaut, wie unsere BASIC-Programme Zeilennummer für Zeilennummer, Befehl für Befehl, im Speicher abgelegt werden, aber nichts über die Verarbeitung zu den Befehlen und zu dieser Art der Abspeicherung hin kennengelernt. Dabei wurde aber bereits erwähnt, daß der MSX-Computer wie jeder andere Computer auch auf die Grundstellung zurückgelangt: an oder aus bzw. in binärer Form: 0 oder 1. Aus dem Erkennen einer Vielzahl von aufeinanderfolgenden 0 en oder 1 en erkennt der Computer dann ein Zeichen, manchmal auch einen Befehl.

Unser BASIC nennt sich eine höhere Programmiersprache: sie ist schon mit einer Vielzahl verständlicher Begriffe aufgebaut, so daß wir bei der Benutzung dieser Befehle vielmehr eine Vorstellung davon bekommen, was der Computer tun kann, als wenn wir nur stur 0 en oder 1 en in den Speicher eingeben würden.

Der in diesem Programm untersuchte Assemblercode ist eine Stufe zwischen BASIC und Maschinensprache. Wir haben hier eine Vielzahl von Befehlen zur Verfügung (beim Prozessor Z80A im MSX-Computer sind es gar mehr als 600), die allerdings bei weitem nicht so wirkungsvoll arbeiten wie die BASIC-Befehle. Im Speicher benötigen diese Befehle teils 1 Byte, teils 2 Bytes, manchmal auch gar 3 Bytes (1 Byte = 1 Speicherplatz). Das Programm 'Disassembler' hat sämtliche Z80A-Assemblercodes (man nennt sie auch Mnemoniks) gespeichert und untersucht den von uns

vorgegebenen Speicherraum nach deren Vorkommen. Dabei untersucht es das Vorkommen bestimmter Zeichenfolgen (ähnlich wie bei dem 'Speicher 4'-Programm, wo wir nach der Zeichenfolge 'Laenge' gesucht haben); werden diese Zahlenfolgen irgendwo angetroffen, wird der entsprechende Assemblercode auf den Bildschirm (oder Drucker) geschrieben.

Da wir nicht davon ausgehen können, daß Sie Tag für Tag sehr viele Maschinenprogramme schreiben, haben wir einen Lerneffekt in unser 'Disassembler'-Proramm eingebaut: In Zeile 3530 steht eine DATA-Anweisung, gefolgt von meheren Hexadezimalzahlen. Diese Zahlen sind ein kleines Maschinenprogramm, in dem zwei Zahlen zusammengezählt und an einer bestimmten Speicherstelle abgelegt werden. Sie können sich die Umwandlung dieser Zahlen in ein Assemblerprogramm anschauen, indem Sie nach dem Programmstart auf alle Fragen ohne weitere Eingabe nur mit (ENTER) antworten. Nur auf die Frage nach dem Ausgabemedium müssen Sie mit 'B' wie Bildschirm antworten, falls Sie keinen schriftlichen Ausdruck benötigen.

