

MOD III ROM COMMENTED

©1981 **ssm** SOFT SECTOR MARKTING,
INCORPORATED

*Not a rehash
of old information*

**but an explanation
of ROMs
in the latest machine
from Tandy**

*Most every location of the
14K ROMs listed, with comments.*

MOD III ROM COMMENTED

COPYRIGHT © 1981 Soft Sector Marketing, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means electronic, mechanical, or photo copying, recording, or otherwise without written permission from the author. Printed in the United States of America.

Soft Sector Marketing, Inc. Is in no way attempting to infringe upon the proprietary rights of the Tandy Corporation by this publication. The express purpose of the MOD III ROM COMMENTED is to aid users of MOD III microcomputers in using and understanding their machines.

IMPORTANT NOTICE

Soft Sector Marketing, Inc. assumes no liability with the respect to the use of this publication no or any damages resulting from the use of any information contained herein. MOD III ROM COMMENTED is sold in an "as is" condition and is not represented as being error free.

First Printing – 3-1981

Soft Sector Marketing, Inc.

6250 Middlebelt Road
Garden City, Michigan 48135
(313) 425-4020

N O T I C E

Soft Sector Marketing Inc. has worked hard to provide you with as much information as possible about the MOD III Rom.

Much of the code has been left for you to examine but a small portion has been removed to protect the copyright of TANDY corp.

MOD III COMMENTED (c) 1981 Soft Sector Marketing Inc.

MOD III Commented was written under the direction of Victor Andrews of Soft Sector Marketing Inc. The following persons and software were used to create this book.

Decoding by	Rick Papo
Data Entry	Pactolus Group
Editing by	Brian Stone
	Vivtor Andrews
Software used	RSM (by small system software) patched with REM SOFT patch Disassembler (by Apparat inc) Lazy writer (by ABC SALES)

MAIN ROM

<u>ADDR</u>				
0000	F3	DI		DISABLE INTERRUPTS!
0001	AF	XOR	A	CLEAR STATUS
0002	C31530	JP	3015H	GO TO BOOTSTRAP
0005	C30040	JP	4000H	UNUSED
0008	C30040	JP	4000H	RST 08H VECTOR
000B	E1	POP	HL	UNUSED
000C	E9	JP	(HL)	RETURN ROUTINE
000D	C31230	JP	3012H	WARM BOOT VECTOR
0010	C30340	JP	4003H	RST 010H VECTOR.
0013	C5	PUSH	BC	SAVE BC
0014	0601	LD	B,01H	SELECT READ BYTE
0016	182E	JR	0046H	GO TO I/O DISPATCHER.
0018	C30640	JP	4006H	RST 018H VECTOR.
001B	C5	PUSH	BC	SAVE BC
001C	0602	LD	B,02H	SELECT WRITE BYTE
001E	1826	JR	0046H	GO TO I/O DISPATCHER
0020	C30940	JP	4009H	EST 020H VECTOR
0023	C5	PUSH	BC	SAVE BC
0024	0604	LD	B,04H	
0026	181E	JR	0046H	GO TO I/O DISPATCHER
0028	C30C40	JP	400CH	RST 028H VECTOR
002B	111540	LD	DE,4015H	DE => KEYBOARD DCB
002E	18E3	JR	0013H	INPUT BYTE
0030	C30F40	JP	400FH	RST 030H VECTOR
0033	111D40	LD	DE,401DH	DE => VIDEO DCB
0036	18E3	JR	001BH	OUTPUT BYTE
0038	C31240	JP	4012H	RST 038H VECTOR
003B	112540	LD	DE,4025H	DE => PRINTER DCB
003E	18DB	JR	001BH	OUTPUT BYTE
0040	C3D905	JP	05D9H	INPUT TEXT FROM KEYBOARD
0043	C9	RET		UNUSED CALL
0044	00	NOP		(FUTURE EXPANSION)
0045	00	NOP		(FUTURE EXPANSION)
0046	C37406	JP	0674H	GO TO I/O DISPATCHER
0049	CD2B00	CALL	002BH	CHECK KEYBOARD
004C	B7	OR	A	KEY HIT?
004D	C0	RET	NZ	RETURN IF SO.
004E	18F9	JR	0049H	LOOP IF NOT.
0050	11E541	LD	DE,41E5H	DE => RS-232 INPUT DCB
0053	18BE	JR	0013H	INPUT BYTE
0055	11ED41	LD	DE,41EDH	DE => RS-232 OUTPUT DCB
0058	18C1	JR	001BH	OUTPUT BYTE
005A	11F541	LD	DE,41F5H	DE =>RS-232 CONTROLLER DCB
005D	18BC	JR	001BH	SET UP RS-232
005F	00	NOP		UNUSED
0060	C3FB01	JP	01FBH	GO TO DELAY ROUTINE
0063	20FB	JR	NZ,0060H	CONDITIONAL DELAY.
0065	C9	RET		
0066	C33930	JP	3039H	GO TO NON-MASKABLE INTERRUPT ROUTINE
0069	C35204	JP	0452H	INITIALIZE I/O DRIVERS
006C	111D42	LD	DE,421DH	DE => \$ ROUTINE DCB
006F	18AA	JR	001BH	SET I/O ROUTING
0071	00	NOP		UNUSED
0072	C3CC06	JP	06CCH	

0075	118040	LD	DE,4080H	SET SOME POINTERS
0078	21F718	LD	HL,18F7H	(TAPE BASIC)
007B	012700	LD	BC,0027H	
007E	EDB0	LDIR		
0080	21E542	LD	HL,42E5H	
0083	363A	LD	(HL),3AH	
0085	23	INC	HL	
0086	70	LD	(HL),B	
0087	23	INC	HL	
0088	362C	LD	(HL),2CH	
008A	23	INC	HL	
008B	22A740	LD	(40A7H),HL	
008E	112D01	LD	DE,012DH	DE => L3 ERROR ROUTINE.
0091	061C	LD	B,1CH	SET 28 VECTOR
0093	215241	LD	HL,4152H	FROM 4152H UP.
0096	36C3	LD	(HL),0C3H	JP INSTRUCTION
0098	23	INC	HL	
0099	73	LD	(HL),E	JUMP ADDRESS
009A	23	INC	HL	
009B	72	LD	(HL),D	
009C	23	INC	HL	
009D	10F7	DJNZ	0096H	LOOP UNTIL DONE.
009F	0615	LD	B,15H	SET 15 RETURNS
00A1	36C9	LD	(HL),0C9H	FROM DOS LINKS
00A3	23	INC	HL	
00A4	23	INC	HL	
00A5	23	INC	HL	
00A6	10F9	DJNZ	00A1H	LOOP UNTIL DONE
00A8	21E843	LD	HL,43E8H	ZERO BYTE 43E8H
00AB	70	LD	(HL),B	
00AC	31F842	LD	SP,42F8H	SET STACK
00AF	CD8F1B	CALL	1B8FH	SET NEW STACK
00B2	00	NOP		UNUSED
00B3	00	NOP		UNUSED
00B4	00	NOP		UNUSED
00B5	210501	LD	HL,0105H	PRINT "MEMORY SIZE?";
00B8	CDA728	CALL	28A7H	
00BB	CDB31B	CALL	1BB3H	INPUT RESPONSE
00BE	38F5	JR	C,00B5H	RESTART IF <BREAK>
00C0	D7	RST	10H	SCAN FOR FIRST CHARACTER
00C1	B7	OR	A	NULL ?
00C2	2012	JR	NZ,00D6H	SKIP IF NOT
00C4	214C44	LD	HL,444CH	START OF MEMORY TEST.
00C7	23	INC	HL	NEXT BYTE
00C8	7C	LD	A,H	END OF MEMORY
00C9	B5	OR	L	
00CA	281B	JR	Z,00E7H	SKIP OUT IF SO.
00CC	7E	LD	A,(HL)	GET LOCATION CONTENTS
00CD	47	LD	B,A	SAVE IN B.
00CE	2F	CPL		INVERT CONTENTS
00CF	77	LD	(HL),A	SAVE TEST BYTE.
00D0	BE	CP	(HL)	SEE IF STILL SAME
00D1	70	LD	(HL),B	RESTORE OLD CONTENTS
00D2	28F3	JR	Z,00C7H	TEST NEXT LOCATION IF GOOD
00D4	1811	JR	00E7H	SKIP OUT IF BAD
00D6	CD5A1E	CALL	1E5AH	DECODE RESPONSE
00D9	B7	OR	A	NULL ? (AFTER RESPONSE)

```

00DA C29719    JP      NZ,1997H    SYNTAX ERROR IF NOT
00DD EB       EX      DE,HL      HL = LAST LOCATION + 1
00DE 2B       DEC     HL          HL = LAST LOCATION
00DF 3E8F     LD      A,8FH      TEST LOCATION
00E1 46       LD      B,(HL)
00E2 77       LD      (HL),A
00E3 BE       CP      (HL)
00E4 70       LD      (HL),B
00E5 20CE     JR      NZ,00B5H    REDO IF NOT GOOD
00E7 2B       DEC     HL          HL = LAST LOCATION
00E8 111445   LD      DE,4514H    MUST BE ≥ 4514H
00EB DF       RST     18H         COMPARE HL,DE
00EC DA7A19   JP      C,197AH    OM ERROR IF NOT ENOUGH
00EF 11CEFF   LD      DE,0FFCEH  RESERVE 50 BYTE STRING SPACE
00F2 22B140   LD      (40B1H),HL SAVE END OF MEMORY
00F5 19       ADD     HL,DE       COMPUTE END OF AVAILABLE MEMORY.
00F6 22A040   LD      (40A0H),HL SAVE END OF AVAILABLE MEMORY.
00F9 CD4D1B   CALL   1B4DH
00FC 211101   LD      HL,0111H   PRINT "RADIO SHACK ..."
00FF C3EB37   JP      37EBH      CONTINUE
0102 C3191A   JP      1A19H      UNUSED JUMP
0105          "MEMORY SIZE"
0110 00
0111          "RADIO SHACK MODEL-III BASIC"
012C 0D
012D 1E2C     LD      E,2CH      ERROR 23 "L3" ERROR
012F C3A219   JP      19A2H      GO TO ERROR ROUTINE
0132 D7       RST     10H        SCAN FOR NEXT CHARACTER
0133 AF       XOR     A          A = 0 (POINT)
0134 01
0135 3E80     LD      A,80H      A = 80H (SET)
0137 01
0137 3E01     LD      A,01H      A = 01H (RESET)
013A F5       PUSH   AF          SAVE CALL TYPE
013B CF       RST     08H        SYNTAX ERROR IF
013C 28
013D CD1C2B   CALL   2B1CH      DECODE X
0140 FE80     CP      80H        X = ≥ 128 ?
0142 D24A1E   JP      NC,1E4AH   FC ERROR IF SO.
0145 F5       PUSH   AF          SAVE X
0146 CF       RST     08H        SYNTAX ERROR IF
0147 2C       INC     L          CURRENT CHARACTER NOT "("
0148 CD1C2B   CALL   2B1CH      DECODE Y
014B FE30     CP      30H        Y ≥ 48?
014D D24A1E   JP      NC,1E4AH   FC ERROR IF SO.
0150 16FF     LD      D,0FFH     D = Y/3
0152 14       INC     D
0153 D603     SUB    03H
0155 30FB     JR      NC,0152H
0157 C603     ADD    A,03H
0159 4F       LD      C,A        C = Y MOD 3
015A F1       POP    AF          RESTORE X
015B 87       ADD    A,A         E = X * 2
015C 5F       LD      E,A
015D 0602     LD      B,02H     DE = (256*Y/3
015F 7A       LD      A,D        +X*2)/4
0160 1F       RRA          CARRY = X MOD 2
    
```

```

0161 57      LD      D,A
0162 7B      LD      A,E
0163 1F      RRA
0164 5F      LD      E,A
0165 10F8    DJNZ    015FH
0167 79      LD      A,C      B = (Y MOD 3)
0168 8F      ADC      A,A      + (X MOD 2)
0169 3C      INC      A      + 1
016A 47      LD      B,A
016B AF      XOR      A      SET BIT B IN A,
016C 37      SCF
                     PUT IN C.
016D 8F      ADC      A,A
016E 10FD    DJNZ    016DH
0170 4F      LD      C,A
0171 7A      LD      A,D      DE = DE + START
0172 F63C    OR      3CH      OF SCREEN
0174 57      LD      D,A
0175 1A      LD      A,(DE)    GET BYTE ON SCREEN
0176 B7      OR      A      CHECK IT
0177 FA7C01  JP      M,017CH    SKIP IF GRAPHIC
017A 3E80    LD      A,80H      GRAPHIC BLANK
017C 47      LD      B,A      SAVE SCREEN BYTE
017D F1      POP     AF      GET CALL TYPE
017E B7      OR      A      CHECK IT
017F 78      LD      A,B      RESTORE SCREEN
0180 2810    JR      Z,0192H    JUMP TO POINT IF ZERO
0182 12      LD      (DE),A    STORE GRAPHIC CHARACTER.
0183 FA8F01  JP      M,018FH    JUMP IF SET
0186 79      LD      A,C      INVERT BIT MASK
0187 2F      CPL
0188 4F      LD      C,A
0189 1A      LD      A,(DE)    RESET SELECTED BIT.
018A A1      AND     C
018B 12      LD      (DE),A    STORE NEW GRAPHIC CHARACTER.
018C CF      RST    08H      SYNTAX ERROR IF
018D 29
                     CHARACTER NOT ") "
018E C9      RET
                     DONE
018F B1      OR      C      SET SELECTED BIT.
0190 18F9    JR      018BH      CONTINUE.
0192 A1      AND     C      MASK OFF IRRELEVANT BITS.
0193 C6FF    ADD     A,0FFH    ADD TRUE VALUE.
0195 9F      SBC     A,A      CARRY = TRUE/FALSE
0196 E5      PUSH   HL      RETURN TO BASIC
0197 CD8D09  CALL   098DH
019A E1      POP     HL
019B 18EF    JR      018CH      CONTINUE
019D D7      RST    10H      SCAN FOR NEXT CHARACTER
019E E5      PUSH   HL      SAVE LOCATION
019F 3A9940  LD      A,(4099H)  CHECK INKEY $ STORAGE
01A2 B7      OR      A
01A3 2006    JR      NZ,01ABH   SKIP IF NOT NULL.
01A5 CD5803  CALL   0358H      SCAN FOR KEY
01A8 B7      OR      A      NULL?
01A9 2811    JR      Z,01BCH   SKIP IF SO.
01AB F5      PUSH   AF      SAVE CHARACTER.
01AC AF      XOR      A      CLEAR INKEY $ STORAGE
01AD 329940  LD      (4099H),A
    
```


01B0	3C	INC	A	CREATE 1 BYTE STRING
01B1	CD5728	CALL	2857H	
01B4	F1	POP	AF	RESTORE CHARACTER.
01B5	2AD440	LD	HL, (40D4H)	GET STRING POINTER
01B8	77	LD	(HL), A	STORE CHARACTER.
01B9	C38428	JP	2884H	FINISH UP & RETURN
01BC	212819	LD	HL, 1928H	CREATE
01BF	222141	LD	(4121H), HL	NULL STRING
01C2	3E03	LD	A, 03H	TYPE = 3 (STRING)
01C4	32AF40	LD	(40AFH), A	
01C7	E1	POP	HL	RESTORE LOCATION
01C8	C9	RET		DONE.
01C9	3E1C	LD	A, 1CH	HOME CURSOR
01CB	CD3A03	CALL	033AH	SMALL CHARACTER
01CE	3E1F	LD	A, 1FH	CLEAR TO END OF
01D0	C33A03	JP	033AH	SCREEN
01D3	ED5F	LD	A, R	GET RANDOM SEED.
01D5	32AB40	LD	(40ABH), A	SAVE IT.
01D8	C9	RET		DONE
01D9	21003C	LD	HL, 3C00H	POINT TO SCREEN.
01DC	7E	LD	A, (HL)	GET CHARACTER.
01DD	FE80	CP	80H	GRAPHIC?
01DF	3802	JR	C, 01E3H	SKIP IF NOT.
01E1	3E2E	LD	A, 2EH	CHARACTER = "."
01E3	CD3B00	CALL	003BH	PRINT CHARACTER
01E6	23	INC	HL	NEXT CHARACTER
01E7	CB74	BIT	06H, H	DONE?
01E9	2029	JR	NZ, 0214H	SKIP OUT IF SO.
01EB	7D	LD	A, L	END OF LINE ?
01EC	E63F	AND	3FH	
01EE	20EC	JR	NZ, 01DCH	LOOP IF NOT
01F0	CD1402	CALL	0214H	NEWLINE
01F3	18E7	JR	01DCH	LOOP
01F5	10FE	DJNZ	01F5H	UNUSED SUBROUTINE
01F7	C9	RET		
01F8	C30C30	JP	300CH	CASSETTE OFF
01FB	7F	LD	A, A	DELAY SUBROUTINE.
01FC	0B	DEC	BC	
01FD	78	LD	A, B	
01FE	B1	OR	C	
01FF	20FA	JR	NZ, 01FBH	
0201	C9	RET		
0202			"(C) "80 TANDY"	COPYRIGHT
020F	0D			MESSAGE
0210	1E3D	LD	E, 3DH	UNUSED SUBROUTINE
0212	AF	XOR	A	CLEAR STATUS
0213	C9	RET		DONE.
0214	3E0D	LD	A, 0DH	CARRIAGE RETURN
0216	CD3B00	CALL	003BH	TO PRINTER
0219	AF	XOR	A	CLEAR
021A	C9	RET		DONE
021B	7E	LD	A, (HL)	GET CHARACTER
021C	23	INC	HL	POINT TO NEXT CHARACTER.
021D	FE03	CP	03H	END OF MESSAGE?
021F	C8	RET	Z	EXIT IF SO.
0220	CD3300	CALL	0033H	DISPLAY CHARACTER.
0223	FE0D	CP	0DH	END OF MESSAGE?

```

0225 20F4    JR      NZ,021BH    LOOP IF NOT
0227 C9      RET
0228 E3      EX      (SP),HL    PUT RETURN IN STACK
0229 C32A30  JP      302AH      & RESTORE HL.
022C 18E4    JR      0212H      MODEL-1 CASSETTE ON
022E FB      EI
022F C3191A  JP      1A19H      GO TO "READY"
0232 3F      CCF
0233 3C      INC     A          ROUTINE
0234 C9      RET
0235 D5      PUSH   DE          SAVE REGISTERS
0236 C5      PUSH   BC          DE & BC
0237 E5      PUSH   HL
0238 2A0E42  LD      HL,(420EH)  GET READ VECTOR
023B E3      EX      (SP),HL    SAVE IT AND RESTORE HL
023C C9      RET
023D E5      PUSH   HL          SAVE HL
023E 210030  LD      HL,3000H    SET RETURNS TO 3000H
0241 18E5    JR      0228H      CONTINUE
0243 F3      DI
0244 CD0F30  CALL   300FH      START TAPE
0247 E5      PUSH   HL          SAVE HL
0248 210630  LD      HL,3006H    SET RETURN TO 3006H
024B 18DB    JR      0228H      CONTINUE
024D E5      PUSH   HL          SAVE HL
024E 2A0C42  LD      HL,(420CH)  GET WRITE VECTOR
0251 E3      EX      (SP),HL    RESTORE HL & SAVE RETURN
0252 C9      RET
0253 E3      EX      (SP),HL    GET RETURN & SAVE HL
0254 3A1142  LD      A,(4211H)   SLOW OR FAST?
0257 B7      OR      A
0258 2803    JR      Z,025DH     SKIP IF SLOW
025A 23      INC     HL          MOVE TO FAST VECTOR
025B 23      INC     HL
025C 23      INC     HL
025D E3      EX      (SP),HL    SAVE RETURN & RESTORE HL
025E C9      RET
025F C1      POP    BC
0260 C9      RET
0261 CD6402  CALL   0264H
0264 18E7    JR      024DH      WRITE BYTE TO TAPE
0266 3C 3C 18  TIME DATA (60,60,24)
0269 1F 1C 1F 1E  MONTH LENGTHS
026D 1F 1E 1F 1F  FOR DATE$
0271 1E 1F 1E 1F
0275 00 00  UNUSED
0277 1D 1E  CLEAR LINE
0279      "DISKETTE?"     MESSAGE
0282 03      END OF MESSAGE
0283 F2      UNUSED
0286 C38702  JP      0287H      MODEL-I CASSETTE ON
0287 F3      DI
0288 CD0F30  CALL   300FH      TURN ON TAPE
028B 18B0    JR      023DH      SET WRITE VECTORS
028D 3A4038  LD      A,(3840H)  CHECK <BREAK>
0290 E604    AND    04H
0292 C9      RET
  
```

0293	C34302	JP	0243H	CASSETTE HEADER READ.
0296	18AB	JR	0243H	CASSETTE HEADER READ
0298	3A1042	LD	A, (4210H)	SET CLOCK ON BIT.
029B	CBC7	SET	00H,A	BIT
029D	321042	LD	(4210H),A	SAVE FLAG
02A0	C9	RET		
02A1	3A1042	LD	A, (4210H)	RESET CLOCK ON
02A4	CB87	RES	00H,A	BIT.
02A6	18F5	JR	029DH	
02A8	C9	RET		UNUSED
02A9	CD1403	CALL	0314H	LOAD EXECUTION
02AC	22DF40	LD	(40DFH),HL	ADDRESS & SAVE
02AF	CDF801	CALL	01F8H	TURN TAPE OFF.
02B2	CDE241	CALL	41E2H	DOS LINK
02B5	318842	LD	SP,4288H	SET SYSTEM STACK
02B8	CDFE20	CALL	20FEH	NEW DISPLAY LINE
02BB	3E2A	LD	A,2AH	DISPLAY "*"
02BD	CD2A03	CALL	032AH	
02C0	CDB31B	CALL	1BB3H	PRINT "? "; INPUT RESPONSE
02C3	DACC06	JP	C,06CCH	JUMP IF <BREAK>
02C6	D7	RST	10H	SCAN FOR CHARACTER.
02C7	CA9719	JP	Z,1997H	SYNTAX ERROR IF NONE
02CA	FE2F	CP	2FH	"/"?
02CC	284F	JR	Z,031DH	SKIP IF SO
02CE	CD9302	CALL	0293H	READER HEADER & SYNC
02D1	CD3502	CALL	0235H	READ CHARACTER.
02D4	FE55	CP	55H	SYSTEM FLAG?
02D6	20F9	JR	NZ,02D1H	LOOP IF NOT
02D8	0606	LD	B,06H	6 BYTE NAME
02DA	7E	LD	A,(HL)	GET GIVEN CHARACTER.
02DB	B7	OR	A	NULL?
02DC	2809	JR	Z,02E7H	SKIP IF SO
02DE	CD3502	CALL	0235H	READ TAPE CHARACTER.
02E1	BE	CP	(HL)	SAME AS GIVEN?
02E2	23	INC	HL	POINT TO NEXT
02E3	20EC	JR	NZ,02D1H	TRY AGAIN IF NOT.
02E5	10F3	DJNZ	02DAH	LOOP UNTIL DONE.
02E7	CD2C02	CALL	022CH	DUMMY CALL
02EA	CD3502	CALL	0235H	READY BYTE
02ED	FE78	CP	78H	TRANSFER ADDRESS HEADER?
02EF	28B8	JR	Z,02A9H	JUMP IF SO.
02F1	FE3C	CP	3CH	DATA HEADER?
02F3	20F5	JR	NZ,02EAH	LOOP IF NOT
02F5	CD3502	CALL	0235H	READ BYTE
02F8	47	LD	B,A	B = DATA FIELD LENGTH
02F9	CD1403	CALL	0314H	HL = DATA LOAD POINT
02FC	85	ADD	A,L	START CHECKSUM
02FD	4F	LD	C,A	
02FE	CD3502	CALL	0235H	READ BYTE
0301	77	LD	(HL),A	STORE IT
0302	23	INC	HL	POINT TO NEXT LOCATION
0303	81	ADD	A,C	UPDATE CHECKSUM
0304	4F	LD	C,A	
0305	10F7	DJNZ	02FEH	LOOP UNTIL FIELD LOADED.
0307	CD3502	CALL	0235H	READ CHECKSUM
030A	B9	CP	C	COMPARE WITH COMPUTED
030B	28DA	JR	Z,02E7H	LOOP IF GOOD

030D	3E43	LD	A,43H	DISPLAY "C" FOR
030F	323E3C	LD	(3C3EH),A	CHECK SUM ERROR.
0312	18D6	JR	02EAH	CONTINUE ANYHOW
0314	CD3502	CALL	0235H	READ LSB
0317	6F	LD	L,A	PUT IN L
0318	CD3502	CALL	0235H	READ MSB
031B	67	LD	H,A	PUT IN H
031C	C9	RET		DONE
031D	EB	EX	DE,HL	DE = EXECUTION ADDRESS
031E	2ADF40	LD	HL,(40DFH)	
0321	EB	EX	DE,HL	
0322	D7	RST	10H	SCAN FOR NEXT CHARACTER
0323	C45A1E	CALL	NZ,1E5AH	DECADE ADDRESS IF GIVEN
0326	208A	JR	NZ,02B2H	JUMP BACK ON ERROR
0328	EB	EX	DE,HL	PUT EXECUTION ADDRESS IN HL
0329	E9	JP	(HL)	JUMP TO IT.
032A	C5	PUSH	BC	SAVE BC
032B	4F	LD	C,A	PUT CHARACTER IN C.
032C	CDC141	CALL	41C1H	LINK TO DOS
032F	3A9C40	LD	A,(409CH)	GET DEVISE FLAG.
0332	B7	OR	A	TEST IT.
0333	79	LD	A,C	PUT CHARACTER BACK
0334	C1	POP	BC	RESTORE BC
0335	FA6402	JP	M,0264H	TYPE = -1 GO TO TAPE
0338	2062	JR	NZ,039CH	TYPE = 1 GO TO PRINTER
033A	D5	PUSH	DE	SAVE DE
033B	CD3300	CALL	0033H	GO TO VIDEO
033E	F5	PUSH	AF	SAVE AF
033F	CD4803	CALL	0348H	UPDATE CURSOR POS.
0342	32A640	LD	(40A6H),A	SAVE LINE POSITION
0345	F1	POP	AF	RESTORE REGISTERS
0346	D1	POP	DE	
0347	C9	RET		DONE
0348	3A3D40	LD	A,(403DH)	DOUBLE-SIZE?
034B	E608	AND	08H	
034D	3A2040	LD	A,(4020H)	GET CURSOR POS.
0350	2803	JR	Z,0355H	SKIP IF SMALL CHARACTER.
0352	0F	RRCA		DIVIDE POS BY 2
0353	E61F	AND	1FH	KEEP ON LINE
0355	E63F	AND	3FH	KEEP ON LINE
0357	C9	RET		DONE
0358	CDC441	CALL	41C4H	LINK TO DOS
035B	D5	PUSH	DE	SAVE DE
035C	CD2B00	CALL	002BH	SCAN KEYBOARD
035F	D1	POP	DE	RESTORE DE
0360	C9	RET		DONE
0361	AF	XOR	A	
0362	329940	LD	(4099H),A	INKEY\$ = ""
0365	32A640	LD	(40A6H),A	TAB POS = 0
0368	CDAF41	CALL	41AFH	LINK TO DOS
036B	C5	PUSH	BC	SAVE BC
036C	2AA740	LD	HL,(40A7H)	HL => INPUT BUFFER
036F	06F0	LD	B,0F0H	UP TO 240 CHARACTER.
0371	CDD905	CALL	05D9H	INPUT TEXT TO BUFFER
0374	F5	PUSH	AF	SAVE STATUS
0375	48	LD	C,B	HL=> END OF TEXT
0376	0600	LD	B,00H	

0378	09	ADD	HL,BC	
0379	3600	LD	(HL),00H	PUT NULL AT END OF TEXT
037B	2AA740	LD	HL,(40A7H)	GET BUFFER LOCATION
037E	F1	POP	AF	RESTORE STATUS
037F	C1	POP	BC	RESTORE BC
0380	2B	DEC	HL	BACK UP BEFORE BUFFER
0381	D8	RET	C	CARRY = <BREAK>
0382	AF	XOR	A	CLEAR STATUS
0383	C9	RET		DONE
0384	CD5803	CALL	0358H	CHECK KEYBOARD
0387	B7	OR	A	
0388	C0	RET	NZ	RETURN IF KEY PRESSED
0389	18F9	JR	0384H	LOOP OTHERWISE
038B	AF	XOR	A	RESET DEVICE TO VIDEO
038C	329C40	LD	(409CH),A	
038F	3A9B40	LD	A,(409BH)	PRINTER AT LEFT MARGIN?
0392	B7	OR	A	
0393	C8	RET	Z	RETURN IF SO
0394	3E0D	LD	A,0DH	RETURN CARRIAGE TO
0396	D5	PUSH	DE	LEFT MARGIN
0397	CD9C03	CALL	039CH	
039A	D1	POP	DE	
039B	C9	RET		DONE
039C	F5	PUSH	AF	SAVE REGISTERS
039D	D5	PUSH	DE	
039E	C5	PUSH	BC	
039F	4F	LD	C,A	SAVE CHARACTER IN C
03A0	1E00	LD	E,00H	
03A2	FE0C	CP	0CH	FORM FEED?
03A4	2810	JR	Z,03B6H	JUMP IF SO
03A6	FE0A	CP	0AH	LINE FEED?
03A8	2003	JR	NZ,03ADH	SKIP IF NOT
03AA	3E0D	LD	A,0DH	MAKE IT A CARRIAGE
03AC	4F	LD	C,A	RETURN
03AD	FE0D	CP	0DH	CARRIAGE RETURN
03AF	2805	JR	Z,03B6H	JUMP IF SO
03B1	3A9B40	LD	A,(409BH)	UPDATE PRINTER
03B4	3C	INC	A	LINE WIDTH
03B5	5F	LD	E,A	PUT IN E.
03B6	7B	LD	A,E	GET PRINTER LINE WIDTH
03B7	329B40	LD	(409BH),A	SAVE IT
03BA	79	LD	A,C	GET CHARACTER TO PRINT
03BB	CD3B00	CALL	003BH	PRINT IT
03BE	C1	POP	BC	RESTORE REGISTERS
03BF	D1	POP	DE	
03C0	F1	POP	AF	
03C1	C9	RET		DONE
03C2	79	LD	A,C	PUT CHARACTER IN A.
03C3	FE20	CP	20H	CONTROL CHARACTER?
03C5	301E	JR	NC,03E5H	JUMP IF SO
03C7	FE0D	CP	0DH	CARRIAGE RETURN?
03C9	282A	JR	Z,03F5H	JUMP IF SO
03CB	FE0C	CP	0CH	FORM FEED
03CD	2030	JR	NZ,03FFH	JUMP IF NOT
03CF	DD7E03	LD	A,(IX+03H)	B = # LINES LEFT IN PAGE.
03D2	DD9604	SUB	(IX+04H)	
03D5	47	LD	B,A	

03D6	CD4004	CALL	0440H	WAIT FOR PRINTER
03D9	3E0A	LD	A,0AH	OUTPUT LINE FEED
03DB	D3F8	OUT	(0F8H),A	
03DD	10F7	DJNZ	03D6H	LOOP UNTIL PAGE FEED
03DF	DD360500	LD	(IX+05H),00H	# CHARACTERS PRINTED = 0
03E3	1854	JR	0439H	SET # LINES PRINTED & EXIT
03E5	FE80	CP	80H	GRAPHICS CHAR?
03E7	3030	JR	NC,0419H	JUMP IF SO.
03E9	0600	LD	B,00H	MSB = 0
03EB	D620	SUB	20H	ADJUST CHARACTER TO TABLE
03ED	4F	LD	C,A	BC = ADJUSTED VALUE
03EE	214531	LD	HL,3145H	HL => CHARACTER TABLE
03F1	09	ADD	HL,BC	HL => CHARACTER
03F2	4E	LD	C,(HL)	GET NEW CHARACTER
03F3	180E	JR	0403H	CONTINUE
03F5	DD7E05	LD	A,(IX+05H)	GET # CHARACTERS PRINTED
03F8	B7	OR	A	NONE?
03F9	79	LD	A,C	RESTORE CHARACTER TO A.
03FA	2003	JR	NZ,03FFH	SKIP IF SOME PRINTED
03FC	3E0A	LD	A,0AH	CHAR = LINE FEED
03FE	4F	LD	C,A	
03FF	FE20	CP	20H	CONTROL CHARACTER?
0401	3816	JR	C,0419H	JUMP IF SO.
0403	DD7E06	LD	A,(IX+06H)	GET MAX PRINT WIDTH.
0406	3C	INC	A	UNLIMITED?
0407	2810	JR	Z,0419H	SKIP IF SO.
0409	DDBE05	CP	(IX+05H)	LINE FULL?
040C	300B	JR	NC,0419H	SKIP IF NOT
040E	CD4004	CALL	0440H	WAIT FOR PRINTER
0411	3E0D	LD	A,0DH	CARRIAGE RETURN
0413	D3F8	OUT	(0F8H),A	OUTPUT IT.
0415	DD360500	LD	(IX+05H),00H	# CHARACTERS PRINTED = 0
0419	CD4004	CALL	0440H	WAIT FOR PRINTER
041C	79	LD	A,C	RESTORE CHARACTER.
041D	D3F8	OUT	(0F8H),A	OUTPUT IT
041F	DD3405	INC	(IX+05H)	INCREMENT # CHARACTERS PRINTED
0422	FE0D	CP	0DH	CARRIAGE RETURN
0424	2804	JR	Z,042AH	SKIP IF SO.
0426	FE0A	CP	0AH	LINE FEED?
0428	2013	JR	NZ,043DH	SKIP IF NOT
042A	DD360500	LD	(IX+05H),00H	EMPTY LINE
042E	DD3404	INC	(IX+04H)	INCREMENT # LINES PRINTED
0431	DD7E04	LD	A,(IX+04H)	
0434	DDBE03	CP	(IX+03H)	ON NEXT PAGE?
0437	2004	JR	NZ,043DH	SKIP IF NOT
0439	DD360401	LD	(IX+04H),01H	TOP OF PAGE?
043D	AF	XOR	A	CLEAR STATUS
043E	79	LD	A,C	RESTORE CHARACTER.
043F	C9	RET		DONE
0440	CD4B04	CALL	044BH	CHECK PRINTER
0443	C8	RET	Z	RETURN IF READY
0444	CD8D02	CALL	028DH	CHECK <BREAK>
0447	28F7	JR	Z,0440H	LOOP IF NOT PRESSED
0449	F1	POP	AF	EXIT CALLING SUBROUTINE
044A	C9	RET		
044B	DBF8	IN	A,(0F8H)	GET PRINTER STATUS
044D	E6F0	AND	0F0H	MASK OFF IRRELEVANT BITS

