

Disassembled Handbook for TRS-80

Richcraft Engineering Ltd.
Drawer 1065
Chautauqua, New York 14722

DEUTSCHE AUSGABE

Copyright © 1980

"TRS-80 Is A Registered Trademark of TANDY CORP."

Erschienen im Eigenverlag

Guenther Daubach Ing. (grad)
Ingenieurbuero fuer Datentechnik

Franziskanerplatz 4

D-8933 Klosterlechfeld

Telefon 08232/6320

T R S - 8 0 H A N D B U C H

B A N D 1

Deutsche Ausgabe des

'Disassembled Handbook for TRS-80 Vol.1'

1. Auflage Januar 1981

Copyright (C) 1980 der englischen Ausgabe:

Richcraft Engineering Ltd.
Drawer 1065
Chautauqua, New York 14722
USA

Alle Rechte der deutschen Ausgabe:

Guenther Daubach Ing. (grad)
Ingenieurbuero fuer Datentechnik
Franziskanerplatz 4
D-8933 Klosterlechfeld

Dieses Buch darf ohne ausdrueckliche schriftliche Genehmigung des Urhebers in keiner Form kopiert oder vervielfaeltigt oder auf Informationstraeger jeder Art aufgezichnet werden.

TRS-80 ist ein Warenzeichen der Fa. TANDY Corporation. Die Anwender der in diesem Buch beschriebenen Level-II ROM Routinen muessen notwendigerweise Level-II BASIC gekauft und damit die Nutzungsrechte von den Inhabern des Urheberrechts dieses Programms erworben haben. Alle Angaben in diesem Buch haben reinen Charakter von Erlaeuterungen und dienen als Ergaenzung der Originaldokumentation, wie von Radio Shack geliefert. Sowohl Fa. Richcraft als auch der Uebersetzer liefern grundsaeztlich keine Kopie dieser Dokumentation. Damit ist sichergestellt, dass alle Benutzer dieses Buches das Level-II ROM von Radio Shack erworben haben und damit die entsprechende finanzielle Leistung erbracht wurde. Weiterhin sind in den Ausdrucken des Level-II ROM eine angemessene Zahl von Speicherinhalten ausgelassen worden, so dass dieser Ausdruck fuer die Leser nutzlos ist, die nicht im Besitz des urheberrechtlich geschuetzten Programms sind.

Vorwort zur deutschen Auflage

Seit etwa zwei Jahren setzte ich den TRS-80, Level II der Fa. Radio Shack für die unterschiedlichsten Aufgaben mit gutem Erfolg ein. Dabei hat sich oft gezeigt, daß bestimmte Anwendungen einfacher zu realisieren gewesen wären, wenn eine ausreichende Dokumentation über die bereits im System enthaltene Software zur Verfügung gestanden hätte.

Bruchstückhafte Informationen wurden im Laufe der Zeit aus den hauptsächlich in den USA erscheinenden Fachzeitschriften zusammengestellt. Kein Wunder also, daß ich sofort nach Erscheinen der 'Disassembled Handbooks' mir diese so schnell wie möglich beschafft habe. Die Notwendigkeit, die dort enthaltene Information einem größeren Benutzerkreis im deutschsprachigen Raum zugänglich zu machen, wurde bald erkannt. Um die Sprachbarriere abzubauen, fiel dann die Entscheidung, eine deutsche Uebersetzung anzufertigen.

Nach langen Stunden vor dem Bildschirm liegt nun, lieber Leser, die erste Auflage der deutschen Ausgabe vor Ihnen. Hier sei der Fa. Radio Shack ein Kompliment gemacht: ohne das ausgezeichnete Textverarbeitungsprogramm 'SCRIPSIT' hätte die Unterlage nie in der kurzen Zeit erstellt werden können.

Es bleibt nun noch, Mr. Bob Richardson zu danken, daß er die Fülle von Informationen zusammengestellt hat und er sofort bereit war, einer Uebersetzung in die deutsche Sprache zuzustimmen.

Ihnen, lieber Leser, wünsche ich zunächst viel Freude beim Durcharbeiten des Buches und dann viel Erfolg für Ihre weitere Programmierarbeit, bei der Sie die hier vermittelten Informationen möglichst oft anwenden mögen.

Zum Schluß noch eine Bitte: Ihre Anregungen, wie diese Buchreihe verbessert und erweitert werden kann, sind eine große Hilfe. Behalten Sie sie daher nicht für sich, sondern fühlen Sie sich frei, mir diese mitzuteilen. Der Druckfehler wird seine Spuren sicherlich auch in diesem Buch hinterlassen haben; damit ich ihn austreiben kann, würde ich mich auch über solche Hinweise freuen.

Klosterlechfeld im Januar 1981


Günther Daubach

v. S. 64 → sehr nett!

Vorwort zur englischen Ausgabe

Dieses Handbuch ist aus einer Reihe von Manuskripten entstanden, die der Verfasser als "Füllroutinen" erstellt hat. Diese wurden anlässlich einer Vortragsreise gehalten, bei der er sein neues Buch "Gunnplexer Cookbook - a 10 GHz Mikrowave Primer" vorstellte. Es ist ein ziemlich weiter Sprung (in Wellenlänge oder Frequenz) von den 1,77 MHz Taktfrequenz zum 10250 MHz-Band, in dem der Gunnplexer arbeitet, aber erstaunlicherweise haben der Computer- und der Mikrowellenenthusiast mehr gemeinsam, als man meinen sollte. So müssen beide eine Art von Kommunikation durchführen, sei es mit Hilfe der Frequenz-Sprachmodulation oder im ASCII-Code über eine Datenverbindung.

Es war gleichzeitig zufriedenstellend und überraschend, daß viele Mikrowellen-Funkamateure genausoviel Interesse für die Programmierung des TRS-80 in Assemblersprache zeigten, wie es TRS-80 Freunde für den Aufbau einer preisgünstigen Datenübertragungsstrecke in Mikrowellentechnik hatten. Die Welt ist nun einmal klein und die meisten Dinge hängen in irgend einer Weise zusammen.

Dieses Handbuch ist NICHT für den Anfänger der Assemblerprogrammierung gedacht, der gewiss zunächst die grundlegenden Programmieretechniken kennenlernen sollte, bevor er versucht, die Möglichkeiten der vielen ausgezeichneten Level-II ROM-Routinen zur Verkürzung seiner Programme anzuwenden. Der einigermaßen erfahrene Assembler-Programmierer, der den Unterschied zwischen JP, JR, SET, RESET und Bits und Bytes versteht, wird bald erkennen, daß die Benutzung der ROM-Routinen ein völlig neues Bild der "wunderbaren Welt" der Assembler-Programmierung schafft. Viele nervenraubende, aber notwendige Routinen lassen sich nun oft durch einen einzigen CALL-Befehl ersetzen und ersparen zeilen- oder seitenlange Programme, die Zeile für Zeile eingegeben werden mußten. Programmierer aufgepasst! Schüttelt das Joch der Ignoranz von Euch und entdeckt die Freude am Erstellen von Programmen in Assemblersprache, die bis zu 300 mal schneller arbeiten und nur ein Zehntel des Speicherbedarfes haben, wie gleichwertige BASIC-, COBOL-, FORTRAN- oder PASCAL-Programme.

Ein besonderes DANKESCHÖN gilt Mr. Charles Tandy für seinen Mut und seine Investition, mit dem TRS-80 ein so gutes Gerät geschaffen zu haben.

E I N F U E H R U N G

Disassemblierte Maschinenprogramme -in jeder Art- OHNE Kommentare sind genauso wertlos, wie Ihr TRS-80 ohne Netzspannung. Konstante, Tabellen mit Sprungadressen und Listen mit Daten werden von einem Disassembler in unsinnigen oder sogar verwirrenden Nonsens uebersetzt. So wird z.B. der Inhalt der Level II-ROM-Adressen 005AH, 1479H, 1619H, 18BAH, 18BCH u.s.w. mit EX AF,AF' uebersetzt, obwohl in keinem Abschnitt des Level II-BASIC die zweite Registerbank des Z-80 verwendet wird.

Will man ein Objekt-Programm decodieren und einen gewissen Sinn in dieses Programm bringen, so ist das nur moeglich, wenn man zumindest eine gewisse Ahnung von dem hat, was das Programm eigentlich bewirken soll. Zum Glueck wissen wir beim Level II ROM aber sehr genau, welche Funktionen und Moeglichkeiten hier programmiert sind. Somit ist unser zunaechst unmoegliches Vorhaben zu einem "nur noch" ausserordentlich schwierigen Unternehmen geworden. In der Tat so schwierig, dass anscheinend sogar die Computerabteilung von Radio Shack in Fort Worth, Texas nicht so ganz genau dieses exzellente Programm versteht. An dieser Stelle sei vermerkt dass das Programm von Paul Allen und Bill Gates bei Microsoft, USA geschrieben wurde. Ein Beispiel, das die Behauptung ueber Radio Shack erhaertet, ist das von ihnen herausgegebene Buch "TRS-80 Assembly Language Programming", das Mitte 1979 erschien und praktisch keine Hinweise auf die vielseitigen Level II ROM Routinen gibt. Das ist nahezu unglaublich und fast der Gipfel der Absurditaet. Entweder beruht das auf einer boeswilligen Absicht von Radio Shack oder auf Unkenntnis. Nehmen wir "zu Gunsten des Angeklagten" Letzteres an!

Wie gross die Schwierigkeiten sind, die beim Decodieren des Level II ROM's auftreten, mag man daran erkennen, dass nach nunmehr einigen Jahren, in denen der TRS-80 weltweit von ueber 200.000 Benutzern eingesetzt wird (vom Anfaenger bis zum fortgeschrittenen Programmierer) noch nie ALLE Routinen und Funktionen des Level II ROM's zusammengestellt und mit Kommentaren veroeffentlicht wurden. Eine Ausnahme war ein Vortrag der von Mr. Andrew Hildebrand vor einem kleinen Zuhorerkreis gehalten wurde

Dieses Handbuch ist Mr. Hildebrand und seiner Unermuedlichkeit gewidmet, das verwirrte Netz der Level II Routinen zu entwirren. Die einzige Muehe des Autors (und des Uebersetzers) war es, die von ihm gewonnenen Erkenntnisse in einen Text umzusetzen, der es einem grossen Leserkreis ermoeglicht, Licht in einige der "schwarzen Loecher" zu bringen. Dieses Handbuch kann auch fuer all die Leser interessant sein, die einen anderen Computer benutzen, der ebenfalls Microsoft-BASIC benutzt (z.B.: Apple, PET, CBM, KIM, Heathkit u.s.w.), da diese BASIC-Versionen aehnlich sind.

Jeder Abschnitt dieses Handbuchs ist so aufgebaut, dass er sich hoffentlich selbst erklart. Der Hex-Code des Level II ROM ist in Kapitel 6 abgedruckt; obwohl sehr lang, wurde er als notwendige Referenz abgedruckt. Einige Wiederholungsfragen und -antworten in Kapitel 9 und 10 erlauben Ihnen, lieber Leser, ihren Lernfortschritt nach jedem Kapitel zu ueberpruefen.

ZUSAMMENFASSUNG:

Kapitel 1 beschreibt wie die Adressen der meisten ROM-Routinen decodiert werden koennen.

Kapitel 2 enthaelt drei Quellen- und Objektprogramme und eine Beschreibung zum Gebrauch der einfachen arithmetischen Routinen (+ - x /) fuer ganze, einfach und doppelt genaue Zahlen.

Kapitel 3 zeigt die Anwendung der trigonometrischen, Exponent- und Logarithmus-Funktionen sowie u.a. der Routinen CINT, CSNG und CDBL im Zusammenhang mit dem RAM-Akkumulator und dem CDBL-Speicher sowie ein Demonstrationsprogramm.

Kapitel 4 bringt eine Reihe weiterer nuetzlicher Level II Routinen und

Kapitel 5 ist eine Zusammenfassung praktisch aller BASIC-Funktionsaufrufe.

Kapitel 6 listet den Hexadezimal-Code des ROM's auf.

Kapitel 7 beschreibt ein sehr nuetzliches Programm zur Umwandlung von einem Zahlensystem in ein anderes, wie es von jedem ernsthaften Assembler Programmierer benoetigt wird. Dieses Programm ist in BASIC geschrieben, so dass es gut verstaendlich ist und vom Anwender veraendert werden kann und enthaelt folgende Umwandlungen:

- dezimal nach binaer (D)
- binaer nach dezimal (B)
- hexadezimal nach binaer (HB)
- dezimal nach hexadezimal (DH)
- hexadezimal nach dezimal (HD)
- zweiteilig dezimal nach dezimal (SP)
- dezimal nach zweiteilig hexadezimal (DS)
- zweiteilig hexadezimal nach dezimal (SD)

Die ersten fuef Umwandlungen verstehen sich von selbst. Die sechste Umwandlung wird benoetigt, wenn mit PEEK Adressdaten von zwei aufeinanderfolgenden Speicherplaetzen gelesen werden. Diese beiden Werte werden in hexadezimal umgewandelt, gemaess Z-80 Adressierungsverfahren vertauscht und aus der so gewonnenen, vierstelligen Hexadezimalzahl wieder eine Dezimalzahl berechnet. Die weiteren Umwandlungen arbeiten entsprechend.

Kapitel 8 ist ein nuetzliches Hilfsprogramm fuer Anwender mit Kugelkopfdruckern um alle Nullen mit einem Schraegstrich zu drucken.

Kapitel 9 enthaelt die Selbsttestfragen und Kapitel 10 bringt ein Literaturverzeichnis und die Antworten zu den Selbsttestfragen.

K A P I T E L 1

Einführung:

Während der letzten Jahre ist das Level II BASIC, das von Microsoft für den TRS-80 geschrieben wurde, de facto zu einem Standard-BASIC geworden, und es wird von fast allen wichtigen Computerherstellern verwendet. Bis Januar 1980 wird die Anzahl der an Endverbraucher ausgelieferten Mikrocomputer mit Level II BASIC oder leicht geänderten Versionen auf ca. 300.000 Stück geschätzt. Betrachtet man die Vielzahl der unterschiedlichen konkurrierenden BASIC-Versionen am Markt, Hewlett-Packard BASIC, General Electric BASIC und auch das IBM VS-BASIC, um einige zu nennen, so muß es eine ganze Reihe von Gründen geben, warum das Microsoft-BASIC sooft in den unterschiedlichen Rechnern eingesetzt wurde:

1. Ist es preiswert? Antwort: Nein, tatsächlich ist dieses BASIC sogar ausgesprochen teuer. Heathkit berechnet nicht umsonst seinen Hobby-Kunden 100 Dollar für dieses Programm, nur damit sie Spiele programmieren können.

2. Ist es effektiv? Antwort: Sie können sicher sein, es ist! Andere Ausgaben von BASIC-Interpretern, die der Autor untersucht hat, erreichen selbst mit einem Umfang von 22 bis 32 KByte nicht die gleichen Funktionen.

3. Ist es kosteneffektiv? Antwort: trotz des hohen Lizenzpreises ist es dennoch sehr kosteneffektiv, wenn man berücksichtigt, daß von den maximal 64 KByte an Speicherkapazität, die die heutigen 8 Bit Prozessoren adressieren können, möglichst viel für Anwenderprogramme freibleiben sollte. Einige der anderen BASIC-Programme lassen, wenn man sie mit einem Diskettenbetriebssystem lädt, oft nur einen Platz von 18 KByte für Anwenderprogramme. Das ist bei einer maximalen Speicherkapazität von 64 KByte schon fast unzumutbar.

Lassen Sie uns, lieber Leser nun einen Blick auf die unglaubliche Anzahl der Funktionen des Level II ROM und ihre Aufrufadressen in diesem Dschungel von Maschinencode werfen. Wenn man die vielen Programmsprünge und -verzweigungen sieht, könnte man meinen, es sei die Absicht der Autoren, Paul Allen und Bill Gates gewesen, das Programm so zu verschlüsseln, daß man es nicht decodieren kann. Verschlüsselung kostet jedoch Speicherplatz und damit indirekt auch Geld, wie oben gesagt. Der Hauptgrund für diesen Programmaufbau dürfte die beabsichtigte Kompaktheit sein. Nichts ist überflüssig, kein einziges Byte ist ungenutzt und es gibt keine leicht zu entziffernden Zeiger die sagen "hier bin ich, nun benutze mich!"

WO SIND DIE BASIC-FUNTIONSNAMEN GESPEICHERT ?

Die Namen der Level II-Funktionen kann man noch am leichtesten finden. Nur ein sehr schlechter Schüler würde die Daten im Speicherbereich von 5712 bis 6172 übersehen. In Abb. 1 ist ein einfaches Programm gezeigt, das Ihnen diese Funktionsnamen und deren Lage im ROM auf den Bildschirm bringt. Vom jeweils ersten Buchstaben eines Namens wird 128 subtrahiert, um das entsprechende ASCII-Zeichen zu erzeugen. Abb.2 zeigt einen Ausdruck dieses Programms. Beachten Sie bitte, daß die angegebenen Adressen die Position des Namens im ROM angeben, NICHT die Adresse, unter der die Funktion aufgerufen werden kann.

```

10 'MICROSOFT BASIC LISTE DER FUNKTIONSNAMEN IM ROM
15 '
20 CLS:FOR N=5712 TO 6175 : Y=PEEK(N) : IF Y > 127 THEN
  Y=Y-128 : M=N
25 Z=N+1 : IF PEEK(Z) > 127 THEN PRINT CHR$(Y);" =";
  ELSE 35
30 PRINT M, : GOTO 40
35 PRINT CHR$(Y);
40 NEXT
  
```

- Abb. 1 -

END	= 5712	FOR	= 5715	RESET	= 5718	SET	= 5723
CLS	= 5726	CMD	= 5729	RANDOM	= 5732	NEXT	= 5738
DATA	= 5742	INPUT	= 5746	DIM	= 5751	READ	= 5754
LET	= 5758	GOTO	= 5761	RUN	= 5765	IF	= 5768
RESTORE	= 5770	GOSUB	= 5777	RETURN	= 5782	REM	= 5783
STOP	= 5791	ELSE	= 5795	TRON	= 5799	TROFF	= 5803
DEFSTR	= 5808	DEFINT	= 5814	DEFENG	= 5820	DEFDBL	= 5826
LINE	= 5832	EDIT	= 5836	ERROR	= 5840	RESUME	= 5845
OUT	= 5851	ON	= 5854	OPEN	= 5856	FIELD	= 5860
GET	= 5865	PUT	= 5868	CLOSE	= 5871	LOAD	= 5876
MERGE	= 5880	NAME	= 5885	KILL	= 5889	LSET	= 5893
RSET	= 5897	SAVE	= 5901	SYSTEM	= 5905	LPRINT	= 5911
DEF	= 5917	POKE	= 5920	PRINT	= 5924	CONT	= 5929
LIST	= 5933	LLIST	= 5937	DELETE	= 5942	AUTO	= 5948
CLEAR	= 5952	CLOAD	= 5957	CSAVE	= 5962	NEW	= 5967
TAB(= 5970	TO	= 5974	FN	= 5978	USING	= 5978
VARPTR	= 5983	USR	= 5989	ERL	= 5992	ERR	= 5995
STRING\$	= 5998	INSTR	= 6005	POINT	= 6010	TIME\$	= 6015
MEM	= 6020	INKEY\$	= 6025	THEN	= 6029	NOT	= 6033
STEP	= 6036	+	= 6040	-	= 6041	*	= 6042
/	= 6043	[= 6044	AND	= 6045	OR	= 6048
=	= 6051	>	= 6050	<	= 6052	SCN	= 6053
INT	= 6056	ABS	= 6059	FRE	= 6062	INP	= 6065
POS	= 6068	SQR	= 6070	RND	= 6074	LOC	= 6077
EXP	= 6080	COS	= 6083	SIN	= 6086	TAN	= 6089
ATN	= 6092	PEEK	= 6095	CVI	= 6099	CVS	= 6102
CVD	= 6103	EOF	= 6106	LOC	= 6111	LOF	= 6114
MKI\$	= 6117	MKS\$	= 6120	MKD\$	= 6125	CINT	= 6129
CSNG	= 6133	CDBL	= 6137	FIX	= 6141	LEN	= 6144
STR\$	= 6147	VAL	= 6150	ASC	= 6154	CHR\$	= 6157
LEFT\$	= 6161	RIGHT\$	= 6165	MID\$	= 6172		

WO STEHEN DIE ADRESSEN DER BASIC-FUNKTIONEN ?

Nun wird die Decodierung schon ein wenig interessanter, aber noch nicht schwierig, da hier keine besonderen Tricks angewendet werden. Man muß nicht weit im ROM suchen, um die CALL-Adressen der BASIC-Funktionen zu finden.

Diese Adressen sind in zwei Gruppen unterteilt. Die erste Gruppe beginnt einige Bytes nach dem Ende der Namensliste bei der Adresse 6178 und geht bis 6351. Diese Gruppe enthält alle Funktionen von 'END' bis 'kleiner als'. Die zweite Gruppe beginnt bei Adresse 5640 und endet mit Adresse 5711. Diese Gruppe enthält alle BASIC-Funktionen von 'SGN' bis zum Ende der Funktionsliste, also 'MID\$'.

Mit Ausnahme der Funktionen von 'TAB' bis 'kleiner als' werden alle CALL-Adressen im ROM unter dem üblichen Z-80 Format gespeichert, d.h. erst LSB, dann MSB in der nächstfolgenden Speicheradresse.

Abb. 3 zeigt ein kleines BASIC-Programm, das die einzelnen Funktionsnamen, ein Gleichheitszeichen, die Adresse unter der die jeweilige CALL-Adresse gespeichert ist und schließlich die CALL-Adresse selbst in der Form LSB MSB ausgibt.

```
10 'PROGRAMM ZUM AUSDRUCK VON BASIC-FUNKTIONEN MIT
15 'ZUGEHÖRIGEN CALL-ADRESSEN
20 CLS:PRINT"FUNKT=ADRESSE      LSB-MSB      . FUNKT=ADRESSE
  LSB-MSB"
25 A=6176:FOR X=5712 TO 6175:Y=PEEK(X):IF Y > 127 THEN
  Y=Y-128
30 Z=X+1:IF PEEK(Z) > 127 THEN PRINT CHR$(Y); "=";ELSE 45
35 A=A+2:IF A=6352 THEN A=5640
40 PRINT A, PEEK(A);"-";PEEK(A+1), : GOTO 50
45 PRINT CHR$(Y);
50 NEXT
55 END
```

In Abb. 4 ist ein Ausdruck dieses Programms dargestellt. Es werden hier nur die Adressen angegeben, deren LSB/MSB direkt eine CALL-Adresse ausdrückt. Die übrigen Funktionen werden in Kapitel 5 berücksichtigt. Abb. 5 ist ein Ausdruck der ROM Adressen 1600H bis 18FFH der zeigt, wie die CALL-Adressen der Funktionen gefunden werden können.

