

Maschinen sprache buch

**für TRS-80 Model I&III
Genie I, II, III, IIs & IIIs
Colour-Genie**

von Luidger Röckrath

RÖCKRATH
MIKROCOMPUTER

Luidger Röckrath:

Das Maschinensprachebuch

für TRS-80-Model-I + III
Genie I, II, III, IIs, IIIs
Speedmaster 5.3
Komtek I, LNW 80
Colour-Genie

ISBN 3-925074-07-4

Best.-Nr.: 06-007

2. korr. Auflage, Januar 1985

(1. Auflage, Dezember 1984, ISBN 3-925074-06-6

(c) 1984 by

RÖCKRATH
MICROCOMPUTER

Noppiusstr. 19
5100 Aachen

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeinem Verfahren reproduziert oder in EDV-Anlagen verarbeitet werden.

Herstellung: Fotodruck J. Mainz GmbH, Aachen

Hinweis:

Alle in diesem Buch wiedergegebenen Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt und dürfen nicht gewerblich genutzt werden.

Alle Angaben in diesem Buch wurden mit größtmöglicher Sorgfalt erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, daß sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung eventueller Fehler sind Autor und Verlag jederzeit dankbar.

Vorwort

=====

Vor über 3 Jahren erschien das TRS-80/Genie-ROM-Listing und etwas später Bearbeitungen für TRS-80 Model III und Colour Genie. Vielen Programmierern auf den genannten Systemen waren diese Publikationen eine wertvolle Hilfe.

Für Programmierer, die noch nicht so tief in die Geheimnisse der Maschinenprogrammierung eingedrungen sind, trotzdem aber eigene Maschinenprogramme schreiben wollen, ist dieses Buch gedacht. Es enthält zu sehr vielen ROM-Routinen, die sinnvoll und vereinfachend in eigenen Maschinenprogrammen benutzt werden können, genaue Beschreibungen und Beispiele. Hierdurch wird die Verwendung des gesamten Basicinterpreters in eigenen Programmen möglich, und nicht nur die einiger Ein-/Ausgabe-Routinen, auf die sich die meisten Publikationen beschränken.

Weiterhin soll dieses Buch die Unterschiede zwischen den verschiedenen mehr oder weniger kompatiblen Systemen aufzeigen. Mit diesen Kenntnissen können Programme geschrieben werden, die auf allen Systemen laufen, oder Programme aus Zeitschriften etc. adaptiert werden.

Außerdem enthält dieses Buch die Source-Listings von zahlreichen Utilities, die als RELOSFT seit Jahren verkauft werden. An diesen Programmen werden die Möglichkeiten des Eingriffes in Betriebssystem und Basicinterpreter genau erläutert.

Aachen, im Oktober 1984

Luidger Röckrath

Inhaltsverzeichnis
=====

Teil 1: ROM-Routinen	5
1.1 Ein-/Ausgaberroutinen	6
1.2 Verwaltung des Programmtextes	16
1.3 Verwaltung der Variablentabelle	21
1.4 Arithmetik	26
1.5 Stringverarbeitung	34
1.6 Ausdrucksauswertung	36
1.7 Kassettenroutinen	38
1.8 Restart's	45
Teil 2: RAM-Adressen	48
Teil 3: Vergleich der verschiedenen ROM-Versionen	50
Teil 4: Beschreibungen der Programme	55
4.1 Bildschirmeditor SEDIT	56
4.2 Basicerweiterung NEWBAS	62
4.3 Double Precision Funktionen DOUBLE	72
4.4 Basicprogrammpacker PACK	75
4.5 Feldaufzeichnung ARRAY	77
4.6 Umlauterweiterung für SCRIPSIT	80
4.7 Assembler/Disassembler Z80AD	83
Teil 5: Listings der Beispielprogramme	87
5.1 Beispiel 1: Durchschnitt eines Feldes	87
5.2 Beispiel 2: Ausgabe einer Codetabelle	89
5.3 Beispiel 3: POP-Anweisung	90
5.4 Beispiel 4: Zahlenein-/ausgabe	91
5.5 Beispiel 5: Systemfileaufzeichnung	92
Teil 6: Listings der Programme aus Teil 4	95
6.1 Bildschirmeditor SEDIT	95
6.2 Basicerweiterung NEWBAS	101
6.3 Double Precision Funktionen DOUBLE	114
6.4 Basicprogrammpacker PACK	124
6.5 Feldaufzeichnung ARRAY	127
6.6 Umlauterweiterung für SCRIPSIT	131
6.7 Assembler/Disassembler Z80AD	133
6.8 Konvertierungsprogramme /SRC <-> /MAC	148

Teil 1: ROM-Routinen
=====

1.1 Ein-/Ausgaberoutinen

```
*****  
*                                                                 *  
* NAME      : Beschreibung                                       *  
*                                                                 *  
*      E    : Einsprung                                           *  
*      B    : Basic notwendig ?                                   *  
*      R    : Registerbeeinflussung                               *  
*      I    : Aufrufparameter                                     *  
*      O    : Ergebnisparameter                                   *  
*                                                                 *  
*****
```

Alle folgenden Beschreibungen der ROM-Routinen fangen mit einem derartigen Block an:

NAME : Der Name der Routine, der aus max. 6 Buchstaben und Ziffern besteht. Dahinter kurz zusammengefasst eine Beschreibung der Funktion.

E : Die Einsprungadresse (ADDRESS)
Die Routine kann entweder mit

```
NAME EQU ADDRESS  
CALL NAME
```

oder

```
CALL ADDRESS
```

aufgerufen werden.

B : Basic notwendig ?
Hier kann "ja" oder "nein" stehen.
Routinen mit "nein" können immer benutzt werden, während Routinen mit "ja" nur dann, wenn das BASIC initialisiert ist. Bei Kassettensystemen ist dies immer der Fall, aber bei Diskettensystemen nur nach Aufruf des BASICs. Lediglich auf dem Colour-Genie ist das Basic auch unter Diskette immer aktiv.

R : Registerbeeinflussung.
Hier sind alle Register verzeichnet, die durch die Routine verändert werden können. Nimmt ein spezielles Register immer einen speziellen Wert an, wenn die Routine durchlaufen wird, ist dieser angegeben.

I : Aufrufparameter
Hier ist aufgelistet, in welchen Registern bzw. Speicherzellen, welche Parameter für die Routine übergeben werden müssen.

O : Ergebnisparameter
In den angegebenen Registern und Speicherzellen werden die Ergebnisse der Routine dem Aufrufer zurückgegeben.

```

*****
*
* OUTCH : Zeichen auf Bildschirm ausgeben
*
* E : 0033H
* B : nein
* R : F, DE=401DH
* I : A = auszugebendes Zeichen
* O : /
*
*****

```

Funktion:

Das Zeichen im Akku wird auf der Cursorposition (CURSOR) im Bildwiederholtspeicher (VIDRAM) eingetragen. Dabei werden folgende Steuerzeichen akzeptiert:

- 07H : BELL = Erzeugt ein Peep (nur Genie III + IIIs)
- 08H : BS = Backspace, letztes Zeichen löschen
- 0AH : LF = 0DH
- 0BH : = 0DH
- 0CH : FF = 0DH
- 0DH : CR = Cursor auf Anfang der nächsten Zeile
- 0EH : = Cursor einschalten (sichtbar)
- 0FH : = Cursor ausschalten (unsichtbar)
- 15H : = Umschaltung Space compression/Special Character (nur TRS-80 Model III)
- 16H : = Umschaltung Invers/Normal (nur Genie III(s))
- 17H : = 32 cpl einschalten (nicht bei Colour Genie)
- 18H : = Cursor 1 Zeichen nach links
- 19H : = Cursor 1 Zeichen nach rechts
- 1AH : = Cursor 1 Zeile runter
- 1BH : = Cursor 1 Zeile hoch
- 1CH : = Cursor in linke, obere Bildschirmecke
- 1DH : = Cursor zum Zeilenanfang
- 1EH : = Von Cursorposition bis Zeilenende löschen
- 1FH : = Von Cursorposition bis Bildschirmende löschen

Die Codes von 20H-7FH werden als ASCII-Zeichen interpretiert, die Zeichen zwischen 80H-FFH als Graphikzeichen, Spacekompression oder inverse Zeichen:

Code	TRS-80 M I	TRS-80 M III	Genie III(s)	Colour Genie
	Genie I+II			

```

-----
20H-3FH Zeichen unverändert in Bildwiederholtspeicher
40H-5FH -> 0H-1FH Zeichen unverändert in Bildwiederhols.
60H-7FH -> 0H-1FH " " " "
80H-BFH Grafik Grafik Grafik abh. CHAR
C0H-FFH Space compr. Space compr. Inverse " "
o. Spec. Ch.

```

Zur Verwaltung des Bildwiederholtspeichers wird ein DCB (Device control block) geführt:

- VIDDCB = 401DH : DCB-Typ = 07H
- + 1 = 401EH : Treiberadresse (LSB)
- + 2 = 401FH : dito (MSB)
- + 3 = 4020H : Cursoradresse (LSB)
- + 4 = 4021H : dito (MSB)

- + 5 = 4022H : = 0, wenn Cursor abgeschaltet (0FH, s. o.)
= Zeichen auf Cursorposition, wenn Cursor eingeschaltet (0EH, s. o.)
- + 6 = 4023H : Cursorzeichen (TRS-80 Model III)
Farbe in LGR-Mode (Colour Genie)
- + 7 = 4024H : Mode für Zeichen >= C0H
= 0: Space compression
<>0: Special characters
(nur TRS-80 Model III)
letzte gedrückte Taste für <RPT>
(nur Colour Genie)

Beispiel:

```
LD      A,' '      ; Leerzeichen nach A
PUSH    DE         ; DE wird durch OUTCH zers.
CALL    OUTCH      ; Leerzeichen ausgeben
POP     DE
```

```
*****
*
* OUTCHF : Zeichen auf Bildschirm/Drucker oder Kassette
*         ausgeben
*
*      E : 032AH
*      B : ja
*      R : F, (evtl. A, siehe WRITEB)
*      I : A = auszugebendes Zeichen
*          (OUTFLG) = 00H : Bildschirm
*                   01H : Drucker
*                   80H : Kassettenrecorder
*      O : /
*
*****
```

Funktion:

In Abhängigkeit von (OUTFLG) wird das Zeichen im Akku auf Bildschirm, Drucker oder Kassettenrecorder ausgegeben. Dazu werden die Routinen OUTCH, PRINT1, WRITEB benutzt. Bei der Ausgabe auf den Kassettenrecorder muß zuvor mit INITW oder WRITES der Header geschrieben werden (siehe Abschnitt 1.7). Außerdem wird bei Ausgabe auf dem Bildschirm die Cursorspalte in (CURSP) abgelegt. Diese Routine wird von der PRINT-Anweisung benutzt um bei LPRINT auf den Drucker und bei PRINT #-1 auf die Kassette ausgeben zu können.

Beispiel:

```
LD      A,1        ; Ausgabe auf Drucker
LD      (OUTFLG),A
LD      A,'0'      ; '0' auf Drucker
CALL    OUTCHF
```

```
*****
*
* CLS      : Bildschirm löschen
*
*      E : 01C9H
*      B : nein
*      R : A = 1FH, F
*      I : /
*      O : /
*
*****
```

Funktion:

1CH und 1FH werden ausgegeben, wodurch der gesamte Bildschirm gelöscht wird.

Beispiel:

```
CALL CLS
```

```
*****
*
* PRINT   : Zeichen auf Drucker ausgeben
*
*      E : 003BH
*      B : nein
*      R : F, DE=4025H, (M III: A, wenn BREAK gedrückt)
*      I : A = auszugebendes Zeichen
*      O : /
*
*****
```

Funktion:

Zeichen im Akku wird auf Drucker ausgegeben. Die Zeichen 00H, 0BH und 0CH werden nicht unverändert an den Drucker weitergegeben. 00H wird nie an den Drucker ausgegeben. 0BH und 0CH bewirken einen FF und werden dazu vom Druckertreiber in eine Folge von CR umgesetzt.

Der Drucker wird bei allen Systemen, außer dem Colour-Genie, durch eine einfache Schaltung mit dem System verbunden. Diese Schaltung erzeugt bei der Ausgabe eines Zeichens automatisch den Strobe-Impuls für die Centronics-Schnittstelle und ermöglicht die Abfrage der Statussignale:

- Bit 7 : Busy
- Bit 6 : Out of Paper
- Bit 5 : Unit select
- Bit 4 : Fault

Diese Schaltung ist entweder als Port oder wie eine Speicherzelle mit dem Systembus verbunden. Die Adressen, unter denen Sie angesprochen werden können, sind verschieden:

	Output:	Input (Status):
TRS-80 Model I :	37E8H	37E8H
TRS-80 Model III :	Port F8H	37E8H, Port F8H
Genie I+II+III :	Port FDH	Port FDH
Genie IIIs :	37E8H, Ports E8H+FDH wie Output	

Mit folgendem Maschinenprogramm kann ein Zeichen an den Drucker ausgegeben werden:

```
DRUCK  PUSH  AF          ; Zeichen retten
DRUCK1 LD    A,(PTRADR)  ; Status prüfen
;      IN    A,(PTRADR)  ; Alternative für Portadr.
;      BIT   7,A         ; Busy ?
;      JR    NZ,DRUCK1   ; ja, -->
;      POP  AF          ; Zeichen zurück
;      LD    (PTRADR),A  ; auf Druckeradresse ausg.
;      OUT  (PTRADR),A  ; Alternative für Portadr.
;      RET
```

Beim Colour-Genie wird der Drucker über den im PSG eingebauten Parallelport angesprochen. Die dazu notwendige Software ist durch die notwendige Programmierung des PSG etwas komplizierter (siehe ROM-Listing).

Für den Druckertreiber wird ein Drucker-DCB benutzt:

```
PTRDCB = 4025H : DCB-Type = 06H
+ 1 = 4026H : Treiberadresse (LSB)
+ 2 = 4027H : dito (MSB)
+ 3 = 4028H : Anzahl der Zeilen pro Seite + 1
+ 4 = 4029H : Zeilenzähler
+ 5 = 402AH : Zeichenzähler (nur TRS-80 M III)
+ 6 = 402BH : Zeichen / Zeile (nur TRS-80 M III)
          (0FFH = keine Begrenzung)
```

Beispiel:

```
LD      A,' '          ; Leerzeichen
PUSH    DE             ; wichtig, wenn DE benutzt
CALL    PRINT         ; Zeichen ausgeben
POP     DE
```

```
*****
*
* PRINT1 : Zeichen auf Drucker ausgeben
*
*      E : 039CH
*      B : nein
*      R : /
*      I : A = auszugebendes Zeichen
*      O : (PTRPOS) verändert
*
*****
```

Funktion:

Zeichen in A wird auf Drucker ausgegeben. Die Druckkopffosition (PTRPOS) wird incrementiert und bei 0AH, 0CH und 0DH auf 0 zurückgesetzt. Statt 0AH wird 0DH

ausgegeben. PRINT1 wird bei allen Druckerausgaben durch den Basicinterpreter mittels OUTCHF (s. o.) benutzt und ruft selbst PRINT auf. Mittels LPRINT ist daher die Ausgabe von 00H, 0AH, 0BH und 0CH nicht möglich ! Besonders bei der Grafikprogrammierung ist dies zu beachten. Durch folgendes Basicunterprogramm kann jedes Zeichen an den Drucker ausgegeben werden:

TRS-80 M I:

```
1000 IF PEEK(14312)<128 THEN POKE 14312,X:RETURN ELSE 1000
```

Genie, TRS-80 M III:

```
1000 IF INP(Y)<128 THEN OUTY,X:RETURN ELSE 1000
```

mit Y=248 bei TRS-80 Model 3 und Y=253 bei Genie

Beispiel:

```
LD      A,'A'          ; A auf Drucker ausgeben
CALL   PRINT1
```

```
*****
*
* INCH   : Zeichen von Tastatur holen
*
*   E : 002BH
*   B : nein
*   R : AF, DE=4015H
*   I : /
*   O : A = 0, wenn keine Taste neu gedrückt ist
*       A = ASCII-Code der neu gedrückten Taste
*
*****
```

Funktion:

Die Tastatur wird abgefragt und, wenn eine Taste neu gedrückt ist, wird deren ASCII-Code in A zurückgegeben, ansonsten enthält A bei Rückkehr eine 0. Die Tastatur ist in einer Matrix angeordnet, die direkt mit Adreß- und Datenbus verbunden ist:

Tastaturmatrix TRS-80 Model I, Genie I+II:

		I Datenleitung							
Adresse:	I Bit 0	1	2	3	4	5	6	7	
3801H	I	ALPHA	A	B	C	D	E	F	G
3802H	I	H	I	J	K	L	M	N	O
3804H	I	P	Q	R	S	T	U	V	W
3808H	I	X	Y	Z					
3810H	I	0	1	2	3	4	5	6	7
3820H	I	8	9	:	;	,	-	*	/
3840H	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHT	SPACE
3880H	I	SHIFT							

Tastaturmatrix TRS-80 Model III (Amerikanische Version):

		I Datenleitung							
Adresse:	I	Bit 0	1	2	3	4	5	6	7
3801H	I	ALPHA	A	B	C	D	E	F	G
3802H	I	H	I	J	K	L	M	N	O
3804H	I	P	Q	R	S	T	U	V	W
3808H	I	X	Y	Z					
3810H	I	0	1	2	3	4	5	6	7
3820H	I	8	9	:	;	,	-	*	/
3840H	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHT	SPACE
3880H	I	LSHFT	RSHFT						

Tastaturmatrix TRS-80 Model III (Deutsche Version):

		I Datenleitung							
Adresse:	I	Bit 0	1	2	3	4	5	6	7
3801H	I	>	A	B	C	D	E	F	G
3802H	I	H	I	J	K	L	M	N	O
3804H	I	P	Q	R	S	T	U	V	W
3808H	I	X	Y	Z	U	+	Ø	X	#
3810H	I	0	1	2	3	4	5	6	7
3820H	I	8	9	B	'	ALPHA	,	.	-
3840H	I	ENTER	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHT	SPACE
3880H	I	LSHFT	RSHFT						

Tastaturmatrix Genie III:

		I Datenleitung							
Adresse:	I	Bit 0	1	2	3	4	5	6	7
3801H	I		A	B	C	D	E	F	G
3802H	I	H	I	J	K	L	M	N	O
3804H	I	P	Q	R	S	T	U	V	W
3808H	I	X	Y	Z	;	X	:	ALPHA	
3810H	I	0	1	2	3	4	5	6	7
3820H	I	8	9	B	Ø	,	-	.	/
3840H	I	NEW LN	CLEAR	BREAK	HOCH	RUNTER	LINKS	RECHT	SPACE
3880H	I	SHIFT	CTRL						LP
38A0H	I	F1	F2	F3	F4	F5	F6	F7	F8
38C0H	I	0	1	2	3	4	5	6	7
38E0H	I	8	9	00	LOCK	,	-	.	

Tastaturmatrix Genie IIIs:

```

-----
          I Datenleitung
Adresse: I Bit 0   1   2   3   4   5   6   7
-----
3801H  I  ALPHA  A   B   C   D   E   F   G
3802H  I    H   I   J   K   L   M   N   O
3804H  I    P   Q   R   S   T   U   V   W
3808H  I    X   Y   Z   X   ö   ü   ß   _
3810H  I    0   1   2   3   4   5   6   7
3820H  I    8   9   :   ;   ,   -   .   /
3840H  I  ENTER CLEAR BREAK HOCH RUNTER LINKS RECHT SPACE
3880H  I  SHIFT CTRL  ESC   P5   P4   P3   P2   P1
38A0H  I    F1   F2   F3   F4   F5   F6   F7   F8
38C0H  I    0   1   2   3   4   5   6   7
38E0H  I    8   9   00  LOCK ,   -   .   PRINT
-----

```

Tastaturmatrix Colour Genie:

```

-----
          I Datenleitung
Adresse: I Bit 0   1   2   3   4   5   6   7
-----
F801H  I  ALPHA  A   B   C   D   E   F   G
F802H  I    H   I   J   K   L   M   N   O
F804H  I    P   Q   R   S   T   U   V   W
F808H  I    X   Y   Z           F1   F2   F3   F4
F810H  I    0   1   2   3   4   5   6   7
F820H  I    8   9   :   ;   ,   -   .   /
F840H  I  ENTER CLEAR BREAK HOCH RUNTER LINKS RECHT SPACE
F880H  I  SHIFT MODSEL          RPT CTRL          LP
-----

```

(LP = Light pen, ALPHA = @)

Durch die Routine INCH wird die Tastaturmatrix abgefragt, und wenn eine neue Taste gedrückt ist, deren ASCII-Code ermittelt. Nach einmaligen Erkennen wird eine für längere Zeit gedrückte Taste ignoriert (Ausnahme: Autorepeat). Von Maschinenprogrammen aus ist es oft sinnvoll festzustellen, ob gerade eine bestimmte Taste gedrückt ist. Dies ist mit folgendem kleinen Programm möglich:

```

LD      A,(3840H)      ; Tastaturzeile ENTER-SPACE
BIT     2,A            ; BREAK gedrückt?
JR      NZ,LABEL1     ; ja, -->
; Hier geht weiter, wenn BREAK nicht gedrückt ist
.
.
.
LABEL1 ...            ; Verarbeitung bei BREAK

```



```
*****
*
* INBUFF : Zeile in Buffer eingeben
*
*      E : 05D9H
*      B : nein
*      R : AF, BC, DE=401DH
*      I : B = max. Anzahl der einzugebenden Zeichen
*          HL = Bufferanfang
*      O : A = letzter Tastendruck (CR oder BREAK)
*          CY = 1 bei Abbruch mit BREAK
*          B = Anzahl der eingegebenen Zeichen (ohne CR)
*          C = B bei Eintritt (max. Anzahl)
*          (HL      ) = 1. Zeichen
*          (HL+1  ) = 2. Zeichen
*          .
*          (HL+B-1) = B. Zeichen
*          (HL+B  ) = 0D (Endmarkierung CR)
*
*****
```

Funktion:

Der Cursor erscheint und es wird die Eingabe von max. B Zeichen (ohne CR) ermöglicht. Dabei können die üblichen Controlzeichen benutzt werden. Die Routine kann nur durch ENTER oder BREAK verlassen werden, wobei letztere Abbruchbedingung dem Aufrufer durch das gesetzte CY-Flag angezeigt wird. HL wird nicht verändert, daher ist die Bufferadresse nach Rückkehr noch verfügbar.

Beispiel:

```
INPUT  LD      HL,41E8H      ; Zeiger auf I/O-Buffer
        LD      B,1         ; max. 1 Zeichen
        CALL   INBUFF      ; Eingabe
        JR     C,LABEL     ; BREAK-Abbruch? --> LABEL
        LD      A,(HL)     ; 1. Zeichen laden
        CP     'J'         ; 'J' ?
        JR     Z,JA        ; ja, -->
        CP     'N'         ; 'N' ?
        JR     Z,NEIN     ; nein, -->
        JR     INPUT      ; weder noch? --> wdh.
        .
        ; Programmteil bei Eingabe 'J'
JA      ...
        ; Programmteil bei Eingabe 'N'
NEIN   ...
        ; Programmteil bei Abbruch mit BREAK
LABEL  ...
```

```
*****  
*  
*  INLINE : 240 Zeichen in I/O-Buffer eingeben      *  
*  
*      E : 0361H                                     *  
*      B : ja                                       *  
*      R : AF, DE=401DH, HL                         *  
*      I : /                                         *  
*      O : A = letzter Tastendruck (CR oder BREAK)  *  
*            CY = 1 bei Abbruch mit BREAK           *  
*            HL = (40A7H) : I/O-Bufferanfang       *  
*            (HL      ) = 1. Zeichen                *  
*            (HL+1   ) = 2. Zeichen                *  
*            .      .                               *  
*            (HL+x-1) = x. Zeichen                  *  
*            (HL+x   ) = 0 (Endmarkierung)         *  
*  
*****
```

Funktion:

INLINE ermöglicht die Eingabe von max. 240 in den I/O-Buffer (Anfang: (40A7)). BREAK-Abbruch wird wie bei INBUFF angezeigt. Außerdem ist beim korrektem Ende mit CR der Akku auf Null gesetzt und das Z-Flag gesetzt. INLINE wird zur Zeileneingabe durch die Hauptschleife und zur Eingabe bei INPUT benutzt.

1.2 Verwaltung des Programmtextes

Das Basicprogramm ist im Speicher Zeile für Zeile, beginnend bei (TXTANF) abgelegt. Jede Zeile beginnt mit Zeilenzeiger auf die nächste Zeile, gefolgt von der Zeilennummer und dem Text im Zwischencode mit Tokens, der mit einer 0H abgeschlossen wird. Nach der letzten Zeile folgt ein Null-Zeilenzeiger, der das Programmtextende anzeigt:

```
10 FORI=1TO9
20 PRINTI
30 NEXT
```

Speicherauszug 42E9H ff. mit den Zeigern, die die Größe des Programmtextes definieren. Untereinander ist jeweils die Adresse (nur LSB und in der 1. Spalte auch MSB), der Inhalt der Speicherzelle und dessen Bedeutung angegeben:

```
+- (TXTANF)
V
+-42-+-----+-----+-----+-----+-----+-----+-----+-----+
! E9 ! EA ! EB ! EC ! ED ! EE ! EF ! F0 ! F1 ! F2 ! F3 !
+-----+-----+-----+-----+-----+-----+-----+-----+
! F4 ! 42 ! 0A ! 00 ! 81 ! 49 ! D5 ! 31 ! B0 ! 39 ! 00 !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Zeil.zg.! ZeilenNr! FOR! I ! = ! 0 ! TO ! 9 ! EOL!
+-----+-----+-----+-----+-----+-----+-----+-----+
!
V
+-42-+-----+-----+-----+-----+-----+-----+
! F4 ! F5 ! F6 ! F7 ! F8 ! F9 ! FA !
+-----+-----+-----+-----+-----+-----+-----+-----+
! FB ! 42 ! 14 ! 00 ! B2 ! 49 ! 00 !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Zeil.zg.! ZeilenNr!PRIN! I ! EOL!
+-----+-----+-----+-----+-----+-----+-----+-----+
!
V
+-42-+-----+-----+-----+-----+-----+-----+-----+-----+
! FB ! FC ! FD ! FE ! FF ! 00 !
+-----+-----+-----+-----+-----+-----+-----+-----+
! 01 ! 43 ! 1E ! 00 ! 87 ! 00 !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Zeil.zg.! ZeilenNr!NEXT! EOL!
+-----+-----+-----+-----+-----+-----+-----+-----+
!
! +- (TXTEND)
V V
+-43-+-----+-----+
! 01 ! 02 ! 03 !
+-----+-----+-----+
! 00 ! 00 ! ? !
+-----+-----+-----+
! Zeil.zg.! !
+-----+-----+-----+-----+-----+-----+-----+-----+
```


Verwendbar sind vor allem die Routinen zur Umwandlung von Klartext in Zwischencode und umgekehrt und zur Suche einer Zeile im Programmtext. Außerdem ist die Generierung des Zeilenzeigersystems manchmal sinnvoll einzusetzen:

```
*****
*
* ENCTXT : Zwischencode erzeugen
*
*   E : 1BC0H
*   B : Ja
*   R : AF, BC, DE, HL
*   I : HL = Zeiger auf umzuwandelnden Text
*         mit 0H abgeschlossen
*         (HL) = 1. Byte des Textes
*   O : BC = Länge des Zwischencode + 5
*         (entspricht Länge der Zeile: je 2 Bytes
*         für Zeilenzeiger und -Nr, 1 Byte für EOL)
*   HL = Zeiger auf Byte vor Zwischencode
*         = (40A7H) - 3
*   (HL+1) = 1. Byte des Zwischencode
*
*****
```

Funktion:

Der Text im Speicherbereich, auf dessen Anfang HL weist, wird in Zwischencode umgewandelt. Der Zwischencode wird bei (IOBUFF)-2 beginnend abgelegt. Der Text hinter DATA und REM-Anweisungen, sowie in Anführungsstriche eingeschlossene Zeichenketten, werden nicht verändert. Ansonsten werden alle Schlüsselwörter, die erkannt werden, in Tokens (1 Byte 80H-FFH) umgewandelt. ' wird in :REM' und ELSE in :ELSE umgewandelt. Beim Colour Genie werden manche Schlüsselwörter (insbes. Grafik- und Soundanweisungen) durch Doppeltokens beginnend mit einem 0FFH ersetzt. Die Tokentabellen können den ROM-Listings entnommen werden.

Beispiel:

```
LD      HL,TEXT      ; Zeiger auf einen Text
CALL   ENCTXT       ; Zwischencode erzeugen
...
TEXT   DEFM  'FORI=1T09'
       DEFB  0
```

```
*****
*
* GENTXT : Klartext aus Zwischencode erzeugen
*
*     E : 2B7EH
*     B : ja
*     R : AF, BC, DE, HL
*     I : HL = Zeiger auf Zwischencode
*           mit 0H abgeschlossen
*           (HL) = 1. Byte des Zwischencode
*     O : BC = Zeiger auf letztes Byte des Klartext
*           HL = Zeiger auf erstes Byte hinter 0H des
*           Zwischencode (Ausnahme siehe Funktion)
*           (HL-1) = 0H
*           ((IOBUFF)) = 1. Zeichen des Klartext
*           (BC) = 0 : letztes Zeichen des Klartext
*
*****
```

Funktion:

Der Zwischencode, auf den HL weist, wird in Klartext zurückgewandelt. Überschreitet der Klartext den I/O-Buffer, wird er nach 255 Zeichen abgeschnitten. Nach der Rückkehr liegt der Klartext, abgeschlossen mit einer 0H (auf die BC weist), im I/O-Buffer vor. HL ist bei der Rückkehr so fortgeschrieben, daß es auf das Byte hinter der abschließenden Null weist. Wurde die Umwandlung wegen Erreichens des I/O-Bufferendes abgebrochen, zeigt HL auf die Stelle des Zwischencodes, bei der die Umwandlung abgebrochen wurde.

Beispiel:

```
LD      HL,ZCODE      ; Zeiger auf Zwischencode
CALL    GENTXT        ; Klartext in I/O-Buffer
...
ZCODE   DEFB  81H
        DEFB  'I'
        DEFB  0CDH
        DEFB  '1'
        DEFB  0B0H
        DEFB  '9'
        DEFB  0
```

```
*****
*
* SLINE : Programmzeile suchen
*
*   E : 1B2CH
*   B : ja
*   R : AF, BC, HL
*   I : DE = Zeilennummer der gesuchten Zeile
*   O : gesuchte Zeile vorhanden :
*       CY=1, Z=1
*       BC = Adresse der gesuchten Zeile
*       HL = Adresse der nächsten Zeile
*   gesuchte Zeile nicht vorhanden :
*       CY=0, Z=0
*       BC = Adresse der nächst größeren Zeile
*       HL = Adresse der darauffolgenden Zeile
*   gesuchte Zeile nicht vorhanden und keine größer
*       CY=0, Z=1
*       BC = (TXTEND) - 2
*       HL = (TXTEND) - 2
*
*****
```

Funktion:

Die Zeile zur Zeilennummer in DE wird gesucht. Wird sie gefunden, wird ihre Speicheradresse und die der folgenden Zeile zurückgegeben, und das positive Vergleichsergebnis durch ein Flag (Z, CY) angezeigt. Existiert die gesuchte Zeile nicht, aber eine mit einer größeren Zeilennummer, wird bei deren Auffinden die Suche abgebrochen, ansonsten bei Erreichen des Programmtextendes.

Wird die Adresse SLINE+3 statt SLINE angesprungen, wird die Suche nicht am Programmtextanfang (TXTANF), sondern bei der Zeile begonnen, auf die HL bei Aufruf weist. HL muß auf den Anfang einer Zeile (= Zeilenzeiger) weisen.

Beispiel:

```
LD      DE,1000      ; ZeilenNr = 1000
CALL    SLINE        ; Zeile suchen
JR      NC,LABEL     ; nicht vorhanden, -->
INC     BC           ; Zeiger auf Zeilentext
INC     BC
INC     BC
INC     BC
...
; Programmteil für Fehlen der Zeile
LABEL   ...
```

```
*****
*
* LPNT   : Zeilenzeigersystem erneuern
*
*       E : 1AF8H
*       B : Ja
*       R : AF, DE, HL
*       I : /
*       O : HL = (TXTEND) - 1
*
*****
```

Funktion:

Alle Zeilenzeiger im gesamten Programmtext werden neu ermittelt und abgespeichert. Diese Routine wird nach jedem Einfügen oder Löschen einer Zeile und jedem Laden eines Programmes von einem Massenspeicher (Kassette oder Diskette) aufgerufen.

Wie bei SLINE kann durch Aufruf bei LPNT+3 auch eine teilweise Erneuerung der Zeilenzeiger erreicht werden. Dabei gelten die gleichen Eintrittsbedingungen wie oben (SLINE+3).

Beispiel:

```
CALL LPNT ; Zeilenzeiger kompl. ern.
```

1.3 Die Variablentabelle

Die Variablentabelle besteht aus 2 Blöcken, je einem für die einfachen und die Feldvariablen: Jede einfache und Feldvariable beginnt mit Typ und Variablenname in 3 Bytes. Bei den einfachen Variablen folgt dann sofort der Wert, der so viele Bytes belegt, wie der Typ angibt. Bei den Feldvariablen folgen zunächst die Informationen zur Größe (in Bytes) und Dimensionierung des Feldes, und dann die Werte aller Feldelemente in einer bestimmten Reihenfolge.

einfache Variablen:

! Typ!Name ! Wert --->

Integer:

+-----+-----+-----+-----+-----+
! 02 ! 2Z ! 1Z !LSB !MSB !
+-----+-----+-----+-----+-----+

Single:

+-----+-----+-----+-----+-----+-----+
! 04 ! 2Z ! 1Z !LSB !LSB !MSB !EXP !
+-----+-----+-----+-----+-----+-----+

Double:

+-----+-----+-----+-----+-----+-----+-----+-----+
! 08 ! 2Z ! 1Z !LSB !LSB !LSB !LSB !LSB !LSB !MSB !EXP !
+-----+-----+-----+-----+-----+-----+-----+-----+

String:

+-----+-----+-----+-----+-----+
! 03 ! 2Z ! 1Z !Län !LSB !MSB !
+-----+-----+-----+-----+-----+

Beispiele:

Integer: A%=100

+-----+-----+-----+-----+-----+
! 02 ! 00 ! 41 ! 64 ! 00 !
+-----+-----+-----+-----+-----+
! Typ! Name=A !Wert=100 !
+-----+-----+-----+-----+-----+

Single: S1!=4.5

+-----+-----+-----+-----+-----+-----+
! 04 ! 31 ! 52 ! 00 ! 00 ! 10 ! 83 !
+-----+-----+-----+-----+-----+-----+
! Typ! Name=S1 ! Wert = 0.1001*2X3 !
+-----+-----+-----+-----+-----+-----+

Double: DD#=.1#

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! 08 ! 44 ! 44 ! CD ! CC ! CC ! CC ! CC ! CC ! 4C ! 7D !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! 08 ! Name=DD ! Wert = 0.11001100...11001101*2X-3      !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

String: D\$="ABCDEF"

```

+-----+-----+-----+-----+-----+-----+
! 03 ! 00 ! 44 ! 06 ! 74 ! 52 !
+-----+-----+-----+-----+-----+-----+
! 03 ! Name=D  !L=6 !Adr.=5274!
+-----+-----+-----+-----+-----+-----+

```

Feldvariablen:

Feldvariablen bestehen aus Feldkopf und den Werten. Der Feldkopf enthält Typ, Name, Größe, Anzahl der Dimensionen (nd) und die Tiefen aller Dimensionen (beginnend mit der letzten). Die Tiefen entsprechen den Indizes der DIM-Anweisung + 1. Die Größe entspricht der Länge des Feldes in Bytes beginnend mit dem ihr folgenden Byte (nd) einschließlich aller Feldelemente:

Feldkopf:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! Typ! 2Z ! 1Z ! Größe   ! nd ! Tiefe nd! .. ! Tiefe 1 !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Die Feldwerte werden nach folgenden Schema abgespeichert: Der Tiefe 1 folgt das Feldelement (0,...,0). Die Adresse der anderen Feldelemente errechnet sich (3-DIM):

$$\text{Adr.}(I1, I2, I3) = \text{Adr.}(0, 0, 0) + ((I3 * T2 + I2) * T1 + I1) * \text{Typ}$$

Ii sind die Indizes und Ti die Tiefen der Dimensionen. Der Typ entspricht der Anzahl der Bytes, den ein Wert eines Feldelementes beansprucht und hat deshalb auch seinen Platz in der Formel. Die Größe errechnet sich folgendermaßen:

$$\text{Größe} = 1 + 2 * \text{nd} + \text{Typ} * T1 * T2 * \dots * \text{Tnd}$$

Beispiel:

```

10 DIM A%(1,2,3)
20 I=0
30 FOR I3=0 TO 3
40 FOR I2=0 TO 2
50 FOR I1=0 TO 1
60 A%(I1,I2,I3)=I
70 I=I+1.
80 NEXT I1,I2,I3

```

Feldkopf:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
! 02 ! 00 ! 41 ! 37 ! 00 ! 03 ! 04 ! 00 ! 03 ! 00 ! 02 ! 00 !
+-----+-----+-----+-----+-----+-----+-----+-----+
! 08 ! Name=A ! Größe ! nd ! Tiefe 3 ! Tiefe 2 ! Tiefe 1 !
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Feldwerte:

```

+-----+-----+-----+-----+-----+-----+-----+-----+
! 00 ! 00 ! 01 ! 00 ! 02 ! 00 ! .. ! 16 ! 00 ! 17 ! 00 !
+-----+-----+-----+-----+-----+-----+-----+-----+
!A%(0,0,0)!A%(1,0,0)!A%(0,1,0)! .. !A%(0,2,3)!A%(1,2,3)!
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Die Variablentabelle ist durch 3 Zeiger folgendermaßen definiert:

A%=100:DIM A%(1,2,3)

```

          +- (TXTEND)      (VAREND)  -+          (ARREND)  -+
          V                  V                  V
+-----+-----+-----+-----+-----+-----+-----+-----+
! E9 ! EA ! EB ! EC ! ED ! EF ! F0 ! F1 ! F2 ! .. ! 2C ! 2D !
+-----+-----+-----+-----+-----+-----+-----+-----+
! 00 ! 00 ! 02 ! 00 ! 41 ! 64 ! 00 ! 02 ! 00 ! .. ! 00 ! ? !
+-----+-----+-----+-----+-----+-----+-----+-----+
!0-Zeil.z.! Typ! Name=A !Wert=100 ! Typ!2.Z ! .. ! .. ! .. !
+-----+-----+-----+-----+-----+-----+-----+-----+
! Text      ! einfache Variablen      ! Feldvariablen      !

```

```

*****
*
* SVAR   : Variable suchen und einrichten
*
*     E : 260DH
*     B : ja
*     R : AF, BC, DE, HL=Pointer
*     I : HL = zeigt auf Variablennamen
*         (HL) = 1. Zeichen des Variablennamens
*     O : DE zeigt auf Wert der Variablen
*         (bei Feldern auf Element (I1,...Ind))
*
*           Integer Single Double String
*     (DE)  =   LSB     LSB Man  LSB M   Länge
*     (DE+1) =   MSB     LSB tis  LSB a   LSB Adr.
*     (DE+2) =           MSB se   LSB n   MSB Adr.
*     (DE+3) =           EXP      LSB t
*     (DE+4) =           LSB i
*     (DE+5) =           LSB s
*     (DE+6) =           MSB se
*     (DE+7) =           EXP
*     (TYPX) = Typcode der Variablen
*
*           2       4       8       3
*
*     HL zeigt hinter Variablenname
*     (HL) = 1. signifikantes Zeichen hinter Varname
*
*   -> Außerdem bei einfachen Variablen:
*     (DE-3) = Typcode der Variablen
*     (DE-2) = 2. Zeichen des Variablennamens(oder 0)
*     (DE-1) = 1. Zeichen des Variablennamens
*
*   -> Außerdem bei Feldvariablen:
*     BC = Adr. (0,...,0) Adresse des Feldelementes
*         (0,...,0)
*
*   -> Bei neu eingerichteten Variablen:
*     (VAREND) + Typ : einfache Variablen
*     (ARREND) + Typ :      "      "
*     (ARREND) + Größe + 5 : Feldvariablen
*
*****

```

Funktion:

Bei Eintritt muß HL auf das 1. Zeichen des Variablennamens weisen. SVAR analysiert Variablenname, -Typ und unterscheidet zwischen einfachen und Feldvariablen. Ist (ARRFLG) gesetzt, werden keine Feldvariablen akzeptiert. Die Variable wird gesucht und ein Zeiger auf den Wert zurückgegeben, falls sie gefunden wird. Andernfalls wird sie neu eingerichtet (mit 2 Ausnahmen: Aufruf von 24F1H (VARPTR) oder 2543H (EXP), da beide Fälle in selbst entwickelten Programmen nicht vorkommen, spielen sie hier keine Rolle, siehe ROM-Listing) und danach ebenfalls der Zeiger auf den Wert zurückgegeben. Felder werden mit einer Tiefe von 11 in jeder Dimension eingerichtet. Bei Feldern wird in DE ein Zeiger auf das spezifizierte Element zurückgegeben. Will man ein komplettes Feld bearbeiten, so gibt man für alle Dimensionen den höchsten Index an und erhält nach Aufruf von SVAR in DE die Adresse des letzten Feldelementes. Alle Feldelemente belegen dann den Speicherbereich (BC..DE + Typ - 1) (siehe Beispiel Durchschnitt eines Feldes).

Beispiel:

```

LD      HL,TEXT      ; Pointer setzen
CALL   SVAR         ; Variable suchen/einr.
LD     A,100        ; Wert (%) = 100
LD     (DE),A       ; (LSB)
LD     A,0
LD     (DE),A INC   DE ; (MSB)
...
TEXT:  DEFM 'A%'    ; Text
*****
*
* SVARBC : einfache Variablen suchen und einrichten *
*
*   E : 2664H *
*   B : ja *
*   R : AF, BC, DE *
*   I : B = 1. Zeichen des Variablennamens *
*       C = 2. Zeichen des Varnamens (0, wenn nur 1 Z.)*
*       E = Typ der Variablen (2, 3, 4, 8) *
*           (Auch alle anderen Werte erlaubt !) *
*   O : DE zeigt auf Wert der Variablen *
*       (DE) = s. o. SVAR *
*       (TYPX) wird nicht (!) gleich E gesetzt ! *
*
*****

```

Funktion:

SVARBC ist ein weiterer Einsprung der SVAR-Routine, bei dem die Analyse des Variablennamens übersprungen wird. Außerdem wird die Abfrage auf Feldvariablen ausgelassen. Daher kann nur nach einfachen Variablen gesucht werden, die wie bei SVAR ggf. neu eingerichtet werden. Die Register B und C müssen den Variablennamen und E den Typ enthalten. Für die Typen Integer, Single, Double und String erhält E den Wert 2, 3, 8 oder 4. Mit anderen Werten von E kann ein beliebiger eigener Typ definiert werden. E gibt dabei wie bei den Standardtypen die Länge des Speicherbereiches für den Variablenwert an. Diese eigenen Typen können nur mit der Routine SVARBC angesprochen werden und sind für den Basicinterpreter unsichtbar.

Beispiel:

```

LD     B,'A'        ; Varname = A
LD     C,0          ; 2. Zeichen = 0
LD     E,2          ; Typ = Integer
CALL   SVAR         ; Variable suchen/einr.
LD     A,100        ; Wert (%) = 100
LD     (DE),A       ; (LSB)
LD     A,0
LD     (DE),A INC   DE ; (MSB)

```

1.4 Arithmetik

Die Formate für die drei numerischen Datentypen wurden schon im Abschnitt 1.3 erläutert. Integerzahlen werden in 2-Komplementdarstellung abgespeichert:

Beispiele:

Speicher:	! binär:	! dezimal:
LSB MSB	! MSB LSB	!
01H 00H	! 0000 0000 0000 0001B	! 1
0AH 00H	! 0000 0000 0000 1010B	! 10
E8H 03H	! 0000 0011 1110 1000B	! 1000
FFH FFH	! 1111 1111 1111 1111B	! -1

Single und Double werden als Fließkommazahlen mit 2-Exponent abgespeichert:

Speicher:	! binär	! dezimal:
Mantisse EXP	! Exponent Mantisse	!
LSB MSB	! MSB LSB	!
? ? ? 00H	! 0000 0000 ???? ???? ...	! 0
00H 00H 00H 81H	! 1000 0001 0000 0000 ...	! 0.5 * 2 ^{X1} = 1
00H 00H 80H 81H	! 1000 0001 0000 0000 ...	! -0.5 * 2 ^{X1} = -1
CDH CCH 4CH 7DH	! 0111 1101 0100 1100 ...	! 0.8 * 2 ^{X-3} = .1

Zum dritten Beispiel:

Die Mantisse lautet:

$$0.110011001100110011001101B = 0.5 + 0.25 + 0.03125 + 0.15625 \\ \dots = 0.8$$

Das höchste Bit der Mantisse ist immer gesetzt und daher im obigen Speicherformat durch das Vorzeichen (0 = plus, 1 = minus) ersetzt. Der Exponent ist mit einem Offset von 80H abgespeichert. Ein Exponent 0 zeigt den Wert 0 an, in diesem Fall ist der Wert der Mantisse irrelevant.

Das Fließkommaformat in doppelter Genauigkeit ist genauso aufgebaut, mit 4 zusätzliche Bytes für die Mantisse.

Die Fließkommaakkumulatoren:

Für die Rechenoperationen, Ausdrucksauswertung, etc werden zwei Pseudoakkumulatoren im Speicher benutzt, in denen Werte aller 4 Datentypen abgelegt werden können:

FACX:

	Integer	String	Single	Double
411DH				LSB M
411EH				LSB a
411FH				LSB n
4120H				LSB t
4121H	LSB	LSB Zei	LSB Ma	LSB i

4122H	MSB	MSB ger	LSB ti	LSB s
4123H			MSB sse	MSB se
4124H			EXP	EXP

FACY:

	Integer	String	Single	Double
4127H	LSB	LSB Zei	LSB Ma	LSB M
4128H	MSB	MSB ger	LSB ti	LSB a
4129H			MSB sse	LSB n
412AH			EXP	LSB t
412BH				LSB i
412CH				LSB s
412DH				MSB se
412EH				EXP

Für Strings wird in FACX und FACY nur der Zeiger auf den Descriptor (Stringlänge und -zeiger) in der Variablen-tabelle oder im Descriptoren-zwischenspeicher abgelegt.

Die Rechenoperationen:

```

*****
*
* I??? : Operationen mit Integerzahlen
*
* E : 0BD2 IADD Addition (FACX) = DE + HL
* 0BC7 ISUB Subtraktion (FACX) = DE - HL
* 0BF2 IMLT Multiplikation (FACX) = DE * HL
* 2490 IDIV Division (FACX) = DE / HL
*
* B : ja
* R : AF, BC, DE, HL
* I : DE = 1. Operand
* HL = 2. Operand
* O : (FACX) = Ergebnis
* (TYPX) = Typ des Ergebnisses (2 oder 4)
* und wenn (TYPX) = 2:
* HL = Ergebnis
*
*****

```

Funktion:

Summe, Differenz, Produkt oder Quotient von HL und DE werden berechnet. Überschreitet das Ergebnis nicht die Grenzen für Integerzahlen (-32768..32767) wird es in HL und (FACX) zurückgegeben. Ansonsten werden die Operanden in Single umgewandelt und die Singleoperationen durchgeführt. Die Division wird immer nur in Single ausgeführt, HL = 0 führt zu Fehlerabbruch.

Beispiel:

```

LD HL,(ZAH1) ; 1. Faktor laden
LD DE,(ZAH2) ; 2. Faktor laden
CALL IMLT ; Multiplikation
LD A,(TYPX) ; Ergebnis Single ?
CP 4

```

```
JP      Z,FEHLER      ; ja, --> Fehlerbehandlung
LD      (ERG),HL      ; Ergebnis abspeichern
```

```
*****
*
* S???? : Operationen mit Singlezahlen
*
*      E : 0716 SADDY Addition      (FACX) = BCDE + (FACX)*
*          0713 SSUBY Subtraktion   (FACX) = BCDE - (FACX)*
*          0847 SMLTY Multiplikation(FACX) = BCDE * (FACX)*
*          08A2 SDIVY Division      (FACX) = BCDE / (FACX)*
*      B : ja
*      R : AF, BC, DE, HL
*      I : BCDE = 1. Operand
*          B = EXP, C = MSB, DE = LSB Mantisse
*          (FACX) = 2. Operand
*      O : (FACX) = Ergebnis
*
*****
```

Funktion:

Summe, Differenz, Produkt oder Quotient von BCDE und (FACX) werden berechnet. Überlauf (EXP-Ergebnis > FFH) wird mit "OV"-Fehler geahndet. Bei Unterlauf (EXP-Ergebnis < 01H) wird das Ergebnis durch 0 ersetzt. Bei Aufruf der Division mit (FACX)=0 wird ein "/0"-Fehler erkannt.

Beispiel:

```
LD      B,81H          ; BCDE = 1
LD      C,0H
LD      DE,0
CALL    SADDY          ; Addition (FACX) + 1
```

```
*****
*
* D??? : Operationen mit Doublezahlen
*
*      E : 0C77 DADD Addition      (FACX) = (FACX)+(FACY)*
*          0C70 DSUB Subtraktion   (FACX) = (FACX)-(FACY)*
*          0DA1 DMLT Multiplikation(FACX) = (FACX)*(FACY)*
*          0DE5 DDIV Division      (FACX) = (FACX)/(FACY)*
*      B : ja
*      R : AF, BC, DE, HL
*      I : (FACY) = 1. Operand
*          (FACX) = 2. Operand
*      O : (FACX) = Ergebnis
*
*****
```

Funktion:

Summe, Differenz, Produkt oder Quotient von (FACX) und (FACY) werden berechnet. Weiteres s. o.

Beispiel:

```
CALL    DADD          ; Addition
```

Die Funktionen:

Elementare Funktionen mit Argumenten verschiedenen Typs

```
*****
*
* ????   : Funktionen mit Argumenten verschiedenen Typs
*
*   E : 0B37 INT   (FACX) = INT  ( (FACX) )
*       0977 ABS   (FACX) = ABS  ( (FACX) )
*       098A SGN   (FACX) = SGN  ( (FACX) )
*       0B26 FIX   (FACX) = FIX  ( (FACX) )
*       14C9 RND   (FACX) = RND  ( (FACX) )
*       0A7F CINT  (FACX) = CINT ( (FACX) )
*       0AB1 CSNG  (FACX) = CSNG ( (FACX) )
*       0ADB CDBL  (FACX) = CDBL ( (FACX) )
*
*   B : ja
*   R : AF, BC, DE, HL
*   I : (FACX) = Argument
*       (TYPX) = Typ des Argumentes
*   O : (FACX) = Ergebnis
*       (TYPX) = Typ des Ergebnisses
*
*****
```

Funktion:

Das Argument beliebigen Typs muß in (FACX) vorliegen. Das Ergebnis liegt nach Aufruf wiederum in (FACX) vor. Der Typ des Argumentes und Ergebnisses wird durch (TYPX) angegeben. Die Beschreibung der Funktion kann dem Basichandbuch entnommen werden.

Beispiel:

```
LD     HL,(ZAHL1)      ; ZAHL1 --> (FACX)
LD     (FACX),HL
LD     A,2             ; (TYPX) = Integer
LD     (TYPX),A
CALL   SGN             ; SGN-Funktion
                        ; (Ergebnis immer Integer)
LD     HL,(FACX)      ; Ergebnis nach HL
```

```
*****
*
* ???    : Tranzendente Funktionen
*
*   E : 13E7 SQR   (FACX) = SQR  ( (FACX) )
*       1439 EXP   (FACX) = EXP  ( (FACX) )
*       0809 LOG   (FACX) = LOG  ( (FACX) )
*       1547 SIN   (FACX) = SIN  ( (FACX) )
*       1541 COS   (FACX) = COS  ( (FACX) )
*       15A8 TAN   (FACX) = TAN  ( (FACX) )
*       15BD ATN   (FACX) = ATN  ( (FACX) )
*
*   B : ja
*   R : AF, BC, DE, HL
*   I : (FACX) = Argument
*   O : (FACX) = Ergebnis
*
*****
```

Funktion:

Das Argument muß immer im Singleformat in (FACX) vorliegen.
Das Ergebnis liegt nach Aufruf wiederum in (FACX) als
Single vor.

Beispiel:

```
LD      BC,8400H      ; BCDE = 8
LD      DE,0000H
LD      (FACX+4),DE   ; BCDE nach (FACX)
LD      (FACX+6),BC
CALL    SQR           ; SQR-Funktion
```

Umwandlung von Zeichenfolgen in Zahlen:

```
*****
*
* LVAL   : Zeichenfolge in Zahl umwandlen
*
*       E : 0E6CH
*       B : ja
*       R : AF, BC, DE, HL=Pointer
*       I : HL zeigt auf Zeichenfolge
*           (HL) = 1. Zeichen
*       O : (FACX) = Zahl
*           (TYPX) = Typ der Zahl
*           HL zeigt hinter Zeichenfolge
*           (HL) = 1. Zeichen, welches nicht sinnvoll als
*               Zahl zu interpretieren ist
*
*****
```

Funktion:

Die Zeichenfolge, auf die HL weist, wird in eine Zahl in Binärdarstellung umgewandelt. Es wird der "kleinste" Zahlentyp benutzt, der eine möglichst exakte Darstellung ergibt.

Beispiel:

```
LD      HL,TEXT      ; Textzeiger
CALL    LVAL
LD      A,(TYPX)     ; Typ = Integer ?
CP      2
JR      NZ,...       ; nein, -->
LD      HL,(FACX+4)  ; Zahl nach HL
...
TEXT   DEFM  '1534'
       DEFB  0
```

```

*****
*
* FSTR   : (FACX) in Zeichenfolge umwandeln
*
*       E : 0FBEB
*       B : ja
*       R : AF, BC, DE, HL
*       I : (FACX) = umzuwandelnde Zahl
*           (TYPX) = Typ von (FACX)
*           A = Formatflag
*           Bit 7 = 0 : keine Formatierung (B, C n. v. )
*           Bit 7 = 1 : Formatierung (PRINT USING)
*               Bit 0 = 1 : Exponentenausgabe
*               Bit 1 = 1 : n. v.
*               Bit 2 = 1 : Vorzeichen hinter Zahl
*               Bit 3 = 1 : '+' auch drucken
*               Bit 4 = 1 : '$' vor Zahl drucken
*               Bit 5 = 1 : führende ' ' durch '*' ersetzen
*               Bit 6 = 1 : ',' zwischen Dreiergruppen
*           B = Vorkommafeldlänge (max. Anz. der
*                               Vorkommastellen
*           C = Nachkommafeldlänge (max. Anz der Nach-
*                               kommastellen + 1)
*       O : HL zeigt auf Zeichenfolge
*           (HL)   = 1. Zeichen
*           (HL+n) = 0 (Endmarkierung)
*           HL zeigt immer auf den NUMBUF (412FH-4149H)
*           (FACX) wird verändert !
*
*****

```

Funktion:

Die Zahl in (FACX) (beliebiger Typ) wird in eine Zeichenkette umgewandelt, die im NUMBUF abgelegt wird. Der Beginn der Zeichenkette stimmt nicht immer mit dem Anfang von NUMBUF überein, da führende Nullen entfernt werden. Bei Rückkehr von FSTR zeigt HL auf den Anfang der Zeichenkette, die mit einer 0H als Endmarkierung abgeschlossen wird. Ist beim Eintritt in FSTR Bit 7 des Akkus = 0, wird keine Formatierung durchgeführt. Andernfalls wird der Zahlenstring formatiert. Die Parameter der Formatierung entsprechen den verschiedenen Möglichkeiten der USING-Anweisung (s. Basicanleitung).

Beispiel:

```

; Zahl in (FACX) formatiert ausdrucken
LD     B,2           ; 2 Vorkommastellen
LD     C,3           ; 2 Nachkommastellen
LD     A,80H        ; Formatierung erzwingen
CALL   FSTR         ; Zahlenstring in NUMBUF
LD     A,(HL)       ; 1. Zeichen des Zahlenstr.
...

```



```
*****
*
* USTR   : (FACX) in Zeichenfolge umwandeln
*
*       E : 0FBDBH
*       B : ja
*       R : AF, BC, DE, HL
*       I : (FACX) = umzuwandelnde Zahl
*           (TYPX) = Typ von (FACX)
*       O : HL zeigt auf Zeichenfolge
*           (HL)   = 1. Zeichen
*           (HL+n) = 0 (Endmarkierung)
*           HL zeigt immer auf den NUMBUF (412FH-4149H)
*           (FACX) wird verändert !
*
*****
```

Funktion:

Wie FSTR, mit der Ausnahme, daß eine Formatierung nicht möglich ist.

Beispiel:

```
        ; Zahl in HL als Integer in 2-Komplement ausgeben
CALL    ILDXHL      ; HL nach (FACX), (TYPX)=2
CALL    USTR        ; Zeichenfolge erzeugen
CALL    PRINTT      ; und ausgeben
```

```
*****
*
* PRZNR  : HL als Integer ohne Vorzeichen ausgeben
*
*       E : 0FAFH
*       B : ja
*       R : AF, BC, DE, HL
*       I : HL = auszugebende Zahl
*       O : /
*
*****
```

Funktion:

HL wird als Integer ohne Vorzeichen (also 0..65535) ausgegeben.

Beispiel:

```
LD      HL,(NUMMER) ; NUMMER ausgeben
CALL    PRZNR
```

1.5 Stringverarbeitung

Die Strings werden in der Variablen-tabelle (siehe 1.4) nur durch Länge und Adresse definiert. Auf der dort genannten Adresse stehen in den der Länge entsprechenden Bytes die einzelnen Zeichen der Zeichenkette. Die meisten Strings liegen im Stringspace, ein spezieller Speicherbereich. Dieser wird von oben beginnend mit den Strings gefüllt. Ist bei der Eintragung eines neuen Strings nicht mehr genügend Platz frei, wird durch Reorganisation des Stringspace (Garbage collection) der freie Platz gesammelt.

Das obere Ende des Stringspace (SSPEND) entspricht der letzten RAM-Zelle, bzw. der Eingabe auf "MEM-SIZE". Der Anfang des Stringspaces (SSPANF) wird durch die CLEAR n-Anweisung definiert. (SSPNT) zeigt die momentane Belegung des Stringspaces an und weist immer auf das letzte freie Byte.

Von den im Basicinterpreter vorhandenen Routinen zur Stringverarbeitung sind die meisten in eigenen Maschinenprogrammen nicht sinnvoll einzusetzen, daher seien hier nur zwei Routinen näher beschrieben.