Sie können nun auch andere Zahlen (in hexadezimaler Form, bitte!) in die DATA-Zeile schreiben - oder Sie finden ein längeres Maschinenprogramm für Z80A-Prozessor in einer Computerzeitschrift ... Sie können auch ruhig mehrere DATA-Zeilen mit Zahlen anfüllen ... Viel Spaß dabei!

```
10 REM Disassembler
20 REM MSX Programmsammlung
30 REM Copyright 1984 DATA BECKER &
                      Rainer Lueers
40 REM Speicherplatz ab 55000 fuer
       Maschinenroutine frei halten
50 CLEAR 2000,54999!:GOSUB 3480
60 DIM I$(255),BL(255):N=0:M=0:SY=1
70 GOTO 1290
80 SCREEN 0:WIDTH 40:COLOR 15,1:C=0:PRIN
T TAB(3) "* * * Z-80 Disassembler * * *"
90 REM Wuensche des Benutzers
       abklaeren
100 PRINT:INPUT "Titel ";F$
110 IF LEN(F$)>80 THEN GOTO 100
120 PRINT
130 REM Anfangsadresse der
        abgespeicherten Maschinenroutine
        in DATA-Zeilen
140 A=0:INPUT "Startadresse (in Dez.) ";
A:IF A<0 OR A>65535! THEN GOTO 120 ELSE
IF A=0 THEN A=55000!
150 INPUT "Endadresse (in Dez.)    ";B
160 IF B<E OR B>65535! OR B<0 THEN GOTO
150 ELSE IF B=0 THEN B=ZA
170 PRINT:INPUT "Datafelder vorhanden (J
/ ) ";V$
180 IF LEFT$(V$,1)="J" OR LEFT$(V$,1)="j
" THEN V$="J"
190 IF V$="J" THEN GOSUB 2150
200 PRINT:INPUT "Drucker oder Bildschirm
 (D/B) ";B$
210 REM Festlegen der Variablen 'b$',
        die fuer die Ansprache des
        PRINT- bzw. LPRINT-Befehls
        zustaendig ist
220 IF LEFT$(B$,1)="D" OR LEFT$(B$,1)="d
" THEN B$="D" ELSE IF LEFT$(B$,1)="b" OR
 LEFT$(B$,1)="B" THEN B$="B"
230 IF B$="D" THEN LPRINT ELSE PRINT
```

```
240 REM Ueberschrift = Titel, der durch
        die Formel in der Mitte der
        Zeile erscheint
250 IF B$="D" THEN LPRINT TAB((80-LEN(F$
))/2-5) F$ ELSE PRINT TAB((40-LEN(F$))/2
-5) F$
260 IF B$="D" THEN LPRINT ELSE PRINT
270 REM Beginn des eigentlichen
        Disassemblers. Suksessives
        Einlesen des angegebenen
        Speicherbereichs und
        Interpretieren der Daten
        in Maschinenbefehle=Mnemoniks
280 Q=A
290 H1=INT(A/4096):H2=INT((A-H1*4096)/25
6)
300 L1=INT((A-H1*4096-H2*256)/16)
310 L2=A-H1*4096-H2*256-L1*16
320 IF M<>0 THEN GOTO 340
330 GOTO 390
340 IF A>=A(N) AND A<=B(N) THEN GOTO 360
350 GOTO 390
360 D=PEEK(A):GOSUB 3390:F$="D":FL$="A":
GH$="T":GL$="A":T$=I$(D):I$(D)=""
370 IF A=B(N) THEN N=N+1:M=M-1:GOTO 400
380 GOTO 400
390 D=PEEK(A):GOSUB 660
400 V=D:GOSUB 2750
410 DH$=H$:DL$=L$
420 IF H1>=10 THEN GOTO 570
430 H1$=STR$(H1):H1$=MID$(H1$,2,1)
440 IF H2>=10 THEN GOTO 580
450 H2$=STR$(H2):H2$=MID$(H2$,2,1)
460 IF L1>=10 THEN GOTO 590
470 L1$=STR$(L1):L1$=MID$(L1$,2,1)
480 IF L2>=10 THEN GOTO 600
490 L2$=STR$(L2):L2$=MID$(L2$,2,1)
```

```
500 REM Ausgabe des disassemblierten
        Speicherbereichs