FUNKT=ADRESSE	LSB-MSB	FUNKT=ADRESSE	LSB-MSB
END = 6173	174 - 29	FOR = 6180	161 - 28
RESET = 6182	56 - 1	SET = 6184	53 - 1
CLS = 6186	201 - 1	CMD = 6188	115 - 65
RANDOM = 6190	211 - 1	NEXT = 6192	182 - 34
DATA = 6194	5 - 31	INPUT = 6196	154 - 33
DIM = 6198	8 - 29	READ = 6200	239 - 33
LET = 6202	33 - 31	GOTO = 6204	194 - 30
RUN = 6206	163 - 30	IF = 6208	57 - 32
RESTORE = 6210	145 - 29	GOSUB = 6212	177 - 30
RETURN = 6214	222 - 30	REM = 6216	7 - 31
STOP = 6218	169 - 29	ELSE = 6220	7 - 31
TRON = 6222	247 - 29	TROFF = 6224	248 - 29
DEFSTR = 6226	0 - 30	DEFINT = 6228	3 - 30
DEFSNG = 6230	6 - 30	DEFDBL = 6232	9 - 30
LINE = 6234	163 - 65	EDIT = 6236	96 - 46
ERROR = 6238	244 - 31	RESUME = 6240	175 - 31
CUT = 6242	251 - 42	ON = 6244	108 - 31
OPEN = 6246	121 - 65	FIELD = 6248	124 - 65
GET = 6250	127 - 65	PUT = 6252	130 - 65
CLOSE = 6254	133 - 65	LOAD = 6256	136 - 65
MERCE = 6258	139 - 65	NAME = 6260	142 - 65
KILL = 6262	145 - 65	LSET = 6264	151 - 65
RSET = 6266	154 - 65	SAVE = 6268	160 - 65
SYSTEM = 6270	178 - 2	LPRINT = 6272	103 - 32
DEF = 6274	91 - 65	POKE = 6276	177 - 44
PRINT = 6278	111 - 32	CONT = 6280	228 - 29
LIST = 6282	46 - 43	LLIST = 6284	41 - 43
DELETE = 6286	198 - 43	AUTO = 6288	8 - 32
CLEAR = 6290	122 - 30	CLOAD = 6292	31 - 44
CSAVE = 6294	245 - 43	NEW = 6296	73 - 27
TAB(= 6298	121 - 121	TO = 6300	124 - 124
FN = 6302	127 - 80	USING = 6304	70 - 219
VARPTR = 6306	10 - 0	USR = 6308	0 - 127
ERL = 6310	10 - 244	ERR = 6312	10 - 177
STRING\$ = 6314	10 - 119	INSTR = 6316	12 - 112
POINT = 6318	12 - 161	TIME\$ = 6320	13 - 229
MEM = 6322	13 - 120	INKEY\$ = 6324	10 - 22
THEN = 6326	7 - 19	NOT = 6328	7 - 71
STEP = 6330	8 - 162	+ = 6332	8 - 12
- = 6334	10 - 210	* = 6336	11 - 199
/ = 6338	11 - 242	[= 6340	11 - 144
AND = 6342	36 - 57	> = 6344	70 - 83
OR = 6344	10 - 78	= = 6348	78 - 82
< = 6350	71 - 79	SGN = 5640	138 - 9
INT = 5642	55 - 11	ABS = 5644	119 - 9
FRE = 5646	212 - 39	INP = 5648	239 - 42
POS = 5650	245 - 39	SQR = 5652	231 - 19
RND = 5654	201 - 20	LOG = 5656	9 - 8
EXP = 5658	57 - 20	COS = 5660	65 - 21
SIN = 5662	71 - 21	TAN = 5664	168 - 21
ATN = 5666	189 - 21	PEEK = 5668	170 - 44
CVI = 5670	82 - 65	CVS = 5672	88 - 65
CVD = 5674	94 - 65	EOF = 5676	97 - 65
LOC = 5678	100 - 65	LOF = 5680	103 - 65
MKI\$ = 5682	106 - 65	MKS\$ = 5684	109 - 65
MKD\$ = 5686	112 - 65	CINT = 5688	127 - 10
CSNG = 5690	177 - 10	CDBL = 5692	219 - 10
FIX = 5694	38 - 11	LEN = 5696	3 - 42
STR\$ = 5698	54 - 40	VAL = 5700	197 - 42
ASC = 5702	15 - 42	CHR\$ = 5704	31 - 42
LEFT\$ = 5706	97 - 42	RIGHT\$ = 5708	145 - 42
MID\$ = 5710	154 - 42		

1600	6CAA	AA7F	0000	0081	8A09	370B	7709	D4277.....'
1610	EF2A	F527	E713	C914	0908	3914	4115	4715	.*.'.....9.A.G.
1620	A815	BD15	AA2C	5241	5841	5E41	6141	6441,RAXA.A.A.A
1630	6741	6A41	6D41	7041	7F0A	B10A	DB0A	260B	.A.A.A.A.....&
1640	032A	3628	C52A	0F2A	1F2A	612A	912A	9A2A	.*6(*.*.*.*.*.*
1650	C54E	44C6	4F52	D245	5345	54D3	4554	C34C	.ND.OR.ESET.ET.L
1660	53C3	4D44	D241	4E44	4F4D	CE45	5854	C441	S.MD.ANDOM.EXT.A
1670	5441	C94E	5055	54C4	494D	D245	4144	CC45	TA.NPUT.IM.EAD.E
1680	54C7	4F54	4FD2	554E	C946	D245	5354	4F52	T.OTO.UN.F.ESTOR
1690	45C7	4F53	5542	D245	5455	524E	D245	4DD3	E.OSUB.ETURN.EM.
16A0	544F	50C5	4C53	45D4	524F	4ED4	524F	4646	TOP.LSE.RON.ROFF
16B0	C445	4653	5452	C445	4649	4E54	C445	4653	.EFSTR.EFINT.EFS
16C0	4E47	C445	4644	424C	CC49	4E45	C544	4954	NG.EFDBL.INE.DIT
16D0	C552	524F	52D2	4553	554D	45CF	5554	CF4E	.RROR.ESUME.UT.N
16E0	CF50	454E	C649	454C	44C7	4554	D055	54C3	.PEN.IELD.ET.UT.
16F0	4C4F	5345	CC4F	4144	CD45	5247	45CE	414D	LOSE.OAD.ERGE.AM
1700	45CB	494C	4CCC	5345	54D2	5345	54D3	4156	E.ILL.SET.SET.AV
1710	45D3	5953	5445	4DCC	5052	494E	54C4	4546	E.YSTEM.PRINT.EF
1720	D04F	4B45	D052	494E	54C3	4F4E	54CC	4953	.OKE.RINT.ONT.IS
1730	54CC	4C49	5354	C445	4C45	5445	C155	544F	T.LIST.ELETE.UTO
1740	C34C	4541	52C3	4C4F	4144	C353	4156	45CE	.LEAR.LOAD.SAVE.
1750	4557	D441	4228	D44F	C64E	D553	494E	47D6	EW.AB(.O.N.SING.
1760	4152	5054	52D5	5352	C552	4CC5	5252	D354	ARPTR.SR.RL.RR.T
1770	5249	4E47	24C9	4E53	5452	D04F	494E	54D4	RING\$.NSTR.OINT.
1780	494D	4524	CD45	4DC9	4E4B	4559	24D4	4845	IME\$.EM.NKEY\$.HE
1790	4ECE	4F54	D354	4550	ABAD	AAAF	DBC1	4E44	N.OT.TEP.....ND
17A0	CF52	BEBD	BCD3	474E	C94E	54C1	4253	C652	.R....GN.NT.BS.R
17B0	45C9	4E50	D04F	53D3	5152	D24E	44CC	4F47	E.NP.OS.QR.ND.OG
17C0	C558	50C3	4F53	D349	4ED4	414E	C154	4ED0	.XP.OS.IN.AN.TN.
17D0	4545	4BC3	5649	C356	53C3	5644	C54F	46CC	EEK.VI.VS.VD.OF.
17E0	4F43	CC4F	46CD	4B49	24CD	4B53	24CD	4B44	OC.OF.KI\$.KS\$.KD
17F0	24C3	494E	54C3	534E	47C3	4442	4CC6	4958	\$.INT.SNG.DBL.IX
1800	CC45	4ED3	5452	24D6	414C	C153	43C3	4852	.EN.TR\$.AL.SC.HR
1810	24CC	4546	5424	D249	4748	5424	CD49	4424	\$.EFT\$.IGHT\$.ID\$
1820	A780	AE1D	A11C	3801	3501	C901	7341	D3018.5....A..
1830	B622	051F	9A21	0826	EF21	211F	C21E	A31E	."...!.&!!.....
1840	3920	911D	B11E	DE1E	071F	A91D	071F	F71D	9.....
1850	F81D	001E	031E	061E	091E	A341	602E	F41FA....
1860	AF1F	FB2A	6C1F	7941	7C41	7F41	8241	8541	...*...A.A.A.A.A
1870	8841	8B41	8E41	9141	9741	9A41	A041	B202	.A.A.A.A.A.A.A..
1880	6720	5B41	B12C	6F20	E41D	2E2B	292B	C62B	...A.,.....+)+.+
1890	0820	7A1E	1F2C	F52B	491B	7979	7C7C	7F50,+I.....P
18A0	46DB	0A00	007F	0AF4	0AB1	0A77	0C70	0CA1	F.....
18B0	0DE5	0D78	0A16	0713	0747	08A2	080C	0AD2G.....
18C0	0BC7	0BF2	0B90	2439	0A4E	4653	4E52	474F\$9.NFSNRGO
18D0	4446	434F	564F	4D55	4C42	5344	442F	3049	DFCOVOMULBSDD/OI
18E0	4454	4D4F	534C	5353	5443	4E4E	5252	5755	DTMOSLSSTCNRWU
18F0	454D	4F46	444C	33D6	006F	7CDE	0067	78DE	EMOFDL3.....

WIE LAUTEN DIE CALL-ADRESSEN IN DEZIMAL UND HEXADEZIMAL ?

Die Umwandlung der in Abb. 4 angegebenen zweiteiligen Adressen in dezimale bzw. hexadezimale Werte ist sicherlich nicht schwer, aber trotzdem recht mühsam. Wir haben aber zum Glück einen Rechner und ein Programm, mit denen wir uns die Arbeit erleichtern können.

Das Zahlenumwandlungsprogramm aus Kapitel 7 macht das Umwandeln zu einer richtigen Freude. Dieses Programm könnte auf Wunsch leicht geändert werden, um die LSB- und MSB-Werte aus dem Programm nach Abb. 3 zu übernehmen und automatisch die Hexadezimal- und Dezimaladressen der Funktionsaufrufe zu erzeugen. Der Einfachheit halber benutzen wir aber zuerst das Programm so wie in Kapitel 7 gezeigt, um die CALL-Adresse der ersten Funktion zu berechnen, die in Abb. 4 abgebildet ist. Es handelt sich um 'END'.

Geben Sie das Umwandlungsprogramm ein und vergessen Sie nicht, es anschließend auf Kassette oder Diskette für eine spätere Anwendung zu speichern. Wir benutzen die Funktion 'ZWEITEILIG DEZIMAL NACH DEZIMAL' und geben dazu 'SP' ein. Aus Abb. 4 entnehmen wir für 'END' die Werte LSB=174 und MSB=29. Auf die Frage 'DEZIMAL ?' geben wir '174' ein und nach einem Moment erscheint wieder 'DEZIMAL ?'. Nun geben wir '29' ein. Fast schneller als der Schall erscheint die Antwort 'HEXADEZIMAL 1DAE' und eine Weile danach folgt 'DEZIMAL 7598'. Aha, das funktioniert also !

Der Ablauf des Umwandlungsprogramms ist folgender:

1. Umwandlung von '174' nach hexadezimal und Speicherung des Wertes.
2. Ebenso Umwandlung und Speicherung von '29'.
3. Vertauschen der beiden Hexadezimalwerte in die Reihenfolge MSB-LSB.
4. Anzeige der Hexadezimalzahl.
5. Umwandlung dieser Zahl in eine Dezimalzahl.
6. Anzeige der Dezimalzahl.

Abb. 6 zeigt einen Ausdruck der BASIC CALL-Adressen gem. Abb. 4 in dezimaler und hexadezimaler Form. Die Adressen über 12288 dienen natürlich zum Aufruf von Disketten-Funktionen.

So weit, so gut ! Machen wir nun einen Sprung in's Kapitel 2 (JP Kapitel 2) und versuchen, einige Erfahrungen über die Anwendung arithmetischer Routinen zu sammeln. Zu Anfang werden wir allerdings nur (wie in den ersten Volksschulklassen) einfache arithmetische Operationen ausführen, also Addition, Subtraktion, Multiplikation und Division ausführen. Aber... das machen wir mit einigen wenigen CALL-Befehlen anstelle mit seitenlangen selbstgeschriebenen Assemblerprogrammen.

BASIC-FUNKTIONEN MIT CALL ADRESSEN

FUNKTION	HEX	DEZ.	FUNKTION	HEX	DEZ.
END	1DAE	7599	FOR	1CA1	7329
RESET	0138	312	SET	0135	309
CLS	01C9	457	CMD	4173	16755
RANDOM	01D3	467	NEXT	22B6	8886
DATA	1F05	7941	INPUT	219A	8602
DIM	2608	9736	READ	21EF	8687
LET	1F21	7959	COTO	1EC2	7874
RUN	1EA3	7843	IF	2039	8249
RESTORE	1D91	7569	COSUB	1EB1	7857
RETURN	1EDE	7902	REM	1F07	7943
STOP	1DA9	7593	ELSE	1F07	7943
TRON	1DF7	7671	TROFF	1DF8	7672
DEFSTR	1E00	7680	DEFINT	1E03	7683
DEFSGN	1E06	7686	DEFDEL	1E09	7689
LINE	41A3	16803	EDIT	2E60	11872
ERROR	1FF4	8180	RESUME	1FAF	8111
OUT	2AFB	11003	ON	1F6C	8044
OPEN	4179	16761	FIELD	417C	16764
GET	417F	16767	PUT	4182	16770
CLOSE	4185	16773	LOAD	4188	16776
MERGE	418B	16779	NAME	418E	16782
KILL	4191	16782	LSET	4197	16791
RSET	419A	16794	SAVE	41A0	16800
SYSTEM	02B2	690	LPRINT	2067	8295
DEF	415B	16731	POKE	2CB1	11441
PRINT	206F	8303	CONT	1DE4	7652
LIST	2B2E	11054	LLIST	2B29	11049
DELETE	2BC6	11206	AUTO	2008	8200
CLEAR	1E7A	7802	CLOAD	2C1F	11295
CSAVE	2BF5	11253	NEW	1B49	6985
INT	0B37	2871	ABS	0977	2423
FRE	27D4	10196	INP	2AEF	10991
POS	27F5	10229	SQR	13E7	5095
RND	14C9	5321	LOG	0809	2057
EXP	1439	5177	COS	1541	5441
SIN	1547	5447	TAN	15A8	5544
ATN	15BD	5565	PEEK	2CAA	11424

K A P I T E L 2

EINFUEHRUNG

Nachdem man sich erst einmal von dem Schock erholt hat, die Grundlagen der Assemblerprogrammierung zu erlernen, ist es unsinnig, immer wieder neu 'das Rad zu erfinden' indem man Dutzende von Programmzeilen oder -seiten schreibt, nur, um simple arithmetische Berechnungen durchzuführen, wenn diese Routinen bereits alle im Level-II ROM enthalten sind und mit einem einfachen CALL aufgerufen werden können.

Programme in Assemblersprache mit dem resultierenden Maschinencode arbeiten bis zu 300 mal schneller als entsprechende BASIC-Programme und benötigen außerdem nur etwa ein Zehntel des Speicherplatzes. Somit kann man sagen, daß Assemblerprogrammierung das 'non plus Ultra' für den ernsthaften Programmierer darstellt, der über die nun einmal vorhandenen Grenzen der höheren Programmiersprachen hinauskommen will. Vor Erscheinen dieses Buches mußten interessierte Programmierer mit allen möglichen, umständlichen Verfahren versuchen, hinter die Geheimnisse des Level-II Codes zu kommen, um all diese hilfreichen Routinen ausnutzen zu können.

Nun, Assembler-Programmierer aufgepaßt, das Geheimnis des Level-II ROM ist keines mehr. Die Kryptographierer wissen: jeder Code hat einen Schlüssel; bei einigen dauert es nur etwas länger, den richtigen zu finden. Aus irgendeinem perversen (also evtl. finanziellen) Grund waren weder Radio Shack noch Microsoft bereit, den über 200.000 Benutzern des TRS-80 zu sagen, wie man die unzähligen ROM-Routinen benutzt (oder sind sie am chinesischen Geheimniskrämer-Syndrom erkrankt?). Diese Behauptung wird am besten durch das Buch "TRS-80 Assembly Language Programming" bewiesen. Aus welchem Grund auch immer wird der angehende Assembler-Programmierer in TBUG eingeführt und dann werden endlos lange Programme zum Abfragen der Tastatur, Ausgabe auf dem Bildschirm, Datenverschieben und Rechnen gezeigt, deren Funktionen mit einem einzigen ROM-CALL ausgeführt werden könnten. Lassen Sie uns annehmen, das Radio Shack selbst nicht die leiseste Ahnung über die Leistungsfähigkeit des Level-II ROM hatte. Im anderen Fall müßte man ihnen eine große Nachlässigkeit vorwerfen und das gegenüber mehr als 200.000 Mitgliedern der TRS-80 Gemeinde. Wenn Sie Level-II-BASIC beherrschen, werden Sie viel Freude an dem hier beschriebenen Weg zur Assemblerprogrammierung haben. Dadurch, daß Sie Level-II-BASIC gemeistert haben, ist bewiesen, daß Sie das Zeug und die Ausdauer besitzen, nach einigen Wochen fleißigen Arbeitens ein guter Assembler-Programmierer zu werden. Die angeblichen "Experten" im Bereich der Assembler Programmierung haben sich mit einer Aura von Geheimnissen umgeben, die völlig unbegründet ist. Es scheint, sie wollen damit ihre Selbstachtung steigern. Ein Philosoph nannte das einmal "Stiergebrüll". Lassen Sie uns nun mit einigen einfachen arithmetischen Programmen sehen, wie leicht das Erstellen von Assemblerprogrammen ist, wenn man fast ausschließlich Level-II ROM Routinen verwendet.

GRUNDLAGEN DER LEVEL-II ROM ARITHMETIK:

Die Arithmetik-Routinen sind praktisch identisch mit den Routinen, die Sie schreiben müßten, wären sie nicht im ROM verfügbar. Das trifft für alle Arten von Routinen zu, egal ob wir Addition, Subtraktion, Multiplikation, Division oder irgendeine trigonometrische, Exponential- oder Logarithmusfunktion mit ganzen, einfach oder doppelt genauen Zahlen betrachten. Allgemein trifft das für alle Funktionen im Level-II ROM zu. Lassen Sie uns mit einer Einführung in die Addition, Subtraktion, Multiplikation und Division beginnen.

ROM (read-only-memory) meint genau das, was es sagt: ein Speicher, aus dem nur gelesen werden kann. Demnach kann hier nichts gespeichert werden. Daher verwendet das Level-II ROM die RAM-Speicheradressen von 14302 bis 17129 (dezimal) für alle internen Speicheraufgaben. Die Tastatur von Adresse 14336 bis 15360 ist kein wirklicher RAM-Speicher, sondern eine einfache Matrixanordnung der Tastenkontakte, allerdings erscheint dieser Bereich dem Rechner wie ein Speicher. Der Bildschirmspeicher belegt den Adressbereich von 15360 bis 16383. Mit Ausnahme der Speicherplätze 14302 bis 14336 werden alle internen Speicherfunktionen im Bereich von 16384 bis 17129 durchgeführt (ausgenommen Disketten-Betriebsprogramme). Drei Speicherbereiche im RAM sind von ganz besonderer Wichtigkeit, wenn man mit arithmetischen Unterprogrammen arbeitet. Es sind dies der AKKUMulator, der CDBL-Speicher (abgekürzt: CS) und der Speicher für den Nummerntyp (NT). Arithmetische Werte, die im RAM gespeichert werden, haben folgende Formate: Integer (ganzzahlig) = erst LSB, dann MSB in Zweierkomplementdarstellung. Einfach- und doppelt genaue Gleitkommazahlen in normalisierter Exponentialdarstellung mit 129 addiert zum Exponent-Byte und dem MSB der Mantisse als Vorzeichenbit. Jetzt wissen die meisten von Ihnen, liebe Leser, sicher genausoviel wie vorher, aber machen Sie sich keine Sorgen, Sie brauchen sich um diese Formate nicht zu kümmern, da unser Level-II ROM alle Umwandlungen unter Berücksichtigung der Formate richtig ausführt, wenn wir die Umwandlungsroutinen richtig anwenden. Der AKKUMulator belegt die Speicherplätze von 411DH bis 4124H (8 Bytes) und der CDBL-Speicher liegt im Bereich von 4127H bis 412EH, also ebenfalls 8 Bytes. Womit wir uns allerdings befassen sollten ist mit dem Nummerntyp-Speicher. Wenn wir versuchen, arithmetische Operationen zwischen Zahlen mit unterschiedlichem Nummerntyp auszuführen, werden wir unser blaues Wunder erleben (z.B.: Addition einer Ganzzahl zu einer doppelt genauen Gleitkommazahl). Aber verlieren Sie nicht den Mut, wir können die CINT, CSNG und CDBL-Funktionen mit einem einzigen CALL aufrufen, um Zahlen, die wir verwenden wollen, passend zu machen.

CINT = CALL 0A7FH CSNG = CALL 0AB1H CDBL = CALL 0ADBH

Die drei Programme in diesem Kapitel sehen diese Routinen bei jeder Funktion vor, so daß es schwerfallen wird, einen Fehler zu machen, wenn Sie nur ganz einfache Regeln beherzigen. Der Nummerntyp-Speicher (NT) ist ein Ein-Byte

Speicherplatz im RAM mit der Adresse 40AF H. NT enthält 2 für Ganzzahl-Variable, 3 für Text-Variable, 4 für einfach- und 8 für doppelt genaue Gleitkommazahlen. Um den jeweiligen Wert in den ASCII-Code umzuwandeln, damit er auf dem Bildschirm angezeigt werden kann, addieren Sie einfach 30 H zum Inhalt des Speicherplatzes 40AFH und geben ihn auf dem Bildschirm mit folgendem Programm aus:

```
LD      A,(40AFH)      ; NT-SPEICHER
ADD     A,30H          ; UMWANDLUNG IN ASCII
CALL    032AH         ; BILDSCHIRMAUSGABE
```

ARITHMETIK MIT GANZEN ZAHLEN (+ - x /):

In Abb. 7 ist der Quellen- und Objektcode für ein Demonstrationsprogramm angegeben, das Ihnen die Addition, Subtraktion, Multiplikation und Division von Ganzzahlen erlaubt, wobei durchweg ROM-Routinen verwendet werden. Sie können wetten, das Programm ist sehr schnell; sobald Sie ENTER drücken, haben Sie das Ergebnis. Bedenken Sie bitte, Ihr TRS-80 mit einer Taktfrequenz von ca. 1,774 MHz ist jetzt wirklich keine lahme Ente mehr. Anstelle im gemütlichen BASIC unterhalten Sie sich jetzt direkt mit dem Z-80 Mikroprozessor in seiner eigenen Sprache. Ohne den Interpreter arbeitet der Rechner mit einer erheblich höheren Geschwindigkeit. Die einzige Aufgabe, des angegebenen Programms ist es, Ihre erste Eingabe im DE-Register abzulegen, ihre zweite Eingabe in das HL-Register zu laden und dann die von Ihnen eingegebene Rechenoperation durch CALL aufzurufen. Das Programm ist recht einfach und geradlinig programmiert mit Ausnahme der Zeilen 330 und 400. Die Befehle PUSH HL und POP DE haben die Aufgabe, den zuerst eingegebenen Wert (er steht im HL-Register) auf den Stack zu 'retten'. Danach wird der Inhalt von HL durch den zweiten eingegebenen Wert überschrieben. Durch POP DE wird der zuerst eingegebene, alte Wert von HL einfach vom Stack in das DE-Register übertragen. Den Stack selber kümmert es wenig, wohin seine Inhalte gehen, er ist einfach ein LIFO-Speicher (last in first out - zuletzt 'rein' zuerst 'raus'). Er liegt - sofern Sie keine Änderungen mit dem Befehl LD SP vornehmen - ab Adresse 4288H im RAM.

Merksenswert ist folgende Regel:

Bei Ganzzahl-Arithmetik wird der 'E'rstes Operand ins 'D'E Register geladen, der 'Z'weite Operand ins 'H'L Register. Danach legen Sie die gewünschte Operation durch einen der folgenden Befehle fest:

```
Addition:          CALL OBD2H
Subtraktion:        CALL OBD7H
Multiplikation:     CALL OBF2H
Division:           CALL 2490H
```

Das Ergebnis der Operation wird jeweils in den AKKUMULATOR geladen. Um das Ergebnis anzuzeigen, schreiben Sie einfach:

```
CALL    OFBDH          ; UMWANDLG. AKKUM NACH ASCII
CALL    28A7H          ; ANZEIGE ASCII AUF BILDSCHIRM
```

Man sollte sich bei einfachen Additionen auch an den Befehl ADD HL, (Registerpaar) erinnern, mit dem man Werte bis zu 65535 verarbeiten kann. Sie sehen, mit Ganzzahlen ist tatsächlich leicht zu arbeiten.

ARITHMETIK MIT EINFACH GENAUEN GLEITKOMMAZAHLEN (+ - x /):

Hier läuft es ähnlich ab, wie bei ganzen Zahlen, allerdings will das ROM nun den Ersten Operanden in den Registern BC und DE und den zweiten im AKKUMULATOR sehen. Die benötigte Operation wird durch folgende CALLs aufgerufen:

Addition:	CALL 0716H
Subtraktion:	CALL 0713H
Multiplikation:	CALL 0847H
Division:	CALL 08A2H

Zur Speicherung von Zwischenergebnissen benutzen wir wieder den Stack, wie in den Zeilen 340 und 350 (PUSH) und in den Zeilen 420 und 430 (POP) gezeigt. Das entsprechende Programm finden Sie unter der Abbildung 8.

ARITHMETIK MIT DOPPELT GENAUEN GLEITKOMMAZAHLEN (+ - x /)

Ein besonders großer Unterschied zu den vorangehenden Funktionen besteht nicht, außer daß das ROM nun den Ersten Operanden im AKKUMULATOR und den zweiten im CDBL-Speicher erwartet. Die gewünschte Funktion wird wie folgt aufgerufen:

Addition:	CALL 077CH
Subtraktion:	CALL 0C70H
Multiplikation:	CALL 0DA1H
Division:	CALL 0DE5H

Die Abbildung 9 zeigt das entsprechende Demonstrationsprogramm.