044F	FE30	CP	30H	READY?
0451	C9	RET		
0452	21BF36	LD	HL, 36BFH	INITIALIZE
0455	111540	LD	DE, 4015H	KI, DO, PR
0458	011800	LD	BC, 0018H	
045B	EDB0	LDIR		
045D	21F936	LD	HL, 36F9H	INITIALIZE
0460	11E541	LD	DE, 41E5H	RI, RO, RN
0463	011800	LD	BC, 0018H	
0466	EDB0	LDIR		
0468	C9	RET		
0469	20			UNUSED
046A	DA			UNUSED
046B	AF	XOR	A	CLEAR STATUS
046C	321442	LD	(4214H), A	UNPROTECT SCREEN
046F	2AA440	LD	HL, (40A4H)	POINT TO START OF PROGRAM
0472	C9	RET		DONE
0473	F3	DI		DISABLE INTERRUPTS
0474	DD6E03	LD	L, (IX+03H)	GET CURSOR POSITION
0477	DD6604	LD	H, (IX+04H)	
047A	DD7E05	LD	A, (IX+05H)	GET CHARACTER AT CURSOR
047D	B7	OR	A	CURSOR ON?
047E	2801	JR	Z, 0481H	SKIP IF OFF
0480	77	LD	(HL), A	DISPLAY CHARACTER AT CURSOR
0481	79	LD	A, C	GET CHARACTER TO DISPLAY
0482	FE20	CP	20H	CONTROL CHARACTER?
0484	DA2105	JP	C, 0521H	JUMP IF SO.
0487	FEC0	CP	0C0H	TAB/SPECIAL CHARACTER?
0489	302C	JR	NC, 04B7H	JUMP IF SO
048B	CD7605	CALL	0576H	DISPLAY CHARACTER.
048E	7C	LD	A, H	MAKE SURE CURSOR
048F	E603	AND	03H	IS ON SCREEN
0491	F63C	OR	3CH	
0493	67	LD	H, A	
0494	56	LD	D, (HL)	GET CHARACTER AT CURSOR
0495	DD7E05	LD	A, (IX+05H)	CURSOR ON?
0498	B7	OR	A	
0499	280D	JR	Z, 04A8H	SKIP IF NOT
049B	DD7205	LD	(IX+05H), D	SAVE CHARACTER UNDER CURSOR
049E	DD7E06	LD	A, (IX+06H)	GET CURSOR CHARACTER.,
04A1	FE20	CP	20H	CONTROL CHARACTER?
04A3	3002	JR	NC, 04A7H	USE IT IF NOT
04A5	3EB0	LD	A, 0B0H	USE DEFAULT CURSOR.
04A7	77	LD	(HL), A	DISPLAY CURSOR
04A8	DD7503	LD	(IX+03H), L	SAVE CURSOR POSITION
04AB	DD7404	LD	(IX+04H), H	
04AE	AF	XOR	A	CLEAR STATUS
04AF	79	LD	A, C	RESTORE CHARACTER.
04B0	FB	EI		ENABLE INTERRUPTS
04B1	C9	RET		DONE
04B2	7D	LD	A, L	CURSOR TO START
04B3	E6C0	AND	0C0H	OF LINE
04B5	6F	LD	L, A	
04B6	C9	RET		
04B7	DD7E07	LD	A, (IX+07H)	TABS OR SPECIAL CHARACTERS?
04BA	B7	OR	A	
04BB	79	LD	A, C	RESTORE CHARACTER

04BC	20CD	JR	NZ,048BH	DISPLAY SPECIAL CHARACTER IF SET
04BE	D6C0	SUB	0C0H	COMPUTE TAB.
04C0	28CC	JR	Z,048EH	EXIT IF TAB (0)
04C2	47	LD	B,A	DISPLAY REQUESTED
04C3	3E20	LD	A,20H	# OF SPACES
04C5	CD7605	CALL	0576H	
04C8	10F9	DJNZ	04C3H	
04CA	18C2	JR	048EH	CONTINUE
04CC	7E	LD	A,(HL)	STORE CHARACTER AT CURSOR
04CD	DD7705	LD	(IX+05H),A	STORE CHARACTER BELOW CURSOR
04D0	C9	RET		
04D1	AF	XOR	A	SET CURSOR OFF
04D2	18F9	JR	04CDH	
04D4	21003C	LD	HL,3C00H	HOME CURSOR
04D7	3A1042	LD	A,(4210H)	SMALL CHARACTER
04DA	E6FB	AND	0FBH	
04DC	CD7005	CALL	0570H	
04DF	3A1442	LD	A,(4214H)	GET # LINES TO PROTECT
04E2	E607	AND	07H	MASK IT
04E4	C8	RET	Z	RETURN WHEN DONE
04E5	CD0405	CALL	0504H	MOVE DOWN ONE LINE
04E8	3D	DEC	A	ONE LESS LEFT
04E9	18F9	JR	04E4H	LOOP UNTIL DONE
04EB	2B	DEC	HL	BACKUP CURSOR
04EC	3A1042	LD	A,(4210H)	CHECK IF LARGE CHARACTER.
04EF	E604	AND	04H	
04F1	2801	JR	Z,04F4H	SKIP IF NOT
04F3	2B	DEC	HL	BACKUP AGAIN
04F4	3620	LD	(HL),20H	BLANK OUT POSITION
04F6	C9	RET		DONE
04F7	3A1042	LD	A,(4210H)	DOUBLE SIZE CHARACTERS?
04FA	E604	AND	04H	
04FC	C4FF04	CALL	NZ,04FFH	DO IT TWICE IF SO
04FF	7D	LD	A,L	CHECK POSITION
0500	E63F	AND	3FH	IN LINE
0502	2B	DEC	HL	BACKUP
0503	C0	RET	NZ	RETURN IF NOT AT START
0504	114000	LD	DE,0040H	MOVE DOWN ONE LINE
0507	19	ADD	HL,DE	
0508	C9	RET		
0509	23	INC	HL	MOVE CURSOR FORWARD
050A	7D	LD	A,L	CHECK POSITION
050B	E63F	AND	3FH	IN LINE
050D	C0	RET	NZ	RETURN IF NOT AT END
050E	11C0FF	LD	DE,0FFC0H	MOVE UP ONE LINE
0511	19	ADD	HL,DE	
0512	C9	RET		
0513	3A1042	LD	A,(4210H)	SET DOUBLE SIZE CHARACTERS
0516	F604	OR	04H	
0518	CD7005	CALL	0570H	
051B	23	INC	HL	NEXT POSITION
051C	7D	LD	A,L	SET TO EVEN POSITION
051D	E6FE	AND	0FEH	BECAUSE LARGE CHARACTERS
051F	6F	LD	L,A	ONLY DISPLAY EVERY
0520	C9	RET		OTHER POSITION.
0521	118E04	LD	DE,048EH	SET RETURN
0524	D5	PUSH	DE	

0525	FE08	CP	08H	BACKSPACE?
0527	28C2	JR	Z,04EBH	JUMP IF SO.
0529	FE0A	CP	0AH	LINE FEED?
052B	CAAF05	JP	Z,05AFH	JUMP IF SO
052E	FE0D	CP	0DH	CARRIAGE RETURN?
0530	CAAF05	JP	Z,05AFH	JUMP IF SO
0533	FE0E	CP	0EH	CURSOR ON?
0535	2895	JR	Z,04CCH	JUMP IF SO.
0537	FE0F	CP	0FH	CURSOR OFF?
0539	2896	JR	Z,04D1H	JUMP IF SO
053B	D615	SUB	15H	SWAP TAB/SPECIAL CHARACTERS
053D	2821	JR	Z,0560H	JUMP IF SO
053F	3D	DEC	A	SPECIAL/ALTERNATES?
0540	2829	JR	Z,056BH	JUMP IF SO
0542	3D	DEC	A	DOUBLE SIZE?
0543	28CE	JR	Z,0513H	JUMP IF SO
0545	3D	DEC	A	CURSOR BACK?
0546	28AF	JR	Z,04F7H	JUMP IF SO
0548	3D	DEC	A	CURSOR FORWARD
0549	28BE	JR	Z,0509H	JUMP IF SO.
054B	3D	DEC	A	CURSOR DOWN?
054C	28B6	JR	Z,0504H	JUMP IF SO
054E	3D	DEC	A	CURSOR UP?
054F	28BD	JR	Z,050EH	JUMP IF SO
0551	3D	DEC	A	HOME CURSOR
0552	CAD404	JP	Z,04D4H	JUMP IF SO
0555	3D	DEC	A	RESTART LINE?
0556	CAB204	JP	Z,04B2H	JUMP IF SO
0559	3D	DEC	A	CLEAR TO END OF LINE?
055A	2860	JR	Z,05BCH	JUMP IF SO
055C	3D	DEC	A	CLEAR TO END OF SCREEN?
055D	2866	JR	Z,05C5H	JUMP IF SO
055F	C9	RET		DONE
0560	DD7E07	LD	A, (IX+07H)	TOGGLE TABS/ALTERNATE
0563	E601	AND	01H	CHARACTERS FLAG
0565	EE01	XOR	01H	
0567	DD7707	LD	(IX+07H), A	
056A	C9	RET		DONE
056B	3A1042	LD	A, (4210H)	TOGGLE SPECIAL
056E	EE08	XOR	08H	ALTERNATE CHARACTERS
0570	321042	LD	(4210H), A	SAVE MASK
0573	D3EC	OUT	(0ECH), A	OUTPUT IT
0575	C9	RET		DONE
0576	77	LD	(HL), A	DISPLAY CHARACTER.
0577	23	INC	HL	ADVANCE CURSOR
0578	3A1042	LD	A, (4210H)	DOUBLE SIZE?
057B	E604	AND	04H	
057D	2801	JR	Z,0580H	SKIP IF NOT
057F	23	INC	HL	ADVANCE AGAIN
0580	7C	LD	A, H	OFF-SCREEN?
0581	FE40	CP	40H	
0583	C0	RET	NZ	RETURN IF NOT
0584	CD0E05	CALL	050EH	PUT CURSOR BACK ON.
0587	E5	PUSH	HL	SAVE CURSOR POSITION
0588	3A1442	LD	A, (4214H)	GET # LINES TO PROTECT
058B	E607	AND	07H	
058D	21003C	LD	HL, 3C00H	HL => START OF SCREEN

0590	110004	LD	DE,0400H	DE = SIZE OF SCREEN
0593	C5	PUSH	BC	SAVE BC
0594	014000	LD	BC,0040H	BC = # CHARACTERS/LINE
0597	3C	INC	A	ADJUST # LINES TO PROTECT
0598	09	ADD	HL,BC	MOVE START DOWN ONE LINE
0599	EB	EX	DE,HL	REDUCE SIZE BY ONE LINE
059A	B7	OR	A	
059B	ED42	SBC	HL,BC	
059D	EB	EX	DE,HL	
059E	3D	DEC	A	ONE LESS LINE TO PROTECT
059F	20F7	JR	NZ,0598H	LOOP UNTIL DONE.
05A1	D5	PUSH	DE	SAVE DE & HL
05A2	E5	PUSH	HL	
05A3	B7	OR	A	MOVE START BACK UP
05A4	ED42	SBC	HL,BC	ONE LINE
05A6	EB	EX	DE,HL	SOURCE = START + ONE LINE
05A7	E1	POP	HL	DEST = START OF SCREEN
05A8	C1	POP	BC	COUNT = SCREEN SIZE - ONE LINE
05A9	EDB0	LDIR		SCROLL UNPROTECTED SCREEN
05AB	C1	POP	BC	RESTORE BC
05AC	EB	EX	DE,HL	HL = CURSOR POSITION
05AD	1817	JR	05C6H	CLEAR TO END OF SCREEN
05AF	CDB204	CALL	04B2H	MOVE TO START OF LINE
05B2	E5	PUSH	HL	SAVE CURSOR POSITION
05B3	CD0405	CALL	0504H	NINE DOWN ONE LINE
05B6	7C	LD	A,H	OFF - SCREEN
05B7	FE40	CP	40H	
05B9	28CD	JR	Z,0588H	SCROLL IF SO.
05BB	D1	POP	DE	GET OLD CURSOR POSITION
05BC	E5	PUSH	HL	SAVE NEW CURSOR POSITION
05BD	54	LD	D,H	DE => END OF LINE
05BE	7D	LD	A,L	
05BF	F63F	OR	3FH	
05C1	5F	LD	E,A	
05C2	13	INC	DE	DE => START OF NEXT LINE
05C3	1804	JR	05C9H	CLEAR TO END OF LINE
05C5	E5	PUSH	HL	SAVE CURSOR POSITION.
05C6	110040	LD	DE,4000H	DE => JUST OFF SCREEN
05C9	3620	LD	(HL),20H	CLEAR POSITION
05CB	23	INC	HL	POINT TO NEXT POSITION
05CC	DF	RST	18H	DONE?
05CD	20FA	JR	NZ,05C9H	LOOP UNTIL DONE
05CF	E1	POP	HL	RESTORE CURSOR POSITION
05D0	C9	RET		DONE
05D1	"RON"			(UNUSED
05D4	E6F0	AND	0F0H	CODE)
05D6	FE30	CP	30H	(UNUSED
05D8	C9	RET		CODE)
05D9	E5	PUSH	HL	SAVE BUFFER POINTER
05DA	3E0E	LD	A,0EH	TURN CURSOR ON
05DC	CD3300	CALL	0033H	
05DF	48	LD	C,B	C = BUFFER SIZE
05E0	CD4900	CALL	0049H	WAIT FOR INPUT
05E3	FE20	CP	20H	CONTROL CHARACTER?
05E5	3025	JR	NC,060CH	JUMP IF NOT
05E7	FE0D	CP	0DH	CARRIAGE RETURN?
05E9	CA6206	JP	Z,0662H	JUMP IF SO

05EC	FE1F	CP	1FH	CLEAR?
05EE	2829	JR	Z,0619H	JUMP IF SO
05F0	FE01	CP	01H	BREAK?
05F2	286D	JR	Z,0661H	JUMP IF SO
05F4	11E005	LD	DE,05E0H	SET RETURN
05F7	D5	PUSH	DE	
05F8	FE08	CP	08H	BACKSPACE?
05FA	2834	JR	Z,0630H	JUMP IF SO.
05FC	FE18	CP	18H	NON-DESTRUCTIVE BACKSPACE
05FE	282B	JR	Z,062BH	JUMP IF SO.
0600	FE09	CP	09H	TAB?
0602	2842	JR	Z,0646H	JUMP IF SO.
0604	FE19	CP	19H	LARGE CHARACTERS?
0606	2839	JR	Z,0641H	JUMP IF SO
0608	FE0A	CP	0AH	LINE FEED?
060A	C0	RET	NZ	RETURN IF NOT.
060B	D1	POP	DE	FIX STACK
060C	77	LD	(HL),A	STORE CHARACTER IN BUFFER
060D	78	LD	A,B	BUFFER FULL?
060E	B7	OR	A	
060F	28CF	JR	Z,05E0H	LOOP BACK IF SO
0611	7E	LD	A,(HL)	RESTORE CHARACTER.
0612	23	INC	HL	POINT TO NEXT STORAGE
0613	CD3300	CALL	0033H	DISPLAY CHARACTER.
0616	05	DEC	B	DECREASE AREA LEFT
0617	18C7	JR	05E0H	LOOP FOR MORE
0619	CDC901	CALL	01C9H	CLEAR SCREEN
061C	41	LD	B,C	CLEAR BUFFER
061D	E1	POP	HL	
061E	E5	PUSH	HL	
061F	C3E005	JP	05E0H	LOOP FOR MORE
0622	CD3006	CALL	0630H	BACKSPACE
0625	2B	DEC	HL	GET CHARACTER BEFORE
0626	7E	LD	A,(HL)	CURRENT POSITION
0627	23	INC	HL	
0628	FE0A	CP	0AH	IS IT A LINE FEED?
062A	C8	RET	Z	RETURN OF SO
062B	78	LD	A,B	IS THE BUFFER EMPTY?
062C	B9	CP	C	
062D	20F3	JR	NZ,0622H	LOOP IF NOT
062F	C9	RET		DONE
0630	78	LD	A,B	BUFFER EMPTY?
0631	B9	CP	C	
0632	C8	RET	Z	RETURN IF SO
0633	2B	DEC	HL	GET CHARACTER BEFORE
0634	7E	LD	A,(HL)	CURRENT POSITION
0635	FE0A	CP	0AH	IS IT A LINE FEED?
0637	23	INC	HL	FIX BUFFER POSITION
0638	C8	RET	Z	RETURN IF IT WAS
0639	2B	DEC	HL	BACKUP ONE POSITION
063A	3E08	LD	A,08H	BACK CURSOR UP
063C	CD3300	CALL	0033H	
063F	04	INC	B	INCREASE AREA LEFT
0640	C9	RET		DONE
0641	3E17	LD	A,17H	GO TO LARGE CHARACTERS.
0643	C33300	JP	0033H	
0646	CD4803	CALL	0348H	GET LINE POSITION

```

0649 E607      AND      07H      MASK OFF TOP 5 BITS
064B 2F       CPL       INVERT IT
064C 3C       INC       A        INCREMENT IT
064D C608     ADD      A,08H     AND ADD 8 TO IT
064F 5F       LD        E,A      E IS NOW TAB LENGTH
0650 78       LD        A,B      BUFFER FULL?
0651 B7       OR        A
0652 C8       RET      Z        RETURN IF SO
0653 3E20     LD      A,20H     PUT SPACE IN BUFFER
0655 77       LD      (HL),A
0656 23       INC     HL        POINT TO NEXT BUFFER LOCATION
0657 D5       PUSH   DE        DISPLAY SPACE
0658 CD3300   CALL  0033H
065B D1       POP    DE
065C 05       DEC    B         DECREASE AREA LEFT
065D 1D       DEC    E         DECREASE TAB LEFT
065E C8       RET    Z         RETURN IF DONE
065F 18EF     JR     0650H     LOOP UNTIL DONE
0661 37       SCF                   CARRY MEANS <BREAK>
0662 F5       PUSH  AF         SAVE STATUS
0663 3E0D     LD    A,0DH     MARK END OF BUFFER
0665 77       LD    (HL),A
0666 CD3300   CALL  0033H     MOVE TO NEW LINE
0669 3E0F     LD    A,0FH
066B CD3300   CALL  0033H     MOVE TO NEW LINE
066E 79       LD    A,C      B = # CHARACTERS READ
066F 90       SUB    B
0670 47       LD    B,A
0671 F1       POP   AF         RESTORE STATUS
0672 E1       POP   HL         RESTORE BUFFER LOCATION
0673 C9       RET                   DONE.
0674 E5       PUSH  HL         SAVE HL, IX, DE
0675 DDE5     PUSH  IX
0677 D5       PUSH  DE
0678 DDE1     POP   IX         IX = DE => DCB
067A D5       PUSH  DE
067B 219406   LD    HL,0694H   SET RETURN
067E E5       PUSH  HL
067F 4F       LD    C,A      C = CHARACTER TO OUTPUT
0680 1A       LD    A,(DE)    GET DRIVER TYPE
0681 CB7F     BIT   07H,A     DISK FILE?
0683 2805     JR    Z,068AH   SKIP IF NOT
0685 A0       AND   B         REQUESTED OPERATION
0686 B8       CP    B         AVAILABLE?
0687 C23340   JP    NZ,4033H  GO TO DOS IF SO.
068A A0       AND   B         SET Z = WRITE
068B FE02     CP    02H      NZ = READ
068D DD6E01   LD    L,(IX+01H) HL = DRIVER ADDRESS
0690 DD6602   LD    H,(IX+02H)
0693 E9       JP    (HL)     GO TO DRIVER
0694 D1       POP   DE      RESTORE RESISTERS
0695 DDE1     POP   IX
0697 E1       POP   HL
0698 C1       POP   BC
0699 C9       RET                   DONE
069A AF     XOR   A         CLEAR
069B 329F40   LD    (409FH),A FLAG BITS
    
```

069E	16FF	LD	D,0FFH	BUFFER 255 BYTES
06A0	C38D2B	JP	2B8DH	CONTINUE
06A3	E6FD	AND	0FDH	RESET
06A5	329F40	LD	(409FH),A	DATA FLAG
06A8	3E3A	LD	A,3AH	RESTORE CHARACTER
06AA	B7	OR	A	RESERVED WORD?
06AB	F2E206	JP	P,06E2H	JUMP IF NOT
06AE	3A9F40	LD	A,(409FH)	CURRENTLY IN
06B1	1F	RRA		QUOTED STRING?
06B2	382E	JR	C,06E2H	JUMP IF SO
06B4	1F	RRA		IN
06B5	1F	RRA		REM LINE?
06B6	303E	JR	NC,06F6H	JUMP IF NOT
06B8	7E	LD	A,(HL)	GET TOKEN
06B9	FEFB	CP	0FBH	REM?
06BB	E5	PUSH	HL	SAVE POSITION IN TEXT
06BC	C5	PUSH	BC	SAVE POSITION IN BUFFER
06BD	21DF06	LD	HL,06DFH	RETURN TO 06DFH
06C0	E5	PUSH	HL	
06C1	C0	RET	NZ	CONTINUE IF NOT
06C2	0B	DEC	BC	BACKUP IN BUFFER
06C3	0A	LD	A,(BC)	GET CHARACTER
06C4	FE4D	CP	4DH	"M"?
06C6	C0	RET	NZ	CONTINUE IF NOT
06C7	0B	DEC	BC	BACKUP
06C8	0A	LD	A,(BC)	GET CHARACTER.
06C9	FE45	CP	45H	"E"?
06CB	C0	RET	NZ	CONTINUE IF NOT
06CC	0B	DEC	BC	BACKUP
06CD	0A	LD	A,(BC)	GET CHARACTER
06CE	FE52	CP	52H	"R"
06D0	C0	RET	NZ	CONTINUE IF NOT
06D1	0B	DEC	BC	BACKUP
06D2	0A	LD	A,(BC)	GET CHARACTER
06D3	FE3A	CP	3AH	COLON?
06D5	C0	RET	NZ	CONTINUE IF NOT
06D6	F1	POP	AF	CLEAR RETURN
06D7	F1	POP	AF	CLEAR STACK
06D8	E1	POP	HL	GET POSITION
06D9	14	INC	D	DECREASE
06DA	14	INC	D	BUFFER
06DB	14	INC	D	SIZE BY 4.
06DC	14	INC	D	
06DD	1825	JR	0704H	GET CHARACTER & CONTINUE
06DF	C1	POP	BC	GET BUFFER POSITION
06E0	E1	POP	HL	GET TEXT POSITION
06E1	7E	LD	A,(HL)	GET CHARACTER
06E2	C3892B	JP	2B89H	CONTINUE
06E5	3A9F40	LD	A,(409FH)	GET FLAG
06E8	F602	OR	02H	SET DATA BIT
06EA	329F40	LD	(409FH),A	SAVE FLAG
06ED	AF	XOR	A	CLEAR STATUS
06EE	C9	RET		DONE
06EF	3A9F40	LD	A,(409FH)	GET FLAG
06F2	F604	OR	04H	SET REM BIT
06F4	18F4	JR	06EAH	CONTINUE
06F6	17	RLA		IN DATA STATEMENT

06F7	38E9	JR	C,06E2H	JUMP IF SO
06F9	7E	LD	A, (HL)	GET CHARACTER
06FA	FE88	CP	88H	DATA?
06FC	CCE506	CALL	Z,06E5H	SET FLAG IF SO.
06FF	FE93	CP	93H	REM?
0701	CCEF06	CALL	Z,06EFH	SET FLAG IF SO.
0704	7E	LD	A, (HL)	GET CHARACTER
0705	C3A02B	JP	2BA0H	CONTINUE
0708	218013	LD	HL,1380H	HL=> 0.5E0
070B	CDC209	CALL	09C2H	BCDE = (HL)
070E	1806	JR	0716H	ADD BCDE & ACCUM.
0710	CDC209	CALL	09C2H	BCDE = (HL)
0713	CD8209	CALL	0982H	BCDE = -BCDE
0716	78	LD	A,B	GET EXPONENT OF BCDE
0717	B7	OR	A	CHECK IT
0718	C8	RET	Z	0? ANSWER = ACCUM
0719	3A2441	LD	A, (4124H)	GET EXPONENT OF ACCUM
071C	B7	OR	A	CHECK IT
071D	CAB409	JP	Z,09B4H	0? ANSWER = BCDE
0720	90	SUB	B	TAKE EXPONENT DIFFERENCE
0721	300C	JR	NC,072FH	SKIP IF POSITIVE
0723	2F	CPL		DIFFERENCE = - DIFFERENCE
0724	3C	INC	A	
0725	EB	EX	DE,HL	SAVE DE TEMPORARILY
0726	CDA409	CALL	09A4H	PUT ACCUM ON STACK
0729	EB	EX	DE,HL	RESTORE DE
072A	CDB409	CALL	09B4H	PUT BCDE IN ACCUM
072D	C1	POP	BC	GET BCDE FROM STACK
072E	D1	POP	DE	
072F	FE19	CP	19H	IF DIFFERENCE > 24, EXIT
0731	D0	RET	NC	ANSWER = # WITH GREATER EXPONENT
0732	F5	PUSH	AF	SAVE EXPONENT DIFFERENCE
0733	CDDF09	CALL	09DFH	TURN ON MSB, BOTH NUMBERS
0736	67	LD	H,A	H = RE RUT SIGN
0737	F1	POP	AF	RESTORE - EXPONENT DIFFERENCE
0738	CDD707	CALL	07D7H	SHIFT CDE RIGHT
073B	B4	OR	H	CHECK SIGN
073C	212141	LD	HL,4121H	HL => ACCUM
073F	F25407	JP	P,0754H	SKIP IF SIGN NEGATIVE
0742	CDB707	CALL	07B7H	CDE = CDE + ACCUM
0745	D29607	JP	NC,0796H	EXIT IF NO OVERFLOW
0748	23	INC	HL	INCREASE EXPONENT
0749	34	INC	(HL)	BY 1
074A	CAB207	JP	Z,07B2H	ERROR IF EXPONENT = 0
074D	2E01	LD	L,01H	SHIFT COEFFICIENT
074F	CDEB07	CALL	07EBH	RIGHT ONE BIT
0752	1842	JR	0796H	NORMALIZE & RETURN
0754	AF	XOR	A	CLEAR STATUS
0755	90	SUB	B	CDE = ACCUM - CDE
0756	47	LD	B,A	
0757	7E	LD	A, (HL)	
0758	9B	SBC	A,E	
0759	5F	LD	E,A	
075A	23	INC	HL	
075B	7E	LD	A, (HL)	
075C	9A	SBC	A,D	
075D	57	LD	D,A	

075E	23	INC	HL	
075F	7E	LD	A, (HL)	
0760	99	SBC	A, C	
0761	4F	LD	C, A	
0762	DCC307	CALL	C, 07C3H	INVERT SIGN IF NECESSARY
0765	68	LD	L, B	SAVE B & E
0766	63	LD	H, E	IN L & H
0767	AF	XOR	A	LEFT SHIFT = 0
0768	47	LD	B, A	B = BITS LEFT SHIFTED
0769	79	LD	A, C	TEST BYTE
076A	B7	OR	A	
076B	2018	JR	NZ, 0785H	NORMALIZE IF NON-ZERO
076D	4A	LD	C, D	C <= D ROTATE
076E	54	LD	D, H	D <= H TO
076F	65	LD	H, L	H <= L NEXT
0770	6F	LD	L, A	L <= A REGISTER.
0771	78	LD	A, B	A <= B
0772	D608	SUB	08H	A = A - 8
0774	FEE0	CP	0E0H	A = -32
0776	20F0	JR	NZ, 0768H	LOOP IF NOT.
0778	AF	XOR	A	SET ACCUM
0779	322441	LD	(4124H), A	TO ZERO
077C	C9	RET		DONE
077D	05	DEC	B	COUNT # LEFT SHIFTS
077E	29	ADD	HL, HL	SHIFT HL LEFT
077F	7A	LD	A, D	SHIFT D LEFT + CARRY.
0780	17	RLA		
0781	57	LD	D, A	
0782	79	LD	A, C	SHIFT C LEFT + CARRY
0783	8F	ADC	A, A	
0784	4F	LD	C, A	
0785	F27D07	JP	P, 077DH	LOOP IF MSB = 0
0788	78	LD	A, B	A = # BITS LEFT SHIFTED.
0789	5C	LD	E, H	RESTORE E & B
078A	45	LD	B, L	
078B	B7	OR	A	NO LEFT SHIFTS?
078C	2808	JR	Z, 0796H	SKIP IF SO
078E	212441	LD	HL, 4124H	POINT TO EXPONENT
0791	86	ADD	A, (HL)	ADD # OF LEFT SHIFTS
0792	77	LD	(HL), A	SAVE NEW EXPONENT
0793	30E3	JR	NC, 0778H	
0795	C8	RET	Z	RETURN IF ZERO.
0796	78	LD	A, B	LOAD LSB
0797	212441	LD	HL, 4124H	POINT TO ACCUM
079A	B7	OR	A	IS LSB NEGATIVE?
079B	FCA807	CALL	M, 07A8H	TEST FOR OVERFLOW IF SO
079E	46	LD	B, (HL)	GET EXPONENT TO B.
079F	23	INC	HL	NEXT LOCATION
07A0	7E	LD	A, (HL)	GET SIGN
07A1	E680	AND	80H	MASK OFF IRRELEVANT BITS
07A3	A9	XOR	C	TOGGLE COEFFICIENT SIGN
07A4	4F	LD	C, A	SAVE IT.
07A5	C3B409	JP	09B4H	ACCUM = BCDE DONE.
07A8	1C	INC	E	IF CDE IS NEGATIVE
07A9	C0	RET	NZ	CONVERT ALL
07AA	14	INC	D	TRAILING ONES TO
07AB	C0	RET	NZ	ZEROS

```

07AC 0C      INC      C
07AD C0      RET      NZ
07AE 0E80   LD        C,80H      NOW CONVERT BACK
07B0 34      INC      (HL)
07B1 C0      RET      NZ          RETURN IF NO OVERFLOW
07B2 1E0A   LD        E,0AH      OV ERROR 6.
07B4 C3A219 JP        19A2H      GO TO ERROR ROUTINE
07B7 7E      LD        A,(HL)     E = E + (HL)
07B8 83      ADD      A,E
07B9 5F      LD        E,A
07BA 23      INC      HL          NEXT HL
07BB 7E      LD        A,(HL)     D = D + (HL) + CARRY
07BC 8A      ADC      A,D
07BD 57      LD        D,A
07BE 23      INC      HL          NEXT HL
07BF 7E      LD        A,(HL)     C=C+(HL + CARRY
07C0 89      ADC      A,C
07C1 4F      LD        C,A
07C2 C9      RET
    
```

NET RESULT : CDE = (HL) + CDE

```

07C3 212541 LD      HL,4125H     INVERT SIGN FLAG
07C6 7E      LD        A,(HL)
07C7 2F      CPL
07C8 77      LD        (HL),A
07C9 AF      XOR      A          CLEAR STATUS
07CA 6F      LD        L,A      L = 0
07CB 90      SUB      B          B = 0 - B
07CC 47      LD        B,A
07CD 7D      LD        A,L      E = 0 - E - CARRY
07CE 9B      SBC      A,E
07CF 5F      LD        E,A
07D0 7D      LD        A,L      D = 0 - D - CARRY
07D1 9A      SBC      A,D
07D2 57      LD        D,A
07D3 7D      LD        A,L      C = 0 - C - CARRY
07D4 99      SBC      A,C
07D5 4F      LD        C,A
07D6 C9      RET          DONE
    
```

NET RESULT : CDE = - CDE. (SIGN INVERTED)

```

07D7 0600   LD        B,00H     B = 0
07D9 D608   SUB      08H      SHIFT 8 BITS?
07DB 3807   JR        C,07E4H   BIT SHIFT IF LESS
07DD 43      LD        B,E      BYTE SHIFT B <= E
07DE 5A      LD        E,D      E <= D
07DF 51      LD        D,C      D <= C
07E0 0E00   LD        C,00H     C <= 0
07E2 18F5   JR        07D9H     LOOP UNTIL DONE
07E4 C609   ADD      A,09H     L = # BITS TO SHIFT
07E6 6F      LD        L,A
07E7 AF      XOR      A          CLEAR STATUS
07E8 2D      DEC      L          ONE LESS BIT TO SHIFT
07E9 C8      RET      Z          RETURN IF DONE
07EA 79      LD        A,C      SHIFT C RIGHT TO CARRY.
07EB 1F      RRA
07EC 4F      LD        C,A
07ED 7A      LD        A,D      SHIFT D RIGHT TO CARRY
    
```



```

07EE 1F      RRA
07EF 57      LD      D,A
07F0 7B      LD      A,E      SHIFT E RIGHT TO CARRY
07F1 1F      RRA
07F2 5F      LD      E,A
07F3 78      LD      A,B      SHIFT B RIGHT TO CARRY
07F4 1F      RRA
07F5 47      LD      B,A
07F6 18EF    JR      07E7H    LOOP UNTIL DONE.
07F8 00 00 00 81  SINGLE PRECISION 1.0
07FC 03      # LN COEFFICIENTS.
07FD AA 56 19 80  S.P. 0.5988
0801 F1 22 76 80  S.P. 0.96145
0805 45 AA 38 82  S.P. 2.88539
0809 CD5509    CALL   0955H    GET SIGN OF ACCUM
080C B7      OR      A      CHECK IT
080D EA4A1E    JP      PE,1E4AH  ERROR IF NEGATIVE
0810 212441    LD      HL,4124H  GET EXPONENT
0813 7E      LD      A,(HL)
0814 013580    LD      BC,8035H  BCDE = 0.707092
0817 11F304    LD      DE,04F3H  LOG OF 2
081A 90      SUB     B      GET DIFFERENCE IN EXPONENTS
081B F5      PUSH    AF     SAVE IT
081C 70      LD      (HL),B  MAKE EXPONENTS SAME
081D D5      PUSH    DE     SAVE BCDE
081E C5      PUSH    BC
081F CD1607    CALL   0716H    ACCUM = ACCUM + BCDE
0822 C1      POP     BC     RESTORE BCDE
0823 D1      POP     DE
0824 04      INC     B      BCDE = 1.4142E0
0825 CDA208    CALL   08A2H    ACCUM = 1.4142/ACCUM
0828 21F807    LD      HL,07F8H  HL => 1.0E0
082B CD1007    CALL   0710H    ACCUM = ACCUM +1.0
082E 21FC07    LD      HL,07FCH  HL => LOG COEFFICIENTS
0831 CD9A14    CALL   149AH    EXPAND POWER SERIES
0834 018080    LD      BC,8080H  BCDE = -0.5E0
0837 110000    LD      DE,0000H
083A CD1607    CALL   0716H    ACCUM = ACCUM - 0.5
083D F1      POP     AF     RESTORE SCALE FACTOR
083E CD890F    CALL   0F89H    SCALE ACCUM
0841 013180    LD      BC,8031H  BCDE = 0.693115E0
0844 111872    LD      DE,7218H  MULTIPLY BY ACCUM & RETURN
0847 CD5509    CALL   0955H    TEST SIGN & EXPONENT
084A C8      RET     Z      RETURN IF ZERO
084B 2E00      LD      L,00H    ADD EXPONENTS
084D CD1409    CALL   0914H
0850 79      LD      A,C      SAVE MSB OF BCDE
0851 324F41    LD      (414FH),A
0854 EB      EX     DE,HL    SAVE REST OF BCDE
0855 225041    LD      (4150H),HL
0858 010000    LD      BC,0000H  BC = 0
085B 50      LD      D,B      DE = 0
085C 58      LD      E,B
085D 216507    LD      HL,0765H  CONVERT TO SP WHEN
0860 E5      PUSH    HL      DONE
0861 216908    LD      HL,0869H  CALL 0869 THREE TIMES &
0864 E5      PUSH    HL      UNPACK ALL
    
```

0865	E5	PUSH	HL	3 BYTES OF COEFFICIENT
0866	212141	LD	HL,4121H	HL => ACCUM
0869	7E	LD	A, (HL)	GET BYTE
086A	23	INC	HL	POINT TO NEXT BYTE
086B	B7	OR	A	TEST BYTE.
086C	2824	JR	Z,0892H	JUMP IF ZERO.
086E	E5	PUSH	HL	SAVE POINTER
086F	2E08	LD	L,08H	RIGHT SHIFT 8 BITS
0871	1F	RRA		SHIFT BYTE SO CARRY
0872	67	LD	H,A	SAVE SHIFTED BYTE
0873	79	LD	A,C	PUT MSB OF BCDE IN A
0874	300B	JR	NC,0881H	SKIP IF NO ADD
0876	E5	PUSH	HL	SAVE HL
0877	2A5041	LD	HL, (4150H)	ADD SHIFTED DE
087A	19	ADD	HL,DE	TO OLD DE
087B	EB	EX	DE,HL	SAVE IN DE
087C	E1	POP	HL	RESTORE HL
087D	3A4F41	LD	A, (414FH)	ADD SHIFTED C
0880	89	ADC	A,C	TO OLD C.
0881	1F	RRA		SHIFT C RIGHT
0882	4F	LD	C,A	
0883	7A	LD	A,D	SHIFT D RIGHT
0884	1F	RRA		
0885	57	LD	D,A	
0886	7B	LD	A,E	SHIFT E RIGHT
0887	1F	RRA		
0888	5F	LD	E,A	
0889	78	LD	A,B	SHIFT B RIGHT
088A	1F	RRA		
088B	47	LD	B,A	
088C	2D	DEC	L	ONE LESS BIT TO SHIFT
088D	7C	LD	A,H	
088E	20E1	JR	NZ,0871H	LOOP UNTIL DONE.
0890	E1	POP	HL	RESTORE POINTER
0891	C9	RET		GO TO NEXT STEP
0892	43	LD	B,E	ROTATE BCDE
0893	5A	LD	E,D	LEFT ONE BYTE
0894	51	LD	D,C	
0895	4F	LD	C,A	
0896	C9	RET		
0897	CDA409	CALL	09A4H	PUT AOCUM ON STACK.
089A	21D80D	LD	HL,0DD8H	HL => 10.0E0
089D	CDB109	CALL	09B1H	PUT (HL) IN ACCUM
08A0	C1	POP	BC	LOAD BCDE
08A1	D1	POP	DE	FROM STACK
08A2	CD5509	CALL	0955H	TEST SIGN & EXPONENT
08A5	CA9A19	JP	Z,199AH	JUMP IF ERROR /0
08A8	2EFF	LD	L,0FFH	SUBTRACT EXPONENTS,
08AA	CD1409	CALL	0914H	SET MSB'S & SIGN
08AD	34	INC	(HL)	DIVIDEND =
08AE	34	INC	(HL)	DIVIDEND * 4
08AF	2B	DEC	HL	POINT TO MSB
08B0	7E	LD	A, (HL)	GET IT
08B1	328940	LD	(4089H), A	SAVE IT
08B4	2B	DEC	HL	POINT TO NMSB
08B5	7E	LD	A, (HL)	GET IT
08B6	328540	LD	(4085H), A	SAVE IT.