510 IF B$="D" THEN QT=QT+1:IF QT>105 THE
N QT=0:FOR QI=1 TO 9:LPRINT:NEXT QI:GOSU
B 3410:SY=SY+1:LPRINT TAB(30) "Seite";SY
:LPRINT:GOTO 510 ELSE LPRINT Q;LEN(DH$+D
L$+EH$+EL$+FH$+FL$+GH$+GL$)/2;"Bytes";TA
B(18);
520 IF B$="D" THEN LPRINT USING "   !!!!
";" ",H1$,H2$,L1$,L2$;:LPRINT "
  ";:LPRINT USING "!!!!!!!!";" ",DH$,DL$
,EH$,EL$,FH$,FL$,GH$,GL$;:LPRINT TAB(50)
;I$(D)
530 IF B$="B" THEN PRINT USING "!!!!";"
",H1$,H2$,L1$,L2$;:PRINT "  ";:PRINT USI
NG "!!!!!!!!";" ",DH$,DL$,EH$,EL$,FH$,FL
$,GH$,GL$;:PRINT TAB(15);I$(D)
540 A=A+1:C=C+1:IF A>=B+1 THEN GOTO 610
550 IF GH$="T" THEN I$(D)=T$
560 GOTO 280
570 X=H1-10+65:H1$=CHR$(X):GOTO 440
580 X=H2-10+65:H2$=CHR$(X):GOTO 460
590 X=L1-10+65:L1$=CHR$(X):GOTO 480
600 X=L2-10+65:L2$=CHR$(X):GOTO 510
610 PRINT:PRINT "* *  Ausdruck fertig  *
 *"
620 REM Frage nach eventuellem
        Neustart
630 PRINT:F$="":INPUT "Nochmal (J/ ) ";F
$:IF LEFT$(F$,1)="J" OR LEFT$(F$,1)="j"
THEN F$="J"
640 IF F$="J" THEN RUN
650 END
660 IF D<64 OR D>127 THEN GOTO 720
670 IF D=118 THEN RETURN
680 DH=INT(D/16):DL=D-DH*16
690 G=DL AND 7:F=((D AND 56)/8)
700 I$(D)="LD         "+J$(F)+","
710 I$(D)=I$(D)+J$(G):GOSUB 3390:RETURN
720 IF D<128 OR D>191 THEN GOTO 840
```

```
730 DH=INT(D/16):DL=D-(DH*16)
740 G=DL AND 7:F=((D AND 120)/8)
750 IF F=0 THEN I$(D)="ADD      A,"
760 IF F=1 THEN I$(D)="ADC      A,"
770 IF F=2 THEN I$(D)="SUB      "
780 IF F=3 THEN I$(D)="SBC      A,"
790 IF F=4 THEN I$(D)="AND      "
800 IF F=5 THEN I$(D)="XOR      "
810 IF F=6 THEN I$(D)="OR       "
820 IF F=7 THEN I$(D)="CP       "
830 I$(D)=I$(D)+J$(G):GOSUB 3390:RETURN
840 IF BL(D)=1 THEN GOTO 890
850 IF BL(D)=2 THEN GOTO 900
860 IF BL(D)=3 THEN GOTO 910
870 IF BL(D)=4 THEN GOTO 1780
880 IF D=221 OR D=253 THEN GOTO 2230
890 GOSUB 3390:RETURN
900 A=A+1:Z=PEEK(A):GOTO 920
910 A=A+1:Z=PEEK(A):A=A+1:Z1=PEEK(A)
920 V=Z:GOSUB 2750:EH$=H$:EL$=L$
930 IF BL(D)=3 THEN GOTO 950
940 GOSUB 3400:GOTO 960
950 V=Z1:GOSUB 2750:FH$=H$:FL$=L$
960 P1$=FH$+FL$+EH$+EL$:P2$=EH$+EL$:GH$=
"":GL$=""
970 F=D AND 7:G=D AND 56:G=G/8
980 IF F=6 AND (D AND 192)=0 THEN GOTO 1
240
990 IF F=2 AND (D AND 192)=192 THEN GOTO
 1270
1000 IF F=4 THEN GOTO 1280
1010 IF F=0 AND (G<>2) THEN GOTO 1250
1020 I$(33)="LD         HL,"+P1$
1030 I$(34)="LD         ("+P1$+"),HL"
1040 I$(50)="LD         ("+P1$+"),A"
1050 I$(205)="CALL       "+P1$
1060 I$(195)="JP         "+P1$
1070 I$(58)="LD         A,("+P1$+")"
1080 I$(254)="CP         "+P2$
1090 I$(42)="LD         HL,("+P1$+")"
```

```
1100 I$(49)="LD          SP,"+P1$
1110 I$(17)="LD          DE,"+P1$
1120 I$(16)="DJNZ        "+P2$
1130 I$(1)="LD          BC,"+P1$
1140 I$(198)="ADD         A,"+P2$
1150 I$(206)="ADC         A,"+P2$