ZUSAMMENFASSUNG:

Jedes der Quellprogramme kann durch den Anwender in ca. 5 Minuten (ohne Kommentare) mit dem ausgezeichneten Editor/Assembler von Radio Shack eingegeben werden. Was bisher ein monatelanges Studium und seitenlange Programme zu Arithmetik mit doppelter Genauigkeit erforderte, kann nun der einigermaßen geübte TRS-80 Anwender in wenigen Stunden verstehen und anwenden.

```

100 ; DEMONSTRATIONSPROGRAMM FUER
105 ; GANZZAHLEN-ARITHMETIK MIT
110 ; LEVEL-II ROM ROUTINEN + - * /
120 ;
130
140 BEGIN ORG 7D00H ;=32000 DEZIMAL
150 LD A,4FH ;=ASCII 'O'
160 CALL 032AH ;ANZEIGEN AUF VIDEO
170 LD A,3FH ;=ASCII '?'
180 CALL 032AH ;ANZEIGEN AUF VIDEO
190 LD A,20H ;=ASCII LEERZEICHEN
200 CALL 032AH ;ANZEIGEN AUF VIDEO
210 CALL 0049H ;TASTATUREINGABE
220 CALL 032AH ;ANZEIGE D. FUNKTION
230 LD (FUNCT),A ;BENOETIGTE FUNKTION
240 ;RETEN
250 LD A,0DH ;=ASCII ZEILENSCHALTC.
260 CALL 032AH ;VIDEO
270 LD A,45H ;=ASCII 'E'
280 CALL 032AH ;VIDEO
290 CALL 1BB3H ;TASTATUREINGABE MIT ECHO AUF
300 ;VIDEO
310 RST 10H ;ABTASTEN UND C-FLAG SETZEN
320 CALL 0E6CH ;ASCII NACH ACCUM UND
330 ;MINDEST-NT BESTIMMEN
340 CALL 0A7FH ;UMWANDLUNG IN INTEGER
350 PUSH HL ;RETEN AUF STAPEL
360 LD A,5AH ;=ASCII 'Z'
370 CALL 032AH ;VIDEO
380 CALL 1BB3H ;TASTATUREING. M. ECHO
390 RST 10H ;ABTASTEN U. C-FLAG SETZEN
400 CALL 0E6CH ;ASCII NACH ACCUM UND
410 ;MINDEST-NT BESTIMMEN
420 CALL 0A7FH ;UMWANDELN IN INTEGER
430 POP DE ;ALTEN WERT V. HL NACH DE
440 LD A,(FUNCT) ;RUECKHOLEN GEWAEHLTE FUNKTION
450 CP 2BH ;='+'?
460 JP Z,ADD ;JA, DANN ADDIEREN
470 CP 2DH ;='-'?
480 JR Z,SUB ;JA, DANN SUBTRAHIEREN
490 CP 2AH ;='*'?
500 JR Z,MULT ;JA, DANN MULTIPLIZIEREN
510 CP 2FH ;='/'?
520 JR Z,DIVIDE ;JA, DANN DIVIDIEREN
530 VIDEO: LD A,3DH ;=ASCII '='
540 CALL 032AH ;VIDEO
550 LD A,20H ;=ASCII LEERZEICHEN
560 CALL 032AH ;VIDEO
570 CALL 0FBDH ;UMWANDLUNG ACCUM NACH STRING
580 CALL 28A7H ;ANZEIGE VON STRING AUF VIDEO
590 LD A,0DH ;=ASCII ZEILENSCHALTUNG
600 CALL 032AH ;VIDEO
610 JR BEGIN ;NOCHMAL DAS GANZE
620 ADD CALL 0BD2H ;ADDIEREN DE + HL
630 JR VIDEO ;ERGEBNIS ANZEIGEN
640 SUB CALL 0BC7H ;SUBTRAHIEREN DE - HL
650 JR VIDEO ;ERGEBNIS ANZEIGEN
660 MULT CALL 0BF2H ;MULTIPLIZ. DE * HL
670 JR VIDEO ;ERGEBNIS ANZEIGEN
680 DIVIDE CALL 2490H ;DIVIDIEREN DE / HL
690 JR VIDEO ;ERGEBNIS ANZEIGEN
700 FUNCT DEFB 1
710 END BEGIN

```

```

00001 ;
00002 ;
00003 ; DEMONSTRATIONSPROGRAMM FUER EINFACHGENAUE ARITHMETIK MI
00004 ; LEVEL II ROM - ROUTINEN + - * / VON W4UCH
00005 ;
7D00 W4UCH EQU 7D00H ; = 32000 DEZIMAL
7D00 00007 ORG W4UCH ; PROGRAMMSTART
7D00 3E4F 00008 BEGIN LD A,4FH ; ASCII "O" (OPERATION)
7D02 CD2A03 00009 CALL 032AH ; ANZEIGEN AUF VIDEO
7D05 3E3F 00010 LD A,3FH ; ASCII "?"
7D07 CD2A03 00011 CALL 032AH ; ANZEIGEN
7D0A 3E20 00012 LD A,20H ; ASCII LEERZEICHEN
7D0C CD2A03 00013 CALL 032AH ; ANZEIGEN AUF VIDEO
7D0F CD4903 00014 CALL 0049H ; TASTATUR-EINGABE (+ - * /)
7D12 CD2A03 00015 CALL 032AH ; FUNKTION ANZEIGEN
7D15 32807D 00016 LD (FUNCT),A ; BENOETIGTE FUNKTION RETTEN
7D18 3E0D 00017 LD A,0DH ; ASCII ZEILENSCHALTUNG
7D1A CD2A03 00018 CALL 032AH ; AUSGEBEN AUF VIDEO
7D1D 3E45 00019 LD A,45H ; "E" - ERSTE ZAHL
7D1F CD2A03 00020 CALL 032AH ; ANZEIGEN AUF VIDEO
7D22 CDB31B 00021 CALL 1BB3H ; EINGABE V. TASTATUR M. ANZEIGE
7D25 D7 00022 RST 10H ; ABTASTEN UND CARRY SETZEN
7D26 CD6C0E 00023 CALL 0E6CH ; ASCII-->ACCUM NT AUF
00024 ; MINDESTWERT SETZEN
7D29 CDB10A 00025 CALL 0AB1H ; UMWANDLUNG EINFACHGENAUIGK.
7D2C CDBF09 00026 CALL 09BEH ; BC, DE VON ACCUM LADEN
7D2F C5 00027 PUSH BC ; SPEICHERN IM STACK
7D30 D5 00028 PUSH DE ; SPEICHERN IM STACK
7D31 3E5A 00029 LD A,5AH ; "Z" - ZWEITE ZAHL
7D33 CD2A03 00030 CALL 032AH ; ANZEIGEN AUF VIDEO
7D36 CDB31B 00031 CALL 1BB3H ; EINGABE VON TASTATUR M. ANZEIGE
7D39 D7 00032 RST 10H ; ABTASTEN UND CARRY SETZEN
7D3A CD6C0E 00033 CALL 0E6CH ; ASCII-->ACCUM NT AUF
00034 ; MINDESTWERT SETZEN
7D3D CDB10A 00035 CALL 0AB1H ; UMWANDLUNG EINFACHGENAUIGK.
7D40 D1 00036 POP DE ; REGISTER VOM STACK ZU-
7D41 C1 00037 POP BC ; RUECKHOLEN
7D42 3A807D 00038 LD A,(FUNCT) ; GEWUENSCHTE FUNKTION RUECKHOLEN
7D45 FE2B 00039 CP 2BH ; GLEICH "+" ?
7D47 2823 00040 JR Z,ADD ; JA, DANN ADDIEREN
7D49 FE2D 00041 CP 2DH ; GLEICH "-" ?
7D4B 2824 00042 JR Z,SUB ; JA, DANN SUBTRAHIEREN
7D4D FE2A 00043 CP 2AH ; GLEICH "*" ?
7D4F 2825 00044 JR Z,MULT ; JA, DANN MULTIPLIZIEREN
7D51 FE2F 00045 CP 2FH ; GLEICH "/" ?
7D52 2826 00046 JR Z,DIVIDE ; JA, DANN DIVIDIEREN
7D53 3E3D 00047 VIDEO LD A,3DH ; ASCII "="
7D57 CD2A03 00048 CALL 032AH ; VIDEO
7D5A 3E14 00049 LD A,20 ; ASCII LEERZEICHEN
7D5C CD2A03 00050 CALL 032AH ; VIDEO
7D5F CDED0F 00051 CALL 0FBDH ; UMWANDLG. ACCUM-->STRING
7D62 CDA728 00052 CALL 28A7H ; ANZEIGEN AUF VIDEO
7D65 3E0D 00053 LD A,0DH ; ZEILENSCHALTUNG
7D67 CD2A03 00054 CALL 032AH ; VIDEO
7D6A 1894 00055 JR BEGIN ; DAS GANZE NOCHMAL
7D6C CD1607 00056 ADD CALL 0716H ; ADDIERT BC, DE ZU ACCUM
7D6F 18E4 00057 JR VIDEO ; ERGEBNIS ANZEIGEN
7D71 CD1307 00058 SUB CALL 0713H ; SUBTRAKTION
7D74 18DF 00059 JR VIDEO ; ERGEBNIS ANZEIGEN
7D76 CD4708 00060 MULT CALL 0847H ; MULTIPLIKATION
7D79 18DA 00061 JR VIDEO ; ERGEBNIS ANZEIGEN
7D7B CDA208 00062 DIVIDE CALL 08A2H ; DIVISION
7D7E 18D5 00063 JR VIDEO ; ERGEBNIS ANZEIGEN
7D80 00 00064 FUNCT DEFB ; SPEICHER FUER GEWAELHTE
00065 ; FUNKTION
00066 ;
00067 ;
00068 ;
00069 ;
00070 ;
7D80 00071 END W4UCH
00000 TOTAL ERRORS

```

- ABB. 8 -

```

ADD 7D6C 00056 00040
BEGIN 7D00 00008 00055
DIVIDE 7D7B 00062 00046
FUNCT 7D80 00064* 00016 00038
MULT 7D76 00060 00044
SUB 7D71 00058 00042
VIDEO 7D53 00047 00057 00059 00061 00063
W4UCH 7D00 00006 00007 00071

```

```

00001
00002
00003 DEMONSTRATIONSROCR. FUER DOPPELTGENAUE ARITHMETIK MIT
00004 ,LEVEL II ROM - ROUTINEN + - * / VON W4UCH
00005
00006 W4UCH EQU 7D00H ;= 32000 DEZIMAL
00007 ORG W4UCH ;PROGRAMMSTART
00008 BEGIN LD A,4FH ;ASCII "O" (OPERATION)
00009 CALL 032AH ;ANZEIGEN AUF VIDEO
00010 LD A,3FH ;ASCII "?"
00011 CALL 032AH ;ANZEIGEN
00012 LD A,20H ;ASCII LEERZEICHEN
00013 CALL 032AH ;ANZEIGEN AUF VIDEO
00014 CALL 0049H ;TASTATUR-EINGABE (+ - * /)
00015 CALL 032AH ;FUNKTION ANZEIGEN
00016 LD (FUNCT),A ;BENOETIGTE FUNKTION RETTEN
00017 LD A,0DH ;ASCII ZEILENSCHALTUNG
00018 CALL 032AH ;AUSGEBEN AUF VIDEO
00019 LD A,45H ;"E" - ERSTE ZAHL
00020 CALL 032AH ;ANZEIGEN AUF VIDEO
00021 CALL 1BB3H ;EINGABE V. TASTATUR M. ANZEIGE
00022 RST 10H ;ABTASTEN UND CARRY SETZEN
00023 CALL 0E65H ;ASCII-->ACCUM UMWANDLG. IN
00024 ;DOPPELTE GENAUIGKEIT
00025 LD DE,411DH ;DATEN AUS RAM-ACCUM NACH
00026 LD HL,TACCUM ;ZWISCHENSPEICHER IM RAM
00027 LD B,9 ;ANZAHL DER BYTES
00028 CALL 09D7H ;UEBERTRAGEN
00029 LD A,5AH ;"Z" - ZWEITE ZAHL
00030 CALL 032AH ;ANZEIGEN AUF VIDEO
00031 CALL 1BB3H ;EINGABE VON TASTATUR M. ANZEIGE
00032 RST 10H ;ABTASTEN UND CARRY SETZEN
00033 CALL 0E65H ;ASCII-->ACCUM DOPP. GENAUIGK.
00034 CALL 09FCH ;ACCUM NACH CDBL UEBERTRAGEN
00035 LD DE,TACCUM ;WERT AUS ZWISCHENSPEICH. NACH
00036 LD HL,411DH ;ACCUM ZURUECKHOLEN
00037 LD B,8 ;ANZAHL DER BYTES
00038 CALL 09D7H ;VERSCHIEBE-ROUTINE
00039 LD A,(FUNCT) ;GEWUENSCHTE FUNKTION RUECKHOLEN
00040 CP 2BH ;GLEICH "+" ?
00041 JR Z,ADD ;JA, DANN ADDIEREN
00042 CP 2DH ;GLEICH "-" ?
00043 JR Z,SUB ;JA, DANN SUBTRAHIEREN
00044 CP 2AH ;GLEICH "*" ?
00045 JR Z,MULT ;JA, DANN MULTIPLIZIEREN
00046 CP 2FH ;GLEICH "/" ?
00047 JR Z,DIVIDE ;JA, DANN DIVIDIEREN
00048 VIDEO LD A,3DH ;ASCII "="
00049 CALL 032AH ;VIDEO
00050 LD A,20 ;ASCII LEERZEICHEN
00051 CALL 032AH ;VIDEO
00052 CALL 0FBDH ;UMWANDLG. ACCUM-->STRING
00053 CALL 28A7H ;ANZEIGEN AUF VIDEO
00054 LD A,0DH ;ZEILENSCHALTUNG
00055 CALL 032AH ;VIDEO
00056 JR BEGIN ;DAS GANZE NOCHMAL
00057 ADD CALL 0C77H ;ADDIERT ACCUM UND CDBL
00058 JR VIDEO ;ERGESNIS ANZEIGEN
00059 SUB CALL 0C70H ;SUBTRAKTION
00060 JR VIDEO ;ERGESNIS ANZEIGEN
00061 MULT CALL 0DA1H ;MULTIPLIKATION
00062 JR VIDEO ;ERGESNIS ANZEIGEN
00063 DIVIDE CALL 0DE5H ;DIVISION
00064 JR VIDEO ;ERGESNIS ANZEIGEN
00065 FUNCT DEFB ;SPEICHER FUER GEWAELHTE FUNKT.
00066 TACCUM DEFS 8 ;ACCUM ZWISCHENSPEICHER
00067
00068
00069 - ABB. 9 -
00070
00071
00072
7D00 END W4UCH
00000 TOTAL ERRORS

```

```

ADD 7D78 00057 00041
BEGIN 7D00 00003 00056
DIVIDE 7D37 00063 00047
FUNCT 7D8C 00065 00016 00039
MULT 7D82 00061 00045
SUB 7D7D 00059 00043
TACCUM 7D8D 00066 00026 00035
VIDEO 7D61 00043 00058 00060 00062 00064
W4UCH 7D00 00006 00007 00072

```

K A P I T E L 3

DIE VERWENDUNG VON LEVEL-II ROM ROUTINEN BEI FORTGESCHRITTENER ASSEMBLER-PROGRAMMIERUNG

- TRIGONOMETRISCHE-, LOGARITHMUS-, EXPONENT- UND SONSTIGE FUNKTIONEN -

Es folgt ein interessantes Programm für den fortgeschrittenen Assembler-Programmierer. Es erlaubt Ihnen, die vielzähligen Routinen zu testen, die in dem exzellenten Level-II ROM vorhanden sind.

Wenn man beginnt, Assembler-Programmierung zu lernen, sollte man sich natürlich auch mit dem Wie, Warum und Wozu befassen und grundlegende arithmetische oder trigonometrische Funktionen selbst schreiben, aber wenn diese Techniken als Teil des Lernprozesses erst einmal beherrscht werden, ist es sicherlich ineffektiv, zeitaufwendig und ziemlich unsinnig, wieder einmal 'das Rad zu erfinden' indem man die Routinen nochmals schreibt, die ohnehin im Level-II ROM enthalten sind.

In Tabelle 1 sind die Funktionen mit ihren Adressen angegeben, die über das beschriebene 'Miniprogramm' von nur 144 Bytes angesprochen und getestet werden können. Sie können das Programm mit dem Editor/Assembler in 5 Minuten eingeben.

Die Abb. 13 und 14 zeigen einen Ausdruck des Quell- bzw. Objektprogramms. Wie man leicht sehen kann, werden zum größten Teil Routinen das Level-II ROM verwendet. Hätte man diese Routinen nicht verwendet und eigene Routinen schreiben müssen, so würde das Programm ca. zehnmal soviel Speicherplatz belegen und etwa 550 anstelle von 55 Zeilen Assemblerprogramm benötigen.

PROGRAMM-ABLAUF:

Die Kommentare, die im Quellprogramm enthalten sind, erläutern die Aufgabe jeder Programmzeile, wir müssen sie also hier nicht wiederholen, da sie sich von selbst erläutern. Das Programm arbeitet ebensogut mit Level-II, TRSDOS und NEWDOS. Das Programm wird wie folgt benutzt:

1. Laden Sie das Programm mit SYSTEM oder mit LOAD unter DOS. Sie können ihm einen beliebigen Namen geben, wir haben es 'DISCOV' genannt, denn für unsere Entdeckungsreise wollen wir es ja verwenden. Nachdem das Programm geladen ist, starten Sie es mit SYSTEM und /32000 (Bei DOS müssen Sie erst BASIC laden).
2. Der Buchstabe 'N' erscheint auf dem Bildschirm. Das Programm wartet auf die Eingabe eines Wertes, der verarbeitet werden soll. Jede Zahl bis maximal 16 Stellen, je nach auszuführender Funktion ist zulässig. Lassen Sie uns mit einem einfachen Beispiel anfangen, indem wir die Zahl 10000, eine nette runde Sache, eingeben (mit ENTER abschließen).

3. Die Zahlen '2' und '10000' erscheinen auf der nächsten Zeile der Anzeige. Die Ziffer '2' drückt den Zahlentyp aus, vom Level-II ROM brilliant berechnet. Da wir zur Zeit nur mit Zahlen arbeiten, interessieren uns nur die Typen 2 = Ganzzahl, 4 = einfach- und 8 = doppeltgenau. In Tabelle 1 sind die Operationen mit den entsprechenden Zahlentypen aufgelistet, für die die Operation jeweils erlaubt ist. So ist es z.B. gegen die Regel, die Wurzel aus einer ganzen Zahl zu ziehen. Die Zahl müssen wir zuerst in eine einfach genaue Gleitkommazahl umwandeln.
4. In der folgenden Zeile der Videoanzeige sehen Sie 'C ?'. Das Programm fragt Sie nach dem Typ der Zahlenumwandlung, den Sie wünschen. Lassen Sie uns den Wert 2737, also die Adresse der Umwandlungsroutine CSNG eingeben, um unsere Zahl vom Ganzzahntyp in eine einfachgenaue Fließkommazahl umzuwandeln. Dann drücken Sie 'ENTER'. Die nächste Zeile zeigt nun '4' '1000' an. Daran sehen wir, daß wir nun eine einfachgenaue Zahl haben, mit der wir weiterarbeiten können. Nun berechnen wir die Quadratwurzel aus dieser Zahl, indem wir '5095' = SQRT eingeben; wir beenden die Eingabe mit 'ENTER' und wie der Blitz erscheint das Ergebnis '100' in der nächsten Zeile. Sie haben soeben ein Wunder der modernen Computerwissenschaft erlebt; und Sie werden sicher zugeben, das war einfacher als hätten Sie ein eigenständiges Programm zum Berechnen der Quadratwurzel schreiben müssen. Lassen Sie uns noch einen Versuch machen: geben Sie nochmals '5095 ENTER' ein und wieder zeigt die nächste Zeile die Quadratwurzel, in diesem Fall die '10'.
5. Aber das ist nur der Anfang! Wenn Sie eine neue Zahl eingeben wollen, mit der Sie das Programm ausprobieren wollen, geben Sie einfach '32000 ENTER' ein. Das bringt uns auf hervorragende Weise wieder dahin wo wir begonnen haben; wir erhalten erneut die Anzeige 'N ?'. Ist denn 32000 eine Unterroutine? Natürlich, Sie haben sie ja selbst geschrieben. Unser Assemblerprogramm unterscheidet nicht zwischen RAM und ROM.
6. Wir könnten nun beliebig weitermachen, indem wir z.B. den natürlichen Logarithmus einer Zahl berechnen und danach die Zahl wiedergewinnen, indem wir die EXP-Funktion aufrufen oder den Tangens einer Zahl berechnen und danach den Arcus Tangens und wieder den Tangens und nochmal den Arcus Tangens und.....so weiter. Sie können das Programm jederzeit verlassen, indem Sie einfach die Funktion '6681 ENTER' aufrufen. Damit gelangen Sie wieder an den Anfang des BASIC-Interpreters, der sich mit 'READY' meldet. Wollen Sie wieder das Maschinenprogramm aufrufen, geben Sie 'SYSTEM ENTER' und dann /32000 ein.

Dieser Abschnitt behandelt nur einige der vielen Routinen, die im Level II ROM enthalten sind und die in Tabelle 1 aufgeführt sind.

Programmieren in Assemblersprache ist sozusagen das Non Plus Ultra der ernsthaften Programmierung und stellt den Mount Everest unseres Hobbys dar. Die Tatsache, daß solche Programme bis zu 300 mal schneller arbeiten als ein vergleichbares BASIC-Programm und dabei nur etwa ein Zehntel des Speicherplatzes erfordern ist ein nicht zu verachtendes Ergebnis.

Eine der wichtigsten Erfahrungen, die sie machen können ist es, mit Ihrem Computer anstelle in BASIC, FORTRAN, PASCAL u.s.w. in seiner eigenen Sprache mit ihm zu reden und Sie werden es bei ein wenig Geduld und Ausdauer sicherlich lernen.

Anhang:

Das Demonstrationsprogramm aus Abb. 13 ist leicht in der Lage, viel mehr Funktionen auszuführen, als in der kurzen Tabelle 1 angegeben. Sie sollten sich eine Notiz machen, daß Sie später nochmals an diese Stelle zurückgehen, wenn sie sich mit den Routinen zur Datenverschiebung und Zahlenumwandlung vertraut gemacht haben.

Die meisten arithmetischen Routinen (+ - / x), die mit ganzen Zahlen, einfacher oder doppelter Genauigkeit arbeiten, können verwendet werden, indem man einfach eine Zahl in dem CDBL-Speicher überträgt. Durch CALL 2556 = 09FCH werden Daten vom ACCUM in den CDBL-Speicher geladen. Die zweite Zahl wird dann durch Aufruf von '32000' in den ACCUM gebracht.

Achten Sie in jedem Fall genau darauf, daß sie nur Zahlen gleichen Typs miteinander verrechnen. Wenden Sie gegebenenfalls vorher die entsprechenden Umwandlungsroutinen an. Nun ist es eigentlich ganz einfach, die unterschiedlichsten Berechnungen mit dem Demonstrationsprogramm auszuführen, wenn Sie zudem noch die Datenverschiebe-Routinen aus dem nächsten Kapitel beherrschen.

Obwohl sie für eine bestimmte Anwendung sicherlich nie mehr ein Programm schreiben werden, das dem hier zu Demonstrationszwecken gezeigten entspricht, können Sie es doch recht gut verwenden, um die in Kapitel 4 gezeigten Routinen auszutesten. Außerdem stellt das Programm eine gewisse Herausforderung dar.....und Herausforderungen bereiten Freude, wenn Sie sie gewinnen.

Anm.: Nummerntypen 2 = ganzzahlig, 4 = einfach genau
8 = doppelt genau

FUNKTION	NUMMERNTYP	DEZIMAL	HEXADEZIMAL
ABS	2 - 4 - 8	2423	0977
ATN	4 - 8	5565	15BD
BASIC	(zurück n. L II)	6681	1A19
BASIC	(zurück n. DOS-BASIC)	112	0075
BREAK	.(RST - Adresse)	16396	400C
CDBL	2 - 4	2779	0ADB
CINT	4 - 8	2687	0A7F
CLS	2 - 4 - 8	457	01C9
COS	4 - 8	5441	1541
CSNG	2 - 8	2737	0AB1
EXP	4 - 8	5177	1439
FIX	2 - 4	2854	0B26
INT	2	2871	0B37
LOG	4 - 8	2057	0809
MEM.SIZE	-	181	00R5
RANDOM	2 - 4 - 8	467	01D3
RND (siehe Einschränkung)	2 - 4 - 8	5321	14C9
SGN	2	2442	098A
SIN	4 - 8	5447	1547
SQU	4 - 8	5095	13E7
TAN	4 - 8	5544	15A8
VORZ.UMKEHR	2	3153	0C51
VORZ.UMKEHR	4 - 8	2434	0982
zurück nach Demoprogramm		32000	7D00

- Tabelle 1 -

	00100				
	00110				
	00120				
7D00	00130	W4UCH	EQU	7D00H	;=32000 DEZIMAL
7D00	00140		ORG	W4UCH	;PROGRAMMSTART HIER
7D00	00150		LD	A,5AH	;ASCII "Z"
7D02	00160		CALL	032AH	;VIDEO
7D05	00170		CALL	1BB3H	;TASTATUR-EINGABE
7D08	00180		RST	10H	;TEXTABTASTEN, CARRY SETZEN
7D09	00190		CALL	0E6CH	;ASCII-->ACCUM, MIN. NT
7D0C	00200	RETURN	EX	AF,AF'	;REGISTER TAUSCHEN, UM WERTE
7D0D	00210		EXX		;ZU SICHERN
7D0E	00220		LD	DE,411DH	;DATEN VON ACCUM IN ZWISCHEN-
7D11	00230		LD	HL,STORE	;SPEICHER UEBERTRAGEN
7D14	00240		LD	B,8	;ANZAHL DER BYTES
7D16	00250		CALL	09D7H	;VERSCHIEBEROUTINE
7D19	00260		LD	A,(40AFH)	;ZAHLENTYP-SPEICHER
7D1C	00270		LD	(FLAG),A	;IN ZWISCHENSPEICHER
7D1F	00280		ADD	A,48	;UMWANDLUNG NACH ASCII
7D21	00290		CALL	032AH	;AUSGEBEN AUF VIDEO
7D24	00300		LD	A,20H	;ASCII LEERZEICHEN
7D26	00310		CALL	032AH	;VIDEO
7D29	00320		CALL	0FBDH	;UMWANDLG. ACCUM-->ASCII
7D2C	00330		CALL	28A7H	;UND ANZEIGEN
7D2F	00340		LD	A,0DH	;ZEILENSCHALTUNG
7D31	00350		CALL	032AH	;AUSGEBEN
7D34	00360		LD	A,43	;ASCII "C"
7D36	00370		CALL	032AH	;VIDEO
7D39	00380		CALL	1BB3H	;TASTATUREINGABE
7D3C	00390		RST	10H	;TEXT ABTASTEN, CARRY SETZEN
7D3D	00400		CALL	0E6CH	;UMWANDLG. ASCII-->ACCUM
7D40	00410		CALL	0A7FH	;UMWANDLUNG NACH GANZZAHL
7D43	00420		LD	(CONV),HL	;SPEICHERN D. UMWANDLUNGSADRESSE
7D46	00430		LD	DE,CDBL	;DATEN VOM CDBL-ZWISCHENSPEICHER
7D49	00440		LD	HL,4127H	;ZURUECKHOLEN
7D4C	00450		LD	B,8	;ANZAHL DER BYTES
7D4E	00460		CALL	09D7H	;VERSCHIEBERROUTINE
7D51	00470		LD	DE,STORE	;DATEN VOM ACCUM-ZWISCHENSPEICHER
7D54	00480		LD	HL,411DH	;ZURUECKHOLEN
7D57	00490		LD	B,8	;ANZAHL DER BYTES
7D59	00500		CALL	09D7H	;VERSCHIEBERROUTINE
7D5C	00510		LD	A,(FLAG)	;NT VON ZWISCHENSPEICHER
7D5F	00520		LD	(40AFH),A	;HOLEN
7D62	00530		LD	HL,RETURN	;RUECKSPRUNGADRESSE WIRD IN
7D65	00540		PUSH	HL	;STACK GELADEN
7D68	00550		LD	HL,(CONV)	;UMWANDLUNGSADRESSE IN DEN
7D69	00560		PUSH	HL	;STAPEL LADEN
7D6A	00570		EX	AF,AF'	;REGISTER RUECKTAUSCHEN
7D6B	00580		EXX		
7D6C	00590		RET		
0001	00600	FLAG	DEFS	1	;NT ZWISCHENSPEICHER
0002	00610	CONV	DEFS	2	;ZW.SPEICHER UMWANDLG.S. ADRESSE
0008	00620	CDBL	DEFS	8	;ZWISCHENSPEICHER FUER CDBL
0008	00630	STORE	DEFS	8	;ZWISCHENSPEICHER FUER ACCUM
7D00	00640		END	W4UCH	;FUNKAMATEUR-RUFZEICHEN
00000	TOTAL	ERRORS			

CDBL	7D70	00620	00430	
CONV	7D6E	00610	00420	00550
FLAG	7D6D	00600	00270	00510
RETURN	7D0C	00200	00530	
STORE	7D78	00630	00230	00470
W4UCH	7D00	00130	00140	00640

K A P I T E L 4

HILFSROUTINEN IM LEVEL-II ROM

EINFUEHRUNG:

In den Kapiteln 2 und 3 wurden bereits eine Anzahl von Hilfsroutinen verwendet, deren Bedeutung mit Ausnahme der wenigen Worte in den Kommentaren der Quellprogramme nicht erläutert wurde. Will man die Level-II Routinen für mathematische Funktionen verwenden, so sollte man auch einigermaßen gute Kenntnisse der Hilfsroutinen des Level-II ROM haben, die ja nur darauf warten, verwendet zu werden. Um nur einige aufzuzählen: Tastatur-Eingabe, Datenverschiebung, Datenvergleich, Datenumwandlung, Bildschirm- und Druckerausgabe. Dieses Kapitel wird die CALL-Adressen und eine kurze Beschreibung der am meisten gebrauchten Routinen bringen und damit wird für Sie das Schreiben von Assemblerprogrammen in 'Kurzschrift' Wirklichkeit.