08B9	2B	DEC	HL	POINT TO LSB
08BA	7E	LD	A, (HL)	GET IT
08BB	328140	LD	(4081H), A	SAVE IT
08BE	41	LD	B, C	B = MSB OF DIVISOR
08BF	EB	EX	DE, HL	HL = ISB, DE => ACCUM
08C0	AF	XOR	A	SET REGISTER
08C1	4F	LD	C, A	VALUE TO ZERO
08C2	57	LD	D, A	
08C3	5F	LD	E, A	
08C4	328C40	LD	(408CH), A	OVERFLOW COUNT = 0
08C7	E5	PUSH	HL	SAVE OLD REGISTER VALUE (BHL).
08C8	C5	PUSH	BC	
08C9	7D	LD	A, L	GET OLD REGISTER LSB
08CA	CD8040	CALL	4080H	BHL = OHL - ACCUM
08CD	DE00	SBC	A, 00H	OVERFLOW = OVERFLOW - CARRY
08CF	3F	CCF		INVERT CARRY FLAG
08D0	3007	JR	NC, 08D9H	SKIP IF ACCUM ≥ BHL
08D2	328C40	LD	(408CH), A	SAVE OVERFLOW COUNT
08D5	F1	POP	AF	DISCARD OLD BHL
08D6	F1	POP	AF	
08D7	37	SCF		MASK TO 08DBH
08D8	D2			
08D9	C1	POP	BC	RESTORE OLD BHL
08DA	E1	POP	HL	
08DB	79	LD	A, C	TEST FOR OVERFLOW
08DC	3C	INC	A	INTO C
08DD	3D	DEC	A	
08DE	1F	RRA		
08DF	FA9707	JP	M, 0797H	NORMALIZE & RETURN IF SO.
08E2	17	RLA		SHIFT CDE LEFT
08E3	7B	LD	A, E	
08E4	17	RLA		
08E5	5F	LD	E, A	
08E6	7A	LD	A, D	
08E7	17	RLA		
08E8	57	LD	D, A	
08E9	79	LD	A, C	
08EA	17	RLA		
08EB	4F	LD	C, A	
08EC	29	ADD	HL, HL	SHIFT BHL LEFT
08ED	78	LD	A, B	
08EE	17	RLA		
08EF	47	LD	B, A	
08F0	3A8C40	LD	A, (408CH)	UPDATE OVERFLOW
08F3	17	RLA		COUNTER
08F4	328C40	LD	(408CH), A	
08F7	79	LD	A, C	TEST CDE (DIVISOR)
08F8	B2	OR	D	
08F9	B3	OR	E	
08FA	20CB	JR	NZ, 08C7H	LOOP IF NON-ZERO
08FC	E5	PUSH	HL	SAVE HL
08FD	212441	LD	HL, 4124H	DECREMENT EXPONENT
0900	35	DEC	(HL)	OF ACCUM
0901	E1	POP	HL	RESTORE HL
0902	20C3	JR	NZ, 08C7H	LOOP IF EXPONENT GOOD.
0904	C3B207	JP	07B2H	EXPONENT ZERO ERROR.
0907	3EFF	LD	A, 0FFH	SET TO NEGATE SIGN

0909	2E			MAKE TO 090BH
0909	AF	XOR	A	
090B	212D41	LD	HL,412DH	HL => MSB OF ACCUM 2
090E	4E	LD	C, (HL)	C =MSB
090F	23	INC	HL	HL => EXPONENT OF ACCUM 2
0910	AE	XOR	(HL)	MAKE IT + OR - AS REQUIRED
0911	47	LD	B,A	B = NEW EXPONENT
0912	2E00	LD	L,00H	L = MASK
0914	78	LD	A,B	GET EXPONENT
0915	B7	OR	A	CHECK IT
0916	281F	JR	Z,0937H	EXIT IF ZERO
0918	7D	LD	A,L	GET MASK
0919	212441	LD	HL,4124H	HL => EXPONENT OF ACCUM
091C	AE	XOR	(HL)	CHANGE SIGN AS NECESSARY
091D	80	ADD	A,B	ADD EXPONENTS
091E	47	LD	B,A	SAVE B
091F	1F	RRA		TEST FOR CARRY
0920	A8	XOR	B	
0921	78	LD	A,B	RESTORE EXPONENT TO A
0922	F23609	JP	P,0936H	EXIT IF OUT OF RANGE
0925	C680	ADD	A,80H	SET BIT 7
0927	77	LD	(HL),A	SAVE NEW EXPONENT
0928	CA9008	JP	Z,0890H	JUMP IF 0
092B	CDDF09	CALL	09DFH	TURN ON MSB
092E	77	LD	(HL),A	SAVE EXPONENT
092F	2B	DEC	HL	HL => MSB
0930	C9	RET		DONE
0931	CD5509	CALL	0955H	GET SIGN OF ACCUM
0934	2F	CPL		INVERT IT
0935	E1	POP	HL	FIX STACK
0936	B7	OR	A	SET STATUS FLAGS
0937	E1	POP	HL	FIX STACK
0938	F27807	JP	P,0778H	SET TO ZERO IF NECESSARY
093B	C3B207	JP	07B2H	OV ERROR
093E	CDBF09	CALL	09BFH	LOAD ACCUM TO BCDE
0941	78	LD	A,B	GET EXPONENT
0942	B7	OR	A	TEST IT
0943	C8	RET	Z	RETURN IF ZERO
0944	C602	ADD	A,02H	BCDE = BCDE * 4
0946	DAB207	JP	C,07B2H	OVERFLOW ERROR IF TOO LARGE
0949	47	LD	B,A	SAVE EXPONENT
094A	CD1607	CALL	0716H	ACCUM = ACCUM + BCDE * 4
094D	212441	LD	HL,4124H	POINT TO EXPONENT
0950	34	INC	(HL)	ACCUM = ACCUM * 2
0951	C0	RET	NZ	RETURN IF NO OVERFLOW
0952	C3B207	JP	07B2H	ERROR IF OVERFLOW
0955	3A2441	LD	A,(4124H)	GET EXPONENT OF ACCUM
0958	B7	OR	A	TEST IT
0959	C8	RET	Z	RETURN 0 IF ZERO
095A	3A2341	LD	A,(4123H)	GET MSB OF ACCUM
095D	FE			MASK TO 095FH
095E	2F	CPL		INVERT
095F	17	RLA		CARRY = SIGN BIT
0960	9F	SBC	A,A	A = 0 - CARRY
0961	C0	RET	NZ	RETURN - 1 IF NEGATIVE
0962	3C	INC	A	+ 1 IF POSITIVE
0963	C9	RET		DONE

NET RESULT:
 GIVEN A SINGLE OR DOUBLE
 PRECISION NUMBER IN
 ACCUM, THIS ROUTINE RETURNS
 A = SGN (ACCUM) .

0964	0688	LD	B,88H	B = INITIAL EXPONENT
0966	110000	LD	DE,0000H	INITIALIZE DE
0969	212441	LD	HL,4124H	HL => EXPONENT OF ACCUM
096C	4F	LD	C,A	C = MSB OF INTEGER
096D	70	LD	(HL),B	SAVE EXPONENT
096E	0600	LD	B,00H	INITIALIZE B
0970	23	INC	HL	POINT TO SIGN BYTE
0971	3680	LD	(HL),80H	SIGN = +
0973	17	RLA		PUT SIGN IN CARRY
0974	C36207	JP	0762H	NORMALIZE & RETURN
0977	CD9409	CALL	0994H	GET SIGN OF ACCUM
097A	F0	RET	P	RETURN IF POSITIVE
097B	E7	RST	20H	GET TYPE OF ACCUM
097C	FA5B0C	JP	M,0C5BH	JUMP IF INTEGER
097F	CAF60A	JP	Z,0AF6H	ERROR IF STRING
0982	212341	LD	HL,4123H	MAKE FLOATING POINT
0985	7E	LD	A,(HL)	NUMBER POSITIVE
0986	EE80	XOR	80H	
0988	77	LD	(HL),A	
0989	C9	RET		DONE
098A	CD9409	CALL	0994H	GET SIGN OF ACCUM
098D	6F	LD	L,A	PUT IT IN L.
098E	17	RLA		PUT SIGN IN CARRY
098F	9F	SBC	A,A	H = 0 OR -1
0990	67	LD	H,A	ACCORDING TO SIGN
0991	C39A0A	JP	0A9AH	PUT HL IN ACCUM
0994	E7	RST	20H	FIND DATA TYPE
0995	CAF60A	JP	Z,0AF6H	ERROR - STRING JUMP.
0998	F25509	JP	P,0955H	JUMP IF FLOATING POINT
099B	2A2141	LD	HL,(4121H)	GET INTEGER IN ACCUM
099E	7C	LD	A,H	TEST IF ZERO
099F	B5	OR	L	
09A0	C8	RET	Z	RETURN IF SO
09A1	7C	LD	A,H	FIX ACCUMULATOR
09A2	18BB	JR	095FH	DETERMINE SIGN & RETURN
09A4	EB	EX	DE,HL	SAVE HL FOR NOW
09A5	2A2141	LD	HL,(4121H)	GET LSB & MSB OF ACCUM
09A8	E3	EX	(SP),HL	GET RETURN & SAVE DATA
09A9	E5	PUSH	HL	SAVE RETURN
09AA	2A2341	LD	HL,(4123H)	GET MSB & EXPONENT OF ACCUM
09AD	E3	EX	(SP),HL	GET RETURN & SAVE DATA
09AE	E5	PUSH	HL	SAVE RETURN
09AF	EB	EX	DE,HL	RESTORE HL
09B0	C9	RET		DONE
09B1	CDC209	CALL	09C2H	BCDE = (HL)
09B4	EB	EX	DE,HL	SAVE HL. STORE DE
09B5	222141	LD	(4121H),HL	IN ACCUM
09B8	60	LD	H,B	STORE BC
09B9	69	LD	L,C	IN
09BA	222341	LD	(4123H),HL	ACCUM

09BD	EB	EX	DE,HL	RESTORE HL
09BE	C9	RET		DONE
09BF	212141	LD	HL,4121H	HL => ACCUM
09C2	5E	LD	E, (HL)	BCDE = (HL)
09C3	23	INC	HL	
09C4	56	LD	D, (HL)	
09C5	23	INC	HL	
09C6	4E	LD	C, (HL)	
09C7	23	INC	HL	
09C8	46	LD	B, (HL)	
09C9	23	INC	HL	
09CA	C9	RET		DONE
09CB	112141	LD	DE,4121H	MOVE FROM ACCUM
09CE	0604	LD	B,04H	FOUR BYTES
09D0	1805	JR	09D7H	MOVE IT
09D2	EB	EX	DE,HL	SOURCE & DESTINATION GIVEN
09D3	3AAF40	LD	A,(40AFH)	GET DATA LENGTH
09D6	47	LD	B,A	PUT IT IN B
09D7	1A	LD	A,(DE)	GET DATA
09D8	77	LD	(HL),A	SAVE DATA
09D9	13	INC	DE	POINT TO NEXT SOURCE
09DA	23	INC	HL	POINT TO NEXT DESTINATION
09DB	05	DEC	B	ONE LESS BYTE TO MOVE
09DC	20F9	JR	NZ,09D7H	LOOP UNTIL DONE
09DE	C9	RET		DONE
09DF	212341	LD	HL,4123H	POINT TO MSB OF ACCUM
09E2	7E	LD	A,(HL)	GET MSB
09E3	07	RLCA		CARRY = COEFFICIENT SIGN
09E4	37	SCF		SET HIGHEST BIT
09E5	1F	RRA		SAVE BIT & RESTORE SIGN
09E6	77	LD	(HL),A	SAVE NEW MSB
09E7	3F	CCF		INVERT OLD SIGN
09E8	1F	RRA		PUT IN BIT 7
09E9	23	INC	HL	HL => SIGN BYTE
09EA	23	INC	HL	
09EB	77	LD	(HL),A	SAVE NEW SIGN
09EC	79	LD	A,C	GET MSB OF ACCUM
09ED	07	RLCA		CARRY = COEFFICIENT SIGN
09EE	37	SCF		SET HIGHEST BIT
09EF	1F	RRA		SAVE BIT AND RESTORE SIGN
09F0	4F	LD	C,A	SAVE NOR MSB
09F1	1F	RRA		PUT SIGN IN BIT 7
09F2	AE	XOR	(HL)	COMPUTE COMBINED SIGN
09F3	C9	RET		DONE
09F4	212741	LD	HL,4127H	MOVE DATA FROM
09F7	11D209	LD	DE,09D2H	ACCUM 2
09FA	1806	JR	0A02H	TO ACCUM
09FC	212741	LD	HL,4127H	NONE DATA TO
09FF	11D309	LD	DE,09D3H	ACCUM 2 FROM ACCUM
0A02	D5	PUSH	DE	SAVE RETURN ADDRESS
0A03	112141	LD	DE,4121H	FROM/TO ACCUM
0A06	E7	RST	20H	CHECK DATA TYPE
0A07	D8	RET	C	CONTINUE IF NOT D.P.
0A08	111D41	LD	DE,411DH	ADJUST TO D.P.
0A0B	C9	RET		CONTINUE
0A0C	78	LD	A,B	REGISTER VALUE 0?
0A0D	B7	OR	A	

0A0E	CA5509	JP	Z,0955H	RETURN SIGN OF ACCUM
0A11	215E09	LD	HL,095EH	SET RETURN THROUGH
0A14	E5	PUSH	HL	SIGN ROUTINE
0A15	CD5509	CALL	0955H	GET SIGN OF ACCUM
0A18	79	LD	A,C	GET MSB OF BCDE
0A19	C8	RET	Z	RETURN IF NOT S.P.
0A1A	212341	LD	HL,4123H	HL => ACCUM MSB
0A1D	AE	XOR	(HL)	COMPARE MSBS
0A1E	79	LD	A,C	FIX MSB
0A1F	F8	RET	M	EXIT IF DIFFERENT SIGNS
0A20	CD260A	CALL	0A26H	COMPARE BCDE & ACCUM
0A23	1F	RRA		GET CARRY FROM COMPARISON
0A24	A9	XOR	C	COMBINE WITH BCDE SIGN
0A25	C9	RET		CONTINUE
0A26	23	INC	HL	HL => EXPONENT OF ACCUM
0A27	78	LD	A,B	COMPARE EXPONENTS
0A28	BE	CP	(HL)	B - (HL)
0A29	C0	RET	NZ	RETURN ANSWER IF NON-ZERO
0A2A	2B	DEC	HL	HL => MSB OF ACCUM
0A2B	79	LD	A,C	COMPARE: C - (HL)
0A2C	BE	CP	(HL)	
0A2D	C0	RET	NZ	RETURN ANSWER IF NON-ZERO
0A2E	2B	DEC	HL	HL => MSB OF ACCUM
0A2F	7A	LD	A,D	COMPARE: D - (HL)
0A30	BE	CP	(HL)	
0A31	C0	RET	NZ	RETURN ANSWER IF NON-ZERO
0A32	2B	DEC	HL	HL => LSB OF ACCUM
0A33	7B	LD	A,E	COMPARE: E - (HL)
0A34	96	SUB	(HL)	
0A35	C0	RET	NZ	RETURN ANSWER IF NON-ZERO
0A36	E1	POP	HL	SKIP SIGN ROUTINE
0A37	E1	POP	HL	
0A38	C9	RET		RETURN 0 FOR MATCH
0A39	7A	LD	A,D	COMPARE SIGN OF
0A3A	AC	XOR	H	DE & HL
0A3B	7C	LD	A,H	PUT H IN A
0A3C	FA5F09	JP	M,095FH	EXIT A = -1 (SIGNS UNLIKE)
0A3F	BA	CP	D	COMPARE D & H
0A40	C26009	JP	NZ,0960H	RETURN SIGN IF UNLIKE
0A43	7D	LD	A,L	COMPARE E & L
0A44	93	SUB	E	
0A45	C26009	JP	NZ,0960H	RETURN SIGN IF UNLIKE
0A48	C9	RET		RETURN 0 IF ALIKE
0A49	212741	LD	HL,4127H	MOVE GIVEN DATA
0A4C	CDD309	CALL	09D3H	TO ACCUM 2
0A4F	112E41	LD	DE,412EH	TEST D.P. EXPONENT
0A52	1A	LD	A,(DE)	OF ACCUM 2
0A53	B7	OR	A	
0A54	CA5509	JP	Z,0955H	0? RESULT IS SIGN OF ACCUM
0A57	215E09	LD	HL,095EH	RETURN THROUGH
0A5A	E5	PUSH	HL	SIGN ROUTINE
0A5B	CD5509	CALL	0955H	GET SIGN OF ACCUM
0A5E	1B	DEC	DE	GET MSB OF ACCUM2
0A5F	1A	LD	A,(DE)	
0A60	4F	LD	C,A	SAVE IT
0A61	C8	RET	Z	ACCUM = 0? RETURN SIGN OF ACCUM 2
0A62	212341	LD	HL,4123H	POINT TO MSB OF ACCUM

0A65	AE	XOR	(HL)	COMPARE SIGNS
0A66	79	LD	A,C	RESTORE MSB OF ACCUM 2
0A67	F8	RET	M	RETURN -1 IF SIGNS DIFFERENT
0A68	13	INC	DE	POINT TO
0A69	23	INC	HL	EXPONENTS
0A6A	0608	LD	B,08H	COMPARE 8 BYTES
0A6C	1A	LD	A,(DE)	COMPARE BYTES
0A6D	96	SUB	(HL)	
0A6E	C2230A	JP	NZ,0A23H	CONTINUE IF DIFFERENT
0A71	1B	DEC	DE	POINT TO
0A72	2B	DEC	HL	NEXT BYTES
0A73	05	DEC	B	ONE LESS TO COMPARE
0A74	20F6	JR	NZ,0A6CH	LOOP UNTIL DONE
0A76	C1	POP	BC	EXIT DIRECTLY TO CALLER
0A77	C9	RET		DONE
0A78	CD4F0A	CALL	0A4FH	COMPARE ACCUM & ACCUM2
0A7B	C25E09	JP	NZ,095EH	SET FLAGS IN NOT SAME
0A7E	C9	RET		DONE
0A7F	E7	RST	20H	GET TYPE OF ACCUM
0A80	2A2141	LD	HL,(4121H)	HL = INTEGER VALUE
0A83	F8	RET	M	RETURN IF INTEGER
0A84	CAF60A	JP	Z,0AF6H	TM ERROR IF STRING
0A87	D4B90A	CALL	NC,0AB9H	CONVERT D.P. TO S.P.
0A8A	21B207	LD	HL,07B2H	RETURN THROUGH
0A8D	E5	PUSH	HL	OV ERROR ROUTINE
0A8E	3A2441	LD	A,(4124H)	GET EXPONENT
0A91	FE90	CP	90H	SPECIAL CASE?
0A93	300E	JR	NC,0AA3H	HANDLE IT IF NECESSARY
0A95	CDFB0A	CALL	0AFBH	CONVERT S.P. TO INTEGER
0A98	EB	EX	DE,HL	PUT INTEGER IN HL
0A99	D1	POP	DE	ELIMINATE OV ERROR EXIT
0A9A	222141	LD	(4121H),HL	SAVE INTEGER VALUE
0A9D	3E02	LD	A,02H	TYPE = INTEGER
0A9F	32AF40	LD	(40AFH),A	SAVE TYPE FLAG
0AA2	C9	RET		DONE
0AA3	018090	LD	BC,9080H	IS ACCUM = -32768?
0AA6	110000	LD	DE,0000H	
0AA9	CD0C0A	CALL	0A0CH	
0AAC	C0	RET	NZ	OV ERROR IF NOT
0AAD	61	LD	H,C	HL = -32768
0AAE	6A	LD	L,D	
0AAF	18E8	JR	0A99H	CONTINUE
0AB1	E7	RST	20H	GET TYPE OF ACCUM
0AB2	E0	RET	PO	RETURN IF ALREADY S.P.
0AB3	FACC0A	JP	M,0ACCH	JUMP IF INTEGER
0AB6	CAF60A	JP	Z,0AF6H	TM ERROR IF STRING
0AB9	CDBF09	CALL	09BFH	BCDE = ACCUM
0ABC	CDEF0A	CALL	0AEFH	TYPE =SINGLE PRECISION
0ABF	78	LD	A,B	ZERO?
0AC0	B7	OR	A	
0AC1	C8	RET	Z	RETURN IF SO
0AC2	CDDF09	CALL	09DFH	UNPACK BCDE
0AC5	212041	LD	HL,4120H	GET 4TH MOST
0AC8	46	LD	B,(HL)	SIGNIFICANT BYTE
0AC9	C39607	JP	0796H	NORMALIZE & RETURN
0ACC	2A2141	LD	HL,(4121H)	GET INTEGER VALUE

0ACF	CDEF0A	CALL	0AEFH	SET TO SINGLE PRECISION
0AD2	7C	LD	A,H	SET UP
0AD3	55	LD	D,L	FOR
0AD4	1E00	LD	E,00H	CONVERSION
0AD6	0690	LD	B,90H	INITIAL EXPONENT
0AD8	C36909	JP	0969H	FINISH CONVERSION & RETURN
0ADB	E7	RST	20H	GET ACCUM TYPE
0ADC	D0	RET	NC	RETURN IF ALREADY D.P.
0ADD	CAF60A	JP	Z,0AF6H	TM ERROR IF STRING
0AE0	FCC00A	CALL	M,0ACCH	CONVERT INTEGER
0AE3	210000	LD	HL,0000H	ZERO THE FOUR
0AE6	221D41	LD	(411DH),HL	LEAST SIGNIFICANT BYTE
0AE9	221F41	LD	(411FH),HL	OF THE MANTISSA
0AEC	3E08	LD	A,08H	DOUBLE PRECISION
0AEE	01			MASK TO 0AF1H
0AEF	3E04	LD	A,04H	SINGLE PRECISION
0AF1	C39F0A	JP	0A9FH	SAVE TYPE
0AF4	E7	RST	20H	GET TYPE OF ACCUM
0AF5	C8	RET	Z	RETURN IF STRING
0AF6	1E18	LD	E,18H	TM ERROR 13
0AF8	C3A219	JP	19A2H	GO TO ERROR ROUTINE
0AFB	47	LD	B,A	CLEAR BCDE TO
0AFC	4F	LD	C,A	EXPONENT BYTES
0AFD	57	LD	D,A	
0AFE	5F	LD	E,A	
0AFF	B7	OR	A	WAS EXPONENT ZERO?
0B00	C8	RET	Z	RETURN IF SO.
0B01	E5	PUSH	HL	SAVE HL
0B02	CDBF09	CALL	09BFH	BCDE = ACCUM
0B05	CDDF09	CALL	09DFH	UNPACK BCDE
0B08	AE	XOR	(HL)	ADJUST SIGN
0B09	67	LD	H,A	SAVE SIGN
0B0A	FC1F0B	CALL	M,0B1FH	HANDLE NEGATIVE SIGN
0B0D	3E98	LD	A,98H	MAXIMUM EXPONENT
0B0F	90	SUB	B	SUBTRACT REAL EXPONENT
0B10	CDD707	CALL	07D7H	CONVERT CDE TO INTEGER
0B13	7C	LD	A,H	RESTORE ORIGINAL SIGN
0B14	17	RLA		PUT IT INTO CARRY
0B15	DCA807	CALL	C,07A8H	CONVERT IF NEGATIVE
0B18	0600	LD	B,00H	SET B TO 0
0B1A	DCC307	CALL	C,07C3H	INVERT SIGN IF NECESSARY
0B1D	E1	POP	HL	RESTORE HL
0B1E	C9	RET		DONE
0B1F	1B	DEC	DE	DE = DE -1
0B20	7A	LD	A,D	DE = FFFF?
0B21	A3	AND	E	
0B22	3C	INC	A	
0B23	C0	RET	NZ	RETURN IF NOT
0B24	0B	DEC	BC	ROUND DOWN
0B25	C9	RET		DONE
		FIX		
0B26	E7	RST	20H	CHECK TYPE OF ACCUM
0B27	F8	RET	M	RETURN IF INTEGER
0B28	CD5509	CALL	0955H	CHECK SIGN
0B2B	F2370B	JP	P,0B37H	SKIP IF POSITIVE TO INT
0B2E	CD8209	CALL	0982H	MAKE POSITIVE

0B31	CD370B	CALL	0B37H	ACCUM = INT(ACCUM)
0B34	C37B09	JP	097BH	MAKE NEGATIVE & RETURN
<u>INT</u>				
0B37	E7	RST	20H	CHECK TYPE OF ACCUM
0B38	F8	RET	M	RETURN IF INTEGER
0B39	301E	JR	NC,0B59H	SKIP IF POSITIVE
0B3B	28B9	JR	Z,0AF6H	TM ERROR IF STRING
0B3D	CD8E0A	CALL	0A8EH	ACCUM = CINT (ACCUM)
0B40	212441	LD	HL,4124H	RETURN NO OV ERROR
0B43	7E	LD	A,(HL)	GET EXPONENT OF ACCUM
0B44	FE98	CP	98H	IS IT ≥ 98H?
0B46	3A2141	LD	A,(4121H)	GET LSB OF ACCUM
0B49	D0	RET	NC	RETURN IF ≥ 98H
0B4A	7E	LD	A,(HL)	GET EXPONENT OF ACCUM
0B4B	CDFB0A	CALL	0AFBH	CONVERT TO INTEGER
0B4E	3698	LD	(HL),98H	SAVE NEW EXPONENT
0B50	7B	LD	A,E	GET INTEGER PART
0B51	F5	PUSH	AF	SAVE IT
0B52	79	LD	A,C	GET SIGN
0B53	17	RLA		
0B54	CD6207	CALL	0762H	APPLY SIGN & NORMALIZE
0B57	F1	POP	AF	RESTORE INTEGER
0B58	C9	RET		DONE
0B59	212441	LD	HL,4124H	GET EXPONENT OF
0B5C	7E	LD	A,(HL)	ACCUM
0B5D	FE90	CP	90H	SMALL ENOUGH FOR INTEGER?
0B5F	DA7F0A	JP	C,0A7FH	INTEGER IF SO
0B62	2014	JR	NZ,0B78H	JUMP IF > 2 * * 16
0B64	4F	LD	C,A	EXPONENT = +16 SAVE IT
0B65	2B	DEC	HL	GET MSB
0B66	7E	LD	A,(HL)	
0B67	EE80	XOR	80H	INVERT SIGN BIT
0B69	0606	LD	B,06H	SCAN NEXT 6 BYTES
0B6B	2B	DEC	HL	NEXT BYTE
0B6C	B6	OR	(HL)	COMBINE WITH MSB
0B6D	05	DEC	B	ONE LESS BYTE
0B6E	20FB	JR	NZ,0B6BH	LOOP UNTIL DONE
0B70	B7	OR	A	ZERO?
0B71	210080	LD	HL,8000H	HL = -32768
0B74	CA9A0A	JP	Z,0A9AH	INTEGER IF SO
0B77	79	LD	A,C	RESTORE EXPONENT
0B78	FEB8	CP	0B8H	TOO LARGE TO FIX?
0B7A	D0	RET	NC	RETURN IF SO
0B7B	F5	PUSH	AF	SAVE EXPONENT & STATUS
0B7C	CDBF09	CALL	09BFH	BCDE = ACCUM
0B7F	CDDF09	CALL	09DFH	UNPACK BCDE
0B82	AE	XOR	(HL)	TEST SIGN
0B83	2B	DEC	HL	POINT TO EXPONENT
0B84	36B8	LD	(HL),0B8H	EXPONENT = + 56.(MAX)
0B86	F5	PUSH	AF	SAVE SIGN
0B87	FCA00B	CALL	M,0BA0H	CONVERT TRAILING 1'S IF -
0B8A	212341	LD	HL,4123H	POINT TO MSB
0B8D	3EB8	LD	A,0B8H	A = MAX EXPONENT
0B8F	90	SUB	B	SUBTRACT REAL EXPONENT
0B90	CD690D	CALL	0D69H	RIGHT JUSTIFY MANTISSA

0B93	F1	POP	AF	RESTORE SIGN
0B94	FC200D	CALL	M,0D20H	CONVERT TRAILING 0'S IF -
0B97	AF	XOR	A	CLEAR A
0B98	321C41	LD	(411CH),A	CLEAR MANTISSA SIGN
0B9B	F1	POP	AF	RESTORE STATUS
0B9C	D0	RET	NC	EXIT IF TOO LARGE
0B9D	C3D80C	JP	0CD8H	NORMALIZE & RETURN

CONVERT TRAILING ONES TO ZEROES (D.P.)

0BA0	211D41	LD	HL,411DH	POINT TO LSB
0BA3	7E	LD	A,(HL)	GET BYTE
0BA4	35	DEC	(HL)	ADJUST BYTE
0BA5	B7	OR	A	TEST PREVIOUS BYTE
0BA6	23	INC	HL	POINT TO NEXT
0BA7	28FA	JR	Z,0BA3H	LOOP IF NOT DONE
0BA9	C9	RET		DONE

DE = DE * BS ERROR IF OVERFLOW

0BAA	E5	PUSH	HL	SAVE HL
0BAB	210000	LD	HL,0000H	RESULT = 0
0BAE	78	LD	A,B	BC = 0?
0BAF	B1	OR	C	
0BB0	2812	JR	Z,0BC4H	RESULT = 0 IF SO, EXIT.
0BB2	3E10	LD	A,10H	SHIFT 16 TIMES
0BB4	29	ADD	HL,HL	SHIFT RESULT LEFT
0BB5	DA3D27	JP	C,273DH	BS ERROR IF OVERFLOW
0BB8	EB	EX	DE,HL	SHIFT DE LEFT
0BB9	29	ADD	HL,HL	
0BBA	EB	EX	DE,HL	
0BBB	3004	JR	NC,0BC1H	SKIP IF NO ADD
0BBD	09	ADD	HL,BC	RESULT = RESULT + BC
0BBE	DA3D27	JP	C,273DH	BS ERROR IF OVERFLOW
0BC1	3D	DEC	A	ONE LESS SHIFT TO DO
0BC2	20F0	JR	NZ,0BB4H	LOOP UNTIL DONE
0BC4	EB	EX	DE,HL	RESULT IN DE
0BC5	E1	POP	HL	RESTORE HL
0BC6	C9	RET		DONE

DE = DE - HL

0BC7	7C	LD	A,H	B = SIGN OF HL
0BC8	17	RLA		
0BC9	9F	SBC	A,A	
0BCA	47	LD	B,A	
0BCB	CD510C	CALL	0C51H	HL = -HL
0BCE	79	LD	A,C	A = -SIGN OF HL
0BCF	98	SBC	A,B	
0BD0	1803	JR	0BD5H	ADD & RETURN

DE = DE + HL

0BD2	7C	LD	A,H	A = SIGN OF HL
0BD3	17	RLA		
0BD4	9F	SBC	A,A	
0BD5	47	LD	B,A	B = SIGN OF HL
0BD6	E5	PUSH	HL	SAVE HL
0BD7	7A	LD	A,D	A = SIGN OF DE
0BD8	17	RLA		
0BD9	9F	SBC	A,A	

0BDA 19	ADD	HL,DE	HL = HL + DE
0BDB 88	ADC	A,B	TEST FOR OVERFLOW
0BDC 0F	RRCA		
0BDD AC	XOR	H	
0BDE F2990A	JP	P,0A99H	EXIT IF NONE HL = RESULT
0BE1 C5	PUSH	BC	SAVE SIGN OF HL
0BE2 EB	EX	DE,HL	SWAP DE & HL
0BE3 CDCF0A	CALL	0ACFH	ACCUM = CSNG(HL)
0BE6 F1	POP	AF	CLEAR STACK
0BE7 E1	POP	HL	RESTORE HL
0BE8 CDA409	CALL	09A4H	PUT ACCUM ON STACK
0BEB EB	EX	DE,HL	PUT OLD HL IN DE
0BEC CD6B0C	CALL	0C6BH	ACCUM = CSNG(DE)
0BEF C38F0F	JP	0F8FH	ADD & RETURN.