```
*****
*
* GARCOL : Stringspace reorganisieren (Garbage collection)*
*
*   E : 28E6H
*   B : ja
*   R : AF = 0, BC, DE, HL = 0
*   I : /
*   O : /
*
*****
```

Funktion:

Der Stringspace wird reorganisiert indem alle noch benötigten Zeichenketten nach oben so dicht wie möglich zusammengeschoben werden.

Beispiel:

```
CALL    GARCOL           ; Garbage collection
LD      HL,(SSPNT)       ; freien Platz im String-
LD      DE,(SSPANF)     ; space berechnen
OR      A                ; eigentlich überflüssig !
SBC     HL,DE            ; --> HL
CALL    PRZNR            ; ausgeben
```

```
*****  
*  
* PRINTT : String ausgeben  
*  
*     E : 2B75H  
*     B : ja  
*     R : AF, HL  
*     I : HL = Zeiger auf String  
*         (HL) = 1. Zeichen des Strings  
*     O : HL = Zeiger auf Null am Ende des Strings  
*         (HL) = 0  
*  
*****
```

Funktion:

Der String auf den HL weist wird Zeichen für Zeichen über die Routine OUTCHF auf das durch (OUTFLG) definierte Ausgabemedium ausgegeben.

Beispiel:

```
LD     HL,TEXT      ; Zeiger laden  
CALL  PRINTT       ; String ausgeben  
;  
TEXT  DEFM  'Das ist ein Beispiel'  
      DEFB  0
```

1.6 Ausdruckauswertung

Bei vielen Anweisungen werden die Parameter in Form von Ausdrücken angegeben, die es ermöglichen, zahlreiche Operanden durch verschiedene Operatoren und Funktionen zu verknüpfen.

```
*****
*
* EXP      : Ausdruck auswerten
*      E : 2337H
*      B : ja
*      R : AF, BC, DE, HL = Pointer
*      I : HL = Zeiger auf Ausdruck
*          (HL) = 1. Zeichen des Ausdruckes
*      O : HL = Zeiger hinter Ausdruck
*          (HL) = 1. Zeichen, welches nicht als Bestand-
*              teil des Ausdruckes interpretiert
*              werden kann.
*          (FACX) = Ausdrucksergebnis (wenn kein Fehler)
*              Bei Strings, Zeiger auf Descriptoren-
*              zwischenspeicher
*          (TYPX) = Typ des Ergebnisses
*
*****
```

Funktion:

Der Ausdruck, auf den HL weist, wird ausgewertet und das Ergebnis in (FACX) zurückgegeben. Der Descriptor von Stringergebnissen wird im Descriptorenzwischenspeicher abgelegt und der Zeiger darauf in (FACX).

Beispiel:

```
FACX EQU 4121H ; FACX = 4121H
      LD HL,TEXT ; Zeiger auf Ausdruck
      CALL EXP ; Ausdruck auswerten
      LD A,(TYPX) ; Ergebnis String ?
      CP 3
      JR NZ,... ; nein, -->
      LD HL,(FACX) ; Zeiger laden
      LD A,(HL) ; Stringlänge nach A
      INC HL
      LD E,(HL) ; Stringadresse nach DE
      INC HL
      LD D,(HL)
      ...
TEXT DEFM 'SQR(2*2+3*3)'
      DEFB 0
```

Es existieren weiterhin verschiedene Routinen, die sich ihrerseits der EXP-Routine bedienen, aber deren Ergebnis speziell weiterverarbeiten, um es dem Aufrufer in einer geeigneteren Form zur Verfügung zu stellen:

```
*****
*
* INTEXP : Ausdruck auswerten und Erg. in Integer umwand. *
*
*     E : 2B02H
*     B : ja
*     R : AF, BC, DE, HL = Pointer
*     I : siehe EXP
*     O : HL siehe EXP
*         (FACX) = Ausdrucksergebnis (wenn kein Fehler)
*         (TYPX) = 2 (Integer)
*         DE = Ergebnis
*         A = MSB Ergebnis (= D)
*
*****
```

Funktion:

Nach der Ausdrucksauswertung durch EXP wird das Ergebnis in Integer umgewandelt und nach DE geladen.

Beispiel:

```
LD      HL,TEXT      ; Ausdruck
CALL    INTEXP       ; auswerten
PUSH    HL           ; Pointer retten
EX      DE,HL        ; Ergebnis ausgeben
CALL    PRZNR
POP     HL           ; Pointer zurück
```

```
*****
*
* IEXPA  : Ausdruck auswerten und Erg. in Integer umwand. *
*
*     E : 2B1CH
*     B : ja
*     R : AF, BC, DE, HL = Pointer
*     I : siehe EXP
*     O : HL siehe EXP
*         (FACX) = Ausdrucksergebnis (wenn kein Fehler)
*         (TYPX) = 2 (Integer)
*         A = Ergebnis (< 256, sonst Fehler)
*
*****
```

Funktion:

Nach der Ausdrucksauswertung durch EXP wird das Ergebnis in Integer umgewandelt und nach A geladen, wenn es kleiner als 256 ist, ansonsten wird mit Fehler abgebrochen.

Beispiel:

```
LD      HL,TEXT      ; Ausdruck
CALL    IEXPA        ; auswerten
LD      (EXP),A      ; Ergebnis abspeichern
```

1.7 Aufzeichnung auf Kassette

Für die Kassettenaufzeichnung gibt es drei Formate:

- 1) Systemfileformat für Maschinenprogramme
- 2) Basicprogrammformat
- 3) Datenformat

Alle Formate beginnen mit einem Header und einem Synchronisationsbyte. Diese unterscheiden sich zwischen den verschiedenen Systemen:

	Header	Synchronbyte
TRS-80 Model I, Genie I, II, III 500 Baud	255 * 00H	1 * A5H
TRS-80 Model III 500 Baud	83 * 00H	1 * A5H
1500 Baud	256 * 55H (beides ohne Taktimpuls)	1 * 7FH
Colour-Genie 1200 Baud	255 * AAH	1 * 66H

Header und Synchronbyte führen jede Kassettenaufzeichnung an. Dahinter folgen die eigentlichen Daten in einem der drei Formate: Diese Formate sind bei allen Systemen identisch, trotzdem ist ein Austausch von Kassetten zwischen den Formaten mit verschiedenen Baudraten nicht möglich.

1) Systemformat

Filenamblock:	Filenameheader	55H
	Filename	6 Zeichen
N Datenblöcke:	Datenblockheader	3CH
	Blocklänge	0-0FFH (0 = 256 Bytes)
	Anfangsadresse	LSB/MSB
	Datenbytes	1 - 256 Bytes
	Prüfsumme	1 Byte
	über alle Datenbytes und Anfangsadresse	
Entrypointblock:	Entrypointheader	78H
	Entrypoint	LSB/MSB

Vor jedem Block können beliebig viele Bytes (<> 55H vor Filenamblock, <> 3CH und <> 78H vor den anderen Blöcken) stehen, die beim Lesen ignoriert werden.

Beispiel:

Aufzeichnung mit NAME "BSP",5,10,0 (siehe Beispiel 5)

```

Filenamblock:  Filenameheader  55H
                Filename        42H 53H 50H 20H 20H 20H
1 Datenblock:  Datenblockheader 3CH
                Blocklänge      06H
                Anfangsadresse  05H 00H
                Datenbytes      C3H 00H 40H C3H 00H 40H
                Prüfsumme       0BH
Entrypointblock: Entrypointheader 78H
                  Entrypoint     00H 00H

```

2) Basicprogrammfiles

Das Basicprogramm wird genau so wie es im Speicher steht auf Kassette aufgezeichnet. Zuvor wird 3 * D3H und ein Byte für den Filenam aufgezeichnet. Beim Colour-Genie entfallen die 3 * D3Hs.

3) Datenformat

Beim Datenformat wird direkt nach dem Header die Ausgabe, so wie sie auch auf dem Bildschirm erscheinen würde, auf Kassette aufgezeichnet. Abgeschlossen wird die Aufzeichnung mit einem CR (=0DH). Aufgezeichnet werden können beliebig viele Byte. Die INPUT#-1 akzeptiert aber nur 250 Bytes, der Rest wird abgeschnitten. Beim ersten Auftreten des CR wird ebenfalls abgebrochen.

Die einzelnen Kassettenroutinen sind zwar funktional weitgehend identisch, haben aber bei Colour-Genie und TRS-80 Model III andere Einsprünge:

```

*****
*
* CASON : Kassettenrecorder einschalten
*
*   E : 0215H (TRS-80 Model I, Genie I, II, III)
*       31E8H (TRS-80 Model III)
*       ---- (Colour Genie)
*   B : nein
*   R : AF
*   I : /
*   O : /
*
*****

```

Funktion:

Der Kassettenrecorder wird eingeschaltet, indem das Bit 2 von Port 0FFH (TRS-80 M I, ...) oder Bit 1 von Port 0E8H (M III) gesetzt wird.

Beispiel:

```

PUSH  AF           ; Akku wird verändert !
CALL  CASON       ; Kassettenrecorder ein
POP   AF           ; Akku zurück

```

```
*****
*
* CASOFF : Kassettenrecorder ausschalten
*
*     E : 01F8H   (TRS-80 Model I, Genie I, II, III)
*         01F8H   (TRS-80 Model III)
*         ----- (Colour Genie)
*
*     B : nein
*
*     R : AF
*
*     I : /
*
*     O : /
*
*****
```

Funktion:

Der Kassettenrecorder wird ausgeschaltet, indem das entsprechende Bit des Ports zurückgesetzt werden.

Beispiel: s. o.

```
*****
*
* READB  : Byte von Kassette lesen
*
*     E : 0235H   (TRS-80 Model I, Genie I, II, III)
*         0235H   (TRS-80 Model III)
*         01EDH   (Colour Genie)
*
*     B : nein, (bei M III: ja, denn Abbruch möglich)
*
*     R : AF, (bei Colour: B', DE')
*
*     I : /
*
*     O : A : eingelesenes Byte
*
*****
```

Funktion:

Ein Byte von Kassette wird gelesen. Die Routine kehrt nicht eher zurück bis 8 Bits von Kassette gelesen werden konnten. Nur beim TRS-80 Model III kann durch Drücken der BREAK-Taste abgebrochen werden, wodurch ein Sprung in die Hauptschleife (1A19H) hervorgerufen wird.

Beispiel:

```
CALL    READB          ; Byte von Kassette lesen
LD      (HL),A         ; abspeichern
INC     HL
...
```



```

*****
*
* WRITEB : Byte auf Kasette schreiben
*
*     E : 0264H   (TRS-80 Model I, Genie I, II, III)
*         0264H   (TRS-80 Model III)
*         021FH   (Colour Genie)
*     B : nein
*     R : /, (Bei Colour BC', D')
*     I : A : aufzuzeichnendes Byte
*     O : /
*
*****

```

Funktion:

Das Byte im Akku wird auf Kasette aufgezeichnet.

Beispiel

```

LD      A,3CH      ; Blockheader
CALL    WRITEB     ; auf Kasette aufzeichnen

```

```

*****
*
* READS  : Header und Synchronbyte lesen
*
*     E : 0296H   (TRS-80 Model I, Genie I, II, III)
*         ----- (TRS-80 Model III)
*         024CH   (Colour Genie)
*     B : nein
*     R : A=2AH, F
*     I : /
*     O : /
*
*****

```

Funktion:

Der Header wird überlesen und auf das erste Erscheinen des Synchronbytes gewartet (s. o.).

Beispiel:

```

CALL    READS     ; Synchronbyte suchen
CALL    READB     ; Byte lesen
CP      55H       ; Systemfile ?
...

```

```
*****
*
* WRITES : Header und Synchronbyte schreiben
*
*     E : 0287H   (TRS-80 Model I, Genie I, II, III)
*         ----- (TRS-80 Model III)
*         023FH   (Colour Genie)
*
*     B : nein
*
*     R : AF, B=0
*
*     I : /
*
*     O : /
*
*****
```

Funktion:

Der Header gefolgt von dem Synchronbyte (s. o.) wird auf Kassette aufgezeichnet.

Beispiel:

```
CALL    WRITES           ; Header & Synchronbyte
LD      A,0D3H           ; 3 * D3 schreiben
CALL    WRITEB           ; für Basicprogramm
CALL    WRITEB
CALL    WRITEB
LD      A,(HL)           ; Filename schreiben
CALL    WRITEB
```

```
*****
*
* INITR  : Kass.Nr. decodieren, Kassettenrec. an, READS
*
*     E : 0293H   (TRS-80 Model I, Genie I, II, III)
*         0293H   (TRS-80 Model III)
*         ----- (Colour Genie)
*
*     B : nein, (ja wenn (HL) = "#", oder bei M III)
*
*     R : AF, BC, DE, HL=Pointer
*
*     I : HL = Zeiger auf Programmtext
*         1. (HL) <> '#'
*         2. (HL) = '#'
*
*     O : 1. unverändert
*         2. HL steht hinter Komma (#-n,), oder Abbruch
*            mit 'Syntax'-Fehler
*
*****
```

Funktion:

Weist HL bei Eintritt in INITR auf das '#' wird die folgende Kassettennummer decodiert und der entsprechende Kassettenrecorder angeschaltet. Folgt nach der Kassetten kein ',', wird mit 'Syntax'-Fehler abgebrochen, andernfalls weist HL nachher auf das erste signifikante Zeichen nach dem ',',.

Weist HL beim Eintritt in INITR nicht auf ein '#', wird der Kassettenrecorder 0 eingeschaltet und danach sofort READS aufgerufen.

Beim TRS-80 Model III wird durch INITR auch die Baudrate festgelegt. Weist HL auf einen '#', wird - neben obigen

Operationen - die Baudrate unabhängig von (BDRATE) auf 500 Baud festgelegt, andernfalls in Abhängigkeit von (BDRATE) auf 500 oder 1500 Baud festgesetzt. READB liest danach automatisch in der gewählten Baudrate, da INITR den Vektor in (READBV) einträgt.

Beispiel:

```

LD      HL,0           ; (HL) <> '#'
LD      A,0           ; Baudrate = 500 (0) oder
LD      (BDRATE),A    ; 1500 (<> 0), nur M III
CALL    INITR

```

```

*****
*
* INITW  : Kass.Nr. decodieren, Kassettenrec. an, WRITES
*
*      E : 0284H   (TRS-80 Model I, Genie I, II, III)
*           0284H   (TRS-80 Model III)
*           ----- (Colour Genie)
*      B : nein, (ja wenn (HL) = "#", oder bei M III)
*      R : AF, BC, DE, HL=Pointer
*      I : HL = Zeiger auf Programmtext
*           1. (HL) <> '#'
*           2. (HL) = '#'
*      O : 1. unverändert
*           2. HL steht hinter Komma (#-n,), oder Abbruch
*              mit 'Syntax'-Fehler
*
*****

```

Funktion:

Analog INITR.

```

*****
*
* BLINK  : Stern in rechter, oberer Bildecke umschalten
*
*      E : 022CH   (TRS-80 Model I, Genie I, II, III)
*           ----- (TRS-80 Model III)
*           01E4H   (Colour Genie)
*      B : nein
*      R : AF
*      I : /
*      O : /
*
*****

```

Funktion:

Der Stern in der rechten, oberen Bildecke wird gegen ein Space ausgetauscht, bzw. umgekehrt.

Beispiel:

```

AND     A             ; Byte = 0?
PUSH    AF           ; AF retten
CALL    Z,BLINK     ; Stern umschalten

```

POP AF ; AF zurück

```
*****
*
* READW : HL von Kassette lesen
*
* E : 0314H
* B : nein (s. READB)
* R : AF
* I : /
* O : HL = eingelesene Bytes
*
*****
```

Funktion:

Zwei Bytes werden von Kassette gelesen. Das erste (LSB) wird in L und das zweite (MSB) in H übergeben.

Beispiel:

```
CALL READB ; Byte lesen
CP 78H ; Entrypointheader ?
JR NZ,... ; nein, --> Sprung
CALL READW ; Entrypoint lesen
JP (HL) ; und anspringen
```

```
*****
*
* READBI : Bit von Kassette lesen
*
* E : 0241H (TRS-80 Model I, Genie I, II, III) 1.
* 31E8H (TRS-80 Model III, 500 Baud) 2.
* 01FAH (Colour Genie) 3.
* B : nein
* R : AF 1.
* AF, D 2.
* DE 3.
* I : A = Byte 1.
* D = Byte 2., 3.
* O : A = Byte * 2 + eingelesenes Bit 1.
* D = Byte * 2 + eingelesenes Bit 2., 3.
*
*****
```

Funktion:

Ein Bit wird von Kassette gelesen und von rechts in das Byte eingeschoben. Die 8 Bits eines Bytes sind auf der Kassette in der Reihenfolge b7, b6, ... , b0 aufgezeichnet.

Beispiel:

```
; aus READS (TRS-80 M I)
XOR A ; Byte = 0
LOOP CALL READBI ; Bit lesen
CP 0A5H ; Synchronbyte ?
JR NZ,LOOP ; nein, --> weiter
... ; ja, File lesen
```

1.8 Die Restart's

Der Z80 stellt acht 1-Byte-CALL-Befehle zur Verfügung, die für Hardwareinterrupts gedacht sind. Die RST-Befehle lauten RST 00H, 08H, 10H, 18H, 20H, 28H, 30H und 38H. Diese entsprechen einem CALL 00H, 08H, 10H, 18H, 20H, 28H und 38H, sparen aber 2 Bytes pro Befehl. Aus diesem Grund werden die meisten RST's beim TRS-80, ... für besonders häufig benutzte Unterprogramme benutzt.

```
*****
*
* RST08H : Syntaxcheck
*
*   E : 0008H Aufruf mit RST 08H (1C96H)
*   B : ja
*   R : AF, HL = Pointer
*   I : HL = Pointer auf Text
*       (PC) = Zeichen nach RST 8H Befehl
*   O : (HL) = (PC) : HL = Zeiger auf nächstes Zeichen
*       ansonsten Syntax-Fehler
*
*****
```

Funktion:

Der RST 08H ist ein Pseudo 2-Byte-Befehl. Das dem RST 08H folgende Byte wird mit dem Zeichen verglichen, auf das HL weist. Sind beide identisch, wird der RST 10H angesprungen, andernfalls mit Syntax-Fehler abgebrochen.

Beispiel:

```
LD      HL,TEXT      ; Zeiger auf Text
RST     8H           ; Syntaxcheck
DEFB    ', '         ; Komma
; A enthält jetzt '1'
...
TEXT    DEFM    ', 10 , 20 , 30'
```

```
*****
*
* RST10H : CHRGET-Routine
*
*   E : 0010H Aufruf mit RST 10H (1D78H)
*   B : nein
*   R : AF, HL = Pointer
*   I : HL = Pointer auf Text
*   O : HL + n : n >=1 und (HL + n) <> ' ', 0AH
*       A = (HL + n)
*       CY = 1, wenn Ziffer in A
*       Z = 1, wenn A = ':' oder 0
*
*****
```

Funktion:

HL wird incrementiert, bis es auf ein Zeichen ungleich ' ' und LF (= 0AH) weist. Dieses Zeichen ist bei der Rückkehr im Akku. Außerdem sind Flags gesetzt, wenn es eine Ziffer oder Anweisungstrennung (':' und 0) ist.

Beispiel:

```

RST    10H           ; nächstes Zeichen holen
JR     C,...        ; Ziffer ?, ja -->
JR     Z,...        ; Anweisungstr. ?, ja -->

```

```

*****
*
* RST18H : Vergleich HL - DE
*
*     E : 0018H Aufruf mit RST 18H (1C90H)
*     B : nein
*     R : AF
*     I : HL, DE Zahlen
*     O : HL > DE : Z = 0, CY = 0
*           HL = DE : Z = 1, CY = 0, A = 0
*           HL < DE : Z = 0, CY = 1
*
*****

```

Funktion:

HL und DE werden verglichen. Die Flags geben das Vergleichsergebnis an.

Beispiel:

```

RST    18H           ; HL = DE ?
JR     Z,...        ; ja, -->

```

```

*****
*
* RST20H : Typtest (FACX)
*
*     E : 0020H Aufruf mit RST 20H (25D9H)
*     B : nein
*     R : AF
*     I : (TYPX) = Typ von FACX
*     O : A = (TYPX) - 3
*           Integer (2) : S=1, P=1, CY=1
*           Single (4) : CY=1
*           Double (8) : P=1
*           String (3) : Z=1, CY=1, P=1
*
*****

```

Funktion:

Je nach Typ von (FACX), der (TYPX) entnommen wird, werden verschiedene Kombinationen von Flags gesetzt. Dann kann durch einen einfachen bedingten Sprung, der Typ abgefragt werden:

Beispiel:

```
RST      20H                ; Typ testen
; Die beiden ersten Abfragen sind eindeutig !
JP       M,...             ; Integer ?, -->
JP       Z,...             ; String ?, -->
; Diese Abfragen dürfen erst danach folgen !
JP       C,...             ; Single ?, -->
JP       ...               ; Double ?, -->
```

Die RST's 28H, 30H und 38H haben im ROM-Basic keine Funktion. Der RST 28H wird von INCH bei Drücken der BREAK-Taste aufgerufen. Aber erst unter Diskette bewirkt er auch eine Aktion. Außerdem dient er unter Diskette zum Aufruf der Overlay-Module des Betriebssystems.

Der RST 30H dient dem DOS-Debugger als Breakpoint.

Der RST 38H ist für den einzigen Hardwareinterrupt reserviert, der nur für die Clock und den FDC genutzt wird.

Alle RST-Einsprungadressen liegen im ROM. Dort stehen sofort Sprünge auf RAM-Vektoren, die ihrerseits Sprünge auf die oben in Klammern angegebenen Adressen beinhalten. Diese RAM-Vektoren (RST08V...RST38V, s. Teil 2), werden auch unter Diskette immer initialisiert, so daß die RST's außer RST 08H immer benutzt werden können.

Teil 2 : RAM-Adressen
=====

2.1 Ein-/Ausgabe

VIDDCB	401DH	Video-DCB
OUTFLG	409CH	Ausgabe-Flag für OUTCHF = 00H : Bildschirm = 01H : Drucker = 80H : Kassette
VIDRAM	3C00H	Bildschirmspeicher (TRS-80/Genie 64 Z.)
	4400H	Colour-Genie Text
	4800H	Colour-Genie Grafik
CURSP	40A6H	Cursorspalte
PTRDCB	4025H	Drucker-DCB
PTRPOS	409BH	Position des Druckkopfes
KEYDCB	4015H	Tastatur-DCB
IOBUFF	40A7H	I/O-Buffer
(IOBUFF)	41E8H	normaler Anfang TRS-80 M I, Genie I+II
	41E8H	" " Colour-Genie
	42E8H	" " TRS-80 Model III

2.2 Programmtext

TXTANF	40A4H	Anfang des Programmtextes
(TXTANF)	42E9H	normaler Anfang TRS-80 M I, Genie I+II
	5800H	" " Colour-Genie
	43E9H	" " TRS-80 Model III
TXTEND	40F9H	Ende des Programmtextes

2.3 Variablentabelle

VAREND	40FBH	Ende der einfachen Variablen
ARREND	40FDH	Ende der Feldvariablen
ARRFLG	40DCH	Flag zur Sperrung der Feldvariablen für SVAR = 00H : Felder freigegeben > 00H : Felder gesperrt

2.4 Arithmetik

FACX	411DH	Fließkommaakku X
FACY	4127H	Fließkommaakku Y
TYPX	40AFH	Typ von (FACX) = 2, 3, 4 oder 8

2.5 Stringverarbeitung

SSPEND	40B1H	Ende des Stringsplace
SSPANF	40A0H	Anfang des Stringsplace
SSPNT	40D6H	Stringsplacezeiger

2.7 Kassetten-I/O

BDRATE	4211H	nur TRS-80-Model-III Kassetten-Baudrate = 0 : 500 Baud > 0 : 1500 Baud
WRITBV	420CH	Vektor für WRITB zeigt auf Routine für 500/1500 Baud
READBV	420EH	Vektor für READB zeigt auf Routine für 500/1500 Baud

2.8 Restart's

RST08V	4000H	RST-08H-Vektor
(RST08V+1)	1C96H	normale Adresse der RST-08H-Routine
RST10V	4003H	RST-10H-Vektor
(RST10V+1)	1D78H	normale Adresse der RST-10H-Routine
RST18V	4006H	RST-18H-Vektor
(RST18V+1)	1C90H	normale Adresse der RST-18H-Routine
RST20V	4009H	RST-20H-Vektor
(RST20V+1)	25D9H	normale Adresse der RST-20H-Routine
RST28V	400CH	RST-28H-Vektor
RST30V	400FH	RST-30H-Vektor
RST38V	4012H	RST-38H-Vektor

Teil 3 : Vergleich der ROM's der verschiedenen Systemen:

Die Basicinterpreter in ROM oder RAM von 6 Systemen wurden mit dem Basicinterpreter des TRS-80 Model I (neue Version mit 'MEM SIZE') verglichen. Ein X in der entsprechenden Spalte zeigt einen Unterschied an.

Systeme:

1. TRS-80 Model I (alte Version mit 'MEMORY SIZE')
2. Genie I+II
3. TRS-80 Model III (deutsche und amerikanische Version)
4. Genie III
5. Genie III s
6. Colour-Genie

Das ROM des Vergleichssystem und der Systeme 1. und 2. ist im TRS-80-Model-I/Genie I+II-ROM-Listing enthalten. Unter Beachtung der geringfügigen Unterschiede, die in untenstehender Tabelle gekennzeichnet sind, ist dieses Buch auch für Genie III und IIIs geeignet.

Für das TRS-80-Model-III (3.) und das Colour-Genie (6.) existieren eigene ROM-Listings (siehe Literaturverzeichnis).

Die meisten Unterschiede zwischen den Systemen liegen im Bereich zwischen 0000H-0707H. Dort liegen die Treiberroutinen und die Systeminitialisierung, die bei den verschiedenen Systemen verschieden ausfallen. Der Bereich zwischen 0708H-2FFFH ist weitgehend identisch. Dort sind ein paar Texte verändert und in der PRINT-Anweisung einige Änderungen, die sich aus dem unterschiedlichen Bildschirmadressen und Bildaufbau ergeben. Beim Colour-Genie sind weiterhin in der PRINT, INPUT, CSAVE und CLOAD-Routine einige Änderungen der Aufrufe der Kassettenroutinen notwendig. Außerdem sind durch die Verwendung von Doppeltokens Eingriffe in die Routinen ENCTXT und DECTXT sowie in die Interpretationsschleife notwendig.

Adresse: von-bis	System:	Beschreibung
0003-0004:	X	: Sprung Systeminitialisierung
000E-000F:	X X	: Sprung DOS-Initialisierung
0047-0048:	X	: Sprung DCB-Routine
0050-0055:	X	: Tastaturdecodierungstabelle
0056-0056:	XXX	: "
0057-0058:	X	: "
0059-0059:	XXXXXX	: "
005A-0062:	X	: ", DELAY
0066-0070:	XXXX	: NMI (RESET)
0071-0071:	XXX	: "
0072-0072:	X	: "
0073-0073:	XXX	: "
0074-0081:	X	: Basicinitialisierung
0082-0082:	X X	: I/O-Bufferdefinition
0083-008A:	X	: Basicinitialisierung
008C-008C:	X	: Programtextdefinition

008E-0095: X : Basicinitialisierung
0097-0097: X : "
0099-009E: X : "
00A0-00A9: X : "
00AA-00AA: X X : Programmtextanfang
00AB-00AD: X : Basicinitialisierung
00AE-00AE: X X : Stackpointer für Initialisierung
00AF-00B1: X : Basicinitialisierung
00B2-00B4: X X : CLS-Aufruf
00B5-00BA: X : Basicinitialisierung
00BC-00C5: X : Mem Size festlegen
00C6-00C6: X X : Anfangswert für Speicherendesuche
00C7-00E7: X : Mem Size festlegen
00E9-00E9: X : " " "
00EA-00EA: X X : " " "
00EB-00F8: X : Speicherbereiche für Basic festlegen
00FA-00FC: X : Basicinitialisierung
00FD-00FD: XXXXXX : Zeiger auf Copyright Text
00FE-00FE: X : Basicinitialisierung
00FF-0101: X : "
0102-0104: X : "
0105-0105: X XXX : Texte 'Memory Size', ...
0106-0106: XXX : "
0107-0107: X XXX : "
0108-010C: XXXXXX : "
010D-010D: X X : "
010E-0116: XXXX : "
0117-0117: XXX : "
0118-011A: XXXX : "
011C-012C: XXXXXX : Tastaturentprellung
012D-0133: X : L3-Fehler, POINT-Funktion
0135-0136: X : SET-Anweisung
0138-018B: X : RESET-Anweisung
018D-019C: X : "
01D9-01D9: X : PULSE, UP für WRITEB
01DA-01DF: X : "
01E0-01E0: XXXX : Zeitkonstante
01E1-01E9: X X : PULSE
01EA-01EA: XXXX : Zeitkonstante
01EB-01EF: X X : PULSE
01F0-01F0: X : "
01F1-01F1: X X : "
01F2-01F2: X : "
01F3-01F3: X X : "
01F4-01F4: XXXX : Zeitkonstante
01F6-01F7: X : PULSE
01F8-020F: X X : CASOFF, CASNO
0210-0211: X : CASNO
0212-0214: XXXXXX : Kassettenrecorder selektieren
0215-0231: X X : CASNO, CASON, RESINP, PORTCT, BLINK
0232-0234: X : BLINK
0235-0245: X X : READB, READBI
0246-0246: X : READBI
0247-0247: X : "
0248-0248: X X : "
0249-0249: XXXXXX : Zeitkonstante
024A-024F: X X : READBI
0250-0250: XXXXXX : Zeitkonstante
0251-025E: X X : READBI
025F-0063: X : READBI, WRBTW
0264-027E: X X : WRITEB

```
027F-027F: ??XXXX : Zeitkonstante
0280-0282:   X  X : WRITEB
0283-0283:     X :   "
0284-0287:   X  X : INITW
0288-0288:   X   : WRITES
0289-02A7:   X  X : WRITES, INITR, READS
02A8-02B1:     X : SYSTEM-Anweisung (Entrypoint einlesen)
02C4-02C5:     X :   " (Sprung Warmstart)
02CF-02CF:     X :   " (Aufruf INITR)
02D2-02D3:     X :   " (Aufruf READB)
02DF-02E0:     X :   " (Aufruf READB)
02E2-02E4:   X   :   "
02E8-02E9:     X :   " (Aufruf BLINK)
02EB-02EC:     X :   " (Aufruf READB)
02F0-02F0:     X :   "
02F6-02F7:     X :   " (Aufruf READB)
02FF-0300:     X :   " (Aufruf READB)
0308-0309:     X :   " (Aufruf READB)
0310-0311:     X :   " ('C' auf Bildschirm)
0315-0316:     X : READW (Aufruf READB)
0319-031A:     X :   "   "
0335-0335:     X : OUTCHF
0337-033E:     X :   "
0346-0346:     X :   "
0348-0357:     X : CURPOS (Cursorspalte ermitteln)
03C2-03E2:     X : I/O-Treiberaufruf über DCB
03E3-03E9:   X  X : Tastaturabfrage (INCH)
03EA-03EA:     X :   "
03EB-03FA:   X  X :   "
03FB-03FC: XXXXXX :   "
03FD-03FD: XXXXXX :   "
03FE-041C:   X  X :   "
041D-041D:   X   :   "
041E-0422:   X  X :   "
0423-0423: XXXX   :   "
0424-043F:   X  X :   "
0440-044A: XXXX   :   "
044B-044B:   X  X :   "
044C-044E: XXXX   :   "
044F-0468:   X  X : Tastaturabfrage, Bildschirmausgabe
0469-046A:     X : Bildschirmausgabe (OUTCH)
046B-0472:   X  X :   "
0473-0473: XX  X :   "
0474-047C: XXXX   :   "
047D-0494:   X  X :   "
0495-0495:     X :   "
0496-049E:   X  X :   "
049F-049F:     X :   "
04A0-04AF:   X  X :   "
04B0-04B1:   X   :   "
04B2-04B7:   X  X :   "
04B8-04B8:     X :   "
04B9-04BF:   X  X :   "
04C0-04C0:   X   :   "
04C1-04C9:   X  X :   "
04CA-04CA:   X   :   "
04CB-0509:   X  X :   "
050A-050C: XXXX   :   "
050D-050D:   XXX :   "
050E-052C: XXXX   :   "
052D-052D:   X  X :   "
```

052E-0532: XXXX : "
0533-0533: XXX : "
0534-0540: XXXX : "
0541-05AC: X X : Bildschirmausgabe, Druckertr. (PRINT)
05AD-05AF: XXXXX : Ausgabebefehl (LD/OUT)
05B0-05BA: X X : Durckertreiber
05BB-05BD: XXXXX : Ausgabebefehl (LD/OUT)
05BC-05CF: X X : Druckertreiber
05D0-05D0: X : "
05D1-05D3: XXXXX : Druckerstatus lesen (LD/IN)
05D4-05D8: X : Druckertreiber
05E0-05E2: X : INBUFF
0674-0692: X X : Systeminitialisierung
0693-069D: X XX : "
069E-069E: X X : "
069F-06AC: XXXX : "
06AD-06AE: XX X : "
06AF-06C9: XXXX : "
06CA-06CA: X X : "
06CB-06CB: XXXX : "
06CC-06D1: X X : Basic-Warmstart
06D2-06EA: X : Daten für RAM (Verktoren, DCB's)
06EB-06EC: X X : " " " (Keyboard DCB)
06ED-06ED: XXXX : " " " (Keyboard DCB)
06EE-06EE: XXX : " " " (Keyboard DCB)
06EF-06EF: X : " " " (Display DCB)
06F0-06F1: X X : " " " (Display DCB)
06F2-06F2: X : " " " (Display DCB)
06F3-06F4: X X : " " " (Display DCB)
06F5-06F5: XXXX : " " " (Display DCB)
06F6-06F6: X X : " " " (Display DCB)
06F7-06F7: X : " " " (Printer DCB)
06F8-06F9: X X : " " " (Printer DCB)
06FA-06FA: XXXX : " " " (Printer DCB)
06FB-06FC: X : " " " (Printer DCB)
06FD-06FD: XXX : " " " (Printer DCB)
06FE-0707: X : " " "
124C-124D: XX XXX : FSTR
18F5-18F6: X XXX : Text für Fehlermeldung L3
1914-1915: X : Daten für RAM (Bildbreites, letzte TAB)
1916-1916: XX : " " " (n. v.)
1918-1918: X : " " " (Stringspaceanfang Initi)
191B-191B: X : " " " (Programmtextanfang)
191C-191C: X X : " " " (")
1A07-1A08: X : Ausgabe ' ERROR '
1A1F-1A21: X : Haupschleife: Kassettenrecorder aussch.
1A29-1A2A: X : " 'READY' ausgeben
1B5D-1B5F: X : NEU-Anweisung
1C15-1C17: X : ENCTXT (Erweiterung Doppeltokens)
1D67-1D69: X : Interpretationsschleife (Doppeltokens)
1DF2-1DF4: X : CONT-Anweisung (CRTC Initialisieren)
206D-206D: XX XXX : PRINT-Anweisung
2073-2073: XX XXX : "
2075-2075: XX XXX : "
2077-20B8: XX XXX : "
20BC-20BC: XX XXX : "
20F7-20F7: XX XXX : "
2131-2131: X : " (' , ')
213A-213A: XX X : " (TAB)
213B-213B: XXX : " (TAB)
213C-213C: X : " (TAB)

Teil 4 : Beschreibungen der Programme

=====

Alle folgenden Programme sind auf allen Systemen außer TRS-80-Model-III und Colour-Genie lauffähig. SEDIT erfordert auf beiden Systemen größere Änderungen, weiterhin sind PACK, NEWBAS und Z80AD nur nach kleineren Modifikationen auf dem Colour-Genie lauffähig. Im Frühjahr '85 erscheint ein Maschinensprachebuch speziell für das Colour-Genie, welches unter anderem auch die Listings von COLBAS (NEWBAS mit Listingformatierer und PACK), SEDIT, COPBIT (Universalkassettenkopierprogramm), SAVETAPE (universeller Tape/Disk-Konverter) u. v. m. enthält.

Die Sourcelisting in den Teilen 5 und 6 sind für den EDTASM (Kassettenversion, oder unter NEWDOS) und ASM (unter GDOS) ausgelegt. Die ORG-Adressen sind so festgelegt, daß die Programme im höchsten Speicher residieren. Wer über keinen dieser Assembler verfügt, kann auch mittels eines Monitor (z. B. SBUG oder CBUG) die Programme direkt in den Speicher schreiben. Wird dabei die Startadresse (ORG) geändert werden, muß bei allen Befehlen, die davon beeinflusst werden, die Adresse entsprechend geändert werden. Wer mit dem Assembler MACRO-80 arbeitet, kann mit den Konvertierungsprogrammen in Teil 6.8 Sourcen für den EDTASM in M80-Sourcen und umgekehrt umwandeln.

Auf der Sourcendiskette für DM 99,-- DM sind folgende Sourcen aufgezeichnet:

B1/SRC
B2/SRC
B3/SRC
B4/SRC
B5/SRC
SEEDIT/SRC
DOUBLE/SRC
PACK/SRC
ARRAY/SRC
ZAP/SRC
Z80AD/SRC Diskettenversion
Z80AD/BAS "
Z80AD/CMD "
CSRCMAC/BAS
CMACSRC/BAS

Die Diskette hat folgendes Format:

40 Spur, einseitig, einfache Dichte

4.1 Bildschirmeditor SEDIT

Das kurze Maschinenprogramm SEDIT ermöglicht volles Bildschirmeditieren von Basicprogrammen, ähnlich wie bei den CBM-Rechnern. SEDIT vereinigt große Effizienz mit einfacher Bedienung, da komplizierte Manipulationen mit wenigen einfachen und selbstverständlichen Grundfunktionen durchgeführt werden können.

Funktion:

SEdit wird durch Verändern des Displayvektors an den Basicinterpreter angebunden. Der Aufruf der OUTCH-Routine in der Hauptschleife zur Ausgabe des Promptzeichen kann so erkannt und unterdrückt werden. Stattdessen wird der Bildschirmeditor aufgerufen und beim Drücken von <ENTER> diejenige logische Zeile, in der sich der Cursor befindet, im I/O-Buffer an den Basicinterpreter übergeben.

Außerdem muß die OUTCH-Routine angezapft werden, um die Funktion des CR zu verändern. CR (=0DH) bewirkt normalerweise bei der Ausgabe per OUTCH das Positionieren des Cursors auf den Anfang der folgenden Zeile und das Löschen dieser Zeile. Unter SEDIT wird vor der Positionierung der Rest der Zeile, in der sich der Cursor befindet, gelöscht und das Ende der Zeile durch ein 80H (Grafikspace, unsichtbar) markiert. Die folgende Zeile, auf deren Anfang der Cursor positioniert wird, wird nicht verändert.

Durch die Markierung des Ende jeder logischen Zeile ist die Verarbeitung von beliebig langen Zeilen durch SEDIT möglich. Die Markierungen werden beim Scrollen automatisch mitgenommen und durch SEDIT beim Einfügen, Löschen, ... richtig modifiziert, wenn dies notwendig ist.

Der CBM-Bildschirmeditor arbeitet nicht mit solchen Markierung sondern mit Bitmasken, die angeben, ob aufeinanderfolgende Bildschirmzeilen zusammengehören. Da diese Bitmasken bei jedem Scroll verändert werden müssen, verbat sich dieses Vorgehen beim TRS-80, wenn eine völlige Neuprogrammierung der OUTCH-Routine vermieden werden soll. Außerdem ermöglicht das Konzept mit den Markierungen interessante Möglichkeiten zum Editieren, wie z. B. die Verknüpfung zweier Zeilen.

Das Sourcelisting in Teil 6.1 zeigt die Kassettenversion. Diese Version läuft mit gewissen Einschränkungen auch auf Diskettensystemen. Eine ideal auf alle Betriebssysteme abgestimmte Version, kann bei uns bezogen werden.

So arbeiten Sie mit SEDIT:

Sobald SEDIT geladen und initialisiert ist, erscheint der neue Cursor: ein blinkendes Quadrat. Dieser erscheint fortan immer, wenn im normalen Betrieb das Promptzeichen erscheinen würde, also nicht unter AUTO, SYSTEM, EDIT oder während der Programmausführung.

Sobald der blinkende Cursor zu sehen ist, eröffnet sich Ihnen eine Vielfalt neuer Möglichkeiten das Programm zu verändern. Da der Cursor mit den vier Pfeiltasten zerstörungsfrei über den gesamten Bildschirm bewegt werden kann, wird auf der Cursorposition abwechselnd das Cursorzeichen (weißes Quadrat) und das Zeichen, welches sich auf dieser Position befand, angezeigt.

Auf der Cursorposition kann der Bildschirminhalt nun einfach überschrieben werden, aber alle Änderungen werden erst in den Speicher übernommen, wenn die ENTER-Taste gedrückt wird (s.u.).

Mit den vier Pfeiltasten kann der Cursor bewegt werden. Stößt er bei Vor- oder Rückwärtsbewegung auf das Zeilenende oder den -anfang, geht er automatisch in die vorhergehende oder nachfolgende Zeile. Bei der Auf- und Abwärtsbewegung bleibt er am Bildschirmrand stehen.

(Da <↑> zur Cursorbewegung dient, ergibt <SHIFT><↑> fortan den Exponentiationspfeil.)

Bei den im folgenden dargestellten Funktionen werden von SEDIT keine physikalischen Zeilen (d.h. die 64 Zeichen einer Bildschirmzeile), sondern logische Zeilen (d.h. Zeilen, die zwar länger als 64 Zeichen sein können, aber sinngemäß zusammengehören.) verarbeitet.

Erläuterungen zu den folgenden Beispielen:

_ : Cursorposition

\$: Folgende Taste festhalten, bis nächste Cursorposition erreicht

<SHIFT><←> (DELETE):

Das Zeichen auf der Cursorposition wird gelöscht; der Rest der logischen Zeile rückt nach.

Beispiel:

```
10 FOR I=0 TO 10:PRINT I:NEXT      <SHIFT><←>
10 FOR I=0 TO 10:RINT I:NEXT      <SHIFT><←>
10 FOR I=0 TO 10:INT I:NEXT       <SHIFT><←>
10 FOR I=0 TO 10:NT I:NEXT       <SHIFT><←>
10 FOR I=0 TO 10:T I:NEXT        <SHIFT><←>
10 FOR I=0 TO 10:I:NEXT         <SHIFT><←>
10 FOR I=0 TO 10: I:NEXT        <SHIFT><←>
10 FOR I=0 TO 10::NEXT         <SHIFT><←>
10 FOR I=0 TO 10:NEXT          <SHIFT><←>
```

(Anstatt 8 mal die <SHIFT> und <←>-Taste zu drücken, können Sie diese Tasten auch einfach festhalten, bis der 2. Doppelpunkt unter dem Cursor erscheint (s. u. Repeatfunktion).

<SHIFT><→> (INSERT):

Nach einmaligem Drücken von <SHIFT><→> befindet sich SEDIT im Insert-Mode, der erst bei Drücken eines nichtdruckenden Zeichen (z.B. Cursorbewegung oder ENTER) wieder verlassen wird. Im Insert-Mode wird jedes druckende Zeichen an der Cursorposition eingefügt und der Rest der logischen Zeile wird weitergeschoben. Auch bei überschreiten des Zeilenendes wird der Bildschirminhalt richtig korrigiert.

Beispiel:

```
10 A=MID$(A$,2)                   <SHIFT><→>
10 A=MID$(A$,2)                   <,>
10 A=MID$(A$,2,_)                 <1>
10 A=MID$(A$,2,1)                 <ENTER>
```

(Durch das einmalige <SHIFT><→> wird SEDIT in den Insert-Modus versetzt. Alle druckenden (d.h. direkt auf dem Bildschirm darstellbaren Zeichen) werden jetzt an der Cursorposition eingefügt, und der Rest der Zeile wird verschoben. Durch Eingabe eines nichtdruckenden Zeichen (wie z.B.: <ENTER>) wird dieser Modus verlassen und im Beispiel die Zeile in Ihrer neuen Form abgespeichert.

<SHIFT><@> (SKIP):

Vom Cursor an wird der Rest der logischen Zeile gelöscht.

Beispiel:

```
10 ? I,I*I_I*I*I           <SHIFT><@>
10 ? I,I*I_
```

<BREAK>:

Vom Bildschirmende wird die erste freie Zeile zur bequemen Befehlseingabe gesucht, und der Cursor dort positioniert.

<CLEAR>:

Wie gehabt.

<ENTER>:

Der Inhalt derjenigen logischen Zeile, in der sich der Cursor befindet, wird über den Tastaturtreiber an den Basicinterpreter übergeben (Ist die Zeile länger als 240 Zeichen, wird ein LS-Error erkannt.). Der Cursor befindet sich danach in der nächsten logischen Zeile.

Tastenwiederholung

Alle Tasten unterliegen einem Autorepeat, der nach ca. 0,5 Sekunden mit ca 15 Hz einsetzt. Während der Wiederholung kann die SHIFT-Taste losgelassen werden. Wird eine neue Taste gedrückt bevor die alte losgelassen wurde, setzt die Wiederholung der neuen Taste ein, die allerdings aufhört, wenn die erste Taste losgelassen wird.

Besondere Tricks

Mit SEDIT können Programmmodifikationen, die sonst kompliziert und langwierig sind, einfach und komfortabel erledigt werden. Hier sind einige wichtige Tricks zusammengetragen:

1) Verschieben von Programmzeilen

Mit SEDIT können einfach Zeilennummern geändert und somit Programmzeilen verschoben oder dupliziert werden.

Beispiel:

```

10 ? I".ELEMENT"A(I)           20<ENTER>
20 ? I".ELEMENT"A(I)           LIST<ENTER>
...

```

2) Editieren nach Syntax-Fehlern

Bei Syntax-Fehlern springt der TRS-80 automatisch in den Zeileneditor. Unter SEDIT verläßt man diesen am besten sofort mit ENTER (was einen Ausdruck der Zeile bewirkt) und editiert sie sodann mit SEDIT.

3) Verknüpfen von Programmzeilen

Zwei Programmzeilen können einfach zu einer einzigen verbunden werden:

3.1) Überschreiben des letzten Zeichens der 1.logischen Zeile (letzte Bildschirmspalte). Der Cursor befindet sich sodann in der 2.Zeile und beide bilden nun eine logische Zeile. Mit Delete werden nun nur noch die überflüssigen Leerzeichen und die 2.Zeilenummer entfernt.

3.2) Löschen des letzten Zeichen (s.o.) mit Delete. Die 2.Zeile rückt dann nach. Weiter wie bei 3.1).

Beispiel:

```

10 FOR I=0 TO 10
20 NEXT
10 FOR I=0 TO 10
0 NEXT
10 FOR I=0 TO 10_
0 NEXT
10 FOR I=0 TO 10_20 NEXT
10 FOR I=0 TO 10_0 NEXT
10 FOR I=0 TO 10_NEXT
10 FOR I=0 TO 10_NEXT

```

Bildschirmrand