TASTATUR-ROUTINEN:

Jedem fortgeschrittenen Assembler Programmierer ist bekannt, daß die Tastatur nichts weiter ist als eine Matrix mit Tastenschaltern, die dem Level-II ROM wie ein RAM erscheint. Die Tastatur belegt acht Speicherplätze, deren Adressen unten in Dezimalwerten angegeben sind. Dabei bedeuten PO = Pfeil nach oben, PU = Pfeil nach unten, PL = Pfeil nach links und PR = Pfeil nach rechts.

PEEK (14337) =	⓪	A	B	C	D	E	F	G	
PEEK (14338) =	H	I	J	K	L	M	N	O	
PEEK (14340) =	P	Q	R	S	T	U	V	W	
PEEK (14344) =	X	Y	Z						
PEEK (14352) =	0	1	2	3	4	5	6	7	
PEEK (14368) =	8	9	:	;	,	-	.	/	
PEEK (14400) =	ENT	CLR	BRK	PO	PU	PL	PR	LEERT.	

WERT	=	1	2	4	8	16	32	64	128

Die Tastenschaltermatrix gibt den unter 'Wert' ausgedruckten Wert aus, wenn eine einzelne Taste gedrückt ist und der Speicherplatz adressiert wird, dem diese Taste zugeordnet ist. Werden gleichzeitig mehrere Tasten einer Reihe gedrückt, so sind die Werte der einzelnen Tasten zu addieren; z.B. "JKL" = 4 + 8 + 16 = 28 unter Adresse 14338 dezimal. Mit diesem Wissen ist es leicht zu erkennen, wie einfach bei NEWDOS+ die Funktion zum Ausdruck des Bildschirminhaltes auf dem Drucker aufgerufen wird, wenn 'J', 'K' und 'L' gleichzeitig gedrückt werden. Nun was halten Sie davon, ein kurzes Maschinenprogramm zu schreiben, das die entsprechende Funktion auch ohne DOS ausführt? Nur Geduld, im Band 2 ist ein solches Programm abgedruckt.

Für eingefleischte Programmierer hier ein kurzer BASIC-Einzeiler:

```
10 X=PEEK(14338) : PRINT @478,X : GOTO 10
```

Geben Sie das Programm ein und drücken Sie dann beliebige Tastenkombinationen in der PEEK(14338)-Reihe.

Nun folgen die drei wichtigsten Level-II Hilfsroutinen für die Tastatur. Alle CALL-Adressen verstehen sich in Hexadezimaldarstellung.

CALL 002B: Dies ist die grundlegende Tastatur-Routine, sie liest die gesamte Tastatur aus und liefert ein ASCII-Zeichen im Akkumulator, wenn eine Taste gedrückt wurde. Um die Routine zu wiederholen, bis eine Taste gedrückt wird, muß eine JR Z Schleife geschrieben werden:

```
TAST      CALL      002BH
          CP        00
          JR        Z,TAST
```

Immer dann wenn eine Taste gedrückt wird, ergibt der Vergleich CP 00 nicht den Wert Null, so daß die Programmschleife verlassen wird und das Programm nach dem JR-Befehl fortgesetzt wird. Diese Routine wurde bisher von allen Programmierern verwendet, die es nicht besser wussten. Wir werden sie in Zukunft kaum noch benötigen.

CALL 0049: Diese Routine ähnelt sehr stark der INKEY\$ - Funktion von BASIC. Sie prüft automatisch die Tastatur solange ab, bis eine Taste gedrückt wird und überträgt das entsprechende Codezeichen in den Akkumulator. Sie brauchen also keine eigene Warteschleife zu programmieren. Das ist schon einmal ein großer Fortschritt gegenüber dem 002BH CALL.

CALL 1BB3: Das ist die 'Super-Tastatur-Routine', die Sie wohl am häufigsten verwenden werden. Zunächst zeigt sie auf dem Bildschirm ein '?' an. Danach wird der über die Tastatur eingegebene Text als Textvariable formatiert und mit 'OOH' abgeschlossen (Maximal-Länge 240 bytes). Die Textvariable wird im RAM beginnend mit Adresse 40A7H abgespeichert. Normalerweise folgt nach dem CALL 1BB3 ein RST 10-Befehl, der das "Carry"-Flag setzt. In den Programmen der Kapitel 2 und 3 wird diese Routine verwendet. Die Routine überträgt außerdem den eingegebenen Text automatisch auf das unter Speicherplatz 409CH angegebene Gerät (-1 = Kassette, 0 = Bildschirm, +1 = Zeilendrucker). Beim Einschalten wird der Wert 0 gewählt. Diese Routine ist sicherlich eine der nützlichsten, zeitsparensten und am häufigsten gebrauchten Unterprogramme und Sie werden sie daher ebenfalls oft anwenden. Ein einziger CALL 1BB3H ersetzt Dutzende von Programmzeilen, wenn nicht -seiten.

DATENVERSCHIEBUNG:

Im Level-II ROM existieren Routinen zum Verschieben bzw. Uebertragen der Daten von und zu fast jedem beliebigen Speicherbereich. Diese Routinen können sehr oft mit nur einem CALL-Befehl aufgerufen werden. Sie sparen eine enorme Menge Zeit und Programmzeilen ein, daher ist es die Mühe wert, sich mit diesen Routinen vertraut zu machen.

Eine der nützlichsten Routinen ist die in der folgenden Tabelle in der ersten Zeile aufgeführte (CALL 09A4H). Dieses Unterprogramm überträgt automatisch entweder eine Ganzzahl oder eine einfach genaue Zahl aus dem ACCUM, der im Speicherbereich 411DH bis 4124H liegt, in den Stapelspeicher (Stack). Durch die Befehlsfolge POP BC und POP DE wird die Zahl wiedergewonnen. Bei einer ganzen Zahl ist ihr Wert im DE-Registerpaar enthalten, bei einer einfach genauen Zahl steht ihr Wert in den Registerpaaren BC und DE.

Obwohl alle Datenverschieberoutinen nützlich sind, verdienen die Zeilen J. und K. der folgenden Tabelle besondere Beachtung. Jede ist in der Lage, einen Speicherbereich von maximal 255 Byte von (DE) nach (HL) zu übertragen (Registerpaar DE enthält die Anfangsadresse der Ursprungsblocks und HL enthält die Anfangsadresse des Zielblocks). Das Programm J. entnimmt die Anzahl der zu übertragenden Zeichen dem A-Register (Akkumulator); beim Programm K. muß die Anzahl im B-Register stehen.

- TABELLE DER DATENVERSCHIEBE-ROUTINEN -

Nr.	von	nach	CALL	NT
A.	ACCUM	STACK	09A4H/2468	2,4
B.	(HL)+	ACCUM	09B1H/2481	4
C.	BCDE	ACCUM	09B4H/2484	4
D.	ACCUM	BCDE	09BFH/2495	4
E.	(HL)+	BCDE	09C2H/2498	4
F.	ACCUM	(HL)+	09CBH/2507	4
G.	(DE)+	(HL)+	09CEH/2510	4
H.	(HL)+	(DE)+	09D2H/2514	2,4,8
I.	(DE)+	(HL)+	09D3H/2515	2,4,8
J.	(DE)+	(HL)+	09D6H/2518	A REG
K.	(DE)+	(HL)+	09D7H/2519	B REG
L.	"CS"	ACCUM	09F4H/2548	2,4,8
M.	ACCUM	"CS"	09FCH/2556	2,4,8
N.	HL	ACCUM	0A9AH/2714	2
O.	DE	HL	EX DE,HL	2
P.	HL	DE	EX DE,HL	2
Q.	BC	STACK	PUSH BC	2,4
R.	DE	STACK	PUSH DE	2,4
S.	HL	STACK	PUSH HL	2
T.	STACK	HL	POP HL	2
U.	STACK	DE	POP DE	2,4
V.	STACK	BC	POP BC	2,4

Nr. gibt den Nummerntyp an, 2 = Ganzzahl, 4 = einfach genaue und 8 = doppelt genaue Zahl).

Anm.: die Zeilen "O" bis "V" stellen gewöhnliche Z-80 Befehle dar, aber sie sind hier mit aufgeführt, um Sie daran zu erinnern, daß man bei der Bearbeitung von ganzen oder einfach genauen Zahlen mit diesen Befehlen oft am einfachsten Daten verschieben und zwischenspeichern kann. Beachten Sie die Programme in Kapitel 2, in denen PUSH und POP-Befehle verwendet wurden, um solche Zahlen zu "retten", während die Register für andere Zwecke verwendet werden mußten.

DATENVERGLEICHE:

Einige der am meisten verwendeten Funktionen bei der Programmierung von Computern sind Vergleiche wie "gleich", 'kleiner als' und 'größer als'. Im Level-II ROM sind diese Funktionen ebenfalls enthalten und können mit einem CALL-Befehl aufgerufen werden. Das Ergebnis der Vergleiche wird im "A"-Register abgelegt und ist "0", wenn es "gleich" lautet, "+1", bei "größer als" und "255" bei "kleiner als".

- TABELLE DER VERGLEICHSPROUTINEN -

NR.	OPERAND 1	MINUS	OPERAND 2	CALL	NT
A.	ACCUM	-	BCDE	OA0CH	4
B.	HL	-	DE	OA39H	2
C.	ACCUM	-	"CS"	OA4FH	8
D.	"CS"	-	ACCUM	OA78H	8
E.	ACCUM	-	Vorzeichenbest.	O994H	2,4,8

Anm.: Zeile E. entspricht dem BASIC-Befehl SGN aber liefert das Ergebnis im A-Register, A = 0, wenn ACCUM = 0; A = +1, wenn ACCUM größer als Null und A = 255 (OFFH), wenn ACCUM kleiner als Null.

DATENUMWANDLUNGEN:

Datenumwandlungen sind einfach durchzuführen und werden bei fast allen arithmetischen Operationen benötigt, da die Nummerntypen verschiedener Zahlen gleich sein müssen, bevor mit ihnen untereinander gerechnet wird und sie müssen auf die jeweiligen Routinen abgestimmt sein. Die am meisten benötigten Routinen sind:

CALL	von NT	nach NT	Register/Speicher	Name
OA7FH	beliebig	ganzz.	ACCUM	CINT
OAB1H	beliebig	einfach	ACCUM	CSNG
OACCH	ganzz.	einfach	ACCUM	--
OACFH	ganzz.	einfach	von HL nach ACCUM	--
OADBH	beliebig	doppelt	ACCUM	--
OE65H	String	doppelt	ACCUM	--
OE6CH	String	mind. erforderl.	ACCUM	--

ZUSAMMENFASSUNG DER ARITHMETISCHEN ROUTINEN:

	ganzzahlig	einfach	doppelt
Addition	OBD2H/3026 DE + HL	O716H/1814 BCDE + ACCUM	OC77H/3191 ACCUM+"CS"
Subtraktion	OBC7H/3015 DE - HL	O713H/1811 BCDE - ACCUM	OC70H/3184 ACCUM-"CS"
Multiplikation	OBF2H/3058 DE x HL	O847H/2119 BCDE x ACCUM	O8A2H/2210 ACCUM x CS
Division	2490H/10560 DE / HL	O8A2H/2210 BCDE / ACCUM	ODE5H/3557 ACCUM / CS

Anm.: NT (Nummerntyp) in Speicherplatz 40AFH muß der aufgerufenen Routine entsprechen. NT: 2 = ganzzahlig, 3 = String, 4 = einfach und 8 = doppelt genau.

ACCUM liegt im Bereich von 411DH bis 4124H
"CS" liegt im Bereich von 4127H bis 412EH.

VIDEO ANZEIGE

Fast alle Routinen, die die Anzeige auf dem Bildschirm betreffen sind den 'Computer-Experten' nun schon seit zwei Jahren bekannt, den 'Fundamental-Aufruf' CALL O32AH eingeschlossen, der den Inhalt des A-Registers auf den Bildschirm bringt, wenn der Speicherplatz 409CH den Wert Null enthält, der auch beim Stromeinschalten dort gespeichert wird. Besonders WICHTIG in diesem Zusammenhang ist, daß CALL O32AH außerdem auch noch unter Adresse 40A6H die aktuelle neue Position des Cursors abspeichert. Das ist sehr nützlich und beseitigt die Notwendigkeit, daß sie ein entsprechendes Programmstück selber schreiben müssen.

Eine der wichtigsten Routinen, um eine Tasteneingabe auf dem Bildschirm wiederzugeben ist der CALL 1BB3H, den wir schon vorher besprochen haben. Diese Routine zeigt einen Text auf dem Bildschirm an, der bei (HL) beginnt (HL enthält die Anfangsadresse des Textes) und der mit 00 endet, sofern Speicheradresse 409CH den Wert 00 enthält. Der Video-Kontrollblock, wie er im Level-II-Handbuch beschrieben ist, zusammen mit der Cursor-Position in 40A6H erlaubt ihnen, die Bildschirmanzeige beliebig zu verändern und zu benutzen.

Eine faszinierende Übung, ein Assemblerprogramm zu schreiben ist ein Programm, das den Bildschirm in zwei unabhängige Anzeigefelder unterteilt. Diese Funktion könnte z.B. für ein Programm zum Empfangen und Senden von Morse- oder Fernschreibzeichen verwendet werden. Die obere Hälfte des Bildschirms kann dabei zur Anzeige des empfangenen und die untere Hälfte zur Anzeige des gesendeten Textes dienen. Den 'Senderteil' könnte man dabei als FIFO-Speicher auslegen um mit beliebiger Geschwindigkeit schreiben, aber mit konstanter Geschwindigkeit zu senden. Wenn Sie eine

Groß/Kleinschrift-Modifikation eingebaut haben, ist es leicht möglich, den bereits gesendeten Text in Großschrift und den noch zu sendenden Text in Kleinschrift anzuzeigen. Eine andere Alternative ist, einen zweiten Cursor z.B. CHR (170) einzublenden, der anzeigt, welcher Text gerade gesendet wird. Beide Bildschirmhälften arbeiten völlig unabhängig voneinander, d.h. von ihrem eigenen Video-Speicher im RAM mit eigener Roll-Funktion (scrolling) u.s.w. Anm.: ein solches Programm wird im Band 2 beschrieben.

ZEILENDRUCKER.

Genauso wie bei der Bildschirmanzeige gibt es auch bei den ROM-Routinen zur Druckersteuerung einige neue Ueber-raschungen. Wieder ist es der Wert, der unter 409CH im RAM gespeichert ist, der bestimmt, wohin der durch die Tasta-tur-Routinen CALL 032AH und 1BB3H erzeugte Text übertragen wird. Wenn 409CH den Wert +1 enthält, geht der Text auf den Drucker. Wie im Level-II-Handbuch gezeigt, ist die Adresse des Druckers 37E8H und der Drucker-Kontrollblock liegt im Bereich von 4025H bis 402CH. Der Speicherplatz 37E8H ent-hält den Wert 63 dezimal (= ASCII '?') wenn Ihr Drucker in der Lage ist, ein weiteres Zeichen zu empfangen. Voraus-setzung ist, daß der Drucker ein entsprechendes 'Hand-shake'-Signal erzeugt. Einige billige Drucker, die ge-braucht erhältlich sind, haben diese Funktion nicht und somit ist von deren Verwendung abzuraten. Sobald in Adresse 37E8H das '?' vom Drucker erscheint, ist es möglich, das nächste zu druckende Zeichen unter 37E8H über das A-Regis-ter mittels einem LD-Befehl abzuspichern. Eine Ausnahme zu dieser Regel wird im Kapitel 7 beim 'Null mit Schräg-strich'-Programm gezeigt, bei dem nach dem Druck einer Null und erfolgtem Rückschritt eine zusätzliche 20 ms-Verzö-gerung einsetzt, um den Druckmechanismus zur Ruhe kommen zu lassen.

Das 'Null mit Schrägstrich'-Programm wurde vorher noch nirgendwo veröffentlicht und bringt einige interessante Punkte im Zusammenhang mit der Programmierung von Druckern. Es unterbricht den Ausdruck des nächsten Zeichens, indem die Adressen bei 4026H und 4027H geändert werden, um ein kleines Programm einzufügen. Zum Glück ist das nächste zu druckende Zeichen immer im C-Register enthalten, unabhängig davon, ob der Rechner ohne DOS, mit TRS-DOS, NEWDOS+ oder mit der 'JKL'-Funktion von NEWDOS arbeitet. Somit können durch einfaches Testen des C-Registers beliebige Funktionen durch den Programmierer eingefügt werden.

Um nur einige Möglichkeiten anzureißen, lassen Sie uns annehmen, daß Sie eine IBM-Kugelkopfmachine als Drucker verwenden, eine hervorragende Maschine, wenn Sie einen sauberen Ausdruck für die Textverarbeitung benötigen. Alerdings haben alle verwendbaren Kugelköpfe einen Nachteil, sie drucken die Null ohne Schrägstrich, so daß eine Verwechslung mit dem großen O leicht möglich ist. In einem Geschäftsbrief mag eine Null mit Schrägstrich zwar etwas merkwürdig aussehen, in einem Programmausdruck ist es jedoch eine große Hilfe, wenn man zwischen Null und 'O' unterscheiden kann. Das Programm aus Kapitel 7 druckt nun

alle Nullen mit Schrägstrich und alle Zeilen mit 64 Zeichen. Sie können es z.B. auch so modifizieren, daß alle '<' als KL und alle '>' als GR gedruckt werden. Nachdem es ein Lehrprogramm ist, bei dem es auf Uebersichtlichkeit ankommt, kann es noch beträchtlich komprimiert werden (Versuchen Sie es doch einmal, das Programm um 1/3 zu kürzen).

K A P I T E L 5

ZUSAMMENSTELLUNG DER LEVEL-II ROM ROUTINEN IN
ALPHABETISCHER FOLGE

Diese Zusammenstellung enthält auch die Adressen der CALLs, die eine Verbindung zu DOS und DISK-BASIC herstellen, da diese über Adressen im ROM aufgerufen werden. Zusätzlich ist die Uebergabe-Adresse für die BREAK-Funktion enthalten, so daß diese bei 400CH abgefangen werden kann bevor ein RST-Befehl ausgeführt wird. Somit kann die BREAK-Taste unwirksam gemacht oder für andere Zwecke ganz nach Ihrem Belieben verwendet werden.

Hier noch ein Extra-Bonus für Besitzer von Disketten unter der Ueberschrift "MASTER":

'F3GUM' ist ein Schutzwort (Password), mit dem Sie Zugriff auf ALLE geschützten Dateien unter TRS-DOS oder NEWDOS+ erhalten. Ein besonderes Dankeschön an Manny Garcia, der diesen Trick herausgefunden hat....zu jedem Schloß findet sich ein Schlüssel! "F3GUM" erlaubt Ihnen Dateien zu laden, löschen, kopieren oder was sonst auch immer nötig ist, unabhängig davon, ob eine Datei SIP-geschützt ist.

BASIC Funktion	CALL-Adresse hexadezimal	BASIC Funktion	CALL-Adresse hexadezimal
ABS	09 17	&H	4194
ASC	2A0F	ATN	15BD
AUTO	2008	BREAK	400C
CDBL	0ADB	CHR\$	2A1F
CINT	0A7F	CLEAR	1E7A
CLOAD	2C1F	CLOSE	4185
CLS	01C9	CMD	4173
CONT	1DE4	COS	1541
CSAVE	2BF5	CSNG	0AB1
CVD	415E	CVI	4152
CVS	4158	DATA	1F05
DEF	415B	DEFDBL	1E09
DEFINT	1E03	DEFSNG	1E06
DEFSTR	1E00	DELETE	2BC6
DIM	2608	EDIT	2E60
ELSE	1F07	END	1DAE
EOF	4161	ERL	24DD
ERR	24CF	ERROR	1FF4
EXP	1439	FIELD	417C
FIX	0B26	FOR	1CA1
FN	4155	FRE	27D4
GET	417F	GOSUB	1EB1
GOTO	1EC2	IF	2039
INKEY\$	019D	INP	2AEF
INPUT	219A	INSTR	419D
INT	0B37	KILL	4191
LEFT\$	2A61	LEN	2A03
LET	1F21	LINE	41A3
LIST	2B2E	LOAD	4188

LOC	4164	LOF	4167
LOG	0809	LLIST	2B29
LPRINT	2067	LSET	4197
MASTER	F3GUM	MEM	27C9
MERGE	418B	MID\$	2A9A
MKD\$	4170	MKI\$	416A
MKS\$	416D	NAME	418E
NEW	1B49	NEXT	22B6
NOT	25C4	ON	1F6C
OPEN	4179	OUT	2AFB
PEEK	2CAA	POINT	0133
POKE	2CB1	POS	27F5
PRINT	206F	PUT	4218
RANDOM	01D3	READ	21EF
REM	1F07	RESET	0138
RESTORE	1D91	RETURN	1EDE
RESUME	1FAF	RIGHT\$	2A91
RND	14C9	RSET	419A
RUN	1EA3	SAVE	41A0
SET	0135	SGN	09 4
SIN	1547	SQR	13E7
STOP	1DA9	STR\$	2836
STRING\$	2A2F	SYSTEM	02B2
TAN	15A8	TIME\$	4176
TROFF	1DF8	TRON	1DF7
USR	27FE	VARPTR	24EB
VAL	2AC5		

Anm.: wenn Sie viele Assmeblerprogramme schreiben, ist es empfehlenswert, diese Tabelle zu kopieren und die beiden Blätter zusammenzukleben. Sie erhalten dann eine übersichtliche Tabelle, die hier aus drucktechnischen Gründen nicht so darzustellen ist.

ELSE = 1F07 ist zweifelhaft, obwohl das ROM auf diese Adresse zeigt, aber das ROM zeigt für 'REM' auf dieselbe Adresse, was richtig erscheint. Evtl. ist nur das hier vorhandene Level-II-ROM fehlerhaft.

läuft es denn?

0000	F3AF	C374	06C3	0040	C300	40E1	E9C3	9F06@..@.....
0010	C303	40C5	0601	182E	C306	40C5	0602	1826	..@.....@.....&
0020	C309	40C5	0604	181E	C30C	4011	1540	18E3	..@.....@..@..
0030	C30F	4011	1D40	18E3	C312	4011	2540	18DB	..@..@.....@.*@..
0040	C3D9	05C9	0000	C3C2	03CD	2B00	B7C0	18F9+.....
0050	0D0D	1F1F	0101	5B1B	0A1A	0818	0919	2020
0060	0B78	B120	FBC9	3100	063A	EC37	3CFE	02D21...:7<...
0070	0000	C3CC	0611	8040	21F7	1801	2700	EDB0@!...!...
0080	21E5	4136	3A23	7023	362C	2322	A740	112D	!.A6:##.#6,##".@.-
0090	0106	1C21	5241	36C3	2373	2372	2310	F706	...!RA6.##.##.##...
00A0	1536	C923	2323	10F9	21E8	4270	31F8	41CD	.6.###...!.B.1.A.
00B0	8F1B	CDC9	0121	0501	CDA7	28CD	B31B	38F5!.....(...8.
00C0	D7B7	2012	214C	4323	7CB5	281B	7E47	2F77!LC#..(..G/.
00D0	BE70	28F3	1811	CD5A	1EB7	C297	19EB	2B3E	..(.....Z.....+>
00E0	8F46	77BE	7020	CE2B	1114	44DF	DA7A	1911	.F.....+..D.....
00F0	CEFF	22B1	4019	22A0	40CD	4D1B	2111	01CD	..".@."@.M.!...
0100	A728	C319	1A4D	454D	4F52	5920	5349	5A45	.(...MEMORY.SIZE
0110	0052	4144	494F	2053	4841	434B	204C	4556	.RADIO.SHACK.LEV
0120	454C	2049	4920	4241	5349	430D	001E	2CC3	EL.II.BASIC....,
0130	A219	D7AF	013E	8001	3E01	F5CF	28CD	1C2B>..>...(..+
0140	FE80	D24A	1EF5	CF2C	CD1C	2BFE	30D2	4A1E	...J....,..+.0.J.
0150	16FF	14D6	0330	FBC6	034F	F187	5F06	027A0...O.....
0160	1F57	7B1F	5F10	F879	8F3C	47AF	378F	10FD	.W.....<G.7...
0170	4F7A	F63C	571A	B7FA	7C01	3E80	47F1	B778	O..<W.....>.G...
0180	2810	12FA	8F01	792F	4F1A	A112	CF29	C9B1	(...../O.....)...
0190	18F9	A1C6	FF9F	E5CD	8D09	E118	EFD7	E53A:.....
01A0	9940	B720	06CD	5803	B728	11F5	AF32	9940	.@....X..(..2.@
01B0	3CCD	5728	F12A	D440	77C3	8428	2128	1922	<.W(*.@...(!("
01C0	2141	3E03	32AF	40E1	C93E	1CCD	3A03	3E1F	!A>.2.@..>...>.
01D0								2.@.!...!...
01E0	0B10	FE21	02FC	CD21	0206	0B10	FE21	00FC	...!...!...!...!
01F0	CD21	0206	5C10	FEC9	E521	00FB	181B	7ED6	.!.....!.....
0200	233E	0020	0DCD	012B	CF2C	7BA2	C602	D24A	#>.....+.,.....J
0210	1E3D	32E4	37E5	2104	FFCD	2102	E1C9	2100	.=2.7.!...!...!
0220	FF3A	3D40	A4B5	D3FF	323D	40C9	3A3F	3CEE	..=@....2=@.:?<.
0230	0A32	3F3C	C9C5	E506	08CD	4102	10FB	E1C1	.2?<.....A.....
0240	C9C5	F5DB	FF17	30FB	0641	10FE	CD1E	02060..A.....
0250	7610	FEDB	FF47	F1CB	1017	F5CD	1E02	F1C1G.....
0260	C9CD	6402	E5C5	D5F5	0E08	57CD	D901	7A07W.....
0270	5730	0BCD	D901	0D20	F2F1	D1C1	E1C9	0687	W0.....
0280	10FE	18F2	CDFE	0106	FFAF	CD64	0210	FB3E>
0290	A518	D1CD	FE01	E5AF	CD41	02FE	A520	F93EA.....>
02A0	2A32	3E3C	323F	3CE1	C9CD	1403	22DF	40CD	*2><2?<.....".@.
02B0	F801	CDE2	4131	8842	CDFE	203E	2ACD	2A03A1.B...>*.*
02C0	CDB3	1BDA	CC06	D7CA	9719	FE2F	284F	CD93/O..
02D0	02CD	3502	FE55	20F9	0606	7EB7	2809	CD35	..5..U.....(..5
02E0	02BE	20ED	2310	F3CD	2C02	CD35	02FE	7828#.....5... (
02F0	B8FE	3C20	F5CD	3502	47CD	1403	854F	CD35	..<...5.G....O.5