DE = DE*HL

0BF2 7C	LD	A,H	HL = 0
0BF3 B5	OR	L	
0BF4 CA9A0A	JP	Z,0A9AH	RESULT = 0 IF SO
0BF7 E5	PUSH	HL	SAVE ORIGINAL
0BF8 D5	PUSH	DE	VALUES JUST IN CASE
0BF9 CD450C	CALL	0C45H	RESULT INTEGER B = SIGN
0BFC C5	PUSH	BC	SAVE SIGN OF RESULT
0BFD 44	LD	B,H	BC = HL
0BFE 4D	LD	C,L	
0BFF 210000	LD	HL,0000H	HL = RESULT = 0
0C02 3E10	LD	A,10H	16 SHIFTS TO DO
0C04 29	ADD	HL,HL	SHIFT RESULT LEFT
0C05 381F	JR	C,0C26H	JUMP OUT IF OVERFLOW
0C07 EB	EX	DE,HL	SHIFT TO THE LEFT
0C08 29	ADD	HL,HL	
0C09 EB	EX	DE,HL	
0C0A 3004	JR	NC,0C10H	SKIP IF NO ADD
0C0C 09	ADD	HL,BC	RESULT = RESULT + BC
0C0D DA260C	JP	C,0C26H	JUMP OUT IF OVERFLOW
0C10 3D	DEC	A	ONE LESS SHIFT TO DO
0C11 20F1	JR	NZ,0C04H	LOOP UNTIL DONE
0C13 C1	POP	BC	RESTORE SIGN OF RESULT
0C14 D1	POP	DE	RESTORE DE
0C15 7C	LD	A,H	CHECK SIGN OF RESULT
0C16 B7	OR	A	BEFORE ADJUSTING
0C17 FA1F0C	JP	M,0C1FH	SKIP IF NEGATIVE
0C1A D1	POP	DE	CLEAR STACK
0C1B 78	LD	A,B	GET SIGN OF RESULT
0C1C C34D0C	JP	0C4DH	SAVE RESULT & EXIT
0C1F EE80	XOR	80H	INVERT SIGN OF RESULT
0C21 B5	OR	L	ZERO?
0C22 2813	JR	Z,0C37H	SKIP IF SO
0C24 EB	EX	DE,HL	SAVE RESULT IN DE
0C25 01			
0C26 C1	POP	BC	RESTORE COMBINED SIGN
0C27 E1	POP	HL	RESTORE OLD DE TO HL
0C28 CDCF0A	CALL	0ACFH	ACCUM = CSNG(HL)
0C2B E1	POP	HL	RESTORE OLD HL
0C2C CDA409	CALL	09A4H	PUT ACCUM ON STACK
0C2F CDCF0A	CALL	0ACFH	ACCUM = CSNG(HL)
0C32 C1	POP	BC	LOAD BCDE

0C33	D1	POP	DE	FROM STACK
0C34	C34708	JP	0847H	MULTIPLY & RETURN
0C37	78	LD	A,B	GET SIGN OF RESULT
0C38	B7	OR	A	TEST IT
0C39	C1	POP	BC	CLEAR STACK
0C3A	FA9A0A	JP	M,0A9AH	RETURN VALUE IF ALREADY NEGATIVE
0C3D	D5	PUSH	DE	SAVE DE
0C3E	CDCF0A	CALL	0ACFH	ACCUM = CSNG (RESULT)
0C41	D1	POP	DE	RESTORE DE
0C42	C38209	JP	0982H	MAKE IT NEGATIVE & RETURN
0C45	7C	LD	A,H	COMPUTE COMBINED
0C46	AA	XOR	D	SIGN & SAVE IN B
0C47	47	LD	B,A	
0C48	CD4C0C	CALL	0C4CH	MAKE HL POSITIVE
0C4B	EB	EX	DE,HL	MAKE DE POSITIVE
0C4C	7C	LD	A,H	GET MSB OF HL
0C4D	B7	OR	A	TEST IT
0C4E	F29A0A	JP	P,0A9AH	ACCUM = HL INTEGER
0C51	AF	XOR	A	A = 0
0C52	4F	LD	C,A	C = 0
0C53	95	SUB	L	HL = -HL
0C54	6F	LD	L,A	
0C55	79	LD	A,C	
0C56	9C	SBC	A,H	
0C57	67	LD	H,A	
0C58	C39A0A	JP	0A9AH	ACCUM=HL INTEGER
0C5B	2A2141	LD	HL,(4121H)	GET HL FROM ACCUM
0C5E	CD510C	CALL	0C51H	HL = -HL. C=0
0C61	7C	LD	A,H	INVERT SIGN OF HL
0C62	EE80	XOR	80H	
0C64	B5	OR	L	HL ZERO?
0C65	C0	RET	NZ	RETURN IF NOT
0C66	EB	EX	DE,HL	PUT IN DE
0C67	CDEF0A	CALL	0AEFH	SET TYPE S.P.
0C6A	AF	XOR	A	CLEAR STATUS
0C6B	0698	LD	B,98H	B = NEW EXPONENT
0C6D	C36909	JP	0969H	NORMALIZE & RETURN

DOUBLE PRECISION SUBTRACTION

0C70	212D41	LD	HL,412DH	INVERT
0C73	7E	LD	A,(HL)	SIGN
0C74	EE80	XOR	80H	OF
0C76	77	LD	(HL),A	ACCUM 2

DOUBLE PRECISION ADDITION

0C77	212E41	LD	HL,412EH	ACCUM 2 = 0?
0C7A	7E	LD	A,(HL)	
0C7B	B7	OR	A	
0C7C	C8	RET	Z	RETURN IF SO
0C7D	47	LD	B,A	SAVE ACCUM 2 EXPONENT
0C7E	2B	DEC	HL	GET MSB OF
0C7F	4E	LD	C,(HL)	ACCUM 2
0C80	112441	LD	DE,4124H	ACCUM = 0
0C83	1A	LD	A,(DE)	
0C84	B7	OR	A	
0C85	CAF409	JP	Z,09F4H	SWAP & EXIT IF SO
0C88	90	SUB	B	GET DIFFERENCE OF EXPONENTS

0C89	3016	JR	NC,0CA1H	SKIP IF POSITIVE
0C8B	2F	CPL		DIFFERENCE =
0C8C	3C	INC	A	- DIFFERENCE
0C8D	F5	PUSH	AF	SAVE DIFFERENCE
0C8E	0E08	LD	C,08H	SWAP 8 BYTES
0C90	23	INC	HL	POINT TO ACCUM 2 EXPONENT
0C91	E5	PUSH	HL	SAVE POINTER
0C92	1A	LD	A,(DE)	GET BYTE FROM ACCUM
0C93	46	LD	B,(HL)	GET BYTE FROM ACCUM 2
0C94	77	LD	(HL),A	SAVE BYTE FROM ACCUM
0C95	78	LD	A,B	SAVE BYTE FROM ACCUM 2
0C96	12	LD	(DE),A	
0C97	1B	DEC	DE	BUMP POINTERS
0C98	2B	DEC	HL	
0C99	0D	DEC	C	ONE LESS BYTE TOGO
0C9A	20F6	JR	NZ,0C92H	LOOP UNTIL DONE
0C9C	E1	POP	HL	RESTORE POINTER
0C9D	46	LD	B,(HL)	GET POINTER OF ACCUM 2
0C9E	2B	DEC	HL	
0C9F	4E	LD	C,(HL)	GET MSB OF ACCUM 2
0CA0	F1	POP	AF	RESTORE EXPONENT DIFFERENCE
0CA1	FE39	CP	39H	TOO BIG A MAGNITUDE
0CA3	D0	RET	NC	DIFFERENCE? RETURN IF SO
0CA4	F5	PUSH	AF	SAVE DIFFERENCE
0CA5	CDDF09	CALL	09DFH	UNPACK ACCUM
0CA8	23	INC	HL	ZERO WORKSPACE
0CA9	3600	LD	(HL),00H	FOR NORMALIZATION
0CAB	47	LD	B,A	B = COMBINED SIGN
0CAC	F1	POP	AF	RESTORE DIFFERENCE
0CAD	212D41	LD	HL,412DH	POINT TO MSB OF ACCUM 2
0CB0	CD690D	CALL	0D69H	UNPACK ACCUM 2 JUSTIFY
0CB3	3A2641	LD	A,(4126H)	ZERO ACCUM
0CB6	321C41	LD	(411CH),A	WORKSPACE
0CB9	78	LD	A,B	CHECK
0CBA	B7	OR	A	COMBINED SIGN
0CBB	F2CF0C	JP	P,0CCFH	SKIP IF POSITIVE
0CBE	CD330D	CALL	0D33H	ACCUM = ACCUM + ACCUM 2
0CC1	D20E0D	JP	NC,0D0EH	FINISH UP IF NO OVERFLOW
0CC4	EB	EX	DE,HL	OTHERWISE, INCREASE
0CC5	34	INC	(HL)	EXPONENT
0CC6	CAB207	JP	Z,07B2H	CV ERROR IF OVERFLOW
0CC9	CD900D	CALL	0D90H	SHIFT ACCUM RIGHT 1 BIT
0CCC	C30E0D	JP	0D0EH	FINISH UP & RETURN
0CCF	CD450D	CALL	0D45H	ACCUM = ACCUM - ACCUM2
0CD2	212541	LD	HL,4125H	POINT TO SIGN OF ACCUM
0CD5	DC570D	CALL	C,0D57H	INVERT ACCUM IF NECESSARY
0CD8	AF	XOR	A	CLEAR FLAG
0CD9	47	LD	B,A	SAVE FLAG
0CDA	3A2341	LD	A,(4123H)	CHECK MSB OF ACCUM
0CDD	B7	OR	A	ZERO?
0CDE	201E	JR	NZ,0CFEH	SKIP OUT IF NOT
0CE0	211C41	LD	HL,411CH	SHIFT ACCUM LEFT ONE BYTE
0CE3	0E08	LD	C,08H	8 BYTES TO SHIFT
0CE5	56	LD	D,(HL)	PUT (HL) IN D
0CE6	77	LD	(HL),A	PUT A IN (HL)
0CE7	7A	LD	A,D	PUT D IN A
0CE8	23	INC	HL	MOVE TO NEXT BYTE

0CE9	0D	DEC	C	ONE LESS TO DO
0CEA	20F9	JR	NZ,0CE5H	LOOP UNTIL DONE
0CEC	78	LD	A,B	EXPONENT =
0CED	D608	SUB	08H	EXPONENT - 8
0CEF	FEC0	CP	0C0H	ALL EIGHT BYTES CHECKED?
0CF1	20E6	JR	NZ,0CD9H	LOOP IF FOOT
0CF3	C37807	JP	0778H	ACCUM = 0 RETURN
0CF6	05	DEC	B	EXPONENT = EXPONENT - 1
0CF7	211C41	LD	HL,411CH	POINT TO ACCUM
0CFA	CD970D	CALL	0D97H	SHIFT ACCUM LEFT 1 BIT
0CFD	B7	OR	A	TEST MSB
0CFE	F2F60C	JP	P,0CF6H	LOOP IF BIT 7<>0
0D01	78	LD	A,B	CHECK EXPONENT CHANGE
0D02	B7	OR	A	
0D03	2809	JR	Z,0D0EH	FINISH UP IF ZERO
0D05	212441	LD	HL,4124H	ADD CHANGE TO
0D08	86	ADD	A,(HL)	ORIGINAL EXPONENT
0D09	77	LD	(HL),A	SAVE IT
0D0A	D27807	JP	NC,0778H	ZERO IF NO OVERFLOW
0D0D	C8	RET	Z	DONE IF ZERO
0D0E	3A1C41	LD	A,(411CH)	CHECK SIGN OF
0D11	B7	OR	A	MANTISSA
0D12	FC200D	CALL	M,0D20H	CONVERT TRAILING ZEROES IF NEGATIVE
0D15	212541	LD	HL,4125H	GET SIGN OF ACCUM
0D18	7E	LD	A,(HL)	
0D19	E680	AND	80H	
0D1B	2B	DEC	HL	PUT IT IN
0D1C	2B	DEC	HL	THE
0D1D	AE	XOR	(HL)	MANTISSA
0D1E	77	LD	(HL),A	
0D1F	C9	RET		DONE
0D20	211D41	LD	HL,411DH	ACCUM =
0D23	0607	LD	B,07H	ACCUM + 1
0D25	34	INC	(HL)	(CONVERT TRAILING
0D26	C0	RET	NZ	ZEROES TO ONES)
0D27	23	INC	HL	
0D28	05	DEC	B	
0D29	20FA	JR	NZ,0D25H	
0D2B	34	INC	(HL)	
0D2C	CAB207	JP	Z,07B2H	OV ERROR IF OVERFLOW
0D2F	2B	DEC	HL	FIX MSB OF ACCUM
0D30	3680	LD	(HL),80H	
0D32	C9	RET		DONE
0D33	212741	LD	HL,4127H	ACCUM
0D36	111D41	LD	DE,411DH	ACCUM + ACCUM 2
0D39	0E07	LD	C,07H	SEVEN BYTES TO PROCESS
0D3B	AF	XOR	A	CLEAR STATUS
0D3C	1A	LD	A,(DE)	ADD BYTES
0D3D	8E	ADC	A,(HL)	INCLUDING OVERFLOW
0D3E	12	LD	(DE),A	AND SAVE
0D3F	13	INC	DE	POINT TO
0D40	23	INC	HL	NEXT BYTES
0D41	0D	DEC	C	ONE LESS BYTE TO DO
0D42	20F8	JR	NZ,0D3CH	LOOP UNTIL DONE
0D44	C9	RET		DONE
0D45	212741	LD	HL,4127H	ACCUM =
0D48	111D41	LD	DE,411DH	ACCUM - ACCUM 2

0D4B	0E07	LD	C, 07H	SEVEN BYTES TO PROCESS
0D4D	AF	XOR	A	CLEAR STATUS
0D4E	1A	LD	A, (DE)	SUBTRACT BYTES
0D4F	9E	SBC	A, (HL)	INCLUDING OVERFLOW
0D50	12	LD	(DE), A	AND SAVE
0D51	13	INC	DE	POINT TO
0D52	23	INC	HL	NEXT BYTES
0D53	0D	DEC	C	ONE LESS BYTE TO GO
0D54	20F8	JR	NZ, 0D4EH	LOOP UNTIL DONE
0D56	C9	RET		DONE
0D57	7E	LD	A, (HL)	INVERT SIGN BYTE
0D58	2F	CPL		
0D59	77	LD	(HL), A	
0D5A	211C41	LD	HL, 411CH	POINT TO ACCUM
0D5D	0608	LD	B, 08H	INVERT 8 BYTES
0D5F	AF	XOR	A	CLEAR STATUS
0D60	4F	LD	C, A	C = 0
0D61	79	LD	A, C	A = C = 0
0D62	9E	SBC	A, (HL)	(HL) = - (HL)
0D63	77	LD	(HL), A	
0D64	23	INC	HL	GO TO NEXT BYTE
0D65	05	DEC	B	ONE LESS TO DO
0D66	20F9	JR	NZ, 0D61H	LOOP UNTIL DONE
0D68	C9	RET		DONE
0D69	71	LD	(HL), C	SAVE MSB
0D6A	E5	PUSH	HL	SAVE POINTER
0D6B	D608	SUB	08H	ONE LESS BYTE TO SHIFT RIGHT
0D6D	380E	JR	C, 0D7DH	JUMP IF THERE WAS LESS LEFT
0D6F	E1	POP	HL	RESTORE POINTER
0D70	E5	PUSH	HL	SAVE POINTER
0D71	110008	LD	DE, 0800H	DE = 8 : E = 0
0D74	4E	LD	C, (HL)	MOVE (HL) TO C.
0D75	73	LD	(HL), E	MOTE E TO (HL)
0D76	59	LD	E, C	MOVE C TO E
0D77	2B	DEC	HL	NEXT BYTES
0D78	15	DEC	D	ONE LESS TO DO
0D79	20F9	JR	NZ, 0D74H	LOOP UNTIL DONE
0D7B	18EE	JR	0D6BH	SHIFT MORE IF NECESSARY
0D7D	C609	ADD	A, 09H	A = # BITS TO RIGHT SHIFT + 1
0D7F	57	LD	D, A	SAVE BIT COUNT
0D80	AF	XOR	A	CLEAR STATUS
0D81	E1	POP	HL	RESTORE POINTER
0D82	15	DEC	D	ONE LESS BIT TO SHIFT
0D83	C8	RET	Z	RETURN IF DONE
0D84	E5	PUSH	HL	SAVE POINTER
0D85	1E08	LD	E, 08H	SHIFT 8 BYTES
0D87	7E	LD	A, (HL)	GET BYTE
0D88	1F	RRA		SHIFT IT RIGHT
0D89	77	LD	(HL), A	SAVE IT
0D8A	2B	DEC	HL	NEXT BYTES
0D8B	1D	DEC	E	ONE LESS TO DO
0D8C	20F9	JR	NZ, 0D87H	LOOP UNTIL DONE
0D8E	18F0	JR	0D80H	LOOP UNTIL DONE
0D90	212341	LD	HL, 4123H	SHIFT ACCUM
0D93	1601	LD	D, 01H	RIGHT 1 BIT
0D95	18ED	JR	0D84H	GO TO IT
0D97	0E08	LD	C, 08H	SHIFT 8 BYTES

0D99	7E	LD	A, (HL)	POINTED TO BY HL,
0D9A	17	RLA		LEFT ONE BIT
0D9B	77	LD	(HL), A	
0D9C	23	INC	HL	
0D9D	0D	DEC	C	ONE LESS TO DO
0D9E	20F9	JR	NZ, 0D99H	ONE LOOP UNTIL DONE
0DA0	C9	RET		DONE

DOUBLE PRECISION MULTIPLICATION

0DA1	CD5509	CALL	0955H	ACCUM = 0?
0DA4	C8	RET	Z	RETURN IF SO
0DA5	CD0A09	CALL	090AH	ADD EXPONENTS
0DA8	CD390E	CALL	0E39H	SET UP WORKSPACE
0DAB	71	LD	(HL), C	ZERO TAIL BYTE
0DAC	13	INC	DE	POINT TO WORKSPACE LSB
0DAD	0607	LD	B, 07H	MULTIPLY 7 BYTES
0DAF	1A	LD	A, (DE)	GET BYTE
0DB0	13	INC	DE	POINT TO NEXT BYTE
0DB1	B7	OR	A	CHECK BYTE
0DB2	D5	PUSH	DE	SAVE POINTER
0DB3	2817	JR	Z, 0DCCH	JUMP IF BYTE ZERO
0DB5	0E08	LD	C, 08H	LOOP OVER ALL 8 BITS
0DB7	C5	PUSH	BC	SAVE BC
0DB8	1F	RRA		ROTATE BYTE TO CARRY
0DB9	47	LD	B, A	SAVE BYTE
0DBA	DC330D	CALL	C, 0D33H	IF BIT SET, ACCUM = ACCUM + ACCUM 2
0DBD	CD900D	CALL	0D90H	SHIFT ACCUM RIGHT
0DC0	78	LD	A, B	RESTORE BYTE
0DC1	C1	POP	BC	RESTORE BC
0DC2	0D	DEC	C	ONE LESS BIT TO GO
0DC3	20F2	JR	NZ, 0DB7H	LOOP UNTIL DONE
0DC5	D1	POP	DE	RESTORE POINTER
0DC6	05	DEC	B	ONE LESS BYTE TO DO
0DC7	20E6	JR	NZ, 0DAFH	LOOP UNTIL DONE
0DC9	C3D80C	JP	0CD8H	NORMALIZE & RETURN
0DCC	212341	LD	HL, 4123H	SHIFT ACCUM
0DCF	CD700D	CALL	0D70H	RIGHT ONE BYTE
0DD2	18F1	JR	0DC5H	DONE
0DD4	000000000002084			10.0D0
0DDC	11D40D	LD	DE, 0DD4H	DIVIDE D.P. ACCUM
0DDF	212741	LD	HL, 4127H	BY 10
0DE2	CDD309	CALL	09D3H	

DOUBLE PRECISION DIVISION

0DE5	3A2E41	LD	A, (412EH)	DOES ACCUM 2
0DE8	B7	OR	A	EQUAL ZERO?
0DE9	CA9A19	JP	Z, 199AH	DIVISION BY 0 ERROR IF SO
0DEC	CD0709	CALL	0907H	SUBTRACT EXPONENTS
0DEF	34	INC	(HL)	INCREASE RESULTING
0DF0	34	INC	(HL)	EXPONENT BY TWO
0DF1	CD390E	CALL	0E39H	SET UP WORKSPACE
0DF4	215141	LD	HL, 4151H	ZERO EXPONENTS
0DF7	71	LD	(HL), C	
0DF8	41	LD	B, C	
0DF9	114A41	LD	DE, 414AH	ACCUM 2 =
0DFC	212741	LD	HL, 4127H	ACCUM 2
0DFE	CD4B0D	CALL	0D4BH	- WORKSPACE

0E02 1A	LD	A, (DE)	WAS ACCUM 2 ≥
0E03 99	SBC	A, C	WORKSPACE?
0E04 3F	CCF		
0E05 380B	JR	C, 0E12H	SKIP IF IT WAS.
0E07 114A41	LD	DE, 414AH	WORKSPACE =
0E0A 212741	LD	HL, 4127H	WORKSPACE + ACCUM 2
0E0D CD390D	CALL	0D39H	
0E10 AFDA			MASK TO 0E14H
0E12 12	LD	(DE), A	SAVE NEW EXPONENT
0E13 04	INC	B	ONE MORE SUBTRACTION
0E14 3A2341	LD	A, (4123H)	GET MSB OF ACCUM
0E17 3C	INC	A	IS
0E18 3D	DEC	A	CARRY
0E19 1F	RRA		SET? (DONE?)
0E1A FA110D	JP	M, 0D11H	JUMP IF SO
0E1D 17	RLA		RESTORE CARRY
0E1E 211D41	LD	HL, 411DH	HL => ACCUM LSB
0E21 0E07	LD	C, 07H	SHIFT 7 BYTES LEFT
0E23 CD990D	CALL	0D99H	SHIFT THEM
0E26 214A41	LD	HL, 414AH	HL => WORKSPACE LSB
0E29 CD970D	CALL	0D97H	SHIFT LEFT ONE
0E2C 78	LD	A, B	GET # SUBTRACTIONS
0E2D B7	OR	A	TEST
0E2E 20C9	JR	NZ, 0DF9H	JUMP IF SOME DONE
0E30 212441	LD	HL, 4124H	DECREMENT EXPONENT
0E33 35	DEC	(HL)	OF ACCUM
0E34 20C3	JR	NZ, 0DF9H	LOOP IF NON-ZERO
0E36 C3B207	JP	07B2H	OV ERROR IF OVERFLOW
0E39 79	LD	A, C	SAVE MSB
0E3A 322D41	LD	(412DH), A	OF ACCUM 2
0E3D 2B	DEC	HL	POINT TO MSB OF ACCUM
0E3E 115041	LD	DE, 4150H	MOVE ACCUM TO
0E41 010007	LD	BC, 0700H	WORKSPACE, ZERO ACCUM
0E44 7E	LD	A, (HL)	GET BYTE
0E45 12	LD	(DE), A	SAVE BYTE TO NEW LOCATION
0E46 71	LD	(HL), C	ZERO OLD LOCATION
0E47 1B	DEC	DE	POINT TO
0E48 2B	DEC	HL	NEXT BYTES
0E49 05	DEC	B	ONE LESS TO DO
0E4A 20F8	JR	NZ, 0E44H	LOOP UNTIL DONE
0E4C C9	RET		DONE
0E4D CDFC09	CALL	09FCH	ACCUM 2 = ACCUM
0E50 EB	EX	DE, HL	HL => LSB OF ACCUM
0E51 2B	DEC	HL	BACK UP TO EXPONENT
0E52 7E	LD	A, (HL)	EXPONENT ZERO?
0E53 B7	OR	A	
0E54 C8	RET	Z	RETURN IF SO
0E55 C602	ADD	A, 02H	ACCUM = ACCUM*4
0E57 DAB207	JP	C, 07B2H	OV ERROR IF OVERFLOW
0E5A 77	LD	(HL), A	SAVE EXPONENT
0E5B E5	PUSH	HL	SAVE HL
0E5C CD770C	CALL	0C77H	ACCUM = ACCUM*4 + ACCUM 2
0E5F E1	POP	HL	
0E60 34	INC	(HL)	ACCUM = ACCUM*10
0E61 C0	RET	NZ	RETURN IF NO OVERFLOW
0E62 C3B207	JP	07B2H	OV ERROR IF OVERFLOW

DECODE AN ASCII NUMBER

0E65	CD7807	CALL	0778H	ZERO EXPONENT
0E68	CDEC0A	CALL	0AECH	DOUBLE PRECISION
0E6B	F6			MASK TO 0E6DH
0E6C	AF	XOR	A	CLEAR STATUS
0E6D	EB	EX	DE,HL	SAVE HL
0E6E	01FF00	LD	BC,00FFH	
0E71	60	LD	H,B	HL = 0
0E72	68	LD	L,B	
0E73	CC9A0A	CALL	Z,0A9AH	FLAG AS INTEGER + 0
0E76	EB	EX	DE,HL	RESTORE HL DE=0
0E77	7E	LD	A, (HL)	GET CHARACTER.
0E78	FE2D	CP	2DH	"-"?
0E7A	F5	PUSH	AF	SAVE RESULT (SIGN)
0E7B	CA830E	JP	Z,0E83H	SKIP IF SO
0E7E	FE2B	CP	2BH	"+"?
0E80	2801	JR	Z,0E83H	SKIP IF SO
0E82	2B	DEC	HL	BACK UP ONE CHARACTER.
0E83	D7	RST	10H	GET NEXT CHARACTER
0E84	DA290F	JP	C,0F29H	JUMP IF NUMBER
0E87	FE2E	CP	2EH	". "?
0E89	CAE40E	JP	Z,0EE4H	JUMP IF SO
0E8C	FE45	CP	45H	"E"?
0E8E	2814	JR	Z,0EA4H	JUMP IF SO
0E90	FE25	CP	25H	"%"?
0E92	CAEE0E	JP	Z,0EEEH	JUMP IF SO.
0E95	FE23	CP	23H	"#"?
0E97	CAF50E	JP	Z,0EF5H	JUMP IF SO.
0E9A	FE21	CP	21H	"!"?
0E9C	CAF60E	JP	Z,0EF6H	JUMP IF SO
0E9F	FE44	CP	44H	"D"?
0EA1	2024	JR	NZ,0EC7H	JUMP IF NOT
0EA3	B7	OR	A	NZ = DOUBLE PRECISION
0EA4	CDFB0E	CALL	0EFBH	MAKE FLOATING POINT
0EA7	E5	PUSH	HL	SAVE HL
0EA8	21BD0E	LD	HL,0EBDH	SET RETURN ADDRESS
0EAB	E3	EX	(SP),HL	SAVE RETURN & RESTORE HL
0EAC	D7	RST	10H	GET NEXT CHARACTER
0EAD	15	DEC	D	D = -1 (NEGATIVE)
0EAE	FECE	CP	0CEH	"-"?
0EB0	C8	RET	Z	RETURN IF SO
0EB1	FE2D	CP	2DH	"-"?
0EB3	C8	RET	Z	RETURN IF SO
0EB4	14	INC	D	D = 0 = POSITIVE
0EB5	FECD	CP	0CDH	"+"?
0EB7	C8	RET	Z	RETURN IF SO
0EB8	FE2B	CP	2BH	"+"?
0EBA	C8	RET	Z	RETURN IF SO
0EBB	2B	DEC	HL	BACK UP ONE CHARACTER
0EBC	F1	POP	AF	CLEAR RETURN FROM STACK
0EBD	D7	RST	10H	GET NEXT CHARACTER
0EBE	DA940F	JP	C,0F94H	JUMP IF NUMERIC
0EC1	14	INC	D	0 = NEGATIVE 1 = POSITIVE
0EC2	2003	JR	NZ,0EC7H	SKIP IF POSITIVE
0EC4	AF	XOR	A	MAKE EXPONENT
0EC5	93	SUB	E	NEGATIVE (E = -E)
0EC6	5F	LD	E,A	SAVE EXPONENT

0EC7	E5	PUSH	HL	SAVE HL
0EC8	7B	LD	A,E	GET EXPONENT IN A
0EC9	90	SUB	B	LESS # DIGITS RIGHT OF DECIMAL
0ECA	F40A0F	CALL	P,0F0AH	ACCUM = ACCUM * 10
0ECD	FC180F	CALL	M,0F18H	ACCUM = ACCUM / 10
0ED0	20F8	JR	NZ,0ECAH	CONTINUE UNTIL A = 0
0ED2	E1	POP	HL	RESTORE HL
0ED3	F1	POP	AF	GET SIGN OF COEFFICIENT
0ED4	E5	PUSH	HL	SAVE HL ONCE MORE
0ED5	CC7B09	CALL	Z,097BH	INVERT ACCUM IF NECESSARY
0ED8	E1	POP	HL	RESTORE HL AGAIN
0ED9	E7	RST	20H	CHECK TYPE OF ACCUM
0EDA	E8	RET	PE	RETURN IF S.P.
0EDB	E5	PUSH	HL	SAVE HL (AGAIN!!)
0EDC	219008	LD	HL,0890H	RETURN THROUGH
0EDF	E5	PUSH	HL	ROUTINE 0890H
0EE0	CDA30A	CALL	0AA3H	HANDLE SPECIAL CASE
0EE3	C9	RET		DONE
0EE4	E7	RST	20H	GET TYPE OF ACCUM
0EE5	0C	INC	C	INCREMENT # DECIMALS
0EE6	20DF	JR	NZ,0EC7H	FINISH UP IF TWO OR MORE
0EE8	DCF80E	CALL	C,0EFBH	MAKE S.P. IF NOT D.P.
0EEB	C3830E	JP	0E83H	CONTINUE SCAN
0EEE	E7	RST	20H	GET TYPE OF ACCUM
0EEF	F29719	JP	P,1997H	SYNTAX ERROR IF NOT INTEGER
0EF2	23	INC	HL	GO TO NEXT CHARACTER
0EF3	18D2	JR	0EC7H	FINISH UP
0EF5	B7	OR	A	NZ = DOUBLE PRECISION
0EF6	CDF80E	CALL	0EFBH	MAKE ACCUM FLOATING POINT
0EF9	18F7	JR	0EF2H	CONTINUE
0EFB	E5	PUSH	HL	SAVE REGISTERS
0EFC	D5	PUSH	DE	
0EFD	C5	PUSH	BC	
0EFE	F5	PUSH	AF	SAVE STATUS
0EFF	CCB10A	CALL	Z,0AB1H	MAKE S.P. IF ZERO
0F02	F1	POP	AF	RESTORE STATUS
0F03	C4DB0A	CALL	NZ,0ADBH	MAKE D.P. IF NON-ZERO
0F06	C1	POP	BC	RESTORE REGISTER
0F07	D1	POP	DE	
0F08	E1	POP	HL	
0F09	C9	RET		DONE. CONTINUE SCAN
0F0A	C8	RET	Z	RETURN IF ZERO
0F0B	F5	PUSH	AF	SAVE STATUS
0F0C	E7	RST	20H	CHECK TYPE OF ACCUM
0F0D	F5	PUSH	AF	SAVE STATUS
0F0E	E43E09	CALL	PO,093EH	ACCUM = ACCUM*10 (SP)
0F11	F1	POP	AF	RESTORE STATUS
0F12	EC4D0E	CALL	PE,0E4DH	ACCUM = ACCUM*10 (DP)
0F15	F1	POP	AF	RESTORE STATUS
0F16	3D	DEC	A	DECREMENT A
0F17	C9	RET		CONTINUE
0F18	D5	PUSH	DE	SAVE REGISTERS
0F19	E5	PUSH	HL	
0F1A	F5	PUSH	AF	
0F1B	E7	RST	20H	GET TYPE OF ACCUM
0F1C	F5	PUSH	AF	SAVE STATUS
0F1D	E49708	CALL	PO,0897H	ACCUM = ACCUM/10 (S.P.)

0F20	F1	POP	AF	RESTORE STATUS
0F21	ECDC0D	CALL	PE,0DDCH	ACCUM = ACCUM/10 (D.P.)
0F24	F1	POP	AF	RESTORE REGISTERS
0F25	E1	POP	HL	
0F26	D1	POP	DE	
0F27	3C	INC	A	INCREMENT A
0F28	C9	RET		CONTINUE
0F29	D5	PUSH	DE	SAVE DE
0F2A	78	LD	A,B	B = B + 1 IF
0F2B	89	ADC	A,C	TO RIGHT OF
0F2C	47	LD	B,A	DECIMAL POINT
0F2D	C5	PUSH	BC	SAVE BC
0F2E	E5	PUSH	HL	SAVE HL
0F2F	7E	LD	A, (HL)	GET CHARACTER.
0F30	D630	SUB	30H	GET VALUE OF CHARACTER
0F32	F5	PUSH	AF	SAVE IT
0F33	E7	RST	20H	CHECK TYPE OF ACCUM
0F34	F25D0F	JP	P,0F5DH	JUMP IF NOT INTEGER
0F37	2A2141	LD	HL, (4121H)	GET VALUE
0F3A	11CD0C	LD	DE,0CCDH	IS IT
0F3D	DF	RST	18H	≥ 3277?
0F3E	3019	JR	NC,0F59H	MAKE S.P. IF SO
0F40	54	LD	D,H	DE = HL
0F41	5D	LD	E,L	
0F42	29	ADD	HL,HL	HL = HL*2
0F43	29	ADD	HL,HL	HL = HL*4
0F44	19	ADD	HL,DE	HL = HL*5
0F45	29	ADD	HL,HL	HL = HL*10
0F46	F1	POP	AF	RESTORE CHARACTER VALUE
0F47	4F	LD	C,A	SAVE VALUE IN BC
0F48	09	ADD	HL,BC	HL = OLD*10 + NEW
0F49	7C	LD	A,H	IS HL OVERFLOWED?
0F4A	B7	OR	A	
0F4B	FA570F	JP	M,0F57H	MAKE S.P. IF SO
0F4E	222141	LD	(4121H),HL	SAVE NEW VALUE
0F51	E1	POP	HL	RESTORE REGISTERS
0F52	C1	POP	BC	
0F53	D1	POP	DE	
0F54	C3830E	JP	0E83H	CONTINUE SCAN
0F57	79	LD	A,C	GET VALUE OF CHARACTER
0F58	F5	PUSH	AF	SAVE IT
0F59	CDCC0A	CALL	0ACCH	MAKE ACCUM SINGLE PRECISION
0F5C	37	SCF		NO JUMP
0F5D	3018	JR	NC,0F77H	JUMP IF DOUBLE PRECISION
0F5F	017494	LD	BC,9474H	IS ACCUM
0F62	110024	LD	DE,2400H	≥ 1,000,000?
0F65	CD0C0A	CALL	0A0CH	
0F68	F2740F	JP	P,0F74H	MAKE D.P. IF SO
0F6B	CD3E09	CALL	093EH	ACCUM = ACCUM*10
0F6E	F1	POP	AF	RESTORE VALUE OF CHARACTER
0F6F	CD890F	CALL	0F89H	
0F72	18DD	JR	0F51H	
0F74	CDE30A	CALL	0AE3H	SET UP D.P.
0F77	CD4D0E	CALL	0E4DH	ACCUM = ACCUM*10
0F7A	CDFC09	CALL	09FCH	ACCUM 2 = ACCUM
0F7D	F1	POP	AF	RESTORE VALUE OF CHARACTER
0F7E	CD6409	CALL	0964H	ACCUM = VALUE OF CHARACTER

0F81	CDE30A	CALL	0AE3H	SET UP AS D.P.
0F84	CD770C	CALL	0C77H	ACCUM = ACCUM + ACCUM 2
0F87	18C8	JR	0F51H	CONTINUE SCAN
0F89	CDA409	CALL	09A4H	PUT ACCUM ON STACK
0F8C	CD6409	CALL	0964H	BC ACCUM = VALUE OF CHARACTER
0F8F	C1	POP	BC	PUT OLD ACCUM
0F90	D1	POP	DE	IN BCDE
0F91	C31607	JP	0716H	ACCUM = ACCUM + VALUE
0F94	7B	LD	A,E	GET EXPONENT
0F95	FE0A	CP	0AH	IS IT 10 OR MORE?
0F97	3009	JR	NC,0FA2H	FORCE TO 50 IF SO
0F99	07	RLCA		E = E*10 + VALUE
0F9A	07	RLCA		
0F9B	83	ADD	A,E	
0F9C	07	RLCA		
0F9D	86	ADD	A, (HL)	
0F9E	D630	SUB	30H	
0FA0	5F	LD	E,A	
0FA1	FA			MASK TO 0FA4H
0FA2	1E32	LD	E,32H	FORCE EXPONENT TO 50
0FA4	C3BD0E	JP	0EBDH	CONTINUE EXPONENT SCAN
0FA7	E5	PUSH	HL	SAVE HL
0FA8	212419	LD	HL,1924H	HL => "IN " *
0FAB	CDA728	CALL	28A7H	PRINT "IN "; :
0FAE	E1	POP	HL	RESTORE HL
0FAF	CD9A0A	CALL	0A9AH	SAVE HL IN ACCUM
0FB2	AF	XOR	A	SET TO NO-EDIT
0FB3	CD3410	CALL	1034H	SET UP TO PRINT
0FB6	B6	OR	(HL)	SET NZ
0FB7	CDD90F	CALL	0FD9H	FORMAT FOR PRINTOUT
0FBA	C3A628	JP	28A6H	OUTPUT & RETURN
0FBD	AF	XOR	A	CLEAR TO NO-EDIT
0FBE	CD3410	CALL	1034H	SAVE EDIT FLAGS
0FC1	E608	AND	08H	INCLUDE THE SIGN?
0FC3	2802	JR	Z,0FC7H	SKIP IF NOT
0FC5	362B	LD	(HL),2BH	DISPLAY A "+".
0FC7	EB	EX	DE,HL	SAVE HL
0FC8	CD9409	CALL	0994H	GET SIGN OF ACCUM
0FCB	EB	EX	DE,HL	RESTORE HL
0FCC	F2D90F	JP	P,0FD9H	SKIP IF POSITIVE
0FCF	362D	LD	(HL),2DH	DISPLAY A "-".
0FD1	C5	PUSH	BC	SAVE BC
0FD2	E5	PUSH	HL	SAVE HL
0FD3	CD7B09	CALL	097BH	MAKE ACCUM POSITIVE
0FD6	E1	POP	HL	RESTORE HL
0FD7	C1	POP	BC	RESTORE BC
0FD8	B4	OR	H	SET NZ
0FD9	23	INC	HL	NEXT POSITION
0FDA	3630	LD	(HL),30H	SAVE "0" IN IT
0FDC	3AD840	LD	A,(40D8H)	GET EDIT FLAGS
0FDF	57	LD	D,A	PUT THEM IN D
0FE0	17	RLA		EDIT?
0FE1	3AAF40	LD	A,(40AFH)	GET TYPE FLAG
0FE4	DA9A10	JP	C,109AH	JUMP IF EDIT
0FE7	CA9210	JP	Z,1092H	JUMP IF EDIT FLAG ZERO
0FEA	FE04	CP	04H	SINGLE PRECISION?
0FEC	D23D10	JP	NC,103DH	JUMP IF FLOATING POINT