```

-
2
20

```

```

<SHIFT><<>
$<<>
$<SHIFT><<>
<SHIFT><<>
<SHIFT><<>
<.:>
<ENTER>

```

4) Übernahme von Programmausdrucken

Mit SEDIT können auch Ausdrücke von Programmen einfach in den Speicher übernommen werden: z.B. können Maschinenprogramme von einem kleinen Programm mit Zeilennummer und DATA-Anweisung auf den Bildschirm gedruckt werden und dann mit SEDIT in den Speicher übernommen werden.

5) Wiederholung von ähnlichen Direktanweisungen

Direktanweisungen können ohne aufwendige Tipperei bequem mit verschiedenen Parametern durchlaufen werden.

Zusammenfassung

<→>	Cursor 1 Zeichen nach rechts
<←>	Cursor 1 Zeichen nach links
<↑>	Cursor 1 Zeile hoch
<↓>	Cursor 1 Zeile runter
<SHIFT><←>	DELETE: Zeichen auf Cursorposition löschen Rest der Zeile nach links rücken.
<SHIFT><→>	INSERT: Insert-Modus setzen. Folgende Zeichen an Cursorposition einfügen
<SHIFT><Ⓞ>	SKIP: Rest der Zeile löschen
<BREAK>	CMD: Cursor in erste freie Zeile vom Bildschirmende her positionieren
<CLEAR>	CLEAR: Bildschirm löschen
<ENTER>	RETURN: Zeile an Basicinterpreter übergeben.

4.2 Basicerweiterung NEWBAS

NEWBAS bieten dem Cassettenbenutzer viele der Möglichkeiten, die sonst erst dem Diskettenbenutzer offenstehen.

Es ist folgendes unter NEWBAS möglich:

- Umnummerierung von Basicprogrammen
- Flexibles Zusammenladen verschiedener Basicprogramme von Cassette
- Suchen von Zeichenfolgen im Basicprogrammtext
- Hexadezimale Konstanten
- USR0 bis USR9
- DEFUSR0 bis DEFUSR9
- DEFFN
- LINE INPUT
- MID\$ als Anweisung
- INSTR
- ROUTING
- RESTORE n

Funktion:

Die meisten Erweiterungen von NEWBAS werden über die unbenutzten Diskbasicbefehlswoorte aufgerufen. Diese Anbindung ist ähnlich wie beim Programm PACK beschrieben programmiert, aber hier noch einfacher. Beim Programm PACK mußte in der Diskettenversion der Befehl KILL zum Löschen von Files erhalten bleiben. Daher wird nach Auftreten des KILL zunächst das Argument geprüft. Wenn es eine Zeichenkette ist, wird ein File gelöscht, andernfalls der Programmtext gepackt.

Der RESTORE N-Befehl wird durch Veränderung des RST 10H-Vektors implementiert. Dazu wird der RST 10H-Aufruf in der Interpretationsschleife, kurz vor dem Anspringen der einzelnen Routinen für die verschiedenen Anweisungen abgefangen.

NAME (UMNUMERIEREN VON BASICPROGRAMMEN)

Mit NAME können Sie die Zeilennummern Ihrer Basicprogramme verändern. Dies kann in folgenden Fällen notwendig werden:

- 1) Sie möchten nachträglich zwischen zwei Zeilen - z.B. mit Zehner-Abstand - einen Block von mehr als neun Basiczeilen einschieben.
- 2) Sie möchten zwei Basicprogramme zusammenladen (Unter NEWBAS ist das möglich) und müssen dafür sorgen, daß das zweite Programm bezüglich seiner Zeilennummern über dem Ersten zu liegen kommt.
- 3) Sie möchten der Übersichtlichkeit oder einer Standardisierung halber gewisse Programmsegmente (z.B. Unterprogramme) ab einer bestimmten Zeilennummer (meist Tausender-Nummern) beginnen lassen.

Der Befehl NAME hat folgendes Format: NAME n1,n2,inc

Mit: n1 = Erste neue Zeilennummer
 n2 = Alte Zeilennummer, ab welcher die Umnummerierung
 beginnen soll. (Die alte Zeile n2 hat also nach
 der Umnummerierung die Nummer n1)
 inc = Abstand zwischen den neuen Zeilen

Z.B. NAME 1000,200,5 bedeutet, daß alle Zeilen, die vorher ab 200 (einschl.) standen, nun ab Zeile 1000 (einschl.) zu liegen kommen. Der Abstand zwischen den neuen Zeilen ist 5.

Sie können auch ein oder mehrere Argumente (n1,n2,inc) weglassen. Der Rechner nimmt dann folgende Standard Argumente an:

 n1 = 10
 n2 = Erste vorhandene alte Zeilennummer (also gesamten
 Basicprogrammtext umnummerieren)
 inc= 10

Z.B. NAME 100 bedeutet, daß alle Zeilen so umnummeriert werden sollen, daß 100 die erste neue Zeilennummer ist und der Abstand zwischen den neuen Zeilen 10 ist.

NAME 5,,2 bedeutet, daß alle Zeilen so umnummeriert werden sollen, daß 5 die erste neue Zeilennummer ist und der Abstand zwischen den neuen Zeilen 2 ist.

Aber auch 'NAME 500,200' oder nur 'NAME' sind mögliche Eingaben.

MERGE und RSET (ZUSAMMENLADEN MEHRERER BASICPROGRAMME)

Mit MERGE und RSET wird es möglich mehrere schon auf Cassette vorhandene Basicprogramme zusammen in den Rechner zu bringen. Dies erweist sich als besonders wertvoll, wenn Sie in neu zu schreibende Programme bereits vorhandene Programmsegmente (insbesondere Unterprogramme aus einer schon vorhandenen Unterprogrammammlung) einfügen möchten.

Durch Eingabe von 'MERGE' wird das momentan im Rechner befindliche Basicprogramm "unsichtbar" gemacht. Der Speicher scheint leer zu sein, wovon Sie sich durch Eingabe von 'LIST' überzeugen können. Sie können nun das zweite Programm wie gewohnt mittels 'CLOAD' laden. Danach steht Ihnen dieses zweite Programm so im Rechner zur Verfügung, wie Sie es nach normalen Ladevorgängen gewohnt sind. Das heißt, Sie können das Programm laufen lassen, listen

Um nun das "unsichtbare" Programm zusammen mit dem zugeladen zweiten Programm zum Vorschein zu bringen, müssen Sie dafür sorgen, daß die niedrigste Zeilennummer des zweiten Programmes größer ist als die höchste Zeilennummer des ersten (momentan "unsichtbaren") Programmes. Falls dies nicht ohnehin der Fall ist, kann mit Hilfe des 'NAME' Befehls leicht dafür gesorgt werden, daß das zweite Programm die erforderlichen höheren Zeilennummern erhält. (Bei dieser Umnummerierung ist es auch möglich, ein zugeladenes Unterprogramm nachträglich an das aufrufende Programm anzupassen.)

Durch Eingabe von 'RSET' wird nun das erste Programm neben dem Zweiten wieder "sichtbar" gemacht. Es stehen nun beide Programme zusammen im Speicher.

Nach diesem Verfahren können Sie nun noch beliebig viele weitere Programme zuladen.

Solange sich ein "unsichtbares" Programm im Rechner befindet erscheint zusätzlich zu jeder 'NEWBAS READY' Meldung der Text '* MERGE MODE *' auf dem Bildschirm. Dies soll Sie an das noch vorhandene, "unsichtbare" Programm erinnern.

GET (SUCHEN VON ZEICHENFOLGEN IM BASICPROGRAMMTEXT)

Mit GET können beliebige Zeichenfolgen im Basicprogrammtext gesucht werden. Dies ist nützlich wenn Sie z.B. feststellen wollen ob, bzw. wo bestimmte Variablennamen im Basicprogrammtext benutzt werden. Oder Sie möchten in einem längeren Basicprogrammtext eine bestimmte Stelle, deren Zeilennummer Sie nicht kennen, schnell finden.

Sie geben 'GET' ein, und der Rechner meldet sich mit 'STRING='. Nun können Sie die Zeichenfolge, nach welcher gesucht werden soll, eingeben. Hierbei müssen zwei Fälle unterschieden werden:

- 1) Sie suchen eine Variable (z.B. AS\$, N, N%) oder einen Ausdruck mit Basicschlüsselwörtern (z.B. GOTO 200, RETURN). In diesem Fall geben Sie die gesuchte Zeichenfolge ein, und schließen mit 'ENTER' ab.
- 2) Sie suchen Zeichenfolgen, die sich innerhalb von Anführungsstrichen ("), in DATA-Anweisungen oder in Bemerkungen (hinter REM bzw. ') befinden. Hier geben Sie vor der eigentlichen Zeichenfolge '@@' ein und schließen dann mit 'ENTER' ab.
(Wenn Sie also z.B. die Zeichenfolge "COPYRIGHT BY" suchen, geben sie '@@COPYRIGHT BY' ein.)

In beiden Fällen wird der Rechner darauf hin die Zeilennummern, in welchen die gesuchte Zeichenfolge auftritt, ausgeben. Sollte die Zeichenfolge nicht im Basicprogrammtext vorhanden sein, so wird dies anstelle der Zeilennummern gemeldet.

Bemerkung: Tritt die gesuchte Zeichenfolge in einer Zeile mehrmals auf, so wird auch dieselbe Zeilennummer entsprechend oft ausgegeben. Dabei ist zu beachten, daß z.B. in der Zeichenfolge "****" die Zeichenfolge "*" dreimal vorkommt.

Bei 1) wird aus dem Suchtext vor dem Vergleich mit dem Basicprogramm der Zwischencode gebildet.
Bei 2) wird der Suchtext nicht verändert.

& (HEXADEZIMALE KONSTANTEN)

Beim Arbeiten mit "maschinennahen" Befehlen (PEEK,POKE,IN,OUT usw.) ist es oft angenehmer mit hexadezimalen, als mit dezimalen Größen zu arbeiten.

Um dies zu ermöglichen erlaubt NEWBAS die Eingabe von hexadezimalen Konstanten. Eine Hexzahl kennzeichnen Sie einfach durch ein vorgestelltes '&'.
'&'

So bedeutet z.B.

POKE &3C00, &3F

das gleiche wie

POKE 15360,63

Bemerkung: Hexadezimale Konstanten dürfen nicht in INPUT- oder DATA-Anweisungen benutzt werden!

Statt &3F kann man auch &H3F schreiben.
Letztere Schreibweise ist kompatibel zum Diskbasic.

USR0 bis USR9 (AUFRUF VON 10 VERSCHIEDENEN MASCHINENPROGRAMMEN)

War es bislang für Sie nur möglich ein Maschinenprogramm aus einem Basicprogramm heraus aufzurufen, so können unter NEWBAS bis zu 10 verschiedene Maschinenprogramme aufgerufen werden.

Die Arbeitsweise der USR0 bis USR9 Funktionen ist genauso, wie in Ihrem Basichandbuch unter USR erläutert.

So rufen Sie z.B. mit

A=USR7(0)

das 7. Maschinenprogramm auf.

DEFUSR0 bis DEFUSR9 (STARTADRESSEN VON MASCHINENPROGRAMMEN)

Bislang mußte die Startadresse des Maschinenprogramms, welches mittels USR aufgerufen werden sollte, in zwei spezielle Speicherzellen "gePOKEt" werden.

Diese unpraktische Vorgehensweise wird unter NEWBAS durch die Befehle DEFUSR0 bis DEFUSR9 ersetzt.

Sie geben z.B.

DEFUSR7=15360 (oder hexadezimal DEFUSR7=&3C00)

ein und weisen damit dem 7. Maschinenprogramm die Startadresse 15360 (hex 3C00) zu.

DEFFN (VOM ANWENDER DEFINIERBARE FUNKTIONEN)

Mit DEFFN können Sie Funktionen, welche häufig in einem Programm benutzt werden, einmal definieren und dann im Laufe des Programms beliebig oft durch den Funktionsnamen aufrufen.

DEFFN hat folgendes Format:

DEFFN FKTN (VARLIST) = AUSDRUCK

FKTN ist der Name der Funktion, die hier definiert wird. Für diesen Namen gelten die gleichen Kriterien wie für Variablennamen. (Also erstes Zeichen alphabetisch, alle weiteren alphanumerisch, nur die ersten beiden Zeichen werden vom Basicinterpreter berücksichtigt)

Weiterhin muß im Funktionsnamen gekennzeichnet werden, von welchem Typ das Ergebniss der Funktion sein soll. Die erfolgt wiederum nach den gleichen Kriterien, wie sie schon von den Variablennamen her bekannt sind. (Also % für ganzzahlige, ! für einfachgenaue, # für doppeltgenaue und \$ für String-Ergebnisse. Weglassen ergibt einfachgenaue Ergebnisse.)

VARLIST besteht aus sovielen Variablen (jeweils durch Komma getrennt), wie der Ausdruck auf der rechten Seite an Argumenten benötigt.

AUSDRUCK ist die Funktion selbst. Sie darf nur eine Anweisung von einer zeile Länge sein. (Durch Doppelpunkt getrennte Anweisungen sind nicht erlaubt.)

Zum besseren Verständnis folgende Beispiele:

```
10 DEFFN CUBIC (X) = X*X*X
20 INPUT A
30 B = FN CUBIC (A)
40 PRINT B
```

In 10 wird die einfache Funktion CUBIC definiert. Sie liefert ein Ergebniss einfacher Genauigkeit und benötigt nur eine Variable in der Variablenliste, da sie nur ein Argument (X) im Ausdruck hat.

In 30 wird die Funktion CUBIC aufgerufen. Hier ist das Argument die Variable A. (Es könnte dort aber auch jeder beliebige Ausdruck einfacher Genauigkeit stehen.)

Oder:

```
10 DEFFN TE$ (X$,Y$) = X$+"/"+Y$
20 INPUT "VORWAHL";V$
30 INPUT "RUFNUMMER";R$
40 G$ = FN TE$ (V$,R$)
50 PRINT G$
```

Hier wird eine Funktion mit zwei Stringargumenten definiert, welche in diesem Fall auch wieder ein Stringergebniss liefert. (Was nicht immer der Fall sein muß!)

LINE INPUT (ZEILENEINGABE)

Mit dem normalen INPUT Befehl ist es nicht möglich in eine Stringvariablen Kommata oder Anführungsstriche einzugeben. Außerdem werden eingegebene, führende Leerzeichen nicht in die Stringvariable übernommen. Weiterhin wurde jede Eingabe mittels Ausgabe eines '?' aufgefordert. Dadurch verbot sich die Anwendung des INPUT Befehls in vielen Fällen. (Insbesondere Textverarbeitungsprogramme in Basic!)

LINE INPUT umgeht diese Probleme und hat folgendes Format:

```
LINE INPUT "TEXT"; VAR$
```

mit: TEXT ist ein beliebiger Text, welcher vor der Eingabe auf den Bildschirm geschrieben wird. (Kann einschließ-lich nachfolgendem Semikolon entfallen!)

VAR\$ ist die Stringvariable, welche die Eingabe über-nehmen soll.

Kurzes Beispiel:

```
10 LINE INPUT "IHRE EINGABE BITTE!"; A$
20 PRINT A$
```

Hier erscheint nach der Ausgabe des Textes nur der Cursor in der nächsten Zeile. (kein Fragezeichen mehr!) Sie können nun beliebige Zeichen eingeben (incl. führender Leerzeichen, Kommata, Anführungsstriche usw.) bis Sie mittels <ENTER> die Eingabe abschließen.

MID\$ (ALS STATEMENT)

MID\$ als Statement ist eine Art Umkehrung der MID\$ Funktion. Mit dem MID\$ Statement wird ein Teil eines Strings durch einen anderen String ersetzt.

Das Format ist wie folgt:

```
MID$(VAR$,n1,n2)=AUSDRUCK$
```

mit: VAR\$ ist die zu verändernde Stringvariable.

n1 ist die Position, ab welcher die Veränderung in VAR\$ beinhalten soll. (Erste Position im String ist 1!)

n2 gibt die Anzahl der zu verändernden Zeichen an. Kann auch entfallen. (einschl. vorstehendem Komma!) Dann wird AUSDRUCK\$ soweit eingesetzt, daß VAR\$ ihre ursprüngliche Länge beibehält.

AUSDRUCK\$ ist ein beliebiger Stringausdruck.

Beachten Sie, daß die Ersetzung immer nur soweit erfolgt, daß die Länge des zu verändernden String erhalten bleibt!

Für folgende Beispiele wird vorausgesetzt: A\$="ABCDEFGH"

MID\$(A\$,2,2)="12345"	liefert:	A\$="A12DEFGH"
MID\$(A\$,5) ="12345"	"	A\$="ABCD123H"
MID\$(A\$,4,4)=""	"	A\$="ABCDEFGH"

INSTR (STRINGSUCH FUNKTION)

Mit INSTR können Sie Teilstrings innerhalb eines String suchen.

INSTR hat folgendes Format:

INSTR(n,AUSDRUCK1\$,AUSDRUCK2\$)

mit: n ist die Position in AUSDRUCK1\$, ab welcher gesucht werden soll. Kann entfallen. (einschl. nachfolgendem Komma) In diesem Fall wird ab der 1. Position gesucht.

AUSDRUCK1\$ ist der Stringausdruck, in welchem gesucht werden soll.

AUSDRUCK2\$ ist der Stringausdruck, nach welchem gesucht werden soll.

Der Aufruf der INSTR Funktion liefert als Ergebniss die Position, ab welcher der gesuchte String zuerst im zu durchsuchenden String auftritt. Wird der Teilstring nicht gefunden, so ergibt sich 0 als Ergebniss.

Bei den folgenden Beispielen wird wieder A\$="ABCDEFGH" vorausgesetzt:

INSTR(A\$,"EF")	liefert	5
INSTR(A\$,"234")	"	0
INSTR(3,A\$,"AB")	"	0

RESTORE N

Mit RESTORE N ist direkte Zugriff auf jede beliebige DATA-Zeile möglich. N muß nicht wie bei GOTO, GOSUB, usw. eine Ganzzahlkonstante sein, sondern kann ein beliebiger Ausdruck sein. Die RESTORE-Anweisung ohne Zeilennummer ist natürlich auch weiterhin verfügbar.

Beispiel:

```
10 DATA 10
20 DATA 20
30 DATA 30
40 RESTORE: READ A: PRINT A;
50 RESTORE 30: READ A: PRINT A;
60 RESTORE 3*10: READ A: PRINT A;
```

ergibt beim Starten mit RUN:

```
10 30 30
```

ROUTING

Routing ermöglicht Ihnen das Umleiten von Ausgaben. Sie können z.B. Ausgaben, welche eigentlich auf den Bildschirm gehen sollen, auf den Drucker umleiten.

Der Befehl hat folgendes Format:

```
CMD quelle TO ziel
```

mit: quelle ist, PRINT oder LPRINT

ziel ist V (Video), P (Printer) oder B (Beide)

Der Befehl kann sowohl im Direktmodus als auch im Programm verwendet werden. Sie können also den Routingmodus während des Programmablaufes beliebig ändern. Beachten Sie aber bitte, daß im Direktmodus kein Routing durchgeführt wird. Es bleibt aber der alte Routingmodus erhalten, wenn Sie wieder aus dem Direktmodus zum Programmlauf übergehen.

Beispiele:

```
CMD PRINT TO B
```

Diese Eingabe im Direktmodus bewirkt, daß alle Ausgaben mit dem PRINT Befehl während der nächsten Programmläufe nicht nur auf den Bildschirm, sondern auch auf den Drucker gehen.

```
100 CMD LPRINT TO V : CMD PRINT TO P
```

Ab dieser Zeile werden alle Drucker Ausgaben auf den Bildschirm, und alle Bildschirmausgaben auf den Drucker umgeleitet.

Achtung!

Die Formatierung von Drucker- und Bildschirmausgabe kann unter Umständen differieren. (Unterschiedliche Zeichenzahl pro Zeile der beiden Ausgabeelemente, auf dem Drucker kann kein PRINT @ simuliert werden, usw.)

4.3 Double-Precision-Functions DOUBLE

DOUBLE erweitert die Funktionen (SQR, LOG, EXP, SIN, COS, TAN, ATN) auf 16-stellige Genauigkeit. Durch die hohe Geschwindigkeit und die gleiche Handhabung wie die Funktionen einfacher Genauigkeit bedeutet DOUBLE eine erhebliche Erleichterung bei der Programmierung mathematischer, wissenschaftlicher und kaufmännischer Anwendungen.

Funktionweise:

Die Algorithmen zur Berechnung der transzendenten Funktionen können nicht Gegenstand dieses Buches sein (bitte nehmen Sie hierzu (4) aus dem Literaturverzeichnis zur Hand), sondern nur die Methoden der Anbindung an den Basicinterpreter.

Da eine Erweiterung der Genauigkeit der transzendenten Funktionen durch den Hersteller nicht geplant war, ist kein spezieller Aussprung ins RAM vorhanden, an den eine derartige Erweiterung angehängt werden könnte. In solchen Fällen kann einer der RST-Befehle benutzt werden, die praktisch in allen Programmsegmenten des Basicinterpreters vorkommen. Dieser Eingriff über die RST-Befehle ist völlig unabhängig von deren Funktion, und dient nur dazu, an einer bestimmten Stelle im ROM eine "Umleitung" einzubauen. Dazu wird der RST-Vektor im RAM geändert und dort die Adresse einer eigenen Routine eingesetzt. Beim Aufruf des RST wird dann zunächst diese Routine aufgerufen, die aus dem Stack die Rücksprungadresse entnimmt. Anhand der Rücksprungadresse kann festgestellt werden, woher der RST aufgerufen wurde. Wurde er von derjenigen Stelle aufgerufen, in die eine Umleitung eingebaut werden soll, wird das neue Programmsegment angesprungen. Andernfalls wird die RST-Routine, deren Adresse vor Verbiegen des RST-Vektors notiert wurde, angesprungen. Vor Ansprung der RST-Routine müssen alle Register wiederhergestellt werden, außer dem Akku und den Flags (AF). Diese Register werden durch alle RST-Routinen verändert, und keine erwartet dort einen Parameter.

Beispiel:

```
RST10V EQU      4004H

LOWADR EQU      0A0H      ; Adresse des RST's + 1
HIGHADR EQU     24H      ; = 24A0H

INIT   LD      HL,(RST10V) ; Adresse RST 10H-Routine
       LD      (RETRST+2),HL ; für Rücksprung retten
       LD      HL,RST10N   ; Adresse der neuen Routine
       LD      (RST10V),HL ; im RST 10H-Vektor ablegen
       RET

RST10N EX      (SP),HL    ; Rücksprungadresse nach HL
```



```
LD      A,L      ; HL retten !
CP      LOWADR   ; LSB RET-Adresse nach A
JR      NZ,RETRST ; = ges. Adresse ?
LD      A,H      ; MSB RET-Adresse nach A
CP      HIGHADR  ; = ges. Adresse ?
RETRST: EX      (SP),HL ; HL zurück und
                ; Rücksprungadr. auf Stack
JP      NZ,0     ; Sprung in RST-Routine
                ; hier folgt die Umleitung
...

```

Für DOUBLE müssen gleich 2 RST's umgebogen werden. Der RST 10H für die Funktionen. Dazu wird der RST 10H zu Beginn der Routine umgeleitet, die die Operanden für die Ausdrucksauswertung (EXP) bereitstellt. Der zweite RST ist der RST 20H. Er muß zur Erweiterung der Exponentiation herangezogen werden. Beim Auftreten dieses Operators in einem Ausdruck wird der Operand vor ihm in Single umgewandelt. Dazu wird die Routine CSNG aufgerufen. Da diese Routine ihrerseits den RST 20H benötigt, kann dort eine Umleitung eingebaut werden. Beim Aufruf des veränderten RST 20H muß nun zunächst geprüft werden, ob er von der Routine CSNG aufgerufen und dann ob diese Routine von einer bestimmten Stelle der Ausdrucksauswertung angesprungen wurde. Treffen beide Bedingungen zu, wird die Erweiterung angesprungen.

Handhabung

Sobald DOUBLE geladen und initialisiert ist, stehen die obengenannten Funktionen auch in 16-stelliger Genauigkeit zur Verfügung. Das bedeutet, wenn als Argument ein Wert in doppelter Genauigkeit auftritt, wird der Funktionswert in doppelter Genauigkeit berechnet. Wenn Sie nicht mehr ganz sicher sind, wann eine Konstante als Konstante doppelter Genauigkeit angesehen wird und welche Probleme bei der Umwandlung zwischen den drei numerischen Typen entstehen, schlagen Sie bitte in den entsprechenden Kapiteln im Level-2-Handbuch nach, um nicht durch Programmierfehler die neugewonnene Genauigkeit wieder zunichte zu machen. Auch der ↑ -Operator liefert 16-stellige Resultate, wenn mindestens einer der beiden Operanden eine doppelgenaue Zahl ist.

Genauigkeit

Ungenauigkeiten treten normalerweise nur in der letzten Stelle auf. Aufgrund der internen Darstellung der Funktionen können bei bestimmten Funktionen und bestimmten Argumenten schonmal grössere Ungenauigkeiten auftreten, welche unvermeidbar sind. Da die gleichen Ungenauigkeiten auch bei den Funktionen einfacher Genauigkeit auftreten, dürften Sie dem erfahrenen Programmierer bekannt sein. Einige wichtige Fälle seien trotzdem kurz angegeben: bei

den trigonometrischen Funktionen sinkt die Genauigkeit bei Erhöhung des Argumentes (Periodizität) und beim Cosinus auch bei Argumenten sehr nahe bei Null (wegen: $\text{COS}(X) = \text{SIN}(X + \text{PI}/2)$). Weiterhin sinkt die Genauigkeit der EXP-Funktion bei wachsendem Argument ab.

Geschwindigkeit

Normalerweise werden zur Berechnung eines Funktionswertes die folgenden (ziemlich konstanten) Zeiten bei 1,7 MHz Taktfrequenz benötigt benötigt:

SQR	150 ms	(0)
LOG	360 ms	(1)
EXP	420 ms	(0)
SIN	330 ms	(0)
COS	330 ms	(- PI/2)
TAN	700 ms	(0)
ATN	270-750 ms	(0)

(Bei den in Klammern genannten Argumenten reduzieren sich die Ausführungzeiten drastisch auf einige Millisekunden.)

4.4 Basicprogrammpacker PACK

PACK entfernt aus Basicprogrammen überflüssige Leerzeichen und Kommentare ohne an der Funktion des Programmes etwas zu ändern.

Funktionweise:

Die Anbindung der PACK-Routine an den Basicinterpreter ist in der einfachsten Möglichkeit realisiert. Ein Befehlsword, bei dessen Auftreten immer auf einen RAM-Vektor verzweigt wird, wird dazu benutzt. Der Sprungvektor im RAM kann leicht geändert und dadurch der neue Befehl implementiert werden. Von der Interpreterschleife wird beim Ansprung in HL der Zeiger auf das 1. signifikante Zeichen nach dem Befehls-Token übergeben. Beim Rücksprung in die Interpreterschleife muß HL auf einen Doppelpunkt oder eine Null (=0H) als Anweisungstrennung weisen. Andernfalls erkennt die Interpreterschleife einen Syntax-Fehler. Diese einfache Möglichkeit neue Befehle zu implementieren ist in vorliegenden Beispiel nicht in ihrer reinen Form enthalten. Da das Befehlsword KILL im Diskbasic eine Bedeutung hat, die auch nach Laden von PACK erhalten bleiben soll, wird in Abhängigkeit vom folgenden Parameter entweder ein File gelöscht oder der Basicprogrammtext gepackt.

Der Aufruf:

Wenn PACK geladen ist steht ein neuer Befehl zur Verfügung:

KILL < ZEILE < - ZEILE >>

Durch diesen Befehl werden im ganzen Programmtext (- oder bei Angabe der optionalen Parameter, im spezifizierten Bereich -) alle Leerzeichen und Kommentare entfernt.

Beim Packen des Programmes werden alle Vorsichtsmaßnahmen getroffen, um die Funktion des Programmes nicht zu beeinträchtigen:

1) Reine Kommentarzeilen werden nur dann vollständig entfernt, wenn sie nicht ein Sprungziel bilden. Andernfalls bleiben Zeilennummer und REM erhalten.

```
10 GOTO 30          10 GOTO30
20 REM *****    30 REM
30 REM ENDE
```

2) In Hochkommata eingeschlossene Zeichenketten werden vom Packen nicht tangiert.

```
10 PRINT "EINGABE ";          10 PRINT"EINGABE ";
```

3) Nicht in Hochkommata eingeschlossene DATA-Strings werden so gepackt, daß sich der durch READ gelesene String nicht ändert.

```
10 DATA 1 , 2 , 3 , 4          10 DATA1 ,2 ,3 ,4
```

In seltenen Fällen, die von PACK nicht erkannt werden, kann das Packen unter ungünstigen Voraussetzungen zu einer Änderung der Funktion des Programmes führen. Ein Beispiel soll das Problem verdeutlichen:

```
10 IF A=F OR I=10 THEN 100
wird zu
10 IFA=FORI=10THEN100
```

gepackt. Wird die Zeile 10 nach dem Packen editiert und dadurch neu in den Zwischencode übersetzt, dann wird das Keyword FOR erkannt und eine Ausführung der Zeile ergäbe einen Syntax-Error.

Vor und nach dem Packen wird der Speicherbedarf des Programmes ausgegeben, um eine Abschätzung des eingesparten Speicherplatzes zu ermöglichen.

4.5 Feldaufzeichnung ARRAY

ARRAY ermöglicht das schnelle Aufzeichnen und Laden von Feldern auf Kassette oder Diskette durch einen einzigen Befehl, wobei für das ganze Feld nur ein Header auf Kassette aufgezeichnet wird.

Funktionsweise:

ARRAY schreibt numerische Felder wie sie im Speicher stehen auf Kassette bzw. Diskette. Bei Stringfeldern wird jeder String - angeführt von seiner Länge - in der Reihenfolge, in der die Strings im Feld angeordnet sind, aufgezeichnet. Die Betriebssystemanbindung ist ähnlich wie bei PACK.

Der Aufruf:

** Kassettenversion:

ARRAY stellt zwei neue Befehle zur Verfügung, die folgendermaßen aufgerufen werden:

```
FIELD PRINT #-1,A(20),B$(20),C(10,10),...
```

```
FIELD INPUT #-1,A(20),B$(20),C(10,10),...
```

Durch den FIELD PRINT-Befehl werden die Feldelemente A(0)..A(20), B\$(0..20), C(0,0)..C(10,10) auf Kassette aufgezeichnet. Die Variablenliste kann beliebig lang sein. Bei eindimensionalen ARRAY kann auch nur ein Teil aufgezeichnet werden:

```
10 DIM A(20)
.
.
100 FIELD PRINT #-1,A(10)
```

Hier werden nur die ersten 11 Elemente aufgezeichnet. Bei mehrdimensionalen Feldern ist es zwar auch möglich nur einen Teil aufzuzeichnen, aber da immer der Bereich im Speicher zwischen Feld(0,0,...,0) und Feld(X1,X2,...,XN) aufgezeichnet wird, kann im allgemeinen nicht davon ausgegangen werden, daß die Elemente Feld(0..X1,0..X2,...,0..XN) darunter sind.

Mit der FIELD INPUT-Anweisung müssen genausoviele Elemente eingelesen werden, wie aufgezeichnet wurden, ansonsten erfolgt ein BS-Error. Reicht bei Feldern der Platz im Stringspace nicht aus, wird der Ladevorgang mit OS-Error abgebrochen. Ist ein Garbage-Collect notwendig, dann hält der Kassettenrekorder für diese Zeit an.

**** Diskettenversion:**

ARRAY stellt zwei neue Befehle zur Verfügung, die folgendermaßen aufgerufen werden:

```
FIELD PRINT #1,A(20),B$(20),C(10,10),...
```

```
FIELD INPUT #1,A(20),B$(20),C(10,10),...
```

Durch den FIELD PRINT-Befehl werden die Feldelemente A(0)..A(20), B\$(0..20), C(0,0)..C(10,10) auf Diskette (File 1) aufgezeichnet. Die Variablenliste kann beliebig lang sein. Bei eindimensionalen ARRAY kann auch nur ein Teil aufgezeichnet werden:

```
10 DIM A(20)
.
.
100 FIELD PRINT #1,A(10)
```

Hier werden nur die ersten 11 Elemente aufgezeichnet. Bei mehrdimensionalen Feldern ist es zwar auch möglich nur einen Teil aufzuzeichnen, aber da immer der Bereich im Speicher zwischen Feld(0,0,...,0) und Feld(X1,X2,...,XN) aufgezeichnet wird, kann im allgemeinen nicht davon ausgegangen werden, daß die Elemente Feld(0..X1,0..X2,...,0..XN) darunter sind.

Mit der FIELD INPUT-Anweisung müssen genausoviele Elemente eingelesen werden, wie aufgezeichnet wurden, ansonsten erfolgt ein BS-Error. Reicht bei Zeichenkettenfeldern der Platz im Stringspace nicht aus, wird der Ladevorgang mit OS-Error abgebrochen. Ist ein Garbage-Collect notwendig, kann es vorkommen, daß das Laufwerk für diese Zeit anhält.

Mit dem folgenden Basicprogramm soll das Schreiben und Lesen von Felder auf und von Diskette mit ARRAY demonstriert. Außerdem zeigt es, wie ein mit ARRAY erzeugtes File, ohne ARRAY gelesen werden kann:

```
10 REM Lesen mit ARRAY
20 CLEAR6000
30 DIM A(10),B$(100)
40 OPEN"I",1,"TEST/TXT"
50 FOR I=1 TO 100:B$(I)=STRING$(59,59):NEXT
60 PRINTFRE(A$)
70 GOSUB 100
80 FIELD INPUT #1,A(10),B$(100)
90 GOSUB 100:END
100 FOR I=0 TO 10:PRINTA(I);:NEXT:PRINT
110 FOR I=1 TO 100:PRINTB$(I):NEXT:RETURN
120 REM Schreiben mit ARRAY
130 CLEAR 6000
140 DIM A(10),B$(100)
150 OPEN"O",1,"TEST/TXT:1"
160 FOR I=1 TO 100:B$(I)=STRING$(I,I+32):NEXT
170 PRINTFRE(A$)
180 FOR I=1 TO 10:A(I)=I:NEXT
190 FIELD PRINT #1,A(10),B$(100)
200 CLOSE
210 END
220 REM Lesen ohne ARRAY
230 CLEAR 6000
240 OPEN"I",1,"TEST/TXT:1","FI"
250 GET1,,N%;
260 N%=N%/4-1 ' 4 für Single
270 DIM A(10)
280 FOR I=0 TO N%
290 GET 1,,A!(I);
300 NEXT
310 GET1,,N%;
320 N%=N%/3-1 ' 3 für String
330 GET 1,,S%;
340 IF FRE(A$)<S% THEN PRINT"Fehler!":END
350 DIM B$(N%)
360 FOR I=0 TO N%
370 GET 1,,(1)J$;
380 J=ASC(J$)
390 IF J>0 THEN GET 1,,(J)B$(I);
400 NEXT
410 GOSUB 100:END
```

4.6 SCRIPSIT-Erweiterung für Umlaute und Druckersteuerung

Diese SCRIPSIT-Modifikation bietet die Möglichkeit der Eingabe der deutschen Umlaute und des scharfen ß und volle Steuerung jedes beliebigen Druckers. Die hervorragenden Möglichkeiten dieser Modifikation zeigt der folgende Text.

Funktionsweise:

Alle Druckerausgaben des SCRIPSIT werden durch Zap's (CD6CFF) auf die Erweiterung umgelenkt. Die Druckeroutine nimmt eine beliebige Umkodierung (1:1) vor und interpretiert die zwei Zeichen, die einem 7FH folgen, als Hexziffern. Auch die Abfrage des Druckerstatus wird auf ein kleines Unterprogramm in der Erweiterung umgeleitet. Dadurch kann das Umlaut-SCRIPSIT durch zwei Zap's in der Erweiterung auf die geänderte Druckeransteuerung beim Genie I, ... angepasst werden. Durch einen kleinen Eingriff (C3ACFF00) in die Tastaturabfrage des SCRIPSIT's wird die Eingabe der Umlaute ermöglicht.

Die SCRIPSIT-Erweiterung wird folgendermaßen implementiert: Die Erweiterung wird mit EDTASM assembliert und der Objektcode unter ZAP/CMD auf Diskette abgelegt. Dann wird ein Original-SCRIPSIT kopiert und die Erweiterung angehängt:

```
COPY SCRIPSIT/LC SCRIPSIT/UP
APPEND ZAP/CMD SCRIPSIT/UP
```

Mit SUPERZAP werden schließlich im SCRIPSIT/UP noch folgende Zap's gemacht. Diese Zap's sind im Format ss,bb,dddddd... notiert, welches bedeutet das im Sektor ss das Byte bb (zu erreichen mit MOD bb) und die folgenden Bytes durch die angegebenen Bytes zu ersetzen sind.

```
00,48,CD6CFF
13,9B,CDCBFF
15,9D,C3ACFF00
20,93,CDCBFF
20,A4,CDCBFF
20,B2,CD6CFF
21,7A,CD6CFF
41,3B,CD6CFF
41,42,CD6CFF
41,49,06B5
```


Handhabung

Die Handhabung ist denkbar einfach. Die deutschen Umlaute und das scharfe ß werden durch folgende Tastenkombinationen erzeugt (§ = @):

ä = § a	= 5BH
ö = § o	= 5CH
ü = § u	= 5DH
Å = § A = § SHIFT a	= 7BH
Ö = § O = § SHIFT o	= 7CH
Ü = § U = § SHIFT u	= 7DH
ß = § :	= 7EH

Bei den großen Umlauten (Å, Ö, Ü) ist die Reihenfolge der Tastendrücke: § SHIFT a, o, u wichtig, wenn man den Zustand des SHIFT-LOCK nicht verändern will.

Druckersteuerung

Die Druckersteuerung ist sehr einfach aufgebaut und unabhängig vom Druckertyp zu verwenden. Bei gleichzeitigen Drücken der Tasten § und P wird der Code 7FH erzeugt, der auf dem Bildschirm als graues Quadrat erscheint. Bei der Ausgabe auf den Drucker wird dieser aber nicht als solcher weitergegeben, sondern in Verbindung mit den folgenden beiden Bytes wird an den Drucker nur ein Byte ausgegeben. Dieses Byte entspricht dem Wert des hinter dem § P in hexadezimaler Schreibweise notierten Bytes.

z.B.:

§1B = gibt an den Drucker nur ein Byte mit dem Wert 1BH aus.

Da der modifizierte Druckertreiber bei dieser Zeichenkombination aus drei ein Zeichen macht, kommt die Zeichenzählung unter Umständen durcheinander.

Anpassung der Zeichencodes an den Druckertyp

Oben können Sie sehen, welche Codes für die Umlaute und das scharfe B benutzt werden. Sollten Sie einen Drucker oder Charaktergenerator mit anderer Zeichencodierung besitzen, können Sie diese nach einer kleinen Tabellenänderung weiterbenutzen. Die Zeichencodierung darf sogar zwischen Drucker und Bildschirm differieren. Die Tabellen können leicht mit dem Superzap-Programm geändert werden. Alle Tabellen sind im File SCRIPSIT/UP (bzw. VP für VG) im Sektor 42 angeordnet. Im folgenden ist jeweils nur noch über den entsprechenden Tabellen die Adresse des ersten Bytes der Tabelle im Sektor angegeben.

	Tastatur 75H	Bildschirm 86H	Drucker 95H
Ä	41H	5BH	5BH
ö	4FH	5CH	5CH
ü	55H	5DH	5DH
ä	61H	7BH	7BH
ö	6FH	7CH	7CH
ü	75H	7DH	7DH
B	3AH	7EH	7EH
	00H	00H	00H
	.	.	.
	00H	00H	00H

Hat zum Beispiel ein Drucker den Code B1H für ä, dann müßte (im NEWDOS-Format) folgendes Zap angebracht werden:

SCRIPSIT/UP,42,95 aus 5B wird B1.

Die 8 Nullen am Ende jeder Tabelle können dazu genutzt werden entweder weitere Funktionen zu implentieren (wenn in die entsprechenden Bytes der drei Tabellen 54H,5FH,0FH eingetragen wird, kann durch Drücken von § SHIFT T das Unterstreichungszeichen erzeugt werden, und beim Drucken schaltet der MX-80 auf Engschrift um.), oder weitere Inkompatibilitäten zwischen der Zeichencodierung auf Bildschirm und Drucker auszumerzen (z.B.: durch 00H,23H,B5H, wird für ein # der Code B5H auf den Drucker ausgegeben).

4.7 Assembler/Disassembler Z80AD

A) Funktionsweise

Das Programmpaket Z80AD besteht aus einem Maschinenprogramm und einem Basicprogramm. Im Teil 6.7 ist jeweils die Kassettenversion aufgelistet, auf der Sourcendiskette (siehe Einleitung Teil 4) ist jeweils die Diskettenversion enthalten. Die Diskettenversion ist auf NEWDOS bzw. GDOS abgestimmt, unter anderen Betriebssystemen sind unter Umständen geringfügige Änderungen notwendig.

Das Maschinenprogramm zu Z80AD wurde nicht mit einem Assembler geschrieben. Das aufgelistete Quellprogrammlisting wurde aus dem Maschinenprogramm mit Hilfe des NEWDOS-Disassembler erstellt (STD). Mit einem Zusatzprogramm wurden die Tabellen generiert. In einer Tabelle sind alle Mnemonics und die jeweils möglichen Operandenkombinationen aufgeführt. Die Operanden sind in einer zweiten Tabellen zusammengefaßt. Nur so konnte der komplexe Z80-Assembler sinnvoll aufgegliedert werden. Der Disassembler benutzt die gleichen Tabellen.

B) Bedienung des Programmpaketes Z80AD

1) <E> ditor

Keine Editierung innerhalb der Zeile möglich. Sortierung nach Zeilennummer wie Basicprogramm. Label muß direkt hinter Zeilennummer beginnen. Nach dem Label folgt ein Leerzeichen und dann der Maschinenbefehl. Enthält eine Zeile kein Label, muß der Maschinenbefehl von einem Leerzeichen angeführt werden.

LIST

Listet gesamten Quellprogrammtext auf.

EXIT

Verläßt den Editor

Nur Kassettenversion:

SAVE

Speichert Quellprogrammtext auf Kassette

LOAD

Lädt Quellprogrammtext von Kassette

Nur Diskettenversion:

SAVE Filename

Speichert Quellprogrammtext auf Diskette

LOAD Filename

Lädt Quellprogrammtext von Diskette

CSAVE

wie SAVE bei Kassettenversion

CLOAD

wie LOAD bei Kassettenversion

2) <A>sssembler

Übersetzt den - mit dem Editor eingegebenen Quellprogrammtext - in Objektcode, der direkt im Speicher unter der mit ORG spezifizierten Adresse abgelegt wird. Außer den Z80-Mnemonics können auch Pseudoopcodes benutzt werden:

Beispiel eines Quellprogrammes:

```
10 LABEL: LD HL, LABEL ; Endlosschleife
20 JP (HL)
30 LD A, 12 ; Konstanten (dezimal)
40 LD A, 0CH ; (hexadezimal)
50 LD A, 'L' ; (ASCII-Zeichen)
60 LD A, 1100B ; (binär)
70 LD HL, $+3 ; Endlosschleife
80 JP (HL)
90 ORG 0F000H ; Speicheradresse des Programmes festlegen
100 DEFB 10 ; 0AH in Speicher ablegen
110 DEFW 20 ; 14H, 00H in aufeinanderfolgende Speicherzellen
120 DEFS 30 ; 30 Speicherzellen reservieren
130 DEFM 'Textstring' ; Text in Speicher ablegen
140 END ; Ende des Quellprogrammes
```

Die Assemblierung wird in 2 Durchläufen (passes) durchgeführt. Während beider Durchläufe können verschiedene Fehlermeldungen erscheinen.

Die Assemblierung kann jederzeit durch Drücken einer beliebigen Taste abgebrochen werden.

3) <D> isassembler

Nach Starten des Disassembler muß die Anfangsadresse eingegeben werden, und zwar immer 4-stellig hexadezimal. Durch Drücken von <ENTER> wird jeweils ein disassemblierter Befehl angezeigt. <D> bewirkt einen Neustart des Disassembler mit einer anderen Adresse, und <E> Verlassen

des Disassemblers.

4) <M> onitor

Der Monitor ermöglicht Anzeigen/Ändern des Speichers und Sichern und Starten von Maschinenprogrammen. Alle Adressen im Monitor müssen 4-stellig hexadezimal eingegeben werden und zwischen dem Befehl und den Parametern muß jeweils ein Leerzeichen stehen.

M XXXX

Der Inhalt der Speicheradresse XXXX wird angezeigt und kann durch Eingabe einer 2-stelligen Hexadezimalzahl überschrieben werden. Durch <ENTER> kann direkt die nächste Speicherzelle angezeigt und durch <.> der M-Mode verlassen werden.

J XXXX

Die Adresse XXXX wird angesprungen und dadurch das dort stehende Maschinenprogramm gestartet. Zuvor wird eine Zahl abgefragt, die in HL an das Maschinenprogramm übergeben wird. Endet das Maschinenprogramm mit RET wird der Wert von HL angezeigt.

S XXXX YYYY ZZZZ NNNNNN

Der Speicherbereich von XXXX bis YYYY mit dem Entrypoint ZZZZ wird auf Kassette unter dem Namen NNNNNN aufgezeichnet.

L (XXXX+-)

Ein Systemfile von Kassette wird mit einem optional anzugebenden Offset in den Speicher geladen

Nur Diskettenversion:

N Doskommando

Unter NEUDOS oder GDOS ist es mit diesem Kommando möglich ein beliebiges DOS-Kommando aufzurufen. So können z. B. Maschinenprogramme von Diskette geladen oder auf Diskette aufgezeichnet werden:

```
N DUMP TEST/CMD 0F900H 0FA00H 0000H<ENTER>
```

Literaturverzeichnis:

- (1) Luidger Röckrath:
TRS-80 Model I, Genie I u. II - ROM-Listing
für Level-II-BASIC-Interpreter.
129 S., DIN A4, kartoniert, 1981, 69,50 DM
ISBN 3-925074-01-5
- (2) Hartmut Grosser, Luidger Röckrath:
TRS-80 Model III ROM-Listing
für Level-II-BASIC-Interpreter.
182 S., DIN A4, kartoniert, 1983, 79,-- DM
ISBN 3-925074-02-3
- (3) Norbert Heicke, Luidger Röckrath:
Colour-Genie ROM-Listing
128 S., DIN A4, kartoniert, 1983, 59,-- DM
ISBN 3-925074-03-1
- (4) Luidger Röckrath:
Microsoftbasic verstanden -
Konzepte, Algorithmen, Datenstrukturen (Arbeitstitel)
erscheint vor. Frühjahr 1985 im Franzis-Verlag, München
- (5) Rodnay Zaks:
Die Programmierung des Z80
Sybex, Düsseldorf
- (6) Luidger Röckrath:
Assemblertricks auf dem Colour-Genie

Die Fortsetzung des Maschinensprachebuches speziell für Colour-Genie Anwender. Mit allen wichtigen Informationen zur Maschinenprogrammierung mit Diskzugriff, Grafikansteuerung, frei programmierbaren Zeichen, Sound u. v. m.. Außerdem die Sourcelistings von SEDIT (Colour-Genie-Version), COLBAS (mit Listingformatierer und PACK), COPBIT (Universeller Kassettenkopierer), SAVETAPE (Universeller Tape/Disk-Konverter).
Erscheint vor. Frühjahr 1985 und wird wahrscheinlich auch nur 49,-- DM kosten.

5.1 Durchschnitt eines Feldes

```

00010 ;*****
00020 ;*
00030 ;* Beispiel 1
00040 ;*
00050 ;* Berechnung des Durchschnittes eines Feldes
00060 ;*
00070 ;* DU=USR(0)A(N)
00080 ;*
00090 ;* DU=(A(1)+A(2)+...+A(N))/N
00100 ;*
00110 ;* (c) 1984 Luidger Röckrath
00120 ;*
00130 ;*****
00140 ;
00150 ; Definitionen
00160 ;
260D 00170 SVAR EQU 260DH ; Variable suchen
0BD2 00180 IADD EQU 0BD2H ; Integeraddition
0716 00190 SADDY EQU 0716H ; Singleaddition
0C77 00200 DADD EQU 0C77H ; Doubleaddition
08A2 00210 SDIVY EQU 08A2H ; Singledivision
0DE5 00220 DDIV EQU 0DE5H ; Doubledivision
0ADB 00230 CDBLX EQU 0ADBH ; CDBL-Funktion
09D2 00240 MOVE EQU 09D2H ; (HL), ... -> (DE), ...
09F4 00250 LDXY EQU 09F4H ; (FACY) --> (FACX)
09FC 00260 LDYX EQU 09FCH ; (FACX) --> (FACY)
09F7 00270 LDXM EQU 09F7H ; (HL)--> (FACX)
0ACF 00280 ICSNGH EQU 0ACFH ; HL --> (FACX) (Single)
0ACC 00290 ICSNGX EQU 0ACCH ; (FACX) (Int.) --> (FACX) (Sin.)
0AEF 00300 TYPXS EQU 0AEFH ; (TYPX) = 3 (Single)
4121 00310 FACX EQU 4121H ; Adresse FACX
4127 00320 FACY EQU 4127H ; Adresse FACY
40AF 00330 TYPX EQU 40AFH ; Type von (FACX)
00340 ; 2 = Integer
00350 ; 3 = String
00360 ; 4 = Single
00370 ; 8 = Double
40BE 00380 USRV EQU 40BEH ; Vektor für USR-Routine
41A9 00390 USRC EQU 41A9H ; Wird von USR aufgerufen
00400 ;
00410 ; Initialisierung
00420 ;
FF00 00430 ORG 0FF00H
00440 ;
FF00 210DFE 00450 INIT LD HL,DURCH
FF03 228E40 00460 LD (USRV),HL
FF06 3EC9 00470 LD A,0C9H ; USR n disablen !
FF08 32A941 00480 LD (USRC),A
FF0B AF 00490 XOR A
FF0C C9 00500 RET
00510 ;
00520 ; Durchschnittberechnung
00530 ;
FF0D DDE1 00540 DURCH POP IX ; RET-Adresse nach IX
FF0F E1 00550 POP HL ; Pointer nach HL
FF10 CD0D26 00560 CALL SVAR ; Variable suchen
FF13 E5 00570 PUSH HL ; Pointer retten
FF14 DDE5 00580 PUSH IX ; RET-Adresse retten
FF16 3AAF40 00590 LD A,(TYPX) ; Typ von X nach A
FF19 6F 00600 LD L,A
FF1A 2600 00610 LD H,0 ; Typ (= Wertlänge) auf
FF1C 19 00620 ADD HL,DE ; Adresse von A(N) addieren
FF1D ED42 00630 SBC HL,BC ; - Adresse A(0) =
00640 ; Länge der Feldwerte
FF1F 3AAF40 00650 LD A,(TYPX) ; Länge d. FW / Typ von X
FF22 CB3C 00660 SCHIEB SRL H ; HL (= LdFW) / 2

```

```

FF24 CB1D      00670      RR      L
FF26 CB3F      00680      SRL     A          ; Typ von X / 2
FF28 FE01      00690      CP      1          ; = 1
FF2A 20F6      00700      JR      NZ,SCHIEB ; nein, weiter
FF2C 5D        00710      LD      E,L       ; N nach DE
FF2D 54        00720      LD      D,H
          00730      ;
FF2E 210000    00740      LD      HL,0      ; FACX = 0
FF31 222141    00750      LD      (FACX),HL
FF34 222341    00760      LD      (FACX+2),HL
FF37 60        00770      LD      H,B       ; Adresse A(0) nach HL
FF38 69        00780      LD      L,C
FF39 3AAF40    00790      LD      A,(TYPX)  ; Typ von X nach A
FF3C 4F        00800      LD      C,A
FF3D 0600      00810      LD      B,0
FF3F 09        00820      ADD     HL,BC     ; Adresse A(1) nach HL
FF40 1B        00830      DEC     DE       ; - 1 (für A(0))
FF41 D5        00840      PUSH   DE
          00850      ;
FF42 D5        00860      LOOP   PUSH DE
FF43 E5        00870      PUSH   HL
FF44 CDFC09    00880      CALL   LDYX      ; (FACX) --> (FACY)
FF47 3AAF40    00890      LD      A,(TYPX)
FF4A FE02      00900      CP      2
FF4C 2006      00910      JR      NZ,L1
FF4E 2A2341    00920      LD      HL,(FACX+2)
FF51 222941    00930      LD      (FACY+2),HL
FF54 E1        00940      L1     POP  HL
FF55 CDF709    00950      CALL   LDXM      ; A(I) --> (FACX)
FF58 EB        00960      EX     DE,HL
FF59 E5        00970      PUSH   HL
FF5A 3AAF40    00980      LD      A,(TYPX)
FF5D FE02      00990      CP      FF
FF5F F5        01000      PUSH   AF
FF60 2A2141    01010      LD      HL,(FACX)
FF63 CCD20A    01020      CALL   Z,ICSNGH+3
FF66 F1        01030      POP    AF
FF67 FE08      01040      CP      8
FF69 F5        01050      PUSH   AF
FF6A CC770C    01060      CALL   Z,DADD
FF6D F1        01070      POP    AF
FF6E ED5B2741  01080      LD      DE,(FACY)
FF72 ED4B2941  01090      LD      BC,(FACY+2)
FF76 C41607    01100      CALL   NZ,SADDY
FF79 E1        01110      POP    HL
FF7A D1        01120      POP    DE
FF7B 1B        01130      DEC     DE
FF7C 7A        01140      LD      A,D      ; Alle Elemente bearbeitet ?
FF7D B3        01150      OR     E
FF7E 20C2      01160      JR      NZ,LOOP  ; nein, --> Weiter
          01170      ;
FF80 CDFC09    01180      CALL   LDYX      ; (FACX) --> (FACY)
FF83 3AAF40    01190      LD      A,(TYPX)
FF86 FE02      01200      CP      2
FF88 2006      01210      JR      NZ,D1
FF8A 2A2341    01220      LD      HL,(FACX+2)
FF8D 222941    01230      LD      (FACY+2),HL
FF90 E1        01240      D1     POP  HL      ; N nach HL
FF91 CDD20A    01250      CALL   ICSNGH+3 ; als Single nach HL
FF94 3AAF40    01260      LD      A,(TYPX)
FF97 FE08      01270      CP      8        ; Double ?
FF99 F5        01280      PUSH   AF
FF9A CCDB0A    01290      CALL   Z,CDBLX   ; CDBL ((FACX))
FF9D F1        01300      POP    AF      ; Single- oder Doubledivision
FF9E ED5B2741  01310      LD      DE,(FACY)
FFA2 ED4B2941  01320      LD      BC,(FACY+2)

```

FFA6	C4EF0A	01330	CALL	NZ,TYPXS
FFA9	C2A208	01340	JP	NZ,SDIVY
FFAC	114A41	01350	LD	DE,414AH
FFAF	211D41	01360	LD	HL,FACX-4
FFB2	D5	01370	PUSH	DE
FFB3	CDD209	01380	CALL	MOVE
FFB6	CDF409	01390	CALL	LDXY
FFB9	112741	01400	LD	DE,FACY
FFBC	E1	01410	POP	HL
FFBD	CDD209	01420	CALL	MOVE
FFC0	CDE50D	01430	CALL	DDIV
FFC3	C9	01440	RET	
FFC0		01450	END	INIT
00000 TOTAL ERRORS				
31713 TEXT AREA BYTES LEFT				

5.2 Ausgabe einer Codetabelle ⁻⁸⁹⁻