1160 I$(211)="OUT         "+P2$+",A"
1170 I$(214)="SUB         "+P2$
1180 I$(219)="IN          A,"+P2$
1190 I$(222)="SBC         A,"+P2$
1200 I$(230)="AND         "+P2$
1210 I$(238)="XOR         "+P2$
1220 I$(246)="OR          "+P2$
1230 RETURN
1240 I$(D)="LD          "+J$(G)+","+P2$:RE
TURN
1250 IF G=3 THEN I$(D)="JR          "+P2$:
RETURN
1260 G=G-4:I$(D)="JR          "+S$(G)+","+
P2$:RETURN
1270 I$(D)="JP          "+S$(G)+","+P1$:RE
TURN
1280 I$(D)="CALL        "+S$(G)+","+P1$:RE
TURN
1290 FOR S=0 TO 63:BL(S)=1:NEXT S
1300 FOR S=192 TO 255:BL(S)=1:NEXT S
1310 BL(118)=1:BL(6)=2:BL(14)=2:BL(16)=2
:BL(22)=2:BL(24)=2:BL(30)=2:BL(32)=2:BL(
38)=2:BL(40)=2:BL(46)=2:BL(48)=2:BL(54)=
2:BL(56)=2
1320 BL(62)=2:BL(198)=2:BL(206)=2:BL(211
)=2:BL(214)=2:BL(219)=2:BL(222)=2:BL(230
)=2:BL(238)=2:BL(246)=2:BL(254)=2
1330 BL(1)=3:BL(17)=3:BL(33)=3:BL(34)=3:
BL(42)=3:BL(49)=3:BL(50)=3:BL(58)=3:BL(1
94)=3:BL(195)=3:BL(196)=3:BL(202)=3:BL(2
04)=3:BL(205)=3:BL(210)=3:BL(212)=3:BL(2
18)=3:BL(220)=3
```

```
1340 BL(226)=3:BL(228)=3:BL(234)=3:BL(23
6)=3:BL(242)=3:BL(244)=3:BL(250)=3:BL(25
2)=3:BL(203)=3:BL(237)=3:BL(221)=0:BL(25
3)=3
1350 J$(0)="B":J$(1)="C":J$(2)="D":J$(3)
="E":J$(4)="H":J$(5)="L"
1360 J$(6)="(HL)":J$(7)="A"
1370 S$(0)="NZ":S$(1)="Z":S$(2)="NC":S$(
3)="C":S$(4)="PO":S$(5)="PE":S$(6)="P":S
$(7)="M"
1380 P$(0)="SBC         HL,":P$(1)="ADC
     HL,"
1390 P$(2)="SBC         HL,":P$(3)="ADC
     HL,"
1400 P$(4)="SBC         HL,":P$(5)="ADC
     HL,"
1410 P$(7)="ADC         HL,":N$(0)="BC":N$
(1)="BC":N$(2)="DE"
1420 N$(3)="DE":N$(4)="HL":N$(5)="HL":N$
(7)="SP"
1430 L$(0)="N":L$(1)="I":M$(0)="0":M$(2)
="1":M$(3)="2"
1440 O$(0)="I,A":O$(1)="R,A":O$(2)="A,I"
:O$(3)="A,R"
1450 Q$(0)="LD":Q$(1)="CP":Q$(2)="IN":Q$
(3)="OUT"
1460 I$(0)="NOP":I$(2)="LD          (BC),A
":I$(3)="INC       BC"
1470 I$(7)="RLC         A":I$(8)="EX
  AF,AF'"
1480 I$(9)="ADD         HL,BC":I$(10)="LD
       A,(BC)"
1490 I$(11)="DEC          BC":I$(15)="RRC
     A"
1500 I$(18)="LD          (DE),A":I$(19)="I
NC        DE"
1510 I$(23)="RLA":I$(25)="ADD          HL,D
E":I$(26)="LD          A,(DE)"
1520 I$(27)="DEC          DE":I$(31)="RRA":
I$(35)="INC         HL"
```

```
1530 I$(39)="DAA":I$(41)="ADD        HL,H
L":I$(43)="DEC        HL"
1540 I$(47)="CPL":I$(51)="INC        SP":
I$(55)="SCF"
1550 I$(57)="ADD        HL,SP":I$(59)="DE
C        SP":I$(63)="CCF"
1560 I$(43)="DEC        HL"
1570 I$(192)="RET        NZ":I$(197)="PUS
H       BC"
1580 I$(199)="RST        0":I$(200)="RET
       Z":I$(207)="RST        8"
1590 I$(208)="RET        NC":I$(209)="POP
        DE"
1600 I$(213)="PUSH       DE":I$(215)="RST
        10H"
1610 I$(216)="RET        C":I$(223)="RST
       18H