0FEF	010000	LD	BC,0000H	NO COMMAS OR DECIMALS
0FF2	CD2F13	CALL	132FH	CONVERT INTEGER TO ASCII
0FF5	213041	LD	HL,4130H	POINT TO BUFFER
0FF8	46	LD	B,(HL)	GET SIGN, IF ANY
0FF9	0E20	LD	C,20H	C = SPACE
0FFB	3AD840	LD	A,(40D8H)	GET EDIT FLAGS
0FFE	5F	LD	E,A	AND PUT IN E
0FFF	E620	AND	20H	LEADING ASTERISKS?
1001	2807	JR	Z,100AH	SKIP IF NOT
1003	78	LD	A,B	GET SIGN
1004	B9	CP	C	IS IT A SPACE?
1005	0E2A	LD	C,2AH	GET ASTERISK
1007	2001	JR	NZ,100AH	SKIP IF NO SPACE
1009	41	LD	B,C	SIGN = ASTERISK
100A	71	LD	(HL),C	DISPLAY SIGN
100B	D7	RST	10H	GET NEXT CHARACTER.
100C	2814	JR	Z,1022H	JUMP IF END OF LINE.
100E	FE45	CP	45H	"E"?
1010	2810	JR	Z,1022H	SKIP IF SO.
1012	FE44	CP	44H	"D"?
1014	280C	JR	Z,1022H	SKIP IF SO
1016	FE30	CP	30H	"0"?
1018	28F0	JR	Z,100AH	NEXT CHARACTER IF SO.
101A	FE2C	CP	2CH	","?
101C	28EC	JR	Z,100AH	NEXT CHARACTER IF SO.
101E	FE2E	CP	2EH	"."?
1020	2003	JR	NZ,1025H	NEXT CHARACTER. IF SO.
1022	2B	DEC	HL	BACK UP ONE CHARACTER.
1023	3630	LD	(HL),30H	CHARACTER = "0".
1025	7B	LD	A,E	PRINT
1026	E610	AND	10H	LEADING "\$"?
1028	2803	JR	Z,102DH	SKIP IF NOT
102A	2B	DEC	HL	BACK UP ONE CHARACTER
102B	3624	LD	(HL),24H	CHARACTER.
102D	7B	LD	A,E	PRINT LEADING
102E	E604	AND	04H	SIGN?
1030	C0	RET	NZ	RETURN IF NOT
1031	2B	DEC	HL	BACK UP ONE CHARACTER
1032	70	LD	(HL),B	CHARACTER = SIGN.
1033	C9	RET		DONE.
1034	32D840	LD	(40D8H),A	SAVE EDIT FLAGS.
1037	213041	LD	HL,4130H	HL => FORMATTING BUFFER
103A	3620	LD	(HL),20H	PUT A SPACE IN IT.
103C	C9	RET		DONE
103D	FE05	CP	05H	SET CARRY IF S.P.
103F	E5	PUSH	HL	SAVE HL
1040	DE00	SBC	A,00H	D = FIELD WIDTH
1042	17	RLA		= 7 FOR S.P.
1043	57	LD	D,A	= 17 FOR D.P.
1044	14	INC	D	
1045	CD0112	CALL	1201H	SCALE ACCUM
1048	010003	LD	BC,0300H	NO COMMAS. DECIMAL IN COL 3
104B	82	ADD	A,D	SCALED DOWN MORE THAN
104C	FA5710	JP	M,1057H	D PLACES? JUMP IF SO
104F	14	INC	D	D = 8 (SP) OR 18 (DP)
1050	BA	CP	D	SCALED AT ALL?
1051	3004	JR	NC,1057H	JUMP IF SCALED UP

1053	3C	INC	A	B = # DIGITS IN VALUE
1054	47	LD	B,A	
1055	3E02	LD	A,02H	FORCE EXPONENT TO ZERO
1057	D602	SUB	02H	COMPUTE EXPONENT
1059	E1	POP	HL	RESTORE POSITION IN BUFFER
105A	F5	PUSH	AF	SAVE EXPONENT
105B	CD9112	CALL	1291H	APPLY DECIMAL
105E	3630	LD	(HL),30H	STORE "0".
1060	CCC909	CALL	Z,09C9H	MOVE TO NEXT CHARACTER IF NO SCALING
1063	CDA412	CALL	12A4H	CONVERT ACCUM TO ASCII.
1066	2B	DEC	HL	BACK UP ONE CHARACTER.
1067	7E	LD	A,(HL)	IS IT A "0"?
1068	FE30	CP	30H	
106A	28FA	JR	Z,1066H	LOOP BACK IF SO.
106C	FE2E	CP	2EH	DECIMAL POINT?
106E	C4C909	CALL	NZ,09C9H	GO FORWARD IF NOT
1071	F1	POP	AF	GET EXPONENT
1072	281F	JR	Z,1093H	JUMP IF NO SCALING
1074	F5	PUSH	AF	SAVE EXPONENT
1075	E7	RST	20H	CHECK TYPE OF ACCUM
1076	3E22	LD	A,22H	A ="D" FOR D.P.
1078	8F	ADC	A,A	="E" FOR S.P.
1079	77	LD	(HL),A	SAVE EXPONENT CHARACTER.
107A	23	INC	HL	NEXT CHARACTER.
107B	F1	POP	AF	RESTORE EXPONENT
107C	362B	LD	(HL),2BH	STORE "+"
107E	F28510	JP	P,1085H	SKIP IF POSITIVE
1081	362D	LD	(HL),2DH	STORE "-".
1083	2F	CPL		INVERT EXPONENT
1084	3C	INC	A	
1085	062F	LD	B,2FH	B = "0" - 1
1087	04	INC	B	B = B + 1
1088	D60A	SUB	0AH	A = A - 10
108A	30FB	JR	NC,1087H	LOOP UNTIL A<0
108C	C63A	ADD	A,3AH	A = A + "0" + 10
108E	23	INC	HL	GO TO NEXT CHARACTER.
108F	70	LD	(HL),B	STORE FIRST DIGIT OF EXPONENT
1090	23	INC	HL	GO TO NEXT CHARACTER.
1091	77	LD	(HL),A	STORE SECOND DIGIT
1092	23	INC	HL	GO TO NEXT CHARACTER.
1093	3600	LD	(HL),00H	MARK END OF STRING.
1095	EB	EX	DE,HL	DE => END OF BUFFER
1096	213041	LD	HL,4130H	HL => START OF BUFFER
1099	C9	RET		RETURN TO CALLER
109A	23	INC	HL	NEXT POSITION
109B	C5	PUSH	BC	SAVE BC
109C	FE04	CP	04H	CHECK ACCUM TYPE
109E	7A	LD	A,D	GET EDIT FLAGS.
109F	D20911	JP	NC,1109H	JUMP IF FLOATING POINT.
10A2	1F	RRA		EXPONENTIAL NOTATION?
10A3	DAA311	JP	C,11A3H	JUMP IF SO
10A6	010306	LD	BC,0603H	SET COMMAS. DECIMAL IN COL 6.
10A9	CD8912	CALL	1289H	CHECK IF COMMAS SELECTED.
10AC	D1	POP	DE	GET OLD BC
10AD	7A	LD	A,D	GET # DIGITS LEFT OF DECIMAL
10AE	D605	SUB	05H	LESS 5.
10B0	F46912	CALL	P,1269H	ZERO LEADING ZEROES

10B3	CD2F13	CALL	132FH	CONVERT TO 5 INTEGER DIGITS
10B6	7B	LD	A,E	GET # DIGITS RIGHT OF DECIMAL
10B7	B7	OR	A	
10B8	CC2F09	CALL	Z,092FH	BACK UP IF ZERO
10BB	3D	DEC	A	ONE LESS ZERO (ALLOW FOR ".")
10BC	F46912	CALL	P,1269H	ADD TRAILING ZEROS.
10BF	E5	PUSH	HL	SAVE POSITION IN BUFFER.
10C0	CD50F	CALL	0FF5H	APPLY \$,* AS NECESSARY
10C3	E1	POP	HL	RESTORE POSITION IN BUFFER
10C4	2802	JR	Z,10C8H	SKIP IF LEADING SIGN
10C6	70	LD	(HL),B	APPLY TRAILING SIGN
10C7	23	INC	HL	NEXT CHARACTER.
10C8	3600	LD	(HL),00H	MARK END OF STRING.
10CA	212F41	LD	HL,412FH	GO TO START OF BUFFER.
10CD	23	INC	HL	NEXT CHARACTER.
10CE	3AF340	LD	A,(40F3H)	IS # OF DIGITS TO LEFT
10D1	95	SUB	L	OF DECIMAL CORRECT
10D2	92	SUB	D	
10D3	C8	RET	Z	RETURN IF SO
10D4	7E	LD	A,(HL)	GET CHARACTER.
10D5	FE20	CP	20H	SPACE?
10D7	28F4	JR	Z,10CDH	ONE LESS TO LEFT IF SO.
10D9	FE2A	CP	2AH	ASTERISK.
10DB	28F0	JR	Z,10CDH	ONE LESS TO LEFT IF SO
10DD	2B	DEC	HL	BACK UP ONE CHARACTER.
10DE	E5	PUSH	HL	SAVE POSITION IN BUFFER.
10DF	F5	PUSH	AF	SAVE CHARACTER.
10E0	01DF10	LD	BC,10DFH	SET RETURN TO LOOP
10E3	C5	PUSH	BC	
10E4	D7	RST	10H	GET NEXT CHARACTER.
10E5	FE2D	CP	2DH	"-"?
10E7	C8	RET	Z	CONTINUE SCAN IF SO
10E8	FE2B	CP	2BH	"+"?
10EA	C8	RET	Z	CONTINUE SCAN IF SO.
10EB	FE24	CP	24H	"\$"?
10ED	C8	RET	Z	CONTINUE SCAN IF SO.
10EE	C1	POP	BC	CLEAR RETURN
10EF	FE30	CP	30H	SPACE?
10F1	200F	JR	NZ,1102H	OVERFLOW IF NOT.
10F3	23	INC	HL	SKIP CHARACTER.
10F4	D7	RST	10H	GET NEXT CHARACTER.
10F5	300B	JR	NC,1102H	OVERFLOW IF NOT NUMERIC
10F7	2B	DEC	HL	BACKUP
10F8	01			SKIP TO 10FBH
10F9	2B	DEC	HL	MOVE CHARACTER. BACK
10FA	77	LD	(HL),A	
10FB	F1	POP	AF	GET CHARACTER.
10FC	28FB	JR	Z,10F9H	LOOP TO END OF FIELD
10FE	C1	POP	BC	CLEAR STACK.
10FF	C3CE10	JP	10CEH	TRY AGAIN.
1102	F1	POP	AF	CLEAR STACK.
1103	28FD	JR	Z,1102H	
1105	E1	POP	HL	RESTORE POSITION IN BUFFER
1106	3625	LD	(HL),25H	STORE "%" (OVERFLOW)
1108	C9	RET		DONE HL => CONVERTED NUMBER
1109	E5	PUSH	HL	SAVE POSITION IN BUFFER
110A	1F	RRA		EXPONENTIAL NOTATION?

110B	DAAA11	JP	C,11AAH	JUMP IF SO.
110E	2814	JR	Z,1124H	JUMP IF S.P.
1110	118413	LD	DE,1384H	IS ACCUM
1113	CD490A	CALL	0A49H	LESS THAN 10**16?
1116	1610	LD	D,10H	16 DIGIT FIELD
1118	FA3211	JP	M,1132H	JUMP IF SO.
111B	E1	POP	HL	CLEAR
111C	C1	POP	BC	STACK
111D	CDBD0F	CALL	0FBDH	RE-ENTER EDIT ROUTINE
1120	2B	DEC	HL	STORE "%"
1121	3625	LD	(HL),25H	BEFORE NUMBER (OVERFLOW)
1123	C9	RET		RETURN TO CALLER.
1124	010EB6	LD	BC,0B60EH	IS
1127	11CA1B	LD	DE,1BCAH	ACCUM
112A	CD0C0A	CALL	0A0CH	≥ 10**16?
112D	F21B11	JP	P,111BH	SKIP IF SO.
1130	1606	LD	D,06H	SIX DIGIT NUMBER
1132	CD5509	CALL	0955H	GET SIGN OF ACCUM
1135	C40112	CALL	NZ,1201H	SCALE IF NON-ZERO.
1138	E1	POP	HL	RESTORE POSITION IN BUFFER
1139	C1	POP	BC	RESTORE BC
113A	FA5711	JP	M,1157H	JUMP IF SCALED UP
113D	C5	PUSH	BC	SAVE BC
113E	5F	LD	E,A	SAVE SCALE FACTOR.
113F	78	LD	A,B	COMPUTE #
1140	92	SUB	D	OF LEADING ZEROES.
1141	93	SUB	E	
1142	F46912	CALL	P,1269H	STORE LEADING ZEROES.
1145	CD7D12	CALL	127DH	SET UP COMMAS & DECIMAL
1148	CDA412	CALL	12A4H	CONVERT ACCUM TO ASCII
114B	B3	OR	E	WAS NUMBER SCALED?
114C	C47712	CALL	NZ,1277H	STORE TRAILING ZEROS IF SO.
114F	B3	OR	E	WAS NUMBER SCALED?
1150	C49112	CALL	NZ,1291H	APPLY DECIMAL IF NECESSARY
1153	D1	POP	DE	RESTORE OLD BC
1154	C3B610	JP	10B6H	FINISH UP
1157	5F	LD	E,A	SAVE SCALE FACTOR.
1158	79	LD	A,C	GET # DIGITS
1159	B7	OR	A	FOLLOWING DECIMAL
115A	C4160F	CALL	NZ,0F16H	DECREMENT IF NOT ZERO.
115D	83	ADD	A,E	ADD SCALE FACTOR
115E	FA6211	JP	M,1162H	SKIP IF RESULT NEGATIVE.
1161	AF	XOR	A	NO SCALING DOWN.
1162	C5	PUSH	BC	SAVE BC.
1163	F5	PUSH	AF	SAVE NEW FACTOR
1164	FC180F	CALL	M,0F18H	IF NEGATIVE, ACCUM ACCUM/10
1167	FA6411	JP	M,1164H	LOOP UNTIL POSITIVE
116A	C1	POP	BC	GET FACTOR
116B	7B	LD	A,E	A = OLD - NEW SCALE FACTOR
116C	90	SUB	B	
116D	C1	POP	BC	RESTORE BC
116E	5F	LD	E,A	IS NUMBER ALL RIGHT
116F	82	ADD	A,D	IF DECIMAL?
1170	78	LD	A,B	GET # DIGITS ON LEFT
1171	FA7F11	JP	M,117FH	SKIP IF ALL RIGHT.
1174	92	SUB	D	COMPUTE # LEADING.
1175	93	SUB	E	ZEROES LEFT OF DECIMAL

1176	F46912	CALL	P,1269H	STORE LEADING ZEROES.
1179	C5	PUSH	BC	SAVE BC
117A	CD7D12	CALL	127DH	SET UP COMMAS & DECIMAL
117D	1811	JR	1190H	CONTINUE.
117F	CD6912	CALL	1269H	STORE LEADING ZEROES.
1182	79	LD	A,C	GET # DIGITS TO RIGHT
1183	CD9412	CALL	1294H	STORE DECIMAL POINT
1186	4F	LD	C,A	SAVE # DIGITS ON RIGHT
1187	AF	XOR	A	COMPUTE # OF
1188	92	SUB	D	LEADING ZEROES
1189	93	SUB	E	RIGHT OF DECIMAL
118A	CD6912	CALL	1269H	STORE ZEROES.
118D	C5	PUSH	BC	SAVE BC.
118E	47	LD	B,A	TURN OFF DECIMAL
118F	4F	LD	C,A	AND COMMA FLAGS
1190	CDA412	CALL	12A4H	CONVERT ACCUM TO ASCII
1193	C1	POP	BC	RESTORE BC
1194	B1	OR	C	GET # DIGITS ON RIGHT
1195	2003	JR	NZ,119AH	SKIP IF NON-ZERO
1197	2AF340	LD	HL,(40F3H)	POSITION TO DECIMAL POINT.
119A	83	ADD	A,E	COMPUTE # OF
119B	3D	DEC	A	TRAILING ZEROES.
119C	F46912	CALL	P,1269H	STORE TRAILING ZEROES.
119F	50	LD	D,B	SAVE # DIGITS ON LEFT
11A0	C3BF10	JP	10BFH	FINISH UP.
11A3	E5	PUSH	HL	SAVE POSITION IN BUFFER
11A4	D5	PUSH	DE	SAVE DE
11A5	CDCC0A	CALL	0ACCH	CONVERT ACCUM TO S.P.
11A8	D1	POP	DE	RESTORE DE.
11A9	AF	XOR	A	FORCE S.P. JUMP
11AA	CAB011	JP	Z,11B0H	JUMP IF S.P.
11AD	1E10	LD	E,10H	16 DIGIT FIELD
11AF	01			MASK TO 11B2H
11B0	1E06	LD	E,06H	6 DIGIT FIELD
11B2	CD5509	CALL	0955H	GET SIGN OF ACCUM.
11B5	37	SCF		JUMP AT 11F3H UNLESS SCALED.
11B6	C40112	CALL	NZ,1201H	SCALE ACCUM IF NECESSARY
11B9	E1	POP	HL	RESTORE POSITION
11BA	C1	POP	BC	RESTORE BC.
11BB	F5	PUSH	AF	SAVE JUMP FLAG.
11BC	79	LD	A,C	GET # DIGITS
11BD	B7	OR	A	TO RIGHT OF DECIMAL
11BE	F5	PUSH	AF	SAVE A.
11BF	C4160F	CALL	NZ,0F16H	DECREMENT IF NON-ZERO
11C2	80	ADD	A,B	COMPUTE TOTAL
11C3	4F	LD	C,A	# OF DIGITS.
11C4	7A	LD	A,D	DOES SIGN FOLLOW
11C5	E604	AND	04H	NUMBER?
11C7	FE01	CP	01H	D= 0 YES
11C9	9F	SBC	A,A	D=-1 (FF) NO
11CA	57	LD	D,A	
11CB	81	ADD	A,C	ADJUST TOTAL # OF DIGITS
11CC	4F	LD	C,A	TO ACCOMMODATE TRAILING SIGN.
11CD	93	SUB	E	COMPUTE SCALE FACTOR
11CE	F5	PUSH	AF	SAVE SCALE FACTOR
11CF	C5	PUSH	BC	SAVE SIZES.
11D0	FC180F	CALL	M,0F18H	SCALE DOWN

11D3	FAD011	JP	M,11D0H	IF NECESSARY
11D6	C1	POP	BC	RESTORE SIZES
11D7	F1	POP	AF	RESTORE SCALE FACTOR
11D8	C5	PUSH	BC	SAVE BOTH AGAIN.
11D9	F5	PUSH	AF	
11DA	FADE11	JP	M,11DEH	SKIP IF SCALED DOWN
11DD	AF	XOR	A	DO NOT SCALE DOWN
11DE	2F	CPL		SCALE FACTOR =
11DF	3C	INC	A	- SCALE FACTOR
11E0	80	ADD	A,B	COMPUTE
11E1	3C	INC	A	DECIMAL
11E2	82	ADD	A,D	POINT
11E3	47	LD	B,A	POSITION
11E4	0E00	LD	C,00H	NO COMMAS
11E6	CDA412	CALL	12A4H	CONVERT ACCUM TO ASCII
11E9	F1	POP	AF	GET SCALE FACTOR
11EA	F47112	CALL	P,1271H	APPLY TRAILING ZEROES.
11ED	C1	POP	BC	RESTORE BC
11EE	F1	POP	AF	GET # OF DIGITS TO RIGHT
11EF	CC2F09	CALL	Z,092FH	BACKUP IF ZERO
11F2	F1	POP	AF	RESTORE JUMP FLAG.
11F3	3803	JR	C,11F8H	JUMP IF NO SCALING WAS DONE.
11F5	83	ADD	A,E	COMPUTE.
11F6	90	SUB	B	EXPONENT
11F7	92	SUB	D	
11F8	C5	PUSH	BC	SAVE BC
11F9	CD7410	CALL	1074H	APPLY EXPONENT
11FC	EB	EX	DE,HL	RESTORE POSITION
11FD	D1	POP	DE	CLEAR STACK.
11FE	C3BF10	JP	10BFH	FINISH UP.
1201	D5	PUSH	DE	SAVE DE
1202	AF	XOR	A	# DIVISIONS = 0
1203	F5	PUSH	AF	
1204	E7	RST	20H	CHECK TYPE OF ACCUM
1205	E22212	JP	PO,1222H	JUMP IF S.P.
1208	3A2441	LD	A,(4124H)	IS EXPONENT ≥ 91H?
120B	FE91	CP	91H	
120D	D22212	JP	NC,1222H	JUMP IF SO
1210	116413	LD	DE,1364H	ACCUM =
1213	212741	LD	HL,4127H	ACCUM
1216	CDD309	CALL	09D3H	
1219	CDA10D	CALL	0DA1H	10 MILLION
121C	F1	POP	AF	# DIVISIONS =
121D	D60A	SUB	0AH	# DIVISIONS
121F	F5	PUSH	AF	- 10
1220	18E6	JR	1208H	LOOP
1222	CD4F12	CALL	124FH	BRING ACCUM BELOW 999,999
1225	E7	RST	20H	CHECK TYPE OF ACCUM
1226	300B	JR	NC,1233H	JUMP IF D.P.
1228	014391	LD	BC,9143H	IS
122B	11F94F	LD	DE,4FF9H	ACCUM
122E	CD0C0A	CALL	0A0CH	≥ 99,999?
1231	1806	JR	1239H	
1233	116C13	LD	DE,136CH	IS ACCUM
1236	CD490A	CALL	0A49H	≥ + 1.0D15?
1239	F24B12	JP	P,124BH	JUMP IF SO

123C	F1	POP	AF	ACCUM =
123D	CD0B0F	CALL	0F0BH	ACCUM * 10
1240	F5	PUSH	AF	# DIVISIONS = # DIVISIONS - 1
1241	18E2	JR	1225H	TRY AGAIN
1243	F1	POP	AF	ACCUM =
1244	CD180F	CALL	0F18H	ACCUM/10
1247	F5	PUSH	AF	# DIVISIONS = # DIVISIONS + 1
1248	CD4F12	CALL	124FH	REPEAT UNTIL WITHIN RANGE
124B	F1	POP	AF	GET # DIVISIONS
124C	D1	POP	DE	RESTORE DE
124D	B7	OR	A	SET STATUS
124E	C9	RET		RETURN TO CALLER
124F	E7	RST	20H	CHECK TYPE OF ACCUM
1250	EA5E12	JP	PE,125EH	JUMP IF NOT S.P.
1253	017494	LD	BC,9474H	IS
1256	11F823	LD	DE,23F8H	ACCUM
1259	CD0C0A	CALL	0A0CH	≥ 999,999?
125C	1806	JR	1264H	
125E	117413	LD	DE,1374H	IS ACCUM
1261	CD490A	CALL	0A49H	≥ + 1.0D16?
1264	E1	POP	HL	GET RETURN ADDRESS
1265	F24312	JP	P,1243H	JUMP IF RESULT TRUE
1268	E9	JP	(HL)	RETURN
1269	B7	OR	A	STORE
126A	C8	RET	Z	(A)
126B	3D	DEC	A	ZEROES
126C	3630	LD	(HL),30H	IN
126E	23	INC	HL	BUFFER
126F	18F9	JR	126AH	
1271	2004	JR	NZ,1277H	DO AT LEAST ONCE IF NZ
1273	C8	RET	Z	RETURN IF DONE
1274	CD9112	CALL	1291H	APPLY COMMAS AND DECIMAL
1277	3630	LD	(HL),30H	SPORE TRAILING ZERO
1279	23	INC	HL	NEXT LOCATION
127A	3D	DEC	A	ONE LESS TO STORE
127B	18F6	JR	1273H	CONTINUE.
127D	7B	LD	A,E	B = DECIMAL POSITION
127E	82	ADD	A,D	
127F	3C	INC	A	
1280	47	LD	B,A	
1281	3C	INC	A	C = COMMA COUNTER
1282	D603	SUB	03H	
1284	30FC	JR	NC,1282H	
1286	C605	ADD	A,05H	
1288	4F	LD	C,A	
1289	3AD840	LD	A,(40D8H)	RESET COMMA COUNTER
128C	E640	AND	40H	IF COMMAS NOT SELECTED
128E	C0	RET	NZ	
128F	4F	LD	C,A	
1290	C9	RET		RETURN TO CALLER
1291	05	DEC	B	ONE LESS DIGIT ON LEFT
1292	2008	JR	NZ,129CH	SKIP IF NOT AT DECIMAL
1294	362E	LD	(HL),2EH	STORE DECIMAL
1296	22F340	LD	(40F3H),HL	SAVE LOCATION
1299	23	INC	HL	NEXT LOCATION
129A	48	LD	C,B	NO MORE COMMAS
129B	C9	RET		DONE

129C	0D	DEC	C	ONE LESS CHARACTER LEFT TO STORE
129D	C0	RET	NZ	RETURN IF NOT ZERO
129E	362C	LD	(HL), 2CH	STORE A COMMA
12A0	23	INC	HL	NEXT CHARACTER.
12A1	0E03	LD	C, 03H	STORE 3 MORE BEFORE ANOTHER
12A3	C9	RET		CONTINUE
12A4	D5	PUSH	DE	SAVE DE
12A5	E7	RST	20H	CHECK TYPE OF ACCUM
12A6	E2EA12	JP	PO, 12EAH	JUMP IF S.P.
12A9	C5	PUSH	BC	SAVE BC
12AA	E5	PUSH	HL	SAVE HL
12AB	CDFC09	CALL	09FCH	ACCUM 2 = ACCUM
12AE	217C13	LD	HL, 137CH	ACCUM = +0.5D + 00
12B1	CDF709	CALL	09F7H	
12B4	CD770C	CALL	0C77H	ACCUM = ACCUM + ACCUM 2
12B7	AF	XOR	A	CLEAR STATUS
12B8	CD7B0B	CALL	0B7BH	GET WHOLE PART OF ACCUM
12BB	E1	POP	HL	RESTORE HL
12BC	C1	POP	BC	RESTORE BC
12BD	118C13	LD	DE, 138CH	DE => TABLE OF POWERS OF 10
12C0	3E0A	LD	A, 0AH	TABLE HAS 10 ENTRIES
12C2	CD9112	CALL	1291H	APPLY COMMAS & DECIMAL
12C5	C5	PUSH	BC	SAVE COMMA FLAGS
12C6	F5	PUSH	AF	SAVE # ENTRIES LEFT
12C7	E5	PUSH	HL	SAVE POSITION IN BUFFER
12C8	D5	PUSH	DE	SAVE POSITION IN TABLE.
12C9	062F	LD	B, 2FH	B = "0" AT FIRST
12CB	04	INC	B	B = B + 1
12CC	E1	POP	HL	HL = POSITION IN TABLE
12CD	E5	PUSH	HL	SAVE POSITION IN TABLE
12CE	CD480D	CALL	0D48H	ACCUM = ACCUM - TABLE ENTRY
12D1	30F8	JR	NC, 12CBH	LOOP UNTIL ACCUM < 0.
12D3	E1	POP	HL	RESTORE POSITION IN TABLE
12D4	CD360D	CALL	0D36H	ACCUM = ACCUM + TABLE ENTRY
12D7	EB	EX	DE, HL	SAVE TABLE ENTRY LOCATION
12D8	E1	POP	HL	GET POSITION IN BUFFER
12D9	70	LD	(HL), B	STORE CHART.
12DA	23	INC	HL	NEXT LOCATION IN BUFFER
12DB	F1	POP	AF	RESTORE # ENTRIES LEFT
12DC	C1	POP	BC	RESTORE COMMA FLAGS
12DD	3D	DEC	A	ONE LESS ENTRY TO GO.
12DE	20E2	JR	NZ, 12C2H	LOOP UNTIL DONE
12E0	C5	PUSH	BC	SAVE COMMA FLAGS
12E1	E5	PUSH	HL	SAVE POSITION
12E2	211D41	LD	HL, 411DH	MOVE LSB'S TO
12E5	CDB109	CALL	09B1H	MSB POSITION
12E8	180C	JR	12F6H	PROCESS AS S.P.
12EA	C5	PUSH	BC	SAVE COMMA FLAGS
12EB	E5	PUSH	HL	SAVE POSITION
12EC	CD0807	CALL	0708H	ACCUM = ACCUM + 0.5
12EF	3C	INC	A	INCREASE EXPONENT
12F0	CDFB0A	CALL	0AFBH	CONVERT TO INTEGER
12F3	CDB409	CALL	09B4H	SAVE IN ACCUM.
12F6	E1	POP	HL	RESTORE POSITION
12F7	C1	POP	BC	RESTORE COMMA FLAGS
12F8	AF	XOR	A	CLEAR CARRY
12F9	11D213	LD	DE, 13D2H	DE => TABLE OF POWERS OF 10

12FC 3F	CCF		LOOP TWICE
12FD CD9112	CALL	1291H	APPLY COMMAS & DECIMAL
1300 C5	PUSH	BC	SAVE COMMA FLAGS
1301 F5	PUSH	AF	SAVE LOOP FLAGS
1302 E5	PUSH	HL	SAVE POSITION
1303 D5	PUSH	DE	SAVE TABLE ENTRY LOCATION
1304 CDBF09	CALL	09BFH	BCDE = ACCUM
1307 E1	POP	HL	GET TABLE ENTRY LOCATION
1308 062F	LD	B,2FH	B = "0" FIRST TIME AROUND
130A 04	INC	B	B = B + 1
130B 7B	LD	A,E	CDE = CDE - ENTRY
130C 96	SUB	(HL)	
130D 5F	LD	E,A	
130E 23	INC	HL	
130F 7A	LD	A,D	
1310 9E	SBC	A, (HL)	
1311 57	LD	D,A	
1312 23	INC	HL	
1313 79	LD	A,C	
1314 9E	SBC	A, (HL)	
1315 4F	LD	C,A	
1316 2B	DEC	HL	BACK UP TO
1317 2B	DEC	HL	START OF TABLE ENTRY
1318 30F0	JR	NC,130AH	LOOP UNTIL CDE < 0
131A CDB707	CALL	07B7H	CDE = CDE + ENTRY
131D 23	INC	HL	POINT TO NEXT ENTRY
131E CDB409	CALL	09B4H	ACCUM = BCDE
1321 EB	EX	DE,HL	SAVE TABLE ENTRY LOCATION
1322 E1	POP	HL	GET POSITION
1323 70	LD	(HL),B	STORE DIGIT
1324 23	INC	HL	NEXT POSITION
1325 F1	POP	AF	RESTORE LOOP FLAG
1326 C1	POP	BC	RESTORE COMMA FLAG
1327 38D3	JR	C,12FCH	LOOP UNTIL DONE
1329 13	INC	DE	POINT TO
132A 13	INC	DE	NEXT TABLE
132B 3E04	LD	A,04H	USE LAST 4 ENTRIES OF IT
132D 1806	JR	1335H	CONTINUE
132F D5	PUSH	DE	SAVE DE
1330 11D813	LD	DE,13D8H	DE => TABLE OF POWERS OF 10
1333 3E05	LD	A,05H	TABLE HAS 5 ENTRIES
1335 CD9112	CALL	1291H	APPLY COMMAS & DECIMAL
1338 C5	PUSH	BC	SAVE COMMA FLAGS
1339 F5	PUSH	AF	SAVE # ENTRIES LEFT
133A E5	PUSH	HL	SAVE POSITION.
133B EB	EX	DE,HL	HL => TABLE ENTRY
133C 4E	LD	C, (HL)	
133D 23	INC	HL	
133E 46	LD	B, (HL)	
133F C5	PUSH	BC	SAVE TABLE ENTRY
1340 23	INC	HL	
1341 E3	EX	(SP),HL	HL = TABLE ENTRY SAVE HL
1342 EB	EX	DE,HL	DE = TABLE ENTRY
1343 2A2141	LD	HL, (4121H)	HL = ACCUM
1346 062F	LD	B,2FH	B INITIALLY "0"
1348 04	INC	B	B = B+1
1349 7D	LD	A,L	HL = HL - DE

```

134A 93      SUB      E
134B 6F      LD        L,A
134C 7C      LD        A,H
134D 9A      SBC      A,D
134E 67      LD        H,A
134F 30F7    JR        NC,1348H    LOOP IF HL > 0
1351 19      ADD      HL,DE        HL = HL + DE
1352 222141  LD        (4121H),HL  SAVE HL IN ACCUM
1355 D1      POP      DE        RESTORE POSITION IN TABLE
1356 E1      POP      HL        RESTORE POSITION IN BUFFER
1357 70      LD        (HL),B     STORE DIGIT
1358 23      INC      HL        NEXT CHARACTER
1359 F1      POP      AF        RESTORE # ENTRIES LEFT
135A C1      POP      BC        RESTORE COMMA FLAGS
135B 3D      DEC      A         ONE LESS ENTRY TO DO
135C 20D7    JR        NZ,1335H   LOOP UNTIL DONE
135E CD9112  CALL     1291H      APPLY COMMAS & DECIMAL
1361 77      LD        (HL),A    MARK END OF STRING
1362 D1      POP      DE        RESTORE DE
1363 C9      RET
    
```

NUMERICAL DATA

```

1364 00000000F90215A2 + 1D + 10
136C FDF9F31A95F63B2  + 1D + 15
1374 FEFF03BFC91B0EB6 + 1D + 16
137C 00000000        + 0.5D + 00
1380 00000080        + 0.5E + 00
1384 000004BFC91B   + 1D + 16
138A 0EB6            + 0.502778E + 00
138C 0080C6A47E8D03 + 10**15      )
1393 00407A10F35A00 + 10**14      )
139A 00A0724E180900 + 10**13      )
13A1 0010A5D4E80000 + 10**12      )
13A8 00E87648170000 + 10**11      ) 7 WORD INTEGERS FOR USE
13AF 00E40B54020000 + 10**10      ) WITH D.P.
13B6 00CA9A3B000000 + 10**9       )
13BD 00E1F505000000 + 10**8       )
13C4 80969800000000 + 10**7       )
13CB 40420F00000000 + 10**6       )
13D2 A08601          + 10**5       ) 3 WORD INTEGERS FOR USE
13D5 102700          + 10**4       ) WITH S.P.
13D8 1027            + 10**4       ) 2 WORD INTEGERS FOR USE
13DA E803            + 10**3       ) WITH INTEGERS
13DC 6400            + 10**2       )
13DE 0A00            + 10**1       )
13E0 0100            + 10**0       )
    
```

```

13E2 218209    LD        HL,0982H    SET RETURN THROUGH
13E5 E3        EX        (SP),HL  NEGATION ROUTINE
13E6 E9        JP        (HL)      RETURN TO CALLER
    
```

SQUARE-ROOT ROUTINE: ACCUM = SQR (ACCUM)

```

13E7 CDA409    CALL     09A4H      SAVE ARGUMENT ON STACK
13EA 218013    LD        HL,1380H  ACCUM = 0.5
13ED CDB109    CALL     09B1H
13F0 1803     JR        13F5H    COMPUTE (ARGUMENT)**0.5
    
```