0300 0277 2381 4F10 F7CD 3502 B928 DA3E 4332 ..#.0...5..(>C2
0310 3E3C 18D6 CD35 026F CD35 0267 C9EB 2ADF ><...5...5....*.
0320 40EB D7C4 5A1E 208A EBE9 C54F CDC1 413A @...Z.....O..A:
0330 9C40 B779 C1FA 6402 2062 D5CD 3300 F5CD .@.....3...
0340 4803 32A6 40F1 D1C9 3A3D 40E6 083A 2040 H.2.@...:=@...:@
0350 2803 0FE6 1FE6 3FC9 CDC4 41D5 CD2B 00D1 (.....?..A..+..
0360 C9AF 3299 4032 A640 CDAF 41C5 2AA7 4006 ..2.@2.@..A.*.@.
0370 F0CD D905 F548 0600 0936 002A A740 F1C1H...6.*.@..
0380 2BD8 AFC9 CD58 03B7 C018 F9AF 329C 403A +....X.....2.@:
0390 9B40 B7C8 3E0D D5CD 9C03 D1C9 F5D5 C54F .@..>.....O
03A0 1E00 FE0C 2810 FE0A 2003 3E0D 4FFE 0D28(.....>.O..(
03B0 053A 9B40 3C5F 7B32 9B40 79CD 3B00 C1D1 ::.@<..2.@..;...
03C0 F1C9 E5DD E5D5 DDE1 D521 DD03 E54F 1AA0!...O..
03D0 B8C2 3340 FE02 DD6E 01DD 6602 E9D1 DDE1 ..3@.....
03E0 E1C1 C921 3640 0101 3816 000A 5FAE 73A3 ...!6@..8.....
03F0 2008 142C CB01 F2EB 03C9 5F7A 0707 0757 ...,,.....W
0400 0E01 79A3 2005 14CB 0118 F73A 8038 477A:8G.
0410 C640 FE60 3013 CB08 3031 C620 573A 4038 .@..0...01..W:@8
0420 E610 2828 7AD6 6018 22D6 7030 10C6 40FE ..((....."..0..@.
0430 3C38 02EE 10CB 0830 12EE 1018 0E07 CB08 <8.....0.....
0440 3001 3C21 5000 4F06 0009 7E57 01AC 0DCD 0.<!P.O....W....
0450 7AFE 01C0 EFC9 DD6E 03DD 6604 383A8:
0460 DD7E 05B7 2801 7779 FE20 DA06 05FE 8030(.....0
0470 35FE 4038 08D6 40FE 2038 02D6 20CD 4105 5.@8..@..8....A.
0480 7CE6 03F6 3C67 56DD 7E05 B728 05DD 7205<.V....(....
0490 365F DD75 03DD 7404 79C9 DD7E 05B7 C07E 6.....
04A0 C97D E6C0 6FC9 FEC0 38D3 D6C0 28D2 473E8...(>G
04B0 20CD 4105 10F9 18C8 7EDD 7705 C9AF 18F9 ..A.....
04C0 2100 3C3A 3D40 E6F7 323D 40D3 FFC9 2B3A 1.<:=@..2=@...+:
04D0 3D40 E608 2801 2B36 20C9 3A3D 40E6 08C4 =@..(+6...:=@...
04E0 E204 7DE6 3F2B C011 4000 19C9 237D E63F?+..@...#..?
04F0 C011 C0FF 19C9 3A3D 40F6 0832 3D40 D3FF:=@..2=@..
0500 237D E6FE 6FC9 1180 04D5 FE08 28C0 FE0A #.....(....
0510 D8FE 0E38 4F28 A1FE 0F28 A2FE 1728 D7FE ...80(...(....(..
0520 1828 B7FE 1928 C5FE 1A28 BCFE 1B28 C2FE .(....(....(....(..
0530 1C28 8DFE 1DCA A104 FE1E 2837 FE1F 283C .(.....(7..(<
0540 C977 233A 3D40 E608 2801 237C FE40 C011 ..#:=@..(#..@..
0550 C0FF 19E5 1100 3C21 403C C501 C003 EDB0<!@<.....
0560 C1EB 1819 7DE6 C06F E511 4000 197C FE40@....@
0570 28E2 D1E5 547D F63F 5F13 1804 E511 0040 (...T..?.....@
0580 3620 237C BA20 F97D BB20 F5E1 C979 B728 6.#.....(
0590 40FE 0B28 0AFE 0C20 1BAF DDB6 0328 15DD @..(.....(..
05A0 7E03 DD96 0447 CDD1 0520 FB3E 0A32 E837G.....>.2.7
05B0 10F4 1818 F5CD D105 20FB F132 E837 FE0D2.7..
05C0 C0DD 3404 DD7E 04DD BE03 79C0 DD36 0400 ..4.....6..
05D0 C93A E837 E6F0 FE30 C9E5 3E0E CD33 0048 ::.7...0..>..3.H
05E0 CD49 00FE 2030 25FE 0DCA 6206 FE1F 2829 .I...0%.....()
05F0 FE01 286D 11E0 05D5 FE08 2834 FE18 282B ..(.....(4..(+

0600 FE09 2842 FE19 2839 FE0A C0D1 7778 B728 ..(B..(9.....(
0610 CF7E 23CD 3300 0518 C7CD C901 41E1 E5C3 ..#.3.....A...
0620 E005 CD30 062B 7E23 FE0A C878 B920 F3C9 ...0.+.#.....
0630 78B9 C82B 7EFE 0A23 C82B 3E08 CD33 0004 ...+...#.+.>..3..
0640 C93E 17C3 3300 CD48 03E6 072F 3CC6 085F .>..3..H.../<...
0650 78B7 C83E 2077 23D5 CD33 00D1 051D C818 ...>..#.3.....
0660 EF37 F53E 0D77 CD33 003E 0FCD 3300 7990 .7.>...3.>..3...
0670 47F1 E1C9 D3FF 21D2 0611 0040 0136 00ED G.....!.....@.6..
0680 B03D 3D20 F106 2712 1310 FC3A 4038 E604 .==...'......:8..
0690 C275 0031 7D40 3AEC 373C FE02 DA75 003E ...1.@:.7<.....>
06A0 0132 E137 21EC 3711 EF37 3603 0100 00CD .2.7!..7..76.....
06B0 6000 CB46 20FC AF32 EE37 0100 423E 8C77 ...F...2.7..B>..
06C0 CB4E 28FC 1A02 0C20 F7C3 0042 0118 1AC3 .N(.....B.....
06D0 AE19 C396 1CC3 781D C390 1CC3 D925 C900&..
06E0 00C9 FBC9 0001 E303 004B 4907KI..
06F0 5804 003C 0044 4F06 8D05 4300 0050 52C3 X..<.DO...C..PR..
0700 0050 C700 003E 00C9 2180 13CD C209 1806 .P...>..!.....
0710 CDC2 09CD 8209 78B7 C83A 2441 B7CA B409:SA....
0720 9030 0C2F 3CEB CDA4 09EB CDB4 09C1 D1FE .0./<.....
0730 19D0 F5CD DF09 67F1 CDD7 07B4 2121 41F2!!A..
0740 5407 CDB7 07D2 9607 2334 CAB2 072E 01CD T.....#4.....
0750 EB07 1842 AF90 477E 9B5F 237E 9A57 237E ...B..G...#..W#..
0760 994F DCC3 0768 63AF 4779 B720 184A 5465 .O.....G.....JT..
0770 6F78 D608 FEE0 20F0 AF32 2441 C905 297A2\$A..).
0780 1757 798F 4FF2 7D07 785C 45B7 2808 2124 .W..O.....E.(.!\$
0790 4186 7730 E3C8 7821 2441 B7FC A807 4623 A..0...!\$A....F#
07A0 7EE6 80A9 4FC3 B409 1CC0 14C0 OCC0 0E80O.....
07B0 34C0 1E0A C3A2 197E 835F 237E 8A57 237E 4.....#..W#..
07C0 894F C921 2541 7E2F 77AF 6F90 477D 9B5F .O.!\$A./.....G...
07D0 7D9A 577D 994F C906 00D6 0838 0743 5A51 ..W..O.....8.CZQ
07E0 0E00 18F5 C609 6FAF 2DC8 791F 4F7A 1F57-...O..W
07F0 7B1F 5F78 1F47 18EF 0000 0081 03AA 5619G.....V..
0800 80F1 2276 8045 AA38 82CD 5509 B7EA 4A1E .."..E.8..U...J..
0810 2124 417E 0135 8011 F304 90F5 70D5 C5CD !\$A..5.....
0820 1607 C1D1 04CD A208 21F8 07CD 1007 21FC!.....!
0830 07CD 9A14 0180 8011 0000 CD16 07F1 CD89
0840 0F01 3180 1118 72CD 5509 C82E 00CD 1409 ..1.....U.....
0850 7932 4F41 EB22 5041 0100 0050 5821 6507 .2OA."PA...PX!..
0860 E521 6908 E5E5 2121 417E 23B7 2824 E52E .!.....!!A.#.(\$..
0870 081F 6779 300B E52A 5041 19EB E13A 4F410..*PA...:OA
0880 891F 4F7A 1F57 7B1F 5F78 1F47 2D7C 20E1 ..O..W.....G-...
0890 E1C9 435A 514F C9CD A409 21D8 0DCD B109 ..CZQO.....!.....
08A0 C1D1 CD55 09CA 9A19 2EFF CD14 0934 342B ...U.....44+
08B0 7E32 8940 2B7E 3285 402B 7E32 8140 41EB .2.@+.2.@+.2.@A..
08C0 AF4F 575F 328C 40E5 C57D CD80 40DE 003F .OW.2.@.....@..?
08D0 3007 328C 40F1 F137 D2C1 E179 3C3D 1FFA 0.2.@..7.....<=..
08E0 9707 177B 175F 7A17W..O)..G
08F0 3A8C 4017 328C 4079 B2B3 40CB E521 2441 :.@.2.@.....!\$A

0900	35E1	20C3	C3B2	073E	FF2E	AF21	2D41	4E23	5.....>...!-AN#
0910	AE47	2E00	78B7	281F	7D21	2441	AE80	471F	.G....(!!\$A..G.
0920	A878	F236	09C6	8077	CA90	08CD	DF09	772B	...6.....+.
0930	C9CD	5509	2FE1	B7E1	F278	07C3	B207	CDBF	..U./.....
0940	0978	B7C8	C602	DAB2	0747	CD16	0721	2441G...!\$A
0950	34C0	C3B2	073A	2441	B7C8	3A23	41FE	2F17	4.....:\$A.:#A./.
0960	9FC0	3CC9	0688	1100	0021	2441	4F70	0600	..<.....!\$AO...
0970	2336	8017	C362	07CD	9409	F0E7	FA5B	0CCA	#6.....
0980	F60A	2123	417E	EE80	77C9	CD94	096F	179F	..!#A.....
0990	67C3	9A0A	E7CA	F60A	F255	092A	2141	7CB5U.*!A..
09A0	C87C	18BB	EB2A	2141	E3E5	2A23	41E3	E5EB*!A.*#A...
09B0	C9CD	C209	EB22	2141	6069	2223	41EB	C921"!A."#A..!
09C0	2141	5E23	5623	4E23	4623	C911	2141	0604	!A.#V#N#F#..!A..
09D0	1805	EB3A	AF40	471A	7713	2305	20F9	C921:G...#.....!
09E0	2341	7E07	371F	773F	1F23	2377	7907	371F	#A..7..?..##...7.
09F0	4F1F	AEC9	2127	4111	D209	1806	2127	4111	O...!'A.....!'A.
0A00	D309	D511	2141	E7D3	111D	41C9	78B7	CA55!A....A....U
0A10	0921	5E09	E5CD	5509	79C8	2123	41AE	79F8	.!....U...!#A...
0A20	CD26	0A1F	A9C9	2378	BEC0	2B79	BEC0	2B7A	.&....#...+...+.
0A30	BEC0	2B7B	96C0	E1E1	C97A	AC7C	FA5F	09BA	..+.....
0A40	C260	097D	93C2	6009	C921	2741	CDD3	0911!'A....
0A50	2E41	1AB7	CA55	0921	5E09	E5CD	5509	1B1A	.A...U.!....U...
0A60	4FC8	2123	41AE	79F8	1323	0608	1A96	C223	O.!#A....#.....#
0A70	0A1B	2B05	20F6	C1C9	CD4F	0AC2	5E09	C9E7	..+.....O.....
0A80	2A21	41F8	CAF6	0AD4	B90A	21B2	07E5	3A24	*!A.....!....:\$
0A90	41FE	9030	0ECD	FB0A	EBD1	2221	413E	0232	A..0....."!A>.2
0AA0	AF40	C901	8090	1100	00CD	0C0A	C061	6A18	..@.....
0AB0	E8E7	E0FA	CC0A	CAF6	0ACD	BF09	CDEF	0A78
0AC0	B7C8	CDDF	0921	2041	46C3	9607	2A21	41CD!AF...*!A.
0AD0	EF0A	7C55	1E00	0690	C369	09E7	D0CA	F60A	...U.....
0AE0	FCCC	0A21	0000	221D	4122	1F41	3E08	013E	...!.."A".A>..>
0AF0	04C3	9F0A	E7C8	1E18	C3A2	1947	4F57	5FB7GOW..
0B00	C8E5	CDBF	09CD	DF09	AE67	FC1F	0B3E	9890>..
0B10	CDD7	077C	17DC	A807	0600	DCC3	07E1	C91B
0B20	7AA3	3CC0	0BC9	E7F8	CD55	09F2	370B	CD82	..<.....U..7...
0B30	09CD	370B	C37B	09E7	F830	1E28	B9CD	8E0A	..7.....0.(....
0B40	2124	417E	FE98	3A21	41D0	7ECD	FB0A	3698	!\$A....!A.....6.
0B50	7BF5	7917	CD62	07F1	C921	2441	7EFE	90DA!\$A....
0B60	7F0A	2014	4F2B	7EEE	8006	062B	B605	20FB	...O+.....+....
0B70	B721	0080	CA9A	0A79	FEB8	D0F5	CDBF	09CD	.!.....
0B80	DF09	AE2B	36B8	F5FC	A00B	2123	413E	B890	...+6.....!#A>..
0B90	CD69	0DF1	FC20	0DAF	321C	41F1	D0C3	D80C2.A.....
0BA0	211D	417E	35B7	2328	FAC9	E521			!A.5.#(..!....
0BB0	2812	3E10	29DA	3D27	EB29	EB30	0409	DA3D	(.>.)='.)0...=
0BC0	273D	20F0	EBE1	C97C	179F	47CD	510C	7998	'=.....G.Q...
0BD0	1803	7C17	9F47	E57A	179F	1988	0FAC	F299G.....
0BE0	0AC5	EBCD	CF0A	F1E1	CDA4	09EB	CD6B	0CC3
0BF0	8F0F	7CB5	CA9A	0AE5	D5CD	450C	C544	4D21E..DMI

0C00			2938	1FEB	29EB	3004	09DA	260C	..>.)8..).0...&.
0C10	3D20	F1C1	D17C	B7FA	1F0C	D178	C34D	0CEE	=.....M..
0C20	80B5	2813	EB01	C1E1	CDCF	0AE1	CDA4	09CD	..(.....
0C30	CF0A	C1D1	C347	0878	B7C1	FA9A	0AD5	CDCFG.....
0C40	0AD1	C382	097C	AA47	CD4C	0CEB	7CB7	F29AG.L.....
0C50	0AAF	4F95	6F79	9C67	C39A	0A2A	2141	CD51	..O.....*1A.Q
0C60	0C7C	EE80	B5C0	EBCD	EF0A	AF06	98C3	6909
0C70	212D	417E	EE80	7721	2E41	7EB7	C847	2B4E	!-A....!A...G+N
0C80	1124	411A	B7CA	F409	9030	162F	3CF5	0E08	.\$A.....0./<...
0C90	23E5	1A46	7778	121B	2B0D	20F6	E146	2B4E	#..F....+....F+N
0CA0	F1FE	39D0	F5CD	DF09	2336	0047	F121	2D41	..9.....#6.G.!-A
0CB0	CD69	0D3A	2641	321C	4178	B7F2	CF0C	CD33&A2.A.....3
0CC0	0DD2	0E0D	EB34	CAB2	07CD	900D	C30E	0DCD4.....
0CD0	450D	2125	41DC	570D	AF47	3A23	41B7	201E	E.!%A.W..G:#A...
0CE0	211C	410E	0856	777A	230D	20F9	78D6	08FE	!.A..V..#.....
0CF0	C020	E6C3	7807	0521	1C41	CD97	0DB7	F2F6!A.....
0D00	0C78	B728	0921	2441	8677	D278	07C8	3A1C	...(!\$A.....:
0D10	41B7	FC20	0D21	2541	7EE6	802B	2BAE	77C9	A....!%A...++...
0D20	211D	4106	0734	C023	0520	FA34	CAB2	072B	!.A..4.#...4...+
0D30	3680	C921	2741	111D	410E	07AF	1A8E	1213	6..!'A..A.....
0D40	230D	20F8	C921	2741	111D	410E	07AF	1A9E	#.....!'A..A.....
0D50	1213	230D	20F8	C97E	2F77	211C	4106	08AF	..#...../.!A...
0D60	4F79	9E77	2305	20F9	C971	E5D6	0838	0EE1	O...#.....8..
0D70	E511	0008	4E73	592B	1520	F918	EEC6	0957N.Y+.....W
0D80	AFE1	15C8	E51E	087E	1F77	2B1D	20F9	18F0+.....
0D90	2123	4116	0118	ED0E	087E	1777	230D	20F9	!#A.....#...
0DA0	C9CD	5509	C8CD	0A09	CD39	0E71	1306	071A	..U.....9.....
0DB0	13B7	D528	170E	08C5	1F47	DC33	0DCD	900D	...(!.....G.3....
0DC0	78C1	0D20	F2D1	0520	E6C3	D80C	2123	41CD!#A.
0DE0	700D	18F1					11D4	0D21!#A.
0DE0	2741	CDD3	093A	2E41	B7CA	9A19	CD07	0934	'A....:A.....4
0DF0	34CD	390E	2151	4171	4111	4A41	2127	41CD	4.9.!QA.A.JA!'A.
0E00	4B0D	1A99	3F38	0B11	4A41	2127	41CD	390D	K...78..JA!'A.9.
0E10	AFDA	1204	3A23	413C	3D1F	FA11	0D17	211D#A<=.....!
0E20	410E	07CD	990D	214A	41CD	970D	78B7	20C9	A.....!JA.....
0E30	2124	4135	20C3	C3B2	0779	322D	412B	1150	!\$A5.....2-A+.P
0E40	4101	0007	7E12	711B	2B05	20F8	C9CD	FC09	A.....+.....
0E50	EB2B	7EB7	C8C6	02DA	B207	77E5	CD77	0CE1	.+.....
0E60	34C0	C3B2	07CD	7807	CDEC	0AF6	AFEB	01FF	4.....
0E70	0060	68CC	9A0A	EB7E	FE2D	F5CA	830E	FE2B-.....+
0E80	2801	2BD7	DA29	0FFE	2ECA	E40E	FE45	2814	(.+...).....E(.</td></tr>

0F00	B10A	F1C4	DB0A	C1D1	E1C9	C8F5	E7F5	E43E>
0F10	09F1	EC4D	0EF1	3DC9	D5E5	F5E7	F5E4	9708	...M..=.....
0F20	F1EC	DC0D	F1E1	D13C	C9D5	7889	47C5	E57E<....G...
0F30	D630	F5E7	F25D	0F2A	2141	11CD	0CDF	3019	.0.....*!A....0.
0F40	545D	2929	1929	F14F	097C	B7FA	570F	2221	T.))..).O....W."!
0F50	41E1	C1D1	C383	0E79	F5CD	CC0A	3730	1801	A.....70..
0F60	7494	1100	24CD	0C0A	F274	0FCD	3E09	F1CD\$.>...
0F70	890F	18DD	CDE3	0ACD	4D0E	CDFC	09F1	CD64M.....
0F80	09CD	E30A	CD77	0C18	C8CD	A409	CD64	09C1
0F90	D1C3	1607	7BFE	0A30	0907	0783	0786	D6300.....0
0FA0	5FFA	1E32	C3BD	0EE5	2124	19CD	A728	E1CD	...2....1\$...(.
0FB0	9A0A	AFCD	3410	B6CD	D90F	C3A6	28AF	CD344.....(..4
0FC0	10E6	0828	0236	2BEB	CD94	09EB	F2D9	0F36	...(.6+.....6
0FD0	2DC5	E5CD	7B09	E1C1	B423	3630	3AD8	4057	-.....#60:..@W
0FE0	173A	AF40	DA9A	10CA	9210	FE04	D23D	1001	..:..@.....=..
0FF0	0000	CD2F	1321	3041	460E	203A	D840	5FE6	.../.!0AF...:..@..
1000	2028	0778	B90E	2A20	0141	71D7	2814	FE45	.(.....*..A..(..E
1010	2810	FE44	280C	FE30	28F0	FE2C	28EC	FE2E	(..D(..0(..,(...
1020	2003	2B36	307B	E610	2803	2B36	247B	E604	..+60...(+6\$...
1030	C02B	70C9	32D8	4021	3041	3620	C9FE	05E5	..+..2.@!0A6.....
1040	DE00	1757	14CD						...W.....W..
1050	BA30	043C	473E	02D6	02E1	F5CD	9112	3630	.0.<G>.....60
1060	CCC9	09CD	A412	2B7E	FE30	28FA	FE2E	C4C9+..0(.....
1070	09F1	281F	F5E7	3E22	8F77	23F1	362B	F285	..(....>"..#.6+..
1080	1036	2D2F	3C06	2F04	D60A	30FB	C63A	2370	.6-/<./...0...:..#.
1090	2377	2336	00EB	2130	41C9	23C5	FE04	7AD2	#. #6..10A.#.....
10A0	0911	1FDA	A311	0103	06CD	8912	D17A	D605
10B0	F469	12CD	2F13	7BB7	CC2F	093D	F469	12E5	.../...../..=.....
10C0	CDF5	0FE1	2802	7023	3600	212F	4123	3AF3(..#6.1/A#:.:
10D0	4095	92C8	7EFE	2028	F4FE	2A28	F02B	E5F5	@.....(..*(.+.:
10E0	01DF	10C5	D7FE	2DC8	FE2B	C8FE	24C8	C1FE-..+..\$...
10F0	3020	0F23	D730	0B2B	012B	77F1	28FB	C1C3	0..#.0.+.+.:(...
1100	CE10	F128	FDE1	3625	C9E5	1FDA	AA11	2814	...(..6\$.>.....(.
1110	1184	13CD	490A	1610	FA32	11E1	C1CD	BD0FI.....2.....
1120	2B36	25C9	010E	B611	CA1B	CD0C	0AF2	1B11	+6\$.>.....
1130	1606	CD55	09C4	0112	E1C1	FA57	11C5	5F78	...U.....W....
1140	9293	F469	12CD	7D12	CDA4	12B3	C477	12B3
1150	C491	12D1	C3B6	105F	79B7	C416	0F83	FA62
1160	11AF	C5F5	FC18	0FFA	6411	C17B	90C1	5F82
1170	78FA	7F11	9293	F469	12C5	CD7D	1218	11CD
1180	6912	79CD	9412	4FAF	9293	CD69	12C5	474FO.....GO
1190	CDA4	12C1	B120	032A	F340	833D	F469	1250*.@.=...P
11A0	C3BF	10E5	D5CD	CC0A	D1AF	CAB0	111E	1001
11B0	1E06	CD55	0937	C401	12E1	C1F5	79B7	F5C4	...U.7.....
11C0	160F	804F	7AE6	04FE	019F	5781	4F93	F5C5	...O.....W.O...
11D0	FC18	0FFA	D011	C1F1	C5F5	FADE	11AF	2F3C/<
11E0	803C	8247	0E00	CDA4	12F1	F471	12C1	F1CC	.<.G.....
11F0	2F09	F138	0383	9092	C5CD	7410	EBD1	C3BF	/..8.....

1200	10D5	AFF5	E7E2	2212	3A24	41FE	91D2	2212"	:\$A..."
1210	1164	1321	2741	CDD3	09CD	A10D	F1D6	0AF5	...!'A.....	
1220	18E6	CD4F	12E7	300B	0143	9111	F94F	CD0C	...O..0..C...O..	
1230	0A18	0611	6C13	CD49	0AF2	4B12	F1CD	0B0FI..K.....	
1240	F518	E2F1	CD18	0FF5	CD4F	12F1	B7D1	C9E7O.....	
1250	EA5E	1201	7494	11F8	23CD	0C0A	1806	1174#.....	
1260	13CD	490A	E1F2	4312	E9B7	C83D	3630	2318	..I...C.....=60#.	
1270	F920	04C8	CD91	1236	3023	3D18	F67B	823C60#=#.....<	
1280	473C	D603	30FC	C605	4F3A	D840	E640	C04F	G<..0...O:..@..@.O	
1290	C905	2008	362E	22F3	4023	48C9	0DC0	362C6."..@#H...6,	
12A0	230E	03C9	D5E7	E2EA	12C5	E5CD	FC09	217C	#.....!.	
12B0	13CD	F709	CD77	0CAF	CD7B	0BE1	C111	8C13	
12C0	3E0A	CD91	12C5	F5E5	D506	2F04	E1E5	CD48	>...../.....H	
12D0	0D30	F8E1	CD36	0DEB	E170	23F1	C13D	20E2	.0...6....#..=#..	
12E0	C5E5	211D	41CD	B109	180C	C5E5	CD08	073C	..!..A.....<	
12F0	CDFB	0ACD	B409	E1C1	AF11	D213	3FCD	9112?...	
1300	C5F5	E5D5	CDBF	09E1	062F	047B	965F	237A/.....#.	
1310	9E57	2379	9E4F	2B2B	30F0	CDB7	0723	CDB4	.W#..O++0....#..	
1320	09EB	E170	23F1	C138	D313	133E	0418	06D5#..8...>....	
1330	11D8	133E	05CD	9112	C5F5	E5EB	4E23	46C5	...>.....N#F.	
1340	23E3	EB2A	2141	062F	047D	936F	7C9A	6730	#..*!A./.....0	
1350	F719	2221	41D1	E170	23F1	C13D	20D7	CD91	..!"A...#..=#....	
1360	1277	D1C9			F902	15A2	FDFE	9F311	
1370	A95F	63B2	FEFF	03BF	C91B	0EB6			
1380				04BF	C91B	0EB6	0080	C6A4	
1390	7E8D	0300	407A	10F3	5A00	00A0	724E	1809@...Z....N..	
13A0	0000	10A5	D4E8		00E8	7648	1700	H....	
13B0	E40B	5402		00CA	9A3B			E1F5	..T.....;.....	
13C0	0500	0000	8096	9800		0040	420F	@B...	
13D0		A086	0110	2700	1027	E803	6400	0A00'...'.....	
13E0	0100	2182	09E3	E9CD	A409	2180	13CD	B109	..!.....!.....	
13F0	1803	CDB1	0AC1	D1CD	5509	7828	3CF2	0414U..(<...	
1400	B7CA	9A19	B7CA	7907	D5C5	79F6	7FCD	BF09	
1410	F221	14D5	C5CD	400B	C1D1	F5CD	0C0A	E17C	..!....@.....	
1420	1FE1	2223	41E1	2221	41DC	E213	CC82	09D5	.."#A."!A.....	
1430	C5CD	0908	C1D1	CD47	08CD	A409	0138	8111G.....8..	
1440	3BAA	CD47	083A	2441	FE88	D231	09CD	400B	;..G.:\$A...1..@.	
1450	C680	C602	DA31	09F5	21F8	07CD	0B07	CD411..!.....A	
1460	08F1	C1D1	F5CD	1307	CD82	0921	7914	CDA9!....	
1470	1411	0000	C14A	C347	0808	402E	9474	704FJ.G..@....O	
1480	2E77	6E02	887A	E6A0	2A7C	50AA	AA7E	FFFF*..P.....	
1490	7F7F	0000	8081	0000	0081	CDA4	0911	320C2.	
14A0	D5E5	CDBF	09CD	4708	E1CD	A409	7E23	CDB1G.....#..	
14B0	0906	F1C1	D13D	C8D5	C5F5	E5CD	4708	E1CD=.....G...	
14C0	C209	E5CD	1607	E118	E9CD	7F0A	7CB7	FA4AJ	
14D0	1EB5	CAF0	14E5	CDF0	14CD	BF09	EBE3	C5CD	
14E0	CF0A	C1D1	CD47	0821	F807	CD0B	07C3	400BG.!.....@.	
14F0	2190	40E5	1100	004B	2603	2E08	EB29	EB79	l.@....K&.....)	