1620 I$(224)="RET        PO":I$(227)="EX
        (SP),HL"
1630 I$(231)="RST        20H":I$(232)="RE
T        PE"
1640 I$(233)="JP         (HL)":I$(235)="E
X         DE,HL"
1650 I$(239)="RST        28H":I$(240)="RE
T        P"
1660 I$(201)="RET"
1670 I$(241)="POP        AF":I$(243)="DI"
:I$(245)="PUSH       AF"
1680 I$(247)="RST        30H":I$(248)="RE
T        M":I$(251)="EI"
1690 I$(249)="LD         SP,HL":I$(255)="
RST        38H"
1700 I$(197)="PUSH       BC"
1710 I$(193)="POP        BC":I$(229)="PUS
H       HL"
1720 I$(225)="POP        HL"
1730 I$(217)="EXX"
1740 I$(118)="HALT"
1750 L=0:FOR D=5 TO 45 STEP 8:I$(D)="DEC
        "+J$(L):L=L+1:NEXT D
```

```
176Ø L=Ø:FOR D=4 TO 44 STEP 8:I$(D)="INC
      "+J$(L):L=L+1:NEXT D
177Ø I$(6Ø)="INC        A":I$(61)="DEC
    A":GOTO 8Ø
178Ø IF D=2Ø3 THEN GOTO 18ØØ
179Ø IF D=237 THEN GOTO 296Ø
18ØØ A=A+1:Z=PEEK(A)
1811Ø EH=INT(Z/16):EL=Z-EH*16:G=EL AND 7
1821Ø F=INT((EL AND 8)/8):F=F+(EH*2):GOSU
B 184Ø
1831Ø GOTO 199Ø
184Ø IF F=Ø THEN I$(D)="RLC        ":RETU
RN
185Ø IF F=1 THEN I$(D)="RRC        ":RETU
RN
186Ø IF F=2 THEN I$(D)="RL         ":RETU
RN
187Ø IF F=3 THEN I$(D)="RR         ":RETR
UN
188Ø IF F=4 THEN I$(D)="SLA        ":RETU
RN
189Ø IF F=5 THEN I$(D)="SRA        ":RETU
RN
19ØØ IF F=7 THEN I$(D)="SRL        ":RETU
RN
191Ø IF F>=8 AND F<=15 THEN GOTO 195Ø
192Ø IF F>=16 AND F<=23 THEN GOTO 197Ø
193Ø F=F AND 7:F$=CHR$(F+48)
194Ø I$(D)="SET      "+F$+",":RETURN
195Ø F=F AND 7:F$=CHR$(F+48)
196Ø I$(D)="BIT      "+F$+",":RETURN
197Ø F=F AND 7:F$=CHR$(F+48)
198Ø I$(D)="RES      "+F$+",":RETURN
199Ø IF EH>=1Ø THEN GOTO 2Ø4Ø
2ØØØ EH$=STR$(EH):EH$=MID$(EH$,2,1)
2Ø1Ø IF EL>=1Ø THEN GOTO 2Ø5Ø
2Ø2Ø EL$=STR$(EL):EL$=MID$(EL$,2,1)
2Ø3Ø GOSUB 34ØØ:I$(D)=I$(D)+J$(G):RETURN
2Ø4Ø X=EH+55:EH$=CHR$(X):GOTO 2Ø1Ø
2Ø5Ø X=EL+55:EL$=CHR$(X):GOTO 2Ø3Ø
```

```
2060 IF D AND 7=5 THEN GOTO 2090
2070 IF D AND 7=4 THEN GOTO 2090
2080 GOTO 890
2090 L=D AND 56
2100 G=L/8
2110 IF D AND 7=5 THEN I$(D)="DEC
"+J$(G):GOTO 2130
2120 I$(D)="INC        "+J$(G)
2130 GOSUB 3390:RETURN
2140 REM Unterprogramm zum Ausschliessen
        des Disassemblierens von
        bestimmten Speicherbereichen
2150 N=1
2160 INPUT "Startadr.=";A(N)
2170 INPUT "Endadr.=";B(N)
2180 IF A(N)<A OR B(N)>B THEN GOTO 2160
2190 INPUT "Noch ein Teil (J/ ) ";V$
2200 IF LEFT$(V$,1)="j" OR LEFT$(V$,1)="
J" THEN V$="J"
2210 IF V$="J" THEN N=N+1:GOTO 2160
2220 M=N:N=1:RETURN
2230 IF D=221 THEN V$="IX":GOTO 2250
2240 V$="IY"
2250 A=A+1:Z=PEEK(A)
2260 IF Z=203 THEN GOTO 2670
2270 IF Z>=70 AND Z<=190 THEN GOTO 2820
2280 IF Z=33 OR Z=34 OR Z=42 THEN GOTO 2
440
2290 IF Z=52 OR Z=53 THEN GOTO 2540
2300 IF Z=54 THEN GOTO 2600
2310 V=Z:GOSUB 2750:EH$=H$:EL$=L$:GOSUB
3400