EXPONENTIATION ROUTINE: ACCUM = (SP)ACC**

13F2	CDB10A	CALL	0AB1H	MAKE ACCUM S.P.
13F5	C1	POP	BC	BCDE = (SP)
13F6	D1	POP	DE	
13F7	CD5509	CALL	0955H	GET SIGN OF ACCUM.
13FA	78	LD	A,B	GET EXPONENT OF BCDE
13FB	283C	JR	Z,1439H	IF ACCUM = 0 THEN JUMP
13FD	F20414	JP	P,1404H	SKIP IF ACCUM > 0
1400	B7	OR	A	TEST BCDE
1401	CA9A19	JP	Z,199AH	/0 ERROR IF ZERO
1404	B7	OR	A	TEST BCDE
1405	CA7907	JP	Z,0779H	ANSWER = 0 IF ZERO
1408	D5	PUSH	DE	SAVE BASE
1409	C5	PUSH	BC	
140A	79	LD	A,C	CHECK SIGN OF BASE
140B	F67F	OR	7FH	
140D	CDBF09	CALL	09BFH	BCDE = EXPONENT.
1410	F22114	JP	P,1421H	JUMP IF BASE POSITIVE
1413	D5	PUSH	DE	SAVE EXPONENT
1414	C5	PUSH	BC	
1415	CD400B	CALL	0B40H	ACCUM = INT(EXPONENT)
1418	C1	POP	BC	RESTORE EXPONENT
1419	D1	POP	DE	
141A	F5	PUSH	AF	SAVE LSB OF EXPONENT
141B	CD0C0A	CALL	0A0CH	EXPONENT = INT(EXPONENT)
141E	E1	POP	HL	GET LSB OF EXPONENT
141F	7C	LD	A,H	IS EXPONENT EVEN?
1420	1F	RRA		
1421	E1	POP	HL	ACCUM = BASE
1422	222341	LD	(4123H),HL	
1425	E1	POP	HL	
1426	222141	LD	(4121H),HL	
1429	DCE213	CALL	C,13E2H	ADJUST SIGN
142C	CC8209	CALL	Z,0982H	
142F	D5	PUSH	DE	SAVE EXPONENT
1430	C5	PUSH	BC	
1431	CD0908	CALL	0809H	ACCUM = LOG(BASE)
1434	C1	POP	BC	RESTORE EXPONENT
1435	D1	POP	DE	
1436	CD4708	CALL	0847H	ACCUM = LOG (BASE) *EXPONENT

EXP (ACCUM)

1439	CDA409	CALL	09A4H	SAVE ARGUMENT
143C	013881	LD	BC,8138H	ACCUM =
143F	113BAA	LD	DE,0AA3BH	ARGUMENT
1442	CD4708	CALL	0847H	* 1.4427
1445	3A2441	LD	A,(4124H)	OUT OF RANGE?
1448	FE88	CP	88H	
144A	D23109	JP	NC,0931H	OV ERROR IF SO
144D	CD400B	CALL	0B40H	CHECK AGAIN
1450	C680	ADD	A,80H	
1452	C602	ADD	A,02H	
1454	DA3109	JP	C,0931H	OV ERROR IF SO.
1457	F5	PUSH	AF	SAVE INTEGER VALUE OF ARGUMENT
1458	21F807	LD	HL,07F8H	ACCUM =
145B	CD0B07	CALL	070BH	ACCUM + 1
145E	CD4108	CALL	0841H	ACCUM = ACCUM * 0.693115

1461	F1	POP	AF	RESTORE INT(ARGUMENT)
1462	C1	POP	BC	RESTORE ARGUMENT
1463	D1	POP	DE	
1464	F5	PUSH	AF	SAVE INT(ARGUMENT)
1465	CD1307	CALL	0713H	ACCUM = ACCUM - ARGUMENT
1468	CD8209	CALL	0982H	MAKE ACCUM POSITIVE
146B	217914	LD	HL,1479H	EXPAND POWER
146E	CDA914	CALL	14A9H	SERIES
1471	110000	LD	DE,0000H	BCDE = 2/ INT(ARGUMENT)
1474	C1	POP	BC	
1475	4A	LD	C,D	
1476	C34708	JP	0847H	RESULT = ACCUM * BASE

DATA FOR EXP(X) EXPANSION

1479	08			8 COEFFICIENTS
147A	402E9474			- 1/7
147E	704F2E77			+ 1/6
1482	6E02887A			- 1/5
1486	E6A02A7C			+ 1/4
148A	50AAAA7E			- 1/3
148E	FFFF7F7F			+ 1/2
1492	00008081			- 1
1496	00000081			+ 1
149A	CDA409	CALL	09A4H	SAVE ARGUMENT
149D	11320C	LD	DE,0C32H	RETURN THROUGH
14A0	D5	PUSH	DE	A MULTIPLICATION
14A1	E5	PUSH	HL	SAVE DATA POINTER
14A2	CDBF09	CALL	09BFH	SQUARE ARGUMENT
14A5	CD4708	CALL	0847H	
14A8	E1	POP	HL	RESTORE DATA POINTER
14A9	CDA409	CALL	09A4H	SAVE ARGUMENT SQUARED
14AC	7E	LD	A,(HL)	GET # COEFFICIENTS
14AD	23	INC	HL	POINT TO FIRST COEFFICIENTS
14AE	CDB109	CALL	09B1H	ACCUM = FIRST COEFFICIENT
14B1	06			MASK TO 14B3H
14B2	F1	POP	AF	RESTORE # COEFFICIENTS LEFT
14B3	C1	POP	BC	RESTORE ARGUMENT
14B4	D1	POP	DE	RAISED TO A POWER
14B5	3D	DEC	A	ONE LESS TO DO
14B6	C8	RET	Z	RETURN IF DONE
14B7	D5	PUSH	DE	SAVE ARGUMENT
14B8	C5	PUSH	BC	RAISED TO POWER
14B9	F5	PUSH	AF	SAVE # COEFFICIENTS LEFT
14BA	E5	PUSH	HL	SAVE COEFFICIENTS POINTER
14BB	CD4708	CALL	0847H	ACCUM = ACCUM*ARGUMENT**N
14BE	E1	POP	HL	RESTORE COEFFICIENT POINTER
14BF	CDC209	CALL	09C2H	GET NEXT COEFFICIENT
14C2	E5	PUSH	HL	SAVE COEFFICIENT POINTER
14C3	CD1607	CALL	0716H	ADD COEFFICIENT TO ACCUM
14C6	E1	POP	HL	RESTORE COEFFICIENT POINTER
14C7	18E9	JR	14B2H	LOOP UNTIL DONE
14C9	CD7F0A	CALL	0A7FH	HL = ACCUM = CINT(ACCUM)
14CC	7C	LD	A,H	ARGUMENT NEGATIVE
14CD	B7	OR	A	
14CE	FA4A1E	JP	M,1E4AH	FC ERROR IF SO
14D1	B5	OR	L	ARGUMENT ZERO? (RND(0))
14D2	CAF014	JP	Z,14F0H	JUMP IF SO

14D5	E5	PUSH	HL	SAVE ARGUMENT
14D6	CDF014	CALL	14F0H	COMPUTE RANDOM NUMBER
14D9	CDBF09	CALL	09BFH	GET RANDOM NUMBER
14DC	EB	EX	DE,HL	SAVE NUMBER &
14DD	E3	EX	(SP),HL	RESTORE ARGUMENT
14DE	C5	PUSH	BC	
14DF	CDCF0A	CALL	0ACFH	CONVERT ARGUMENT TO S.P.
14E2	C1	POP	BC	RESTORE RANDOM NUMBER
14E3	D1	POP	DE	
14E4	CD4708	CALL	0847H	RESULT = ARGUMENT
14E7	21F807	LD	HL,07F8H	*RANDOM NUMBER
14EA	CD0B07	CALL	070BH	+1
14ED	C3400B	JP	0B40H	CONVERT TO INTEGER & RETURN
14F0	219040	LD	HL,4090H	HL => SEED # 2
14F3	E5	PUSH	HL	SAVE POINTER
14F4	110000	LD	DE,0000H	CDE = 0
14F7	4B	LD	C,E	
14F8	2603	LD	H,03H	# OUTER INTERATIONS
14FA	2E08	LD	L,08H	# INNER INTERATIONS
14FC	EB	EX	DE,HL	CDE
14FD	29	ADD	HL,HL	=
14FE	EB	EX	DE,HL	CDE
14FF	79	LD	A,C	*
1500	17	RLA		2
1501	4F	LD	C,A	
1502	E3	EX	(SP),HL	SAVE COUNTERS TO GET POINTER
1503	7E	LD	A,(HL)	ROTATE BYTE
1504	07	RLCA		IN WORKSPACE
1505	77	LD	(HL),A	LEFT, CHECK BIT 7
1506	E3	EX	(SP),HL	GET COUNTER & SAVE POINTER
1507	D21615	JP	NC,1516H	SKIP IF BIT WAS RESET
150A	E5	PUSH	HL	SAVE COUNTERS
150B	2AAA40	LD	HL,(40AAH)	CDE
150E	19	ADD	HL,DE	=
150F	EB	EX	DE,HL	CDE
1510	3AAC40	LD	A,(40ACH)	+
1513	89	ADC	A,C	SEED # 1
1514	4F	LD	C,A	
1515	E1	POP	HL	RESTORE COUNTERS
1516	2D	DEC	L	ONE LESS INNER INTERATION
1517	C2FC14	JP	NZ,14FCH	LOOP UNTIL DONE
151A	E3	EX	(SP),HL	POINT TO
151B	23	INC	HL	NEXT
151C	E3	EX	(SP),HL	WORKSPACE BYTE
151D	25	DEC	H	ONE LESS OUTER INTERATION
151E	C2FA14	JP	NZ,14FAH	LOOP UNTIL DONE
1521	E1	POP	HL	FIX STACK
1522	2165B0	LD	HL,0B065H	SEED # 1
1525	19	ADD	HL,DE	=
1526	22AA40	LD	(40AAH),HL	CDE
1529	CDEF0A	CALL	0AEFH	+
152C	3E05	LD	A,05H	372837
152E	89	ADC	A,C	
152F	32AC40	LD	(40ACH),A	
1532	EB	EX	DE,HL	CDE = RANDOM # = SEED #1
1533	0680	LD	B,80H	SET EXPONENT (0 ≤ X < 1)
1535	212541	LD	HL,4125H	POINT TO ACCUM

```

1538 70      LD      (HL),B      SIGN POSITIVE
1539 2B      DEC      HL      POINT TO EXPONENT
153A 70      LD      (HL),B      SAVE EXPONENT
153B 4F      LD      C,A      SET UP
153C 0600    LD      B,00H      TO NORMALIZE
153E C36507  JP      0765H      NORMALIZE & RETURN

```

ACCUM = COS (ACCUM)

```

1541 218B15  LD      HL,158BH      ANGLE = ANGLE + 90 DEGREES
1544 CD0B07  CALL   070BH

                                ACCUM = SIN(ACCUM)
1547 CDA409  CALL   09A4H      SAVE ANGLE
154A 014983  LD      BC,8349H      ACCUM = 2 PIE
154D 11DB0F  LD      DE,0FDBH
1550 CDB409  CALL   09B4H
1553 C1      POP    BC      BCDE = ANGLE
1554 D1      POP    DE
1555 CDA208  CALL   08A2H      ANGLE
1558 CDA409  CALL   09A4H      =
155B CD400B  CALL   0B40H      - ANGLE
155E C1      POP    BC      MOD
155F D1      POP    DE      2 PIE
1560 CD1307  CALL   0713H      (ANGLE = ANGLE - 180 DEGREES)
1563 218F15  LD      HL,158FH      ANGLE = ANGLE - 90 DEGREE
1566 CD1007  CALL   0710H
1569 CD5509  CALL   0955H      CHECK SIGN OF ANGLE
156C 37      SCF
                                NO SIGN INVERSION IF POSITIVE
156D F27715  JP      P,1577H      JUMP IF POSITIVE
1570 CD0807  CALL   0708H      ANGLE = ANGLE + 180 DEGREES
1573 CD5509  CALL   0955H      CHECK SIGN OF ANGLE
1576 B7      OR     A      INVERT SIGN WHEN DONE
1577 F5      PUSH  AF      SAVE SIGN FLAG
1578 F48209  CALL   P,0982H      MAKE ANGLE NEGATIVE IF POSITIVE
157B 218F15  LD      HL,158FH      ANGLE = ANGLE + 90 DEGREES
157E CD0B07  CALL   070BH
1581 F1      POP    AF      GET SIGN FLAG
1582 D48209  CALL   NC,0982H      INVERT SIGN IF NECESSARY
1585 219315  LD      HL,1593H      EXPAND POWER SERIES
1588 C39A14  JP      149AH      AND RETURN TO CALLER

```

DATA FOR SIN/COS

```

158B DB0F4981 1.5708      (PIE /2)
158F 0000007F 0.25        (1/4)
1593 05        5          NUMBER OF COEFFICIENTS
1594 BAD71E86 39.7107
1598 64269987 -76.575
159C 58342387 81.6022
15A0 E05DA586 -41.3417
15A4 DA0F4983 6.28319     (2 PIE)

```

ACCUM = TAN (ACCUM)

```

15A8 CDA409  CALL   09A4H      SAVE ANGLE
15AB CD4715  CALL   1547H      COMPUTE SIN (ANGLE)
15AE C1      POP    BC      GET ANGLE TO
15B0 CDA409  CALL   09A4H      SAVE SIN (ANGLE)
15B3 EB      EX     DE,HL      BCDE = ANGLE

```

15B4	CDB409	CALL	09B4H	ACCUM = ANGLE
15B7	CD4115	CALL	1541H	COMPUTE COS (ANGLE)
15BA	C3A008	JP	08A0H	RESULT = SIN/COS. DONE

ACCUM = ATN (ACCUM)

15BD	CD5509	CALL	0955H	CHECK SIGN OF ARGUMENT
15C0	FCE213	CALL	M,13E2H	CHANGE SIGN ON RETURN IF NEGATIVE
15C3	FC8209	CALL	M,0982H	CHANGE SIGN IF NEGATIVE
15C6	3A2441	LD	A,(4124H)	IS ARGUMENT
15C9	FE81	CP	81H	LESS THAN 1.0?
15CB	380C	JR	C,15D9H	SKIP IF SO
15CD	010081	LD	BC,8100H	BCDE = +1.0
15D0	51	LD	D,C	
15D1	59	LD	E,C	
15D2	CDA208	CALL	08A2H	ACCUM = 1/ACCUM
15D5	211007	LD	HL,0710H	ON EXIT, ACCUM =
15D8	E5	PUSH	HL	ACCUM + PIE /2
15D9	21E315	LD	HL,15E3H	EXPAND
15DC	CD9A14	CALL	149AH	POWER SERIES
15DF	218B15	LD	HL,158BH	POINT TO PIE /2
15E2	C9	RET		DONE. FIX SIGN IF NECESSARY

DATA FOR ATN

15E3	09	9	# COEFFICIENTS
15E4	4AD73B78	2.86623E-03	COEFFICIENTS
15E8	026E847B	-0.0161657	
15EC	FEC12F7C	0.0429096	
15F0	74319A7D	-0.0752896	
15F4	843D5A7D	0.106563	
15F8	C87F917E	-0.142089	
15FC	E4BB4C7E	0.199936	
1600	6CAAAA7F	-0.333331	
1604	00000081	1.0	

FUNCTION ADDRESS TABLE

1608	8A09	098A	SGN
160A	370B	0B37	INT
160C	7709	0977	ABS
160E	D427	27D4	FRE
1610	EF2A	2AEF	INP
1612	F527	27F5	POS
1614	E713	13E7	SQR
1616	C914	14C9	RND
1618	0908	0809	LOG
161A	3914	1439	EXP
161C	4115	1541	COS
161E	4715	1547	SIN
1620	A815	15A8	TAN
1622	BD15	15BD	ATN
1624	AA2C	2CAA	PEEK
1626	5241	4152	CVI
1628	5841	4158	CVS
162A	5E41	415E	CVD
162C	6141	4161	EOF
162E	6441	4164	LOC
1630	6741	4167	LOF
1632	6A41	416A	MKI\$

1634 6D41	416D	MKS\$
1636 7041	4170	MKD\$
1638 7F0A	0A7F	CINT
163A B10A	0AB1	CSNG
163C DB0A	0ADB	CDBL
163E 260B	0B26	FIX
1640 032A	2A03	LEN
1642 3628	2836	STR\$
1644 C52A	2AC5	VAL
1646 0F2A	2A0F	ASC
1648 1F2A	2A1F	CHR\$
164A 612A	2A61	LEFT\$
164C 912A	2A91	RIGHT\$
164E 9A2A	2A9A	MID\$

RESERVED WORD TABLE

1650 C54E44	END	80 SEE ADDRESS
1653 C64F52	FOR	81 TABLE AT 1822H
1656 D245534554	RESET	82
165B D34554	SET	83
165E C34C53	CLS	84
1661 C34D44	CMD	85
1664 D2414E444F4D	RANDOM	86
166A CE455854	NEXT	87
166E C4415441	DATA	88
1672 C94E505554	INPUT	89
1677 C4494D	DIM	8A
167A D2454144	READ	8B
167E CC4554	LET	8C
1681 C74F544F	GOTO	8D
1685 D2554E	RUN	8E
1688 C946	IF	8F
168A D24553544F5245	RESTORE	90
1691 C74F535542	GOSUB	91
1696 D2455455524E	RETURN	92
169C D2454D	REM	93
169F D3544F50	STOP	94
16A3 C54C5345	ELSE	95
16A7 D4524F4E	TRON	96
16AB D4524F4646	TROFF	97
16B0 C44546535452	DEFSTR	98
16B6 C44546494E54	DEFINT	99
16BC C44546534E47	DEFSNG	9A
16C2 C4454644424C	DEFDBL	9B
16C8 CC494E45	LINE	9C
16CC C5444954	EDIT	9D
16D0 C552524F52	ERROR	9E
16D5 D24553554D45	RESUME	9F
16DB CF5554	OUT	A0
16DE CF4E	ON	A1
16E0 CF50454E	OPEN	A2
16E4 C649454C44	FIELD	A3
16E9 C74554	GET	A4
16EC D05554	PUT	A5
16EF C34C4F5345	CLOSE	A6
16F4 CC4F4144	LOAD	A7
16F8 CD45524745	MERGE	A8

16FD	CE414D45	NAME	A9
1701	CB494C4C	KILL	AA
1705	CC534554	LSET	AB
1709	D2534554	RSET	AC
170D	D3415645	SAVE	AD
1711	D3595354454D	SYSTEM	AE
1717	CC5052494E54	LPRINT	AF
171D	C44546	DEF	B0
1720	D04F4B45	POKE	B1
1724	D052494E54	PRINT	B2
1729	C34F4E54	CONT	B3
172D	CC495354	LIST	B4
1731	CC4C495354	LLIST	B5
1736	C4454C455445	DELETE	B6
173C	C155544F	AUTO	B7
1740	C34C454152	CLEAR	B8
1745	C34C4F4144	CLOAD	B9
174A	C353415645	CSAVE	BA
174F	CE4557	NEW	BB
1752	D4414228	TAB (BC
1756	D44F	TO	BD
1758	C64E	FN	BE
175A	D553494E47	USING	BF
175F	D64152505452	VARPTR	C0
1765	D55352	USR	C1
1768	C5524C	ERL	C2
176B	C55252	ERR	C3
176E	D35452494E4724	STRING\$	C4
1775	C94E535452	INSTR	C5
177A	D04F494E54	POINT	C6
177F	D4494D4524	TIME\$	C7
1784	CD454D	MEM	C8
1787	C94E4B455924	INKEY\$	C9
178D	D448454E	THEN	CA
1791	CE4F54	NOT	CB
1794	D3544550	STEP	CC
1798	AB	+	CD
1799	AD	-	CE
179A	AA	*	CF
179B	AF	/	D0
179C	DB	**	D1
179D	C14E44	AND	D2
17A0	CF52	OR	D3
17A2	BE	>	D4
17A3	BD	=	D5
17A4	BC	<	D6

TOKEN

17A5	D3474E	SGN	D7 EMBEDDED FUNCTIONS
17A8	C94E54	INT	D8 SEE TABLE 1608H
17AB	C14253	ABS	D9
17AE	C65245	FRE	DA
17B1	C94E50	INP	DB
17B4	D04F53	POS	DC
17B7	D35152	SQR	DD
17BA	D24E44	RND	DE
17BD	CC4F47	LOG	DF

17C0	C55850	EXP	E0
17C3	C34F53	COS	E1
17C6	D3494E	SIN	E2
17C9	D4414E	TAN	E3
17CC	C1544E	ATN	E4
17CF	D045454B	PEEK	E5
17D3	C35649	CVI	E6
17D6	C35653	CVS	E7
17D9	C35644	CVD	E8
17DC	C54F46	EOF	E9
17DF	CC4F43	LOC	EA
17E2	CC4F46	LOF	EB
17E5	CD4B4924	MKI\$	EC
17E9	CD4B5324	MKS\$	ED
17ED	CD4B4424	MKD\$	EE
17F1	C3494E54	CINT	EF
17F5	C3534E47	CSNG	F0
17F9	C344424C	CDBL	F1
17FD	C64958	FIX	F2
1800	CC454E	LEN	F3
1803	D3545224	STR\$	F4
1807	D6414C	VAL	F5
180A	C15343	ASC	F6
180D	C3485224	CHR\$	F7
1811	CC45465424	LEFT\$	F8
1816	D24947485424	RIGHT\$	F9
181C	CD494424	MID\$	FA
1820	A7	/	FB
1821	80	80H	END MARKER

ROUTINE ADDRESS TABLE

1822	AE1D	1DAE	END
1824	A11C	1CA1	FOR
1826	3801	0138	RESET
1828	3501	0135	SET
181A	C901	01C9	CLS
182C	7341	4173	CMD
182E	D301	01D3	RANDOM
1830	B622	22B6	NEXT
1832	051F	1F05	DATA
1834	9A21	219A	INPUT
1836	0826	2608	DIM
1838	EF21	21EF	READ
183A	211F	1F21	LET
183C	C21E	1EC2	GOTO
183E	A31E	1EA3	RUN
1840	3920	2039	IF
1842	911D	1D91	RESTORE
1844	B11E	1EB1	GOSUB
1846	DE1E	1EDE	RETURN
1848	071F	1F07	REM
184A	A91D	1DA9	STOP
184C	071F	1F07	ELSE
184E	F71D	1DF7	TRON
1850	F81D	1DF8	TROFF
1852	001E	1E00	DEFSTR
1854	031E	1E03	DEFINT

1856	061E	1E06	DEFSNG
1858	091E	1E09	DEFDBL
185A	A341	41A3	LINE
185C	602E	2E60	EDIT
185E	F41F	1FF4	ERROR
1860	AF1F	1FAF	RESUME
1862	FB2A	2AFB	OUT
1864	6C1F	1F6C	ON
1866	7941	4179	OPEN
1868	7C41	417C	FIELD
186A	7F41	417F	GET
186C	8241	4182	PUT
186E	8541	4185	CLOSE
1870	8841	4188	LOAD
1872	8B41	418B	MERGE
1874	8E41	418E	NAME
1876	9141	4191	KILL
1878	9741	4197	LSET
187A	9A41	419A	RSET
187C	A041	41A0	SAVE
187E	B202	02B2	SYSTEM
1880	6720	2067	LPRINT
1802	5B41	4158	DEF
1884	B12C	2CB1	POKE
1886	6F20	206F	PRINT
1888	E41D	1DE4	CONT
188A	2E2B	2B2E	LIST
188C	292B	2B29	LLIST
188E	C62B	2BC6	DELETE
1890	0820	2008	AUTO
1892	7A1E	1E7A	CLEAR
1894	1F2C	2C1F	CLAD
1896	F52B	2BF5	CSAVE
1898	491B	1B49	NEW

PRECEDENCE TABLE

189A	79	79	+	PRECEDENCE
189B	79	79	-	TABLE
189C	7C	7C	*	
189D	7C	7C	/	
189E	7F	7F	/**	
189F	50	50	AND	
18A0	46	46	OR	

MATHEMATICAL OPERATOR INDEX

18A1	DB0A	0ADB	CDBL
18A3	0000	0000	UNUSED
18A5	7F0A	0A7F	CINT
18A7	F40A	0AF4	CHECK IF STRING
18A9	B10A	0AB1	CSNG
18AB	770C	0C77	D.P. ADDITION
18AD	700C	0C70	SUBTRACTION
18AF	A10D	0DA1	MULTIPLICATION
18B1	E50D	0DE5	DIVISION
18B3	780A	0A78	COMPARE
18B5	1607	0716	S.P. ADDITION
18B7	1307	0713	SUBTRACTION

18B9 4708	0847	MULTIPLICATION
18BB A208	08A2	DIVISION
18BD 0C0A	0A0C	COMPARE
18BF D20B	0BD2	INTEGER ADDITION
18C1 C70B	0BC7	SUBTRACTION
18C3 F20B	0BF2	MULTIPLICATION
18C5 9024	2490	DIVISION
18C7 390A	0A39	COMPARE

ERROR CODES

18C9 4E46	NF	1	NEXT W/O FOR
18CB 534E	SN	2	SYNTAX
18CD 5247	RG	3	RETURN W/O GOSUB
18CF 4F44	OD	4	OUT OF DATA
18D1 4643	FC	5	FUNCTION CALL
18D3 4F56	OV	6	OVERFLOW
18D5 4F4D	OM	7	OUT OF MEMORY
18D7 554C	UL	8	LINE NOT FOUND
18D9 4253	BS	9	BAD SUBSCRIPT
18DB 4444	DD	10	DOUBLE DIMENSION
18DD 2F30	/0	11	DIVIDE BY ZERO
18DF 4944	ID	12	ILLEGAL DIRECT
18E1 544D	TM	13	TYPE MISMATCH
18E3 4F53	OS	14	OUT OF STRING SPACE
18E5 4C53	LS	15	STRING OVERFLOW
18E7 5354	ST	16	STRING OVER-COMPLEX
18E9 434E	CN	17	CAN'T CONTINUE
18EB 4E52	NR	18	NO RESUME
18ED 5257	RW	19	RESUME W/O ERROR
18EF 5545	UE	20	UNPRINTABLE ERROR
18F1 4D4F	MO	21	MISSING OPERAND
18F3 4644	FD	22	FILE DATA BAD
18F5 4C33	L3	23	DISK BASIC ONLY

RAM INITIALIZATION FOR 4080 - 4066

18F7 D600	SUB	00H	(4080)	PART OF
18F9 6F	LD	L,A		DIVISION SUBROUTINE
18FA 7C	LD	A,H		00S GET REPLACED
18FB DE00	SBC	A,00H		BY DATA IN RAM
18FD 67	LD	H,A		
18FE 78	LD	A,B		
18FF DE00	SBC	A,00H		
1901 47	LD	B,A		
1902 3E00	LD	A,00H		
1904 C9	RET			
1905 4A1E	1E4A		(408E)	USR0 VECTOR (FC ERROR)
1907 40E64D	4DE640		(4090)	RANDOM # SEED #2
190A DB00	IN	A,(00H)	(4093)	INPUT FROM GIVEN PORT
190C C9	RET			DONE
190D D300	OUT	(00H),A	(4096)	OUTPUT TO GIVEN PORT
190F C9	RET			DONE
1910 00000000				
1914 403000				
1917 4C44	444C		(4060)	TOP OF MEMORY
1919 FEFF	FFFE		(4062)	CURRENT BASIC LINE #
191B E943	43E9		(4064)	PROGRAM START

MESSAGES

191D	204572726F7200			" ERROR" 00
1924	20696E2000			" IN " 00
1929	52454144590D00			"READY" 0D 00
1930	427265616B00			"BREAK" 00
1936	210400	LD	HL,0004H	BACK UP 4 BYTES
1939	39	ADD	HL,SP	INTO THE STACK
193A	7E	LD	A, (HL)	GET BYTE
193B	23	INC	HL	POINT TO NEXT BYTE
193C	FE81	CP	81H	FOR DATA?
193E	C0	RET	NZ	RETURN IF NOT
193F	4E	LD	C, (HL)	GET COUNTER POINTER
1940	23	INC	HL	FROM DATA BLOCK
1941	46	LD	B, (HL)	
1942	23	INC	HL	
1943	E5	PUSH	HL	SAVE POSITION IN STACK
1944	69	LD	L,C	HL = COUNTER POINTER
1945	60	LD	H,B	
1946	7A	LD	A,D	WAS THERE A VARIABLE
1947	B3	OR	E	SPECIFIED IN THE NEXT?
1948	EB	EX	DE,HL	USE FOUND VARIABLE IF NOT
1949	2802	JR	Z,194DH	SKIP IF NO VARIABLE SPECIFIED
194B	EB	EX	DE,HL	USE GIVEN VARIABLE?
194C	DF	RST	18H	GIVEN SAME AS FOUND?
194D	010E00	LD	BC,000EH	DISPLACEMENT PAST FOR DATA
1950	E1	POP	HL	RESTORE POSITION IN STACK
1951	C8	RET	Z	RETURN IF NO GIVEN
				OR GIVEN SAME AS FOUND
1952	09	ADD	HL,BC	MOVE PAST FOR BLOCK
1953	18E5	JR	193AH	LOOP
1955	CD6C19	CALL	196CH	CHECK MEMORY SPACE
1958	C5	PUSH	BC	DE = START OF DATA BLOCK
1959	E3	EX	(SP),HL	HL = END OF DATA BLOCK
195A	C1	POP	BC	BC = END OF DESTINATION BLOCK
195B	DF	RST	18H	DONE YET?
195C	7E	LD	A, (HL)	GET BYTE
195D	02	LD	(BC),A	SAVE BYTE
195E	C8	RET	Z	RETURN IF DONE
195F	0B	DEC	BC	NEXT
1960	2B	DEC	HL	LOCATIONS
1961	18F8	JR	195BH	LOOP UNTIL DONE
1963	E5	PUSH	HL	SAVE HL
1964	2AFD40	LD	HL,(40FDH)	HL=> START OF FREE MEMORY
1967	0600	LD	B,00H	BC = # WORDS REQUESTED
1969	09	ADD	HL,BC	HL => NEW START
196A	09	ADD	HL,BC	OF FREE MEMORY
196B	3E			MASK TO 196DH
196C	E5	PUSH	HL	SAVE HL
196D	3EC6	LD	A,0C6H	HL = 0FFC6H
196F	95	SUB	L	- START OF FREE MEMORY
1970	6F	LD	L,A	
1971	3EFF	LD	A,0FFH	
1973	9C	SBC	A,H	
1974	3804	JR	C,197AH	OM ERROR IF START >FFC6
1976	67	LD	H,A	
1977	39	ADD	HL,SP	COMPARE HL & SP

1978	E1	POP	HL	RESTORE HL
1979	D8	RET	C	RETURN IF SP > FREE START
197A	1E0C	LD	E,0CH	OM ERROR 7
197C	1824	JR	19A2H	GO TO ERROR ROUTINE

CHECK FOR NO RESUME AND END OF PROGRAM

197E	2AA240	LD	HL, (40A2H)	DIRECT
1981	7C	LD	A,H	COMMAND?
1982	A5	AND	L	
1983	3C	INC	A	
1984	2808	JR	Z,198EH	END EXECUTION IF SO
1986	3AF240	LD	A, (40F2H)	IN ERROR TRAP?
1989	B7	OR	A	
198A	1E22	LD	E,22H	POSSIBLE NR ERROR
198C	2014	JR	NZ,19A2H	ERROR IF IN TRAP
198E	C3C11D	JP	1DC1H	END EXECUTION

ERROR ROUTINE

1991	2ADA40	LD	HL, (40DAH)	MAKE DATA LINE
1994	22A240	LD	(40A2H), HL	THE CURRENT LINE.
1997	1E02	LD	E,02H	SYNTAX ERROR
1999	01			MASK TO 19A2H
199A	1E14	LD	E,14H	/0 ERROR
199C	01			MASK TO 19A2H
199D	1E00	LD	E,00H	NEXT W/O FOR ERROR
199F	01			MASK TO 19A2H
19A0	1E24	LD	E,24H	RESUME W/O ERROR
19A2	2AA240	LD	HL, (40A2H)	GET CURRENT LINE
19A5	22EA40	LD	(40EAH), HL	SET ERL
19A8	22EC40	LD	(40ECH), HL	SET ERL
19AB	01B419	LD	BC,19B4H	CONTINUATION ADDRESS
19AE	2AE840	LD	HL, (40E8H)	GET STACK POINTER
19B1	C39A1B	JP	1B9AH	INITIALIZE & RETURN
19B4	C1	POP	BC	
19B5	7B	LD	A,E	A = C = ERROR CODE
19B6	4B	LD	C,E	
19B7	329A40	LD	(409AH), A	SAVE ERR.
19BA	2AE640	LD	HL, (40E6H)	GET CURRENT LINE #.
19BD	22EE40	LD	(40EEH), HL	
19C0	EB	EX	DE,HL	
19C1	2AEA40	LD	HL, (40EAH)	GET ERL
19C4	7C	LD	A,H	DIRECT
19C5	A5	AND	L	MODE?
19C6	3C	INC	A	
19C7	2807	JR	Z,19D0H	SKIP IF SO
19C9	22F540	LD	(40F5H), HL	SAVE LAST LINE
19CC	EB	EX	DE,HL	& LAST BYTE EXECUTED
19CD	22F740	LD	(40F7H), HL	
19D0	2AF040	LD	HL, (40F0H)	GET ERROR VECTOR
19D3	7C	LD	A,H	IS IT 0?
19D4	B5	OR	L	
19D5	EB	EX	DE,HL	SAVE VECTOR
19D6	21F240	LD	HL,40F2H	POINT TO TRAP FLAG
19D9	2808	JR	Z,19E3H	SKIP IF ON ERROR GOTO 0
19DB	A6	AND	(HL)	TEST TRAP FLAG
19DC	2005	JR	NZ,19E3H	SKIP IF ALREADY TRAPPING
19DE	35	DEC	(HL)	FLAG IS IN TRAP

19DF	EB	EX	DE,HL	PUT VECTOR IN HL
19E0	C3361D	JP	1D36H	ERROR GO TO
19E3	AF	XOR	A	CLEAR TRAP FLAG
19E4	77	LD	(HL),A	
19E5	59	LD	E,C	E = ERROR CODE
19E6	CD920	CALL	20F9H	PRINT
19E9	21C918	LD	HL,18C9H	HL => ERROR TABLE
19EC	CDA641	CALL	41A6H	DOS LINK
19EF	57	LD	D,A	DE = ERROR CODE
19F0	3E3F	LD	A,3FH	"?"
19F2	CD2A03	CALL	032AH	OUTPUT "?"
19F5	19	ADD	HL,DE	FIND ERROR MESSAGE
19F6	7E	LD	A,(HL)	GET FIRST BYTE
19F7	CD2A03	CALL	032AH	DISPLAY IT
19FA	D7	RST	10H	GET SECOND BYTE
19FB	CD2A03	CALL	032AH	DISPLAY IT
19FE	211D19	LD	HL,191DH	POINT TO "ERROR"
1A01	E5	PUSH	HL	SAVE POINTER
1A02	2AEA40	LD	HL,(40EAH)	GET ERROR LINE #
1A05	E3	EX	(SP),HL	PUT IN STACK & RETRIEVE POINTER
1A06	CDA728	CALL	28A7H	PRINT " ERROR";
1A09	E1	POP	HL	RESTORE ERL
1A0A	11FEFF	LD	DE,0FFFEH	IS ERL = FFFEH?
1A0D	DF	RST	18H	
1A0E	CA7406	JP	Z,0674H	RE-BOOT IF SO
1A11	7C	LD	A,H	IS THIS DIRECT MODE?
1A12	A5	AND	L	
1A13	3C	INC	A	
1A14	C4A70F	CALL	NZ,0FA7H	PRINT LINE # IF NOT
1A17	3E			MASK TO 1A19H
1A18	C1	POP	BC	CLEAR THE STACK
				READY
1A19	CD8B03	CALL	038BH	RESET I/O
1A1C	CDAC41	CALL	41ACH	LINK TO DOS
1A1F	CD801	CALL	01F8H	TURN OFF TAPE
1A22	CD920	CALL	20F9H	NEW LINE
1A25	212919	LD	HL,1929H	PRINT
1A28	CDA728	CALL	28A7H	"READY"
1A2B	3A9A40	LD	A,(409AH)	SYNTAX
1A2E	D602	SUB	02H	ERROR?
1A30	CC532E	CALL	Z,2E53H	EDIT IF SO
1A33	21FFFF	LD	HL,0FFFFH	SET
1A36	22A240	LD	(40A2H),HL	DIRECT MODE
1A39	3AE140	LD	A,(40E1H)	AUTO?
1A3C	B7	OR	A	
1A3D	2837	JR	Z,1A76H	SKIP IF NOT
1A3F	2AE240	LD	HL,(40E2H)	GET CURRENT LINE #
1A42	E5	PUSH	HL	SAVE CURRENT LINE #
1A43	CDAF0F	CALL	0FAFH	DISPLAY LINE #
1A46	D1	POP	DE	DE = CURRENT LINE #
1A47	D5	PUSH	DE	
1A48	CD2C1B	CALL	1B2CH	SEARCH FOR LINE
1A4B	3E2A	LD	A,2AH	DISPLAY "*"
1A4D	3802	JR	C,1A51H	IF ALREADY USED
1A4F	3E20	LD	A,20H	SPACE IF NOT
1A51	CD2A03	CALL	032AH	OUTPUT