1500	174F	E37E	0777	E3D2	1615	E52A	AA40	19EB	.O.....*..@..
1510	3AAC	4089	4FE1	2DC2	FC14	E323	E325	C2FA	::@.O.-.....#.8..
1520	14E1	2165	B019	22AA	40CD	EF0A	3E05	8932	..!...".@....>..2
1530	AC40	EB06	8021	2541	702B	704F	0600	C365	.@...!%A.+..O....
1540	0721	8B15	CD0B	07CD	A409	0149	8311	DB0F	.!.....I....
1550	CDB4	09C1	D1CD	A208	CDA4	09CD	400B	C1D1@...
1560	CD13	0721	8F15	CD10	07CD	5509	37F2	7715	...!.....U.7...
1570	CD08	07CD	5509	B7F5	F482	0921	8F15	CD0BU.....!....
1580	07F1	D482	0921	9315	C39A	14DB	0F49	8100!.....I..
1590	0000	7F05	BAD7	1E86	6426	9987	5834	2387&..X4#.
15A0	E05D	A586	DA0F	4983	CDA4	09CD	4715	C1E1I.....G...
15B0	CDA4	09EB	CDB4	09CD	4115	C3A0	08CD	5509A.....U.
15C0	FCE2	13FC	8209	3A24	41FE	8138	0C01	0081:\$A..8....
15D0	5159	CDA2	0821	1007	E521	E315	CD9A	1421	QY...!...!.....!
15E0	8B15	C909	4AD7	3B78	026E	847B	FEC1	2F7CJ.;...../.
15F0	7431	9A7D	843D	5A7D	C87F	917E	E4BB	4C7E	.1...=Z.....L.
1600	6CAA	AA7F			8A09	370B	7709	D4277.....'
1610	EF2A	F527	E713	C914	0908	3914	4115	4715	.*.'.....9.A.G.
1620	A815	BD15	AA2C	5241	5841	5E41	6141	6441,RAXA.A.A.A
1630	6741	6A41	6D41	7041	7F0A	B10A	DB0A	260B	.A.A.A.A.....&.
1640	032A	3628	C52A	0F2A	1F2A	612A	912A	9A2A	.*6(*.*.*.*.*.*
1650	C54E	44C6	4F52	D245	5345	54D3	4554	C34C	.ND.OR.ESET.ET.L
1660	53C3	4D44	D241	4E44	4F4D	CE45	5854	C441	S.MD.ANDOM.EXT.A
1670	5441	C94E	5055	54C4	494D	D245	4144	CC45	TA.NPUT.IM.EAD.E
1680	54C7	4F54	4FD2	554E	C946	D245	5354	4F52	T.OTO.UN.F.ESTOR
1690	45C7	4F53	5542	D245	5455	524E	D245	4DD3	E.OSUB.ETURN.EM.
16A0	544F	50C5	4C53	45D4	524F	4ED4	524F	4646	TOP.LSE.ROFF
16B0	C445	4653	5452	C445	4649	4E54	C445	4653	.EFSTR.EFINT.EFS
16C0	4E47	C445	4644	424C	CC49	4E45	C544	4954	NG.EFDBL.INE.DIT
16D0	C552	524F	52D2	4553	554D	45CF	5554	CF4E	.RROR.ESUME.UT.N
16E0	CF50	454E	C649	454C	44C7	4554	D055	54C3	.PEN.IELD.ET.UT.
16F0	4C4F	5345	CC4F	4144	CD45	5247	45CE	414D	LOSE.OAD.ERGE.AM
1700	45CB	494C	4CCC	5345	54D2	5345	54D3	4156	E.ILL.SET.SET.AV
1710	45D3	5953	5445	4DCC	5052	494E	54C4	4546	E.YSTEM.PRINT.EF
1720	D04F	4B45	D052	494E	54C3	4F4E	54CC	4953	.OKE.RINT.ONT.IS
1730	54CC	4C49	5354	C445	4C45	5445	C155	544F	T.LIST.ELETE.UTO
1740	C34C	4541	52C3	4C4F	4144	C353	4156	45CE	.LEAR.LOAD.SAVE.
1750	4557	D441	4228	D44F	C64E	D553	494E	47D6	EW.AB(.O.N.SING.
1760	4152	5054	52D5	5352	C552	4CC5	5252	D354	ARPTR.SR.RL.RR.T
1770	5249	4E47	24C9	4E53	5452	D04F	494E	54D4	RING\$.NSTR.OINT.
1780	494D	4524	CD45	4DC9	4E4B	4559	24D4	4845	IME\$.EM.NKEY\$.HE
1790	4ECE	4F54	D354	4550	ABAD	AAAF	DBC1	4E44	N.OT.TEP.....ND
17A0	CF52	BEBD	BCD3	474E	C94E	54C1	4253	C652	.R....GN.NT.BS.R
17B0	45C9	4E50	D04F	53D3	5152	D24E	44CC	4F47	E.NP.OS.QR.ND.OG
17C0	C558	50C3	4F53	D349	4ED4	414E	C154	4ED0	.XP.OS.IN.AN.TN.
17D0	4545	4BC3	5649	C356	53C3	5644	C54F	46CC	EEK.VI.VS.VD.OF.
17E0	4F43	CC4F	46CD	4B49	24CD	4B53	24CD	4B44	OC.OF.KI\$.KS\$.KD
17F0	24C3	494E	54C3	534E	47C3	4442	4CC6	4958	\$.INT.SNG.DBL.IX

1800	CC45	4ED3	5452	24D6	414C	C153	43C3	4852	.EN.TR\$.AL.SC.HR
1810	24CC	4546	5424	D249	4748	5424	CD49	4424	\$.EFT\$.IGHT\$.ID\$
1820	A780	AE1D	A11C	3801	3501	C901	7341	D3018.5....A..
1830	B622	051F	9A21	0826	EF21	211F	C21E	A31E	."...!.&.!!.....
1840	3920	91mD	B11E	DE1E	071F	A91D	071F	F71D	9.....
1850	F81D	001E	031E	061E	091E	A341	602E	F41FA.....
1860	AF1F	FB2A	6C1F	7941	7C41	7F41	8241	8541	...*...A.A.A.A.A
1870	8841	8B41	8E41	9141	9741	9A41	A041	B202	.A.A.A.A.A.A.A..
1880	6720	5B41	B12C	6F20	E41D	2E2B	292B	C62B	...A.,.....+)+.+
1890	0820	7A1E	1F2C	F52B	491B	7979	7C7C	7F50,+I.....P
18A0	46DB			0AF4	0AB1	0A77	0C70	0CA1	F.....
18B0	0DE5	0D78	0A16	0713	0747	08A2	080C	0AD2G.....
18C0	0BC7	0BF2	0B90	2439	0A4E	4653	4E52	474F\$9.NFSNRGO
18D0	4446	434F	564F	4D55	4C42	5344	442F	3049	DFCOVOMULBSDD/OI
18E0	4454	4D4F	534C	5353	5443	4E4E	5252	5755	DTMOSLSSTCNNRRWU
18F0	454D	4F46	444C	33D6	006F	7CDE	0067	78DE	EMOFDL3.....
1900	0047	3E00	C94A	1E40	E64D	DB00	C9D3	00C9	.G>..J.@.M.....
1910			4030	004C	43FE	FFE9	4220	4572@0.LC...B.E.
1920	726F	7200	2069	6E20	0052	4541	4459	0D00READY..
1930	4272	6561	6B00	2104	0039	7E23	FE81	C04E	B.....!..9.#...N
1940	2346	23E5	6960	7AB3	EB28	02FB	DF01	0E00	#F#.....(.....
1950	E1C8	0918	E5CD	6C19	C5E3	C1DF	7E02	C80B
1960	2B18	F8E5	2AFD	4006	0009	093E	E53E	C695	+...*.@.....>.>..
1970	6F3E	FF9C	3804	6739	E1D8	1E0C	1824	2AA2	.>..8..9.....\$*.
1980	407C	A53C	2808	3AF2	40B7	1E22	2014	C3C1	@..<(.....@..".
1990	1D2A	DA40	22A2	401E	0201	1E14	011E	0001	.*."@.....
19A0	1E24	2AA2	4022	EA40	22EC	4001	B419	2AE8	.\$*."@".@...*.
19B0	40C3	9A1B	C17B	4B32	9A40	2AE6	4022	EE40	@.....K2.@*."@
19C0	EB2A	EA40	7CA5	3C28	0722	F540	EB22	F740	*.@..<("@"@
19D0	2AF0	407C	B5EB	21F2	4028	08A6	2005	35EB	*.@...!.@(....5.
19E0	C336	1DAF	7759	CDF9	2021	C918	CDA6	4157	.6...Y...!....AW
19F0	3E3F	CD2A	0319	7ECD	2A03	D7CD	2A03	211D	>?.*.....*...*.!
1A00	19E5	2AEA	40E3	CDA7	28E1	11FE	FFDF	CA74	..*.@... (.....
1A10	067C	A53C	C4A7	0F3E	C1CD	8B03	CDAC	41CD	...<...>.....A.
1A20	F801	CDF9	2021	2919	CDA7	282A	9A40	D602	1) (. @

1B00	B6C8	2323	23AF	BE23	20FC	EB73	2372	18EC	...###...#...#...
1B10			2809	D1CD	4F1E	D528	0BCF	CE11(....O...(....
1B20	FAFF	C44F	1EC2	9719	EBD1	E3E5	2AA4	4044	...O.....*.@D
1B30	4D7E	23B6	2BC8	2323	7E23	666F	DF60	697E	M.#.+##.#.....
1B40	2366	6F3F	C83F	D018	E6C0	CDC9	012A	A440	#..?.?.....*.@
1B50	CDF8	1D32	E140	7723	7723	22F9	402A	A440	...2.@.#"#".@*.@
1B60	2B22	DF40	061A	2101	4136	0423	10FB	AF32	+".@...!.A6.#...2
1B70	F240	6F67	22F0	4022	F740	2AB1	4022	D640	.@..".@".@*."@.
1B80	CD91	1D2A	F940	22FB	4022	FD40	CDBB	41C1	...*."@".@".@..A.
1B90	2AA0	402B	2B22	E840	2323	F921	B540	22B3	*.@++".@##.!.@".
1BA0	40CD	8B03	CD69	21AF	676F	32DC	40E5	C52A	@.....!...2.@..*
1BB0	DF40	C93E	3FCD	2A03	3E20	CD2A	03C3	6103	.@.>?.*.>.*....
1BC0	AF32	B040	4FEB	2AA7	402B	2BEB	7EFE	20CA	.2.@O.*.@++.....
1BD0	5B1C	47FE	22CA	771C	B7CA	7D1C	3AB0	40B7	..G.".....:@.
1BE0	7EC2	5B1C	FE3F	3EB2	CA5B	1C7E	FE30	38s5?>.....08.
1BF0	FE3C	DA5B	1CD5	114F	16C5	013D	1CC5	067F	.<.....O...=....
1C00	7EFE	6138	07FE	7B30	03E6	5F77	4EEB	23B6	...8...0....N.#.
1C10	F20E	1C04	7EE6	7FC8	B920	F3EB	E513	1AB7
1C20	FA39	1C4F	78FE	8D20	02D7	2B23	7EFE	6138	.9.O.....+#...8
1C30	02E6	5FB9	28E7	E118	D348	F1EB	C9EB	79C1(....H.....
1C40	D1EB	FE95	363A	2002	0C23	FEFB	200C	363A6:....#....6:
1C50	2306	9370	23EB	0C0C	181D	EB23	1213	0CD6	#...#.....#.....
1C60	3A28	04FE	4E20	0332	B040	D659	C2CC	1B47	:(.N..2.@.Y...G
1C70	7EB7	2809	B828	E423	120C	1318	F321	0500	..(.(.#.....!..
1C80	4409	444D	2AA7	402B	2B2B	1213	1213	12C9	D.DM*.@+++.....
1C90	7C92	C07D	93C9	7EE3	BE23	E3CA	781D	C397#.....
1CA0	193E	6432	DC40	CD21	1FE3	CD36	19D1	2005	..>.2.@.!.6....
1CB0	09F9	22E8	40EB	0E08	CD63	19E5	CD05	1FE3	..".@.....
1CC0	E52A	A240	E3CF	BDE7	CAF6	0AD2	F60A	F5CD	.*.@.....
1CD0	3723	F1E5	F2EC	1CCD	7F0A	E311			7#.....
1CE0	CCCC	012B	D5E5	EBCD	9E09	1822	CDB1	0ACD	...+.....".....
1CF0	BF09	E1C5	D501	0081	515A	7EFE	CC3E	0120QZ...>..
1D00	0ECD	3823	E5CD	B10A	CDBF	09CD	5509	E1C5	..8#.....U...
1D10	D54F	E747	C5E5	2ADF	40E3	0681	C53v	CD58	.O.G.*.@....3.X
1D20	03B7	C4A0	1D22	E640	ED73	E840	7EFE	3A28".@....@...:(
1D30	29B7	C297	1923	7E23	B6CA	7E19	235E	2356).....#.#.....#.#V
1D40	EB22	A240	3A1B	41B7	280F	D53E	3CCD	2A03	..".@:..A.(..><.*.
1D50	CDAF	0F3E	3ECD	2A03	D1EB	D711	1E1D	D5C8	...>>.*.....
1D60	D680	DA21	1FFE	3CD2	E72A	074F	0600	EB21	...!.<.*.O...!
1D70	2218	094E	2346	C5EB	237E	FE3A	D0FE	20CA	"..N#F..#...:....
1D80	781D	FE0B	3005	FE09	D278	1DFE	303F	3C3D0.....0?<=
1D90	C9EB	2AA4	402B	22FF	40EB	C9CD	5803	B7C8	..*.@+."@...X...
1DA0	FE60	CC84	0332	9940	3DC0	3CC3	B41D	C0F52.@=<.....
1DB0	CCBB	41F1	22E6	4021	B540	22B3	4021	F6FF	..A."@!.@"@!..
1DC0	C12A	A240	E5F5	7DA4	3C28	0922	F540	2AE6	.*.@.....<("@"@*.
1DD0	4022	F740	CD8B	03CD	F920	F121	3019	C206	@".@.....10...
1DE0	1AC3	181A	2AF7	407C	B51E	20CA	A219	EB2A*.@.....*
1DF0	F540	22A2	40EB	C93E	AF32	1B41	C9F1	E1C9	."@..>.2.A....

1E00	1E03	011E	0201	1E04	011E	08CD	3D1E	0197=...
1E10	19C5	D8D6	414F	47D7	FECE	2009	D7CD	3D1EAOG.....=.
1E20	D8D6	4147	D778	91D8	3CE3	2101	4106	0009	..AG....<!.A...
1E30	7323	3D20	FBE1	7EFE	2CC0	D718	CE7E	FE41	.#.=.....,.....A
1E40	D8FE	5B3F	C9D7	CD02	2BF0	1E08	C3A2	197E	...?.....+.....
1E50	FE2E	EB2A	EC40	EBCA	781D	2B11	0000	D7D0	...*.@.....+.....
1E60	E5F5	2198	19DF	DA97	1962	6B19	2919	29F1	..!......).).
1E70	D630	5F16	0019	EBE1	18E4	CA61	1BCD	461E	.0.....F.
1E80	2BD7	C0E5	2AB1	407D	935F	7C9A	57DA	7A19	+...*.@.....W...
1E90	2AF9	4001			D27A	19EB	22A0	40E1	*.@.(.....".@.
1EA0	C361	1BCA	5D1B	CDC7	41CD	611B	011E	1D18A.....
1EB0	100E	03CD	6319	C1E5	E52A	A240	E33E	91F5*.@.>..
1EC0	33C5	CD5A	1ECD	071F	E52A	A240	DFE1	23DC	3..Z.....*.@.#.
1ED0	2F1B	D42C	1B60	692B	D81E	0EC3	A219	C016	/.....+.....
1EE0	FFCD	3619	F922	E840	FE91	1E04	C2A2	19E1	..6..".@.....
1EF0	22A2	4023	7CB5	2007	3ADD	40B7	C218	1A21	".@#.....: @.....!
1F00	1E1D	E33E	E101				4847	7EB7	...>.....HG..
1F10	C8B8	C823	FE22	28F3	D68F	20F2	B88A	5718	...#."(.....W.
1F20	EDCD	0D26	CFD5	EB22	DF40	EBD5	E7F5	CD37	...&....".@.....7
1F30	23F1	E3C6	03CD	1928	CD03	0AE5	2028	2A21	#.....(.....(*!
1F40	41E5	235E	2356	2AA4	40DF	300E	2AA0	40DF	A.#.#V*.@.0.*.@.
1F50	D130	0F2A	F940	DF30	093E	D1CD	F529	EBCD	.0.*.@.0.>....)
1F60	4328	CDF5	29E3	CDD3	09D1	E1C9	FE9E	2025	C(..).....&
1F70	D7CF	8DCD	5A1E	7AB3	2809	CD2A	1B50	59E1Z...(*.PY.
1F80	D2D9	1EEB	22F0	40EB	D83A	F240	B7C8	3A9A".@.....@.....
1F90	405F	C3AB	19CD	1C2B	7E47	FE91	2803	CF8D	@.....+G..(...
1FA0	2B4B	0D78	CA60	1DCD	5B1E	FE2C	C018	F311	+K.....,.....
1FB0	F240	1AB7	CAA0	193C	329A	4012	7EFE	8728	@.....<2.@.....(
1FC0	0CCD	5A1E	C07A	B3C2	C51E	3C18	02D7	C02A	..Z.....<.....*
1FD0	EE40	EB2A	EA40	22A2	40EB	C07E	B720	0423	..@.*@"@.....#
1FE0	2323	2323	7AA3	3CC2	051F	3ADD	403D	CABE	####<.....@=..
1FF0	1DC3	051F	CD1C	2BC0	B7CA	4A1E	3D87	5FFE+...J.=...
2000	2D38	021E	26C3	A219	110A		2817	CD4F	-8..&.....(..O
2010	1EEB	E328	11EB	CF2C	EB2A	E440	EB28	06CD	...(...,*.@.(..
2020	5A1E	C297	19EB	7CB5	CA4A	1E22	E440	32E1	Z.....J."@2.
2030	40E1	22E2	40C1	C333	1ACD	3723	7EFE	2CCC	@."@..3..7#...,
2040	781D	FECA	CC78	1D2B	E5CD	9409	E128	07D7+.....(..
2050	DAC2	1EC3	5F1D	1601	CD05	1FB7	C8D7	FE95>2.@.....
2060	20F6	1520	F318	E83E	0132	9C40	C39B	20CD>2.@.....
2070	CA41	FE40	2019	CD01	2BFE	04D2	4A1E	E521	.A.@.....+...J..!
2080	003C	1922	2040	7BE6	3F32	A640	E1CF	2CFE	.<."@..?2.@...,
2090	2320	08CD	8402	3E80	329C	402B	D7CC	FE20	#.....>2.@+....
20A0	CA69	21FE	BFCA	BD2C	FEBC	CA37	21E5	FE2C	..!......,7!..,
20B0	CA08	21FE	3BCA	6421	C1CD	3723	E5E7	2832	..!.;..!.7#..(2
20C0	CDBD	0FCD	6528	CDCD	412A	2141	3A9C	40B7(..A*1A:..@.
20D0	FAE9	2028	083A	9B40	86FE	8418	093A	9D40	...(..@.....: @
20E0	473A	A640	86B8	D4FE	20CD	AA28	3E20	CD2A	G:..@.....(>..*
20F0	03B7	CCAA	28E1	C39B	203A	A640	B7C8	3E0D(.....: @.>.

2100	CD2A	03CD	D041	AFC9	CDD3	413A	9C40	B7F2	. * . . . A A : . @ . .
2110	1921	3E2C	CD2A	0318	4B28	083A	9B40	FE70	. > , . * . . K (. . . @ . .
2120	C32B	213A	9E40	473A	A640	B8D4	FE20	3034	. + ! : . , @ G : . @ 04
2130	D610	30FC	2F18	23CD	1B2B	E63F	5FCF	292B	. . 0 . / . # . . . + . ? . .) +
2140	E5CD	D341	3A9C	40B7	FA4A	1ECA	5321	3A9B	. . . A : . @ . . . J . . S ! : . .
2150	4018	033A	A640	2F83	300A	3C47	3E20	CD2A	@ @ / . 0 . < G > . . *
2160	0305	20FA	E1D7	C3A0	203A	9C40	B7FC	F801 : . @
2170	AF32	9C40	CDBE	41C9	3F52	4544	4F0D	003A	. 2 . @ . . . A . ? REDO . . :
2180	DE40	B7C2	9119	3AA9	40B7	1E2A	CAA2	19C1	. @ : . @ . . *
2190	2178	21CD	A728	2AE6	40C9	CD28	287E	CDD6	! . ! . . (* . @ . . ((. . .
21A0	41D6	2332	A940	7E20	20CD	9302	E506	FA2A	A . # 2 . @ *
21B0	A740	CD35	0277	23FE	0D28	0210	F52B	3600	. @ . 5 . . # . . (. . . + 6 .
21C0	CDF8	012A	A740	2B18	2201	DB21	C5FE	22C0	. . . * . @ + . " . . ! . . "
21D0	CD66	28CF	3BE5	CDA A	28E1	C9E5	CDB3	1BC1	. . (. ; . . . (.
21E0	DABE	1D23	7EB7	2BC5	CA04	1F36	2C18	05E5	. . . # . . + 6 , . . .
21F0	2AFF	40F6	AF32	DE40	E318	02CF	2CCD	0D26	* . @ . . 2 . @ , . &
2200	E3D5	7EFE	2C28	263A	DE40	B7C2	9622	3AA9 , (& : . @ . . . " : .
2210	40B7	1E06	CAA2	193E	3FCD	2A03	CDB3	1BD1	@ > ? . *
2220	C1DA	BE1D	237E	B72B	C5CA	041F	D5CD	DC41 # . . + A
2230	E7F5	2019	D757	47FE	2228	0516	3A06	2C2B WG . " (. . . . , +
2240	CD69	28F1	EB21	5A22	E3D5	C333	1FD7	F1F5	. . (. . ! Z " . . . 3
2250	0143	22C5	DA6C	0ED2	650E	2BD7	2805	FE2C	. C " + . (. ,
2260	C27F	21E3	2BD7	C2FB	21D1				. . ! . + . . . ! : .
2270	DE40	B7EB	C296	1DD5	CDDF	41B6	2186	22C4	. @ A . ! . "
2280	A728	E1C3	6921	3F45	7874	7261	2069	676E	. (. . . ! ? E
2290	6F72	6564	0D00	CD05	1FB7	2012	237E	23B6 # . # .
22A0	1E06	CAA2	1923	5E23	56EB	22DA	40EB	D7FE # . # V . " . @
22B0	8820	E3C3	2D22			0D26	22DF	40CD - " & " . @ .
22C0	3619	C29D	19F9	22E8	40D5	7E23	F5D5	7E23	6 " . @ . . # . . . #
22D0	B7FA	EA22	CDB1	09E3	E5CD	0B07	E1CD	CB09	. . . "
22E0	E1CD	C209	E5CD	0C0A	1829	2323	2323	4E23) # # # # N #
22F0	4623	E35E	2356	E569	60CD	D20B	3AAF	40FE	F # . . # V @ .
2300	04CA	B207	EBE1	722B	73E1	D55E	2356	23E3 + # V # .
2310	CD39	0AE1	C190	CDC2	0928	09EB	22A2	4069	. 9 (. " . @ .
2320	60C3	1A1D	F922	E840	2ADF	407E	FE2C	C21E " . @ * . @ . , . .
2330	1DD7	CDB9	22CF	282B	1600	D50E	01CD	6319 " . (+
2340	CD9F	2422	F340	2AF3	40C1	7E16	00D6	D438	. . \$ " . @ * . @ 8
2350	13FE	0330	0FFE	0117	AABA	57DA	9719	22D8	. . . 0 W . . . " .
2360	40D7	18E9	7AB7	C2EC	237E	22D8	40D6	CDD8	@ # . " . @
2370	FE07	D05F	3AAF	40D6	03B3	CA8F	2921	9A18 : . @) ! . .
2380	1978	56BA	D0C5	0146	23C5	7AFE	7FCA	D423	. . V # # #
2390	FE51	DAE1	2321	2141	B73A	AF40	3D3D	3DCA	. Q . . # ! ! A . . . @ = = = .
23A0	F60A	4E23	46C5	FAC5	2323	4E23	46C5	F5B7	. . N # F . . . # # N # F . . .
23B0	E2C4	23F1	2338	0321	1D41	4E23	4623	C54E	. . # . # 8 . ! . AN # F # . N
23C0	2346	C506	F1C6	034B	47C5	0106	24C5	2AD8	# F KG . . . \$. * .
23D0	40C3	3A23	CDB1	0ACD	A409	01F2	1316	7F18	@ . : #
23E0	ECD5	CD7F	0AD1	E501	E925	18E1	78FE	64D0 #
23F0	C5D5	1104	6421	B825	E5E7	C295	232A	2141 ! . % # * ! A