2320 IF Z=9 THEN I$(D)="ADD        "+V$+
",BC"
2330 IF Z=25 THEN I$(D)="ADD       "+V$+
",DE"
2340 IF Z=35 THEN I$(D)="INC       "+V$
2350 IF Z=41 THEN I$(D)="ADD       "+V$+
","+V$
2360 IF Z=43 THEN I$(D)="DEC       "+V$
```

```
2370 IF Z=57 THEN I$(D)="ADD        "+V$+
",SP"
2380 IF Z=225 THEN I$(D)="POP        "+V$
2390 IF Z=227 THEN I$(D)="EX         (SP)
,"+V$
2400 IF Z=229 THEN I$(D)="PUSH       "+V$
2410 IF Z=233 THEN I$(D)="JP         ("+V
$+")"
2420 IF Z=249 THEN I$(D)="LD         SP,"
+V$
2430 RETURN
2440 V=Z:GOSUB 2750
2450 EH$=H$:EL$=L$
2460 A=A+1:Z1=PEEK(A):V=Z1:GOSUB 2750
2470 FH$=H$:FL$=L$:A=A+1
2480 Z1=PEEK(A):V=Z1:GOSUB 2750
2490 GH$=H$:GL$=L$
2500 IF Z=33 THEN I$(D)="LD          "+V$+
","+GH$+GL$+FH$+FL$
2510 IF Z=34 THEN I$(D)="LD          ("+GH
$+GL$+FH$+FL$+"),"+V$
2520 IF Z=42 THEN I$(D)="LD          "+V$+
",("+GH$+GL$+FH$+FL$+")"
2530 RETURN
2540 V=Z:GOSUB 2750:EH$=H$:EL$=L$
2550 A=A+1:Z1=PEEK(A):V=Z1:GOSUB 2750
2560 FH$=H$:FL$=L$:GH$="":GL$=""
2570 IF Z=52 THEN I$(D)="INC         ("+V
$+"+"+FH$+FL$+")"
2580 IF Z=53 THEN I$(D)="DEC         ("+V
$+"+"+FH$+FL$+")"
2590 RETURN
2600 V=Z:GOSUB 2750:EH$=H$:EL$=L$
2610 A=A+1:Z1=PEEK(A):V=Z1:GOSUB 2750
2620 FH$=H$:FL$=L$:A=A+1
2630 Z1=PEEK(A):V=Z1:GOSUB 2750
2640 GH$=H$:GL$=L$
2650 I$(D)="LD          ("+V$+"+"+FH$+FL$+
"),"+GH$+GL$
2660 RETURN
```

```
267Ø A=A+1:EH$=CHR$(67):EL$=CHR$(66)
268Ø V=PEEK(A):GOSUB 275Ø
269Ø FH$=H$:FL$=L$
27ØØ A=A+1:Z2=PEEK(A):O=Z2 AND 248:O=O/8
271Ø Q$="("+V$+"+"+FH$+FL$+")"
272Ø F=O:GOSUB 184Ø
273Ø I$(D)=I$(D)+Q$
274Ø V=Z2:GOSUB 275Ø:GH$=H$:GL$=L$:RETUR
N
275Ø H=INT(V/16):L=V-(H*16)
276Ø IF H>=1Ø GOTO 28ØØ
277Ø H$=STR$(H):H$=MID$(H$,2,1)
278Ø IF L>=1Ø THEN GOTO 281Ø
279Ø L$=STR$(L):L$=MID$(L$,2,1):RETURN
28ØØ X1=H+55:H$=CHR$(X1):GOTO 278Ø
281Ø X1=L+55:L$=CHR$(X1):RETURN
282Ø V=Z:GOSUB 275Ø:EH$=H$:EL$=L$:GH$=""
:GL$=""
283Ø A=A+1:Z1=PEEK(A)
284Ø V=Z1:GOSUB 275Ø:FH$=H$:FL$=L$
285Ø IF Z=126 THEN I$(D)="LD        A,("
+V$+"+"+FH$+FL$+")":RETURN
286Ø P=Z:AND 24Ø
287Ø IF P=112 THEN GOTO 291Ø
288Ø IF P>=128 THEN GOTO 293Ø
289Ø P=Z AND 56:P=P/8:GOSUB 33ØØ
29ØØ I$(D)="LD        "+G$+",("+V$+"+"+F
H$+FL$+")":RETURN
2911Ø P=Z AND 7:GOSUB 33ØØ
2922Ø I$(D)="LD        ("+V$+"+"+FH$+FL$+
"),"+G$:RETURN
293Ø P=Z AND 56:P=P/8:GOSUB 33ØØ
294Ø I$(D)=I$(D)+V$+"+"+FH$+FL$+")"
295Ø RETURN
296Ø A=A+1:Z=PEEK(A):V=Z:GOSUB 275Ø:EH$=
H$:EL$=L$
297Ø IF Z=67 OR Z=75 OR Z=83 OR Z=91 OR
Z=115 OR Z=123 THEN GOTO 321Ø
298Ø GOSUB 34ØØ:F=Z AND 248:G=Z AND 7
299Ø IF F=16Ø THEN GOTO 317Ø
```

```
3000 IF F=168 THEN GOTO 3180
3010 IF F=176 THEN GOTO 3190
3020 IF F=184 THEN GOTO 3200
3030 F=Z AND 56:F=F/8:G=Z AND 7
3040 IF F=6 THEN I$(D)="SBC     HL,SP"
:RETURN