1A54	CD6103	CALL	0361H	INS LINE
1A57	D1	POP	DE	RESTORE CURRENT LINE #
1A58	3006	JR	NC,1A60H	SKIP IF NO <BREAK>
1A5A	AF	XOR	A	TURN OFF
1A5B	32E140	LD	(40E1H),A	AUTO
1A5E	18B9	JR	1A19H	GO TO READY
1A60	2AE440	LD	HL,(40E4H)	GET AUTO INCREMENT
1A63	19	ADD	HL,DE	COMPUTE NEXT LINE #
1A64	38F4	JR	C,1A5AH	STOP IF TOO LARGE
1A66	D5	PUSH	DE	SAVE PREVIOUS LINE #
1A67	11F9FF	LD	DE,0FFF9H	IS NEW LINE #
1A6A	DF	RST	18H	GREATER OR EQUAL TO 65529?
1A6B	D1	POP	DE	RARE PREVIOUS LINE #
1A6C	30EC	JR	NC,1A5AH	STOP IF SO
1A6E	22E240	LD	(40E2H),HL	SAVE NEW CURRENT LINE #
1A71	F6FF	OR	0FFH	SET FLAG
1A73	C3EB2F	JP	2FEBH	SAVE LINE & RETURN
1A76	3E3E	LD	A,3EH	DISPLAY ">"
1A78	CD2A03	CALL	032AH	
1A7B	CD6103	CALL	0361H	INPUT LINE
1A7E	DA331A	JP	C,1A33H	RE-START IF <BREAK>.
1A81	D7	RST	10H	GET CHARACTER
1A82	3C	INC	A	SET STATUS
1A83	3D	DEC	A	
1A84	CA331A	JP	Z,1A33H	RE-START IF NULL
1A87	F5	PUSH	AF	SAVE CHARACTER
1A88	CD5A1E	CALL	1E5AH	PROCESS LINE #, IF ANY
1A8B	2B	DEC	HL	BACKUP
1A8C	7E	LD	A,(HL)	OVERTRAILING
1A8D	FE20	CP	20H	BLANKS AFTER LINE #
1A8F	28FA	JR	Z,1A8BH	IF ANY
1A91	23	INC	HL	POINT TO FIRST CHARACTER
1A92	7E	LD	A,(HL)	IGNORE
1A93	FE20	CP	20H	EXACTLY
1A95	CCC909	CALL	Z,09C9H	ONE BLANK
1A98	D5	PUSH	DE	SAVE LINE #
1A99	CDC01B	CALL	1BC0H	ENCODE LINE
1A9C	D1	POP	DE	RESTORE LINE #
1A9D	F1	POP	AF	RESTORE STATUS OF 1ST CHARACTER
1A9E	22E640	LD	(40E6H),HL	SAVE LINE BUFFER ADDRESS
1AA1	CDB241	CALL	41B2H	LINK TO DOS
1AA4	D25A1D	JP	NC,1D5AH	EXECUTE IF DIRECT STATEMENT
1AA7	D5	PUSH	DE	SAVE LINE #
1AA8	C5	PUSH	BC	SAVE LINE LENGTH
1AA9	AF	XOR	A	ZERO
1AAA	32DD40	LD	(40DDH),A	FLAG
1AAD	D7	RST	10H	SCAN LINE FOR FIRST CHARACTER.
1AAE	B7	OR	A	CHECK CHARACTER
1AAF	F5	PUSH	AF	SAVE STATUS
1AB0	EB	EX	DE,HL	ERL = CURRENT LINE
1AB1	22EC40	LD	(40ECH),HL	
1AB4	EB	EX	DE,HL	
1AB5	CD2C1B	CALL	1B2CH	SEARCH FOR LINE
1AB8	C5	PUSH	BC	SAVE POSITION
1AB9	DCE42B	CALL	C,2BE4H	DELETE LINE IF FOUND
1ABC	D1	POP	DE	GET
1ABD	F1	POP	AF	STATUS

1ABE D5	PUSH	DE	OF FIRST CHARACTER.
1ABF 2827	JR	Z,1AE8H	SKIP IF NULL
1AC1 D1	POP	DE	DE = POSITION OF NEW LINE
1AC2 2AF940	LD	HL,(40F9H)	BC = END OF PROGRAM
1AC5 E3	EX	(SP),HL	HL = LENGTH OF NEW LINE
1AC6 C1	POP	BC	
1AC7 09	ADD	HL,BC	HL = NEW END OF PROGRAM
1AC8 E5	PUSH	HL	SAVE NEW END OF PROGRAM
1AC9 CD5519	CALL	1955H	MOVE PROGRAM APART
1ACC E1	POP	HL	RESTORE NEW END OF PROGRAM
1ACD 22F940	LD	(40F9H),HL	SAVE NEW END OF PROGRAM
1AD0 EB	EX	DE,HL	HL => NEW LINE
1AD1 74	LD	(HL),H	SAVE MSB OF LINE POINTER
1AD2 D1	POP	DE	DE = LINE NUMBER
1AD3 E5	PUSH	HL	SAVE START OF LINE
1AD4 23	INC	HL	POINT TO STORAGE
1AD5 23	INC	HL	FOR LINE #
1AD6 73	LD	(HL),E	SAVE LINE #
1AD7 23	INC	HL	
1AD8 72	LD	(HL),D	
1AD9 23	INC	HL	
1ADA EB	EX	DE,HL	DE => INPUT BUFFER
1ADB 2AA740	LD	HL,(40A7H)	
1ADE EB	EX	DE,HL	
1ADF 1B	DEC	DE	
1AE0 1B	DEC	DE	
1AE1 1A	LD	A,(DE)	MOVE DATA UNTIL
1AE2 77	LD	(HL),A	END OF LINE
1AE3 23	INC	HL	
1AE4 13	INC	DE	
1AE5 B7	OR	A	
1AE6 20F9	JR	NZ,1AE1H	
1AE8 D1	POP	DE	RESTORE START OF LINE
1AE9 CDFC1A	CALL	1AFCH	UPDATE LINE POINTERS
1AEC CDB541	CALL	41B5H	LINK TO DOS
1AEF CD5D1B	CALL	1B5DH	FIX OTHER POINTERS
1AF2 CDB841	CALL	41B8H	LINK TO DOS
1AF5 C3331A	JP	1A33H	LOOP FOR MORE
1AF8 2AA440	LD	HL,(40A4H)	DE = START OF PROGRAM
1AFB EB	EX	DE,HL	
1AFC 62	LD	H,D	HL = POSITION TO
1AFD 6B	LD	L,E	START SCAN
1AFE 7E	LD	A,(HL)	FORWARD POINTER
1AFF 23	INC	HL	ZERO?
1B00 B6	OR	(HL)	
1B01 C8	RET	Z	RETURN IF SO
1B02 23	INC	HL	MOVE PAST LINE #
1B03 23	INC	HL	TO TEXT
1B04 23	INC	HL	
1B05 AF	XOR	A	CLEAR STATUS
1B06 BE	CP	(HL)	SCAN PAST TEXT
1B07 23	INC	HL	IN LINE
1B08 20FC	JR	NZ,1B06H	
1B0A EB	EX	DE,HL	STORE NEW FORWARD
1B0B 73	LD	(HL),E	POINTER
1B0C 23	INC	HL	DE = START OF NEW LINE
1B0D 72	LD	(HL),D	

1B0E 18EC	JR	1AFCH	LOOP OVER REST OF PROGRAM
1B10 110000	LD	DE,0000H	DEFAULT IT START LINE #0
1B13 D5	PUSH	DE	SAVE DEFAULT START
1B14 2809	JR	Z,1B1FH	SKIP IF NO LINE #'S GIVEN
1B16 D1	POP	DE	CLEAR STACK
1B17 CD4F1E	CALL	1E4FH	GET FIRST LINE #
1B1A D5	PUSH	DE	SAVE START LINE #
1B1B 280B	JR	Z,1B28H	SKIP IF NOTHING ELSE GIVEN
1B1D CF	RST	08H	SYNTAX ERROR IF
1B1E CE	"-"		NEXT CHARACTER. ISN'T "-".
1B1F 11	LD	BC,0FFFAH	DEFAULT ENDING LINE #
1B22 C44F1E	CALL	NZ,1E4FH	GET SECOND LINE #
1B25 C29719	JP	NZ,1997H	SYNTAX ERROR IF MORE JUNK
1B28 EB	EX	DE,HL	DE = START LINE
1B29 D1	POP	DE	HL = END LINE
1B2A E3	EX	(SP),HL	SAVE ENDING LINE
1B2B E5	PUSH	HL	& RETURN ADDRESS
1B2C 2AA440	LD	HL,(40A4H)	GET START OF PROGRAM
1B2F 44	LD	B,H	SAVE POSITION IN BC
1B30 4D	LD	C,L	
1B31 7E	LD	A,(HL)	IS FORWARD POINTER
1B32 23	INC	HL	NULL ? (END OF
1B33 B6	OR	(HL)	PROGRAM?)
1B34 2B	DEC	HL	
1B35 C8	RET	Z	RETURN IF SO, Z,NC
1B36 23	INC	HL	MOVE TO
1B37 23	INC	HL	LINE #
1B38 7E	LD	A,(HL)	GET
1B39 23	INC	HL	LINE
1B3A 66	LD	H,(HL)	#
1B3B 6F	LD	L,A	
1B3C DF	RST	18H	LINE # CORRECT?
1B3D 60	LD	H,B	RESTORE POSITION
1B3E 69	LD	L,C	FROM BC
1B3F 7E	LD	A,(HL)	GET
1B40 23	INC	HL	FORWARD
1B41 66	LD	H,(HL)	POINTER
1B42 6F	LD	L,A	
1B43 3F	CCF		EXIT Z,C
1B44 C8	RET	Z	IF LINE FOUND
1B45 3F	CCF		EXIT NZ,NC
1B46 D0	RET	NC	IF HIGHER LINE FOUND
1B47 18E6	JR	1B2FH	CONTINUE SEARCHING
1B49 C0	RET	NZ	SYNTAX ERROR IF MORE IN COMMAND
1B4A CDC901	CALL	01C9H	CLEAR SCREEN
1B4D 2AA440	LD	HL,(40A4H)	POINT TO START OF PROGRAM
1B50 CDF81D	CALL	1DF8H	TROFF
1B53 32E140	LD	(40E1H),A	TURN AUTO OFF
1B56 77	LD	(HL),A	MARK
1B57 23	INC	HL	END
1B58 77	LD	(HL),A	OF
1B59 23	INC	HL	PROGRAM
1B5A 22F940	LD	(40F9H),HL	SAVE START OF VARIABLE TABLE
1B5D CD6B04	CALL	046BH	UNPROTECT SCREEN
1B60 2B	DEC	HL	SET EXECUTION
1B61 22DF40	LD	(40DFH),HL	UNPROTECT SCREEN
1B64 061A	LD	B,1AH	SET

1B66	210141	LD	HL,4101H	VARIABLE
1B69	3604	LD	(HL),04H	TYPES
1B6B	23	INC	HL	TO
1B6C	10FB	DJNZ	1B69H	S.P.
1B6E	AF	XOR	A	
1B6F	32F240	LD	(40F2H),A	CLEAR ERROR TRAP FLAG
1B72	6F	LD	L,A	
1B73	67	LD	H,A	
1B74	22F040	LD	(40F0H),HL	ON ERROR GO TO 0
1B77	22F740	LD	(40F7H),HL	LAST BYTE EXECUTED = 0
1B7A	2AB140	LD	HL,(40B1H)	GET MEMORY SIZE
1B7D	22D640	LD	(40D6H),HL	SET NEXT AVAILABLE STRING
1B80	CD911D	CALL	1D91H	RESTORE
1B83	2AF940	LD	HL,(40F9H)	GET END OF PROGRAM
1B86	22FB40	LD	(40FBH),HL	SET ARRAY POINTER
1B89	22FD40	LD	(40FDH),HL	SET FREE SPACE POINTER
1B8C	CDBB41	CALL	41BBH	LINK TO DOS
1B8F	C1	POP	BC	GET RETURN ADDRESS
1B90	2AA040	LD	HL,(40A0H)	GET TOP OF FREE MEMORY
1B93	2B	DEC	HL	SET POINTER
1B94	2B	DEC	HL	TO STACK
1B95	22E840	LD	(40E8H),HL	AT START OF STATEMENT
1B98	23	INC	HL	SET NEW
1B99	23	INC	HL	STACK
1B9A	F9	LD	SP,HL	POINTER
1B9B	21B540	LD	HL,40B5H	CLEAR
1B9E	22B340	LD	(40B3H),HL	WORKSPACE
1BA1	CD8B03	CALL	038BH	RESET I/O
1BA4	CD6921	CALL	2169H	RESET I/O
1BA7	AF	XOR	A	CLEAR A & HL
1BA8	67	LD	H,A	
1BA9	6F	LD	L,A	
1BAA	32DC40	LD	(40DCH),A	NO ACTIVE FORS
1BAD	E5	PUSH	HL	MARK TOP OF STACK
1BAE	C5	PUSH	BC	SAVE RETURN ADDRESS
1BAF	2ADF40	LD	HL,(40DFH)	GET EXECUTION ADDRESS
1BB2	C9	RET		DONE

PRINT "? " & INPUT

1BB3	3E3F	LD	A,3FH	"? "
1BB5	CD2A03	CALL	032AH	DISPLAY
1BB8	3E20	LD	A,20H	" "
1BBA	CD2A03	CALL	032AH	DISPLAY
1BBD	C36103	JP	0361H	INPUT TEXT AND RETURN

ENCODE PROGRAM LINE

1BC0 AF	XOR	A	CLEAR
1BC1 32B040	LD	(40B0H),A	DATA FLAG &
1BC4 4F	LD	C,A	DESTINATION BUFFER SIZE
1BC5 EB	EX	DE,HL	HL = SOURCE BUFFER
1BC6 2AA740	LD	HL,(40A7H)	DE => DESTINATION BUFFER
1BC9 2B	DEC	HL	(2 BYTES BEFORE
1BCA 2B	DEC	HL	SOURCE BYTES)
1BCB EB	EX	DE,HL	
1BCC 7E	LD	A,(HL)	GET SOURCE CHARACTER.
1BCD FE20	CP	20H	SPACE?
1BCF CA5B1C	JP	Z,1C5BH	JUMP IF SO
1BD2 47	LD	B,A	STORE POSSIBLE DELIMITER
1BD3 FE22	CP	22H	DOUBLE QUOTES?
1BD5 CA771C	JP	Z,1C77H	JUMP IF SO?
1BD8 B7	OR	A	END OF LINE
1BD9 CA7D1C	JP	Z,1C7DH	FINISH UP IF SO
1BDC 3AB040	LD	A,(40B0H)	DATA STATEMENT?
1BDF B7	OR	A	
1BE0 7E	LD	A,(HL)	GET CHARACTER.
1BE1 C25B1C	JP	NZ,1C5BH	JUMP IF SO
1BE4 FE3F	CP	3FH	QUESTION MARK? (PRINT?)
1BE6 3EB2	LD	A,0B2H	TOKEN 82 IF SO
1BE8 CA5B1C	JP	Z,1C5BH	AND JUMP IF SO
1BEB 7E	LD	A,(HL)	GET CHARACTER
1BEC FE30	CP	30H	IS CHARACTER PART
1BEE 3805	JR	C,1BF5H	OF A NUMBER?
1BF0 FE3C	CP	3CH	SKIP OUT IF SO
1BF2 DA5B1C	JP	C,1C5BH	
1BF5 D5	PUSH	DE	SAVE DE
1BF6 114F16	LD	DE,164FH	DE = RESERVED WORD TABLE
1BF9 C5	PUSH	BC	SAVE BC
1BFA 013D1C	LD	BC,1C3DH	RETURN TO 1C3DH
1BFD C5	PUSH	BC	
1BFE 067F	LD	B,7FH	INITIAL TOKEN VALUE
1C00 7E	LD	A,(HL)	GET CHARACTER.
1C01 FE61	CP	61H	FOLD
1C03 3807	JR	C,1C0CH	INTO
1C05 FE7B	CP	7BH	UPPER
1C07 3003	JR	NC,1C0CH	CASE
1C09 E65F	AND	5FH	ONLY
1C0B 77	LD	(HL),A	
1C0C 4E	LD	C,(HL)	GET CHARACTER
1C0D EB	EX	DE,HL	HL => RESERVED WORD TABLE
1C0E 23	INC	HL	SCAN FOR
1C0F B6	OR	(HL)	BYTE WITH
1C10 F20E1C	JP	P,1C0EH	BIT 7 SET
1C13 04	INC	B	INCREASE TOKEN VALUE
1C14 7E	LD	A,(HL)	GET CHARACTER FROM WORD
1C15 E67F	AND	7FH	MASK OFF BIT 7
1C17 C8	RET	Z	RETURN IF END OF TABLE
1C18 B9	CP	C	GIVEN CHARACTER MATCH WORD
1C19 20F3	JR	NZ,1C0EH	LOOP TO NEXT WORD IF NOT
1C1B EB	EX	DE,HL	RESTORE POINTERS AS BEFORE
1C1C E5	PUSH	HL	SAVE POSITION IN TEXT
1C1D 13	INC	DE	NEXT CHARACTER IN WORD
1C1E 1A	LD	A,(DE)	GET CHARACTER IN WORD

1C1F B7	OR	A	TEST IT
1C20 FA391C	JP	M,1C39H	SKIP OUT IF NEW WORD
1C23 4F	LD	C,A	SAVE CHARACTER FROM WORD
1C24 78	LD	A,B	GET TOKEN VALUE
1C25 FE8D	CP	8DH	GO TO?
1C27 2002	JR	NZ,1C2BH	SKIP IF NOT
1C29 D7	RST	10H	PASS OVER
1C2A 2B	DEC	HL	EMBEDDED SPACES
1C2B 23	INC	HL	NEXT CHARACTER IN TEXT
1C2C 7E	LD	A, (HL)	GET CHARACTER
1C2D FE61	CP	61H	FOLD
1C2F 3802	JR	C,1C33H	TO
1C31 E65F	AND	5FH	UPPER-CASE
1C33 B9	CP	C	SAME AS CHARACTER IN WORD?
1C34 28E7	JR	Z,1C1DH	LOOP IF SO
1C36 E1	POP	HL	RESTORE POSITION TEXT
1C37 18D3	JR	1C0CH	TRY AGAIN
1C39 48	LD	C,B	PUT TOKEN IN C
1C3A F1	POP	AF	CLEAR STACK
1C3B EB	EX	DE,HL	HL => DESTINATION
1C3C C9	RET		CONTINUE AT 1C3D.
1C3D EB	EX	DE,HL	HL => SOURCE
1C3E 79	LD	A,C	GET TOKEN
1C3F C1	POP	BC	RESTORE BC
1C40 D1	POP	DE	RESTORE DESTINATION
1C41 EB	EX	DE,HL	HL => DESTINATION
1C42 FE95	CP	95H	ELSE?
1C44 363A	LD	(HL),3AH	INSERT A COLON IF SO
1C46 2002	JR	NZ,1C4AH	SKIP IF NOT
1C48 0C	INC	C	DESTINATION SIZE UP
1C49 23	INC	HL	DESTINATION FORWARD
1C4A FEFB	CP	0FBH	REM ABBREVIATOR?
1C4C 200C	JR	NZ,1C5AH	SKIP IF NOT
1C4E 363A	LD	(HL),3AH	INSERT A COLON
1C50 23	INC	HL	MOVE DESTINATION FORWARD
1C51 0693	LD	B,93H	STORE REM TOKEN
1C53 70	LD	(HL),B	
1C54 23	INC	HL	MOVE DESTINATION FORWARD
1C55 EB	EX	DE,HL	HL => SOURCE
1C56 0C	INC	C	INCREASE DESTINATION
1C57 0C	INC	C	SIZE BY 2
1C58 181D	JR	1C77H	CONTINUE
1C5A EB	EX	DE,HL	HL => SOURCE
1C5B 23	INC	HL	NEXT SOURCE BYTE
1C5C 12	LD	(DE),A	STORE TOKEN OR CHARACTER
1C5D 13	INC	DE	NEXT DESTINATION BYTE
1C5E 0C	INC	C	INCREASE DESTINATION SIZE
1C5F D63A	SUB	3AH	WAS CHARACTER A COLON?
1C61 2804	JR	Z,1C67H	SKIP IF SO
1C63 FE4E	CP	4EH	DATA TOKEN?
1C65 2003	JR	NZ,1C6AH	SKIP IF NOT
1C67 32B040	LD	(40B0H),A	FLAG AS DATA
1C6A D659	SUB	59H	REM?
1C6C C2CC1B	JP	NZ,1BCCH	LOOP IF NOT
1C6F 47	LD	B,A	B = 0
1C70 7E	LD	A, (HL)	GET SOURCE CHARACTER.
1C71 B7	OR	A	NULL? (END OF LINE?)

1C72	2809	JR	Z,1C7DH	SKIP OUT IF SO.
1C74	B8	CP	B	DELIMITER FOUND?
1C75	28E4	JR	Z,1C5BH	REJOIN MAIN LOOP IF SO
1C77	23	INC	HL	NEXT SOURCE BYTE.
1C78	12	LD	(DE),A	SAVE BYTE IN DESTINATION
1C79	0C	INC	C	INCREASE DESTINATION SIZE
1C7A	13	INC	DE	NEXT DESTINATION BYTE
1C7B	18F3	JR	1C70H	LOOP UNTIL DONE
1C7D	210500	LD	HL,0005H	HL = 5
1C80	44	LD	B,H	BC = DESTINATION BUFFER SIZE
1C81	09	ADD	HL,BC	HL = TOTAL BUFFER SIZE
1C82	44	LD	B,H	BC = TOTAL BUFFER SIZE
1C83	4D	LD	C,L	
1C84	2AA740	LD	HL,(40A7H)	GET OLD START OF BUFFER
1C87	2B	DEC	HL	BACK UP 3 BYTES
1C88	2B	DEC	HL	
1C89	2B	DEC	HL	
1C8A	12	LD	(DE),A	ZERO 3
1C8B	13	INC	DE	BYTES AFTER
1C8C	12	LD	(DE),A	DESTINATION
1C8D	13	INC	DE	BUFFER
1C8E	12	LD	(DE),A	
1C8F	C9	RET		DONE

(CP HL,DE (RST 18H)

1C90	7C	LD	A,H	COMPARE MSBS
1C91	92	SUB	D	
1C92	C0	RET	NZ	RETURN IF NOT EQUAL
1C93	7D	LD	A,L	COMPARE LSB
1C94	93	SUB	E	
1C95	C9	RET		RETURN WITH STATUS

RST 08 ROUTINE

1C96	7E	LD	A,(HL)	COMPARE (HL)
1C97	E3	EX	(SP),HL	AND (SP)
1C98	BE	CP	(HL)	
1C99	23	INC	HL	RETURN TO FOLLOWING BYTE
1C9A	E3	EX	(SP),HL	FIX STACK, RESTORE HL
1C9B	CA781D	JP	Z,1D78H	RST 10H & RETURN IF MATCH
1C9E	C39719	JP	1997H	SN ERROR 2

FOR

1CA1	3E64	LD	A,64H	FLAG ACTIVE FOR
1CA3	32DC40	LD	(40DCH),A	
1CA6	CD211F	CALL	1F21H	EVALUATE LOWER LIMIT EXPRESSION
1CA9	E3	EX	(SP),HL	SAVE POSITION
1CAA	CD3619	CALL	1936H	SEARCH FOR MATCHING COUNTER
1CAD	D1	POP	DE	RESTORE POSITION TO DE
1CAE	2005	JR	NZ,1CB5H	SKIP OF COUNTER NOT FOUND
1CB0	09	ADD	HL,BC	POSITION
1CB1	F9	LD	SP,HL	STACK POINTER
1CB2	22E840	LD	(40E8H),HL	SAVE STACK POSITION
1CB5	EB	EX	DE,HL	HL => TEXT POSITION
1CB6	0E08	LD	C,08H	REQUEST 8 BYTES
1CB8	CD6319	CALL	1963H	CHECK IF ENOUGH SPACE
1CBB	E5	PUSH	HL	SAVE POSITION
1CBC	CD051F	CALL	1F05H	GET CURRENT LINE

1CBF	E3	EX	(SP),HL	SAVE LINE & RESTORE POSITION
1CC0	E5	PUSH	HL	
1CC1	2AA240	LD	HL,(40A2H)	
1CC4	E3	EX	(SP),HL	
1CC5	CF	RST	08H	SYNTAX ERROR IF
1CC6	BD			NEXT CHARACTER NOT "TO" TOKEN
1CC7	E7	RST	20H	CHECK TYPE OF COUNTER
1CC8	CAF60A	JP	Z,0AF6H	TM ERROR IF STRING
1CCB	D2F60A	JP	NC,0AF6H	TM ERROR IF D.P.
1CCE	F5	PUSH	AF	SAVE TYPE
1CCF	CD3723	CALL	2337H	EVALUATE UPPER LIMIT
1CD2	F1	POP	AF	RESTORE TYPE
1CD3	E5	PUSH	HL	SAVE POSITION
1CD4	F2EC1C	JP	P,1CECH	JUMP IF S.P.
1CD7	CD7F0A	CALL	0A7FH	MAKE UPPER LIMIT INTEGER
1CDA	E3	EX	(SP),HL	SAVE LIMIT & GET POSITION
1CDB	110100	LD	DE,0001H	DEFAULT STEP = 1
1CDE	7E	LD	A,(HL)	GET NEXT TOKEN
1CDF	FECC	CP	0CCH	STEP?
1CE1	CC012B	CALL	Z,2B01H	GET STEP IF SO
1CE4	D5	PUSH	DE	SAVE STEP
1CE5	E5	PUSH	HL	SAVE POSITION
1CE6	EB	EX	DE,HL	CHECK SIGN
1CE7	CD9E09	CALL	099EH	OF STEP
1CEA	1822	JR	1D0EH	CONTINUE
1CEC	CDB10A	CALL	0AB1H	CONVERT UPPER LIMIT TO
1CEF	CDBF09	CALL	09BFH	PUT IT IN BCDE
1CF2	E1	POP	HL	RESTORE POSITION
1CF3	C5	PUSH	BC	SAVE UPPER LIMIT
1CF4	D5	PUSH	DE	
1CF5	010081	LD	BC,8100H	DEFAULT STEP 1.0
1CF8	51	LD	D,C	
1CF9	5A	LD	E,D	
1CFA	7E	LD	A,(HL)	GET NEXT TOKEN
1CFB	FECC	CP	0CCH	STEP?
1CFD	3E01	LD	A,01H	DEFAULT SIGN = +1
1CFF	200E	JR	NZ,1D0FH	SKIP IF NO STEP
1D01	CD3823	CALL	2338H	EVALUATE STEP EXPRESSION
1D04	E5	PUSH	HL	SAVE POSITION
1D05	CDB10A	CALL	0AB1H	CONVERT STEP TO S.P.
1D08	CDBF09	CALL	09BFH	PUT IT IN BCDE
1D0B	CD5509	CALL	0955H	GET SIGN OF STEP
1D0E	E1	POP	HL	RESTORE POSITION
1D0F	C5	PUSH	BC	SAVE STEP
1D10	D5	PUSH	DE	VALUE
1D11	4F	LD	C,A	SAVE SIGN IN C
1D12	E7	RST	20H	CHECK TYPE
1D13	47	LD	B,A	SAVE TYPE IN B
1D14	C5	PUSH	BC	SAVE TYPE & SIGN OF STEP
1D15	E5	PUSH	HL	SAVE POSITION
1D16	2ADF40	LD	HL,(40DFH)	GET COUNTER POINTER
1D19	E3	EX	(SP),HL	SAVE POINTER & GET POSITION
1D1A	0681	LD	B,81H	FOR FLAG
1D1C	C5	PUSH	BC	SAVE FLAG
1D1D	33	INC	SP	TRIM EXTRA BYTE

MAIN PROGRAM EXECUTOR

1D1E	CD5803	CALL	0358H	CHECK KEYBOARD
1D21	B7	OR	A	ANY KEY HIT?
1D22	C4A01D	CALL	NZ,1DA0H	HANDLE SHIFT @ OR <BREAK>
1D25	22E640	LD	(40E6H),HL	SAVE TEXT POSITION
1D28	ED73E840	LD	(40E8H),SP	SAVE STACK POSITION
1D2C	7E	LD	A,(HL)	GET TOKEN
1D2D	FE3A	CP	3AH	COLON?
1D2F	2829	JR	Z,1D5AH	EXECUTE STATEMENT IF SO
1D31	B7	OR	A	END OF LINE?
1D32	C29719	JP	NZ,1997H	SYNTAX ERROR IF NOT
1D35	23	INC	HL	END
1D36	7E	LD	A,(HL)	OF
1D37	23	INC	HL	PROGRAM?
1D38	B6	OR	(HL)	
1D39	CA7E19	JP	Z,197EH	END EXECUTION IF SO
1D3C	23	INC	HL	GET
1D3D	5E	LD	E,(HL)	LINE
1D3E	23	INC	HL	NUMBER
1D3F	56	LD	D,(HL)	INTO
1D40	EB	EX	DE,HL	HL
1D41	22A240	LD	(40A2H),HL	SAVE CURRENT LINE #
1D44	3A1B41	LD	A,(411BH)	TRACE
1D47	B7	OR	A	ON?
1D48	280F	JR	Z,1D59H	SKIP IF NOT
1D4A	D5	PUSH	DE	SAVE POSITION
1D4B	3E3C	LD	A,3CH	DISPLAY
1D4D	CD2A03	CALL	032AH	"<"
1D50	CDAF0F	CALL	0FAFH	DISPLAY LINE #
1D53	3E3E	LD	A,3EH	DISPLAY
1D55	CD2A03	CALL	032AH	">"
1D58	D1	POP	DE	RESTORE POSITION
1D59	EB	EX	DE,HL	HL => POSITION IN TEXT
1D5A	D7	RST	10H	GET NEXT CHARACTER
1D5B	111E1D	LD	DE,1D1EH	RETURN TO
1D5E	D5	PUSH	DE	EXECUTOR WHEN DONE
1D5F	C8	RET	Z	NEW LINE IF END-OF-LINE
1D60	D680	SUB	80H	ADJUST VALUE OF TOKEN
1D62	DA211F	JP	C,1F21H	EVALUATE IF AN EXPRESSION
1D65	FE3C	CP	3CH	TOKEN IN STATEMENT TABLE?
1D67	D2E72A	JP	NC,2AE7H	JUMP IF NOT
1D6A	07	RLCA		BC
1D6B	4F	LD	C,A	=
1D6C	0600	LD	B,00H	TOKEN *2
1D6E	EB	EX	DE,HL	SAVE POSITION
1D6F	212218	LD	HL,1822H	HL = FUNCTION
1D72	09	ADD	HL,BC	ADDRESS
1D73	4E	LD	C,(HL)	GET
1D74	23	INC	HL	ADDRESS
1D75	46	LD	B,(HL)	OF ROUTINE
1D76	C5	PUSH	BC	SAVE AS RETURN ADDRESS
1D77	EB	EX	DE,HL	RESTORE POSITION

RST 10H ROUTINE

1D78	23	INC	HL	MOVE TO NEXT CHARACTER
1D79	7E	LD	A,(HL)	GET CHARACTER

1D7A	FE3A	CP	3AH	NUMERIC?
1D7C	D0	RET	NC	RETURN IF NOT
1D7D	FE20	CP	20H	SPACE?
1D7F	CA781D	JP	Z,1D78H	SKIP OVER SPACES
1D82	FE0B	CP	0BH	IGNORE
1D84	3005	JR	NC,1D8BH	LINE FEEDS
1D86	FE09	CP	09H	AND TABS
1D88	D2781D	JP	NC,1D78H	SKIP IF FOUND
1D8B	FE30	CP	30H	SET CARRY IF NUMERIC
1D8D	3F	CCF		
1D8E	3C	INC	A	SET SIGN
1D8F	3D	DEC	A	NEGATIVE = TOKEN
1D90	C9	RET		DONE

RESTORE

1D91	EB	EX	DE,HL	SAVE POSITION
1D92	2AA440	LD	HL,(40A4H)	POSITION DATA
1D95	2B	DEC	HL	POINTER TO JUST
1D96	22FF40	LD	(40FFH),HL	BEFORE PROGRAM
1D99	EB	EX	DE,HL	RESTORE POSITION
1D9A	C9	RET		DONE

PAUSE KEY HANDLER

1D9B	CD5803	CALL	0358H	SCAN KEYBOARD
1D9E	B7	OR	A	ANY KEY HIT?
1D9F	C8	RET	Z	RETURN IF NOT
1DA0	FE60	CP	60H	SHIFT @?
1DA2	CC8403	CALL	Z,0384H	WAIT FOR KEYBOARD IF SO
1DA5	329940	LD	(4099H),A	SAVE INKEY\$
1DA8	3D	DEC	A	BREAK? RETURN IF NOT

STOP

1DA9	C0	RET	NZ	RETURN IF MORE JUNK ON LINE
1DAA	3C	INC	A	FLAG STOP
1DAB	C3B41D	JP	1DB4H	CONTINUE

END

1DAE	C0	RET	NZ	RETURN IF MORE JUNK ON LINE
1DAF	F5	PUSH	AF	FLAG END. SAVE FLAG
1DB0	CCBB41	CALL	Z,41BBH	DOS LINK
1DB3	F1	POP	AF	RESTORE FLAG
1DB4	22E640	LD	(40E6H),HL	SAVE POSITION
1DB7	21B540	LD	HL,40B5H	CLEAR
1DBA	22B340	LD	(40B3H),HL	WORKSPACE
1DBD	21			MASK TO 1DC0H
1DBE	F6FF	OR	0FFH	FLAG STOP
1DC0	C1	POP	BC	CLEAR STACK
1DC1	2AA240	LD	HL,(40A2H)	GET CURRENT LINE #
1DC4	E5	PUSH	HL	SAVE CURRENT LINE #
1DC5	F5	PUSH	AF	SAVE FLAG
1DC6	7D	LD	A,L	DIRECT
1DC7	A4	AND	H	MODE?
1DC8	3C	INC	A	
1DC9	2809	JR	Z,1DD4H	SKIP IF SO
1DCB	22F540	LD	(40F5H),HL	SAVE CURRENT LINE #
1DCE	2AE640	LD	HL,(40E6H)	GET POSITION
1DD1	22F740	LD	(40F7H),HL	SAVE POSITION

1DD4	CD8B03	CALL	038BH	RESET VIDEO & PRINTER
1DD7	CDF920	CALL	20F9H	MOVE TO NEW LINE
1DDA	F1	POP	AF	RESTORE A
1DDB	213019	LD	HL,1930H	POINT TO "BREAK"
1DDE	C2061A	JP	NZ,1A06H	PRINT "BREAK IN NNN" & READY.
1DE1	C3181A	JP	1A18H	GO TO READY.