2400	E501	8C25	18C7	C179	32B0	4078	FE08	2828	...%....2.@...((
2410	3AAF	40FE	08CA	6024	5778	FE04	CA72	247A	::@....\$W....\$.
2420	FE03	CAF6	0AD2	7C24	21BF	1806	0009	094E\$!.....N
2430	2346	D12A	2141	C5C9	CDDB	0ACD	FC09	E122	#F.*!A....."
2440	1F41	E122	1D41	C1D1	CDB4	09CD	DB0A	21AB	.A.".A.....!
2450	183A	B040	07C5	4F06		C17E	2366	6FE9	::@..O.....#...
2460	C5CD	FC09	F132	AF40	FE04	28DA	E122	21412.@..(."!A
2470	18D9	CDB1	0AC1	D121	B518	18D5	E1CD	A409!.....
2480	CDCF	0ACD	BF09	E122	2341	E122	2141	18E7"#A."!A..
2490	E5EB	CDCF	0AE1	CDA4	09CD	CF0A	C3A0	08D7
24A0	1E28	CAA2	19DA	6C0E	CD3D	1ED2	4025	FECD	.(.....=@%..
24B0	28ED	FE2E	CA6C	0EFE	CECA	3225	FE22	CA66	(.....2%."..
24C0	28FE	CBCA	C425	FE26	CA94	41FE	C320	0AD7	(.....%&..A.....
24D0	3A9A	40E5	CDF8	27E1	C9FE	C220	0AD7	E52A	::@...!.....*
24E0	EA40	CD66	0CE1	C9FE	C020	14D7	CF28	CD0D	.@.....(.
24F0	26CF	29E5	EB7C	B5CA	4A1E	CD9A	0AE1	C9FE	&.).....J.....
2500	C1CA	FE27	FEC5	CA9D	41FE	C8CA	C927	FEC7	...!.....A.....!
2510	CA76	41FE	C6CA	3201	FEC9	CA9D	01FE	C4CA	..A...2.....
2520	2F2A	FEBE	CA55	41D6	D7D2	4E25	CD35	23CF	/*...UA...N%.5#.
2530	29C9	167D	CD3A	232A	F340	E5CD	7B09	E1C9).....:#*.@.....
2540	CD0D	26E5	EB22	2141	E7C4	F709	E1C9	0600	..&."!A.....
2550	074F	C5D7	79FE	4138	16CD	3523	CF2C	CDF4	.O...A8..5#.,..
2560	0AEB	2A21	41E3	E5EB	CD1C	2BEB	E318	14CD	..*!A.....+.....
2570	2C25	E37D	FE0C	3807	FE1B	E5DC	B10A	E111	,%....8.....
2580	3E25	D501	0816	094E	2366	69E9	CDD7	297E	>%....N#.....).
2590	234E	2346	D1C5	F5CD	DE29	D15E	234E	2346	#N#F.....)#N#F
25A0	E17B	B2C8	7AD6	01D8	AFBB	3CD0	151D	0ABE<.....
25B0	2303	28ED	3FC3	6009	3C8F	C1A0	C6FF	9FCD	#.(?...<.....
25C0	8D09	1812	165A	CD3A	23CD	7F0A	7D2F	6F7CZ.:#.../..
25D0	2F67	2221	41C1	C346	233A	AF40	FE08	3005	/"!A..F#::@..0.
25E0	D603	B737	C9D6	03B7	C9C5	CD7F	0AF1	D101	...7.....
25F0	FA27	C5FE	4620	067B	B56F	7CB2	C97B	A56F	.'..F.....
2600	7CA2	C92B	D7C8	CF2C	0103	26C5	F6AF	32AE	...+...,&...2.
2610	4046	CD3D	1EDA	9719	AF4F	D738	05CD	3D1E	@F.=.....O.8..=.
2620	3809	4FD7	38FD	CD3D	1E30	F811	5226	D516	8.O.8..=.0..R&..
2630	02FE	25C8	14FE	24C8	14FE	21C8	1608	FE23	..%...\$...!...#
2640	C878	D641	E67F	5F16	00E5	2101	4119	56E1	...A.....!A.V.
2650	2BC9	7A32	AF40	D73A	DC40	B7C2	6426	7ED6	+..2.@::@...&..
2660	28CA	E926	AF32	DC40	E5D5	2AF9	40EB	2AFB	(..&2.@...*.@.*.
2670	40DF	E128	191A	6FBC	1320	0B1A	B920	0713	@..(.....
2680	1AB8	CACC	263E					&>...&.....
2690	E3F5	D511	F124	DF28	3611	4325	DFD1	2835\$. (6.C%.. (5
26A0	F1E3	E5C5	4F06	00C5	0303	032A	FD40	E509O.....*..@..
26B0	C1E5	CD55	19E1	22FD	4060	6922	FB40	2B36	...U..".@..".@+6
26C0	00DF	20FA	D173	23D1	7323	72EB	13E1	C957#..#.....W
26D0	5FF1	F1E3	C932	2441	C167	6F22	2141	E7202\$A..."!A..
26E0	0621	2819	2221	41E1	C9E5	2AAE	40E3	57D5	..! ("!A...*.@.W.
26F0	C5CD	451E	C1F1	EBE3	E5EB	3C57	7EFE	2C28	..E.....<W...,(

2700	EECF	2922	F340	E122	AE40	D52A	FB40	3E19	..)"@."@.*@>.
2710	EB2A	FD40	EBDF	3AAF	4028	27BE	2320	087E	.*@...@('.#...
2720	B923	2004	7EB8	3E23	235E	2356	2320	E03A	.#...>##.#V#...:
2730	AE40	B71E	12C2	A219	F196	CA95	271E	10C3	@.....'...
2740	A219	7723	5F16	00F1	7123	7023	4FCD	6319	...#.....#.#O...
2750	2323	22D8	4071	233A	AE40	1779	010B	0030	##".@.#:..@.....0
2760	02C1	0371	2370	23F5	CDAA	0BF1	3D20	EDF5	...#.#.....=...
2770	424B	EB19	38C7	CD6C	1922	FD40	2B36	00DF	BK..8.....".@+6..
2780	20FA	03'7	2AD8	405E	EB29	09EB	2B2B	7323	...W*@..)..++.#
2790	7223	F138	3047	4F7E	2316	E15E	2356	23E3	.#.80GO.#...#V#.
27A0	F5DF	D23D	27CD	AA0B	19F1	3D44	4D20	EB3A	...='.....=DM...:
27B0	AF40	444D	29D6	0438	0429	2806	29B7	E2C2	..@DM)..8.)(..)...
27C0	2709	C109	EB2A	F340	C9AF	E532	AF40	CDD4	'.....*@...2.@..
27D0	27E1	D7C9	2AFD	40EB					'.....*@.l..9.....
27E0	DA29	CDE6	282A	A040	EB2A	D640	7D93	6F7C	..)(.*@.*@.....
27F0	9A67	C366	0C3A	A640	6FAF	67C3	9A0A	CDA9:..@.....
2800	41D7	CD2C	25E5	2190	08E5	3AAF	40F5	FE03	A.,%.!.....@...
2810	CCDA	29F1	EB2A	8E40	E9E5	E607	21A1	184F	..)..*@...l..O
2820	0600	09CD	8625	E1C9	E52A	A240	237C	B5E1%...*@#...
2830	C01E	16C3	A219	CDBD	0FCD	6528	CDDA	2901(..).
2840	2B2A	C57E	23E5	CDBF	28E1	4E23	46CD	5A28	+*..#...(.N#F.Z(
2850	E56F	CDCE	29D1	C9CD	BF28	21D3	40E5	7723)(l.@..#
2860	7323	72E1	C92B	0622	50E5	0EFF	237E	0CB7	.#...+."P...#...
2870	2806	BA28	03B8	20F4	FE22	CC78	1DE3	23EB	(..(.....".....#.
2880	79CD	5A28	11D3	403E	D52A	B340	2221	413E	..Z(..@>.*@"!A>
2890	0332	AF40	CDD3	0911	D640	DF22	B340	E17E	.2.@.....@."@..
28A0	C01E	1EC3	A219	23CD	6528	CDDA	29CD	C409#..(..)...
28B0	1415	C80A	CD2A	03FE	0DCC	0321	0318	F2B7*.....!.....
28C0	0EF1	F52A	A040	EB2A	D640	2F4F	06FF	0923	...*.*@*/O...#
28D0	DF38	0722	D640	23EB	F1C9	F11E	1ACA	A219	.8."@#.....
28E0	BFF5	01C1	28C5	2AB1	4022	D640		(*@".@l...
28F0	2AA0	40E5	21B5	40EB	2AB3	40EB	DF01	F728	*@.!.@.*@.....(
2900	C24A	292A	F940	EB2A	FB40	EBDF	2813	7E23	.J)*@.*@..(..#
2910	2323	FE03	2004	CD4B	29AF	5F16	0019	18E6	##.....K).....
2920	C1EB	2AFD	40EB	DFCA	6B29	7E23	CDC2	09E5	..*@.....).#.....
2930	09FE	0320	EB22	D840	E14E	0600	0909	23EE"@.N.....#.
2940	2AD8	40EB	DF28	DA01	3F29	C5AF	B623	5E23	*@..(..?)...#.#
2950	5623	C844	4D2A	D640	DF60	69D8	E1E3	DFE3	V#.DM*@.....
2960	E560	69D0	C1F1	F1E5	D5C5	C9D1	E17D	B4C8
2970	2B46	2B4E	E52B	6E26	0009	5059	2B44	4D2A	+F+N.+.&..PY+DM*
2980	D640	CD58	19E1	7123	7069	602B	C3E9	28C5	..@.X...#...+..(.
2990	E52A	2141	E3CD	9F24	E3CD	F40A	7EE5	2A21	*!A...\$.....*!
29A0	41E5	861E	1CDA	A219	CD57	28D1	CDDE	29E3	A.....W(..).
29B0	CDDD	29E5	2AD4	40EB	CDC6	29CD	C629	2149	..).*.@.....)!)I
29C0	23E3	E5C3	8428	E1E3	7E23	4E23	466F	2C2D	#.....(....#N#F.,-
29D0	C80A	1203	1318	F8CD	F40A	2A21	41EB	CDF5*!A...
29E0	29EB	C0D5	5059	1B4E	2AD6	40DF	2005	4709)...PY.N*@...G.
29F0	22D6	40E1	C92A	B340	2B46	2B4E	2BDF	C022	".@...*@+F+N+.."

2A00	B340	C901	F827	C5CD	D729	AF57	7EB7	C901	.@...'....).W....
2A10	F827	C5CD	072A	CA4A	1E23	5E23	561A	C93E	.'...*.J.#.#V...>
2A20	01CD	5728	CD1F	2B2A	D440	73C1	C384	28D7	..W(..+*.@.....(.
2A30	CF28	CD1C	2BD5	CF2C	CD37	23CF	29E3	E5E7	.(..+...7#.)...
2A40	2805	CD1F	2B18	03CD	132A	D1F5	F57B	CD57	(...+...*......W
2A50	285F	F11C	1D28	D42A	D440	7723	1D20	FB18	(....(*.@.#....
2A60	CACD	DF2A	AFE3	4F3E	E5E5	7EB8	3802	7811	...*...O>.....8...
2A70	0E00	C5CD	BF28	C1E1	E523	4623	6668	0600(....#F#....
2A80	0944	4DCD	5A28	6FCD	CE29	D1CD	DE29	C384	.DM.Z(...).)..
2A90	28CD	DF2A	D1D5	1A90	18CB	EB7E	CDE2	2A04	(..*.....*.
2AA0	05CA	4A1E	C51E	FFFE	2928	05CF	2CCD	1C2B	..J.....)(...+.
2AB0	CF29	F1E3	0169	2AC5	3DEE				.).....*...=.....O..
2AC0	BB47	D843	C9CD	072A	CAF8	275F	237E	2366	.G.C...*...'.#.#.
2AD0	6FE5	1946	72E3	C57E	CD65	0EC1	E170	C9EB	...F.....
2AE0	CF29	C1D1	C543	C9FE	7AC2	9719	C3D9	41CD	.)...C.....A.
2AF0	1F2B	3294	40CD	9340	C3F8	27CD	0E2B	C396	..+2.@..@...'+..
2B00	40D7	CD37	23E5	CD7F	0AEB	E17A	B7C9	CD1C	@..7#.....
2B10	2B32	9440	3297	40CF	2C18	01D7	CD37	23CD	+2.@2.@...7#.
2B20	052B	C24A	1E2B	D77B	C93E	0132	9C40	C1CD	..+J.+...>.2.@..
2B30	101B	C521	FFFF	22A2	40E1	D14E	2346	2378	...!..".@..N#F#.
2B40	B1CA	191A	CDDF	41CD	9B1D	C54E	2346	23C5A.....N#F#.
2B50	E3EB	DFC1	DA18	1AE3	E5C5	EB22	EC40	CDAF".@..
2B60	0F3E	20E1	CD2A	03CD	7E2B	2AA7	40CD	752B	..>...*...+*.@...+
2B70	CDFE	2018	BE7E	B7C8	CD2A	0323	18F7	E52A*.#...*
2B80	A740	444D	E116	FF18	0303	15C8	7EB7	2302	..@DM.....#.
2B90	C8F2	892B	FEFB	2008	0B0B	0B0B	1414	1414	...+.....
2BA0	FE95	CC24	0BD6	7FE5	5F21	5016	7EB7	23F2	...\$.!P...#
2BB0	AC2B	1D20	F7E6	7F02	0315	CAD8	287E	23B7	..+.....(.#
2BC0	F2B7	2BE1	18C6	CD10	1BD1	C5C5	CD2C	1B30	..+.....(,.)
2BD0	0554	5DE3	E5DF	D24A	1E21	2919	CDA7	28C1	..T.....J.l)...(.
2BE0	21E8	1AE3	EB2A	F940	1AC2	0313	DF20	F960	!.....*.@.....
2BF0	6922	F940	C9CD	8402	CD37	23E5	CD13	2A3E	..".@.....7#...*>
2CG0	D3CD	6402	CD61	021A	CD64	022A	A440	EB2A*.@.*
2CH0	F940	1A13	CD64	02DF	20F8	CDF8	01E1	C9CD	..@.....
2CI0	9302	7ED6	B228	02AF	012F	23F5	2BD7	3E00(..../#.+.>.
2CJ0	2807	CD37	23CD	132A	1A6F	F1B7	6722	2141	(..7#...*....."!A
2CK0	CC4D	1B2A	2141	EB06	03CD	3502	D6D3	20F7	..M.*!A....5.....
2CL0	10F7	CD35	021C	1D28	03EB	2037	2AA4	4006	...5....(....7*.@.
2CM0	C3CD	3502	5F96	A220	2173	CD6C	197E	B723	..5.....!.....#
2CN0	20ED	CD2C	0210	EA22	F940	2129	19CD	A728	...,"...@!)...(
2CO0	CDF8	012A	A440	E5C3	E81A	21A5	2CCD	A728	...*.@.....!.,...(
2CP0	C318	1A32	3E3C	0603	CDE5	02B7	20F8	10F8	...2><...5.....
2CQ0	CD9C	0218				CD7F	0A7E	C3F8BAD.....
2CR0	27C1	022B	D5CF	2CCD	1C2B	D112	C9CD	3823	'..+...+.....8#
2CS0	CDF4	0ACF	3BEB	2A21	4118	083A	DE40	B728;.*!A...:.@.(
2CT0	CCD1	EBE5	AF32	DE40	BA15	D546	B0CA	4A1E2.@...F..J.
2CU0	234E	2366	6918	1C58	E5CE	027E	23FE	25CA	#N#....X....#.#.
2CV0	172E	FE20	2003	0C10	F2F1	433E	25CD	492EC>#.I.


```

100 '                                     KAPITEL 7
110 '
120 '
130 CLEAR 200.
140 CMD"!" : BEI BETRIEB OHNE DISKETTEN ZEILE ENTFERNEN
! !
150 PRINT
160 ON ERROR GOTO 170
170 RESUME 180
180   CLS:PRINT"   W   4   U   C   H   PROGRAMM   ZUR
ZAHLENKONVERTIERUNG"
190 TR=0
200 PRINT
210 PRINT"           D   -   DEZIMAL NACH BINAER"
220 PRINT"           B   -   BINAER NACH DEZIMAL"
230 PRINT"           HB  -   HEXADEZIMAL NACH BINAER"
240 PRINT"           DH  -   DEZIMAL NACH HEXADEZIMAL"
250 PRINT"           HD  -   HEXADEZIMAL NACH DEZIMAL"
260 PRINT"           SP  -   SPLIT DEZIMAL NACH DEZIMAL"
270 PRINT"           DS  -   DEZIMAL NACH SPLIT HEXADEZIMAL"
280 PRINT"           SD  -   SPLIT HEXADEZIMAL NACH DEZIMAL"
290 PRINT
300 PRINT"SOLL EINE GEWAELHTE BETRIEBSART VERLASSEN WERDEN,
SOMUSS NACH ANZEIGE DES ERGEBNISSES DIE ZIFFER '9'
EINGEGEBEN WEIDEN, SOLL SIE BEIBEHALTEN WERDEN, NUR
'ENTER'"
310 INPUT AA$ : CLS
320 IF AA$="HB" THEN 1210
330 IF AA$="DH" THEN QB$=""
340 IF AA$="DS" THEN AA$="DH" : QB$="DS"
350 IF AA$="SD" THEN 1160
360 IF AA$="HD" THEN 1210
370 IF AA$="B" THEN 520
380 REM   DEZIMAL NACH BINAER UMWANDLUNG
390 CLS : INPUT"DEZIMALZAHL";X : IF X>65535 THEN 390
400 A=INT(X/2):AA=X-2*A:B=INT(A/2):BB=A-2*B:C=INT(B/2):
CC=B-2*C
410 D=INT(C/2):DD=C-2*D:E=INT(D/2):EE=D-2*E:F=INT(E/2):
FF=E-2*F
420 G=INT(F/2):GG=F-2*G:H=INT(G/2):HH=G-2*H:I=INT(H/2):
II=H-2*I
430 J=INT(I/2):JJ=I-2*J:K=INT(J/2):KK=J-2*K:L=INT(K/2):
LL=K-2*L
440 M=INT(L/2):MM=L-2*M:N=INT(M/2):NN=M-2*N:O=INT(N/2):
OO=N-2*O
450 PP=O-2*INT(O/2)
460 Y$=STR$(PP)+STR$(OO)+STR$(NN)+STR$(MM)+STR$(LL)+
STR$(KK)+STR$(JJ)+STR$(II)+STR$(HH)+STR$(GG)+
STR$(FF)+STR$(EE)+STR$(DD)+STR$(CC)+STR$(BB)+STR$(AA)
470 IF AA$="SP" THEN NO=NO+1 : QB$="DS" : GOTO 860
480 IF AA$="DH" AND QB$="" THEN PRINT"HEXADEZIMAL " :
GOTO 860
490 IF QB$="DS" THEN 860
500 PRINT"BINAER ZAHL "Y$:INPUT TR : IF TR=9 THEN 150
ELSE 390
510 REM   BINAER NACH DEZIMAL UMWANDLUNG
520 CLS   INPUT' 16, 9 ODER 4 STELLIGE BINAERZAHL";AA
530 CLS   PRINT AA;"STELLIGE BINAERZAHL";:INPUT X$
540 IF LEN(X$)<>AA THEN 530
550 FORZ=1 TO AA:X=VAL(MID$(X$,Z,1)):IFZ=1 THEN A=X
560 IF Z=2 THEN B=X
570 IF Z=3 THEN C=X

```

```
580 IF Z=4 THEN D=X
590 IF AA=8 THEN 630 ELSE IF AA=16 THEN 630
600 NEXT
610 Y=A*8+B*4+C*2+D
620 PRINT" DEZIMAL "Y:INPUT TR : IF TR=9
    THEN 150ELSE330
630 IF Z=5 THEN E=X
640 IF Z=6 THEN F=X
650 IF Z=7 THEN C=X
660 IF Z=8 THEN H=X
670 IF AA=16 THEN 710
680 NEXT
690 Y=A*128+B*64+C*32+D*16+E*8+F*4+C*2+H
700 PRINT" DEZIMAL "Y:INPUT TR :IF TR=9
    THEN 150 ELSE530
710 IF Z=9 THEN I=X
720 IF Z=10 THEN J=X
730 IF Z=11 THEN K=X
740 IF Z=12 THEN L=X
750 IF Z=13 THEN M=X
760 IF Z=14 THEN N=X
770 IF Z=15 THEN O=X
780 IF Z=16 THEN P=X
790 NEXT
800 Y=A*32768+B*16384+C*8192+D*4096+E*2048+F*1024+C*512+H*
    256+I*128+J*64+K*32+L*16+M*8+N*4+O*2+P
810 PRINT"DEZIMAL "Y: INPUT TR:IF TR=9 THEN 150
820 IF AA$="SP" THEN 390
830 IF AA$="B" THEN 530
840 IF AA$="HD" THEN 1200
850 IF AA$="SD" THEN 1160
860 DH$=MID$(Y$,1,8) : GOSUB 910
870 DH$=MID$(Y$,9,8) : GOSUB 910
880 DH$=MID$(Y$,17,8) : GOSUB 910
890 DH$=MID$(Y$,25,8) : GOSUB 910
900 PRINT:INPUT TR:IFTR=9THEN150ELSE 390
910 DH=VAL(DH$) : IF DH=0 THEN DH$="0"
920 IF DH=1 THEN DH$="1"
930 IF DH=10 THEN DH$="2"
940 IF DH=11 THEN DH$="3"
950 IF DH=100 THEN DH$="4"
960 IF DH=101 THEN DH$="5"
970 IF DH=110 THEN DH$="6"
980 IF DH=111 THEN DH$="7"
990 IF DH=1000 THEN DH$="8"
1000 IF DH=1001 THEN DH$="9"
1010 IF DH=1010 THEN DH$="A"
1020 IF DH=1011 THEN DH$="B"
1030 IF DH=1100 THEN DH$="C"
1040 IF DH=1101 THEN DH$="D"
1050 IF DH=1110 THEN DH$="E"
1060 IF DH=1111 THEN DH$="F"
1070 IF QB$="DS" THEN QQ=QQ+1
1080 IF QQ=1 THEN RR$=DH$ : RETURN
1090 IF QQ=2 THEN SS$=DH$ : RETURN
1100 IF QQ=3 THEN TT$=DH$ : RETURN
1110 IF QQ=4 THEN UU$=DH$
1120 IF QB$="DS"AND AA$="DH"THEN PRINT"SPLIT HEX. ";
    TT$,UU$,RR$,SS$:QQ=0 : INPUT TR:IFTR=9THEN150 :LSE390
1130 IF QB$="DS" AND AA$="SP" AND NO=1 THEN
    FF$=TT$+UU$:QQ=0:GOTO 390
1140 IF QB$="DE" AND AA$="SP" AND NO=2 THEN SS$=TT$+UU$:
```

```
X$=SS$+FF$ NO=0:QQ=0:PRINT"HEXADEZIMAL ";X$:GOTO 1220
1150 PRINT DH$; RETURN
1160 REM SPLIT HEXADEZIMAL NACH DEZIMAL UMWANDLUNG
1170 CLS:INPUT"SPLIT HEX ZAHL ";SH$
1180 S1$=LEFT$(SH$,2):S2$=RIGHT$(SH$,2)
1190 X$="" X$=S1$+S2$:GOTO 1220
1200 REM HEX NACH DEZIMAL UND HEX NACH BINAER UMWANDLUNG
1210 CLS X$="" INPUT"HEXADEZIMALZAHL ";X$
1220 IF LEN(X$)<>4 THEN 1210
1230 FOR XX=1 TO 4 A$=MID$(X$,XX,1)
1240 IF A$="0" THEN A$="0000"
1250 IF A$="1" THEN A$="0001"
1260 IF A$="2" THEN A$="0010"
1270 IF A$="3" THEN A$="0011"
1280 IF A$="4" THEN A$="0100"
1290 IF A$="5" THEN A$="0101"
1300 IF A$="6" THEN A$="0110"
1310 IF A$="7" THEN A$="0111"
1320 IF A$="8" THEN A$="1000"
1330 IF A$="9" THEN A$="1001"
1340 IF A$="A" THEN A$="1010"
1350 IF A$="B" THEN A$="1011"
1360 IF A$="C" THEN A$="1100"
1370 IF A$="D" THEN A$="1101"
1380 IF A$="E" THEN A$="1110"
1390 IF A$="F" THEN A$="1111"
1400 NN=NN+1
1410 IF NN=1 THEN 1420 ELSE IF NN=2 THEN 1430 ELSE IF NN=3
THEN 1440 ELSE IF NN=4 THEN 1450
1420 BB$=A$ NEXT XX GOTO 1230
1430 CC$=A$ NEXT XX : GOTO 1230
1440 DD$=A$ NEXT XX : GOTO 1230
1450 EE$=A$
1460 X$=BB$+CC$+DD$+EE$ : NN=0
1470 IF AA$="HB" THEN PRINT "BINAER ";X$:INPUT
TR IF TR=9 THEN 150 ELSE CLS : GOTO 1210
1480 AA=16 GOTO 550
1490 END
1500 '
1510 'ANMERKUNG: PROGRAMM BELEGT 3919 BYTES
```

KAPITEL 8

```

010 ;PROGRAMM ZUM DRUCKEN ALLER NULLEN MIT SCHRAEGSTRICH
011 ;FUER IBM-DRUCKER UND ANDERE
012 ;
013 ;MIT AUTOMATISCHEM WAGENRUECKLAUF NACH 64 ZEICHEN BEI
014 ;LLIST
015 ;
016 ORC 7F15H ;=32533 DEZIMAL
017 COUNT: DEFB 00 ;ZEICHENZAEHLER
018 START: PUSH AF ;RETTE DEN REGISTER AUF DEN
019 PUSH BC ;STAPELSPICHER
020 PUSH DE
021 PUSH HL
022 LD A,C ;NAECHSTES ZEICHEN AUS C-REG.
023 CP 0DH ;WAGENRUECKLAUF
024 JR Z,NOINC ;ZAEHLER NICHT INKREMENTIEREN
025 LD A,(COUNT) ;ZAEHLERSTAND HOLEN
026 CP 40H ;BEI 64 DEZ. ZEILENENDE
027 JR Z,CARRET ;WENN JA, WAGENRUECKL.
028 FINIS: LD A,(COUNT) ;ZEICHENZAEHLER
029 INC A ;INKREMENTIEREN
030 LD (COUNT),A
031 LD A,C
032 CP 30H ;30H=ASCII '0'
033 CALL Z,ZERO ;WENN JA ZERO-ROUTINE
034 ELFIN: POP HL ;ALTE REGISTERINHALTE WIEDER
035 POP DE ;VOM STAPELSPICHER ZURUECK-
036 POP BC ;HOLEN
037 POP AF
038 JP 058DH ;SPRUNG ZU ROM-DRUCKER-ROUTINE
039 NOINC: LD A,00H ;ZEICHENZAEHLER RUECKSETZEN
040 LD (COUNT),A
041 JR ELFIN ;RUECKSPRUNG
042 RESET: LD A,00H ;ZEICHENZAEHLER RUECKSETZEN
043 LD (COUNT),A
044 JR FINIS ;FERTIG
045 CARRET: CALL TEST ;TEST, OB DRUCKER BEREIT
046 LD A,0DH ;0DH=ASCII 'WAGENRUECKLAUF'
047 LD (37E8H),A ;AUSGEBEN AUF DRUECKER-PORT
048 RESET ;WEITER MIT RUECKSETZEN DES
049 ;ZEICHENZAEHLERS
050 TEST: LD A,(37E8H) ;DRUECKER-PORT
051 ;ABFRAGEN
052 BIT 7,A ;DRUCKER BEREIT ?
053 JR NZ,TEST ;NEIN, DANN ZURUECK
054 RET
055 ZERO: CALL TEST ;TEST, OB DRUCKER BEREIT
056 LD A,2FH ;2FH=ASCII '/'
057 LD (37E8H),A ; '/' DRUCKEN
058 CALL TEST ;TEST, OB DRUCKER BEREIT
059 CALL DELAY ;VERZOECERUNG 20 MS
060 LD A,08H ;08H=ASCII RUECKSCHRITT
061 LD (37E8H),A ;DRUCKEN
062 DELAY: LD C,0AH ;VERZOEGERUNGSZEIT SETZEN
063 DELAY1: LD B,0
064 DELAY2: DJNZ DELAY2
065 DEC C
066 JP NZ,DELAY1
067 RET
068
069 ORC 4026H ;ADRESSE IN DRUCKERKONTROLL-
070 DEFW START ;BLOCK AENDERN AUF ANFANGS-
071 ;ADRESSE DIESES PROGRAMMS
072 ORC COUNT ;ZEICHENZAEHLER AM ANFANG
073 DEFW 0 ;AUF NULL SETZEN
074 END COUNT

```

K A P I T E L 9

SELBSTTEST-FRAGEN

Die folgenden Seiten mit Selbsttest Fragen umfassen eine Anzahl der wichtigsten Punkte aus den vorangehenden Kapiteln und den Demonstrationsprogrammen. Wenn Sie die Logik und den Programmablauf sowie den Inhalt der Programme und der Texte verstanden haben, sollten Sie keine Schwierigkeit mit der Beantwortung der Fragen haben. Wenn Ihnen die Antwort auf eine Frage nicht möglich ist, sollten Sie das betreffende Kapitel und die entsprechenden Programme nochmals durcharbeiten und mit den Programmen arbeiten, bis Sie -sofern die Zeit vorhanden- langsam von allem Wissen durchdrungen sind. Sollte auch das nicht helfen, versuchen sie ob es nützt das Handbuch unter Ihr Kopfkissen zu legen und darauf zu schlafen.