```

00010 ;*****
00020 ;*
00030 ;* Beispiel 2
00040 ;*
00050 ;* Ausgabe einer Codetabelle 20H-7FH
00060 ;*
00070 ;* (c) 1984 Luidger Röckrath
00080 ;*
00090 ;*****
00100 ;
00110 ; Definitionen
00120 ;
0033 00130 OUTCH EQU 0033H ; Zeichen ausgeben
000D 00140 CR EQU 000DH ; CR
00150 ;
FF00 00160 ORG OFF00H
00170 ;
FF00 2142FF 00180 START LD HL,TITEL ; Titel drucken
FF03 7E 00190 S1 LD A,(HL) ; Zeichen laden
FF04 A7 00200 AND A ; Textende ?
FF05 2806 00210 JR_ Z,S2 ; Ja, -->
FF07 CD3300 00220 CALL OUTCH ; Zeichen ausgeben
FF0A 23 00230 INC HL ; Zeiger inc.
FF0B 18F6 00240 JR S1
00250 ;
FF0D 0602 00260 S2 LD B,2 ; Zeilenzähler
FF0F 0E20 00270 LD C,' ' ; Zeichencode = ' '
FF11 78 00280 S21 LD A,B ; ZeilenNr. ausgeben
FF12 C630 00290 ADD A,'0'
FF14 CD3300 00300 CALL OUTCH
FF17 3E20 00310 LD A,' ' ; ' : ' ausgeben
FF19 CD3300 00320 CALL OUTCH
FF1C 3E3A 00330 LD A,':'
FF1E CD3300 00340 CALL OUTCH
FF21 3E20 00350 LD A,' '
FF23 CD3300 00360 CALL OUTCH
FF26 79 00370 S3 LD A,C ; Zeichen ausgeben
FF27 CD3300 00380 CALL OUTCH
FF2A 3E20 00390 LD A,' '
FF2C CD3300 00400 CALL OUTCH
FF2F 0C 00410 INC C ; nächstes Zeichen
FF30 79 00420 LD A,C ; Zeilenende ?
FF31 E60F 00430 AND 0FH
FF33 20F1 00440 JR NZ,S3 ; nein, -->
FF35 3E0D 00450 LD A,CR
FF37 CD3300 00460 CALL OUTCH
FF3A 04 00470 INC B ; Zeilennummer + 1
FF3B 78 00480 LD A,B
FF3C FE08 00490 CP 8 ; Zeile 8 ? (Code = 80H)
FF3E 20D1 00500 JR NZ,S21 ; nein, -->
FF40 AF 00510 XOR A
FF41 C9 00520 RET
00530 ;
FF42 1C 00540 TITEL DEFB 1CH ; CLS
FF43 1F 00550 DEFB 1FH
FF44 20 00560 DEFM ' '
FF48 30 00570 DEFM '0 1 2 3 4 5 6 7 8 9 A B C D E F'
FF67 0D 00580 DEFB CR
FF68 20 00590 DEFM ' '
FF6C 2D 00600 DEFM '-----'
FF8B 0D 00610 DEFB CR
FF8C 00 00620 DEFB 0 ; Textende
FF00 00630 END START

```

00000 TOTAL ERRORS
33696 TEXT AREA BYTES LEFT

5.3 POP-Anweisung - 90 -

```

00010 ;*****
00020 ;* *
00030 ;* Beispiel 3 *
00040 ;* *
00050 ;* POP-Befehl löscht einen GOSUB vom Stack *
00060 ;* *
00070 ;* Aufruf: NAME *
00080 ;* *
00090 ;* (c) 1984 Luidger Röckrath *
00100 ;* *
00110 ;*****
00120 ;
00130 ; Definitionen
00140 ;
1936 00150 BSTSRH EQU 1936H ; Suche auf Basicstack
40EB 00160 STKTOP EQU 40EBH ; Ende des Maschinenstacks
19A2 00170 ERROR EQU 19A2H ; Fehleroutine
1D1E 00180 INTL EQU 1D1EH ; Interpreterschleife
0004 00190 RGERR EQU 04H ; 'RG'-Fehlercode
0091 00200 GOSUB EQU 91H ; GOSUB-Token
415E 00210 NAMEV EQU 415EH ; Vektor für NAME-Routine
00220 ;
00230 ; Initialisierung
00240 ;
FF00 00250 ORG 0FF00H
00260 ;
FF00 210DF 00270 INIT LD HL,POP
FF03 228F41 00280 LD (NAMEV+1),HL ; NAMEV ändern
FF06 3EC3 00290 LD A,0C3H ; JMP eintragen
FF08 32BE41 00300 LD (NAMEV),A
FF0B AF 00310 XOR A
FF0C C9 00320 RET
00330 ;
00340 ; POP (=NAME) -Befehl
00350 ;
FF0D E5 00360 POP PUSH HL ; Pointer nach IX retten
FF0E DDE1 00370 POP IX
FF10 16FF 00380 LD D,0FFH ; Nächsten GOSUB-Datensatz
FF12 CD3619 00390 CALL BSTSRH ; auf Basicstack suchen
FF15 F9 00400 LD SP,HL ; Maschinenstack neu initil.
FF16 22E840 00410 LD (STKTOP),HL
FF19 FE91 00420 CP GOSUB ; GOSUB-Datensatz
FF1B 1E04 00430 LD E,RGERR
FF1D C2A219 00440 JP NZ,ERROR ; nein, --> Fehler
FF20 E1 00450 POP HL ; ZNR aus GOSUB-Datensatz
FF21 211E1D 00460 LD HL,INTL ; Interpreterschleifeadresse
FF24 E3 00470 EX (SP),HL ; mit Pointer von Stack vert.
FF25 DDE5 00480 PUSH IX ; Pointer (IX) --> HL
FF27 E1 00490 POP HL
FF28 C9 00500 RET
FF00 00510 END INIT
00000 TOTAL ERRORS
33671 TEXT AREA BYTES LEFT

```

5.4 Demo für Zahlenein-/ausgabe - 91 -

```

00010 ;*****
00020 ;*
00030 ;* Beispiel 4
00040 ;*
00050 ;* Demo für Zahlenein-/ausgabe
00060 ;*
00070 ;* (c) 1984 Luidger Röckrath
00080 ;*
00090 ;*****
00100 ;
00110 ; Definitionen
00120 ;
0E6C 00130 LVAL EQU 0E6CH ; Zahlenstring nach (FACX)
0FBE 00140 FSTR EQU 0FBEH ; (FACX) in Zahlenstring
2B75 00150 PRINTT EQU 2B75H ; Zeichenfolge (HL)... ausg.
0033 00160 OUTCH EQU 0033H ; Zeichen ausgeben
411D 00170 FACX EQU 411DH ; FACX-Adresse (Double)
09D2 00180 MOVE EQU 09D2H ; (HL)... --> (DE)...
00190 ;
FF00 00200 ORG 0FF00H
00210 ;
FF00 2133FF 00220 START LD HL,ZAHL ; Zeiger auf Zahl
FF03 CD6C0E 00230 CALL LVAL ; als Double nach (FACX)
00240 ;
00250 ; (FACX) nach (ZAHL)... retten
00260 ;
FF06 211D41 00270 LD HL,FACX
FF09 1133FF 00280 LD DE,ZAHL
FF0C CDD209 00290 CALL MOVE
00300 ;
00310 ; Schleife
00320 ;
FF0F 0E01 00330 LD C,1 ; Nachkommast. = 0
FF11 2133FF 00340 LOOP LD HL,ZAHL ; (ZAHL)... nach (FACX)
FF14 111D41 00350 LD DE,FACX
FF17 CDD209 00360 CALL MOVE
00370 ;
FF1A 3E80 00380 LD A,80H ; Formatflag
FF1C 0601 00390 LD B,1 ; Vorkommastellen
FF1E C5 00400 PUSH BC ; Formatparam retten
FF1F CDBE0F 00410 CALL FSTR ; (FACX) --> Zeichenfolge
FF22 CD752B 00420 CALL PRINTT ; Zeichenfolge ausgeben
FF25 3E0D 00430 LD A,0DH ; CR ausgeben
FF27 CD3300 00440 CALL OUTCH
FF2A C1 00450 POP BC ; Formatparam zurück
FF2B 0C 00460 INC C ; Nachkommast. + 1
FF2C 79 00470 LD A,C ; = 16
FF2D FE10 00480 CP 16
FF2F 20E0 00490 JR NZ,LOOP ; nein, --> weiter
FF31 AF 00500 XOR A
FF32 C9 00510 RET
00520 ;
FF33 31 00530 ZAHL DEFM '1.234567890123456'
FF44 00 00540 DEFB 0
00550 ;
00560 ; Wird auch als Buffer für (FACX) benutzt !
00570 ;
FF00 00580 END START
00000 TOTAL ERRORS
33646 TEXT AREA BYTES LEFT

```

5.5 Dump eines Systemfiles - 92 -

```

00010 ;*****
00020 ;* *
00030 ;* Beispiel 5 *
00040 ;* *
00050 ;* Systemfile auf Kassette auszeichnen *
00060 ;* *
00070 ;* Aufruf: NAME "NAME",Anfang,Ende,Start *
00080 ;* *
00090 ;* (c) 1984 Luidger Röckrath *
00100 ;* *
00110 ;*****
00120 ;
00130 ; Definitionen
00140 ;
2337 00150 EXP EQU 2337H ; Ausdruck auswerten
2B01 00160 INTEXP EQU 2B01H ; INT (Ausdruck) auswerten
06CC 00170 WSTART EQU 06CCH ; Warmstart
0AF4 00180 TESTS EQU 0AF4H ; (FACX) auf String testen
29F5 00190 DSDZS EQU 29F5H ; String aus DZS löschen
4121 00200 FACX EQU 4121H ; FACX-Adresse
00210 ; TRS/Genie / Colour-Genie
0215 00220 CASON EQU 0215H ; (0065H) Kassettrecorder einschalten
01F8 00230 CASOFF EQU 01F8H ; (0065H) Kassettrecorder ausschalten
02B7 00240 WRITES EQU 02B7H ; (023FH) Header schreiben
0264 00250 WRITEB EQU 0264H ; (021FH) Byte schreiben
022C 00260 BLINK EQU 022CH ; (01E4H) Stern umschalten
3C3E 00270 STADR EQU 3C3EH ; (4426H) Adresse der 2 Sterne
418E 00280 NAMEV EQU 418EH ; Vektor für NAME-Routine
00290 ;
00300 ; Initialisierung
00310 ;
FF00 00320 ORG OFF00H
00330 ;
FF00 210DF 00340 INIT LD HL,DUMP
FF03 22BF41 00350 LD (NAMEV+1),HL ; NAMEV ändern
FF06 3EC3 00360 LD A,0C3H ; JMP eintragen
FF0B 32BE41 00370 LD (NAMEV),A
FF0B AF 00380 XOR A
FF0C C9 00390 RET
00400 ;
00410 ; POP (=NAME) -Befehl
00420 ;
FF0D CD3723 00430 DUMP CALL EXP ; NAME$ auswerten
FF10 CDF40A 00440 CALL TESTS ; String? nein --> Fehler
FF13 ED5B2141 00450 LD DE,(FACX) ; Descriptorzeiger retten
FF17 D5 00460 PUSH DE
FF18 DF 00470 RST 18H
FF19 2C 00480 DEFB ',,' ; Syntaxcheck ',,'
FF1A CD022B 00490 CALL INTEXP+1 ; Anfangsadresse ausw.
FF1D D5 00500 PUSH DE
FF1E DF 00510 RST 18H
FF1F 2C 00520 DEFB ',,'
FF20 CD022B 00530 CALL INTEXP+1 ; Endadresse ausw.
FF23 D5 00540 PUSH DE ; retten
FF24 7E 00550 LD A,(HL) ; Anweisungsende ?
FF25 A7 00560 AND A
FF26 11CC06 00570 LD DE,WSTART ; ja, Startadr. = WSTART
FF29 2805 00580 JR Z,D1 ; ja, -->
FF2B DF 00590 RST 18H
FF2C 2C 00600 DEFB ',,' ; Syntaxcheck ',,'
FF2D CD022B 00610 CALL INTEXP+1 ; nein, Startadr. aus Text
FF30 C1 00620 D1 POP BC ; Endadresse nach BC
FF31 DDE1 00630 POP IX ; Anfangsadresse nach IX
FF33 E3 00640 EX (SF),HL ; Descr.z. mit Pointer vertauschen
FF34 D5 00650 PUSH DE ; Startadresse retten
FF35 C5 00660 PUSH BC ; Endadresse retten

```

FF36 DDE5	00670		PUSH	IX	; Anfangsadresse retten
FF38 CD1502	00680		CALL	CASON	; Kassettenrecorder an
FF3B CD8702	00690		CALL	WRITES	; Header schreiben
FF3E 3E55	00700		LD	A,55H	; Kennung für System-File
FF40 CD6402	00710		CALL	WRITEB	; auf Kassette schreiben
FF43 EB	00720		EX	DE,HL	; String aus DZS löschen
FF44 CDF529	00730		CALL	DSDZS	
FF47 EB	00740		EX	DE,HL	
FF48 7E	00750		LD	A,(HL)	; Länge des Namen
FF49 0606	00760		LD	B,6	; Zähler = 6
FF4B B8	00770		CP	B	; Länge < 6 ?
FF4C 3001	00780		JR	NC,D2	; ja, -->
FF4E 47	00790		LD	B,A	; Zähler = Länge
FF4F 23	00800	D2	INC	HL	; Adresse des Namen nach HL
FF50 7E	00810		LD	A,(HL)	; LSB nach A
FF51 23	00820		INC	HL	
FF52 66	00830		LD	H,(HL)	; MSB nach H
FF53 6F	00840		LD	L,A	; LSB nach L
FF54 C5	00850		PUSH	BC	; Zähler retten
FF55 7E	00860	DNAME	LD	A,(HL)	; Zeichen des Namen laden
FF56 CD6402	00870		CALL	WRITEB	; auf Kassette schreiben
FF59 23	00880		INC	HL	
FF5A 10F9	00890		DJNZ	DNAME	; weiter bis fertig
FF5C C1	00900		POP	BC	; Zähler nach A zurück
FF5D 78	00910	D4	LD	A,B	; Zähler nach A
FF5E FE06	00920		CP	6	; 6 Zeichen geschrieben
FF60 2808	00930		JR	Z,D3	; ja, weiter
FF62 3E20	00940		LD	A,' '	; Leerzeichen
FF64 CD6402	00950		CALL	WRITEB	; auf Kassette
FF67 04	00960		INC	B	; Zähler inc.
FF68 18F3	00970		JR	D4	
	00980				
FF6A D1	00990	D3	POP	DE	; Anfangsadresse nach DE
FF6B E1	01000		POP	HL	; Endadresse nach HL
FF6C E5	01010	D5	PUSH	HL	; Endadresse retten
FF6D 23	01020		INC	HL	; Endadresse + 1
FF6E ED52	01030		SBC	HL,DE	; Endadresse - Anfang d. B.
FF70 282F	01040		JR	Z,DENTRY	; ja, fertig
FF72 010001	01050		LD	BC,256	; Blocklänge = 256
FF75 E5	01060		PUSH	HL	; Restlänge retten
FF76 ED42	01070		SBC	HL,BC	; Restlänge < 256
FF78 E1	01080		POP	HL	
FF79 3801	01090		JR	C,D6	; ja, Blocklänge = Restlänge
FF7B 69	01100		LD	L,C	; nein, Blocklänge = 256
FF7C 45	01110	D6	LD	B,L	; Blocklänge nach B
	01120				
	01130				; Block schreiben (DE=Anfang, B=Länge)
	01140				
FF7D 3E3C	01150		LD	A,3CH	; Anfangskennung auf Kassette
FF7F CD6402	01160		CALL	WRITEB	
FF82 78	01161		LD	A,B	; Blocklänge auf Kassette
FF83 CD6402	01162		CALL	WRITEB	
FF86 7B	01170		LD	A,E	; Adresse auf Kassette (LSB)
FF87 CD6402	01180		CALL	WRITEB	
FF8A 7A	01190		LD	A,D	; (MSB)
FF8B CD6402	01200		CALL	WRITEB	
FF8E 83	01210		ADD	A,E	; D + E nach C (Checksum)
FF8F 4F	01220		LD	C,A	
FF90 1A	01230	LOOP	LD	A,(DE)	; Byte laden
FF91 81	01240		ADD	A,C	; auf Prüfsumme addieren
FF92 4F	01250		LD	C,A	
FF93 1A	01260		LD	A,(DE)	; Byte auf Kassette
FF94 CD6402	01270		CALL	WRITEB	
FF97 13	01280		INC	DE	; Zeiger inc.
FF98 10F6	01290		DJNZ	LOOP	; weiter bis Block fertig
FF9A 79	01300		LD	A,C	; Checksum auf Kassette

FF9B	CD6402	01310	CALL	WRITEB	
FF9E	E1	01320	POP	HL	; Endadresse zurück
FF9F	18CB	01330	JR	D5	
		01340	;		
		01350	;	Entry schreiben	
		01360	;		
FFA1	E1	01370	DENTRY	POP	HL ; Endadresse zurück
FFA2	E1	01380	POP	HL	; Startadresse zurück
FFA3	3E78	01390	LD	A,78H	; Kennung für Entry
FFA5	CD6402	01400	CALL	WRITEB	; auf Kasette
FFA8	7D	01410	LD	A,L	; Startadresse (LSB)
FFA9	CD6402	01420	CALL	WRITEB	; auf Kasette
FFAC	7C	01430	LD	A,H	; (MSB)
FFAD	CD6402	01440	CALL	WRITEB	
FFB0	CDFB01	01450	CALL	CASOFF	; Kassettenrecorder aus
FFB3	E1	01460	POP	HL	; Pointer zurück
FFB4	C9	01470	RET		
		01480	;		
FF00		01490	END	INIT	
00000	TOTAL	ERRORS			
30753	TEXT	AREA	BYTES	LEFT	

6.1 Bildschirmeditor SEDIT - 95 -

```

00020 ; ** TRS-80-SCREEN-EDITOR **
00030 ; COPYRIGHT BY LUIDGER ROECKRATH
00040 ; KAISERSTR. 54
00050 ; 4050 MOENCHENGLADBACH 1
00060 ; JUNI 1981
00070 ;BASIC-UNTERPROGRAMME:
002B 00080 INCH EQU 002BH ;TASTATURABFRAGE
0033 00090 DUTCH EQU 0033H ;ZEICHENAUSGABE
054F 00100 SCROLL EQU 054FH ;BILDSCHIRMSCROLL
01C9 00110 CLEAR EQU 01C9H ;BILDSCHIRM LOESCHEN
0573 00120 CLEOL EQU 0573H ;BIS ZEILENENDE LOESCHEN
04A1 00130 CR EQU 04A1H ;CURSOR ZUM ZEILENANFANG
04E7 00140 DOWN EQU 04E7H ;CURSOR EINE ZEILE RUNTER
1A7B 00150 RET EQU 1A7BH ;RUECKSPRUNGADRESSE
19A2 00160 ERROR EQU 19A2H ;BASIC-ERRORROUTINE
0060 00170 DELAY EQU 0060H ;ZEITSCHLEIFE
03FE 00180 DECODE EQU 03FEH ;TASTATURDECODIERUNG
04C3 00190 CPL64 EQU 04C3H ;AUF 64CPL SCHALTEN
00200 ;BASIC-ZEIGER:
4020 00210 CURSOR EQU 4020H ;CURSORADRESSE
4016 00220 KEYVEC EQU 4016H ;KEYBOARD-TREIBER-VEKTOR
401E 00230 DISVEC EQU 401EH ;DISPLAY-TREIBER-VEKTOR
40A7 00240 BUFFER EQU 40A7H ;I/O-BUFFER ANFANG
3801 00250 KEYBD EQU 3801H ;TASTATURADRESSE
4036 00260 LAST EQU 4036H ;LEZTER TASTATURSTATUS
00270 ;INTERNE SYMBOLE
0080 00280 MARKER EQU 80H ;TRENNT LOGISCHE ZEILEN
008F 00290 CURCH EQU 8FH ;CURSORZEICHEN
00300 ;PARAMETER:
0258 00310 BLINK EQU 600 ;BLINKFREQUENZ
0400 00320 EZEIT EQU 400H ;EINSATZZEIT BEI REPEAT
0080 00330 RFREQU EQU 128 ;WIEDERHOLUNGSFREQU.
00340 ;BELEGUNG DER ZWEITREGISTER:
00350 ;A:ZEICHEN AUF CURSORPOSITION
00360 ;BC:ZAEHLER FUER CURSORBLINKEN
00370 ;D:ZEICHEN FUER WIEDERHOLUNG
00380 ;E:FLAG
00390 ;HL:ZAEHLER FUER REPEAT
00400 ; INITIALISIERUNG:
FDAF 00420 ORG OFDAFH
00430 ;REPEAT INITIALISIEREN
FDAF 2A1640 00440 INIT LD HL,(KEYVEC)
FDB2 A7 00450 AND A
FDB3 11E303 00460 LD DE,03E3H
FDB6 ED52 00470 SBC HL,DE
FDB8 200A 00480 JR NZ,NOKEY
FDBA 21B7FF 00490 LD HL,REPEAT ;KEYBOARDVEKTOR AENDERN
FDBD 221640 00500 LD (KEYVEC),HL
FDC0 D9 00510 EXX ;ZEICHENSPEICHER LOESCHEN
FDC1 1600 00520 LD D,0
FDC3 D9 00530 EXX
00540 ;SEIT INITIALISIEREN
FDC4 2A1E40 00550 NOKEY LD HL,(DISVEC)
FDC7 22E5FD 00560 LD (OUT+1),HL
FDCA 21D1FD 00570 LD HL,SEIT ;DISPLAYVEKTOR AENDERN
FDCD 221E40 00580 LD (DISVEC),HL
00590 ;GEGEN BASIC SCHUETZEN (INIT NICHT)
FDD0 C9 00600 RET
00610 ; HAUPTPROGRAMM
FDD1 210E00 00630 SEDIT LD HL,14 ;RUECKSPRUNGADRESSE
FDD4 39 00640 ADD HL,SP ;DURCH STACKMANIPULATION
FDD5 5E 00650 LD E,(HL) ;ERMITTELN
FDD6 23 00660 INC HL
FDD7 56 00670 LD D,(HL)
FDD8 E5 00680 PUSH HL
FDD9 217B1A 00690 LD HL,RET

```

FDDC DF	00700	RST 1BH	;=RET?
FDDD E1	00710	POF HL	
FDDE 2824	00720	JR Z,EDIT	;JA
FDE0 79	00730	LD A,C	;NORMALE AUSGABE
FDE1 D60D	00740	SUB 0DH	;CR?
FDE3 A7	00750	AND A	;CARRY LOESCHEN!
FDE4 C20000	00760 OUT	JP NZ,0	;NEIN
FDE7 DD6E03	00770	LD L,(IX+3)	
FDEA DD6604	00780	LD H,(IX+4)	
FDED DD7E05	00790	LD A,(IX+5)	
FDFO A7	00800	AND A	
FDf1 2801	00810	JR Z,000	
FDf3 77	00820	LD (HL),A	
FDf4 E5	00830 000	PUSH HL	
FDf5 CDA104	00840	CALL CR	
FDf8 CDE704	00850	CALL DOWN	
FDfB EB	00860	EX DE,HL	
FDfC 1B	00870	DEC DE	
FDfD E1	00880	POP HL	
FDfE CDA2FE	00890	CALL SKIP1	
FE01 C3F3FF	00900	JP CR1	
FE04 23	00910 EDIT	INC HL	;NEUER STACK
FE05 F9	00920	LD SP,HL	
FE06 3E0F	00930	LD A,15	;'-'-CURSOR OFF
FE08 CD3300	00940	CALL OUTCH	
FE0B CDC304	00950	CALL CPL64	;ARBEITET NUR MIT 64 CPL
FE0E D9	00960	EXX	;FLAG LOESCHEN
FE0F 1E00	00970	LD E,0	
FE11 D9	00980	EXX	
FE12 2A2040	00990 LOOP	LD HL,(CURSOR)	
FE15 08	01000	EX AF,AF'	;CURSOR ON
FE16 7E	01010	LD A,(HL)	;ZEICHEN NACH A'
FE17 08	01020	EX AF,AF'	
FE18 368F	01030	LD (HL),CURCH	
FE1A D9	01040 LOOP1	EXX	;ZAEHLER FUER CURSOR-
FE1B 015802	01050	LD BC,BLINK	;BLINKEN INITIALISIEREN
FE1E D9	01060	EXX	
FE1F CD2B00	01070 SCAN	CALL INCH	
FE22 4F	01080	LD C,A	;1BH UND 5BH VERTAUSCHEN
FE23 E6BF	01090	AND 0BFH	; (SHIFT A UND A)
FE25 FE1B	01100	CP 1BH	
FE27 79	01110	LD A,C	
FE28 2002	01120	JR NZ,E1	
FE2A EE40	01130	XOR 40H	
FE2C A7	01140 E1	AND A	;TASTE GEDRUECKT?
FE2D 2011	01150	JR NZ,BRANCH	;JA
FE2F D9	01160	EXX	;BLINKZ. DEC.
FE30 0B	01170	DEC BC	
FE31 78	01180	LD A,B	;BLINKZ.=0?
FE32 B1	01190	OR C	
FE33 D9	01200	EXX	
FE34 20E9	01210	JR NZ,SCAN	;NEIN
FE36 7E	01220	LD A,(HL)	
FE37 FE8F	01230	CP CURCH	;CURSOR ON?
FE39 20D7	01240	JR NZ,LOOP	;NEIN
FE3B 08	01250	EX AF,AF'	;CURSOR OFF
FE3C 77	01260	LD (HL),A	; (ZEICHEN ZURUECK)
FE3D 08	01270	EX AF,AF'	
FE3E 18DA	01280	JR LOOP1	;WEITER
FE40 1112FE	01290 BRANCH	LD DE,LOOP	;RUECKSPRUNGADRESSE
FE43 D5	01300	PUSH DE	
FE44 08	01310	EX AF,AF'	;CURSOR OFF
FE45 77	01320	LD (HL),A	
FE46 08	01330	EX AF,AF'	
FE47 FE60	01340	CP 'E'+20H	;SKIP?
FE49 2850	01350	JR Z,SKIP	;JA

FE4B	FE20	01360	CP ' '	; ZEICHEN < ' '
FE4D	380A	01370	JR C,CNTL	; JA
FE4F	D9	01380	EXX	; INSERT-FLAG GESETZT
FE50	CB43	01390	BIT 0,E	
FE52	D9	01400	EXX	
FE53	C2DCFE	01410	JP NZ, INSERT	; JA
FE56	C33300	01420	JP OUTCH	; ZEICHEN AUSGEBEN
FE59	D9	01430	CNTL EXX	; INSERT-FLAG SETZEN
FE5A	CB03	01440	SET 0,E	
FE5C	D9	01450	EXX	
FE5D	FE19	01460	CP 19H	; INSERT?
FE5F	C8	01470	RET Z	; JA
FE60	D9	01480	EXX	; INSERT-FLAG ZURUECKSETZ.
FE61	CB83	01490	RES 0,E	
FE63	D9	01500	EXX	
FE64	FE18	01510	CP 18H	; DELETE?
FE66	2862	01520	JR Z,DELETE	; JA
FE68	FE1F	01530	CP 1FH	; CLEAR?
FE6A	CAC901	01540	JP Z,CLEAR	; JA
FE6D	FE01	01550	CP 1	; BREAK?
FE6F	2838	01560	JR Z,BREAK	; JA
FE71	FE0D	01570	CP 0DH	; ENTER?
FE73	CA4AFF	01580	JP Z,RETURN	; JA
FE76	11C0FF	01590	LD DE,-64	; ZEILE RAUF
FE79	FE1B	01600	CP 1BH	; UP?
FE7B	2812	01610	JR Z,MOVE	; JA
FE7D	11FFFF	01620	LD DE,-1	; ZEICHEN ZURUECK
FE80	D608	01630	SUB 8	; BACK?
FE82	2808	01640	JR Z,MOVE	; JA
FE84	110100	01650	LD DE,1	; ZEICHEN VOR
FE87	3D	01660	DEC A	; ADVANCE?
FE88	2805	01670	JR Z,MOVE	; JA
FE8A	114000	01680	LD DE,64	; ZEILE RUNTER
FE8D	3D	01690	DEC A	; DOWN?
FE8E	C0	01700	RET NZ	; NEIN
FE8F	19	01710	MOVE ADD HL,DE	; ZUM CURSOR ADDIEREN
FE90	7C	01720	LD A,H	; CURSOR NOCH IM VIDEORAM?
FE91	FE40	01730	CP 40H	
FE93	C8	01740	RET Z	; NEIN, AENDERUNG INGN.
FE94	FE3B	01750	CP 3BH	
FE96	C8	01760	RET Z	; NEIN
FE97	222040	01770	LD (CURSOR),HL	; NEUEN CURSOR ABSPEICHERN
FE9A	C9	01780	RET	
FE9B	CD9BFF	01790	SKIP CALL SEARCH	; LETZTES ZEICHEN SUCHEN
FE9E	EB	01800	EX DE,HL	; LETZTES ZEICHEN
FE9F	DF	01810	RST 18H	; HINTER CURSOR?
FEA0	EB	01820	EX DE,HL	
FEA1	D8	01830	RET C	; JA
FEA2	3620	01840	SKIP1 LD (HL),' '	; ZEICHEN LOESCHEN
FEA4	DF	01850	RST 18H	; LETZTES ZEICHEN GEL.?
FEA5	23	01860	INC HL	
FEA6	20FA	01870	JR NZ,SKIP1	; NEIN
FEA8	C9	01880	RET	
FEA9	21FF3F	01890	BREAK LD HL,3FFFH	; LETZTES VIDEORAMBYTE
FEAC	7E	01900	BREAK1 LD A,(HL)	
FEAD	FE20	01910	CP ' '	; BLANK?
FEAF	2006	01920	JR NZ,BREAK2	; NEIN
FEB1	2B	01930	DEC HL	; JA, WEITER ZURUECK
FEB2	7C	01940	LD A,H	; VIDEORAMANF. ERREICHT?
FEB3	FE3B	01950	CP 3BH	
FEB5	20F5	01960	JR NZ,BREAK1	; NEIN
FEB7	FE80	01970	BREAK2 CP 80H	
FEB9	2801	01980	JR Z,BREAK3	
FEBB	23	01990	INC HL	
FEBC	7C	02000	BREAK3 LD A,H	
FEBD	FE40	02010	CP 40H	

FEBF	CC4F05	02020	CALL Z, SCROLL	
FEC2	222040	02030	LD (CURSOR), HL	; NEUER CURSOR
FEC5	3E0D	02040	LD A, 0DH	; CR
FEC7	C33300	02050	JP OUTCH	
FECA	23	02060	DELETE INC HL	; LETZTES ZEICHEN AB
FECB	CD9BFF	02070	CALL SEARCH	; CURSOR+1 SUCHEN
FECE	2B	02080	DEC HL	
FECF	DF	02090	RST 18H	; MIT CURSOR VERGL.
FED0	2807	02100	JR Z, DEL	; SPRUNG, WENN GLEICH
FED2	D0	02110	RET NC	; FERTIG WENN HINTER CURSOR
FED3	54	02120	LD D, H	; CURSOR=ZIELADRESSE
FED4	5D	02130	LD E, L	; FUER BLOCKMOVE
FED5	23	02140	INC HL	; CURSOR+1=QUELLADR.
FED6	EDB0	02150	LDIR	; MOVE
FED8	2B	02160	DEC HL	; ZURUECK ZUM LETZTEN BYTE
FED9	3620	02170	DEL LD (HL), ' '	; MIT SPACE UEBERSCHR.
FEDB	C9	02180	RET	
FEDC	F5	02190	INSERT PUSH AF	; ZEICHEN RETTEN
FEDD	CD9BFF	02200	CALL SEARCH	; LETZTES ZEICHEN SUCHEN
FEE0	7E	02210	LD A, (HL)	; CURSOR AUF MARKER?
FEE1	FE80	02220	CP MARKER	
FEE3	2009	02230	JR NZ, L1	; NEIN
FEE5	D9	02240	EXX	; FLAG SETZEN
FEE6	CBCB	02250	SET 1, E	
FEE8	D9	02260	EXX	
FEE9	54	02270	LD D, H	; ZEIGER AUF LETZTES Z.
FEEA	5D	02280	LD E, L	; =CURSOR-1
FEEB	1B	02290	DEC DE	
FEEC	1805	02300	JR L2	
FEEE	DF	02310	L1 RST 18H	; VERGL. CURSOR-LETZTES Z.
FEFF	2802	02320	JR Z, L2	; SPRUNG, WENN GLEICH
FEF1	3053	02330	JR NC, L3	; CURSOR HINTER L. Z.
FEF3	7B	02340	L2 LD A, E	; L. Z. AM ZEILENENDE?
FEF4	E63F	02350	AND 3FH	
FEF6	FE3E	02360	CP 3EH	
FEF8	203F	02370	JR NZ, L4	; NEIN
FEFA	E5	02380	PUSH HL	; ENDE DER LETZTEN ZEILE
FEFB	21FE3F	02390	LD HL, 3FFE3F	
FEFE	DF	02400	RST 18H	
FEFF	E1	02410	POP HL	
FF00	200D	02420	JR NZ, L5	; NEIN
FF02	D5	02430	PUSH DE	; JA, SCROLL
FF03	CD4F05	02440	CALL SCROLL	
FF06	222040	02450	LD (CURSOR), HL	
FF09	D1	02460	POP DE	
FF0A	21C0FF	02470	LD HL, 0FFC0H	; DE=DE-64
FF0D	19	02480	ADD HL, DE	
FF0E	EB	02490	EX DE, HL	
FF0F	21BE3F	02500	L5 LD HL, 3FBE3F	; MUSS EIN REST NACH
FF12	DF	02510	RST 18H	; UNTEN VERSCHOBEN WERDEN?
FF13	D5	02520	PUSH DE	
FF14	23	02530	INC HL	
FF15	23	02540	INC HL	
FF16	2817	02550	JR Z, L7	; NEIN
FF18	13	02560	INC DE	; NAECHSTES ZEICHEN MARKER?
FF19	1A	02570	LD A, (DE)	
FF1A	FE80	02580	CP MARKER	
FF1C	2011	02590	JR NZ, L7	; NEIN, KEINE VERSCHIEBUNG
FF1E	EB	02600	EX DE, HL	; HL=ANFANG DER NAECHSTEN Z.
FF1F	23	02610	INC HL	
FF20	1B	02620	DEC DE	; QUELLADRESSE (7FBF)
FF21	DD21FF3F	02630	LD IX, 3FFFH	; ZIELADRESSE
FF25	1A	02640	L6 LD A, (DE)	
FF26	DD7700	02650	LD (IX+0), A	
FF29	DF	02660	RST 18H	; FERTIG?
FF2A	1B	02670	DEC DE	

FF2B	DD2B	02680		DEC IX	
FF2D	20F6	02690		JR NZ,L6	;NEIN
FF2F	CD7305	02700	L7	CALL CLEOL	;FREIGEWORDENE ZEILE
FF32	D1	02710		POP DE	;LOESCHEN
FF33	7D	02720		LD A,L	;NEU MARKIEREN
FF34	F63F	02730		OR 3FH	
FF36	6F	02740		LD L,A	
FF37	3680	02750		LD (HL),MARKER	
FF39	D9	02760	L4	EXX	;FLAG PRUEFEN
FF3A	CB4B	02770		BIT 1,E	
FF3C	CB8B	02780		RES 1,E	;UND LOESCHEN
FF3E	D9	02790		EXX	
FF3F	2005	02800		JR NZ,L3	;GESETZT?--KEIN MOVE
FF41	62	02810		LD H,D	;EIN ZEICHEN VORWAERTS
FF42	6B	02820		LD L,E	;VERSCHIEBEN
FF43	13	02830		INC DE	
FF44	EDB8	02840		LDDR	
FF46	F1	02850	L3	POP AF	;NEUES ZEICHEN EINFUEGEN
FF47	C33300	02860		JP OUTCH	
FF4A	CD9BFF	02870	RETURN	CALL SEARCH	;LETZTES ZEICHEN SUCHEN
FF4D	E5	02880		PUSH HL	;CURSOR RETTEN
FF4E	1805	02890		JR R6	
FF50	7E	02900	R1	LD A,(HL)	;ERSTES ZEICHEN SUCHEN
FF51	FE80	02910		CP MARKER	
FF53	2806	02920		JR Z,R4	
FF55	2B	02930	R6	DEC HL	
FF56	7C	02940		LD A,H	
FF57	FE3B	02950		CP 3BH	;ANF. DES VIDEORAM?
FF59	20F5	02960		JR NZ,R1	;NEIN
FF5B	23	02970	R4	INC HL	
FF5C	7E	02980		LD A,(HL)	
FF5D	FE20	02990		CP ' '	
FF5F	28FA	03000		JR Z,R4	
FF61	0600	03010		LD B,0	;ZAEHLER
FF63	DD2AA740	03020		LD IX,(BUFFER)	;IN I/O-BUFFER
FF67	7E	03030	R2	LD A,(HL)	;UEBERTRAGEN
FF68	FE20	03040		CP ' '	
FF6A	3002	03050		JR NC,R8	
FF6C	C640	03060		ADD A,40H	
FF6E	DD7700	03070	R8	LD (IX+0),A	
FF71	DF	03080		RST 18H	;FERTIG?
FF72	23	03090		INC HL	
FF73	DD23	03100		INC IX	
FF75	280D	03110		JR Z,R3	;JA
FF77	3007	03120		JR NC,LEER	;LEERE ZEILE
FF79	10EC	03130		DJNZ R2	;256 ZEICHEN UEBERTR.
FF7B	1E1C	03140		LD E,1CH	;LS-ERROR
FF7D	C3A219	03150		JP ERROR	
FF80	DD2B	03160	LEER	DEC IX	;ZAEHLER AUF ANFANG
FF82	E1	03170		POP HL	;CURSOR ZURUECK
FF83	E5	03180		PUSH HL	
FF84	D1	03190	R3	POP DE	
FF85	DD360000	03200		LD (IX+0),0	;ENDE MARKIEREN
FF89	CDA104	03210		CALL CR	;IN NAECHSTE ZEILE
FF8C	CDE704	03220		CALL DOWN	
FF8F	CDF3FF	03230		CALL CR1	
FF92	2AA740	03240		LD HL,(BUFFER)	;PARAMETER FUER BASIC
FF95	2B	03250		DEC HL	
FF96	AF	03260		XOR A	
FF97	D1	03270		POP DE	;STACK KORRIG.
FF98	C37E1A	03280		JP RET+3	;ZURUECK ZUM BASIC
FF9B	E5	03290	SEARCH	PUSH HL	;CURSOR RETTEN
FF9C	010000	03300		LD BC,0	;ZEICHENZAehler
FF9F	7E	03310	S1	LD A,(HL)	;MARKER SUCHEN
FFA0	FE80	03320		CP MARKER	
FFA2	2808	03330		JR Z,S2	

FFA4 23	03340	INC HL	
FFA5 03	03350	INC BC	
FFA6 7C	03360	LD A,H	
FFA7 FE40	03370	CP 40H	;BILDSCHIRMEDE?
FFA9 20F4	03380	JR NZ,S1	;NEIN
FFAB 2B	03390	DEC HL	;AUF LETZTEN BILDSCHIRML.
FFAC 03	03400 S2	INC BC	
FFAD 2B	03410 S3	DEC HL	;LEERZEICHEN IGNORIEREN
FFAE 0B	03420	DEC BC	
FFAF 7E	03430	LD A,(HL)	
FFB0 FE20	03440	CP ' '	
FFB2 2BF9	03450	JR Z,S3	
FFB4 E3	03460	EX (SP),HL	;CURSOR ZURUECK
FFB5 D1	03470	POP DE	; (DE)=LETZTES ZEICHEN
FFB6 C9	03480	RET	
FFB7 213640	03490 REPEAT	LD HL, LAST	;LETZTEN TASTATURSTATUS
FFBA 01013B	03500	LD BC,KEYBD	; (4036-403C) MIT ZUSTAND
FFBD 1600	03510	LD D,0	;DER TASTATURVERGLEICHEN
FFBF 0A	03520 REP1	LD A,(BC)	
FFC0 5F	03530	LD E,A	
FFC1 AE	03540	XOR (HL)	;AENDERUNG?
FFC2 73	03550	LD (HL),E	
FFC3 2014	03560	JR NZ,REP2	;JA
FFC5 14	03570	INC D	
FFC6 2C	03580	INC L	
FFC7 CB01	03590	RLC C	;ALLE ZEILEN VERGL.
FFC9 F2BFFF	03600	JP P,REP1	;JA
FFCC D9	03610	EXX	;ZAEHLER DEC.
FFCD 2B	03620	DEC HL	
FFCE 7C	03630	LD A,H	;=0?
FFCF B5	03640	OR L	
FFD0 3E00	03650	LD A,0	
FFD2 2003	03660	JR NZ,RETREP	;NEIN
FFD4 2E80	03670	LD L,RFREQU	;ZAEHLER NEU SETZEN
FFD6 7A	03680	LD A,D	;ZEICHEN IN AKKU
FFD7 D9	03690 RETREP	EXX	
FFDB C9	03700	RET	
FFD9 5F	03710 REP2	LD E,A	;ENTPRELLUNG:
FFDA C5	03720	PUSH BC	;19 MS WARTEN
FFDB 010005	03730	LD BC,500H	
FFDE CD6000	03740	CALL DELAY	
FFE1 C1	03750	POP BC	
FFE2 0A	03760	LD A,(BC)	;TASTE NOCH IMMER GEDR.
FFE3 A3	03770	AND E	
FFE4 2806	03780	JR Z,REP3	;NEIN
FFE6 7A	03790	LD A,D	;IN ASCII UMWANDELN
FFE7 07	03800	RLCA	
FFEB 07	03810	RLCA	
FFE9 CDFE03	03820	CALL DECODE	
FFEC D9	03830 REP3	EXX	
FFED 210004	03840	LD HL,EZEIT	;ZAEHLER MIT EINSATZZEIT
FFF0 57	03850	LD D,A	;INITIALISIEREN
FFF1 D9	03860	EXX	;UND ZEICHEN NACH D'
FFF2 C9	03870	RET	
FFF3 7C	03880 CR1	LD A,H	
FFF4 FE40	03890	CP 40H	
FFF6 CC4F05	03900	CALL Z,SCROLL	
FFF9 222040	03910	LD (CURSOR),HL	
FFFC 2B	03920	DEC HL	;ZEILENENDE MARKIEREN
FFFD 3680	03930	LD (HL),MARKER	
FFFF C9	03940	RET	
FDAF	03950	END INIT	

00000 TOTAL ERRORS

26158 TEXT AREA BYTES LEFT

6.2 Basicerweiterung NEWBAS - 101 -