3050 IF G=0 THEN I$(D)="IN      "+J$(F
)+",(C)":RETURN
3060 IF G=1 THEN I$(D)="OUT     (C),"+
J$(F):RETURN
3070 IF G=2 THEN I$(D)=P$(F)+N$(F):RETUR
N
3080 IF G=4 THEN I$(D)="NEG":RETURN
3090 IF G=5 THEN I$(D)="RET"+L$(F):RETUR
N
3100 IF G=6 THEN I$(D)="IM "+M$(F):RETUR
N
3110 IF G<>7 THEN I$(D)="* * *":RETURN
3120 IF F<=3 THEN GOTO 3160
3130 IF F=4 THEN I$(D)="RRD":RETURN
3140 IF F=5 THEN I$(D)="RLD":RETURN
3150 GOTO 3110
3160 I$(D)="LD      "+O$(F):RETURN
3170 I$(D)=Q$(G)+"I":RETURN
3180 I$(D)=Q$(G)+"D":RETURN
3190 I$(D)=Q$(G)+"IR":RETURN
3200 I$(D)=Q$(G)+"DR":RETURN
3210 A=A+1:Z1=PEEK(A):V=Z1:GOSUB 2750:FH
$=H$:FL$=L$
3220 A=A+1:Z1=PEEK(A):V=Z1:GOSUB 2750:GH
$=H$:GL$=L$
3230 GG$=GH$+GL$+FH$+FL$
3240 IF Z=67 THEN I$(D)="LD      ("+GG
$+"),BC":RETURN
3250 IF Z=75 THEN I$(D)="LD      BC,("
+GG$+")":RETURN
3260 IF Z=83 THEN I$(D)="LD      ("+GG
$+"),DE":RETURN
3270 IF Z=91 THEN I$(D)="LD      DE,("
+GG$+")":RETURN
```

```
3280 IF Z=115 THEN I$(D)="LD         ("+G
G$+"),SP":RETURN
3290 IF Z=123 THEN I$(D)="LD         SP,(
"+GG$+")":RETURN
3300 IF P=0 THEN G$="B":I$(D)="ADD
  A,("
3310 IF P=1 THEN G$="C":I$(D)="ADC
A,("
3320 IF P=2 THEN G$="D":I$(D)="SUB      (
"
3330 IF P=3 THEN G$="E":I$(D)="SBC      A
,("
3340 IF P=4 THEN G$="H":I$(D)="AND      (
"
3350 IF P=5 THEN G$="L":I$(D)="XOR      (
"
3360 IF P=6 THEN G$="***":I$(D)="OR
 ("
3370 IF P=7 THEN G$="A":I$(D)="CP       (
"
3380 RETURN
3390 EH$="":EL$="":FH$="":FL$="":GH$="":
GL$="":RETURN
3400 FH$="":FL$="":GH$="":GL$="":RETURN
3410 IF B$="D" THEN LPRINT ELSE PRINT
3420 IF B$="D" THEN LPRINT ELSE PRINT
3430 QT=0:SY=1
3440 IF B$="D" THEN LPRINT ELSE PRINT
3450 RETURN
3460 REM Ende der
         Unterprogrammbibliothek zur
         Erzeugung des Assemblercodes
3470 REM Ueberpruefung der Anzahl der zu
         disassemblierenden Bytes
         hexadezimale Eingabe) in den
         DATA-Zeilen
3480 ON ERROR GOTO 3570:FOR ZA=55000! TO
 65535!:READ A$:NEXT ZA
```

```
3490 REM Eingentliches Einlesen der
         Bytes in der DATA-Zeile in
         den Speicher ab 43776
3500 RESTORE:FOR M=55000! TO ZA-1
3510 READ A$
3520 A=VAL("&h"+A$)
3530 POKE M,A
3540 NEXT M:POKE M,&HC9
3550 RETURN
3560 DATA 3e,04,06,07,80,32,00,7d
3570 IF ERR=4 THEN RESUME 3500 ELSE STOP
```

```
   * * * Z-80 Disassembler * * *
Titel ? Einfache Addition in MC

Startadresse (in Dez.) ?
Endadresse (in Dez.)   ?

Datafelder vorhanden (J/ ) ?

Drucker oder Bildschirm (D/B) ? b

   Einfache Addition in MC

 D6D8    3E04     LD          A,04
 D6DA    0607     LD          B,07
 D6DC    80       ADD         A,B
 D6DD    32007D   LD          (7D00),A
 D6E0    C9       RET

* *  Ausdruck fertig  * *

3560 DATA 3e,04,06,07,80,32,00,7d
Ok
```