FRAGEN ZU KAPITEL 1:

1. Was besagt der Ausdruck 'das MSB maskieren' ?
2. Zwischen welchen dezimalen und hexadezimalen Adressen sind im ROM die Level-II Funktionsnamen gespeichert ?
3. Warum liegen auch die BASIC-Disketten-Funktionsnamen und Sprungadressen im Level-II BASIC ROM ?
4. Wie würden Sie das MSB des Inhaltes einer beliebigen Speicherstelle mit einem einfachen BASIC-Programm maskieren ?
5. Wieviele unterschiedliche Speicherbereichsgruppen belegen die BASIC-Funktions-CALL Adressen ?
6. Wie heißen die beiden klugen Köpfe bei Microsoft, die Level-II und Disk-BASIC geschrieben haben (zum Glück stammen TRS-DOS 2.1 und 2.2 nicht von denen) ?
7. Warum werden die BASIC-Funktions-CALL Adressen in der Form 'erst LSB, dann MSB' gespeichert ?
8. Welchen Inhalt (dezimal) zeigen PEEK (5666) und PEEK (5667) an und welche Level-II-CALL Adresse stellen sie dar ?
9. Welche Speicherplätze in hexadezimal und dezimal entsprechen den folgenden PEEK-Werten:
a. 194-30, b. 15-42, c. 255-255, d. 0-255 ?

FRAGEN ZU KAPITEL 2:

1. Wieviel schneller ist im Mittel ein gutgeschriebenes Assemblerprogramm gegenüber einem BASIC Programm und welcher Speicherplatz ist erforderlich ?
2. Wie nennen wir die beiden Speicherbereiche für arithmetische Operationen im RAM ?
3. Geben Sie den Nummerntyp (NT) für folgende Zahlen an :
a. 10000 b. 33000 c. 1.011 d. 1.1111111
4. Wo wird der Kenner für den Nummerntyp (NT) im RAM gespeichert ?
5. Welche Variablenarten werden durch folgende Nummerntypen dargestellt ?
a. NT=2 b. NT=3 c. NT=4 d. NT=8
6. Was bewirkt die Zeile 290 in Abb. 7 ?
7. Was bewirkt die Zeile 300 in Abb. 7 ?
8. Was bewirken die Zeilen 27 und 28 in Abb. 8 ?
9. Was bewirkt die Zeile 23 in Abb. 8 ?
10. Wieviele signifikante Stellen hat eine doppelt genaue Zahl ?

FRAGEN ZU KAPITEL 3:

1. Verwendet das Level-II ROM die zweite Registerbank des Z-80 (AF', BD', DE' und HL') ?
2. Welche Adresse hat der CALL, mit dem man (B) Bytes von einem Adressbereich in einen anderen verschieben kann ?
((B) = Anzahl der Bytes steht im B-Register)
3. Wiviele Speicherplätze belegen jeweils der ACCUM und der CDBL-Speicher und unter welchen Adressen liegen diese ?
4. Wie lautet der Nummerntyp eines ASCII-String ?
5. Welcher CALL wandelt den Inhalt von ACCUM in einen ASCII-String um ?
6. Welche Funktion erfüllt die Zeile 410 in Abb. 10 und warum ?
7. Was ist der Unterschied zwischen RANDOM und RND ?
8. Was ist der Unterschied zwischen CINT und INT ?
9. Wie wird die Adresse der Umwandlungsroutine im Programm in Abb. 10 gefunden ?
10. Wie wird im obigen Programm nach einer Routine die Rücksprungadresse gefunden ?

FRAGEN ZU KAPITEL 4:

1. Welcher Wert ergibt sich für PEEK (14464) wenn eine Shifttaste gedrückt wird; welcher Wert ergibt sich, wenn beide Tasten gedrückt werden ?
2. Welcher Wert ergibt sich für PEEK (14338) wenn die Tasten H, I, J, K, L, M, N und O zusammen gedrückt werden ?
3. Was ist der wesentliche Unterschied zwischen CALL 002BH und CALL 0049H ?
4. Welche Information ist unter Adresse 409CH gespeichert?
5. Was ist die einfachste Methode, eine ganze oder einfach genaue Zahl im RAM zu speichern (ACCUM und CDBL-Speicher ausgenommen) ?
6. Wo werden die Ergebnisse von Zahlenvergleichen gespeichert und mit welchen Werten ?
7. Welchen Wert enthält das A-Register, nachdem ein CALL 0994H aufgerufen wurde und der ACCUM den Wert -1.9999999999 enthält ?
8. Welche CALL-Adressen werden verwendet, um einen beliebigen Wert in eine Ganzzahl, eine einfach oder doppelt genaue Zahl umzuwandeln ?
9. Welcher Wert wird unter NT bei einer Stringvariablen gespeichert ?
10. Welchen Wert enthält der Speicherplatz 37E8H wenn der Drucker bereit ist. In welchem Speicher oder Register muß das nächste zu druckende Zeichen enthalten sein ?
11. Schreiben Sie ein kurzes Assemblerprogramm, um auf dem Bildschirm einen Text beliebiger Länge ausgeben zu können; benutzen Sie die Routine CALL 28A7H.
Anm.: Der CALL 28A7H gibt ohne DOS-BASIC einen Text von maximal 63 Zeichen aus. Durch Zusammenfügen der Texte mit zusätzlichen DEFM-Anweisungen kann die Textlänge beliebig vergrößert werden und ist nur noch durch die Größe des verfügbaren RAM-Speichers begrenzt.

FRAGEN ZU KAPITEL 5

Ergänzen Sie in untenstehender Tabelle die CALL-Adressen in dezimal; benutzen Sie dazu das Konvertierungsprogramm aus Kapitel 7.

BASIC FUNKTION	CALL ADRESSE	BASIC FUNKTION	CALL ADRESSE
ABS	ASC
ATN	AUTO
CDBL	CHR\$
CINT	CLEAR
CLOAD	CLS
CONT	COS
CSAVE	CSNG
DATA	DEFDBL
DEFINT	DEFSNG
DEFSTR	DELETE
DIM	EDIT
ELSE	END
ERR	ERL
ERROR	EXP
FIX	FOR
FRE	GOSUB
GOTO	IF
INKEY\$	INP
INPUT	INSTR
INT	LEFT\$
LEN	LIST
LOG	LLIST
LPRINT	MEM
MID\$	NEW
NEXT	NOT
ON	OUT
PEEK	POINT
POKE	POS
PRINT	RANDOM
READ	REM
RESET	RESTORE
RESUME	RIGHT\$
RND	RUN
SET	SGN
SIN	SQR
STOP	STR\$
STRING\$	SYSTEM
TAN	TROFF
TRON	USR
VARPTR	VAL

FRAGEN ZU DEN KAPITELN 6, 7 und 8

1. a) Wieviel Byte enthält jeder in Kapitel 6 dargestellte Speicherplatz ?
b) Wieviel Bytes enthält eine Zeile und
c) wieviel Bytes enthält eine Zeile ?
2. Welches Zeichen enthält der Speicherplatz 01 Hexadezimal und in ASCII ?
3. Tragen Sie über der ersten Zeile des Kapitel das LSB des jeweiligen Speicherplatzes ein.
4. Wird die Zeile 510 im Zahlen-Konvertierungs (Kapitel 7) nur bei Disk-BASIC benötigt und ansonsten gelöscht werden ?
5. Wieviele grundlegende Zahlenkonvertierungen Programm in Kapitel 7 ausführen ?
6. Wie wird in diesem Programm die Umwandlung "Dezimal" nach Dezimal durchgeführt ?
7. Schreiben Sie das Programm in Kapitel 8 so in den Zeilen 18 bis 21 und 34 bis 37 die B EX AF,AF' und EXX anstelle der Stack-Operat verwenden.
8. Ändern Sie in obigem Programm die Zeile 52 Sie den CP Befehl anstelle des Befehls BIT verwenden; was ist der Unterschied ?
9. Warum sind die Zeilen 62-66 bei IBM-Drucker erforderlich ?
10. Wie würden Sie das Programm ändern, wenn Si Drucker mit einer Zeilenlänge von 130 Zeich wollen ?

K A P I T E L 10

LOESUNGEN ZU DEN SELBSTTEST FRAGEN

Anmerkung: erfahrene Assembler-Programmierer werden gewiß einige der Fragen trivial gefunden haben, aber für die weniger erfahrenen Leser werden sie sicherlich zum Teil auch nützlich gewesen sein und es ist nun einmal nicht möglich, es jedem recht zu machen, darum bitten wir Sie um Ihr Verständnis.

ANTWORTEN KAPITEL 1

1. Angenommen, $\text{PEEK}(5712) = 192$ dezimal. 5712 entspricht der Adresse 1650H und 197 dezimal ist 11000101 dual. Das am weitesten links stehende Bit ist 1 und soll maskiert (d.h. gelöscht) werden. Dann erhalten wir 01000101 dual was 69 dezimal oder dem ASCII-Zeichen 'E' entspricht. Da 10000000 dual gleich 128 dezimal ist, kann man die Maskierung durch einfache Subtraktion von 128 vom erhaltenen PEEK-Wert erreichen.
2. Von 5712 dezimal/1650H bis 6172 dezimal/181CH.
3. Sie werden alle über das ROM (das direkt keine Disk-Funktionen enthält) angesprungen. Wenn keine Diskette angeschlossen ist erfolgt ein Sprung zur L3-ERROR-Meldung.
4. Subtrahieren Sie 128 vom PEEK-Wert.
5. Zwei.
6. Von 6178 dez./1822H bis 6297 dez./1899H und von 5642 dez./160AH bis 5711 dez./164FH.
7. Aus Gründen der Vereinfachung und um Kompatibilität zwischen dem 8080 und Z-80 Mikroprozessor zu erhalten, die beide von Dr. Federico Faggin entwickelt wurden.
8. 189 und 21 dezimal.
9. A. 7874 dez./1EC2H B. 10767 dez./2AOFH
C. 65535 dez./FFFFH D. 65280 dez./FF00H

ANTWORTEN KAPITEL 2

1. A. 300 bis 350 mal schneller; B. 1/10 des Speicherbedarfs
2. ACCUM und CDBL-Speicher
3. A. 2; B. 4; C. 4; D. 8;
4. NT wird unter Adresse 40AFH gespeichert.
5. 2 = ganzzahlig; 4 = einfach genau; 8 = doppelt genau

6. CALL OE6CH wandelt ein String, das einen numerischen Wert darstellt in den mindest erforderlichen Nummerntyp um und speichert den numerischen Wert im ACCUM sowie den Nummerntyp (NT) in Speicheradresse 40AFH.
7. CALL OA7FH = CINT ändert jeden numerischen Wert im Bereich von +32767 bis -32768 in eine ganze Zahl um.
8. Verwendet den Stapelspeicher im RAM um den einfach genauen Wert abzuspeichern, der in den Registern BC und DE enthalten ist, damit BC und DE in den darauffolgenden Befehlen verwendet werden können.
9. Siehe Antwort zu Frage 6. CALL OE65H ist ähnlich, es wird jedoch eine Zahl mit doppelter Genauigkeit im ACCUM gespeichert und NT wird auf 8 gesetzt.
10. Diese Antwort mag Sie vielleicht überraschen: nur soviele Stellen, wie in einem Argument enthalten sind, maximal jedoch 17 Stellen. Wenn das Argument nur 10 Stellen enthält, sind die restlichen 7 Stellen ohne Bedeutung. Siehe auch Radio Shack "Microcomputer Newsletter, Oktober 1979" mit einer ausgezeichneten Erläuterung zu diesem Thema.

ANTWORTEN KAPITEL 3

1. Nein, das dieses BASIC bereits für 8080-Rechner geschrieben wurde und der 8080 entsprechende Befehle ja nicht kennt. Dadurch ist es möglich, daß Sie über diese Register frei verfügen können und so im Vergleich zu PUSH und POP Befehlen jeweils 4 und mehr Bytes an Befehlen sparen können.
2. CALL O9D7H/2519 dezimal.
3. Jeweils acht.
4. ADD A,48 ;konvertiert nach ASCII
5. CALL OFBDH konvertiert den Inhalt von ACCUM in ein String (Adresse in HL) und setzt NT (40AFH) auf 3.
6. Wandelt alle Zahlen im zulässigen Bereich für CINT in ganze Zahlen um und dient somit zur Erkennung fehlerhafter Eingaben, da alle CALL-Adressen ganzzahlige Werte sind.
7. RANDOM initialisiert den Zufallszahlengenerator; RND mit NT=2 erzeugt eine Zufallszahl zwischen und dem ganzzahligen Wert in ACCUM. RND mit NT=4 erzeugt eine einfach genaue Zufallszahl zwischen Null und der Zahl, die in ACCUM steht (diese muß kleiner oder gleich 1 sein).
8. INT liefert eine gerundete Zahl ohne Stellen hinter dem Komma für JEDE beliebig genaue Zahl; CINT wandelt eine beliebige Zahl mit bis zu sieben Stellen in eine ganze Zahl im Bereich von +32767 bis -32768 um.

9. Zeile 590 enthält einen RET-Befehl, der den obersten Wert vom Stapelspeicher in den Programmzähler überträgt und somit einen Sprung auf die entsprechende Adresse verursacht.

10. Da jede aufgeführte Routine mit einem RET-Befehl endet, wird die Rücksprungadresse vom Stapelspeicher geholt, eine sehr nützliche Funktion.

ANTWORTEN KAPITEL 4

1. A. 1; B. 1;

2. 255 dezimal

3. CALL 002BH prüft die Tastatur EINMAL ab, CALL 0049H hingegen tastet solange ab, bis ein Wert ungleich Null auftritt, also eine Taste gedrückt wird.

4. Unter der Adresse 409DH wird ein Wert gespeichert, der angibt, wohin eine Ausgabe von CALL 1BB3H (und andere) übertragen werden soll. Anfangs auf Null gesetzt (Bildschirmausgabe); +1 = Drucker; -1 = Kassette.

5. A. PUSH DE oder PUSH HL B. PUSH BC und PUSH DE
C. Uebertragen von 8 Bytes von ACCUM in einen Speicherbereich mit CALL 09D6H oder 09D7H.

6. A. im A-Register B. Null C. +1 D. 255/OFFH

7. 255 dezimal = 11111111 dual = OFFH

8. A. CALL 0B37H = INT; B. CALL 0AB1H = CSNG
C. CALL 0ABDH = CDBL.

9. 3 dezimal

10. A. 63 dezimal = ASCII '?' B. 37E8H

```
11. OTEXT    EQU        7DOOH        ;PROGRAMM ZUR TEXTAUSGABE
      ORG        OTEXT        ;7DOOH = 32000 DEZIMAL
      LD         HL,TEXT        ;ANFANGSADRESSE TEXTSPEICHER
      CALL       28A7H        ;UNTERPROGRAMM TEXTANZEIGE
TEXT        DEFM       'DAS IST EIN TEST-TEXT'
      DEFB       0            ;TEXT-ENDE BEGRENZUNG
      END        OTEXT
;ANMERKUNG: SIE KOENNEN BIS ZU 240 BYTES TEXT PRO
;CALL 28A7H ZUSAMMENFUEGEN.
```

ANTWORTEN KAPITEL 5

ABS	2423	ASC	10767
ATN	5565	AUTO	8200
CDBL	2779	CHR\$	10783
CINT	2687	CLEAR	7802
CLOAD	11295	CLS	457
CONT	7652	COS	5441
CSAVE	11253	CSNG	2737
DATA	7941	DEFDBL	7689
DEFINT	7683	DEFSNG	7686
DEFSTR	7680	DELETE	11206
DIM	9736	EDIT	11872
ELSE	7943	END	7698
ERR	9437	ERL	9423
ERROR	8180	EXP	5177
FIX	2854	FOR	7329
FRE	10196	GOSUB	7857
GOTO	7874	IF	8249
INKEY \$	413	INP	10991
INPUT	8602	INSTR	16787
INT	2871	LEFT\$	10849
LEN	10755	LIST	11054
LOG	2057	LLIST	11049
LPRINT	8295	MEM	10185
MID\$	10906	NEW	6985
NEXT	8886	NOT	9668
ON	8044	OUT	11003
PEEK	11434	POINT	307
POKE	11441	POS	10229
PRINT	8303	RANDOM	467
READ	8687	REM	7943
RESET	312	RESTORE	7569
RESUME	8111	RIGHT\$	10897
RND	5321	RUN	7843
SET	309	SGN	2442
SIN	5447	SQR	5095
STOP	7593	STR\$	10294
STRING\$	10799	SYSTEM	690
TAN	5544	TROFF	7672
TRON	7671	USR	10238
VARPTR	9461	VAL	10949

ANTWORTEN KAPITEL 6, 7 und 8

1. A. eins B. 16 Bytes / Zeile C. 48 Zeilen x 16 = 768

2. A. 56H = 86 dez. B. ASCII 'V'

3. O 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000 F3AF C374 06C3 0040 C300 40E1 E9C3 9F06ß..ß.....

4.A. Ja, um den Diskettenbetrieb etwas schneller zu machen
B. Ja, sollte entfernt werden, wenn kein Diskettenbetrieb

5. Vier grundlegende Umwandlungen, die übrigen sind lediglich Anwendungen.

- dezimal nach dual Zeilen 740 - 840
- dual nach dezimal Zeilen 860 - 1150
- dual nach hexadezimal Zeilen 1200 - 1490
- hexadezimal nach dual Zeilen 1550 - 1810

6. Die erste Dezimalzahl wird in eine Hexadezimalzahl umgewandelt und in Zeile 1130 unter FF\$ gespeichert. Dann wird die zweite Zahl nach hexadezimal umgewandelt und in Zeile 1140 unter SS\$ gespeichert. FF\$ und SS\$ werden dann umgedreht und in Zeile 1140 zu X\$ zusammengefügt. Die richtige Hexadezimalzahl, nämlich X\$ wird dann in Zeile 1140 auf dem Bildschirm angezeigt. Diese Zahl wird zuerst nach dual und danach in dezimal umgewandelt und angezeigt. Obwohl dieses Verfahren recht langsam ist, schlägt es doch jeden Hewlett-Packard Taschenrechner.

```
7. 018      START  EX      AF,AF'      ;VERTAUSCHEN DER REGISTER
                                ;AF UND AF'
    019              LD      A,C          ;NAECHST. ZEICHEN NACH A
    020              EXX                    ;AUSTAUSCH BC MIT BC',
                                ;DE MIT DE' UND HL MIT HL'
```

Zeilen 021 und 022 löschen

```
    034      ELFIN  EX      AF,AF'      ;VERTAUSCHEN DER REGISTER
                                ;AF UND AF'
    035              EXX                    ;VERTAUSCHEN DER UEBRIGEN
                                ;REGISTER
```

Zeilen 036 und 037 löschen

ANM.: Diese kleine Uebung sparte 4 Zeilen und vier weitere Bytes ein.

```
8. 052              CP      3FH          ;IST ES ASCII ? = 63 DEZ.
```

9. Durch eine Zeichenrückschaltung werden fast alle IBM-Drucker in kräftige Erschütterung versetzt. Sie sollten jedoch ggf. diese Zeilen entfernen und testen, wie sich Ihr Drucker verhält und sie nur einfügen, wenn erforderlich.

10. Aendern Sie die Zeile 026:

```
    026              CP      082H        ;IST ES 130 DEZ. = ZEILEN-
                                ;ENDE (EOL)
```

LITERATUR:

Wenn man Wertigkeiten für Literatur angeben würde, wie sie bei Warentests verwendet werden, also z.B.: SEHR GUT, MAESSIG, MANGELHAFT und VOM KAUF ABZURATEN, so würde kein verfügbares Buch die Note SEHR GUT erhalten. Leider ist kein Buch für den TRS-80 Assembler-Programmierer auf dem Markt erhältlich, das eine bessere Note als MAESSIG erhalten kann, mit zwei Ausnahmen:

SEHR GUT: Hildebrand, Software Technical Manual
Houston Micro Computer Technologies
5313 Bissonnet Street
Bellaire, Texas 77401

GUT: Lance Leventhal, Z-80 Assembly Language Programming
Adam Osborne & Associates
630 Bancroft Way
Berkeley, California 94710

Beide Bücher setzen allerdings unserer Meinung nach beim Leser eine mindestens zweijährige Erfahrung in der Programmierung des 8080A voraus.

Leider ist nichts wirklich gutes für den Anfänger im Bereich der Assembler-Programmierung verfügbar, Radio-Shacks "TRS-80 Assembly Language Programming" eingeschlossen.

Die folgenden Bücher haben von uns die Note MAESSIG erhalten:

Z-80 Programming For Logic Design

The Z-80 Microcomputer Handbook

Z-80 Software Gourmet Guide & Cookbook

Worauf die über 200.000 TRS-80 Besitzer warten, dürfte ein neues Buch von Dr. David Lien sein, dem Verfasser des "User's Manual for Level I". Dieses Buch ist gewiß eines der besten Lehrbücher für Anfänger in BASIC und das im Herbst 1979 erschienene Buch "Learning Level II" ist ebenso hervorragend. Falls Dave ein entsprechendes Buch über Disk-BASIC und Assembler-Programmierung herausbringen würde, hätte er sicherlich einen weiteren Bestseller.

Wichtige Informationen:

Die nächsten Bände dieser Reihe enthalten weitere nützliche Programme und Tips.

In Band 2 sind unter anderem enthalten:

- Disassemblierter Ausdruck des Level II - ROM
- Beschreibung der Speichermethoden von Variablen im RAM
- Programm zum Bildschirmausdruck (wie JKL in NEWDOS)
- Speichern von Bildschirminhalten im RAM für spätere Verwendung
- Aufteilung des Bildschirms in zwei getrennte Anzeigebereiche
- Zusammenfassung aller wichtigen Level II CALL-Adressen
- Morse Sende- und Empfangsprogramm
- Programm zur Berechnung von Satellitenpositionen
- Selbsttest Fragen und Antworten

Im Band 3 sind unter anderem enthalten:

- Verschiedene schnelle Disassembler-Programme in BASIC
- Verwendung von Interrupts beim TRS-80 mit Demoprogrammen
- Verschiedene Schaltungen und Programme für Ein-Ausgangs-Ports
- Analog/Digital- und Digital/Analog-Wandlung, Schaltungen und Programme
- Schnelles Morse Sende- und Empfangsprogramm
- Funkfern schreiben mit dem TRS-80 von A-Z
- Anschluß des TRS-80 an öffentliche Computer-Informations-Netze
- Selbsttest Fragen und Antworten

Band 4 ist in Vorbereitung und wird hauptsächlich Datenübertragung behandeln.

Informationen über das neue TRS-80 Modell III liegen bereits vor und werden bei Verfügbarkeit des Gerätes mit aufgenommen. Interessenten können diese Informationen bereits jetzt gegen frankierten Rückumschlag (DIN A 4) anfordern.

Damit Sie sich die Mühe sparen können, die in den Handbüchern abgedruckten Programme einzutippen, sind die Programme aus Band 1 und 2 auf Diskette oder Kassette lieferbar, siehe untenstehenden Gutschein. Die Programme aus Band 3 werden ebenfalls in dieser Form lieferbar sein.

Weitere Bände können Sie entweder bei Ihrem Händler oder direkt beim deutschen Verleger bestellen.