```

00130 ;*****
00140 ;*****      N E W B A S      *****
00150 ;*****
00160 ;
00170 ;
00190 ;
012D  00200 L3      EQU      012DH
00210 ;
00220 ;
00230 ;
00240 ;***** INITIALISIERUNG DER BASIC ZEIGER *****
00250 ;
00260 ;
4155  00270          ORG      4155H
4155  C3A7F8        JP      ASC
00290
415B  00300          ORG      415BH
415B  C36EF9        JP      DEF
00320
417F  00330          ORG      GETP
417F  C35EFE        JP      GET
00350
418B  00360          ORG      418BH
418B  C309FD        JP      MERGE
418E  C35CFA        JP      NAME
00390
4194  00400          ORG      4194H
4194  C3ECF9        JP      HEX
00420
419A  00430          ORG      419AH
419A  C336FD        JP      RSET
419D  C3ECFD        JP      INSTR
00460
41A3  00470          ORG      41A3H
41A3  C3C2FD        JP      LINE
00490
4173  00500          ORG      CMD
4173  C34CFF        JP      DIRECT
00520 ;
00530 ;
F837  00540          ORG      0F837H
00550 ;
F837  2A1E40        00560 NEWBAS LD      HL, (DODCB+1) ;ADRESSE DES VIDEO TREIBER
F83A  228DFF        00570 LD      (DDVVR1+1),HL ;IN JP-BEFEHL EINSETZEN
F83D  2A2640        00580 LD      HL, (PRDCB+1) ;ADRESSE DES PRINT TREIBER
F840  22B0FF        00590 LD      (PRDVR1+1),HL ;IN JP-BEFEHL EINSETZEN
00600 ;
F843  2A0440        00610 LD      HL, (RST10V)
F846  22DEFF        00620 LD      (CHV10+1),HL
F849  21CDFF        00630 LD      HL, RESTN
F84C  220440        00640 LD      (RST10V),HL ;RST 10H-VEKTOR AENDERN
00650 ;
F84F  3EC3          00660 LD      A, 0C3H
F851  32A941        00670 LD      (41A9H),A
F854  2192F9        00680 LD      HL, USR
F857  22AA41        00690 LD      (41AAH),HL
F85A  32AC41        00700 LD      (41ACH),A
00720 ;
F85D  21DCFC        00740 LD      HL, READY
F860  22AD41        00750 LD      (41ADH),HL
F863  32D941        00760 LD      (41D9H),A
F866  214CFD        00770 LD      HL, MID
F869  22DA41        00780 LD      (41DAH),HL
F86C  AF           00781 XOR      A
F86D  C9           00790 RET
00800 ;

```

	00820 ;				F8D7 37	01480	SCF	
	00830 ;				F8D8 C36EF8	01490 AS11	JP	AS5
F86E 3AAF40	00840 AS5	LD	A, (40AFH)		F8DB D22324	01500 AS6	JP	NC, 2423H
F871 3D	00850	DEC	A		F8DE D5	01510	PUSH	DE
F872 3D	00860	DEC	A		F8DF EB	01520	EX	DE, HL
F873 3D	00870	DEC	A		F8E0 CD8828	01530	CALL	2888H
F874 B7	00880	OR	A		F8E3 D1	01540	POP	DE
F875 37	00890	SCF			F8E4 AF	01550	XOR	A
F876 CADBF8	00900	JP	Z, AS6		F8E5 E5	01560 AS9	PUSH	HL
F879 4E	00910	LD	C, (HL)		F8E6 F5	01570	PUSH	AF
F87A 23	00920	INC	HL		F8E7 EB	01580	EX	DE, HL
F87B 46	00930	LD	B, (HL)		F8E8 7E	01590	LD	A, (HL)
F87C C5	00940	PUSH	BC		F8E9 FE29	01600	CP	29H
F87D FA98F8	00950	JP	M, AS7		F8EB 20DA	01610	JR	NZ, AS10
F880 23	00960	INC	HL		F8ED 2AF340	01620	LD	HL, (40F3H)
F881 4E	00970	LD	C, (HL)		F8F0 CF	01630	RST	8
F882 23	00980	INC	HL		F8F1 2BE5	01640	JR	Z, AS11
F883 46	00990	LD	B, (HL)		F8F3 2AD840	01650	LD	HL, (40DBH)
F884 C5	01000	PUSH	BC		F8F6 CD0D26	01660 AS13	CALL	260DH
F885 E298F8	01010	JP	PD, AS7		F8F9 E3	01670	EX	(SP), HL
F888 23	01020	INC	HL		F8FA CD2B1F	01680	CALL	1F2BH
F889 DA8FF8	01030	JP	C, AS8		F8FD 7E	01690	LD	A, (HL)
F88C 211D41	01040	LD	HL, 411DH		F8FE FE29	01700	CP	29H
F88F 4E	01050 ASB	LD	C, (HL)		F900 2B07	01710	JR	Z, AS12
F890 23	01060	INC	HL		F902 CF	01720	RST	8
F891 46	01070	LD	B, (HL)		F903 2C	01730	INC	L
F892 23	01080	INC	HL		F904 E3	01740	EX	(SP), HL
F893 C5	01090	PUSH	BC		F905 CF	01750	RST	8
F894 4E	01100	LD	C, (HL)		F906 2C	01760	INC	L
F895 23	01110	INC	HL		F907 18ED	01770	JR	AS13
F896 46	01120	LD	B, (HL)		F909 D7	01780 AS12	RST	16
F897 C5	01130	PUSH	BC		F90A E3	01790	EX	(SP), HL
F898 DA5F8	01140 AS7	JP	C, AS9		F90B CF	01800	RST	8
F89B CF	01150 AS1	RST	8		F90C 29	01810	ADD	HL, HL
F89C BE	01160	CP	(HL)		F90D 3ED5	01820 AS3	LD	A, 0D5H
F89D 3E80	01170	LD	A, 80H		F90F CF	01830	RST	8
F89F 32DC40	01180	LD	(40DCH), A		F910 D5	01840	PUSH	DE
F8A2 B6	01190	OR	(HL)		F911 CD3723	01850	CALL	2337H
F8A3 47	01200	LD	B, A		F914 2B	01860	DEC	HL
F8A4 C31226	01210	JP	2612H		F915 D7	01870	RST	16
F8A7 CD9BF8	01220 ASC	CALL	AS1		F916 C29719	01880	JF	NZ, 1997H
F8AA 3AAF40	01230	LD	A, (40AFH)		F919 E7	01890	RST	32
F8AD B7	01240	OR	A		F91A 282F	01900	JR	Z, AT1
F8AE F5	01250	PUSH	AF		F91C D1	01910 AT6	POP	DE
F8AF 22F340	01260	LD	(40F3H), HL		F91D F1	01920 AT4	POP	AF
F8B2 EB	01270	EX	DE, HL		F91E 283F	01930	JR	Z, AT2
F8B3 7E	01280	LD	A, (HL)		F920 301C	01940	JR	NC, AT3
F8B4 23	01290	INC	HL		F922 E1	01950	POP	HL
F8B5 66	01300	LD	H, (HL)		F923 C1	01960	POP	BC
F8B6 6F	01310	LD	L, A		F924 70	01970	LD	(HL), B
F8B7 B4	01320	OR	H		F925 2B	01980	DEC	HL
F8B8 CAE7F9	01330	JP	Z, AS2		F926 71	01990	LD	(HL), C
F8BB 7E	01340	LD	A, (HL)		F927 FA1DF9	02000	JP	M, AT4
F8BC FE28	01350	CP	28H		F92A 2B	02010	DEC	HL
F8BE C20EF9	01360	JP	NZ, AS3+1		F92B C1	02020	POP	BC
F8C1 D7	01370	RST	16		F92C 70	02030	LD	(HL), B
F8C2 22DB40	01380	LD	(40DBH), HL		F92D 2B	02040	DEC	HL
F8C5 1802	01390	JR	AS4		F92E 71	02050	LD	(HL), C
F8C7 CF	01400 AS10	RST	8		F92F E21DF9	02060	JP	PD, AT4
F8C8 2C	01410	INC	L		F932 2B	02070	DEC	HL
F8C9 0E04	01420 AS4	LD	C, 04		F933 C1	02080	POP	BC
F8CB CD6319	01430	CALL	1963H		F934 70	02090	LD	(HL), B
F8CE 3E80	01440	LD	A, 80H		F935 2B	02100	DEC	HL
F8D0 32DC40	01450	LD	(40DCH), A		F936 71	02110	LD	(HL), C
F8D3 CD0D26	01460	CALL	260DH		F937 2B	02120	DEC	HL
F8D6 EB	01470	EX	DE, HL		F938 C1	02130	POP	BC

F939 70	02140	LD	(HL),B	F9AC F1	02800	POP	AF
F93A 2B	02150	DEC	HL	F9AD EB	02810	EX	DE,HL
F93B 71	02160	LD	(HL),C	F9AE 212141	02820	LD	HL,4121H
F93C 18DF	02170	JR	AT4	F9B1 C9	02830	RET	
F93E D5	02180 AT3	PUSH	DE	F9B2 D7	02840 US1	RST	16
F93F F5	02190	PUSH	AF	F9B3 010000	02850	LD	BC,0
F940 E7	02200	RST	32	F9B6 3005	02860	JR	NC,US2
F941 11D340	02210	LD	DE,40D3H	F9B8 D630	02870	SUB	30H
F944 CC882B	02220	CALL	Z,2888H	F9BA 17	02880	RLA	
F947 F1	02230	POP	AF	F9BB 4F	02890	LD	C,A
F948 C31A2B	02240	JP	281AH	F9BC D7	02900	RST	16
F94B 2AB340	02250 AT1	LD	HL,(40B3H)	F9BD EB	02910 US2	EX	DE,HL
F94E EB	02260	EX	DE,HL	F9BE 21C4F9	02920	LD	HL,STORE
F94F 2A2141	02270	LD	HL,(4121H)	F9C1 09	02930	ADD	HL,BC
F952 DF	02280	RST	24	F9C2 EB	02940	EX	DE,HL
F953 3805	02290	JR	C,AT5	F9C3 C9	02950	RET	
F955 CD432B	02300	CALL	2843H	0014	02960 STORE	DEFS	20
F958 18C2	02310	JR	AT6	F9D8 CDB2F9	02970 DE1	CALL	US1
F95A D1	02320 AT5	POP	DE	F9DB D5	02980	PUSH	DE
F95B 21D340	02330	LD	HL,40D3H	F9DC CF	02990	RST	8
F95E E5	02340	PUSH	HL	F9DD D5	03000	PUSH	DE
F95F CDF529	02350 AT2	CALL	29F5H	F9DE CD022B	03010	CALL	2B02H
F962 7E	02360	LD	A,(HL)	F9E1 E3	03020	EX	(SP),HL
F963 22B340	02370	LD	(40B3H),HL	F9E2 73	03030	LD	(HL),E
F966 E1	02380	POP	HL	F9E3 23	03040	INC	HL
F967 77	02390	LD	(HL),A	F9E4 72	03050	LD	(HL),D
F968 23	02400	INC	HL	F9E5 E1	03060	POP	HL
F969 71	02410	LD	(HL),C	F9E6 C9	03070	RET	
F96A 23	02420	INC	HL	F9E7 1E2E	03080 AS2	LD	E,2EH
F96B 70	02430	LD	(HL),B	F9E9 C3A219	03090	JP	19A2H
F96C 18AF	02440	JR	AT4	F9EC D7	03100 HEX	RST	10H
F96E FEC1	02450 DEF	CP	0C1H	F9ED FE48	03110	CP	'H'
F970 2866	02460	JR	Z,DE1	F9EF CC781D	03120	CALL	Z,1D78H
F972 CD9BF8	02470	CALL	AS1	F9F2 110000	03130	LD	DE,0
F975 CD282B	02480	CALL	2828H	F9F5 2B	03140	DEC	HL
F978 EB	02490	EX	DE,HL	F9F6 0605	03150	LD	B,5
F979 73	02500	LD	(HL),E	F9F8 23	03160 HX6	INC	HL
F97A 23	02510	INC	HL	F9F9 7E	03170	LD	A,(HL)
F97B 72	02520	LD	(HL),D	F9FA CD3E1E	03180	CALL	1E3EH
F97C EB	02530	EX	DE,HL	F9FD EB	03190	EX	DE,HL
F97D 7E	02540	LD	A,(HL)	F9FE 300A	03200	JR	NC,HX3
F97E FE28	02550	CP	28H	FA00 FE3A	03210	CP	3AH
F980 C2051F	02560	JP	NZ,1F05H	FA02 3019	03220	JR	NC,HX4
F983 D7	02570	RST	16	FA04 D630	03230	SUB	30H
F984 CD0D26	02580 DE2	CALL	260DH	FA06 3815	03240	JR	C,HX4
F987 7E	02590	LD	A,(HL)	FA08 1806	03250	JR	HX5
F988 FE29	02600	CP	29H	FA0A FE47	03260 HX3	CP	47H
F98A CA051F	02610	JP	Z,1F05H	FA0C 300F	03270	JR	NC,HX4
F98D CF	02620	RST	8	FA0E D637	03280	SUB	37H
F98E 2C	02630	INC	L	FA10 29	03290 HX5	ADD	HL,HL
F98F C384F9	02640	JP	DE2	FA11 29	03300	ADD	HL,HL
F992 F1	02650 USR	POP	AF	FA12 29	03310	ADD	HL,HL
F993 CDB2F9	02660	CALL	US1	FA13 29	03320	ADD	HL,HL
F996 D5	02670	PUSH	DE	FA14 B5	03330	OR	L
F997 CD2C25	02680	CALL	252CH	FA15 6F	03340	LD	L,A
F99A E3	02690	EX	(SP),HL	FA16 05	03350	DEC	B
F99B 4E	02700	LD	C,(HL)	FA17 CAB207	03360	JP	Z,07B2H
F99C 23	02710	INC	HL	FA1A EB	03370	EX	DE,HL
F99D 46	02720	LD	B,(HL)	FA1B 18DB	03380	JR	HX6
F99E 21E726	02730	LD	HL,26E7H	FA1D CD9A0A	03390 HX4	CALL	0A9AH
F9A1 E5	02740	PUSH	HL	FA20 EB	03400	EX	DE,HL
F9A2 C5	02750	PUSH	BC	FA21 C9	03410	RET	
F9A3 3AAF40	02760	LD	A,(40AFH)	00FF	03420 FCC	EQU	255
F9A6 F5	02770	PUSH	AF	FA22 2AA440	03430 NC0	LD	HL,(40A4H)
F9A7 FE03	02780	CP	3	FA25 7E	03440 NC1	LD	A,(HL)
F9A9 CCDA29	02790	CALL	Z,29DAH	FA26 23	03450	INC	HL

FA27 B6	03460	OR	(HL)	FA92 EB	04120	EX	DE,HL
FA28 23	03470	INC	HL	FA93 C5	04130 NU1	PUSH	BC
FA29 2B2B	03480	JR	Z,NC4	FA94 EB	04140	EX	DE,HL
FA2B 23	03490	INC	HL	FA95 22BFFB	04150	LD	(NU0),HL
FA2C 23	03500	INC	HL	FA98 EB	04160	EX	DE,HL
FA2D 7E	03510 NC2	LD	A, (HL)	FA99 CD2C1B	04170	CALL	1B2CH
FA2E A7	03520	AND	A	FA9C D1	04180	POP	DE
FA2F 23	03530	INC	HL	FA9D D5	04190	PUSH	DE
FA30 2BF3	03540	JR	Z,NC1	FA9E C5	04200	PUSH	BC
FA32 FEFF	03550	CP	FCC	FA9F CD2C1B	04210	CALL	1B2CH
FA34 20F7	03560	JR	NZ,NC2	FAA2 60	04220	LD	H,B
FA36 2B	03570	DEC	HL	FAA3 69	04230	LD	L,C
FA37 E5	03580	PUSH	HL	FAA4 D1	04240	POP	DE
FA38 23	03590 NC3	INC	HL	FAA5 DF	04250	RST	24
FA39 7E	03600	LD	A, (HL)	FAA6 EB	04260	EX	DE,HL
FA3A FEFF	03610	CP	FCC	FAA7 DA4A1E	04270	JP	C,1E4AH
FA3C 2BFA	03620	JR	Z,NC3	FAAA D1	04280	POP	DE
FA3E EB	03630	EX	DE,HL	FAAB C1	04290	POP	BC
FA3F 2AF940	03640	LD	HL, (40F9H)	FAAC F1	04300	POP	AF
FA42 A7	03650	AND	A	FAAD D5	04310	PUSH	DE
FA43 ED52	03660	SBC	HL,DE	FAAE 180E	04320	JR	NU3
FA45 4D	03670	LD	C,L	FAB0 09	04330 NU5	ADD	HL,BC
FA46 44	03680	LD	B,H	FAB1 DA4A1E	04340	JP	C,1E4AH
FA47 EB	03690	EX	DE,HL	FAB4 EB	04350	EX	DE,HL
FA48 D1	03700	POP	DE	FAB5 E5	04360	PUSH	HL
FA49 D5	03710	PUSH	DE	FAB6 21F9FF	04370	LD	HL,0FFF9H
FA4A EDB0	03720	LDIR		FAB9 DF	04380	RST	24
FA4C ED53F940	03730	LD	(40F9H),DE	FABA E1	04390	POP	HL
FA50 E1	03740	POP	HL	FABB DA4A1E	04400	JP	C,1E4AH
FA51 1BDA	03750	JR	NC2	FABE D5	04410 NU3	PUSH	DE
FA53 CDFB1A	03760 NC4	CALL	1AFBH	FABF 5E	04420	LD	E, (HL)
FA56 CDB0FC	03770	CALL	NZB	FAC0 7B	04430	LD	A,E
FA59 C3CC06	03780	JP	RTRN	FAC1 23	04440	INC	HL
FA5C CD611B	03790 NAME	CALL	1B61H	FAC2 56	04450	LD	D, (HL)
FA5F 011E1D	03800	LD	BC,1D1EH	FAC3 B2	04460	OR	D
FA62 2B	03810	DEC	HL	FAC4 EB	04470	EX	DE,HL
FA63 D7	03820	RST	16	FAC5 D1	04480	POP	DE
FA64 C5	03830	PUSH	BC	FAC6 2B07	04490	JR	Z,NU4
FA65 010A00	03840	LD	BC,000AH	FAC8 7E	04500	LD	A, (HL)
FA68 C5	03850	PUSH	BC	FAC9 23	04510	INC	HL
FA69 50	03860	LD	D,B	FACA B6	04520	OR	(HL)
FA6A 5B	03870	LD	E,B	FACB 2B	04530	DEC	HL
FA6B 2B26	03880	JR	Z,NU1	FACC EB	04540	EX	DE,HL
FA6D FE2C	03890	CP	2CH	FACD 20E1	04550	JR	NZ,NU5
FA6F 2B09	03900	JR	Z,NU2	FACF C5	04560 NU4	PUSH	BC
FA71 D5	03910	PUSH	DE	FAD0 CDF2FB	04570	CALL	NZ0
FA72 CD4F1E	03920	CALL	1E4FH	FAD3 CDFBFA	04580	CALL	NU6+1
FA75 42	03930	LD	B,D	FAD6 2ABFFB	04590	LD	HL, (NU0)
FA76 4B	03940	LD	C,E	FAD9 EB	04600	EX	DE,HL
FA77 D1	03950	POP	DE	FADA CD2C1B	04610	CALL	1B2CH
FA78 2B19	03960	JR	Z,NU1	FADD 60	04620	LD	H,B
FA7A CF	03970 NU2	RST	B	FADE 69	04630	LD	L,C
FA7B 2C	03980	INC	L	FADF C1	04640	POP	BC
FA7C CD4F1E	03990	CALL	1E4FH	FAE0 D1	04650	POP	DE
FA7F 2B12	04000	JR	Z,NU1	FAE1 D5	04660 NUB	PUSH	DE
FA81 F1	04010	POP	AF	FAE2 5E	04670	LD	E, (HL)
FA82 CF	04020	RST	B	FAE3 7B	04680	LD	A,E
FA83 2C	04030	INC	L	FAE4 23	04690	INC	HL
FA84 D5	04040	PUSH	DE	FAE5 56	04700	LD	D, (HL)
FA85 CD5A1E	04050	CALL	1E5AH	FAE6 B2	04710	OR	D
FA88 C29719	04060	JP	NZ,1997H	FAE7 2B0D	04720	JR	Z,NU7
FA8B 7A	04070	LD	A,D	FAE9 EB	04730	EX	DE,HL
FA8C B3	04080	OR	E	FAEA E3	04740	EX	(SP),HL
FA8D CA4A1E	04090	JP	Z,1E4AH	FAEB EB	04750	EX	DE,HL
FA90 EB	04100	EX	DE,HL	FAEC 23	04760	INC	HL
FA91 E3	04110	EX	(SP),HL	FAED 73	04770	LD	(HL),E

FAEE 23	04780	INC	HL	FB57 CDA72B	05440	CALL	2BA7H
FAEF 72	04790	LD	(HL),D	FB5A E1	05450	POP	HL
FAFO EB	04800	EX	DE,HL	FB5B CDAFOF	05460	CALL	OFAFH
FAF1 09	04810	ADD	HL,BC	FB5E C1	05470	POP	BC
FAF2 EB	04820	EX	DE,HL	FB5F CDE4FB	05480	CALL	NX1
FAF3 E1	04830	POP	HL	FB62 E1	05490	POP	HL
FAF4 18EB	04840	JR	NU8	FB63 E5	05500	PUSH	HL
FAF6 0122FA	04850 NU7	LD	BC,NCO	FB64 C5	05510	PUSH	BC
FAF9 C5	04860	PUSH	BC	FB65 CDA70F	05520	CALL	OFA7H
FAFA FEF6	04870 NU6	CP	OF6H	FB68 E1	05530 NX6	POP	HL
FAFC AF	04880	XOR	A	FB69 D1	05540 NW6	POP	DE
FAFD 32C1FB	04890	LD	(FB),A	FB6A 2B	05550	DEC	HL
FB00 2AA440	04900	LD	HL,(40A4H)	FB6B 18A0	05560	JR	NW7
FB03 2B	04910	DEC	HL	FB6D FE0D	05570 NW1	CP	ODH
FB04 23	04920 NU9	INC	HL	FB6F 209C	05580	JR	NZ,NW7
FB05 7E	04930	LD	A,(HL)	FB71 D5	05590	PUSH	DE
FB06 23	04940	INC	HL	FB72 CDD8FB	05600	CALL	NW4
FB07 B6	04950	OR	(HL)	FB75 EB	05610	EX	DE,HL
FB08 C8	04960	RET	Z	FB76 23	05620	INC	HL
FB09 23	04970	INC	HL	FB77 23	05630	INC	HL
FB0A 5E	04980	LD	E,(HL)	FB78 23	05640	INC	HL
FB0B 23	04990	INC	HL	FB79 4E	05650	LD	C,(HL)
FB0C 56	05000	LD	D,(HL)	FB7A 23	05660	INC	HL
FB0D D7	05010 NW7	RST	16	FB7B 46	05670	LD	B,(HL)
FB0E B7	05020 NW3	OR	A	FB7C EB	05680	EX	DE,HL
FB0F 28F3	05030	JR	Z,NU9	FB7D 60	05690	LD	H,B
FB11 4F	05040	LD	C,A	FB7E 69	05700	LD	L,C
FB12 3AC1FB	05050	LD	A,(FB)	FB7F CDD1FC	05710	CALL	NX3
FB15 B7	05060	OR	A	FB82 EB	05720	EX	DE,HL
FB16 79	05070	LD	A,C	FB83 13	05730	INC	DE
FB17 2854	05080	JR	Z,NW1	FB84 2AC2FB	05740	LD	HL,(FC)
FB19 FE9E	05090	CP	9EH	FB87 2B	05750	DEC	HL
FB1B 2022	05100	JR	NZ,NW2	FB88 2B	05760	DEC	HL
FB1D D7	05110	RST	16	FB89 2B	05770	DEC	HL
FB1E FE8D	05120	CP	8DH	FB8A 0E05	05780	LD	C,5
FB20 20EC	05130	JR	NZ,NW3	FB8C 1A	05790 NX5	LD	A,(DE)
FB22 D7	05140	RST	16	FB8D B7	05800	OR	A
FB23 FE0E	05150	CP	0EH	FB8E 2840	05810	JR	Z,NX4
FB25 20E7	05160	JR	NZ,NW3	FB90 77	05820	LD	(HL),A
FB27 D5	05170	PUSH	DE	FB91 23	05830	INC	HL
FB28 CDD8FB	05180	CALL	NW4	FB92 13	05840	INC	DE
FB2B 7A	05190	LD	A,D	FB93 0D	05850	DEC	C
FB2C B3	05200	OR	E	FB94 20F6	05860	JR	NZ,NX5
FB2D 2018	05210	JR	NZ,NW5	FB96 D1	05870 NX7	POP	DE
FB2F E5	05220	PUSH	HL	FB97 2B	05880	DEC	HL
FB30 2AC2FB	05230	LD	HL,(FC)	FB98 C30DFB	05890	JP	NW7
FB33 2B	05240	DEC	HL	FB9B 2168FB	05900 NW8	LD	HL,NX6
FB34 3620	05250	LD	(HL),20H	FB9E E5	05910	PUSH	HL
FB36 2B	05260	DEC	HL	FB9F 2AC2FB	05920	LD	HL,(FC)
FB37 3620	05270	LD	(HL),20H	FBA2 E5	05930	PUSH	HL
FB39 2B	05280	DEC	HL	FBA3 2B	05940	DEC	HL
FB3A 3630	05290	LD	(HL),30H	FBA4 70	05950	LD	(HL),B
FB3C E1	05300	POP	HL	FBA5 2B	05960	DEC	HL
FB3D 182A	05310	JR	NW6	FBA6 71	05970	LD	(HL),C
FB3F FE0E	05320 NW2	CP	0EH	FBA7 2B	05980	DEC	HL
FB41 20CA	05330	JR	NZ,NW7	FBA8 77	05990	LD	(HL),A
FB43 D5	05340	PUSH	DE	FBA9 E1	06000	POP	HL
FB44 CDD8FB	05350	CALL	NW4	FBA9 C9	06010	RET	
FB47 E5	05360 NW5	PUSH	HL	FBA9 53	06020 NW9	DEFM	'SPRUNG
FB48 CD2C1B	05370	CALL	1B2CH	FBBE 00	06030	DEFB	0
FB4B 0B	05380	DEC	BC	0002	06040 NU0	DEFS	2
FB4C 3E0D	05390	LD	A,ODH	0001	06050 FB	DEFS	1
FB4E 384B	05400	JR	C,NW8	0002	06060 FC	DEFS	2
FB50 CDF920	05410	CALL	20F9H	FBC4 B7B6	06070 FJ	DEFW	0B6B7H
FB53 21ABFB	05420	LD	HL,NW9	FBC6 9D9F	06080	DEFW	9F9DH
FB56 D5	05430	PUSH	DE	FBC8 C28E	06090 NX2	DEFW	8EC2H

FBCA B4B5	06100	DEFW	0B5B4H	FC2D 1825	06760	JR	NZ4
FBCC 8DCA	06110	DEFW	0CABDH	FC2F E1	06770 NZ6	POP	HL
FBCE 9195	06120	DEFW	9591H	FC30 18CB	06780	JR	NZ3
FBD0 36FF	06130 NX4	LD	(HL), FCC	FC32 2AA440	06790 NZ1	LD	HL, (40A4H)
FBD2 23	06140	INC	HL	FC35 EB	06800	EX	DE, HL
FBD3 0D	06150	DEC	C	FC36 62	06810 NA2	LD	H, D
FBD4 20FA	06160	JR	NZ, NX4	FC37 6B	06820	LD	L, E
FBD6 18BE	06170	JR	NX7	FC38 7E	06830	LD	A, (HL)
FBD8 23	06180 NW4	INC	HL	FC39 23	06840	INC	HL
FBD9 5E	06190	LD	E, (HL)	FC3A B6	06850	OR	(HL)
FBDA 23	06200	INC	HL	FC3B 2873	06860	JR	Z, NZ8
FBDB 56	06210	LD	D, (HL)	FC3D 23	06870	INC	HL
FBDC 23	06220	INC	HL	FC3E 23	06880	INC	HL
FBDD 22C2FB	06230	LD	(FC), HL	FC3F 23	06890	INC	HL
FBE0 2B	06240	DEC	HL	FC40 7E	06900 NA1	LD	A, (HL)
FBE1 C3781D	06250	JP	1D78H	FC41 23	06910	INC	HL
FBE4 D5	06260 NX1	PUSH	DE	FC42 FE0E	06920	CP	0EH
FBE5 C5	06270	PUSH	BC	FC44 2809	06930	JR	Z, NZ9
FBE6 3E20	06280	LD	A, 20H	FC46 B7	06940	OR	A
FBE8 1E05	06290	LD	E, 5	FC47 20F7	06950	JR	NZ, NA1
FBEA 0B	06300 NX8	DEC	BC	FC49 EB	06960	EX	DE, HL
FBEB 02	06310	LD	(BC), A	FC4A 73	06970	LD	(HL), E
FBEC 1D	06320	DEC	E	FC4B 23	06980	INC	HL
FBED 20FB	06330	JR	NZ, NX8	FC4C 72	06990	LD	(HL), D
FBEF C1	06340	POP	BC	FC4D 19E7	07000	JR	NA2
FBF0 D1	06350	POP	DE	FC4F 23	07010 NZ9	INC	HL
FBF1 C9	06360	RET	.	FC50 23	07020	INC	HL
FBF2 2AA440	06370 NZ0	LD	HL, (40A4H)	FC51 19ED	07030	JR	NA1
FBF5 7E	06380 NZ2	LD	A, (HL)	FC53 2B	07040 NA7	DEC	HL
FBF6 23	06390	INC	HL	FC54 D7	07050 NZ4	RST	16
FBF7 B6	06400	OR	(HL)	FC55 E5	07060	FUSH	HL
FBF8 2838	06410	JR	Z, NZ1	FC56 110000	07070	LD	DE, 0
FBFA 23	06420	INC	HL	FC59 300A	07080	JR	NC, NZ11
FBFB 23	06430	INC	HL	FC5B CD5A1E	07090	CALL	1E5AH
FBFC 23	06440	INC	HL	FC5E 2B	07100 NZ10	DEC	HL
FBFD 7E	06450 NZ3	LD	A, (HL)	FC5F 7E	07110	LD	A, (HL)
FBFE B7	06460	OR	A	FC60 FE20	07120	CP	' '
FBFF 23	06470	INC	HL	FC62 28FA	07130	JR	Z, NZ10
FC00 28F3	06480	JR	Z, NZ2	FC64 23	07140	INC	HL
FC02 F2FDFB	06490	JP	P, NZ3	FC65 C1	07150 NZ11	POP	BC
FC05 2B	06500	DEC	HL	FC66 7D	07160	LD	A, L
FC06 11C4FB	06510	LD	DE, FJ	FC67 91	07170	SUB	C
FC09 0E0C	06520	LD	C, 0CH	FC68 2824	07180	JR	Z, NA3
FC0B 1A	06530 NZ5	LD	A, (DE)	FC6A D5	07190	PUSH	DE, NA3
FC0C BE	06540	CP	(HL)	FC6B 5F	07200	LD	E, A
FC0D 2845	06550	JR	Z, NZ4	FC6C 3E02	07210	LD	A, 2
FC0F 13	06560	INC	DE	FC6E F5	07220	FUSH	AF
FC10 0D	06570	DEC	C	FC6F 3E05	07230	LD	A, 5
FC11 20FB	06580	JR	NZ, NZ5	FC71 93	07240	SUB	E
FC13 7E	06590	LD	A, (HL)	FC72 3007	07250	JR	NC, NA4
FC14 FE89	06600	CP	B9H	FC74 2F	07260	CPL	
FC16 23	06610	INC	HL	FC75 3C	07270	INC	A
FC17 20E4	06620	JR	NZ, NZ3	FC76 D1	07280	POP	DE
FC19 E5	06630	FUSH	HL	FC77 C602	07290	ADD	A, 2
FC1A 2B	06640	DEC	HL	FC79 F5	07300	PUSH	AF
FC1B D7	06650	RST	16	FC7A AF	07310	XOR	A
FC1C FE23	06660	CP	23H	FC7B C5	07320 NA4	FUSH	BC
FC1E 200F	06670	JR	NZ, NZ6	FC7C C4BAFC	07330	CALL	NZ, NA5
FC20 D7	06680	RST	16	FC7F E1	07340	POP	HL
FC21 FEF3	06690	CP	0F3H	FC80 360E	07350	LD	(HL), 0EH
FC23 200A	06700	JR	NZ, NZ6	FC82 23	07360	INC	HL
FC25 D7	06710 NZ7	RST	16	FC83 C1	07370	POP	BC
FC26 2807	06720	JR	Z, NZ6	FC84 D1	07380	POP	DE
FC28 FE2C	06730	CP	2CH	FC85 73	07390	LD	(HL), E
FC2A 20F9	06740	JR	NZ, NZ7	FC86 23	07400	INC	HL
FC2C F1	06750	POP	AF	FC87 72	07410	LD	(HL), D

FC88 23	07420 NA6	INC	HL	FD15 ED5BF940	08080 ME1	LD	DE, (40F9H)
FC89 3620	07430	LD	(HL), 20H	FD19 1B	08090	DEC	DE
FC8B 10FB	07440	DJNZ	NA6	FD1A D5	08100	PUSH	DE
FC8D 23	07450 NAB	INC	HL	FD1B C1	08110	POP	BC
FC8E 2B	07460 NA3	DEC	HL	FD1C 1B	08120	DEC	DE
FC8F D7	07470	RST	16	FD1D 1B	08130	DEC	DE
FC90 3C	07480	INC	A	FD1E 1B	08140	DEC	DE
FC91 3D	07490	DEC	A	FD1F 1A	08150	LD	A, (DE)
FC92 38BF	07500	JR	C, NA7	FD20 B7	08160	OR	A
FC94 23	07510	INC	HL	FD21 2005	08170	JR	NZ, ME2
FC95 CAF5FB	07520	JP	Z, NZ2	FD23 ED43F940	08180	LD	(40F9H), BC
FC98 2B	07530	DEC	HL	FD27 1B	08190	DEC	DE
FC99 FE20	07540	CF	20H	FD28 13	08200 ME2	INC	DE
FC9B 28F0	07550	JR	Z, NAB	FD29 13	08210	INC	DE
FC9D FE2C	07560	CF	2CH	FD2A ED53A440	08220	LD	(40A4H), DE
FC9F 28EC	07570	JR	Z, NAB	FD2E 3E3A	08230	LD	A, 42
FCA1 FECE	07580	CF	0CEH	FD30 32F9FC	08240	LD	(GI2), A
FCA3 28E8	07590	JR	Z, NAB	FD33 C3CC06	08250	JP	06CCH
FCA5 FED4	07600	CF	0D4H	FD36 3AF9FC	08260 RSET	LD	A, (GI2)
FCA7 3804	07610	JR	C, NA9	FD39 B7	08270	OR	A
FCA9 FED7	07620	CF	0D7H	FD3A 290B	08280	JR	Z, RS1
FCAB 38E0	07630	JR	C, NAB	FD3C 2A4AFD	08290	LD	HL, (BASBEG)
FCAD C3FDFB	07640 NA9	JP	NZ3	FD3F 22A440	08300	LD	(40A4H), HL
FCB0 2AF940	07650 NZB	LD	HL, (40F9H)	FD42 3E00	08310	LD	A, 0
FCB3 22FB40	07660	LD	(40FBH), HL	FD44 32F9FC	08320	LD	(GI2), A
FCB6 22FD40	07670	LD	(40FDH), HL	FD47 C3CC06	08330 RS1	JP	06CCH
FCB9 C9	07680	RET		0002	08340 BASBEG	DEFS	2
FCBA E5	07690 NA5	PUSH	HL		08350 ;		
FCBB 2AF940	07700	LD	HL, (40F9H)		08360 ;		
FCBE EB	07710	EX	DE, HL		08370 ;		
FCBF 2600	07720	LD	H, 0		08380 ;		
FCC1 6F	07730	LD	L, A		08390 ;		
FCC2 19	07740	ADD	HL, DE		08400 ;***** MID# ALS STATEMENT		
FCC3 22F940	07750	LD	(40F9H), HL		08410 ;		
FCC6 44	07760	LD	B, H		08420 ;		
FCC7 4D	07770	LD	C, L	FD4C D7	08430 MID	RST	10H
FCC8 E1	07780	POP	HL	FD4D CF	08440	RST	08H
FCC9 1A	07790 NB1	LD	A, (DE)	FD4E 2B	08450	DEFM	? (?)
FCCA 02	07800	LD	(BC), A	FD4F CDBCFD	08460	CALL	MID6
FCCB DF	07810	RST	24	FD52 E5	08470	PUSH	HL
FCCC CB	07820	RET	Z	FD53 D5	08480	PUSH	DE
FCCD 1B	07830	DEC	DE	FD54 EB	08490	EX	DE, HL
FCCE 0B	07840	DEC	BC	FD55 23	08500	INC	HL
FCCF 18FB	07850	JR	NB1	FD56 5E	08510	LD	E, (HL)
FCD1 CD9A0A	07860 NX3	CALL	0A9AH	FD57 23	08520	INC	HL
FCD4 AF	07870	XOR	A	FD58 56	08530	LD	D, (HL)
FCD5 CD3410	07880	CALL	1034H	FD59 2AA040	08540	LD	HL, (40A0H)
FCD8 B6	07890	OR	(HL)	FD5C DF	08550	RST	18H
FCD9 C3D90F	07900	JP	0FD9H	FD5D E1	08560	POP	HL
FCDC CDF801	07910 READY	CALL	01F8H	FD5E E5	08570	PUSH	HL
FCDF CDF920	07920	CALL	20F9H	FD5F D4B4FD	08580	CALL	NC, MID5
FCE2 21ECFC	07930	LD	HL, GI1	FD62 E1	08590	POP	HL
FCE5 CDA728	07940	CALL	28A7H	FD63 E3	08600	EX	(SP), HL
FCEB E1	07950	POP	HL	FD64 CF	08610	RST	08H
FCE9 C32B1A	07960	JP	1A2BH	FD65 2C	08620	INC	L
FCEC 4E	07970 GI1	DEFM	'NEWBAS READY'	CD1C2B	08630	CALL	2B1CH
FCFB 0D	07980	DEFB	0DH	FD69 B7	08640	OR	A
FCF9 00	07990 GI2	DEFB	0	FD6A CA4A1E	08650	JP	Z, 1E4AH
FCFA 20	08000	DEFM	'MERGE MODE *'	F5	08660	PUSH	AF
FD07 0D	08010	DEFB	0DH	FD6E 7E	08670	LD	A, (HL)
FD08 00	08020	DEFB	0	FD6F FE2C	08680	CP	2CH
FD09 3AF9FC	08030 MERGE	LD	A, (GI2)	FD71 3EFF	08690	LD	A, OFFH
FD0C B7	08040	OR	A	FD73 2004	08700	JR	NZ, MID1
FD0D 2006	08050	JR	NZ, ME1	FD75 D7	08710	RST	10H
FD0F 2AA440	08060	LD	HL, (40A4H)	FD76 CD1C2B	08720	CALL	2B1CH
FD12 22AFD	08070	LD	(BASBEG), HL	FD79 F5	08730 MID1	PUSH	AF

FD7A	CF	08740	RST	08H	
FD7B	29	08750	DEFM	') '	
FD7C	CF	08760	RST	08H	
FD7D	D5	08770	DEFB	0D5H	; "=" TOKEN
FD7E	CD3723	08780	CALL	2337H	
FD81	E5	08790	PUSH	HL	
FD82	CDD729	08800	CALL	29D7H	
FD85	D1	08810	POP	DE	
FD86	C1	08820	POP	BC	
FD87	F1	08830	POP	AF	
FD88	4F	08840	LD	C, A	
FD89	EB	08850	EX	DE, HL	
FD8A	E3	08860	EX	(SP), HL	
FD8B	D5	08870	PUSH	DE	
FD8C	7E	08880	LD	A, (HL)	
FD8D	23	08890	INC	HL	
FD8E	5E	08900	LD	E, (HL)	
FD8F	23	08910	INC	HL	
FD90	56	08920	LD	D, (HL)	
FD91	69	08930	LD	L, C	
FD92	2600	08940	LD	H, 0	
FD94	19	08950	ADD	HL, DE	
FD95	2B	08960	DEC	HL	
FD96	91	08970	SUB	C	
FD97	DA4A1E	08980	JP	C, 1E4AH	
FD9A	3C	08990	INC	A	
FD9B	B8	09000	CP	B	
FD9C	3001	09010	JR	NC, MID2	
FD9E	47	09020	LD	B, A	
FD9F	E3	09030 MID2	EX	(SP), HL	
FDA0	7E	09040	LD	A, (HL)	
FDA1	23	09050	INC	HL	
FDA2	5E	09060	LD	E, (HL)	
FDA3	23	09070	INC	HL	
FDA4	56	09080	LD	D, (HL)	
FDA5	E1	09090	POP	HL	
FDA6	B8	09100	CP	B	
FDA7	3801	09110	JR	C, MID3	
FDA9	78	09120	LD	A, B	
FDAA	CDAFFD	09130 MID3	CALL	MID4	
FDAD	E1	09140	POP	HL	
FDAE	C9	09150	RET		
		09160 ;			
		09170 ;	UNTERPROGRAMME		
		09180 ;			
FDAF	B7	09190 MID4	OR	A	
FDB0	C2D609	09200	JP	NZ, 09D6H	
FDB3	C9	09210	RET		
FDB4	E5	09220 MID5	PUSH	HL	
FDB5	CD4328	09230	CALL	2843H	
FDB8	E1	09240	POP	HL	
FDB9	C3D309	09250	JP	09D3H	
FDBC	CD0D26	09260 MID6	CALL	260DH	
FDBF	C3F40A	09270	JP	0AF4H	
		09280 ;			
		09290 ;			
		09300 ;			
		09310 ;	***** LINE INPUT BEFEHL *****		
		09320 ;			
		09330 ;			
FDC2	CF	09340 LINE	RST	08H	
FDC3	B9	09350	DEFB	89H	; "INPUT"-TOKEN
FDC4	CD2828	09360	CALL	2828H	
FDC7	7E	09370	LD	A, (HL)	
FDC8	FE23	09380	CP	' # '	
FDCA	CA2D01	09390	JP	Z, 012DH	; LEVEL 3 ERROR

FE31	80	10060	ADD	A, B
FE32	47	10070	LD	B, A
FE33	3017	10080	JR	NC, INSTR4
FE35	04	10090	INC	B
FE36	E5	10100	INSTR2	PUSH HL
FE37	D5	10110	PUSH	DE
FE38	C5	10120	PUSH	BC
FE39	0C	10130	INC	C
FE3A	0D	10140	INSTR3	DEC C
FE3B	2813	10150	JR	Z, INSTR5
FE3D	1A	10160	LD	A, (DE)
FE3E	BE	10170	CP	(HL)
FE3F	13	10180	INC	DE
FE40	23	10190	INC	HL
FE41	28F7	10200	JR	Z, INSTR3
FE43	C1	10210	POP	BC
FE44	D1	10220	POP	DE
FE45	E1	10230	POP	HL
FE46	F1	10240	POP	AF
FE47	3C	10250	INC	A
FE48	23	10260	INC	HL
FE49	F5	10270	PUSH	AF
FE4A	10EA	10280	DJNZ	INSTR2
FE4C	F1	10290	INSTR4	POP AF
FE4D	AF	10300	XOR	A
FE4E	1804	10310	JR	INSTR6
FE50	C1	10320	INSTR5	POP BC
FE51	D1	10330	POP	DE
FE52	E1	10340	POP	HL
FE53	F1	10350	POP	AF
FE54	CDFB27	10360	INSTR6	CALL 27FBH
FE57	E1	10370	INSTR7	POP HL
FE58	C9	10380	RET	
FE59	CF	10390	INSTR8	RST 08H
FE5A	2C	10400	DEFM	' , '
FE5B	C33723	10410	JP	2337H
		10420	;	
		10430	;	
		10450	;	***** SUCHT STRINGS IN BASIC PROGRAMMEN *****
		10460	;	***** AUFRUF DURCH L3 BEFEHL 'GET' *****
032A		10470	SHOCHR	EQU 32AH
0361		10480	GETSTR	EQU 361H
06CC		10490	RTRN	EQU 6CCH
0FAF		10500	SHOLIN	EQU 0FAFH
1BC0		10510	CODE	EQU 1BC0H
28A7		10520	SHOMSG	EQU 28A7H
40A4		10530	BSTP	EQU 40A4H
40A6		10540	CURSP	EQU 40A6H
417F		10550	GETP	EQU 417FH
		10560	;	
		10570	;	
		10580	;	
FE5E	3E01	10590	GET	LD A, 1
FE60	324AFF	10600	LD	(NF), A
FE63	210DFF	10610	LD	HL, STRM
FE66	CDA72B	10620	CALL	SHOMSG
FE69	CD6103	10630	STRIN	CALL GETSTR
FE6C	23	10640	INC	HL
FE6D	7E	10650	LD	A, (HL)
FE6E	B7	10660	OR	A
FE6F	CACC06	10670	JP	Z, RTRN
FE72	FE40	10680	CP	' § '
FE74	200F	10690	JR	NZ, ST2
FE76	23	10700	INC	HL
FE77	7E	10710	LD	A, (HL)
FE78	FE40	10720	CP	' § '

FDCD	CDCD21	09400	CALL	21CDH	FE7A	2008	10730	JR	NZ, ST1	
FDD0	CD0D26	09410	CALL	260DH	FE7C	23	10740	INC	HL	
FDD3	CDF40A	09420	CALL	0AF4H	FE7D	7E	10750	LD	A, (HL)	
FDD6	E5	09430	PUSH	HL	FE7E	B7	10760	OR	A,	
FDD7	D5	09440	PUSH	DE	FE7F	CACC06	10770	JP	Z, RTRN	
FDD8	CD6103	09450	CALL	0361H	FE82	1805	10780	JR	ST3	
FDD8	380A	09460	JR	C, LINE1	FE84	2B	10790	ST1	DEC	
FDDD	0600	09470	LD	B, 0	FE85	CDC01B	10800	ST2	CALL	
FDDF	CD6828	09480	CALL	2868H	FE88	23	10810	INC	HL	
FDE2	E1	09490	POP	HL	FE89	2246FF	10820	ST3	LD	
FDE3	AF	09500	XOR	A	FE8C	2AA440	10830	LD	HL, (BSTP)	
FDE4	C3331F	09510	JP	1F33H	FE8F	7E	10840	LD	A, (HL)	
FDE7	F1	09520	LINE1	POP	AF	FE90	B7	10850	OR	
FDE8	F1	09530	POP	AF	FE91	2006	10860	JR	NZ, SU	
FDE9	C3BE1D	09540	JP	1DBEH	FE93	23	10870	INC	HL	
		09550	;		FE94	7E	10880	LD	A, (HL)	
		09560	;		FE95	B7	10890	OR	A	
		09570	;		FE96	285C	10900	JR	Z, NOPGM	
		09580	;	***** INSTR BEFEHL *****	FE98	2B	10910	DEC	HL	
		09590	;		FE99	2B	10920	SU	DEC	
FDEC	D7	09600	INSTR	RST	10H	FE9A	23	10930	GETPTR	INC
FDED	CD3523	09610	CALL	2335H	FE9B	5E	10940	LD	E, (HL)	
FDF0	E7	09620	RST	20H	FE9C	23	10950	INC	HL	
FDF1	3E01	09630	LD	A, 01H	FE9D	56	10960	LD	D, (HL)	
FDF3	C41F2B	09640	CALL	NZ, 2B1FH	FE9E	AF	10970	XOR	A	
FDF6	B7	09650	OR	A	FE9F	B3	10980	OR	E	
FDF7	CA4A1E	09660	JP	Z, 1E4AH	FEA0	C2A7FE	10990	JP	NZ, GETLIN	
F DFA	F5	09670	PUSH	AF	FEA3	B2	11000	OR	D	
FDFB	E7	09680	RST	20H	FEA4	CAFDFE	11010	JP	Z, ENDBAS	
FDFC	C459FE	09690	CALL	NZ, INSTR8	FEA7	23	11020	GETLIN	INC	
FDFE	CD40A	09700	CALL	0AF4H	FEA8	5E	11030	LD	E, (HL)	
FE02	ED4B2141	09710	LD	BC, (4121H)	FEA9	23	11040	INC	HL	
FE06	C5	09720	PUSH	BC	FEAA	56	11050	LD	D, (HL)	
FE07	CD59FE	09730	CALL	INSTR8	FEAB	ED5348FF	11060	LD	(LIN), DE	
FE0A	CF	09740	RST	08H	FEAF	23	11070	SCAN	INC	
FE0B	29	09750	DEFM	')'	FEB0	0600	11080	LD	B, 0	
FE0C	E3	09760	EX	(SP), HL	FEB2	ED5B46FF	11090	LD	DE, (STR)	
FE0D	E5	09770	PUSH	HL	FEB6	1A	11100	SC1	LD	
FE0E	CDD729	09780	CALL	29D7H	FEB7	B7	11110	OR	A	
FE11	E3	09790	EX	(SP), HL	FEB8	C2DCFE	11120	JP	NZ, SC3	
FE12	CDDD29	09800	CALL	29DDH	FEBB	D9	11130	EXX		
FE15	D1	09810	POP	DE	FEBC	AF	11140	XOR	A	
FE16	C1	09820	POP	BC	FEBD	324AFF	11150	LD	(NF), A	
FE17	F1	09830	POP	AF	FEC0	3AA640	11160	LD	A, (CURSP)	
FE18	C5	09840	PUSH	BC	FEC3	FE39	11170	CP	57	
FE19	F5	09850	PUSH	AF	FEC5	FACDFE	11180	JP	M, SC2	
FE1A	D5	09860	PUSH	DE	FEC8	3E0D	11190	LD	A, 13	
FE1B	3D	09870	DEC	A	FECA	CD2A03	11200	CALL	SHOCHR	
FE1C	46	09880	LD	B, (HL)	FECD	2A48FF	11210	SC2	LD	
FE1D	23	09890	INC	HL	FED0	CDAF0F	11220	CALL	SHQLIN	
FE1E	5E	09900	LD	E, (HL)	FED3	3E20	11230	LD	A, 20H	
FE1F	23	09910	INC	HL	FED5	CD2A03	11240	CALL	SHOCHR	
FE20	56	09920	LD	D, (HL)	FED8	D9	11250	EXX		
FE21	6F	09930	LD	L, A	FED9	C3ECFE	11260	JP	SC4	
FE22	2600	09940	LD	H, 0	FEDC	4E	11270	SC3	LD	
FE24	19	09950	ADD	HL, DE	FEDD	0C	11280	INC	C	
FE25	E3	09960	EX	(SP), HL	FEDE	0D	11290	DEC	C	
FE26	4E	09970	LD	C, (HL)	FEDF	CA9AFE	11300	JP	Z, GETPTR	
FE27	23	09980	INC	HL	FEE2	B9	11310	CP	C	
FE28	5E	09990	LD	E, (HL)	FEE3	C2ECFE	11320	JP	NZ, SC4	
FE29	23	10000	INC	HL	FEE6	04	11330	INC	B	
FE2A	56	10010	LD	D, (HL)	FEE7	23	11340	INC	HL	
FE2B	E1	10020	POP	HL	FEE8	13	11350	INC	DE	
FE2C	81	10030	ADD	A, C	FEE9	C3B6FE	11360	JP	SC1	
FE2D	381D	10040	JR	C, INSTR4	FEEC	48	11370	SC4	LD	
FE2F	ED44	10050	INSTR1	NEG	FEED	AF	11380	XOR	A	

```
FEEE 47      11390      LD      B,A
FEFF ED42    11400      SBC     HL,BC
FEF1 C3AFFE  11410      JP      SCAN
FEF4 2126FF  11420 NOPGM  LD      HL,NPGM
FEF7 CDA72B  11430      CALL   SHOMSG
FEFA C3CC06  11440      JP      RTRN
FEFD 3A4AFF  11450 ENDBAS LD      A,(NF)
FF00 B7      11460      OR     A
FF01 CACC06  11470      JP      Z,RTRN
FF04 2117FF  11480      LD      HL,NFM
FF07 CDA72B  11490      CALL   SHOMSG
FF0A C3CC06  11500      JP      RTRN
FF0D 1D1E    11510 STRM  DEFW   1E1DH
FF0F 53      11520      DEFM   'STRING='
FF16 00      11530      DEFB   0
FF17 4E      11540 NFM  DEFM   'NICHT GEFUNDEN'
FF25 00      11550      DEFB   0
FF26 4B      11560 NPGM  DEFM   'KEIN BASIC PROGRAMM IM SPEICHER'
FF45 00      11570      DEFB   0
FF46 0000    11580 STR  DEFW   0
FF48 0000    11590 LIN  DEFW   0
FF4A 01      11600 NF   DEFB   1
FF4B 00      11610      DEFB   0 ;NULL VOR BASIC PROGRAM
11620 ;
11630 ;***** ROUTING ERWEITERUNG FUER NEWBAS *****
11650 ;
06CC 11660 BASIC EQU    06CCH
1997 11670 SNERR EQU    1997H ;SYNTAX ERROR ENTRY POINT
40A2 11680 AKTZNR EQU    40A2H ;ZEIGER AUF AKTUELLE ZNR.
4173 11690 CMD EQU    4173H ;ADRESSE DES "CMD" VECTOR
401D 11700 DODCB EQU    401DH ;ADRESSE DES VIDEO-DCB
4025 11710 PRDCB EQU    4025H ;ADRESSE DES PRINTER-DCB
00B2 11720 PRINT EQU    0B2H ;PRINT-TOKEN
00AF 11730 LPRINT EQU    0AFH ;LPRINT-TOKEN
00BD 11740 TO EQU    0BDH ;TO-TOKEN
11750 ;
11760 ;
11770 ; ROUTING-ROUTINE
11780 ;
FF4C FEB2    11790 DIRECT CP     PRINT ;PRINT-TOKEN?
FF4E 111E40  11800 LD     DE,DODCB+1 ;DE ZEIGT AUF VIDEO-DCB
FF51 280A    11810 JR     Z,DIR1
FF53 FEAF    11820 CP     LPRINT ;LPRINT-TOKEN?
FF55 112640  11830 LD     DE,PRDCB+1 ;DE ZEIGT AUF PRINT-DCB
FF58 2803    11840 JR     Z,DIR1
FF5A C39719  11850 JP     SNERR ;SONST SYNTAX-ERROR
11860 ;
FF5D D7      11870 DIR1 RST    16 ;NAECHSTES TOKEN HOLEN
FF5E CF      11880 RST    8 ;TESTEN OB ES DAS
FF5F BD      11890 DEFB   TO ;TO-TOKEN IST
11900 ;
FF60 FE56    11910 CP     'V' ;SOLL AUSGABE AUF VIDEO?
FF62 0185FF  11920 LD     BC,DODVR
FF65 280F    11930 JR     Z,DIR2
11940 ;
FF67 FE50    11950 CP     'P' ;SOLL AUSGABE AUF DRUCKER?
FF69 018FFF  11960 LD     BC,PRDVR
FF6C 2808    11970 JR     Z,DIR2
11980 ;
FF6E FE42    11990 CP     'B' ;SOLL AUSGABE AUF BEIDE?
FF70 017DFF  12000 LD     BC,BOTH
12010 ;
FF73 C29719  12020 JP     NZ,SNERR ;SONST FEHLER MELDEN
12030 ;
FF76 D7      12040 DIR2 RST    16 ;ZEIGER AKTUALISIEREN
FF77 EB      12050 EX     DE,HL ;HL ZEIGT AUF DCB
```

```

FF78 71      12060      LD      (HL),C      ;DCB MIT TREIBERADRESSE
FF79 23      12070      INC     HL          ;LADEN
FF7A 70      12080      LD      (HL),B
FF7B EB      12090      EX      DE,HL
FF7C C9      12100      RET
                12110 ;
                12120 ; NEUE TREIBER
                12130 ;
FF7D CDB2FF  12140 BOTH   CALL   DITEST      ;TEST AUF DIREKTMODUS
FF80 F5      12150      PUSH   AF
FF81 CDBFFF  12160      CALL   PRDVR      ;AUSGABE AUF DRUCKER
FF84 F1      12170      POP    AF
                12180 ;
FF85 CDB2FF  12190 DODVR  CALL   DITEST      ;TEST AUF DIREKTMODUS
FF88 DD211D40 12200      LD      IX,DODCB   ;IX ZEIGT AUF VIDEO-DCB
FF8C C30000  12210 DODVR1  JP     0           ;ADR. DES AKTUELLEN VIDEO-TREIBER WIRD DORT
BEI DER INITIALISIERUNG EINGESETZT
                12220 ;
FF8F CDB2FF  12230 PRDVR  CALL   DITEST      ;TEST AUF DIREKTMODUS
FF92 F5      12240      PUSH   AF          ;FLAGS RETTEN
FF93 DDE5    12250      PUSH   IX
FF95 E1      12260      POP    HL          ;ADRESSE DES AUF RUFENDEN DCB NACH HL BRINGE
N
FF96 7D      12270      LD      A,L
FF97 FE25    12280      CP     25H         ;KAM AUFRUF VOM PRINT-DCB
FF99 280F    12290      JR     Z,PR1       ;DANN AUSGABE UEBER NORMALEN DRUCKERTREIBER
                12300 ;
FF9B 79      12310      LD      A,C
FF9C FE20    12320      CP     20H         ;TEST AUF KONTROLL-ZEICHEN
FF9E 300A    12330      JR     NC,PR1      ;FALLS KEIN SOLCHES, NORMALEN DRUCKERTREIBE
ANSPRINGEN
FFA0 FE0D    12340      CP     0DH         ;CR?
FFA2 2806    12350      JR     Z,PR1       ;DANN NORMALER TREIBER
FFA4 FE0A    12360      CP     0AH         ;LF?
FFA6 2802    12370      JR     Z,PR1       ;DANN NORMALER TREIBER
FFA8 F1      12380      POP    AF          ;SONST KEINE AUSGABE AUF DRUCKER
FFA9 C9      12390      RET
                12400 ;
FFAA F1      12410 PR1   POP    AF
FFAB DD212540 12420      LD      IX,PRDCB   ;IX MUSS AUF PRINT-DCB ZEIGEN
FFAF C30000  12430 PRDVR1  JP     0           ;HIER WIRD BEI DER INITIALISIERUNG DIE ADRE
SE DES AKTUELLEN DRUCKERTREIBER EINGEFUEGT
                12440 ;
                12450 ; SUBROUTINE ZUM TEST AUF DIREKTMODUS
                12460 ;
FFB2 F5      12470 DITEST  PUSH   AF
FFB3 2AA240  12480      LD      HL,(AKTZNR) ;AKTUELLE ZNR NACH HL
FFB6 23      12490      INC     HL
FFB7 7C      12500      LD      A,H
FFB8 B5      12510      OR     L
FFB9 2010    12520      JR     NZ,DOROUT   ;FALLS NICHT IM DIREKTMODUS, DANN ROUTING
                12530 ;
FFBB DDE5    12540      PUSH   IX
FFBD E1      12550      POP    HL          ;ADRESSE DES AUFRUFENDEN DCB IN HL
FFBE 7D      12560      LD      A,L
FFBF FE25    12570      CP     25H         ;IST ES PRINT-DCB?
FFC1 2804    12580      JR     Z,NOROUT    ;DANN AUCH AUSGABE AUF DRUCKER (KEIN ROUTIN
)
                12590 ;
FFC3 F1      12600      POP    AF
FFC4 E1      12610      POP    HL          ;STACK JUSTIEREN
FFC5 18C5    12620      JR     DODVR1     ;SONST AUSGABE AUF VIDEO
                12630 ;
FFC7 F1      12640 NOROUT  POP    AF
FFC8 E1      12650      POP    HL          ;STACK JUSTIEREN
FFC9 18E4    12660      JR     PRDVR1     ;ZUM AKTUELLEN DRUCKERTREIBER

```



```
12670 ;
FFCB F1 12680 DOROUT POP AF
FFCC C9 12690 RET ;WEITER MIT VORGESEHENEM ROUTING
12700 ;
12710 ; ** RESTORE N **, 18.2.80 (V 1.1)
12720 ; BEARBEITET FUER EDTASM, 4.9.81
12730 ;
12740 ; BASIC-UNTERPROGRAMME:
1D78 12750 CHRGET EQU 1D78H ; ZEICHENHOLROUTINE
2B01 12760 INTEXP EQU 2B01H ; ZEILENUMMER ERMITTELN
1B2C 12770 SLINE EQU 1B2CH ; ZEILE IN RAM SUCHEN
06CC 12780 WSTART EQU 06CCH ; ZURUECK ZUM BASIC
1D1E 12790 EXEC EQU 1D1EH ; BASICPROGRAMM AUSFUEHREN
12800 ;
12810 ; BASIC-ZEIGER:
40FF 12820 DATAZ EQU 40FFH ; DATAZEIGER
4004 12830 RST10V EQU 4004H ; RST 10H-VEKTOR
12840 ;
12850 ;
12860 ;
12870 ; RESTORE N
12880 ;
FFCD E3 12890 RESTN EX (SP),HL ; AUFRUF VON 1DXXH?
FFCE 7C 12900 LD A,H
FFCF FE1D 12910 CP 1DH
FFD1 2810 12920 JR Z,RESTN1 ; NEIN, RST 10H AUSF.
FFD3 FE20 12930 CP 20H
FFD5 2005 12940 JR NZ,NINT
FFD7 7D 12950 LD A,L
FFD8 FE50 12960 CP 50H
FFDA 2807 12970 JR Z,RESTN1
FFDC E3 12980 NINT EX (SP),HL
FFDD C30000 12990 CHV10 JP 0
FFE0 E1 13000 NINT1 POP HL
FFE1 18FA 13010 JR CHV10
FFE3 E3 13020 RESTN1 EX (SP),HL
FFE4 E5 13030 PUSH HL
FFE5 D7 13040 RST 10H
FFE6 FE90 13050 CP 90H ; =RESTORE?
FFE8 20F6 13060 JR NZ,NINT1 ; NEIN, RST 10H(OHNE INC HL)
FFEA C1 13070 POP BC
FFEB D7 13080 RST 10H
FFEC CA911D 13090 JP Z,1D91H
FFEF CD022B 13100 CALL INTEXP+1 ; ZNR DECODIEREN
FFF2 E5 13110 PUSH HL ; POINTER RETTEN
FFF3 CD2C1B 13120 CALL SLINE ; ZEILENADRESSE
FFF6 0B 13130 DEC BC
FFF7 ED43FF40 13140 LD (DATAZ),BC ; NACH DATAZEIGER
FFFB E1 13150 POP HL ; POINTER ZURUECK
FFFC C1 13160 POP BC ; RUECKSPRUNGADR.
FFFD C31E1D 13170 JF EXEC ; NAECHSTE ; ZEILENADRESSE
FB37 13180 END NEWBAS
00000 TOTAL ERRORS
16015 TEXT AREA BYTES LEFT
```

6.3 Double-Precision-Funktionen DOUBLE

- 114 -