Anhang

Die Token der MSX-Computer (siehe 'Speicher 1' bis 'Speicher 5')

32 bis 126: normaler ASCII-Zeichensatz

129	END	130	FOR	131	NEXT	132	DATA
133	INPUT	134	DIM	135	READ	136	LET
137	GOTO	138	RUN	139	IF	140	RESTORE
141	GOSUB	142	RETURN	143	REM	144	STOP
145	PRINT	146	CLEAR	147	LIST	148	NEW
149	ON	150	WAIT	151	DEF	152	POKE
153	CONT	154	CSAVE	155	CLOAD	156	OUT
157	LPRINT	158	LLIST	159	CLS	160	WIDTH
161	LSE	162	TRON	163	TROFF	164	SWAP
165	ERASE	166	ERROR	167	RESUME	168	DELETE
169	AUTO	170	RENUM	171	DEFSTR	172	DEFINT
173	DEFSNG	174	DEFDBL	175	LINE	176	OPEN
177	FIELD	178	GET	179	PUT	180	CLOSE
181	LOAD	182	MERGE	183	FILES	184	LSET
185	RSET	186	SAVE	187	LFILES	188	CIRCLE
189	COLOR	190	DRAW	191	PAINT	192	BEEP
193	PLAY	194	PSET	195	PRESET	196	SOUND
197	SCREEN	198	VPOKE	199	SPRITE	200	VDP
201	BASE	202	CALL	203	TIME	204	KEY
205	MAX	206	MOTOR	207	BLOAD	208	BSAVE
209	DSKO$	210	SET	211	NAME	212	KILL
213	IPL	214	COPY	215	CMD	216	LOCATE
217	TO	218	THEN	219	TAB(	220	STEP
221	USR	222	FN	223	SPC(	224	NOT
225	ERL	226	ERR	227	STRING$	228	USING
229	INSTR	230	' (nicht ASCII, sondern Befehl 'REM')				
231	VARPTR	232	CSRLIN	233	ATTR$	234	DSKI$
235	OFF	236	INKEY$	237	POINT	238	> (Rechnung)
239	= (Rech.)	240	< (Re.)	241	+ (Rech.)	242	- (Rechnung)
243	* (Rech.)	244	/ (Re.)	245	^ (Rech.)	246	AND
247	OR	248	XOR	249	EQV	250	IMP
251	MOD	252 bis 254: keine sichtbaren Tokens					

Mit dem Vorzeichen '255' entstehen folgende Befehlswörter:

1	LEFT$	2	RIGHT$	3	MID$	4	SGN
5	INT	6	ABS	7	SQR	8	RND
9	SIN	10	LOG	11	EXP	12	COS
13	TAN	14	ATN	15	FRE	16	INP
17	POS	18	LEN	19	STR$	20	VAL
21	ASC	22	CHR$	23	PEEK	24	VPEEK
25	SPACE$	26	OCT$	27	HEX$	28	LPOS
29	BIN$	30	CINT	31	CSNG	32	CDBL
33	FIX	34	STICK	35	STRIG	36	PDL
37	PAD	38	DSKF	39	FPOS	40	CVI
41	CVS	42	CVD	43	EOF	44	LOC
45	LOF	46	MKI$	47	MKS$	48	MKD$

Endlich!

DATA WELT – JETZT ALLE 2 MONATE NEU

Darauf haben alle gewartet: Das aktuelle Computermagazin aus dem Hause DATA BECKER gibt's jetzt alle 2 Monate.
Das heißt für Sie: Mehr Informationen, mehr Tips & Tricks und immer auf dem neuesten Stand.
Aktuelle Berichte, brandheiße News über Hardware und Software runden die Themenvielfalt jeder DATA WELT ab.

DATA WELT gleich besorgen! Am Kiosk und beim Fachhändler.

MSX für Einsteiger sollte das erste Buch zu Ihrem MSX-Computer sein. Es ist eine leichtverständliche Einführung in Handhabung, Einsatz und Programmierung von MSX-Rechnern, die keinerlei Vorkenntnisse voraussetzt. Ein Buch, das zu jedem MSX gehört.
MSX FÜR EINSTEIGER, 1985, ca. 200 Seiten, DM 29,–

DAS STEHT DRIN:

Diese Programmsammlung enthält Spitzenprogramme, vom Disassembler bis zum Sporttabellenprogramm. Mit spannenden Superspielen und kompletten Anwendungsprogrammen. Zusätzlich werden Sie zu den Programmen auf wichtige Programmiertips und Tricks hingewiesen. Die Programme laufen auf allen MSX-Computern sowie auf Spectravideo 318 und 328.

Aus dem Inhalt:

- Hexdump
- Grafikeditor
- Soundeditor
- Deutsche Umlaute
- Computerschrift
- Variablenreferenzliste
- Kalender
- Disassembler
- Daten- und Langspielplattenverwaltung
- Hollow – das Kirschenspiel
- Balkendiagramme
- Sporttabellen

UND GESCHRIEBEN HAT DIESES BUCH:

Rainer Lüers ist ausgebildeter Lehrer, Computer-Fachtrainer bei einer großen Kaufhauskette und erfahrener Autor verschiedener DATA BECKER Bücher (CPC 464 BASIC-Programme, MSX Grafik & Sound).

ISBN 3-89011-090-8