```

4003      00010 RST10V EQU 4003H
4009      00020 RST20V EQU 4009H
FAFF      00030          ORG 0FAFFH
FAFF 2A0440 00040 INIT   LD HL, (RST10V+1)
FB02 2229FB 00050          LD (RETRST+2),HL
FB05 223EFB 00060          LD (RST101+2),HL
FB08 211EFB 00070          LD HL,DOUBLE
FB0B 220440 00080          LD (RST10V+1),HL
FB0E 2A0A40 00090          LD HL, (RST20V+1)
FB11 2267FB 00100          LD (RST1+2),HL
FB14 2273FB 00110          LD (RST2+1),HL
FB17 215CFB 00120          LD HL,RST20N
FB1A 220A40 00130          LD (RST20V+1),HL
FB1D C9      00140          RET
          00150          ; ** DOUBLE PRECISION FUNKTIONEN **, 9.11.80
          00160          ; BEARBEITET FUER EDTASM, 12.4.81
OC77      00170 ADD     EQU OC77H      ;X=X+Y
OC70      00180 SUB     EQU OC70H      ;X=X-Y
0DA1      00190 MLT     EQU 0DA1H      ;X=X*Y
ODE5      00200 DIV     EQU ODE5H      ;X=X/Y
0955      00210 TEST    EQU 0955H
0AB1      00220 CSNG    EQU 0AB1H      ;X=CSNG(X)
0ADB      00230 CDBL    EQU 0ADBH      ;X=CDBL(X)
0AE3      00240 CDBL1   EQU 0AE3H      ;X=CDBL(X) (X SINGLE)
0B59      00250 INT     EQU 0B59H      ;X=INT(X)
0B7B      00260 INT1    EQU 0B7BH      ;X=INT(X) (OHNE NORMALISIERUNG)
0CD8      00270 NORM    EQU 0CD8H      ;NORMALISIERUNG
09A4      00280 SPUSHX  EQU 09A4H      ;X AUF STACK
13E2      00290 ENEG    EQU 13E2H      ;ERGEBNIS * (-1)
0982      00300 NEG     EQU 0982H      ;X=-X
0778      00310 NULL    EQU 0778H      ;X=0
0964      00320 FLOAT   EQU 0964H      ;X=CSNG(A)
0A49      00330 CMP     EQU 0A49H      ;VERGLEICH: X-(DE)
0A4F      00340 DCFXY   EQU 0A4FH      ;VURGLEICH: X-Y
13E7      00350 SQR     EQU 13E7H      ;X=SQR(X) (SINGLE)
13F2      00360 SPOWER  EQU 13F2H      ;X=STACK ** X
252C      00370 ARG     EQU 252CH      ;X=(AUSDRUCK)
411D      00380 X       EQU 411DH      ;X-REGISTER
4127      00390 Y       EQU 4127H      ;Y-REGISTER
1997      00400 SNERR   EQU 1997H      ;SYNTAX ERROR
1E4A      00410 FCERR   EQU 1E4AH      ;ILLEGAL FC
07B2      00420 OVERR   EQU 07B2H      ;OVERFLOW
199A      00430 DOERR   EQU 199AH      ;/0-ERROR
          00440          ;INITIALISIERUNG
FB1E E3      00450 DOUBLE EX (SP),HL
FB1F 7D      00460          LD A,L
FB20 FEA0    00470          CP 0A0H
FB22 2003    00480          JR NZ,RETRST
FB24 7C      00490          LD A,H
FB25 FE24    00500          CP 24H
FB27 E3      00510 RETRST EX (SP),HL
FB28 C20000 00520          JP NZ,0
FB2B E5      00530          PUSH HL
FB2C D7      00540          RST 10H
FB2D FEDD    00550          CP 0DDH
FB2F CA40FB 00560          JP Z, FN
FB32 FEDF    00570          CP 0DFH
FB34 DA3CFB 00580          JP C,RST101
FB37 FEE5    00590          CP 0E5H
FB39 DA40FB 00600          JP C, FN
FB3C E1      00610 RST101 POP HL
FB3D C30000 00620          JP 0
FB40 C1      00630 FN     POP BC
FB41 C1      00640          POP BC
FB42 D6D7    00650          SUB 0D7H
FB44 0600    00660          LD B,0

```

FB46	87	00670	ADD A,A	
FB47	4F	00680	LD C,A	
FB48	C5	00690	PUSH BC	
FB49	D7	00700	RST 10H	
FB4A	CD2C25	00710	CALL ARG	
FB4D	E7	00720	RST 20H	
FB4E	DA7225	00730	JP C,2572H	
FB51	E3	00740	EX (SP),HL	
FB52	113E25	00750	LD DE,253EH	
FB55	D5	00760	PUSH DE	
FB56	0189FE	00770	LD BC,JMPTAB-12	
FB59	C38625	00780	JP 2586H	
FB5C	E3	00790	RST20N EX (SP),HL	
FB5D	7D	00800	LD A,L	
FB5E	FEB2	00810	CP 0B2H	
FB60	2003	00820	JR NZ,RST1	
FB62	7C	00830	LD A,H	
FB63	FE0A	00840	CP 0AH	
FB65	E3	00850	RST1 EX (SP),HL	
FB66	C20000	00860	JP NZ,0	
FB69	E5	00870	PUSH HL	
FB6A	210500	00880	LD HL,5	
FB6D	39	00890	ADD HL,SP	
FB6E	7E	00900	LD A,(HL)	
FB6F	E1	00910	POP HL	
FB70	FE23	00920	CP 23H	
FB72	C20000	00930	RST2 JP NZ,0	
FB75	E1	00940	POP HL	
FB76	E1	00950	POP HL	
FB77	E7	00960	RST 20H	
FB78	F5	00970	PUSH AF	
FB79	DCB10A	00980	CALL C,CSNG	
FB7C	F1	00990	POP AF	
FB7D	3808	01000	JR C,SNG	
FB7F	2A1D41	01010	LD HL,(X)	
FB82	E5	01020	PUSH HL	
FB83	2A1F41	01030	LD HL,(X+2)	
FB86	E5	01040	PUSH HL	
FB87	CDA409	01050	SNG CALL SPUSHX	
FB8A	F5	01060	PUSH AF	
FB8B	167F	01070	LD D,7FH	
FB8D	01ABFC	01080	LD BC,POWER	
FB90	C3CD23	01090	JP 23CDH	
FB93	21F0FF	01100	YDR1 LD HL,DR1	;R1 NACH X
FB96	111D41	01110	XC LD DE,X	;CONST (P:HL) NACH X
FB99	010800	01120	DMOVE LD BC,8	
FB9C	EDB0	01130	LDIR	
FB9E	C9	01140	RET	
FB9F	212741	01150	XY LD HL,Y	;Y NACH X
FBA2	18F2	01160	JR XC	
FBA4	21F0FF	01170	YDR1 LD HL,DR1	;DR1 NACH Y
FBA7	112741	01180	YC LD DE,Y	;CONST (P:HL) NACH Y
FBA8	18ED	01190	JR DMOVE	
FBA9	21F8FF	01200	YDR2 LD HL,DR2	;DR2 NACH Y
FBAF	18F6	01210	JR YC	
FBB1	211D41	01220	YX LD HL,X	;X NACH Y
FBB4	18F1	01230	JR YC	
FBB6	211D41	01240	DR1X LD HL,X	;X NACH DR1
FBB9	11F0FF	01250	DR1C LD DE,DR1	;CONST (P:HL) NACH DR1
FBBC	18DB	01260	JR DMOVE	
FBBE	211D41	01270	DR2X LD HL,X	;X NACH DR2
FBC1	11F8FF	01280	DR2C LD DE,DR2	;CONST (P:HL) NACH DR2
FBC4	18D3	01290	JR DMOVE	
FBC6	211D41	01300	DR3X LD HL,X	;X NACH DR3
FBC9	11E8FF	01310	LD DE,DR3	
FBCB	18CB	01320	JR DMOVE	

FBCE 21EBFF	01330	YDR3	LD HL,DR3	;DR3 NACH Y
FBD1 18D4	01340		JR YC	
FBD3 2167FF	01350	YK1	LD HL,K1	;1 NACH Y
FBD6 18CF	01360		JR YC -	
	01370			;UP ZUR BERECHNUNG DER REIHEN:
	01380			; (HL) = ANZAHL DER GLIEDER
	01390			; (HL+1..) = KOEFFIZIENTEN
	01400			;REIHE1: X=C1*X+C3*X**3+C5*X**5
	01410			;REIHE2: X=C1+C2*X+C3*X**2+C4*X**3
FBDB E5	01420	REIHE1	PUSH HL	;POINTER RETTEN
FBD9 CDBEFB	01430		CALL DR2X	;X IN DR2 RETTEN
FBDC CDB1FB	01440		CALL YX	
FBDF CDA10D	01450	REIHE3	CALL MLT	;X=X*X
FBE2 E1	01460		POP HL	
FBE3 1108FC	01470		LD DE,DRET	
FBE6 D5	01480		PUSH DE	
FBE7 E5	01490	REIHE2	PUSH HL	;POINTER RETTEN
FBE8 CDB6FB	01500		CALL DR1X	;X NACH DR1
FBEB E1	01510		POP HL	
FBEC 7E	01520		LD A,(HL)	;ANZAHL DER GLIEDER
FBED 23	01530		INC HL	
FBEE CD96FB	01540		CALL XC	;1.KOEFFIZIENT NACH X
FBF1 F5	01550		PUSH AF	
FBF2 F1	01560	LOOP2	POP AF	
FBF3 3D	01570		DEC A	;ALLE GLIEDER BERRECHNET?
FBF4 C8	01580		RET Z	;JA,RET
FBF5 F5	01590		PUSH AF	
FBF6 E5	01600		PUSH HL	
FBF7 CDA4FB	01610		CALL YDR1	;MIT X MULTIPLIZIEREN
FBFA CDA10D	01620		CALL MLT	
FBFD E1	01630		POP HL	
FBFE CDA7FB	01640		CALL YC	;KOEFFIZIENT NACH Y
FC01 E5	01650		PUSH HL	
FC02 CD770C	01660		CALL ADD	;ADDIEREN
FC05 E1	01670		POP HL	
FC06 18EA	01680		JR LOOP2	
FC08 CDACFB	01690	DRET	CALL YDR2	;ERG*X (REIHE1)
FC0B C3A10D	01700		JP MLT	
FC0E CD5509	01710	FNSQR	CALL TEST	;X=0?
FC11 C8	01720		RET Z	;JA, FERTIG
FC12 CDB6FB	01730		CALL DR1X	;X NACH DR1
FC15 CDE713	01740		CALL SQR	;1.NAEHERUNG
FC18 CDD80A	01750		CALL CDBL	
FC1B 0602	01760		LD B,2	;2 ITERATIONEN
FC1D C5	01770	DLOOP1	PUSH BC	
FC1E CDB1FB	01780		CALL YX	;XN NACH Y
FC21 CDBEFB	01790		CALL DR2X	;XN NACH DR2
FC24 CD93FB	01800		CALL XDR1	
FC27 CDE50D	01810		CALL DIV	;X=X/XN
FC2A CDACFB	01820		CALL YDR2	;X=X+XN
FC2D CD770C	01830		CALL ADD	
FC30 CDB1FB	01840		CALL YX	
FC33 2157FF	01850		LD HL,K1HALB	;XN+1=.5*(X/XN+XN)
FC36 CD96FB	01860		CALL XC	
FC39 CDA10D	01870		CALL MLT	
FC3C C1	01880		POP BC	
FC3D 10DE	01890		DJNZ DLOOP1	
FC3F C9	01900		RET	
FC40 CD5509	01910	FNLOG	CALL TEST	;X<=0
FC43 B7	01920		OR A	
FC44 EA4A1E	01930		JP PE,FCERR	;JA, FCERR
FC47 1167FF	01940		LD DE,K1	
FC4A CD490A	01950		CALL CMP	;X=1?
FC4D CA7807	01960		JP Z,NULL	;JA, ERG=0
FC50 212441	01970		LD HL,X+7	;EXP X
FC53 7E	01980		LD A,(HL)	

FC54 D680	01990		SUB 80H
FC56 F5	02000		PUSH AF
FC57 3680	02010		LD (HL),80H ;EXP X=0
FC59 21C8FF	02020		LD HL,KSQDR2
FC5C CDA7FB	02030		CALL YC
FC5F CDA10D	02040		CALL MLT ;X=X*SQR(2)
FC62 CDB6FB	02050		CALL DR1X
FC65 CDD3FB	02060		CALL YK1 ;X=(X-1)/(X+1)
FC68 CD770C	02070		CALL ADD
FC6B CDBEFB	02080		CALL DR2X
FC6E CDD3FB	02090		CALL YK1
FC71 CD93FB	02100		CALL XDR1
FC74 CD700C	02110		CALL SUB
FC77 CDACFB	02120		CALL YDR2
FC7A CDE50D	02130		CALL DIV
FC7D 216FFF	02140		LD HL,KLOG ;REIHE BERECHNEN
FC80 CDD8FB	02150		CALL REIHE1
FC83 CDB1FB	02160		CALL YX
FC86 CD770C	02170		CALL ADD ;X=2*X
FC89 CDB6FB	02180		CALL DR1X ;ERG. RETTEN
FC8C F1	02190		POP AF
FC8D CD6409	02200		CALL FLOAT ;EXP X NACH X
FC90 CDE30A	02210		CALL CDBL1
FC93 2157FF	02220		LD HL,K1HALB
FC96 CDA7FB	02230		CALL YC
FC99 CD700C	02240		CALL SUB ;X=X-.5
FC9C 21E0FF	02250		LD HL,KLN2
FC9F CDA7FB	02260		CALL YC
FCA2 CDA10D	02270		CALL MLT ;X=X*LN(2)
FCA5 CDA4FB	02280		CALL YDR1
FCA8 C3770C	02290		JP ADD ;+ REIHENERG.
FCAE E7	02300	POWER	RST 20H
FCAC 3005	02310		JR NC,POPARG
FCAE F1	02320		POP AF
FCAF DAF213	02330		JP C,SPOWER
FCB2 F5	02340		PUSH AF
FCB3 F1	02350	POPARG	POP AF
FCB4 E1	02360		POP HL
FCB5 22F6FF	02370		LD (DR1+6),HL
FCB8 E1	02380		POP HL
FCB9 22F4FF	02390		LD (DR1+4),HL
FCBC 210000	02400		LD HL,0
FCBF 54	02410		LD D,H
FCC0 5D	02420		LD E,L
FCC1 3802	02430		JR C,DBLPOW
FCC3 D1	02440		POP DE
FCC4 E1	02450		POP HL
FCC5 ED53F2FF	02460	DBLPOW	LD (DR1+2),DE
FCC9 22F0FF	02470		LD (DR1),HL
FCCC CDD80A	02480		CALL CDBL
FCCF CD5509	02490		CALL TEST
FCD2 3AF7FF	02500		LD A,(DR1+7)
FCD5 2B5B	02510		JR Z,FNEXP
FCD7 F2DEFC	02520		JP P,TEST2
FCDA B7	02530		OR A
FCDB CA9A19	02540		JP Z,DOERR
FCDE B7	02550	TEST2	OR A
FCDF CA7907	02560		JP Z,NULL+1
FCE2 3AF6FF	02570		LD A,(DR1+6)
FCE5 F67F	02580		OR 7FH
FCE7 CDC6FB	02590		CALL DR3X
FCEA CDB1FB	02600		CALL YX
FCED F220FD	02610		JP P,NINT
FCF0 3A2441	02620		LD A,(X+7)
FCF3 FEB8	02630		CP 0B8H
FCF5 3E00	02640		LD A,0

FCF7	2003	02650	JR NZ, I1	
FCF9	3A1D41	02660	LD A, (X)	
FCFC	04	02670	INC B	
FCFD	05	02680	DEC B	
FCFE	F5	02690	PUSH AF	
FCFF	301C	02700	JR NC, NINT1	
FD01	F1	02710	POP AF	
FD02	A7	02720	AND A	
FD03	CD5509	02730	CALL TEST	
FD06	F5	02740	PUSH AF	
FD07	DC8209	02750	CALL C, NEG	
FD0A	CD7B0B	02760	CALL INT1	
FD0D	3A1D41	02770	LD A, (X)	
FD10	C1	02780	POP BC	
FD11	F5	02790	PUSH AF	
FD12	C5	02800	PUSH BC	
FD13	CDD80C	02810	CALL NORM	
FD16	F1	02820	POP AF	
FD17	DC8209	02830	CALL C, NEG	
FD1A	CD4F0A	02840	CALL DCPXY	
FD1D	E1	02850	POP HL	NINT1
FD1E	7C	02860	LD A, H	
FD1F	1F	02870	RRA	
FD20	CD93FB	02880	CALL XDR1	NINT
FD23	DCE213	02890	CALL C, ENEG	
FD26	CC8209	02900	CALL Z, NEG	
FD29	CD40FC	02910	CALL FNLOG	
FD2C	CDCEFB	02920	CALL YDR3	
FD2F	CDA10D	02930	CALL MLT	
FD32	CD5509	02940	CALL TEST	FNEXP ; X=0?
FD35	2167FF	02950	LD HL, K1	
FD38	CA96FB	02960	JP Z, XC	; JA, ERG=1
FD3B	CDB6FB	02970	CALL DR1X	
FD3E	21E0FF	02980	LD HL, KLN2	; X=X/LN2
FD41	CDA7FB	02990	CALL YC	
FD44	CDE50D	03000	CALL DIV	
FD47	3A2441	03010	LD A, (X+7)	
FD4A	FE88	03020	CP 88H	; X>=128?
FD4C	D283FD	03030	JP NC, OVER	; JA
FD4F	CD590B	03040	CALL INT	
FD52	7D	03050	LD A, L	; X<126 AND X>-127?
FD53	C680	03060	ADD A, 80H	
FD55	C602	03070	ADD A, 2	
FD57	DAB3FD	03080	JP C, OVER	; NEIN
FD5A	F5	03090	PUSH AF	; INT(X)+130
FD5B	CDD80A	03100	CALL CDBL	; X=LN2-X MOD LN2
FD5E	CDD3FB	03110	CALL YK1	
FD61	CD770C	03120	CALL ADD	
FD64	21E0FF	03130	LD HL, KLN2	
FD67	CDA7FB	03140	CALL YC	
FD6A	CDA10D	03150	CALL MLT	
FD6D	CDA4FB	03160	CALL YDR1	
FD70	CD700C	03170	CALL SUB	
FD73	21F6FE	03180	LD HL, KEXP	; TAYLOR-REIHE
FD76	CDE7FB	03190	CALL REIHE2	
FD79	CDD3FB	03200	CALL YK1	
FD7C	F1	03210	POP AF	; EXP X=INT(X/LN2)+130
FD7D	322E41	03220	LD (Y+7), A	
FD80	C3A10D	03230	JP MLT	
FD83	3AF6FF	03240	LD A, (DR1+6)	OVER
FD86	B7	03250	OR A	; X<0?
FD87	FA7807	03260	JP M, NULL	; JA
FD8A	C3B207	03270	JP OVERR	
FD8D	21D0FF	03280	LD HL, KPI2	; X=X+PI/2
FD90	CDA7FB	03290	CALL YC	
FD93	CD770C	03300	CALL ADD	

FD96	CD5509	03310	FNSIN	CALL TEST	; X=0?
FD99	C8	03320		RET Z	; JA
FD9A	F5	03330		PUSH AF	; SGN RETTEN
FD9B	DC8209	03340		CALL C, NEG	; X=ABS(X)
FD9E	21D8FF	03350		LD HL, K2PI	; X=X/(2*PI)
FDA1	CDA7FB	03360		CALL YC	
FDA4	CDE50D	03370		CALL DIV	
FDA7	CDB6FB	03380		CALL DR1X	
FDA8	CD590B	03390		CALL INT	; X=X-INT(X)
FDAD	CDDB0A	03400		CALL CDEL	
FDB0	CDA4FB	03410		CALL YDR1	
FDB3	CD8209	03420		CALL NEG	
FDB6	CD770C	03430		CALL ADD	
FDB9	3A2441	03440		LD A, (X+7)	
FDBC	FE7F	03450		CP 7FH	; ABS(X)<1/4?
FDBE	3824	03460		JR C, L0	; JA
FDC0	CD8209	03470		CALL NEG	; X=1/2-X
FDC3	2157FF	03480		LD HL, K1HALB	
FDC6	CDA7FB	03490		CALL YC	
FDC9	CD770C	03500		CALL ADD	
FDCC	CD5509	03510		CALL TEST	; X<0
FDCF	3013	03520		JR NC, L0	; NEIN
FDD1	3A2441	03530		LD A, (X+7)	
FDD4	FE7F	03540		CP 7FH	; ABS(X)<1/4?
FDD6	380C	03550		JR C, L0	; JA
FDD8	2157FF	03560		LD HL, K1HALB	; X=-(X+1/2)
FDD8	CDA7FB	03570		CALL YC	
FDDE	CD770C	03580		CALL ADD	
FDE1	CD8209	03590		CALL NEG	
FDE4	F1	03600	L0	POP AF	; VORZ. ZURUECK
FDE5	DC8209	03610		CALL C, NEG	
FDE8	21D8FF	03620		LD HL, K2PI	
FDEB	CDA7FB	03630		CALL YC	
FDEE	CDA10D	03640		CALL MLT	
fdf1	21A5FE	03650		LD HL, KSIN	; TAYLOR-REIHE
fdf4	C3D8FB	03660		JP REIHE1	
fdf7	CDC6FB	03670	FNTAN	CALL DR3X	
fdfa	CD8DFD	03680		CALL FNCOS	
fdfd	CD5509	03690		CALL TEST	; COS(X)=0?
FE00	CA4A1E	03700		JP Z, FCERR	; JA
FE03	CDCEFB	03710		CALL YDR3	
FE06	CDC6FB	03720		CALL DR3X	
FE09	CD9FFB	03730		CALL XY	
FE0C	CD96FD	03740		CALL FNSIN	
FE0F	CDCEFB	03750		CALL YDR3	
FE12	C3E50D	03760		JP DIV	; TAN(X)=SIN(X)/COS(X)
FE15	CD5509	03770	FNATN	CALL TEST	; X=0?
FE18	C8	03780		RET Z	; JA
FE19	F5	03790		PUSH AF	
FE1A	DC8209	03800		CALL C, NEG	; X=ABS(X)
FE1D	3A2441	03810		LD A, (X+7)	
FE20	FE81	03820		CP 81H	; X<1?
FE22	380D	03830		JR C, DL1	; JA
FE24	CDB1FB	03840		CALL YX	; X=1/X
FE27	2167FF	03850		LD HL, K1	
FE2A	CD96FB	03860		CALL XC	
FE2D	CDE50D	03870		CALL DIV	
FE30	AF	03880		XOR A	
FE31	F5	03890	DL1	PUSH AF	
FE32	0602	03900		LD B, 2	
FE34	3A2441	03910	DL3	LD A, (X+7)	
FE37	FE7E	03920		CP 7EH	; X<1/8?
FE39	3828	03930		JR C, DL4	; JA
FE3B	C5	03940		PUSH BC	
FE3C	CDC6FB	03950		CALL DR3X	; X=X/(SQR(X*X+1)+1)
FE3F	CDB1FB	03960		CALL YX	

FE42	CDA10D	03970	CALL MLT	
FE45	CDD3FB	03980	CALL YK1	
FE4B	CD770C	03990	CALL ADD	
FE4B	CD0EFC	04000	CALL FNSQR	
FE4E	CDD3FB	04010	CALL YK1	
FE51	CD770C	04020	CALL ADD	
FE54	CDB1FB	04030	CALL YX	
FE57	21E8FF	04040	LD HL,DR3	
FE5A	CD96FB	04050	CALL XC	
FE5D	CDE50D	04060	CALL DIV	
FE60	C1	04070	POP BC	
FE61	10D1	04080	DJNZ DL3	
FE63	C5	04090	PUSH BC	
FE64	2178FE	04100	LD HL,RET2	; TAYLOR-REIHE
FE67	E5	04110	PUSH HL	
FE68	216FFF	04120	LD HL,KLOG	
FE6B	E5	04130	PUSH HL	
FE6C	CDB1FB	04140	CALL YX	
FE6F	CDBEFB	04150	CALL DR2X	
FE72	CD8209	04160	CALL NEG	
FE75	C3DFFB	04170	JP REIHE3	
FE78	212441	04180	LD HL,4124H	
FE7B	C1	04190	POP BC	
FE7C	7E	04200	LD A,(HL)	; X=X**("B)
FE7D	90	04210	SUB B	
FE7E	C602	04220	ADD A,2	
FE80	77	04230	LD (HL),A	
FE81	F1	04240	POP AF	; ARG>1?
FE82	3B0C	04250	JR C,DL2	; NEIN
FE84	CD8209	04260	CALL NEG	; X=PI/2-X
FE87	21D0FF	04270	LD HL,KPI2	
FE8A	CDA7FB	04280	CALL YC	
FE8D	CD770C	04290	CALL ADD	
FE90	F1	04300	POP AF	; ARG<0?
FE91	DA8209	04310	JP C,NEG	; JA
FE94	C9	04320	RET	
FE95	0EFC	04330	DEFW FNSQR	
FE97	0000	04340	DEFW 0	
FE99	40FC	04350	DEFW FNLOG	
FE9B	32FD	04360	DEFW FNEXP	
FE9D	8DFD	04370	DEFW FNCOS	
FE9F	96FD	04380	DEFW FNSIN	
FEA1	F7FD	04390	DEFW FNTAN	
FEA3	15FE	04400	DEFW FNATN	
		04410	; KONSTANTEN:	
FEA5	0A	04420	DEFB 10	; SIN(10)
FEA6	2E17	04430	DEFW 172EH	; -(1/19!-1.55/15!)
FEA8	A9BB	04440	DEFW 0BBA9H	
FEAA	9515	04450	DEFW 1595H	
FEAC	9748	04460	DEFW 4897H	
FEAE	536A	04470	DEFW 6A53H	; 1/17!
FEB0	8581	04480	DEFW 8185H	
FEB2	3B96	04490	DEFW 963BH	
FEB4	4A50	04500	DEFW 504AH	
FEB6	F9C0	04510	DEFW 0C0F9H	; -1/15!
FEB8	9D39	04520	DEFW 399DH	
FEBA	9F3F	04530	DEFW 3F9FH	
FEBC	D758	04540	DEFW 58D7H	
FEBE	4C68	04550	DEFW 684CH	; 1/13!
FEC0	439D	04560	DEFW 9D43H	
FEC2	3092	04570	DEFW 9230H	
FEC4	3060	04580	DEFW 6030H	
FEC6	1C27	04590	DEFW 271CH	; -1/11!
FEC8	AA3F	04600	DEFW 3FAAH	
FECA	2B32	04610	DEFW 322BH	
FECB	D767	04620	DEFW 67D7H	

FECE 9C39	04630	DEFW 399CH	;1/9!
FED0 B62A	04640	DEFW 2AB6H	
FED2 1DEF	04650	DEFW 0EF1DH	
FED4 386E	04660	DEFW 6E38H	
FED6 D000	04670	DEFW 0DOH	; -1/7!
FED8 ODD0	04680	DEFW 0D00DH	
FEDA 000D	04690	DEFW 0D00H	
FEDC D074	04700	DEFW 74D0H	
FEDE 8988	04710	DEFW 8889H	;1/5!
FEE0 8888	04720	DEFW 8888H	
FEE2 8888	04730	DEFW 8888H	
FEE4 087A	04740	DEFW 7A08H	
FEE6 ABAA	04750	DEFW 0AAABH	; -1/3!
FEE8 AAAA	04760	DEFW 0AAAAH	
FEEA AAAA	04770	DEFW 0AAAAH	
FEEC AA7E	04780	DEFW 7EAAH	
FEEE 0000	04790	DEFW 0	;1
FEF0 0000	04800	DEFW 0	
FEF2 0000	04810	DEFW 0	
FEF4 0081	04820	DEFW 8100H	
FEF6 0F	04830 KEXP	DEFB 15	;EXP(15)
FEF7 DA79	04840	DEFW 79DAH	;CA 1/14!
FEF9 41A0	04850	DEFW 0A041H	
FEFB 0FC8	04860	DEFW 0C80FH	
FEFD 405C	04870	DEFW 5C40H	
FEFF 4C68	04880	DEFW 684CH	; -1/13!
FF01 439D	04890	DEFW 9D43H	
FF03 3092	04900	DEFW 9230H	
FF05 B060	04910	DEFW 60B0H	
FF07 BEC4	04920	DEFW 0C4BEH	;1/12!
FF09 C67F	04930	DEFW 7FC6H	
FF0B C776	04940	DEFW 76C7H	
FF0D 0F64	04950	DEFW 640FH	
FF0F 1C27	04960	DEFW 271CH	; -1/11!
FF11 AA3F	04970	DEFW 3FAAH	
FF13 2B32	04980	DEFW 322BH	
FF15 D767	04990	DEFW 67D7H	
FF17 E4FA	05000	DEFW 0FAE4H	;1/10!
FF19 C4BB	05010	DEFW 0BBC4H	
FF1B 7DF2	05020	DEFW 0F27DH	
FF1D 136B	05030	DEFW 6B13H	
FF1F 9C39	05040	DEFW 399CH	; -1/9!
FF21 B62A	05050	DEFW 2AB6H	
FF23 1DEF	05060	DEFW 0EF1DH	
FF25 B86E	05070	DEFW 6EBBH	
FF27 D000	05080	DEFW 0DOH	;1/8!
FF29 ODD0	05090	DEFW 0D00DH	
FF2B 000D	05100	DEFW 0D00H	
FF2D 5071	05110	DEFW 7150H	
FF2F D000	05120	DEFW 0DOH	;1/7!
FF31 ODD0	05130	DEFW 0D00DH	
FF33 000D	05140	DEFW 0D00H	
FF35 D074	05150	DEFW 74D0H	
FF37 B660	05160	DEFW 60B6H	;1/6!
FF39 0BB6	05170	DEFW 0B60BH	
FF3B 600B	05180	DEFW 0B60H	
FF3D 3677	05190	DEFW 7736H	
FF3F 8988	05200	DEFW 8889H	; -1/5!
FF41 8888	05210	DEFW 8888H	
FF43 8888	05220	DEFW 8888H	
FF45 887A	05230	DEFW 7A88H	
FF47 ABAA	05240	DEFW 0AAABH	;1/4!
FF49 AAAA	05250	DEFW 0AAAAH	
FF4B AAAA	05260	DEFW 0AAAAH	
FF4D 2A7C	05270	DEFW 7C2AH	
FF4F ABAA	05280	DEFW 0AAABH	; -1/3!

FF51	AAAA	05290	DEFW	OAAAAH	
FF53	AAAA	05300	DEFW	OAAAAH	
FF55	AA7E	05310	DEFW	7EAAH	
FF57	0000	05320	K1HALB	DEFW 0	;1/2
FF59	0000	05330		DEFW 0	
FF5B	0000	05340		DEFW 0	
FF5D	0080	05350		DEFW 8000H	
FF5F	0000	05360		DEFW 0	;-1
FF61	0000	05370		DEFW 0	
FF63	0000	05380		DEFW 0	
FF65	8081	05390		DEFW 8180H	
FF67	0000	05400	K1	DEFW 0	;1
FF69	0000	05410		DEFW 0	
FF6B	0000	05420		DEFW 0	
FF6D	0081	05430		DEFW 8100H	
FF6F	0B	05440	KLOG	DEFB 11	;LOG,ATN(11)
FF70	C330	05450		DEFW 30C3H	;1/21
FF72	0CC3	05460		DEFW 0C30CH	
FF74	300C	05470		DEFW 0C30H	
FF76	437C	05480		DEFW 7C43H	
FF78	4379	05490		DEFW 7943H	;1/19
FF7A	0DE5	05500		DEFW 0E50DH	
FF7C	3594	05510		DEFW 9435H	
FF7E	577C	05520		DEFW 7C57H	
FF80	F1F0	05530		DEFW 0F0F1H	;1/17
FF82	F0F0	05540		DEFW 0F0F0H	
FF84	F0F0	05550		DEFW 0F0F0H	
FF86	707C	05560		DEFW 7C70H	
FF88	8988	05570		DEFW 8889H	;1/15
FF8A	8888	05580		DEFW 8888H	
FF8C	8888	05590		DEFW 8888H	
FF8E	087D	05600		DEFW 7D08H	
FF90	9ED8	05610		DEFW 0D89EH	;1/13
FF92	899D	05620		DEFW 9D89H	
FF94	D889	05630		DEFW 89D8H	
FF96	1D7D	05640		DEFW 7D1DH	
FF98	2FBA	05650		DEFW 0BA2FH	;1/11
FF9A	EBA2	05660		DEFW 0A2E8H	
FF9C	8B2E	05670		DEFW 2E8BH	
FF9E	3A7D	05680		DEFW 7D3AH	
FFA0	E438	05690		DEFW 38E4H	;1/9
FFA2	8EE3	05700		DEFW 0E38EH	
FFA4	388E	05710		DEFW 08E38H	
FFA6	637D	05720		DEFW 7D63H	
FFA8	9224	05730		DEFW 2492H	;1/7
FFAA	4992	05740		DEFW 9249H	
FFAC	2449	05750		DEFW 4924H	
FFAE	127E	05760		DEFW 7E12H	
FFB0	CDCC	05770		DEFW 0CCCDH	;1/5
FFB2	CCCC	05780		DEFW 0CCCCH	
FFB4	CCCC	05790		DEFW 0CCCCH	
FFB6	4C7E	05800		DEFW 7E4CH	
FFB8	ABAA	05810		DEFW 0AAABH	;1/3
FFBA	AAAA	05820		DEFW 0AAAAH	
FFBC	AAAA	05830		DEFW 0AAAAH	
FFBE	2A7F	05840		DEFW 7F2AH	
FFC0	0000	05850		DEFW 0	;1
FFC2	0000	05860		DEFW 0	
FFC4	0000	05870		DEFW 0	
FFC6	0081	05880		DEFW 8100H	
FFC8	65DE	05890	KSQDR2	DEFW 0DE65H	;SQR(2)
FFCA	F933	05900		DEFW 33F9H	
FFCC	F304	05910		DEFW 4F3H	
FFCE	3581	05920		DEFW 8135H	
FFD0	C468	05930	KPI2	DEFW 68C4H	;PI/2
FFD2	21A2	05940		DEFW 0A221H	

FFD4	DA0F	05950	DEFW	OFDAH	
FFD6	4981	05960	DEFW	8149H	
FFD8	C46B	05970	DEFW	68C4H	;2*PI
FFDA	21A2	05980	DEFW	0A221H	
FFDC	DA0F	05990	DEFW	OFDAH	
FFDE	4983	06000	DEFW	8349H	
FFE0	7BCF	06010	DEFW	0CF7BH	;LOG(2)
FFE2	D1F7	06020	DEFW	0F7D1H	
FFE4	1772	06030	DEFW	7217H	
FFE6	3180	06040	DEFW	8031H	
		06050		;REGISTER	
FFEB	0000	06060	DR3	DEFW	0
FFEA	0000	06070		DEFW	0
FFEC	0000	06080		DEFW	0
FFEE	0000	06090		DEFW	0
FFF0	0000	06100	DR1	DEFW	0
FFF2	0000	06110		DEFW	0
FFF4	0000	06120		DEFW	0
FFF6	0000	06130		DEFW	0
FFF8	0000	06140	DR2	DEFW	0
FFFA	0000	06150		DEFW	0
FFFC	0000	06160		DEFW	0
FFFE	0000	06170		DEFW	0
FAFF		06180		END	INIT
00000	TOTAL ERRORS				
24187	TEXT AREA BYTES LEFT				

6.4 Basicprogrammpacker PACK - 124-

```

00010 ; * BASIC-PROGRAMM-PACKER *
00020 ; ENTFERNT ALLE LEERRAUEME UND KOMMENTARE
00030 ; COPYRIGHT BY LUIDGER ROECKRATH, 26.9.81
00040 ;
00050 ; BASICUNTERPROGRAMME
00060 ;
1D78 00070 CHRGET EQU 1D78H ;RST 10H UP
1F07 00080 SEOL EQU 1F07H ;ZEILENENDE SUCHEN
1B10 00090 LARG EQU 1B10H ;2 ARGUMENTE ANALYSIEREN
1E5B 00100 DECLN1 EQU 1E5BH ;ZNR DECODIEREN
1B2C 00110 SLINE EQU 1B2CH ;ZEILE IM PROGRAMMTEXT SUCHEN
0FA7 00120 PRZNR1 EQU 0FA7H ;' IN ' ZNR AUSGEBEN
0FAF 00130 PRZNR EQU 0FAFH ;ZNR AUSGEBEN
2B75 00140 PRINTT EQU 2B75H ;TEXT AUSGEBEN
20FE 00150 CR EQU 20FEH ;CR AUSGEBEN
1AEB 00160 ENDML EQU 1AEBH ;HAUPTSCHLEIFENENDE
1E4A 00170 FCERR EQU 1E4AH ;FC-ERROR
00180 ;
00190 ; BASICZEIGER
00200 ;
1929 00210 TEXT6 EQU 1929H ;TEXT 'READY'
40A4 00220 TEXTA EQU 40A4H ;ZEIGER AUF PROGRAMMTEXTANFANG
40F9 00230 VARA EQU 40F9H ;ZEIGER AUF ANFANG DER VARTAB
4191 00240 KILLV EQU 4191H
00250 ;
00260 ; TOKENS
00270 ;
008D 00280 GOTO EQU 8DH
008E 00290 RUN EQU 8EH
0091 00300 GOSUB EQU 91H
0093 00310 REM EQU 93H
0095 00320 ELSE EQU 95H
009F 00330 RESUME EQU 9FH
00CA 00340 THEN EQU 0CAH
0088 00350 DATA EQU 88H
00AA 00360 KILLT EQU 0AAH
00370 ;
FDDD 00390 ORG 0FDDDH
00400 ;
00410 ; INTIALISIERUNG
00420 ;
FDDD 2A9241 00430 INIT LD HL, (KILLV+1)
FDE0 22F9FD 00440 LD (KILLCH+1),HL
FDE3 21EAFD 00450 LD HL, KILL
FDE6 229241 00460 LD (KILLV+1),HL
FDE9 C9 00470 RET
00480 ;
00490 ;KILL
00500 ;
FDEA 2B0F 00520 KILL JR Z, KILL2
FDEC 3B0D 00530 JR C, KILL2
FDEE FECE 00540 CP OCEH
FDF0 2B09 00550 JR Z, KILL2
FDF2 FE2E 00560 CP ' '
FDF4 2B05 00570 JR Z, KILL2
FDF6 2B 00580 DEC HL
FDF7 D7 00590 RST 10H
FDFB C30000 00600 KILLCH JP 0
FDFB CD101B 00610 KILL2 CALL LARG
FDFF D1 00620 POP DE
FDFD D5 00630 PUSH DE
FE00 C5 00640 PUSH BC
FE01 CD2C1B 00650 CALL SLINE
FE04 69 00660 LD L, C
FE05 60 00670 LD H, B
FE06 D1 00680 POP DE

```

FE07 DF	00690	RST	18H	FE72 2002	01350	JR	NZ, REF3
FE08 D5	00700	PUSH	DE	FE74 36FF	01360	LD	(HL), OFFH
FE09 DA4A1E	00710	JP	C, FCERR	FE76 E1	01370 REF3	POP	HL
FE0C CD74FF	00720	CALL	LEN	FE77 D7	01380	RST	10H
FE0F 2191FF	00730	LD	HL, TEXT1	FE78 FE2C	01390	CP	' , '
FE12 CD752B	00740	CALL	PRINTT	FE7A 3E8D	01400	LD	A, GOTO
FE15 2AA440	00750	LD	HL, (TEXTA)	FE7C 28D5	01410	JR	Z, JUMP1
FE18 2B	00760	DEC	HL	FE7E 2B	01420	DEC	HL
FE19 D5	00770	PUSH	DE	FE7F 18A5	01430	JR	REF2
FE1A D1	00780 REF1	POP	DE	FE81 21C6FF	01440 UNDEFL	LD	HL, TEXT3
FE1B 23	00790	INC	HL	FE84 CD752B	01450	CALL	PRINTT
FE1C 7E	00800	LD	A, (HL)	FE87 EB	01460	EX	DE, HL
FE1D 23	00810	INC	HL	FE88 CDAF0F	01470	CALL	PRZNR
FE1E B6	00820	OR	(HL)	FE8B D1	01480	POP	DE
FE1F 2877	00830	JR	Z, PACK	FE8C E1	01490	POP	HL
FE21 23	00840	INC	HL	FE8D E5	01500	PUSH	HL
FE22 5E	00850	LD	E, (HL)	FE8E D5	01510	PUSH	DE
FE23 23	00860	INC	HL	FE8F CDA70F	01520	CALL	PRZNR1
FE24 56	00870	LD	D, (HL)	FE92 E1	01530	POP	HL
FE25 D5	00880	PUSH	DE	FE93 CDFE20	01540	CALL	CR
FE26 D7	00890 REF2	RST	10H	FE96 189E	01550	JR	REF2
FE27 A7	00900	AND	A	FE98 21B5FF	01560 PACK	LD	HL, TEXT2
FE28 28F0	00910	JR	Z, REF1	FE9B CD752B	01570	CALL	PRINTT
FE2A FE22	00920	CP	' "'	FE9E E1	01580	POP	HL
FE2C 200B	00930	JR	NZ, JUMP	FE9F D1	01590	POP	DE
FE2E D7	00940 STRING	RST	10H	FEA0 E5	01600	PUSH	HL
FE2F A7	00950	AND	A	FEA1 D5	01610	PUSH	DE
FE30 28E8	00960	JR	Z, REF1	FEA2 2B	01620	DEC	HL
FE32 FE22	00970	CP	' "'	FEA3 23	01630 PACKL	INC	HL
FE34 20F8	00980	JR	NZ, STRING	FEA4 7E	01640 PACKL1	LD	A, (HL)
FE36 FE8D	00990 JUMP	CP	GOTO	FEA5 23	01650	INC	HL
FE38 2819	01000	JR	Z, JUMP1	FEA6 B6	01660	OR	(HL)
FE3A FE91	01010	CP	GOSUB	FEA7 CA3BFF	01670	JP	Z, PACKE
FE3C 2815	01020	JR	Z, JUMP1	FEAA 23	01680	INC	HL
FE3E FE8E	01030	CP	RUN	FEAB 5E	01690	LD	E, (HL)
FE40 2811	01040	JR	Z, JUMP1	FEAC 23	01700	INC	HL
FE42 FE9F	01050	CP	RESUME	FEAD 56	01710	LD	D, (HL)
FE44 280D	01060	JR	Z, JUMP1	FEAE 23	01720	INC	HL
FE46 FE95	01070	CP	ELSE	FEAF E3	01730	EX	(SP), HL
FE48 2804	01080	JR	Z, JUMP2	FEB0 DF	01740	RST	18H
FE4A FECA	01090	CP	THEN	FEB1 E3	01750	EX	(SP), HL
FE4C 20D8	01100	JR	NZ, REF2	FEB2 DA3BFF	01760	JP	C, PACKE
FE4E D7	01110 JUMP2	RST	10H	FEB5 E5	01770	PUSH	HL
FE4F 2B	01120	DEC	HL	FEB6 7E	01780 CHAR	LD	A, (HL)
FE50 30D4	01130	JR	NC, REF2	FEB7 A7	01790	AND	A
FE52 AF	01140	XOR	A	FEB8 286F	01800	JR	Z, EOL
FE53 F5	01150 JUMP1	PUSH	AF	FEBA FE20	01810	CP	' '
FE54 CD5B1E	01160	CALL	DECLN1	FEBC 2824	01820	JR	Z, SPACE
FE57 2B	01170	DEC	HL	FEBE FE0A	01830	CP	OAH
FE58 F1	01180	POP	AF	FEC0 2820	01840	JR	Z, SPACE
FE59 D69F	01190	SUB	RESUME	FEC2 FE93	01850	CP	REM
FE5B B2	01200	OR	D	FEC4 282B	01860	JR	Z, DEL.REM
FE5C B3	01210	OR	E	FEC6 FEFF	01870	CP	OFFH
FE5D 28C7	01220	JR	Z, REF2	FEC8 2826	01880	JR	Z, DELR1
FE5F E5	01230	PUSH	HL	FECA FE88	01890	CP	DATA
FE60 CD2C1B	01240	CALL	SLINE	FECC 2839	01900	JR	Z, PDATA
FE63 301C	01250	JR	NC, UNDEFL	FECE FE3A	01910	CP	' : '
FE65 69	01260	LD	L, C	FED0 2829	01920	JR	Z, COLON
FE66 60	01270	LD	H, B	FED2 FE22	01930 CPSTR	CP	' "'
FE67 23	01280	INC	HL	FED4 23	01940 CHAR1	INC	HL
FE68 23	01290	INC	HL	FED5 20DF	01950	JR	NZ, CHAR
FE69 23	01300	INC	HL	FED7 2B	01960	DEC	HL
FE6A D7	01310	RST	10H	FED8 D7	01970 STR	RST	10H
FE6B FE3A	01320	CP	' ; '	FED9 A7	01980	AND	A
FE6D CC781D	01330	CALL	Z, CHRGET	FEDA 284D	01990	JR	Z, EOL
FE70 FE93	01340	CP	REM	FEDC D622	02000	SUB	' "'

FEDE 20FB	02010	JR	NZ,STR	FF4B 23	02670	INC	HL		
FEE0 18F0	02020	JR	CPSTR	FF49 23	02680	INC	HL		
FEE2 CDE7FE	02030	SPACE	CALL	USPACE	FF4A 7E	02690	LD	A, (HL)	
FEE5 18CF	02040	JR	CHAR	FF4B 3C	02700	INC	A		
FEE7 E5	02050	USPACE	PUSH	HL	FF4C 2002	02710	JR	NZ,NEXTL	
FEE8 D7	02060	RST	10H	FF4E 3693	02720	LD	(HL),REM		
FEE9 D1	02070	POP	DE	FF50 CD071F	02730	NEXTL	CALL	SEOL	
FEEA D5	02080	DEL	PUSH	DE	FF53 18EB	02740	JR	DELREF	
FEEB CD61FF	02090	CALL	DELETE	FF55 CD74FF	02750	END	CALL	LEN	
EEEE E1	02100	POP	HL	FF58 212919	02760	LD	HL,TEXT6		
FEFF C9	02110	RET		FF5B CD752B	02770	CALL	PRINTT		
FEF0 23	02120	DELRI	INC	HL	FF5E C3E81A	02780	JP	ENDML	
FEF1 E5	02130	DELREM	PUSH	HL	FF61 D5	02790	DELETE	PUSH	DE
FEF2 CD071F	02140	CALL	SEOL	FF62 EB	02800	EX	DE,HL		
FEF5 D1	02150	POP	DE	FF63 2AF940	02810	LD	HL,(VARA)		
FEF6 CDEAFE	02160	CALL	DEL	FF66 A7	02820	AND	A		
FEF9 18BB	02170	JR	CHAR	FF67 ED52	02830	SBC	HL,DE		
FEFB E5	02180	COLON	PUSH	HL	FF69 4D	02840	LD	C,L	
FEFC D7	02190	RST	10H	FF6A 44	02850	LD	B,H		
FEFD E1	02200	POP	HL	FF6B EB	02860	EX	DE,HL		
FEFE FE93	02210	CP	REM	FF6C D1	02870	POP	DE		
FF00 28E0	02220	JR	Z,SPACE	FF6D EDB0	02880	LDIR			
FF02 3C	02230	INC	A	FF6F ED53F940	02890	LD	(VARA),DE		
FF03 28DD	02240	JR	Z,SPACE	FF73 C9	02900	RET			
FF05 18CD	02250	JR	CHAR1	FF74 E5	02910	LEN	PUSH	HL	
FF07 23	02260	PDATA	INC	HL	FF75 21E4FF	02920	LD	HL,TEXT4	
FF08 7E	02270	LD	A, (HL)	FF78 CD752B	02930	CALL	PRINTT		
FF09 FE20	02280	CP	' '	FF7B 2AF940	02940	LD	HL,(VARA)		
FF0B CCE7FE	02290	CALL	Z,USPACE	FF7E ED5BA440	02950	LD	DE,(TEXTA)		
FF0E 2B	02300	DEC	HL	FF82 37	02960	SCF			
FF0F D7	02310	RST	10H	FF83 ED52	02970	SBC	HL,DE		
FF10 29A4	02320	JR	Z,CHAR	FF85 2B	02980	DEC	HL		
FF12 FE22	02330	CP	' "'	FF86 CDAFOF	02990	CALL	PRZNR		
FF14 200D	02340	JR	NZ,SC1	FF89 21F8FF	03000	LD	HL,TEXT5		
FF16 D7	02350	DS	RST	10H	FF8C CD752B	03010	CALL	PRINTT	
FF17 A7	02360	AND	A	FF8F E1	03020	POP	HL		
FF18 280F	02370	JR	Z,EOL	FF90 C9	03030	RET			
FF1A FE22	02380	CP	' "'	FF91 5A	03040	TEXT1	DEFM	←	
FF1C 20F8	02390	JR	NZ,DS	FFB3 0D	03050	DEFB	DEFB		
FF1E 18E7	02400	JR	PDATA	FFB4 00	03060	DEFB	DEFB		
FF20 D7	02410	SC	RST	10H	FFB5 50	03070	TEXT2	DEFM	
FF21 2893	02420	JR	Z,CHAR	FFC4 0D	03080	DEFB	DEFB		
FF23 FE2C	02430	SC1	CP	' , '	FFC5 00	03090	DEFB	DEFB	
FF25 20F9	02440	JR	NZ,SC	FFC6 42	03100	TEXT3	DEFM		
FF27 18DE	02450	JR	PDATA	FFE3 00	03110	DEFB	DEFB		
FF29 D1	02460	EDL	POP	DE	FFE4 50	03120	TEXT4	DEFM	
FF2A DF	02470	RST	18H	FFF7 00	03130	DEFB	DEFB		
FF2B C2A3FE	02480	JP	NZ,PACKL	FFF8 20	03140	TEXT5	DEFM		
FF2E 1B	02490	DEC	DE	FFFE 0D	03150	DEFB	DEFB		
FF2F 1B	02500	DEC	DE	FFFF 00	03160	DEFB	DEFB	← INIT	
FF30 1B	02510	DEC	DE	FDDD	03170	END			
FF31 1B	02520	DEC	DE	00000	TOTAL ERRORS				
FF32 D5	02530	PUSH	DE	30127	TEXT AREA BYTES LEFT				
FF33 23	02540	INC	HL						
FF34 CD61FF	02550	CALL	DELETE						
FF37 E1	02560	POP	HL						
FF3B C3A4FE	02570	JP	PACKL1						
FF3B D1	02580	PACKE	POP						
FF3C 2AA440	02590	LD	HL,(TEXTA)						
FF3F 2B	02600	DEC	HL						
FF40 23	02610	DELREF	INC						
FF41 7E	02620	LD	A,(HL)						
FF42 23	02630	INC	HL						
FF43 B6	02640	DR	(HL)						
FF44 CA55FF	02650	JP	Z,END						
FF47 23	02660	INC	HL						

'ZEILENREFERENZEN WERDEN ERMITTELT.'

ODH

0

'PACKEN BEGINNT.'

ODH

0

'BEZUG AUF UNDEFINIESTE ZEILE '

0

'PROGRAMM ERFORDERT '

0

' BYTES'

ODH

0

6.5 Feldaufzeichnung ARRAY (Kass.)- 127 -

FE76	00020	ORG	OFE76H	FEEB A7	00690	AND	A
FE76 2A7D41	00030	INIT	LD HL, (FIELDV)	FEE9 280E	00700	JR	Z, ABF
FE79 2290FE	00040	LD	(AAZ+1), HL	FEEB 0A	00710	ABE	LD A, (BC)
FE7C 2184FE	00050	LD	HL, AAY	FEEC CD6402	00720	CALL	WRITEB
FE7F 227D41	00060	LD	(FIELDV), HL	FEEF CDCBFF	00730	CALL	BREAK
FE82 AF	00070	XOR	A	FEF2 C2C2FF	00740	JP	NZ, RET2
FE83 C9	00080	RET		FEF5 03	00750	INC	BC
FE84 FE89	00100	AAY	CP INPUT	FEF6 15	00760	DEC	D
FE86 CA1EFF	00110	JP	Z, ABI	FEF7 20F2	00770	JR	NZ, ABE
FE89 FEB2	00120	CP	PRINT	FEF9 D1	00780	ABF	POP DE
FE8B 2805	00130	JR	Z, ABA	FEFA 1B	00790	DEC	DE
FE8D 2B	00140	DEC	HL	FEFB 1B	00800	DEC	DE
FE8E D7	00150	RST	10H	FEFC 1B	00810	DEC	DE
FE8F C30000	00160	AAZ	JP 0	FEFD 7A	00820	LD	A, D
FE92 D7	00170	ABA	RST 10H	FEFE B3	00830	OR	E
FE93 CD8402	00180	CALL	INITW	FEFF 20DC	00840	JR	NZ, ABD
FE96 2B	00190	DEC	HL	FF01 E1	00850	POP	HL
FE97 010000	00200	ABB	LD BC, 0	FF02 C315FF	00860	JP	ABH
FE9A CD6000	00210	CALL	DELAY	FF05 0A	00870	ABG	LD A, (BC)
FE9D CD6000	00220	CALL	DELAY	FF06 CD6402	00880	CALL	WRITEB
FEA0 D7	00230	RST	10H	FF09 CDCBFF	00890	CALL	BREAK
FEA1 CDD1FF	00240	CALL	ABR	FF0C C2C4FF	00900	JP	NZ, RET
FEA4 7B	00250	LD	A, E	FF0F 03	00910	INC	BC
FEA5 CD6402	00260	CALL	WRITEB	FF10 1B	00920	DEC	DE
FEA8 7A	00270	LD	A, D	FF11 7A	00930	LD	A, D
FEA9 CD6402	00280	CALL	WRITEB	FF12 B3	00940	OR	E
FEAC 3AAF40	00290	LD	A, (TYPX)	FF13 20F0	00950	JR	NZ, ABG
FEAF FE03	00300	CP	3	FF15 7E	00960	ABH	LD A, (HL)
FEB1 2052	00310	JR	NZ, ABG	FF16 FE2C	00970	CP	' , '
FEB3 E5	00320	PUSH	HL	FF18 CA97FE	00980	JP	Z, ABB
FEB4 C5	00330	PUSH	BC	FF1B C3FB01	00990	JP	CASOFF
FEB5 D5	00340	PUSH	DE	FF1E CD2B2B	01000	ABI	CALL 2B2B
FEB6 210000	00350	LD	HL, 0	FF21 D7	01010	RST	10H
FEB9 0A	00360	ABC	LD A, (BC)	FF22 CD9302	01020	CALL	INITR
FEBA 85	00370	ADD	A, L	FF25 2B	01030	DEC	HL
FEBB 6F	00380	LD	L, A	FF26 D7	01040	ABJ	RST 10H
FEBC 7C	00390	LD	A, H	FF27 CDD1FF	01050	CALL	ABR
FEBD CE00	00400	ADC	A, 0	FF2A CD3502	01060	CALL	READB
FEBF 67	00410	LD	H, A	FF2D BB	01070	CP	E
FEC0 1B	00420	DEC	DE	FF2E C23D27	01080	JP	NZ, BSERR
FEC1 1B	00430	DEC	DE	FF31 CD3502	01090	CALL	READB
FEC2 1B	00440	DEC	DE	FF34 BA	01100	CP	D
FEC3 03	00450	INC	BC	FF35 C23D27	01110	JP	NZ, BSERR
FEC4 03	00460	INC	BC	FF38 3AAF40	01120	LD	A, (TYPX)
FEC5 03	00470	INC	BC	FF3B FE03	01130	CP	3
FEC6 7A	00480	LD	A, D	FF3D 2066	01140	JR	NZ, ABP
FEC7 B3	00490	OR	E	FF3F E5	01150	PUSH	HL
FEC8 20EF	00500	JR	NZ, ABC	FF40 CD3502	01160	CALL	READB
FECA 7D	00510	LD	A, L	FF43 6F	01170	LD	L, A
FECB CD6402	00520	CALL	WRITEB	FF44 CD3502	01180	CALL	READB
FECE 7C	00530	LD	A, H	FF47 67	01190	LD	H, A
FECF CD6402	00540	CALL	WRITEB	FF48 CDE9FF	01200	CALL	ABS
FED2 010000	00550	LD	BC, 0	FF4B 381A	01210	JR	C, ABL
FED5 CD6000	00560	CALL	DELAY	FF4D CDF801	01220	CALL	CASOFF
FED8 CD6000	00570	CALL	DELAY	FF50 C5	01230	PUSH	BC
FEDB D1	00580	POP	DE	FF51 D5	01240	PUSH	DE
FEDC E1	00590	POP	HL	FF52 AF	01250	ABK	XOR A
FEDD D5	00600	ABD	PUSH DE	FF53 02	01260	LD	(BC), A
FEDE 56	00610	LD	D, (HL)	FF54 03	01270	INC	BC
FEDF 23	00620	INC	HL	FF55 1B	01280	DEC	DE
FEE0 4E	00630	LD	C, (HL)	FF56 7A	01290	LD	A, D
FEE1 23	00640	INC	HL	FF57 B3	01300	OR	E
FEE2 46	00650	LD	B, (HL)	FF58 20F8	01310	JR	NZ, ABK
FEE3 23	00660	INC	HL	FF5A E5	01320	PUSH	HL
FEE4 7A	00670	LD	A, D	FF5B CDE62B	01330	CALL	COLLEC
FEE5 CD6402	00680	CALL	WRITEB	FF5E E1	01340	POP	HL

FF5F D1	01350	POP	DE	FFD0 C9	02010	RET	
FF60 CDE9FF	01360	CALL	ABS	FFD1 CD0D26	02020	ABR	CALL SVAR
FF63 C1	01370	POP	BC	FFD4 E5	02030		PUSH HL
FF64 D2FBFF	01380	JP	NC, ABT	FFD5 2AFB40	02040		LD HL, (ARRTAB)
FF67 CD1502	01390	ABL	CALL	FFD8 ED52	02050		SBC HL, DE
FF6A 69	01400	LD	L, C	FFDA D2F60A	02060		JP NC, TMERR
FF6B 60	01410	LD	H, B	FFDD 3AAF40	02070		LD A, (TYPX)
FF6C CD2C02	01420	ABM	CALL	FFE0 6F	02080		LD L, A
FF6F D5	01430	PUSH	DE	FFE1 2600	02090		LD H, 0
FF70 CD3502	01440	CALL	READB	FFE3 19	02100		ADD HL, DE
FF73 77	01450	LD	(HL), A	FFE4 ED42	02110		SBC HL, BC
FF74 23	01460	INC	HL	FFE6 EB	02120		EX DE, HL
FF75 E5	01470	PUSH	HL	FFE7 E1	02130		POP HL
FF76 2AD640	01480	LD	HL, (STRZG)	FFE8 C9	02140		RET
FF79 4F	01490	LD	C, A	FFE9 D5	02150	ABS	PUSH DE
FF7A 0600	01500	LD	B, 0	FFEA E5	02160		PUSH HL
FF7C A7	01510	AND	A	FFEB 2AD640	02170		LD HL, (STRZG)
FF7D ED42	01520	SBC	HL, BC	FFEE A7	02180		AND A
FF7F 22D640	01530	LD	(STRZG), HL	FFEF ED5BA040	02190		LD DE, (STRSPC)
FF82 23	01540	INC	HL	FFF3 ED52	02200		SBC HL, DE
FF83 EB	01550	EX	DE, HL	FFF5 EB	02210		EX DE, HL
FF84 E1	01560	POP	HL	FFF6 13	02220		INC DE
FF85 73	01570	LD	(HL), E	FFF7 E1	02230		POP HL
FF86 23	01580	INC	HL	FFF8 DF	02240		RST 18H
FF87 72	01590	LD	(HL), D	FFF9 D1	02250		POP DE
FF88 23	01600	INC	HL	FFFA C9	02260		RET
FF89 A7	01610	AND	A	FFFB 1E1A	02270	ABT	LD E, OSERR
FF8A 280E	01620	JR	Z, ABO	FFFD C3A219	02280		JP ERROR
FF8C CD3502	01630	ABN	CALL	001A	02290	OSERR	EQU 001AH
FF8F 12	01640	LD	(DE), A	0060	02300	DELAY	EQU 0060H
FF90 CDCBFF	01650	CALL	BREAK	0089	02310	INPUT	EQU 0089H
FF93 C2C2FF	01660	JP	NZ, RET2	00B2	02320	PRINT	EQU 00B2H
FF96 13	01670	INC	DE	01F8	02330	CASOFF	EQU 01F8H
FF97 0D	01680	DEC	C	0215	02340	CASON	EQU 0215H
FF98 20F2	01690	JR	NZ, ABN	022C	02350	BLINK	EQU 022CH
FF9A D1	01700	ABO	POP	0235	02360	READB	EQU 0235H
FF9B 1B	01710	DEC	DE	0264	02370	WRITEB	EQU 0264H
FF9C 1B	01720	DEC	DE	0284	02380	INITW	EQU 0284H
FF9D 1B	01730	DEC	DE	0293	02390	ININTR	EQU 0293H
FF9E 7B	01740	LD	A, E	0AF6	02400	TMERR	EQU 0AF6H
FF9F B2	01750	OR	D	19A2	02410	ERROR	EQU 19A2H
FFA0 20CA	01760	JR	NZ, ABM	260D	02420	SVAR	EQU 260DH
FFA2 E1	01770	POP	HL	273D	02430	BSERR	EQU 273DH
FFA3 1814	01780	JR	ABO	28E6	02440	COLLEC	EQU 28E6H
FFA5 CD3502	01790	ABP	CALL	40A0	02450	STRSPC	EQU 40A0H
FFA8 02	01800	LD	(BC), A	40AF	02460	TYPX	EQU 40AFH
FFA9 A7	01810	AND	A	40D6	02470	STRZG	EQU 40D6H
FFAA CC2C02	01820	CALL	Z, BLINK	40FB	02480	ARRTAB	EQU 40FBH
FFAD CDCBFF	01830	CALL	BREAK	417D	02490	FIELDV	EQU 417DH
FFB0 C2C4FF	01840	JP	NZ, RET	FE76	02500		END INIT
FFB3 03	01850	INC	BC	00000	TOTAL ERRORS		
FFB4 1B	01860	DEC	DE	31621	TEXT AREA BYTES LEFT		
FFB5 7A	01870	LD	A, D				
FFB6 B3	01880	OR	E				
FFB7 20EC	01890	JR	NZ, ABP				
FFB9 7E	01900	ABO	LD				
FFBA FE2C	01910		A, (HL)				
FFBC CA26FF	01920	CP	' , '				
FFBF C3F801	01930	JP	Z, ABJ				
FFC2 D1	01940	RET2	POP				
FFC3 E1	01950	POP	DE				
FFC4 CDF801	01960	RET	CALL				
FFC7 2AE640	01970	LD	HL, (40E6H)				
FFCA C9	01980	RET					
FFCB 3A4038	01990	BREAK	LD				
FFCE CB57	02000	BIT	A, (3840H)				
			Z, A				

Feldaufzeichnung		ARRAY (Disk)	- 129 -					
FE9F	00010	ORG	0FE9FH		FEFB CD2A03	00670	CALL	OUTCH
	00020				FEFE A7	00680	AND	A
FE9F 2AE667	00030	INIT	LD	HL, (FIELDV)	FEFF 290B	00690	JR	Z, ABF
FEA2 22B8FE	00040		LD	(AAZ+1), HL	FF01 0A	00700	LD	A, (BC)
FEA5 21ACFE	00050		LD	HL, AAY	FF02 CD2A03	00710	CALL	OUTCH
FEA8 22E667	00060		LD	(FIELDV), HL	FF05 03	00720	INC	BC
FEAB C9	00070		RET		FF06 15	00730	DEC	D
4004	00080	RST10V	EQU	4004H	FF07 20FB	00740	JR	NZ, ABF
41CA	00090	WDISK	EQU	41CAH	FF09 D1	00750	POP	DE
41BE	00100	WDISKE	EQU	41BEH	FF0A 1B	00760	DEC	DE
032A	00110	OUTCH	EQU	032AH	FF0B 1B	00770	DEC	DE
0013	00120	INCHW	EQU	0013H	FF0C 1B	00780	DEC	DE
5F7B	00130	BUFFER	EQU	5F7BH	FF0D 7A	00790	LD	A, D
	00140				FF0E B3	00800	OR	E
FEAC FEB9	00150	AAY	CP	INPUT	FF0F 20E2	00810	JR	NZ, ABD
FEAE CA28FF	00160		JP	Z, ABI	FF11 E1	00820	POP	HL
FEB1 FEB2	00170		CP	PRINT	FF12 C31FFF	00830	JP	ABH
FEB3 2B05	00180		JR	Z, ABA	FF15 0A	00840	LD	A, (BC)
FEB5 2B	00190		DEC	HL	FF16 CD2A03	00850	CALL	OUTCH
FEB6 D7	00200		RST	10H	FF19 03	00860	INC	BC
FEB7 C30000	00210	AAZ	JP	0	FF1A 1B	00870	DEC	DE
FEBA D7	00220	ABA	RST	10H	FF1B 7A	00880	LD	A, D
FEBC CDCA41	00230		CALL	WDISK	FF1C B3	00890	OR	E
FEBE 2B	00240		DEC	HL	FF1D 20F6	00900	JR	NZ, ABG
FEBF D7	00250	ABB	RST	10H	FF1F 7E	00910	LD	A, (HL)
FEC0 CDBBFF	00260		CALL	SARRAY	FF20 FE2C	00920	CP	' , '
FEC3 7B	00270		LD	A, E	FF22 CABFFE	00930	JP	Z, ABF
FEC4 CD2A03	00280		CALL	OUTCH	FF25 C3BE41	00940	JP	WDISKE
FEC7 7A	00290		LD	A, D	FF28 CD2828	00950	CALL	2828H
FEC8 CD2A03	00300		CALL	OUTCH	FF2B CD7B5F	00960	CALL	BUFFER
FECB 3AAF40	00310		LD	A, (TYPX)	FF2E CF	00970	RST	8H
FECE FE03	00320		CP	3	FF2F 2C	00980	DEFB	' , '
FED0 2043	00330		JR	NZ, ABG	FF30 E5	00990	PUSH	HL
FED2 E5	00340		PUSH	HL	FF31 210D00	01000	LD	HL, 13
FED3 C5	00350		PUSH	BC	FF34 09	01010	ADD	HL, BC
FED4 D5	00360		PUSH	DE	FF35 22FEFF	01020	LD	(FCB), HL
FED5 210000	00370		LD	HL, 0	FF38 E1	01030	POP	HL
FED8 0A	00380	ABC	LD	A, (BC)	FF39 2B	01040	DEC	HL
FED9 85	00390		ADD	A, L	FF3A D7	01050	RST	10H
FEDA 6F	00400		LD	L, A	FF3B CDBBFF	01060	CALL	SARRAY
FEDB 7C	00410		LD	A, H	FF3E CDEAFF	01070	CALL	READB
FEDC CE00	00420		ADC	A, 0	FF41 BB	01080	CP	E
FEDE 67	00430		LD	H, A	FF42 C23D27	01090	JP	NZ, BSERR
FEDF 1B	00440		DEC	DE	FF45 CDEAFF	01100	CALL	READB
FEE0 1B	00450		DEC	DE	FF48 BA	01110	CP	D
FEE1 1B	00460		DEC	DE	FF49 C23D27	01120	JP	NZ, BSERR
FEE2 03	00470		INC	BC	FF4C 3AAF40	01130	LD	A, (TYPX)
FEE3 03	00480		INC	BC	FF4F FE03	01140	CP	3
FEE4 03	00490		INC	BC	FF51 2057	01150	JR	NZ, ABF
FEE5 7A	00500		LD	A, D	FF53 E5	01160	PUSH	HL
FEE6 B3	00510		OR	E	FF54 CDEAFF	01170	CALL	READB
FEE7 20EF	00520		JR	NZ, ABC	FF57 6F	01180	LD	L, A
FEE9 7D	00530		LD	A, L	FF5B CDEAFF	01190	CALL	READB
FEEA CD2A03	00540		CALL	OUTCH	FF5B 67	01200	LD	H, A
FEED 7C	00550		LD	A, H	FF5C CDD3FF	01210	CALL	ABS
FEED CD2A03	00560		CALL	OUTCH	FF5F 3817	01220	JR	C, ABL
FEF1 D1	00570		POP	DE	FF61 C5	01230	PUSH	BC
FEF2 E1	00580		POP	HL	FF62 D5	01240	PUSH	DE
FEF3 D5	00590	ABD	PUSH	DE	FF63 AF	01250	XOR	A
FEF4 56	00600		LD	D, (HL)	FF64 02	01260	LD	(BC), A
FEF5 23	00610		INC	HL	FF65 03	01270	INC	BC
FEF6 4E	00620		LD	C, (HL)	FF66 1B	01280	DEC	DE
FEF7 23	00630		INC	HL	FF67 7A	01290	LD	A, D
FEF8 46	00640		LD	B, (HL)	FF68 B3	01300	OR	E
FEF9 23	00650		INC	HL	FF69 20FB	01310	JR	NZ, ABK
FEFA 7A	00660		LD	A, D	FF6B E5	01320	PUSH	HL

FF6C CDE628	01330	CALL	COLLEC	FFD3 D5	01990	ABS	PUSH	DE
FF6F E1	01340	POP	HL	FFD4 E5	02000		PUSH	HL
FF70 D1	01350	FOP	DE	FFD5 2AD640	02010		LD	HL, (STRZG)
FF71 CDD3FF	01360	CALL	ABS	FFD8 A7	02020		AND	A
FF74 C1	01370	POP	BC	FFD9 EDSBA040	02030		LD	DE, (STRSPC)
FF75 D2E5FF	01380	JP	NC, ABT	FFDD ED52	02040		SBC	HL, DE
FF78 69	01390	ABL	LD	FFDF EB	02050		EX	DE, HL
FF79 60	01400		LD	FFE0 13	02060		INC	DE
FF7A D5	01410	ABM	PUSH	FFE1 E1	02070		POP	HL
FF7B CDEAFF	01420		CALL	FFE2 DF	02080		RST	18H
FF7E 77	01430		LD	FFE3 D1	02090		POP	DE
FF7F 23	01440		INC	FFE4 C9	02100		RET	
FF80 E5	01450		PUSH	FFE5 1E1A	02110	ABT	LD	E, OSERR
FF81 2AD640	01460		LD	FFE7 C3A219	02120		JP	ERROR
FF84 4F	01470		LD	FFEA D5	02130	READB	PUSH	DE
FF85 0600	01480		LD	FFEB ED5BFEFF	02140		LD	DE, (FCB)
FF87 A7	01490		AND	FFEF CD1300	02150		CALL	INCHW
FF88 ED42	01500		SBC	FFF2 D1	02160		POP	DE
FF8A 22D640	01510		LD	FFF3 C8	02170		RET	Z
FF8D 23	01520		INC	FFF4 CBFF	02180		SET	7, A
FF8E EB	01530		EX	FFF6 CD0944	02190		CALL	4409H
FF8F E1	01540		POP	FFF9 1E66	02200		LD	E, 102
FF90 73	01550		LD	FFFB C3A219	02210		JP	ERROR
FF91 23	01560		INC	FFFE 0000	02220	FCB	DEFW	0
FF92 72	01570		LD		02230		;	
FF93 23	01580		INC	001A	02240	OSERR	EQU	001AH
FF94 A7	01590		AND	0089	02250	INPUT	EQU	0089H
FF95 2808	01600		JR	00B2	02260	PRINT	EQU	00B2H
FF97 CDEAFF	01610	ABN	CALL	0AF6	02270	TMERR	EQU	0AF6H
FF9A 12	01620		LD	19A2	02280	ERROR	EQU	19A2H
FF9B 13	01630		INC	260D	02290	SVAR	EQU	260DH
FF9C 0D	01640		DEC	273D	02300	BSERR	EQU	273DH
FF9D 20F8	01650		JR	28E6	02310	COLLEC	EQU	28E6H
FF9F D1	01660	ABO	POP	40A0	02320	STRSPC	EQU	40A0H
FFA0 1B	01670		DEC	40AF	02330	TYPX	EQU	40AFH
FFA1 1B	01680		DEC	40D6	02340	STRZG	EQU	40D6H
FFA2 1B	01690		DEC	40FB	02350	ARRTAB	EQU	40FBH
FFA3 7B	01700		LD	67E6	02360	FIELDV	EQU	67E6H
FFA4 B2	01710		OR	FE9F	02370		END	INIT
FFA5 20D3	01720		JR	00000	TOTAL	ERRORS		
FFA7 E1	01730		POP	31840	TEXT	AREA	BYTES	LEFT
FFA8 180A	01740		JR					
FFAA CDEAFF	01750	ABP	CALL					
FFAD 02	01760		LD					
FFAE 03	01770		INC					
FFAF 1B	01780		DEC					
FFB0 7A	01790		LD					
FFB1 B3	01800		OR					
FFB2 20F6	01810		JR					
FFB4 7E	01820	ABQ	LD					
FFB5 FE2C	01830		CP					
FFB7 CA3AFF	01840		JP					
FFBA C9	01850		RET					
FFBB CD0D26	01860	SARRAY	CALL					
FFBE E5	01870		PUSH					
FFBF 2AFB40	01880		LD					
FFC2 ED52	01890		SBC					
FFC4 D2F60A	01900		JP					
FFC7 3AAF40	01910		LD					
FFCA 6F	01920		LD					
FFCB 2600	01930		LD					
FFCD 19	01940		ADD					
FFCE ED42	01950		SBC					
FFD0 EB	01960		EX					
FFD1 E1	01970		POP					
FFD2 C9	01980		RET					

6.6 Umlauterweiterung für SCRIPSIT - 131 -
 00010 ; ** SCHREIBMASCHINENANSTEUERUNG **

```

00020 ; VON SCRIPSIT
4049 00030 ORG 4049H
4049 6BFF 00040 DEFW ANF-1
FF6C 00050 ORG OFF&CH
FF6C E5 00060 ANF PUSH HL
FF6D C5 00070 PUSH BC
FF6E F5 00080 PUSH AF
FF6F 21A FF 00090 LD HL,FLAG
FF72 7E 00100 LD A,(HL)
FF73 A7 00110 AND A
FF74 2020 00120 JR NZ,HEXOUT
FF76 F1 00130 POP AF
FF77 F5 00140 PUSH AF
FF78 FE7F 00150 CP 7FH
FF7A 2B16 00160 JR Z,INIT
FF7C 21E2 FF 00170 LD HL,NEU+2
FF7F 010F00 00180 LD BC,DRUCK-NEU-2
FF82 EDB1 00190 CPIR
FF82 00200 ORG OFF82H
FF82 EDB1 00210 CPIR
FF84 2005 00220 JR NZ,PRINT
FF86 010E00 00230 LD BC,DRUCK-NEU-1-2
FF89 09 00240 ADD HL,BC
FF8A 7E 00250 LD A,(HL)
FF8B 32EB37 00260 PRINT LD (37E8H),A
FF8E F1 00270 RET POP AF
FF8F C1 00280 POP BC
FF90 E1 00290 POP HL
FF91 C9 00300 RET
FF92 3602 00310 INIT LD (HL),2
FF94 18FB 00320 JR RET
FF96 F1 00330 HEXOUT POP AF
FF97 F5 00340 PUSH AF
FF98 FE3A 00350 CP '9'+1
FF9A 3802 00360 JR C,HO1
FF9C D607 00370 SUB 7
FF9E D630 00380 HO1 SUB '0'
FFA0 23 00390 INC HL
FFA1 ED6F 00400 RLD
FFA3 7E 00410 LD A,(HL)
FFA4 2B 00420 DEC HL
FFA5 35 00430 DEC (HL)
FFA6 20E6 00440 JR NZ,RET
FFAB 18E1 00450 JR PRINT
FFAA 00 00460 FLAG DEFB 0
FFAB 00 00470 DEFB 0
FFAC 21CFFF 00480 LD HL,ALT
FFAF 011100 00490 LD BC,NEU-ALT ;BC=LAENGE DER TABELLE
FFB2 EDB1 00500 CPIR
FFB4 200D 00510 JR NZ,NORMAL
FFB6 011000 00520 LD BC,NEU-ALT-1
FFB9 09 00530 ADD HL,BC
FFBA 7E 00540 LD A,(HL) ;UMCODIERTES ZEICHEN IN D LADEN
FFBB FE7E 00550 CP 7EH
FFBD DA6761 00560 JP C,6167H
FFC0 C37961 00570 JP 6179H
FFC3 FE41 00580 NORMAL CP 'A'
FFC5 DA6761 00590 JP C,6167H
FFC8 C36161 00600 JP 6161H
FFCB 3AEB37 00610 BUSY LD A,(37E8H)
FFCE C9 00620 RET
FFCF 50 00630 ALT DEFB 'P'
FFD0 70 00640 DEFB 70H
FFD1 41 00650 DEFB 41H ;A
FFD2 4F 00660 DEFB 4FH ;0
    
```

FFD3	55	00670	DEFB	55H	;U
FFD4	61	00680	DEFB	61H	;KLEIN A
FFD5	6F	00690	DEFB	6FH	;KLEIN O
FFD6	75	00700	DEFB	75H	;KLEIN U
FFD7	3A	00710	DEFB	':'	
FFD8	00	00720	DEFB	0	
FFD9	00	00730	DEFB	0	
FFDA	00	00740	DEFB	0	
FFDB	00	00750	DEFB	0	
FFDC	00	00760	DEFB	0	
FFDD	00	00770	DEFB	0	
FFDE	00	00780	DEFB	0	
FFDF	00	00790	DEFB	0	
FFE0	7F	00800 NEU	DEFB	7FH	
FFE1	7F	00810	DEFB	7FH	
FFE2	5B	00820	DEFB	5BH	;AE
FFE3	5C	00830	DEFB	5CH	;OE
FFE4	5D	00840	DEFB	5DH	;UE
FFE5	7B	00850	DEFB	7BH	;KLEIN AE
FFE6	7C	00860	DEFB	7CH	;KLEIN OE
FFE7	7D	00870	DEFB	7DH	;KLEIN UE
FFE8	7E	00880	DEFB	7EH	;SZ
FFE9	00	00890	DEFB	0	
FFEA	00	00900	DEFB	0	
FFEB	00	00910	DEFB	0	
FFEC	00	00920	DEFB	0	
FFED	00	00930	DEFB	0	
FFEE	00	00940	DEFB	0	
FFEF	00	00950	DEFB	0	
FFF0	00	00960	DEFB	0	
FFF1	5B	00970 DRUCK	DEFB	5BH	;AE
FFF2	5C	00980	DEFB	5CH	;OE
FFF3	5D	00990	DEFB	5DH	;UE
FFF4	7B	01000	DEFB	7BH	;KLEIN AE
FFF5	7C	01010	DEFB	7CH	
FFF6	7D	01020	DEFB	7DH	
FFF7	7E	01030	DEFB	7EH	
FFF8	00	01040	DEFB	0	
FFF9	00	01050	DEFB	0	
FFFA	00	01060	DEFB	0	
FFFB	00	01070	DEFB	0	
FFFC	00	01080	DEFB	0	
FFFD	00	01090	DEFB	0	
FFFE	00	01100	DEFB	0	
FFFF	00	01110	DEFB	0	
5200		01120	END	5200H	

00000 TOTAL ERRORS

33416 TEXT AREA BYTES LEFT

6,7 Assembler/Disassembler Z80AD Kass., Maschinenprogramm						- 133 -	
42C0	00010	ORG	42C0H	4353 13	00670	ADX	INC DE
42C0	211E1D	00020	START LD HL,1D1EH	4354 23	00680		INC HL
42C3	E5	00030	PUSH HL	4355 1A	00690		LD A,(DE)
42C4	C35D1B	00040	JF 1B5DH	4356 96	00700		SUB (HL)
40A4	00050	ORG	40A4H	4357 CB27	00710		SLA A
40A4	694B	00060	DEFW AJU+1	4359 2810	00720		JR Z,ADZ
41A4	00070	ORG	41A4H	435B 1A	00730	ADY	LD A,(DE)
41A4	0B4B	00080	DEFW LINE	435C 3C	00740		INC A
42E9	00090	ORG	42E9H	435D 13	00750		INC DE
0008	00100	MEM	DEFS 8	435E 20FB	00760		JR NZ,ADY
42F1	2B	00110	ADM DEC HL	4360 1A	00770		LD A,(DE)
42F2	23	00120	ADN INC HL	4361 3C	00780		INC A
42F3	7E	00130	LD A,(HL)	4362 E1	00790		POP HL
42F4	FE20	00140	CP AAO	4363 1B	00800		DEC DE
42F6	2BFA	00150	JR Z,ADN	4364 20EC	00810		JR NZ,ADW
42F8	A7	00160	AND A	4366 DD360402	00820		LD (IX+04H),AAC
42F9	C9	00170	RET	436A C9	00830		RET
42FA	DDCB057E	00180	ADD BIT 07H,(IX+05H)	436B 30E6	00840	ADZ	JR NC,ADX
42FE	062C	00190	LD B,AAT	436D 23	00850		INC HL
4300	48	00200	LD C,B	436E 7E	00860		LD A,(HL)
4301	C8	00210	RET Z	436F FE20	00870		CP AAO
4302	063B	00220	LD B,AAY	4371 2809	00880		JR Z,AEA
4304	0E00	00230	LD C,AAA	4373 FE3B	00890		CP AAY
4306	C9	00240	RET	4375 2805	00900		JR Z,AEA
4307	CD7F0A	00250	ADP CALL ACD	4377 18E2	00910		JR ADY
430A	23	00260	INC HL	4379 00	00920		NOP
430B	5E	00270	LD E,(HL)	437A 00	00930		NOP
430C	23	00280	INC HL	437B 00	00940		NOP
430D	56	00290	LD D,(HL)	437C C1	00950	AEA	POP BC
430E	EB	00300	EX DE,HL	437D 13	00960		INC DE
430F	22EF42	00310	LD (ADL),HL	437E DD7E01	00970	AEB	LD A,(IX+01H)
4312	DD21E942	00320	LD IX,ADH	4381 E6E0	00980		AND ABU
4316	97	00330	SUB A	4383 DD7701	00990		LD (IX+01H),A
4317	DD7700	00340	LD (IX+00H),A	4386 E5	01000		PUSH HL
431A	DD7701	00350	LD (IX+01H),A	4387 1A	01010		LD A,(DE)
431D	DD7704	00360	LD (IX+04H),A	4388 07	01020		RLCA
4320	E5	00370	PUSH HL	4389 3818	01030		JR C,AEE
4321	47	00380	LD B,A	438B DDCB05BE	01040		RES 07H,(IX+05H)
4322	7E	00390	ADQ LD A,(HL)	438F CDE943	01050		CALL AEH
4323	23	00400	INC HL	4392 13	01060		INC DE
4324	FE3A	00410	CP AAX	4393 200B	01070		JR NZ,AED
4326	2016	00420	JR NZ,ADT	4395 13	01080	AEC	INC DE
4328	D1	00430	POP DE	4396 13	01090		INC DE
4329	78	00440	LD A,B	4397 1A	01100		LD A,(DE)
432A	FE07	00450	CP AAH	4398 3C	01110		INC A
432C	3003	00460	JR NC,ADR	4399 E1	01120		POP HL
432E	A7	00470	AND A	439A 20E2	01130		JR NZ,AEB
432F	2005	00480	JR NZ,ADS	439C DD7104	01140		LD (IX+04H),C
4331	DD360405	00490	ADR LD (IX+04H),AAF	439F C9	01150		RET
4335	C9	00500	RET	43A0 DD7700	01160	AED	LD (IX+00H),A
4336	0F	00510	ADS RRCA	43A3 DDCB05FE	01170	AEE	SET 07H,(IX+05H)
4337	0F	00520	RRCA	43A7 CDE943	01180		CALL AEH
4338	0F	00530	RRCA	43AA 28E9	01190		JR Z,AEC
4339	DD7701	00540	LD (IX+01H),A	43AC C1	01200		POP BC
433C	1806	00550	JR ADU	43AD DD8600	01210		ADD A,(IX+00H)
433E	04	00560	ADT INC B	43B0 47	01220		LD B,A
433F	FE20	00570	CP AAO	43B1 13	01230		INC DE
4341	20DF	00580	JR NZ,ADQ	43B2 1A	01240		LD A,(DE)
4343	E1	00590	POP HL	43B3 80	01250		ADD A,B
4344	CDF142	00600	ADU CALL ADM	43B4 DD7700	01260		LD (IX+00H),A
4347	2005	00610	JR NZ,ADV	43B7 DD7E01	01270		LD A,(IX+01H)
4349	DD360401	00620	LD (IX+04H),AAB	43BA E603	01280		AND AAD
434D	C9	00630	RET	43BC FE03	01290		CP AAD
434E	2B	00640	ADV DEC HL	43BE 2005	01300		JR NZ,AEF
434F	11C144	00650	LD DE,AFG	43C0 DD360403	01310		LD (IX+04H),AAD
4352	E5	00660	ADW PUSH HL	43C4 C9	01320		RET

43C5 1B	01330 AEF	DEC	DE	4441 2C	01990	INC	L
43C6 1A	01340	LD	A, (DE)	4442 DD7502	02000	LD	(IX+02H), L
43C7 E660	01350	AND	ABG	4445 E1	02010	POP	HL
43C9 FE40	01360	CP	ABB	4446 E5	02020	PUSH	HL
43CB 0604	01370	LD	B, AAE	4447 1A	02030	LD	A, (DE)
43CD 280D	01380	JR	Z, AEG	4448 07	02040	RLCA	
43CF FE20	01390	CP	AAD	4449 385C	02050	JR	C, AFC
43D1 0603	01400	LD	B, AAD	444B 13	02060	INC	DE
43D3 2807	01410	JR	Z, AEG	444C 23	02070 AET	INC	HL
43D5 A7	01420	AND	A	444D 7E	02080	LD	A, (HL)
43D6 0607	01430	LD	B, AAH	444E FE00	02090	CP	AAA
43D8 2002	01440	JR	NZ, AEG	4450 2866	02100	JR	Z, AFE
43DA 0600	01450	LD	B, AAA	4452 1A	02110	LD	A, (DE)
43DC DD7E01	01460 AEG	LD	A, (IX+01H)	4453 E67F	02120	AND	ABJ
43DF B0	01470	OR	B	4455 BE	02130	CP	(HL)
43E0 DD7701	01480	LD	(IX+01H), A	4456 20F4	02140	JR	NZ, AET
43E3 2AE942	01490	LD	HL, (ADH)	445B C1	02150	POP	BC
43E6 C39A0A	01500	JP	ACP	4459 97	02160 AEU	SUB	A
43E9 D5	01510 AEH	PUSH	DE	445A E5	02170	PUSH	HL
43EA 1A	01520	LD	A, (DE)	445B ED42	02180	SBC	HL, BC
43EB E61F	01530	AND	AAN	445D DD7503	02190	LD	(IX+03H), L
43ED 47	01540	LD	B, A	4460 E1	02200	POP	HL
43EE 2013	01550	JR	NZ, AEM	4461 182D	02210	JR	AFB
43F0 CDF142	01560	CALL	ADM	4463 FE58	02220 AEV	CP	ABF
43F3 2003	01570	JR	NZ, AEJ	4465 2011	02230	JR	NZ, AEX
43F5 04	01580 AEI	INC	B	4467 BE	02240	CP	(HL)
43F6 D1	01590	POP	DE	4468 2006	02250	JR	NZ, AEW
43F7 C9	01600	RET		446A DDCB01C6	02260	SET	00H, (IX+01H)
43F8 D63B	01610 AEJ	SUB	AAY	446E 1808	02270	JR	AEX
43FA 28F9	01620	JR	Z, AEI	4470 3C	02280 AEW	INC	A
43FC 0E01	01630 AEK	LD	C, AAB	4471 BE	02290	CP	(HL)
43FE 0601	01640 AEL	LD	B, AAB	4472 2007	02300	JR	NZ, AEY
4400 05	01650	DEC	B	4474 DDCB01CE	02310	SET	01H, (IX+01H)
4401 D1	01660	POP	DE	4478 BE	02320 AEX	CP	(HL)
4402 C9	01670	RET		4479 2811	02330	JR	Z, AEZ
4403 11E846	01680 AEM	LD	DE, AFN	447B 1A	02340 AEY	LD	A, (DE)
4406 13	01690 AEN	INC	DE	447C 07	02350	RLCA	
4407 1A	01700	LD	A, (DE)	447D 13	02360	INC	DE
4408 3C	01710	INC	A	447E 30FB	02370	JR	NC, AEY
4409 20FB	01720	JR	NZ, AEN	4480 13	02380	INC	DE
440B 10F9	01730	DJNZ	AEN	4481 1A	02390	LD	A, (DE)
440D CDF142	01740	CALL	ADM	4482 E1	02400	POP	HL
4410 2B	01750	DEC	HL	4483 3C	02410	INC	A
4411 0E0A	01760	LD	C, AAJ	4484 1B	02420	DEC	DE
4413 28E9	01770	JR	Z, AEL	4485 208E	02430	JR	NZ, AEO
4415 E5	01780 AEO	PUSH	HL	4487 0E03	02440	LD	C, AAD
4416 13	01790 AEP	INC	DE	4489 C3FE43	02450	JP	AEL
4417 23	01800	INC	HL	448C 1A	02460 AEZ	LD	A, (DE)
4418 1A	01810	LD	A, (DE)	448D 07	02470	RLCA	
4419 E67F	01820	AND	ABJ	448E 3086	02480	JR	NC, AEP
441B FE23	01830	CP	AAP	4490 23	02490 AFA	INC	HL
441D 200A	01840	JR	NZ, AEG	4491 C1	02500	POP	BC
441F DD7E01	01850	LD	A, (IX+01H)	4492 CDF142	02510	CALL	ADM
4422 C608	01860	ADD	A, AAI	4495 CDFA42	02520	CALL	ADO
4424 DD7701	01870	LD	(IX+01H), A	4498 7E	02530	LD	A, (HL)
4427 1810	01880	JR	AES	4499 90	02540	SUB	B
4429 FE24	01890 AEO	CP	AAQ	449A 2805	02550	JR	Z, AFB
442B 2808	01900	JR	Z, AER	449C 7E	02560	LD	A, (HL)
442D FE25	01910	CP	AAR	449D 91	02570	SUB	C
442F 2032	01920	JR	NZ, AEV	449E C2FC43	02580	JP	NZ, AEK
4431 DDCB01DE	01930	SET	03H, (IX+01H)	44A1 3C	02590 AFB	INC	A
4435 DDCB01E6	01940 AER	SET	04H, (IX+01H)	44A2 13	02600	INC	DE
4439 E5	01950 AES	PUSH	HL	44A3 23	02610	INC	HL
443A 97	01960	SUB	A	44A4 1A	02620	LD	A, (DE)
443B ED4BEF42	01970	LD	BC, (ADL)	44A5 D1	02630	POP	DE
443F ED42	01980	SBC	HL, BC	44A6 C9	02640	RET	

44A7	CDFA42	02650	AFC	CALL	ADD	44FF	40	03310	DEFM	'S'
44AA	7E	02660	AFD	LD	A, (HL)	4500	0D	03320	DEFB	0DH
44AB	FE20	02670		CP	AA0	4501	89	03330	DEFB	89H
44AD	280E	02680		JR	Z, AFF	4502	F9	03340	DEFB	0F9H
44AF	BB	02690		CP	B	4503	09	03350	DEFB	9H
44B0	280B	02700		JR	Z, AFF	4504	97	03360	DEFB	97H
44B2	B9	02710		CP	C	4505	2A	03370	DEFM	'*'
44B3	2808	02720		JR	Z, AFF	4506	17	03380	DEFB	17H
44B5	23	02730		INC	HL	4507	89	03390	DEFB	89H
44B6	18F2	02740		JR	AFD	4508	22	03400	DEFM	'''
44B8	C1	02750	AFE	POP	BC	4509	01	03410	DEFB	1H
44B9	C1	02760		POP	BC	450A	97	03420	DEFB	97H
44BA	C3FC43	02770		JP	AEK	450B	3A	03430	DEFM	'.'
44BD	2B	02780	AFF	DEC	HL	450C	17	03440	DEFB	17H
44BE	C1	02790		POP	BC	450D	81	03450	DEFB	81H
44BF	0B	02800		DEC	BC	450E	32	03460	DEFM	'2'
44C0	1897	02810		JR	AEU	450F	07	03470	DEFB	7H
44C1		02820	AFG	EQU	\$-1	4510	87	03480	DEFB	0B7H
44C2	4F	02830		DEFM	'OR'	4511	4B	03490	DEFM	'K'
44C4	C7	02840		DEFB	0C7H	4512	17	03500	DEFB	17H
44C5	F4	02850		DEFB	0F4H	4513	A7	03510	DEFB	0A7H
44C6	01	02860		DEFB	1H	4514	43	03520	DEFM	'C'
44C7	FF	02870		DEFB	0FFH	4515	03	03530	DEFB	3H
44C8	45	02880		DEFM	'EQ'	4516	93	03540	DEFB	93H
44CA	D5	02890		DEFB	0D5H	4517	06	03550	DEFB	6H
44CB	F4	02900		DEFB	0F4H	4518	05	03560	DEFB	5H
44CC	02	02910		DEFB	2H	4519	94	03570	DEFB	94H
44CD	FF	02920		DEFB	0FFH	451A	01	03580	DEFB	1H
44CE	44	02930		DEFM	'DEF'	451B	FF	03590	DEFB	0FFH
44D1	C2	02940		DEFB	0C2H	451C	41	03600	DEFM	'AD'
44D2	F3	02950		DEFB	0F3H	451E	C4	03610	DEFB	0C4H
44D3	03	02960		DEFB	3H	451F	01	03620	DEFB	1H
44D4	FF	02970		DEFB	0FFH	4520	82	03630	DEFB	82H
44D5	44	02980		DEFM	'DEF'	4521	80	03640	DEFB	80H
44D8	D7	02990		DEFB	0D7H	4522	0B	03650	DEFB	0BH
44D9	F4	03000		DEFB	0F4H	4523	87	03660	DEFB	87H
44DA	04	03010		DEFB	4H	4524	09	03670	DEFB	9H
44DB	FF	03020		DEFB	0FFH	4525	0C	03680	DEFB	0CH
44DC	44	03030		DEFM	'DEF'	4526	88	03690	DEFB	88H
44DF	D3	03040		DEFB	0D3H	4527	09	03700	DEFB	9H
44E0	F4	03050		DEFB	0F4H	4528	01	03710	DEFB	1H
44E1	05	03060		DEFB	5H	4529	93	03720	DEFB	93H
44E2	FF	03070		DEFB	0FFH	452A	C6	03730	DEFB	0C6H
44E3	44	03080		DEFM	'DEF'	452B	FF	03740	DEFB	0FFH
44E6	CD	03090		DEFB	0CDH	452C	41	03750	DEFM	'AD'
44E7	FD	03100		DEFB	0FDH	452E	C3	03760	DEFB	0C3H
44E8	06	03110		DEFB	6H	452F	01	03770	DEFB	1H
44E9	FF	03120		DEFB	0FFH	4530	82	03780	DEFB	82H
44EA	45	03130		DEFM	'EN'	4531	88	03790	DEFB	88H
44EC	C4	03140		DEFB	0C4H	4532	0B	03800	DEFB	0BH
44ED	E0	03150		DEFB	0E0H	4533	A7	03810	DEFB	0A7H
44EE	07	03160		DEFB	7H	4534	4A	03820	DEFM	'J'
44EF	FF	03170		DEFB	0FFH	4535	01	03830	DEFB	1H
44F0	48	03180	AFH	DEFM	'HAL'	4536	93	03840	DEFB	93H
44F3	D4	03190		DEFB	0D4H	4537	CE	03850	DEFB	0CEH
44F4	80	03200		DEFB	80H	4538	FF	03860	DEFB	0FFH
44F5	76	03210		DEFM	'v'	4539	53	03870	DEFM	'SU'
44F6	FF	03220		DEFB	0FFH	453B	C2	03880	DEFB	0C2H
44F7	4C	03230		DEFM	'L'	453C	82	03890	DEFB	82H
44F8	C4	03240		DEFB	0C4H	453D	90	03900	DEFB	90H
44F9	92	03250		DEFB	92H	453E	93	03910	DEFB	93H
44FA	00	03260		DEFB	0H	453F	D6	03920	DEFB	0D6H
44FB	B0	03270		DEFB	0B0H	4540	FF	03930	DEFB	0FFH
44FC	00	03280		DEFB	0H	4541	53	03940	DEFM	'SB'
44FD	03	03290		DEFB	3H	4543	C3	03950	DEFB	0C3H
44FE	82	03300		DEFB	82H	4544	01	03960	DEFB	1H

4545 82	03970	DEFB	82H	458F FF	04630	DEFB	OFFH
4546 98	03980	DEFB	98H	4590 53	04640	DEFM	'SE'
4547 0B	03990	DEFB	0BH	4592 D4	04650	DEFB	0D4H
4548 A7	04000	DEFB	0A7H	4593 18	04660	DEFB	18H
4549 42	04010	DEFM	'B'	4594 C2	04670	DEFB	0C2H
454A 01	04020	DEFB	1H	4595 C0	04680	DEFB	0C0H
454B 93	04030	DEFB	93H	4596 FF	04690	DEFB	OFFH
454C DE	04040	DEFB	0DEH	4597 52	04700	DEFM	'RE'
454D FF	04050	DEFB	OFFH	4599 D3	04710	DEFB	0D3H
454E 41	04060	DEFM	'AN'	459A 18	04720	DEFB	18H
4550 C4	04070	DEFB	0C4H	459B C2	04730	DEFB	0C2H
4551 82	04080	DEFB	82H	459C 80	04740	DEFB	80H
4552 A0	04090	DEFB	0A0H	459D FF	04750	DEFB	OFFH
4553 93	04100	DEFB	93H	459E 4A	04760	DEFM	'J'
4554 E6	04110	DEFB	0E6H	459F D0	04770	DEFB	0D0H
4555 FF	04120	DEFB	OFFH	45A0 1A	04780	DEFB	1AH
4556 58	04130	DEFM	'X0'	45A1 94	04790	DEFB	94H
4558 D2	04140	DEFB	0D2H	45A2 C2	04800	DEFB	0C2H
4559 82	04150	DEFB	82H	45A3 8A	04810	DEFB	8AH
455A A8	04160	DEFB	0A8H	45A4 E9	04820	DEFB	0E9H
455B 93	04170	DEFB	93H	45A5 94	04830	DEFB	94H
455C EE	04180	DEFB	0EEH	45A6 C3	04840	DEFB	0C3H
455D FF	04190	DEFB	OFFH	45A7 FF	04850	DEFB	OFFH
455E 4F	04200	DEFM	'D'	45A8 43	04860	DEFM	'CAL'
455F D2	04210	DEFB	0D2H	45AB CC	04870	DEFB	0CCH
4560 82	04220	DEFB	82H	45AC 1A	04880	DEFB	1AH
4561 B0	04230	DEFB	0B0H	45AD 94	04890	DEFB	94H
4562 93	04240	DEFB	93H	45AE C4	04900	DEFB	0C4H
4563 F6	04250	DEFB	0F6H	45AF 94	04910	DEFB	94H
4564 FF	04260	DEFB	OFFH	45B0 CD	04920	DEFB	0CDH
4565 43	04270	DEFM	'C'	45B1 FF	04930	DEFB	OFFH
4566 D0	04280	DEFB	0D0H	45B2 4A	04940	DEFM	'J'
4567 82	04290	DEFB	82H	45B3 D2	04950	DEFB	0D2H
4568 B8	04300	DEFB	0B8H	45B4 18	04960	DEFB	18H
4569 93	04310	DEFB	93H	45B5 95	04970	DEFB	95H
456A FE	04320	DEFB	0FEH	45B6 20	04980	DEFM	' '
456B FF	04330	DEFB	OFFH	45B7 95	04990	DEFB	95H
456C 49	04340	DEFM	'IN'	45B8 18	05000	DEFB	18H
456E C3	04350	DEFB	0C3H	45B9 FF	05010	DEFB	OFFH
456F 83	04360	DEFB	83H	45BA 44	05020	DEFM	'DJN'
4570 04	04370	DEFB	4H	45BD DA	05030	DEFB	0DAH
4571 85	04380	DEFB	85H	45BE 95	05040	DEFB	95H
4572 03	04390	DEFB	3H	45BF 10	05050	DEFB	10H
4573 FF	04400	DEFB	OFFH	45C0 FF	05060	DEFB	OFFH
4574 44	04410	DEFM	'DE'	45C1 52	05070	DEFM	'RE'
4576 C3	04420	DEFB	0C3H	45C3 D4	05080	DEFB	0D4H
4577 83	04430	DEFB	83H	45C4 80	05090	DEFB	80H
4578 05	04440	DEFB	5H	45C5 C9	05100	DEFB	0C9H
4579 85	04450	DEFB	85H	45C6 9A	05110	DEFB	9AH
457A 0B	04460	DEFB	0BH	45C7 C0	05120	DEFB	0C0H
457B FF	04470	DEFB	OFFH	45C8 FF	05130	DEFB	OFFH
457C 50	04480	DEFM	'PUS'	45C9 45	05140	DEFM	'E'
457F C8	04490	DEFB	0C8H	45CA D8	05150	DEFB	0D8H
4580 86	04500	DEFB	86H	45CB 91	05160	DEFB	91H
4581 C5	04510	DEFB	0C5H	45CC 00	05170	DEFB	0H
4582 FF	04520	DEFB	OFFH	45CD 0E	05180	DEFB	0EH
4583 50	04530	DEFM	'PD'	45CE 89	05190	DEFB	89H
4585 D0	04540	DEFB	0D0H	45CF E3	05200	DEFB	0E3H
4586 86	04550	DEFB	86H	45D0 FF	05210	DEFB	OFFH
4587 C1	04560	DEFB	0C1H	45D1 49	05220	DEFM	'I'
4588 FF	04570	DEFB	OFFH	45D2 CE	05230	DEFB	0CEH
4589 42	04580	DEFM	'BI'	45D3 04	05240	DEFB	4H
458B D4	04590	DEFB	0D4H	45D4 AF	05250	DEFB	0AFH
458C 18	04600	DEFB	18H	45D5 40	05260	DEFM	'S'
458D C2	04610	DEFB	0C2H	45D6 01	05270	DEFB	1H
458E 40	04620	DEFM	'S'	45D7 96	05280	DEFB	96H

45D8 DB	05290	DEFB	0DBH	4623 43	05950	DEFM	'CPD'
45D9 FF	05300	DEFB	OFFH	4626 D2	05960	DEFB	0D2H
45DA 4F	05310	DEFM	'OU'	4627 A0	05970	DEFB	0A0H
45DC D4	05320	DEFB	0D4H	4628 B9	05980	DEFB	0B9H
45DD 0F	05330	DEFB	0FH	4629 FF	05990	DEFB	OFFH
45DE A4	05340	DEFB	0A4H	462A 43	06000	DEFM	'CP'
45DF 41	05350	DEFM	'A'	462C C9	06010	DEFB	0C9H
45E0 16	05360	DEFB	16H	462D A0	06020	DEFB	0A0H
45E1 81	05370	DEFB	81H	462E A1	06030	DEFB	0A1H
45E2 D3	05380	DEFB	0D3H	462F FF	06040	DEFB	OFFH
45E3 FF	05390	DEFB	OFFH	4630 43	06050	DEFM	'CPI'
45E4 49	05400	DEFM	'I'	4633 D2	06060	DEFB	0D2H
45E5 CD	05410	DEFB	0CDH	4634 A0	06070	DEFB	0A0H
45E6 BC	05420	DEFB	0BCH	4635 B1	06080	DEFB	0B1H
45E7 00	05430	DEFB	0H	4636 FF	06090	DEFB	OFFH
45E8 FF	05440	DEFB	OFFH	4637 43	06100	DEFM	'CP'
45E9 52	05450	DEFM	'RL'	4639 CC	06110	DEFB	0CCH
45EB C3	05460	DEFB	0C3H	463A 80	06120	DEFB	80H
45EC C2	05470	DEFB	0C2H	463B 2F	06130	DEFM	'/'
45ED 00	05480	DEFB	0H	463C FF	06140	DEFB	OFFH
45EE FF	05490	DEFB	OFFH	463D 44	06150	DEFM	'DA'
45EF 52	05500	DEFM	'R'	463F C1	06160	DEFB	0C1H
45F0 CC	05510	DEFB	0CCH	4640 80	06170	DEFB	80H
45F1 C2	05520	DEFB	0C2H	4641 27	06180	DEFB	27H
45F2 10	05530	DEFB	10H	4642 FF	06190	DEFB	OFFH
45F3 FF	05540	DEFB	OFFH	4643 44	06200	DEFM	'D'
45F4 52	05550	DEFM	'RR'	4644 C9	06210	DEFB	0C9H
45F6 C3	05560	DEFB	0C3H	4645 80	06220	DEFB	80H
45F7 C2	05570	DEFB	0C2H	4646 F3	06230	DEFB	0F3H
45F8 08	05580	DEFB	8H	4647 FF	06240	DEFB	OFFH
45F9 FF	05590	DEFB	OFFH	4648 45	06250	DEFM	'E'
45FA 52	05600	DEFM	'R'	4649 C9	06260	DEFB	0C9H
45FB D2	05610	DEFB	0D2H	464A 80	06270	DEFB	80H
45FC C2	05620	DEFB	0C2H	464B FB	06280	DEFB	0FBH
45FD 18	05630	DEFB	18H	464C FF	06290	DEFB	OFFH
45FE FF	05640	DEFB	OFFH	464D 45	06300	DEFM	'EX'
45FF 53	05650	DEFM	'SL'	464F D8	06310	DEFB	0D8H
4601 C1	05660	DEFB	0C1H	4650 80	06320	DEFB	80H
4602 C2	05670	DEFB	0C2H	4651 D9	06330	DEFB	0D9H
4603 20	05680	DEFM	' '	4652 FF	06340	DEFB	OFFH
4604 FF	05690	DEFB	OFFH	4653 49	06350	DEFM	'IN'
4605 53	05700	DEFM	'SR'	4655 C4	06360	DEFB	0C4H
4607 C1	05710	DEFB	0C1H	4656 A0	06370	DEFB	0A0H
4608 C2	05720	DEFB	0C2H	4657 AA	06380	DEFB	0AAH
4609 28	05730	DEFM	' ('	4658 FF	06390	DEFB	OFFH
460A FF	05740	DEFB	OFFH	4659 49	06400	DEFM	'IND'
460B 53	05750	DEFM	'SR'	465C D2	06410	DEFB	0D2H
460D CC	05760	DEFB	0CCH	465D A0	06420	DEFB	0A0H
460E C2	05770	DEFB	0C2H	465E BA	06430	DEFB	0BAH
460F 38	05780	DEFM	'8'	465F FF	06440	DEFB	OFFH
4610 FF	05790	DEFB	OFFH	4660 49	06450	DEFM	'IN'
4611 52	05800	DEFM	'RS'	4662 C9	06460	DEFB	0C9H
4613 D4	05810	DEFB	0D4H	4663 A0	06470	DEFB	0A0H
4614 99	05820	DEFB	99H	4664 A2	06480	DEFB	0A2H
4615 C7	05830	DEFB	0C7H	4665 FF	06490	DEFB	OFFH
4616 FF	05840	DEFB	OFFH	4666 49	06500	DEFM	'INI'
4617 43	05850	DEFM	'CC'	4669 D2	06510	DEFB	0D2H
4619 C6	05860	DEFB	0C6H	466A A0	06520	DEFB	0A0H
461A 80	05870	DEFB	80H	466B B2	06530	DEFB	0B2H
461B 3F	05880	DEFM	'?'	466C FF	06540	DEFB	OFFH
461C FF	05890	DEFB	OFFH	466D 4C	06550	DEFM	'LD'
461D 43	05900	DEFM	'CP'	466F C4	06560	DEFB	0C4H
461F C4	05910	DEFB	0C4H	4670 A0	06570	DEFB	0A0H
4620 A0	05920	DEFB	0A0H	4671 A8	06580	DEFB	0A8H
4621 A9	05930	DEFB	0A9H	4672 FF	06590	DEFB	OFFH
4622 FF	05940	DEFB	OFFH	4673 4C	06600	DEFM	'LDD'

4676 D2	06610	DEFB	0D2H	46CD A0	07270	DEFB	0A0H
4677 A0	06620	DEFB	0A0H	46CE 6F	07280	DEFM	'a'
4678 B8	06630	DEFB	0BBH	46CF FF	07290	DEFB	OFFH
4679 FF	06640	DEFB	OFFH	46D0 52	07300	DEFM	'RR'
467A 4C	06650	DEFM	'LD'	46D2 C1	07310	DEFB	0C1H
467C C9	06660	DEFB	0C9H	46D3 80	07320	DEFB	80H
467D A0	06670	DEFB	0A0H	46D4 1F	07330	DEFB	1FH
467E A0	06680	DEFB	0A0H	46D5 FF	07340	DEFB	OFFH
467F FF	06690	DEFB	OFFH	46D6 52	07350	DEFM	'RRC'
4680 4C	06700	DEFM	'LDI'	46D9 C1	07360	DEFB	0C1H
4683 D2	06710	DEFB	0D2H	46DA 80	07370	DEFB	80H
4684 A0	06720	DEFB	0A0H	46DB 0F	07380	DEFB	0FH
4685 B0	06730	DEFB	0B0H	46DC FF	07390	DEFB	OFFH
4686 FF	06740	DEFB	OFFH	46DD 52	07400	DEFM	'RR'
4687 4E	06750	DEFM	'NE'	46DF C4	07410	DEFB	0C4H
4689 C7	06760	DEFB	0C7H	46E0 A0	07420	DEFB	0A0H
468A A0	06770	DEFB	0A0H	46E1 67	07430	DEFM	'g'
468B 44	06780	DEFM	'D'	46E2 FF	07440	DEFB	OFFH
468C FF	06790	DEFB	OFFH	46E3 53	07450	DEFM	'SC'
468D 4E	06800	DEFM	'NO'	46E5 C6	07460	DEFB	0C6H
468F D0	06810	DEFB	0D0H	46E6 80	07470	DEFB	80H
4690 80	06820	DEFB	80H	46E7 37	07480	DEFM	'7'
4691 00	06830	DEFB	0H	46E8 FF	07490 AFN	DEFB	OFFH
4692 FF	06840	DEFB	OFFH	46E9 FF	07500 AFO	DEFB	OFFH
4693 4F	06850	DEFM	'OTD'	46EA C1	07510	DEFB	0C1H
4696 D2	06860	DEFB	0D2H	46EB 00	07520	DEFB	0H
4697 A0	06870	DEFB	0A0H	46EC FF	07530	DEFB	OFFH
4698 BB	06880	DEFB	0BBH	46ED C2	07540	DEFB	0C2H
4699 FF	06890	DEFB	OFFH	46EE 00	07550	DEFB	0H
469A 4F	06900	DEFM	'OTI'	46EF C3	07560	DEFB	0C3H
469D D2	06910	DEFB	0D2H	46F0 01	07570	DEFB	1H
469E A0	06920	DEFB	0A0H	46F1 C4	07580	DEFB	0C4H
469F B3	06930	DEFB	0B3H	46F2 02	07590	DEFB	2H
46A0 FF	06940	DEFB	OFFH	46F3 C5	07600	DEFB	0C5H
46A1 4F	06950	DEFM	'OUT'	46F4 03	07610	DEFB	3H
46A4 C4	06960	DEFB	0C4H	46F5 C8	07620	DEFB	0C8H
46A5 A0	06970	DEFB	0A0H	46F6 04	07630	DEFB	4H
46A6 AB	06980	DEFB	0ABH	46F7 CC	07640	DEFB	0CCH
46A7 FF	06990	DEFB	OFFH	46F8 05	07650	DEFB	5H
46A8 4F	07000	DEFM	'OUT'	46F9 28	07660	DEFM	'(HL'
46AB C9	07010	DEFB	0C9H	46FC A9	07670	DEFB	0A9H
46AC A0	07020	DEFB	0A0H	46FD 06	07680	DEFB	6H
46AD A3	07030	DEFB	0A3H	46FE 28	07690	DEFM	'(IX+#'
46AE FF	07040	DEFB	OFFH	4703 A9	07700	DEFB	0A9H
46AF 52	07050	DEFM	'RET'	4704 06	07710	DEFB	6H
46B2 C9	07060	DEFB	0C9H	4705 C1	07720	DEFB	0C1H
46B3 A0	07070	DEFB	0A0H	4706 07	07730	DEFB	7H
46B4 4D	07080	DEFM	'M'	4707 FF	07740	DEFB	OFFH
46B5 FF	07090	DEFB	OFFH	4708 C2	07750	DEFB	0C2H
46B6 52	07100	DEFM	'RET'	4709 00	07760	DEFB	0H
46B9 CE	07110	DEFB	0CEH	470A C3	07770	DEFB	0C3H
46BA A0	07120	DEFB	0A0H	470B 08	07780	DEFB	8H
46BB 45	07130	DEFM	'E'	470C C4	07790	DEFB	0C4H
46BC FF	07140	DEFB	OFFH	470D 10	07800	DEFB	10H
46BD 52	07150	DEFM	'RL'	470E C5	07810	DEFB	0C5H
46BF C1	07160	DEFB	0C1H	470F 18	07820	DEFB	18H
46C0 80	07170	DEFB	80H	4710 C8	07830	DEFB	0C8H
46C1 17	07180	DEFB	17H	4711 20	07840	DEFM	' '
46C2 FF	07190	DEFB	OFFH	4712 CC	07850	DEFB	0CCH
46C3 52	07200	DEFM	'RLC'	4713 28	07860	DEFM	'(HL'
46C6 C1	07210	DEFB	0C1H	4717 A9	07870	DEFB	0A9H
46C7 80	07220	DEFB	80H	4718 30	07880	DEFM	'O(IX+#'
46C8 07	07230	DEFB	7H	471E A9	07890	DEFB	0A9H
46C9 FF	07240	DEFB	OFFH	471F 30	07900	DEFM	'0'
46CA 52	07250	DEFM	'RL'	4720 C1	07910	DEFB	0C1H
46CC C4	07260	DEFB	0C4H	4721 38	07920	DEFM	'8'

4722 FF	07930	DEFB	OFFH	476A 30	08590	DEFM	'0'
4723 C2	07940	DEFB	0C2H	476B FF	08600	DEFB	OFFH
4724 00	07950	DEFB	0H	476C 48	08610	DEFM	'H'
4725 C3	07960	DEFB	0C3H	476D CC	08620	DEFB	OCCH
4726 08	07970	DEFB	8H	476E 00	08630	DEFB	0H
4727 C4	07980	DEFB	0C4H	476F 49	08640	DEFM	'I'
4728 10	07990	DEFB	10H	4770 D8	08650	DEFB	0DBH
4729 C5	08000	DEFB	0C5H	4771 00	08660	DEFB	0H
472A 18	08010	DEFB	18H	4772 FF	08670	DEFB	OFFH
472B C8	08020	DEFB	0C8H	4773 28	08680	DEFM	'(HL'
472C 20	08030	DEFM	' '	4776 A9	08690	DEFB	0A9H
472D CC	08040	DEFB	OCCH	4777 00	08700	DEFB	0H
472E 28	08050	DEFM	' ('	4778 28	08710	DEFM	'(IX'
472F C1	08060	DEFB	0C1H	477B A9	08720	DEFB	0A9H
4730 38	08070	DEFM	'B'	477C 00	08730	DEFB	0H
4731 FF	08080	DEFB	OFFH	477D FF	08740	DEFB	OFFH
4732 42	08090	DEFM	'B'	477E 48	08750	DEFM	'H'
4733 C3	08100	DEFB	0C3H	477F CC	08760	DEFB	OCCH
4734 00	08110	DEFB	0H	4780 00	08770	DEFB	0H
4735 44	08120	DEFM	'D'	4781 FF	08780	DEFB	OFFH
4736 C5	08130	DEFB	0C5H	4782 49	08790	DEFM	'I'
4737 10	08140	DEFB	10H	4783 D8	08800	DEFB	0DBH
4738 48	08150	DEFM	'H'	4784 00	08810	DEFB	0H
4739 CC	08160	DEFB	OCCH	4785 FF	08820	DEFB	OFFH
473A 20	08170	DEFM	'I'	4786 53	08830	DEFM	'S'
473C D8	08180	DEFB	0DBH	4787 D0	08840	DEFB	0D0H
473D 20	08190	DEFM	'S'	4788 00	08850	DEFB	0H
473F D0	08200	DEFB	0D0H	4789 FF	08860	DEFB	OFFH
4740 30	08210	DEFM	'0'	478A 28	08870	DEFM	'(SP'
4741 FF	08220	DEFB	OFFH	478D A9	08880	DEFB	0A9H
4742 42	08230	DEFM	'B'	478E 00	08890	DEFB	0H
4743 C3	08240	DEFB	0C3H	478F FF	08900	DEFB	OFFH
4744 00	08250	DEFB	0H	4790 28	08910	DEFM	'(C'
4745 44	08260	DEFM	'D'	4792 A9	08920	DEFB	0A9H
4746 C5	08270	DEFB	0C5H	4793 00	08930	DEFB	0H
4747 10	08280	DEFB	10H	4794 FF	08940	DEFB	OFFH
4748 48	08290	DEFM	'H'	4795 41	08950	DEFM	'A,'
4749 CC	08300	DEFB	OCCH	4797 D2	08960	DEFB	0D2H
474A 20	08310	DEFM	'I'	4798 5F	08970	DEFM	'_R,'
474C D8	08320	DEFB	0DBH	479B C1	08980	DEFB	0C1H
474D 20	08330	DEFM	'A'	479C 4F	08990	DEFM	'0A,'
474F C6	08340	DEFB	0C6H	479F C9	09000	DEFB	0C9H
4750 30	08350	DEFM	'0'	47A0 57	09010	DEFM	'WI,'
4751 FF	08360	DEFB	OFFH	47A3 C1	09020	DEFB	0C1H
4752 42	08370	DEFM	'B'	47A4 47	09030	DEFM	'G'
4753 C3	08380	DEFB	0C3H	47A5 FF	09040	DEFB	OFFH
4754 00	08390	DEFB	0H	47A6 44	09050	DEFM	'DE,H'
4755 44	08400	DEFM	'D'	47AA CC	09060	DEFB	OCCH
4756 C5	08410	DEFB	0C5H	47AB EB	09070	DEFB	0EBH
4757 10	08420	DEFB	10H	47AC 41	09080	DEFM	'AF,AF'
4758 48	08430	DEFM	'H'	47B1 A7	09090	DEFB	0A7H
4759 CC	08440	DEFB	OCCH	47B2 08	09100	DEFB	8H
475A 20	08450	DEFM	'S'	47B3 FF	09110	DEFB	OFFH
475C D0	08460	DEFB	0D0H	47B4 41	09120	DEFM	'A,(BC'
475D 30	08470	DEFM	'0'	47B9 A9	09130	DEFB	0A9H
475E FF	08480	DEFB	OFFH	47BA 0A	09140	DEFB	0AH
475F 42	08490	DEFM	'B'	47BB 41	09150	DEFM	'A,(DE'
4760 C3	08500	DEFB	0C3H	47C0 A9	09160	DEFB	0A9H
4761 00	08510	DEFB	0H	47C1 1A	09170	DEFB	1AH
4762 44	08520	DEFM	'D'	47C2 28	09180	DEFM	'(BC),'
4763 C5	08530	DEFB	0C5H	47C7 C1	09190	DEFB	0C1H
4764 10	08540	DEFB	10H	47C8 02	09200	DEFB	2H
4765 49	08550	DEFM	'I'	47C9 28	09210	DEFM	'(DE),'
4766 D8	08560	DEFB	0DBH	47CE C1	09220	DEFB	0C1H
4767 20	08570	DEFM	'S'	47CF 12	09230	DEFB	12H
4769 D0	08580	DEFB	0D0H	47D0 FF	09240	DEFB	OFFH

47D1 A3	09250	DEFB	0A3H	481E 50	09910	DEFM	'P'
47D2 00	09260	DEFB	0H	481F CF	09920	DEFB	OCFH
47D3 FF	09270	DEFB	OFFH	4820 20	09930	DEFM	'P'
47D4 A4	09280	DEFB	0A4H	4822 C5	09940	DEFB	OC5H
47D5 00	09290	DEFB	0H	4823 28	09950	DEFM	'C'
47D6 FF	09300	DEFB	OFFH	4824 D0	09960	DEFB	OD0H
47D7 A5	09310	DEFB	0A5H	4825 30	09970	DEFM	'0'
47D8 00	09320	DEFB	0H	4826 CD	09980	DEFB	OCDH
47D9 FF	09330	DEFB	OFFH	4827 38	09990	DEFM	'B'
47DA 28	09340	DEFM	'(#'	4828 FF	10000	DEFB	OFFH
47DC A9	09350	DEFB	0A9H	4829 4E	10010	DEFM	'N'
47DD 00	09360	DEFB	0H	482A DA	10020	DEFB	ODAH
47DE FF	09370	DEFB	OFFH	482B 00	10030	DEFB	0H
47DF 28	09380	DEFM	'(\$'	482C DA	10040	DEFB	ODAH
47E1 A9	09390	DEFB	0A9H	482D 08	10050	DEFB	8H
47E2 00	09400	DEFB	0H	482E 4E	10060	DEFM	'N'
47E3 FF	09410	DEFB	OFFH	482F C3	10070	DEFB	OC3H
47E4 B0	09420	DEFB	0B0H	4830 10	10080	DEFB	10H
47E5 00	09430	DEFB	0H	4831 C3	10090	DEFB	OC3H
47E6 B1	09440	DEFB	0B1H	4832 18	10100	DEFB	18H
47E7 08	09450	DEFB	8H	4833 FF	10110	DEFB	OFFH
47E8 B2	09460	DEFB	0B2H	4834 B0	10120	DEFB	0B0H
47E9 10	09470	DEFB	10H	4835 46	10130	DEFM	'F'
47EA B3	09480	DEFB	0B3H	4836 B1	10140	DEFB	0B1H
47EB 18	09490	DEFB	18H	4837 56	10150	DEFM	'V'
47EC B4	09500	DEFB	0B4H	4838 B2	10160	DEFB	0B2H
47ED 20	09510	DEFM	' '	4839 5E	10170	DEFM	'^'
47EE B5	09520	DEFB	0B5H	483A FF	10180	DEFB	OFFH
47EF 28	09530	DEFM	' ('	483B 27	10190	DEFB	27H
47F0 B6	09540	DEFB	0B6H	483C 24	10200	DEFM	'\$'
47F1 30	09550	DEFM	'0'	483D A7	10210	DEFB	0A7H
47F2 B7	09560	DEFB	0B7H	483E 00	10220	DEFB	0H
47F3 38	09570	DEFM	'8'	483F FF	10230	DEFB	OFFH
47F4 FF	09580	DEFB	OFFH	4840 7C	10240	AHK	LD
47F5 30	09590	DEFM	'0'	4841 CD4548	10250		CALL
47F6 C8	09600	DEFB	0C8H	4844 7D	10260		LD
47F7 00	09610	DEFB	0H	4845 F5	10270	AHM	PUSH
47F8 38	09620	DEFM	'8'	4846 07	10280		RLCA
47F9 C8	09630	DEFB	0C8H	4847 07	10290		RLCA
47FA 08	09640	DEFB	8H	4848 07	10300		RLCA
47FB 31	09650	DEFM	'10'	4849 07	10310		RLCA
47FD C8	09660	DEFB	0C8H	484A CD4E48	10320		CALL
47FE 10	09670	DEFB	10H	484D F1	10330		POP
47FF 31	09680	DEFM	'18'	484E E60F	10340	AHN	AND
4801 C8	09690	DEFB	0C8H	4850 FE0A	10350		CP
4802 18	09700	DEFB	18H	4852 3802	10360		JR
4803 32	09710	DEFM	'20'	4854 C607	10370		ADD
4805 C8	09720	DEFB	0C8H	4856 C630	10380	AHO	ADD
4806 20	09730	DEFM	'28'	4858 E5	10390	AHP	PUSH
4809 C8	09740	DEFB	0C8H	4859 D5	10400		PUSH
480A 28	09750	DEFM	'(30'	485A CD3300	10410		CALL
480D C8	09760	DEFB	0C8H	485D D1	10420		POP
480E 30	09770	DEFM	'03B'	485E E1	10430		FOP
4811 C8	09780	DEFB	0C8H	485F C9	10440		RET
4812 38	09790	DEFM	'8'	4860 3E20	10450	AHQ	LD
4813 FF	09800	DEFB	OFFH	4862 CD5848	10460		CALL
4814 4E	09810	DEFM	'N'	4865 10F9	10470		DJNZ
4815 DA	09820	DEFB	ODAH	4867 C9	10480		RET
4816 00	09830	DEFB	0H	4868 1A	10490	AHR	LD
4817 DA	09840	DEFB	ODAH	4869 E61F	10500		AND
4818 08	09850	DEFB	8H	486B E5	10510		PUSH
4819 4E	09860	DEFM	'N'	486C 4F	10520		LD
481A C3	09870	DEFB	OC3H	486D 0600	10530		LD
481B 10	09880	DEFB	10H	486F 21CE48	10540		LD
481C C3	09890	DEFB	OC3H	4872 09	10550		ADD
481D 18	09900	DEFB	18H	4873 46	10560		LD

4874 E1	10570	POP	HL	48D5 CF	11230	DEFB	OCFH
4875 78	10580	LD	A,B	48D6 CF	11240	DEFB	OCFH
4876 C9	10590	RET		48D7 FF	11250	DEFB	OFFH
4877 1A	10600 AHU	LD	A, (DE)	48D8 FF	11260	DEFB	OFFH
4878 E61F	10610	AND	AAN	48D9 FF	11270	DEFB	OFFH
487A 0601	10620	LD	B,AAB	48DA FF	11280	DEFB	OFFH
487C CB	10630	RET	Z	48DB FF	11290	DEFB	OFFH
487D D5	10640	PUSH	DE	48DC FF	11300	DEFB	OFFH
487E CD6848	10650	CALL	AHR	48DD FF	11310	DEFB	OFFH
4881 2F	10660	CPL		48DE 00	11320	DEFB	OH
4882 A6	10670	AND	(HL)	48DF 00	11330	DEFB	OH
4883 DD7703	10680	LD	(IX+03H),A	48E0 00	11340	DEFB	OH
4886 1A	10690	LD	A, (DE)	48E1 FF	11350	DEFB	OFFH
4887 E61F	10700	AND	AAN	48E2 FF	11360	DEFB	OFFH
4889 47	10710	LD	B,A	48E3 FF	11370	DEFB	OFFH
488A 11E946	10720	LD	DE,AFD	48E4 FF	11380	DEFB	OFFH
488D 1A	10730 AHV	LD	A, (DE)	48E5 FF	11390	DEFB	OFFH
488E 3C	10740	INC	A	48E6 C7	11400	DEFB	OC7H
488F 13	10750	INC	DE	48E7 C7	11410	DEFB	OC7H
4890 20FB	10760	JR	NZ,AHV	48E8 C7	11420	DEFB	OC7H
4892 10F9	10770	DJNZ	AHV	48E9 E7	11430	DEFB	OE7H
4894 D5	10780 AHW	PUSH	DE	48EA 00	11440	DEFB	OH
4895 1A	10790 AHX	LD	A, (DE)	48EB 00	11450	NOP	
4896 07	10800	RLCA		48EC 00	11460	NOP	
4897 13	10810	INC	DE	48ED 00	11470	NOP	
4898 30FB	10820	JR	NC,AHX	48EE E5	11480	PUSH	HL
489A 1A	10830	LD	A, (DE)	48EF 3E0D	11490	LD	A,AAK
489B DDBE03	10840	CP	(IX+03H)	48F1 CD5848	11500	CALL	AHP
489E C1	10850	POP	BC	48F4 CD4048	11510	CALL	AHK
489F 2808	10860	JR	Z,AHZ	48F7 0604	11520	LD	B,AAE
48A1 13	10870 AHY	INC	DE	48F9 CD6048	11530	CALL	AHQ
48A2 1A	10880	LD	A, (DE)	48FC DD21E942	11540	LD	IX,ADH
48A3 3C	10890	INC	A	4900 0E00	11550	LD	C,AAA
48A4 20EE	10900	JR	NZ,AHW	4902 7E	11560	LD	A, (HL)
48A6 47	10910	LD	B,A	4903 FEDD	11570	CP	ABT
48A7 D1	10920	POP	DE	4905 200B	11580	JR	NZ,AIE
48A8 C9	10930	RET		4907 CBC1	11590	SET	00H,C
48A9 0A	10940 AHZ	LD	A, (BC)	4909 23	11600 AID	INC	HL
48AA FE49	10950	CP	ABD	490A 7E	11610	LD	A, (HL)
48AC 2807	10960	JR	Z,AIA	490B FECE	11620	CP	ABS
48AE 03	10970	INC	BC	490D 2003	11630	JR	NZ,AIE
48AF 0A	10980	LD	A, (BC)	490F 23	11640	INC	HL
48B0 0B	10990	DEC	BC	4910 1813	11650	JR	AIH
48B1 FE49	11000	CP	ABD	4912 FEFD	11660 AIE	CP	ABY
48B3 2004	11010	JR	NZ,AIB	4914 2004	11670	JR	NZ,AIF
48B5 DDCB00D6	11020 AIA	SET	02H, (IX+00H)	4916 CBC9	11680	SET	01H,C
48B9 DD7E00	11030 AIB	LD	A, (IX+00H)	4918 18EF	11690	JR	AID
48BC E603	11040	AND	AAD	491A FEED	11700 AIF	CP	ABW
48BE 280C	11050	JR	Z,AIC	491C 2003	11710	JR	NZ,AIG
48C0 0A	11060	LD	A, (BC)	491E CBE9	11720	SET	05H,C
48C1 FE48	11070	CP	ABC	4920 23	11730	INC	HL
48C3 28DC	11080	JR	Z,AHY	4921 FECE	11740 AIG	CP	ABS
48C5 03	11090	INC	BC	4923 2003	11750	JR	NZ,AII
48C6 0A	11100	LD	A, (BC)	4925 CBF1	11760 AIH	SET	06H,C
48C7 FE48	11110	CP	ABC	4927 23	11770	INC	HL
48C9 0B	11120	DEC	BC	4928 DD7100	11780 AII	LD	(IX+00H),C
48CA 28D5	11130	JR	Z,AHY	492B 11F044	11790	LD	DE,AFH
48CC D1	11140 AIC	POP	DE	492E ED53EA42	11800 AIJ	LD	(ADI),DE
48CD C9	11150	RET		4932 1A	11810 AIK	LD	A, (DE)
48CE FF	11160 MASKE	DEFB	OFFH	4933 07	11820	RLCA	
48CF FF	11170	DEFB	OFFH	4934 13	11830	INC	DE
48D0 F8	11180	DEFB	OF8H	4935 30FB	11840	JR	NC,AIK
48D1 C7	11190	DEFB	OC7H	4937 DDCB00FE	11850 AIL	SET	07H, (IX+00H)
48D2 C7	11200	DEFB	OC7H	493B DDCB0096	11860	RES	02H, (IX+00H)
48D3 CF	11210	DEFB	OCFH	493F CD6848	11870	CALL	AHR
48D4 CF	11220	DEFB	OCFH	4942 A6	11880	AND	(HL)

4943	F5	11890	PUSH	AF	49B9	13	12550	AIT	INC	DE
4944	1A	11900	LD	A, (DE)	49BA	1A	12560		LD	A, (DE)
4945	07	11910	RLCA		49BB	3C	12570		INC	A
4946	06FF	11920	LD	B, ABZ	49BC	C23749	12580		JP	NZ, AIL
4948	380B	11930	JR	C, AIM	49BF	13	12590		INC	DE
494A	DDCB00BE	11940	RES	07H, (IX+00H)	49C0	1A	12600		LD	A, (DE)
494E	13	11950	INC	DE	49C1	3C	12610		INC	A
494F	CD684B	11960	CALL	AHR	49C2	C22E49	12620		JP	NZ, AIJ
4952	F1	11970	POP	AF	49C5	E1	12630	AIU	POP	HL
4953	A0	11980	AND	B	49C6	00	12640	AIV	NOP	
4954	47	11990	LD	B, A	49C7	00	12650		NOP	
4955	13	12000	INC	DE	49C8	C9	12660		RET	
4956	1A	12010	LD	A, (DE)	49C9	7A	12670	AIW	LD	A, D
4957	B8	12020	CP	B	49CA	3D	12680		DEC	A
4958	205F	12030	JR	NZ, AIT	49CB	C8	12690		RET	Z
495A	1B	12040	DEC	DE	49CC	1A	12700	AIX	LD	A, (DE)
495B	1A	12050	LD	A, (DE)	49CD	E67F	12710		AND	ABJ
495C	DDAE00	12060	XOR	(IX+00H)	49CF	FE58	12720		CP	ABF
495F	E660	12070	AND	ABG	49D1	2007	12730		JR	NZ, AIY
4961	2055	12080	JR	NZ, AIS	49D3	DDCB004E	12740		BIT	01H, (IX+00H)
4963	CD774B	12090	CALL	AHU	49D7	2801	12750		JR	Z, AIY
4966	78	12100	LD	A, B	49D9	3C	12760		INC	A
4967	A7	12110	AND	A	49DA	FE23	12770	AIY	CP	AAP
4968	284E	12120	JR	Z, AIS	49DC	2017	12780		JR	NZ, AJA
496A	C5	12130	PUSH	BC	49DE	DDCB0076	12790		BIT	06H, (IX+00H)
496B	1B	12140	DEC	DE	49E2	280A	12800		JR	Z, AIZ
496C	DDCB007E	12150	BIT	07H, (IX+00H)	49E4	2B	12810		DEC	HL
4970	0601	12160	LD	B, AAB	49E5	2B	12820		DEC	HL
4972	2006	12170	JR	NZ, AID	49E6	7E	12830		LD	A, (HL)
4974	CD774B	12180	CALL	AHU	49E7	CD454B	12840		CALL	AHM
4977	13	12190	INC	DE	49EA	23	12850		INC	HL
4978	78	12200	LD	A, B	49EB	23	12860		INC	HL
4979	A7	12210	AND	A	49EC	1833	12870		JR	AJE
497A	FDE1	12220	POP	IY	49EE	7E	12880	AIZ	LD	A, (HL)
497C	283A	12230	JR	Z, AIS	49EF	CD454B	12890		CALL	AHM
497E	DD7E00	12240	LD	A, (IX+00H)	49F2	23	12900		INC	HL
4981	E603	12250	AND	AAD	49F3	182C	12910		JR	AJE
4983	2806	12260	JR	Z, AIP	49F5	FE24	12920	AJA	CP	AAQ
4985	DDCB0056	12270	BIT	02H, (IX+00H)	49F7	200D	12930		JR	NZ, AJB
4989	283A	12280	JR	Z, AIU	49F9	D5	12940		PUSH	DE
498B	FDE5	12290	PUSH	IY	49FA	5E	12950		LD	E, (HL)
498D	C5	12300	PUSH	BC	49FB	23	12960		INC	HL
498E	23	12310	INC	HL	49FC	56	12970		LD	D, (HL)
498F	ED5BEA42	12320	LD	DE, (ADI)	49FD	EB	12980		EX	DE, HL
4993	0607	12330	LD	B, AAH	49FE	CD404B	12990		CALL	AHK
4995	1A	12340	LD	A, (DE)	4A01	EB	13000		EX	DE, HL
4996	E67F	12350	AND	ABJ	4A02	D1	13010		POP	DE
4998	CD584B	12360	CALL	AHP	4A03	23	13020		INC	HL
499B	1A	12370	LD	A, (DE)	4A04	181B	13030		JR	AJE
499C	07	12380	RLCA		4A06	FE25	13040	AJB	CP	AAR
499D	05	12390	DEC	B	4A08	2014	13050		JR	NZ, AJD
499E	13	12400	INC	DE	4A0A	7E	13060		LD	A, (HL)
499F	30F4	12410	JR	NC, AIQ	4A0B	E5	13070		PUSH	HL
49A1	CD604B	12420	CALL	AHQ	4A0C	FEB0	13080		CP	ABK
49A4	D1	12430	POP	DE	4A0E	0600	13090		LD	B, AAA
49A5	CDC949	12440	CALL	AIW	4A10	3802	13100		JR	C, AJC
49A8	7A	12450	LD	A, D	4A12	06FF	13110		LD	B, ABZ
49A9	3D	12460	DEC	A	4A14	4F	13120	AJC	LD	C, A
49AA	2805	12470	JR	Z, AIR	4A15	09	13130		ADD	HL, BC
49AC	3E2C	12480	LD	A, AAT	4A16	23	13140		INC	HL
49AE	CD584B	12490	CALL	AHP	4A17	CD404B	13150		CALL	AHK
49B1	D1	12500	POP	DE	4A1A	E1	13160		POP	HL
49B2	CDC949	12510	CALL	AIW	4A1B	23	13170		INC	HL
49B5	C1	12520	POP	BC	4A1C	1803	13180		JR	AJE
49B6	180E	12530	JR	AIV	4A1E	CD584B	13190	AJD	CALL	AHP
49B8	13	12540	INC	DE	4A21	1A	13200	AJE	LD	A, (DE)

4A22 07	13210	RLCA		4AAC 3E55	13870	LD	A, ABE
4A23 13	13220	INC	DE	4AAE CD6402	13890	CALL	ACE
4A24 30A6	13230	JR	NC, AIX	4AB1 0606	13890	LD	B, AAG
4A26 C9	13240	RET		4AB3 1A	13900	LD	A, (DE)
4A27 CD7F0A	13250	CALL	ACD	4AB4 CD6402	13910	CALL	ACE
4A2A ED73ED42	13260	LD	(ADK), SF	4AB7 13	13920	INC	DE
4A2E 31684B	13270	LD	SP, AJU	4AB8 10F9	13930	DJNZ	AJN
4A31 013B4A	13280	LD	BC, AJF	4ABA ED5BE942	13940	LD	DE, (ADH)
4A34 C5	13290	PUSH	BC	4ABE 2AEB42	13950	LD	HL, (ADJ)
4A35 DD2AEF42	13300	LD	IX, (ADL)	4AC1 97	13960	SUB	A
4A39 DDE9	13310	JP	(IX)	4AC2 ED52	13970	SBC	HL, DE
4A3B ED7BED42	13320	LD	SP, (ADK)	4AC4 282F	13980	JR	Z, AJR
4A3F C39A0A	13330	JP	ACP	4AC6 010001	13990	LD	BC, ACA
4A42 EB	13340	EX	DE, HL	4AC9 E5	14000	PUSH	HL
4A43 1A	13350	LD	A, (DE)	4ACA ED42	14010	SBC	HL, BC
4A44 ED5BE942	13360	LD	DE, (ADH)	4ACC E1	14020	POP	HL
4A48 FE2D	13370	CP	AAU	4ACD 3802	14030	JR	C, AJP
4A4A 200A	13380	JR	NZ, AJG	4ACF 2E00	14040	LD	L, AAA
4A4C 210000	13390	LD	HL, AAA	4AD1 45	14050	LD	B, L
4A4F ED52	13400	SBC	HL, DE	4AD2 3E3C	14060	LD	A, AAZ
4A51 EB	13410	EX	DE, HL	4AD4 CD6402	14070	CALL	ACE
4A52 ED53E942	13420	LD	(ADH), DE	4AD7 7B	14080	LD	A, B
4A56 AF	13430	XOR	A	4AD8 CD6402	14090	CALL	ACE
4A57 CD1202	13440	CALL	ACC	4ADB 7B	14100	LD	A, E
4A5A CD9602	13450	CALL	AC6	4ADC CD6402	14110	CALL	ACE
4A5D 0607	13460	LD	B, AAH	4ADF 7A	14120	LD	A, D
4A5F CD3502	13470	CALL	ACD	4AE0 CD6402	14130	CALL	ACE
4A62 10FB	13480	DJNZ	AJH	4AE3 83	14140	ADD	A, E
4A64 CD3502	13490	CALL	ACD	4AE4 4F	14150	LD	C, A
4A67 FE3C	13500	CP	AAZ	4AE5 1A	14160	LD	A, (DE)
4A69 202C	13510	JR	NZ, AJL	4AE6 81	14170	ADD	A, C
4A6B CD3502	13520	CALL	ACD	4AE7 4F	14180	LD	C, A
4A6E 47	13530	LD	B, A	4AEB 1A	14190	LD	A, (DE)
4A6F CD3502	13540	CALL	ACD	4AE9 CD6402	14200	CALL	ACE
4A72 6F	13550	LD	L, A	4AEC 13	14210	INC	DE
4A73 CD3502	13560	CALL	ACD	4AED 10F6	14220	DJNZ	AJQ
4A76 67	13570	LD	H, A	4AEF 79	14230	LD	A, C
4A77 85	13580	ADD	A, L	4AF0 CD6402	14240	CALL	ACE
4A78 4F	13590	LD	C, A	4AF3 18C9	14250	JR	AJO
4A79 ED5BE942	13600	LD	DE, (ADH)	4AF5 3E78	14260	LD	A, ABI
4A7D 19	13610	ADD	HL, DE	4AF7 CD6402	14270	CALL	ACE
4A7E CD3502	13620	CALL	ACD	4AFA 3A094B	14280	LD	A, (AJS)
4A81 77	13630	LD	(HL), A	4AFD CD6402	14290	CALL	ACE
4A82 81	13640	ADD	A, C	4B00 3A0A4B	14300	LD	A, (AJT)
4A83 4F	13650	LD	C, A	4B03 CD6402	14310	CALL	ACE
4A84 23	13660	INC	HL	4B06 C3FB01	14320	JP	ACB
4A85 10F7	13670	DJNZ	AJJ	4B09 00	14330	AJS	DEFB
4A87 CD3502	13680	CALL	ACD	4B0A 00	14340	AJT	DEFB
4A8A B9	13690	CP	C	4B0B CF	14350	LINE	RST
4A8B 00	13700	NOP		4B0C 89	14360	DEFB	B9H
4A8C 00	13710	NOP		4B0D CD282B	14370	CALL	2828H
4A8D CD3502	13720	CALL	ACD	4B10 7E	14380	LD	A, (HL)
4A90 FE78	13730	CP	ABI	4B11 FE23	14390	CP	2#
4A92 20D3	13740	JR	NZ, AJI	4B13 CA2D01	14400	JP	Z, 12DH
4A94 C3FB01	13750	JP	ACB	4B16 CDCD21	14410	CALL	21CDH
4A97 3E3F	13760	LD	A, ABA	4B19 CD0D26	14420	CALL	260DH
4A99 323F3C	13770	LD	(ACW), A	4B1C CDF40A	14430	CALL	0AF4H
4A9C 18F6	13780	JR	AJK	4B1F E5	14440	PUSH	HL
4A9E EB	13790	EX	DE, HL	4B20 D5	14450	PUSH	DE
4A9F 13	13800	INC	DE	4B21 CD6103	14460	CALL	0361H
4AA0 1A	13810	LD	A, (DE)	4B24 DA314B	14470	JP	C, LINE1
4AA1 FE20	13820	CP	AAO	4B27 0600	14480	LD	B, 0
4AA3 28FA	13830	JR	Z, AJM	4B29 CD682B	14490	CALL	2868H
4AA5 AF	13840	XOR	A	4B2C E1	14500	POP	HL
4AA6 CD1202	13850	CALL	ACC	4B2D AF	14510	XOR	A
4AA9 CD8702	13860	CALL	ACF	4B2E C3331F	14520	JP	1F33H

4B31 F1	14530	LINE1	POP	AF
4B32 F1	14540		POP	AF
4B33 C3BE1D	14550		JP	1DBEH
0032	14560		DEFS	50
4B68 00	14570	AJU	DEFB	0
0000	14580	AAA	EQU	0000H
0001	14590	AAB	EQU	0001H
0002	14600	AAC	EQU	0002H
0003	14610	AAD	EQU	0003H
0004	14620	AAE	EQU	0004H
0005	14630	AAF	EQU	0005H
0006	14640	AAG	EQU	0006H
0007	14650	AAH	EQU	0007H
0008	14660	AAI	EQU	0008H
000A	14670	AAJ	EQU	000AH
000D	14680	AAK	EQU	000DH
000F	14690	AAL	EQU	000FH
001F	14700	AAN	EQU	001FH
0020	14710	AAO	EQU	0020H
0023	14720	AAP	EQU	0023H
0024	14730	AAQ	EQU	0024H
0025	14740	AAR	EQU	0025H
002C	14750	AAT	EQU	002CH
002D	14760	AAU	EQU	002DH
0030	14770	AAV	EQU	0030H
0033	14780	AAW	EQU	0033H
003A	14790	AAX	EQU	003AH
003B	14800	AAY	EQU	003BH
003C	14810	AAZ	EQU	003CH
003F	14820	ABA	EQU	003FH
0040	14830	ABB	EQU	0040H
0048	14840	ABC	EQU	0048H
0049	14850	ABD	EQU	0049H
0055	14860	ABE	EQU	0055H
0058	14870	ABF	EQU	0058H
0060	14880	ABG	EQU	0060H
0078	14890	ABI	EQU	0078H
007F	14900	ABJ	EQU	007FH
0080	14910	ABK	EQU	0080H
00CB	14920	ABS	EQU	00CBH
00DD	14930	ABT	EQU	00DDH
00E0	14940	ABU	EQU	00E0H
00ED	14950	ABW	EQU	00EDH
00FD	14960	ABY	EQU	00FDH
00FF	14970	ABZ	EQU	00FFH
0100	14980	ACA	EQU	0100H
01FB	14990	ACB	EQU	01FBH
0212	15000	ACC	EQU	0212H
0235	15010	ACD	EQU	0235H
0264	15020	ACE	EQU	0264H
0287	15030	ACF	EQU	0287H
0296	15040	ACG	EQU	0296H
0A7F	15050	ACD	EQU	0A7FH
0A9A	15060	ACP	EQU	0A9AH
3C3F	15070	ACW	EQU	3C3FH
42E9	15080	ADH	EQU	MEM
42EA	15090	ADI	EQU	MEM+1
42EB	15100	ADJ	EQU	MEM+2
42ED	15110	ADK	EQU	MEM+4
42EF	15120	ADL	EQU	MEM+6
42C0	15130	END		START

00000 TOTAL ERRORS

15135 TEXT AREA BYTES LEFT


```

10 REM COPYRIGHT BY LUIDGER ROECKRATH, 2.1.83 (D 1.0)
20 CLS: CLEAR 4500: DEFINIT A-Z: DEF SNG A, V: DEF STRE, I, L, H: DIME (370), L (100), V (100), H (15): EE=" "+CHR$(0):
AC=LOG(10): AB=1.0001: DA=17129
25 FORM=1 TO 10: READE: NEXT: FORM=0 TO 15: READ H(M): NEXT
30 PRINT: PRINT "E=EDITOR / A=ASSEMBLER / D=DISASSEMBLER / M=MONITOR"
80 I=INKEY$: IF I="" THEN 80
90 IF I="E" THEN 200
100 IF I="A" THEN 760
110 IF I="D" THEN 1600
120 IF I="M" THEN 1340
130 F=8: GOSUB 150: GOTO 80
140 DATASYNTAX, INVALID OFCODE, INVALID OPERAND, INVALID NUMBER, INVALID LABEL, UNDEFINED LABEL, MULTI
PLE DECLARATION, INVALID COMMAND, OVERFLOW, MISSING OPERAND, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
150 FO=FO+1: RESTORE: FORM=1 TO FAND 15: READ I: NEXT
160 PRINT I; " ERROR";
170 IF (FAND 16)=16 THEN PRINT " IN LINE" VAL (E(0));
180 PRINT: RETURN
200 PRINT ">"; : LINE INPUT E
210 E1=LEFT$(E, 4)
240 IFE="LIST" THEN 390
250 IFE="LOAD" THEN 430
260 IFE="SAVE" THEN 400
270 IFE="EXIT" THEN PRINT CHR$(15); : GOTO 30
280 A=VAL (E): IFA=0 THEN F=8: GOSUB 150: GOTO 200
290 IF ASC (E) = 32 THEN E="A": GOTO 280
300 IFRIGHT$(E, LEN (E) - INT (LOG (VAL (E)) / AC + AB)) = "" THEN 370
310 IF VAL (E(N)) < ATHEN N=N+1: E(N)=E: GOTO 200
320 FORM=NT00STEP-1: IF VAL (E(M)) = ATHEN (M)=E: GOTO 200
330 IF VAL (E(M)) < ATHEN GOTO 350
340 NEXT: M=M+1
350 FOR O=N+1 TO M+1STEP-1: E(O)=E(O-1): NEXT
360 E(M+1)=E: N=N+1: GOTO 200
370 FORM=0 TO N: IF VAL (E(M)) < ATHEN NEXT: GOTO 200
380 FOR O=M TO N-1: E(O)=E(O+1): NEXT: N=N-1: GOTO 200
390 FORM=0 TO N: PRINT E(M): IF INKEY$="E" THEN 200 ELSE NEXT: GOTO 200
400 M=1: PRINT # -1, N
405 I=""
410 IF LEN (I) + LEN (E(M)) < 247 THEN I=I+E(M)+CHR$(128): M=M+1: IF M <= N THEN 410
420 PRINT # -1, CHR$(34); I; CHR$(34): IF M > N THEN 200 ELSE 405
430 INPUT # -1, N: M=1
440 INPUT # -1, I: B=0
450 FOR O=1 TO LEN (I): IF MID$(I, O, 1) = CHR$(128) THEN E(M)=MID$(I, B+1, O-B-1): B=O: M=M+1
460 NEXT
470 IF M > N THEN 200 ELSE 440
500 IFA > 32767 THEN POKEA-65536, DELSEPOKEA, D
510 A=A+1: RETURN
520 J=0: E=MID$(I, PEEK (DA+2), PEEK (DA+3))
530 L=LEFT$(E, 1): IFL < "A" OR L > "Z" THEN 570
540 IFF=0 THEN F=19: RETURN
550 FORM=0 TO K: IFL (M) = E THEN V=V(M): F=0: RETURN
560 NEXT: F=22: RETURN
570 IF J=0 THEN IF LEFT$(E, 1) = "$" THEN 680
575 F=0: IF ASC (E) = 39 THEN 700
580 IFL < "0" OR L > "9" THEN F=20: RETURN
590 V=0: L=RIGHT$(E, 1): IFL="H" THEN 660
600 IFL="B" THEN 640
610 IFL="D" THEN 630
620 IFL < "0" OR L > "9" THEN F=20: RETURN
630 V=VAL (E): GOTO 715
640 FORM=1 TO LEN (E) - 1: G=ASC (MID$(E, M, 1)) - 48: IF G < 0 OR G > 1 THEN F=20: RETURN
650 V=V+V+G: NEXT: GOTO 715
660 FORM=1 TO LEN (E) - 1: G=ASC (MID$(E, M, 1)) - 48: IF G > 16 AND G < 23 THEN G=G-7 ELSE IF G < 0 OR G > 9 THEN F=20: RETURN
670 V=16*V+G: NEXT: GOTO 715
680 IF LEN (E) < 3 THEN F=17: RETURN ELSE L=MID$(E, 2, 1): IFL="-" THEN J=-1 ELSE IFL="+" J=1 ELSE F=17: RETURN
690 E=RIGHT$(E, LEN (E) - 2): GOSUB 530: V=AA+J*V: RETURN
700 IF LEN (E) = 3 THEN V=ASC (MID$(E, 2, 1)): IF ASC (MID$(E, 3, 1)) = 39 THEN RETURN

```

```
710 F=17:RETURN
715 IFV>65535THENF=25:RETURNELSERETURN
720 O=O+1:IFE(O)=""THENF=11:RETURNELSEI=RIGHT$(E(O),LEN(E(O))-INT(LOG(VAL(E(O)))/AC+AB))+EE
730 IFASC(I)=59THEN720
740 V=USR(VARPTR(I)):F=PEEK(DA+4)
750 B=V/256:C=V-256*B:RETURN
760 O=O:K=-1:A=O:F=O:D=16526:V=DA+30:GOSUB1660
770 AA=A:IFINKEY$("<>")THEN30ELSEGOSUB720:IFF=11THENF=17:GOSUB150:GOTO910ELSEIFFTHENF=F+16:GOSUB150:GOTO770
780 D=(BAND224)/32
790 IFD=0THEN820ELSEL=LEFT$(I,D):IFK<0THEN810
800 FORM=0TOK:IFL(M)=LTHENF=23:GOSUB150:GOTO770ELSENEXT
810 K=K+1:L(K)=L:V(K)=A
820 D=BAND7:IFD=7THEN850
830 A=A+SGN(D)+1:IFD>=5THENA=A+2:GOTO770
840 D=BAND24:IFD=0THEN770ELSEA=A+1:IFD=16THENA=A+1:GOTO770ELSE770
850 IFC=3ORC=4THEN840ELSEIFC=7THEN910
860 IFC=6THEN900ELSEGOSUB520:IFFGOSUB150:GOTO770ELSEF=0:ONCGOSUB870,880,0,0,890:IFFGOSUB150:GOTO770ELSE770
870 A=V:IFBAND224THENV(K)=V:RETURNELSERETURN
880 IFBAND224THENV(K)=V:RETURNELSEF=26:RETURN
890 A=A+V:RETURN
900 A=A+PEEK(DA+3):GOTO770
910 PRINT"PASS 1 COMPLETE":O=O:A=O
920 IFINKEY$("<>")THEN30ELSEGOSUB720:IFF=11THEN1200ELSEIFFTHEN920
930 AA=A:D=BAND7:IFD=7THEN1110
940 X=O:G=O:OND+1GOSUB950,960,970,980,990,1000,1010:GOTO1020
950 RETURN
960 G=221:RETURN
970 G=253:RETURN
980 G=237:RETURN
990 G=203:RETURN
1000 G=221:X=203:RETURN
1010 G=253:X=203:RETURN
1020 D=G:IFGOSUB500
1030 D=X:IFXGOSUB500:A=A+1
1040 D=C:GOSUB500
1050 F=1:D=(BAND24)/8:IFD=0THEN920
1052 IFD=2ANDC=54THEN1170
1055 GOSUB520:IFFGOSUB150:GOTO920
1060 B=V/256:C=V-256*B:IFXTHEND=4
1065 F=0:ONDGOTO1070,1080,1100,1090
1066 IFFGOSUB150
1067 GOTO920
1070 D=C:GOSUB500:IFBTHENF=25:GOTO1066ELSE1066
1080 D=C:GOSUB500:IFB>255THENF=25:GOTO1066ELSED=B:GOSUB500:GOTO1066
1090 A=A-2:D=C:GOSUB500:A=A+1:IFBTHENF=25:GOTO1066ELSE1066
1100 V=V-A-1:D=1:IFV>127ORV<-128THENF=25:GOTO1066ELSEIFV<0THENV=V+256:GOTO1060ELSE1060
1110 F=0:ONCGOTO1130,920,1050,1050,1130,1140,1200
1130 GOSUB520:IFFTHEN920ELSEIFC=1THENA=V:GOTO920ELSEA=A+V:GOTO920
1140 FORM=PEEK(DA+2)TOM+PEEK(DA+3)-1:D=ASC(MID$(I,M,1)):GOSUB500:NEXT:GOTO920
1170 FORM=1TOLEN(I):IFMID$(I,M,1)=""THENF=MELSENEXT
1180 FORM=F+1TOLEN(I):IFMID$(I,M,1)=""THENG=MELSENEXT
1190 E=MID$(I,F+1,G-F-1):F=1:J=0:GOSUB530:IFFTHEN150ELSED=V:IFD>255THENF=25:GOSUB150:GOTO920ELSEGOSUB500:D=1:F=D:GOTO1055
1200 PRINT"ASSEMBLY COMPLETE,"FO"ERROR(S)":GOTO30
1300 C=B/16:PRINTH(C);:C=B-16*C:PRINTH(C);:RETURN
1330 B=A/256:GOSUB1300:B=A-256*B:GOTO1300
1340 INPUTI:I=I+" ":E=LEFT$(I,2):IFE="M "THEN1400
1350 IFE="J "THEN1470
1360 IFLEFT$(I,1)="L"THEN1510
1370 IFE="S "THEN1550
1380 IFE="E "THEN30
1385 F=8:GOSUB150:GOTO1340
1390 IFMID$(I,J+4,1)>"/"THENF=1:GOTO150ELSEE="0"+MID$(I,J,4)+"H":J=1:F=0:GOSUB530:IFFTHENF=F-16:GOTO150ELSERETURN
```

```
1400 J=3:GOSUB1390:A=V:IFFTHEN1340
1410 GOSUB1330:PRINT " ";:IFA<32768THENB=PEEK(A):ELSEB=PEEK(A-65536)
1420 GOSUB1300:I="":INPUT " ";I:IFI=""THENA=A+1:IFA=65536THENA=0:GOTO1410ELSE1410
1430 IFI="."THEN1340
1440 J=1:F=0:E="0"+I+"H":GOSUB530:IFFTHENF=F-16:GOSUB150:GOTO1410
1450 IFV<256THEND=V:GOSUB500:GOTO1410ELSE1410
1470 J=3:GOSUB1390:IFFTHEN1340
1480 GOSUB1500:INPUT"ARGUMENT:";B:B=USR(B):PRINT"ERGBNIS:"B:GOTO1340
1500 D=DA+6:GOSUB1660:D=16526:V=DA+1854:GOTO1660
1510 IFLEN(I)<3THENV=0ELSEI=I+" ":J=3:GOSUB1390:IFFTHEN1340
1520 D=DA:GOSUB1660
1530 V=DA+1881:GOSUB1500:J=6:GOSUB1540:B=USR(A):GOTO1340
1540 B=VARPTR(I):A=PEEK(B+1)+256*PEEK(B+2)+J:IFA>32767THENA=A-65536:RETURNELSEReturn
1550 I=I+CHR$(213)
1560 J=3:GOSUB1390:IFFTHEN1340ELSEDA=DA:GOSUB1660
1570 J=8:GOSUB1390:V=V+1:IFFTHEN1340ELSEDA=DA+2:GOSUB1660
1580 J=13:GOSUB1390:IFFTHEN1340ELSEDA=DA+2080:GOSUB1660
1590 V=DA+1973:GOSUB1500:J=17:GOSUB1540:B=USR(A-1):GOTO1340
1600 V=DA+1541:GOSUB1500
1610 INPUT"ADRESSE:";I:I=I+" ":J=1:GOSUB1390:IFFTHEN1610
1620 IFV>32767THENV=V-65536
1630 A=V:V=USR(V):IFV=ATHENF=2:GOSUB150:GOTO1610
1640 I=INKEY$:IFI=""THEN1640ELSEIFI="E"THEN30
1650 IFI="D"THENPRINT:GOTO1610ELSE1630
1660 IFV<0THENV=V+65536
1670 POKED,V-256*INT(V/256):POKED+1,V/256:RETURN
```

6.8 Konvertierungsprogramme

- 148 -

```
10 REM (C) 1981 Luidger Röckrath
20 CLEAR1000:DEFINT A-Z
30 CLS : PRINT "Umwandlung einer Source für MACRO-80"
40 PRINT "in eine EDTASM-Source"
50 PRINT
60 INPUT "/MAC-File";SF$
70 INPUT "/SRC-File";DF$
80 IF INSTR(SF$,"/")=0 THEN SF$=SF$+"/MAC"
90 D$=DF$:IF INSTR(DF$,"/")=0 THEN DF$=DF$+"/SRC"
100 OPEN "I",1,SF$:OPEN "O",2,DF$
110 PRINT#2,CHR$(%HD3);LEFT$(D$+"      ",6);
120 NR=10
130 LINEINPUT#1,A$
140 PRINTA$
150 GOSUB 310
160 F=0:F1=0
170 FOR I=1 TO LEN(A$)
180 M$=MID$(A$,I,1)
190 IF M$=" " AND F=0 THEN 270
200 IF M$=":" AND F1=0 THEN F1=1:M$=" ";J=I:GOTO 290
210 PRINT#2,M$;
220 IF M$="?" THEN F=ABS(F-1)
230 NEXT
240 PRINT#2,
250 IF NOT EOF(1) THEN 130
260 PRINT#2,CHR$(26);:CLOSE:END
270 J=I
280 IF MID$(A$,I,1) <> " " THEN 300
290 IF INT(I/8)=I/8 THEN PRINT#2,CHR$(9);:IF I=LEN(A$) THEN 240 ELSE I=I+1:GOTO 270 ELSE I=I+1:GOTO 280
300 PRINT#2,STRING$(I-J," ");:F1=1:GOTO 180
310 NR$=MID$(STR$(NR),2)
320 NR$=RIGHT$("000"+NR$,3)
330 FOR K=1 TO 5
340 MID$(NR$,K,1)=CHR$(ASC(MID$(NR$,K,1)) + 128)
350 NEXT
360 PRINT#2,NR$;" ";
370 NR=NR+10
380 RETURN
```

```
50 'CONV/BAS :34
60 '7.4.81
70 'WERNER ZANG
80 '
90 '** CONVERTIERT EDTASM/SRC CODE ZU M80/SRC CODE *
100 CLEAR1000:DEFINTA-Z
110 INPUT"SRC-FILE";SF$
120 INPUT"DEST-FILE";DF$
130 OPEN"I",1,SF$
140 OPEN"O",2,DF$
150 LINEINPUT#1,A$
160 A$=MID$(A$,8,250)
170 GOTO190
180 LINEINPUT#1,A$
190 I=1:K=1:DP$="":PO=1
195 C$=""
200 B$=MID$(A$,7,1)
210 IFB$=CHR$(9)ORB$=" "ORB$="";THEN300
215 IFB$="*"THEN180
220 C$=C$+B$
230 FORJ=2TO8
240 B$=MID$(A$,J+6,1)
245 IFB$="":THENDP$=" ":GOTO260
50 IFB$=CHR$(9)THENTB=INT((J-1)/8)*8+8-J:PO=J:B$=DP$+STRING$(TB," "):K=J+1:I=J+TB+1:J=10:ELSEIF
B$=" "THENPO=J:B$=DP$:K=J+1:I=J+1:J=10
260 C$=C$+B$
270 NEXTJ
300 'BEFEHL SUCHEN
310 LN=LEN(A$)
330 FORK=KTOLN
340 B$=MID$(A$,K+6,1)
350 IFB$=CHR$(9)THENTB=INT((I-1)/8)*8+8-I+1:B$=STRING$(TB," "):I=I+TB:C$=C$+B$:GOTO390
360 IFB$<>" "THEN:L=K:K=LN:GOTO390
370 I=I+1
380 C$=C$+B$
390 NEXTK
500 IFMID$(A$,L+6,3)<>"EGU"THEN600ELSEMID$(C$,PO,1)=" "?:" ENTFERNEN
510 FORJ=L+3TOLN
520 MS$=MID$(A$,J+6,1)
530 IFMS$<>" "ANDMS$<>CHR$(9)THENK=J:J=LN
580 NEXTJ
590 IFMS$=" $"THENL=K+1:MID$(C$,PO,1)=" ":ELSE
600 FORK=LTOLEN(A$)-6
610 B$=MID$(A$,K+6,1)
620 IFB$=CHR$(9)THENTB=INT((I-1)/8)*8+8-I+1:B$=STRING$(TB," "):I=I+TB:ELSEI=I+1
630 C$=C$+B$
640 NEXTK
650 PRINT#2,C$
655 PRINTC$
660 IFNOTEOF(1)THEN180
670 CLOSE
```

Luidger Röckrath:

Das Maschinensprachebuch für TRS-80/
Genie/Colour-Genie

Dieses Buch ist eine wichtige Hilfe in der Maschinenprogrammierung auf den genannten Systemen.

Alle wichtigen ROM-Routinen werden ausführlich beschrieben. Hierbei werden neben der Einsprungsadresse und den Ein-/ausgabe-Parametern auch Beispiele für ihre Anwendung gegeben.

Außerdem werden alle Systeme verglichen, so daß der Leser in die Lage versetzt wird, Programme zu schreiben, die auf allen Systemen laufen.

Schließlich enthält das Buch Sourcelistings zahlreicher Beispielprogramme und Utilities, die wichtige Programmier Techniken erläutern, und zudem selbst sinnvoll einsetzbar sind.

ISBN 3-925074-07-4