CONT

1DE4	2AF740	LD	HL, (40F7H)	GET LAST EXECUTED POSITION
1DE7	7C	LD	A,H	CAN WE
1DE8	B5	OR	L	CONTINUE?
1DE9	1E20	LD	E,20H	POSSIBLE CN ERROR
1DEB	CAA219	JP	Z,19A2H	ERROR IF WE CAN'T
1DEE	EB	EX	DE,HL	DE = LINE #
1DEF	2AF540	LD	HL, (40F5H)	HL = POSITION
1DF2	22A240	LD	(40A2H),HL	SAVE CURRENT LINE #
1DF5	EB	EX	DE,HL	
1DF6	C9	RET		GO TO IT!

TRON

1DF7	3EAF	LD	A,0AFH	SET TRACE FLAG & MASK TO 1DF
------	------	----	--------	------------------------------

TROFF

1DF8	AF	XOR	A	CLEAR TRACE FLAG
1DF9	321B41	LD	(411BH),A	STORE FLAG
1DFC	C9	RET		DONE
1DFD	F1	POP	AF	UNUSED
1DFE	E1	POP	HL	SUBROUTINE
1DFE	C9	RET		

DEFSTR

1E00	1E03	LD	E,03H	TYPE = STRING
1E02	01			MASK TO 1E0BH

DEFINT

1E03	1E02	LD	E,02H	TYPE = INTEGER
1E05	01			MASK TO 1E0BH

DEFSNG

1E06	1E04	LD	E,04H	TYPE = REAL (S.P.)
1E08	01			MASK TO 1E0BH

DEFDBL

1E09	1E08	LD	E,08H	TYPE = REAL (D.P)
------	------	----	-------	-------------------

SET DEFAULT IT VARIABLE TYPES

1E0B	CD3D1E	CALL	1E3DH	IS CHARACTER A-Z?
1E0E	019719	LD	BC,1997H	POSSIBLE SYNTAX ERROR
1E11	C5	PUSH	BC	
1E12	D8	RET	C	SYNTAX ERROR IF NOT
1E13	D641	SUB	41H	COMPUTE TABLE OFFSET
1E15	4F	LD	C,A	C = OFFSET (FIRST)
1E16	47	LD	B,A	B = OFFSET (SECOND)
1E17	D7	RST	10H	CHECK NEXT CHARACTER
1E18	FECE	CP	0CEH	
1E1A	2009	JR	NZ,1E25H	SKIP IF NOT
1E1C	D7	RST	10H	GET NEXT CHARACTER

1E1D	CD3D1E	CALL	1E3DH	IS CHARACTER A - Z
1E20	D8	RET	C	SYNTAX ERROR IF NOT
1E21	D641	SUB	41H	COMPUTE
1E23	47	LD	B,A	SECOND OFFSET
1E24	D7	RST	10H	SCAN FOR NEXT CHARACTER
1E25	78	LD	A,B	IS SECOND OFFSET
1E26	91	SUB	C	LESS THAN FIRST?
1E27	D8	RET	C	SYNTAX ERROR IF SO.
1E28	3C	INC	A	A = SECOND - FIRST + 1
1E29	E3	EX	(SP),HL	SAVE POSITION
1E2A	210141	LD	HL,4101H	COMPUTE
1E2D	0600	LD	B,00H	POSITION
1E2F	09	ADD	HL,BC	IN TABLE
1E30	73	LD	(HL),E	SAVE DEFAULT TYPE
1E31	23	INC	HL	NEXT POSITION
1E32	3D	DEC	A	ONE LESS TO DO
1E33	20FB	JR	NZ,1E30H	LOOP UNTIL DONE
1E35	E1	POP	HL	GET POSITION IN TEXT
1E36	7E	LD	A,(HL)	GET CHARACTER
1E37	FE2C	CP	2CH	", "?"
1E39	C0	RET	NZ	DONE IF NOT
1E3A	D7	RST	10H	GET NEXT CHARACTER
1E3B	18CE	JR	1E0BH	CONTINUE
1E3D	7E	LD	A,(HL)	IS CHARACTER < "A"?
1E3E	FE41	CP	41H	
1E40	D8	RET	C	RETURN ERROR IF SO
1E41	FE5B	CP	5BH	IS CHARACTER > "Z"?
1E43	3F	CCF		
1E44	C9	RET		RETURN ERROR IF SO

SUBSCRIPT EVALUATION

1E45	D7	RST	10H	GET NEXT CHARACTER
1E46	CD022B	CALL	2B02H	EVALUATE EXPRESSION
1E49	F0	RET	P	RETURN IF POSITIVE
1E4A	1E08	LD	E,08H	FC ERROR?
1E4C	C3A219	JP	19A2H	GO TO ERROR ROUTINE

DECODE LINE # REFERENCE

1E4F	7E	LD	A,(HL)	GET NEXT CHARACTER
1E50	FE2E	CP	2EH	PERIOD?
1E52	EB	EX	DE,HL	SAVE POSITION
1E53	2AEC40	LD	HL,(40ECH)	GET "CURRENT" LINE #
1E56	EB	EX	DE,HL	RESTORE POSITION
1E57	CA781D	JP	Z,1D78H	GET NEXT CHARACTER & RETURN
1E5A	2B	DEC	HL	BACKUP
1E5B	110000	LD	DE,0000H	LINE # = 0
1E5E	D7	RST	10H	GET NEXT CHARACTER
1E5F	D0	RET	NC	RETURN IF NOT NUMBER
1E60	E5	PUSH	HL	SAVE POSITION
1E61	F5	PUSH	AF	SAVE CHARACTER
1E62	219819	LD	HL,1998H	IS LINE # > 6552?
1E65	DF	RST	18H	
1E66	DA9719	JP	C,1997H	SYNTAX ERROR IF SO
1E69	62	LD	H,D	LINE #
1E6A	6B	LD	L,E	=
1E6B	19	ADD	HL,DE	LINE #
1E6C	29	ADD	HL,HL	*

1E6D	19	ADD	HL,DE	10
1E6E	29	ADD	HL,HL	
1E6F	F1	POP	AF	RESTORE CHARACTER
1E70	D630	SUB	30H	GET VALUE
1E72	5F	LD	E,A	DE = VALUE
1E73	1600	LD	D,00H	
1E75	19	ADD	HL,DE	LINE # = LINE #*10 + VALUE
1E76	EB	EX	DE,HL	DE = LINE #
1E77	E1	POP	HL	RESTORE POSITION
1E78	18E4	JR	1E5EH	LOOP FOR MORE

CLEAR

1E7A	CA611B	JP	Z,1B61H	RESET VARIABLES IF NO PARAM.
1E7D	CD461E	CALL	1E46H	EVALUATE NUMBER TO CLEAR
1E80	2B	DEC	HL	BACK UP ONE
1E81	D7	RST	10H	GET NEXT CHARACTER
1E82	C0	RET	NZ	RETURN IF MORE JUNK
1E83	E5	PUSH	HL	SAVE POSITION
1E84	2AB140	LD	HL, (40B1H)	GET MEMORY SIZE
1E87	7D	LD	A,L	DE
1E88	93	SUB	E	=
1E89	5F	LD	E,A	MEMORY SIZE
1E8A	7C	LD	A,H	-
1E8B	9A	SBC	A,D	SPACE TO CLEAR
1E8C	57	LD	D,A	
1E8D	DA7A19	JP	C,197AH	OM ERROR IF DE < 0
1E90	2AF940	LD	HL, (40F9H)	HL=
1E93	012800	LD	BC,0028H	END OF PROGRAM
1E96	09	ADD	HL,BC	+40
1E97	DF	RST	18H	COMPARE WITH NEW END OF MEMORY
1E98	D27A19	JP	NC,197AH	OM ERROR IF OVERLAP
1E9B	EB	EX	DE,HL	HL = NEW END OF AVAILABLE MEMORY
1E9C	22A040	LD	(40A0H),HL	SAVE NEW END OF AVAILABLE MEMORY
1E9F	E1	POP	HL	RESTORE POSITION
1EA0	C3611B	JP	1B61H	RESET VARIABLES & RETURN

RUN

1EA3	CA5D1B	JP	Z,1B5DH	CLEAR & GO IF NO PARAM
1EA6	CDC741	CALL	41C7H	LINK TO DOS
1EA9	CD611B	CALL	1B61H	RESET VARIABLES
1EAC	011E1D	LD	BC,1D1EH	BC => EXECUTOR
1EAF	1810	JR	1EC1H	CONTINUE GOTO LINE

GOSUB

1EB1	0E03	LD	C,03H	REQUEST 3 BYTES
1EB3	CD6319	CALL	1963H	CHECK IF ENOUGH SPACE
1EB6	C1	POP	BC	SAVE INTERPRETER RETURN
1EB7	E5	PUSH	HL	SAVE POSITION
1EB8	E5	PUSH	HL	TWICE
1EB9	2AA240	LD	HL, (40A2H)	GET CURRENT LINE #
1EBC	E3	EX	(SP),HL	SAVE LINE # RESTORE POSITION
1EBD	3E91	LD	A,91H	SAVE
1EBF	F5	PUSH	AF	GOSUB TOKEN
1EC0	33	INC	SP	TRIM EXTRA BYTE FROM STACK
1EC1	C5	PUSH	BC	SAVE INTERPRETER RETURN

GOTO

1EC2	CD5A1E	CALL	1E5AH	DECODE LINE #
1EC5	CD071F	CALL	1F07H	SKIP REST OF LINE
1EC8	E5	PUSH	HL	SAVE POSITION
1EC9	2AA240	LD	HL, (40A2H)	GET CURRENT LINE #
1ECC	DF	RST	18H	COMPARE WITH DESTINATION
1ECD	E1	POP	HL	RESTORE POSITION
1ECE	23	INC	HL	MOVE UP TO FORWARD POINTER
1ECF	DC2F1B	CALL	C,1B2FH	SCAN FORWARD IF LATER
1ED2	D42C1B	CALL	NC,1B2CH	SCAN WHOLE PROGRAM OTHERWISE
1ED5	60	LD	H,B	GET POSITION
1ED6	69	LD	L,C	OF LINE & ADJUST
1ED7	2B	DEC	HL	AS CURRENT POSITION
1ED8	D8	RET	C	RETURN IF LINE FORWARD
1ED9	1E0E	LD	E,0EH	UL ERROR 8
1EDB	C3A219	JP	19A2H	TO ERROR ROUTINE
				RETURN
1EDE	C0	RET	NZ	RETURN IF ANY PARAMS
1EDF	16FF	LD	D,0FFH	CLEAR STACK OF
1EE1	CD3619	CALL	1936H	ANY FORS SET UP
1EE4	F9	LD	SP,HL	IN SUBROUTINE
1EE5	22E840	LD	(40E8H),HL	SAVE NEW STACK POSITION
1EE8	FE91	CP	91H	GOSUB TOKEN?
1EEA	1E04	LD	E,04H	POSSIBLE RG ERROR
1EEC	C2A219	JP	NZ,19A2H	ERROR IF NO GOSUB DATA
1EEF	E1	POP	HL	GET RETURN LINE #
1EF0	22A240	LD	(40A2H),HL	SAVE AS CURRENT LINE #
1EF3	23	INC	HL	DIRECT MODE?
1EF4	7C	LD	A,H	
1EF5	B5	OR	L	
1EF6	2007	JR	NZ,1EFFH	SKIP IF NOT
1EF8	3ADD40	LD	A,(40DDH)	INPUTTING TEXT?
1EFB	B7	OR	A	
1EFC	C2181A	JP	NZ,1A18H	EXIT IF SO
1EFF	211E1D	LD	HL,1D1EH	SET MACHINE RETURN
1F02	E3	EX	(SP),HL	GET POSITION & SAVE RETURN
1F03	3E			MASK TO 1F05H
1F04	E1	POP	HL	GET POSITION

**DATA (1F05) (SKIP TO NEXT STATEMENT)
 & ELSE & REM (1F07) (SKIP TO NEXT LINE)**

1F05	013A0E	LD	BC,0E3AH	DELIMITER ":" SKIP TO 1F09
1F08	00	NOP		DELIMITER NULL (END OF LINE)
1F09	0600	LD	B,00H	ALTERNATE DELIMITER NULL
1F0B	79	LD	A,C	SWAP
1F0C	48	LD	C,B	DELIMITERS
1F0D	47	LD	B,A	
1F0E	7E	LD	A,(HL)	GET CHARACTER
1F0F	B7	OR	A	TEST IT
1F10	C8	RET	Z	RETURN IF END OF LINE
1F11	B8	CP	B	DELIMITER?
1F12	C8	RET	Z	RETURN IF SO
1F13	23	INC	HL	GO TO NEXT CHARACTER
1F14	FE22	CP	22H	DOUBLE QUOTES?
1F16	28F3	JR	Z,1F0BH	SWAP DELIMITERS IF SO.
1F18	D68F	SUB	8FH	IF TOKEN?
1F1A	20F2	JR	NZ,1F0EH	LOOP IF NOT

1F1C B8	CP	B	D = D+ 1
1F1D 8A	ADC	A,D	IF NOT IN A
1F1E 57	LD	D,A	QUOTED STRING
1F1F 18ED	JR	1F0EH	LOOP UNTIL DONE.

LET (EVALUATE EXPRESSION)

1F21 CD0D26	CALL	260DH	FIND VARIABLE TO STORE TO
1F24 CF	RST	08H	SYNTAX ERROR IF
1F25 D5			NEXT CHARACTER NOT "=" TOKEN
1F26 EB	EX	DE,HL	SAVE
1F27 22DF40	LD	(40DFH),HL	VARIABLE
1F2A EB	EX	DE,HL	POINTER
1F2B D5	PUSH	DE	SAVE IT IN STACK, TOO
1F2C E7	RST	20H	CHECK TYPE
1F2D F5	PUSH	AF	SAVE TYPE IN STACK
1F2E CD3723	CALL	2337H	EVALUATE EXPRESSION
1F31 F1	POP	AF	RESTORE TYPE
1F32 E3	EX	(SP),HL	SAVE POSITION, RESTORE VARIABLE
1F33 C603	ADD	A,03H	ADJUST TYPE
1F35 CD1928	CALL	2819H	CONVERT RESULT TO PROPER TYPE
1F38 CD030A	CALL	0A03H	DE => ACCUM
1F3B E5	PUSH	HL	SAVE VARIABLE POINTER
1F3C 2028	JR	NZ,1F66H	JUMP IF NOT STRING
1F3E 2A2141	LD	HL,(4121H)	HL => STRING INFO
1F41 E5	PUSH	HL	SAVE STRING INFO POINTER
1F42 23	INC	HL	DE => STRING
1F43 5E	LD	E,(HL)	
1F44 23	INC	HL	
1F45 56	LD	D,(HL)	
1F46 2AA440	LD	HL,(40A4H)	JUMP IF STRING
1F49 DF	RST	18H	IS LOCATED
1F4A 300E	JR	NC,1F5AH	BEFORE THE PROGRAM
1F4C 2AA040	LD	HL,(40A0H)	IS THE STRING
1F4F DF	RST	18H	IN THE SIRING SPACE?
1F50 D1	POP	DE	RESTORE INFO POINTER
1F51 300F	JR	NC,1F62H	JUMP IF NOT
1F53 2AF940	LD	HL,(40F9H)	JUMP IF THE
1F56 DF	RST	18H	STRING IS IN
1F57 3009	JR	NC,1F62H	THE PROGRAM
1F59 3E			MASK TO 1F5BH
1F5A D1	POP	DE	RESTORE INFO POINTER
1F5B CDF529	CALL	29F5H	DE => TEMPORARY STRING
1F5E EB	EX	DE,HL	
1F5F CD4328	CALL	2843H	MOVE TO STRING SPACE
1F62 CDF529	CALL	29F5H	CLEAN UP WORKSPACE
1F65 E3	EX	(SP),HL	SAVE RESULT POINTER. GET DESTINATION
1F66 CDD309	CALL	09D3H	MOVE DATA TO DESTINATION
1F69 D1	POP	DE	RESTORE VARIABLE POINTER
1F6A E1	POP	HL	RESTORE POSITION
1F6B C9	RET		DONE
			ON
1F6C FE9E	CP	9EH	ERROR?
1F6E 2025	JR	NZ,1F95H	JUMP IF NOT
1F70 D7	RST	10H	GET NEXT CHARACTER
1F71 CF	RST	08H	SYNTAX ERROR IF
1F72 8D			NEXT TOKEN NOT "GOTO"

1F73	CD5A1E	CALL	1E5AH	EVALUATE LINE #
1F76	7A	LD	A,D	ZERO?
1F77	B3	OR	E	
1F78	2809	JR	Z,1F83H	SKIP IF SO
1F7A	CD2A1B	CALL	1B2AH	SEARCH FOR LINE #
1F7D	50	LD	D,B	DE => LINE POSITION
1F7E	59	LD	E,C	
1F7F	E1	POP	HL	RESTORE POSITION
1F80	D2D91E	JP	NC,1ED9H	JUMP IF LINE NOT FOUND
1F83	EB	EX	DE,HL	SAVE ERROR
1F84	22F040	LD	(40F0H),HL	TRAP LOCATION
1F87	EB	EX	DE,HL	
1F88	D8	RET	C	RETURN IF LINE FOUND
1F89	3AF240	LD	A,(40F2H)	ALREADY PROCESSING
1F8C	B7	OR	A	AN ERROR?
1F8D	C8	RET	Z	RETURN IF NOT
1F8E	3A9A40	LD	A,(409AH)	GET ERROR CODE
1F91	5F	LD	E,A	
1F92	C3AB19	JP	19ABH	GOTO ERROR ROUTINE
<u>ON N...</u>				
1F95	CD1C2B	CALL	2B1CH	EVALUATE INDEX
1F98	7E	LD	A,(HL)	GET NEXT CHARACTER
1F99	47	LD	B,A	SAVE FOR LATER
1F9A	FE91	CP	91H	GOSUB?
1F9C	2803	JR	Z,1FA1H	SKIP IF SO
1F9E	CF	RST	08H	SYNTAX ERROR IF
1F9F	8D			TOKEN NOT GOTO
1FA0	2B	DEC	HL	BACKUP
1FA1	4B	LD	C,E	GET INDEX
1FA2	0D	DEC	C	ONE LESS LINE # TO SKIP
1FA3	78	LD	A,B	GET TOKEN
1FA4	CA601D	JP	Z,1D60H	EXECUTE BRANCH IF DONE
1FA7	CD5B1E	CALL	1E5BH	SKIP LINE #
1FAA	FE2C	CP	2CH	COMMA?
1FAC	C0	RET	NZ	RETURN IF NOT. NO ACTION
1FAD	18F3	JR	1FA2H	LOOP FOR MORE.
1FAF	11F240	LD	DE,40F2H	WAS THERE
1FB2	1A	LD	A,(DE)	AN ERROR?
1FB3	B7	OR	A	
1FB4	CAA019	JP	Z,19A0H	RW IF NOT
1FB7	3C	INC	A	CLEAR ERROR
1FB8	329A40	LD	(409AH),A	CLEAR ERROR TRAP FLAG
1FBB	12	LD	(DE),A	CLEAR ERROR FLAG
1FBC	7E	LD	A,(HL)	GET NEXT TOKEN
1FBD	FE87	CP	87H	NEXT?
1FBF	280C	JR	Z,1FCDH	JUMP IF SO
1FC1	CD5A1E	CALL	1E5AH	DECODE LINE #
1FC4	C0	RET	NZ	RETURN IF ANY MORE PARAMS
1FC5	7A	LD	A,D	GO TO
1FC6	B3	OR	E	LINE # IF
1FC7	C2C51E	JP	NZ,1EC5H	LINE # <> 0
1FCA	3C	INC	A	
1FCB	1802	JR	1FCFH	
<u>RESUME NEXT</u>				
1FCD	D7	RST	10H	SCAN FOR MORE PARAMS

1FCE C0	RET	NZ	RETURN IF THERE WERE ANY
1FCF 2AEE40	LD	HL, (40EEH)	GET POSITION BEFORE ERROR
1FD2 EB	EX	DE,HL	SAVE TEMPORARILY
1FD3 2AEA40	LD	HL, (40EAH)	GET OLD LINE #
1FD6 22A240	LD	(40A2H),HL	SAVE AS CURRENT LINE #
1FD9 EB	EX	DE,HL	RESTORE POSITION
1FDA C0	RET	NZ	DONE IF RESUME 0
1FDB 7E	LD	A, (HL)	GET NEXT CHARACTER
1FDC B7	OR	A	END OF LINE?
1FDD 2004	JR	NZ, 1FE3H	SKIP IF NOT
1FDF 23	INC	HL	SKIP OVER
1FE0 23	INC	HL	LINE HEADER
1FE1 23	INC	HL	DATA
1FE2 23	INC	HL	
1FE3 23	INC	HL	NEXT CHARACTER
1FE4 7A	LD	A, D	DIRECT MODE?
1FE5 A3	AND	E	
1FE6 3C	INC	A	
1FE7 C2051F	JP	NZ, 1F05H	SKIP REST OF STATEMENT IF SO
1FEA 3ADD40	LD	A, (40DDH)	INPUTTING TEXT
1FED 3D	DEC	A	
1FEE CABE1D	JP	Z, 1DBEH	STOP IF NOT
1FF1 C3051F	JP	1F05H	SKIP REST OF STATEMENT & RETURN

ERROR

1FF4 CD1C2B	CALL	2B1CH	EVALUATE PARAM
1FF7 C0	RET	NZ	RETURN IF ANY MORE
1FF8 B7	OR	A	ERROR 0
1FF9 CA4A1E	JP	Z, 1E4AH	FC ERROR IF SO
1FFC 3D	DEC	A	ERR= (ERROR -1)*2
1FFD 87	ADD	A, A	
1FFE 5F	LD	E, A	STORE ERROR CODE
1FFF FE2D	CP	2DH	VALID ERROR CODE
2001 3802	JR	C, 2005H	SKIP IF SO
2003 1E26	LD	E, 26H	UE ERROR
2005 C3A219	JP	19A2H	GOTO ERROR ROUTINE

AUTO

2008 110A00	LD	DE, 000AH	DEFAULT START = 10
200B D5	PUSH	DE	SAVE DEFAULT START
200C 2817	JR	Z, 2025H	JUMP IF NO PARAMS
200E CD4F1E	CALL	1E4FH	DECODE START LINE #
2011 EB	EX	DE,HL	SAVE START
2012 E3	EX	(SP),HL	LINE # ON STACK
2013 2811	JR	Z, 2026H	JUMP IF NO MORE PARAMS
2015 EB	EX	DE,HL	RESTORE POSITION
2016 CF	RST	08H	SYNTAX ERROR IF NEXT
2017 2C	" , "		CHARACTER IS NOT A COMMA
2018 EB	EX	DE,HL	SAVE POSITION
2019 2AE440	LD	HL, (40E4H)	GET OLD AUTO INCREMENT
201C EB	EX	DE,HL	RESTORE POSITION
201D 2806	JR	Z, 2025H	SKIP IF NO MORE PARAMS
201F CD5A1E	CALL	1E5AH	DECODE INCREMENT
2022 C29719	JP	NZ, 1997H	ERROR IF MORE PARAMS
2025 EB	EX	DE,HL	PUT INCREMENT IN HL
2026 7C	LD	A, H	INCREMENT
2027 B5	OR	L	ZERO?

2028	CA4A1E	JP	Z,1E4AH	FC ERROR IF SO
202B	22E440	LD	(40E4H),HL	SAVE INCREMENT
202E	32E140	LD	(40E1H),A	SAVE AUTO FLAG
2031	E1	POP	HL	SAVE
2032	22E240	LD	(40E2H),HL	START LINE #
2035	C1	POP	BC	CLEAR RETURN
2036	C3331A	JP	1A33H	ENTER COMMAND MODE

IF

2039	CD3723	CALL	2337H	EVALUATE EXPRESSION
203C	7E	LD	A,(HL)	GET CHARACTER OR TOKEN
203D	FE2C	CP	2CH	COMMA?
203F	CC781D	CALL	Z,1D78H	SKIP COMMA
2042	FECA	CP	0CAH	THEN?
2044	CC781D	CALL	Z,1D78H	SKIP THEN
2047	2B	DEC	HL	BACKUP
2048	E5	PUSH	HL	SAVE POSITION
2049	CD9409	CALL	0994H	TEST SIGN OF EXPRESSION
204C	E1	POP	HL	RESTORE POSITION
204D	2807	JR	Z,2056H	JUMP IF FALSE
204F	D7	RST	10H	SCAN TO NEXT CHARACTER
2050	DAC21E	JP	C,1EC2H	GOTO IF NUMERIC
2053	C35F1D	JP	1D5FH	ELUTE STATEMENTS
2056	1601	LD	D,01H	D = # NESTED IFS
2058	CD051F	CALL	1F05H	SCAN TO END OF STATEMENT
205B	B7	OR	A	END OF LINE?
205C	C8	RET	Z	RETURN IF SO
205D	D7	RST	10H	GET TOKEN
205E	FE95	CP	95H	ELSE?
2060	20F6	JR	NZ,2058H	NEXT STATEMENT IF NOT
2062	15	DEC	D	EXIT NEST BY ONE LEVEL
2063	20F3	JR	NZ,2058H	LOOP IF NOT OUT FAR ENOUGH
2065	18E8	JR	204FH	EXECUTE STATEMENTS

LPRINT

2067	3E01	LD	A,01H	ROUTE OUTPUT
2069	329C40	LD	(409CH),A	TO PRINTER
206C	C37C20	JP	207CH	CONTINUE

PRINT

206F	CDCA41	CALL	41CAH	DOS LINK
2072	FE23	CP	23H	PRINT#?
2074	2006	JR	NZ,207CH	SKIP IF NOT
2076	CD8402	CALL	0284H	TURN TAPE ON. NO HEADER
2079	329C40	LD	(409CH),A	ROUTE TO TAPE
207C	2B	DEC	HL	BACKUP
207D	D7	RST	10H	END OF STATEMENT?
207E	CCFE20	CALL	Z,20FEH	NEW LINE IF SO
2081	CA6921	JP	Z,2169H	FINISH UP IF SO
2084	F620	OR	20H	PRINT @?
2086	FE60	CP	60H	
2088	201B	JR	NZ,20A5H	SKIP IF NOT
208A	CD012B	CALL	2B01H	GET SCREEN POSITION
208D	FE04	CP	04H	OFF SCREEN?
208F	D24A1E	JP	NC,1E4AH	FC ERROR IF SO
2092	E5	PUSH	HL	SAVE TEXT POSITIONS
2093	21003C	LD	HL,3C00H	COMPUTE SCREEN

2096	19	ADD	HL,DE	MEMORY POSITION
2097	222040	LD	(4020H),HL	POSITION CURSOR
209A	7B	LD	A,E	COMPUTE &
209B	E63F	AND	3FH	SAVE POSITION
209D	32A640	LD	(40A6H),A	
20A0	E1	POP	HL	RESTORE TEXT POSITION
20A1	CF	RST	08H	SYNTAX ERROR IF NEXT
20A2	2C	" , "		CHARACTER IS NOT A COMMA
20A3	18C7	JR	206CH	NEXT ITEM
20A5	7E	LD	A,(HL)	GET NEXT TOKEN
20A6	FEBF	CP	0BFH	USING?
20A8	CABD2C	JP	Z,2CBDH	JUMP IF SO
20AB	FEBC	CP	0BCH	TAB?
20AD	CA3721	JP	Z,2137H	JUMP IF SO
20B0	E5	PUSH	HL	SAVE TEXT POSITION
20B1	FE2C	CP	2CH	COMMA?
20B3	2853	JR	Z,2108H	JUMP IF SO
20B5	FE3B	CP	3BH	SEMI-COLON?
20B7	285E	JR	Z,2117H	JUMP IF SO
20B9	CD3723	CALL	2337H	EVALUATE EXPRESSION
20BC	E3	EX	(SP),HL	RESTORE POSITION
20BD	E7	RST	20H	CHECK TYPE OF ACCUM
20BE	2832	JR	Z,20F2H	JUMP IF STRING
20C0	CDBD0F	CALL	0FBDH	COVERT NUMBER TO ASCII
20C3	CD6528	CALL	2865H	PRINT ASCII STRING
20C6	CD41	CALL	41CDH	DOS LINK
20C9	2A2141	LD	HL,(4121H)	POINT TO STRING DESCRIPTOR
20CC	3A9C40	LD	A,(409CH)	CHECK DEVICE
20CF	B7	OR	A	
20D0	FAE920	JP	M,20E9H	JUMP IF TO TAPE
20D3	2808	JR	Z,20DDH	JUMP IF TO VIDEO
20D5	3A9B40	LD	A,(409BH)	GET PRINTER POSITION
20D8	86	ADD	A,(HL)	COMPUTE NEW POSITION
20D9	FE84	CP	84H	PAST COLUMN 132?
20DB	1809	JR	20E6H	CONTINUE
20DD	3A9D40	LD	A,(409DH)	GET SIZE OF LINE
20E0	47	LD	B,A	
20E1	3AA640	LD	A,(40A6H)	GET POSITION IN LINE
20E4	86	ADD	A,(HL)	ADD LENGTH OF STRING TO POSITION
20E5	B8	CP	B	TOO FAR DOWN LINE?
20E6	D4FE20	CALL	NC,20FEH	NEW LINE IF SO
20E9	CDAA28	CALL	28AAH	DISPLAY NUMBER
20EC	3E20	LD	A,20H	DISPLAY A SPACE
20EE	CD2A03	CALL	032AH	AFTER THE NUMBER
20F1	B7	OR	A	SET NON-ZERO
20F2	CCAA28	CALL	Z,28AAH	
20F5	E1	POP	HL	RESTORE TEXT POSITION
20F6	C37C20	JP	207CH	CONTINUE
20F9	3AA640	LD	A,(40A6H)	GET POS
20FC	B7	OR	A	TEST IT
20FD	C8	RET	Z	RETURN IF 0
20FE	3E0D	LD	A,0DH	GO TO
2100	CD2A03	CALL	032AH	NEW LINE
2103	CDD041	CALL	41D0H	DOS LINK
2106	AF	XOR	A	CLEAR STATUS
2107	C9	RET		DONE
2108	CDD341	CALL	41D3H	DOS LINK

210B	3A9C40	LD	A, (409CH)	GET DEVICE FLAG
210E	B7	OR	A	TEST IT
210F	F21921	JP	P, 2119H	SKIP IF NOT TAPE
2112	3E2C	LD	A, 2CH	OUTPUT
2114	CD2A03	CALL	032AH	A COMMA
2117	184B	JR	2164H	CONTINUE
2119	2808	JR	Z, 2123H	SKIP IF VIDEO
211B	3A9B40	LD	A, (409BH)	GET PRINTER POSITION
211E	FE70	CP	70H	CHECK IF TOO FAR?
2120	C32B21	JP	212BH	CONTINUE
2123	3A9E40	LD	A, (409EH)	GET VIDEO WIDTH
2126	47	LD	B, A	PUT IN B
2127	3AA640	LD	A, (40A6H)	GET VIDEO POSITION
212A	B8	CP	B	TOO FAR DOWN LINE?
212B	D4FE20	CALL	NC, 20FEH	NEW LINE IF SO
212E	3034	JR	NC, 2164H	SKIP OUT IF SO
2130	D610	SUB	10H	COMPUTE # SPACES
2132	30FC	JR	NC, 2130H	TO TAB
2134	2F	CPL		INVERT IT
2135	1823	JR	215AH	CONTINUE
2137	CD1B2B	CALL	2B1BH	EVALUATE TAB
213A	E67F	AND	7FH	MASK BIT 7 OFF
213C	5F	LD	E, A	SAVE FOR LATER
213D	CF	RST	08H	IF NEXT CHARACTER NOT
213E	29) ") " THEN SYNTAX ERROR
213F	2B	DEC	HL	BACKUP
2140	E5	PUSH	HL	SAVE POSITION
2141	CDD341	CALL	41D3H	DOS LINK
2144	3A9C40	LD	A, (409CH)	CHECK
2147	B7	OR	A	ROUTING
2148	FA4A1E	JP	M, 1E4AH	FC ERROR IF TAPE
214B	CA5321	JP	Z, 2153H	SKIP IF VIDEO
214E	3A9B40	LD	A, (409BH)	GET PRINTER HEAD POSITION
2151	1803	JR	2156H	CONTINUE
2153	3AA640	LD	A, (40A6H)	GET VIDEO POSITION
2156	2F	CPL		COMPUTE TAB-POS-1
2157	83	ADD	A, E	
2158	300A	JR	NC, 2164H	IGNORE IF PAST TAB
215A	3C	INC	A	B = TAB - POS
215B	47	LD	B, A	
215C	3E20	LD	A, 20H	SPACE
215E	CD2A03	CALL	032AH	OVER
2161	05	DEC	B	LOOP UNTIL
2162	20FA	JR	NZ, 215EH	DONE
2164	E1	POP	HL	RESTORE POSITION
2165	D7	RST	10H	GET NEXT CHARACTER
2166	C38120	JP	2081H	LOOP
2169	3A9C40	LD	A, (409CH)	CHECK
216C	B7	OR	A	ROUTING
216D	FCF801	CALL	M, 01F8H	TURN OFF TAPE
2170	AF	XOR	A	RESET ROUTING
2171	329C40	LD	(409CH), A	TO VIDEO
2174	CDBE41	CALL	41BEH	DOS LINK
2177	C9	RET		DONE
2178	3F5245444F	OD00		"?REDO" 0D 00
217F	3ADE40	LD	A, (40DEH)	READ ERROR?
2182	B7	OR	A	

2183	C29119	JP	NZ,1991H	SYNTAX ERROR IF SO
2186	3AA940	LD	A,(40A9H)	TAPE INPUT ERROR
2189	B7	OR	A	
218A	1E2A	LD	E,2AH	FD ERROR
218C	CAA219	JP	Z,19A2H	IF SO
218F	C1	POP	BC	CLEAR STACK
2190	217821	LD	HL,2178H	PRINT
2193	CDA728	CALL	28A7H	"REDO"
2196	2AE640	LD	HL,(40E6H)	RESTORE POSITION
2199	C9	RET		CONTINUE

INPUT

219A	CD2828	CALL	2828H	CHECK FOR ILLEGAL DIRECT
219D	7E	LD	A,(HL)	GET CHARACTER
219E	CDD641	CALL	41D6H	DOS LINK
21A1	D623	SUB	23H	INPUT#?
21A3	32A940	LD	(40A9H),A	SAVE STATUS
21A6	7E	LD	A,(HL)	GET CHARACTER
21A7	2020	JR	NZ,21C9H	JUMP IF KEYBOARD INPUT

TAPE INPUT

21A9	CD9302	CALL	0293H	START TAPE & READ HEADER
21AC	E5	PUSH	HL	SAVE TEXT POSITION
21AD	06FA	LD	B,0FAH	250 BYTE MAX.
21AF	2AA740	LD	HL,(40A7H)	HL => BUFFER
21B2	CD3502	CALL	0235H	READ BYTE
21B5	77	LD	(HL),A	SAVE BYTE
21B6	23	INC	HL	POINT TO NEXT BYTE
21B7	FE0D	CP	0DH	END OF LINE?
21B9	2802	JR	Z,21BDH	SKIP OUT IF SO
21BB	10F5	DJNZ	21B2H	LOOP TO BUFFER FULL
21BD	2B	DEC	HL	MARK END
21BE	3600	LD	(HL),00H	OF INPUT STRING
21C0	CD801	CALL	01F8H	STOP TAPE
21C3	2AA740	LD	HL,(40A7H)	HL =>JUST BEFORE BUFFER
21C6	2B	DEC	HL	
21C7	1822	JR	21EBH	DECODE INPUT
21C9	01DB21	LD	BC,21DBH	SET RETURN
21CC	C5	PUSH	BC	TO 21DBH
21CD	FE22	CP	22H	DOUBLE QUOTES?
21CF	C0	RET	NZ	RETURN IF NOT
21D0	CD6628	CALL	2866H	PROCESS QUOTED STRING
21D3	CF	RST	08H	SYNTAX ERROR IF NEXT
21D4	3B		","	CHARACTER NOT SEMI-COLON
21D5	E5	PUSH	HL	SAVE POSITION
21D6	CDAA28	CALL	28AAH	OUTPUT STRING
21D9	E1	POP	HL	RESTORE POSITION
21DA	C9	RET		RETURN
21DB	E5	PUSH	HL	SAVE POSITION IN TEXT
21DC	CDB31B	CALL	1BB3H	PRINT "?" & INPUT DATA
21DF	C1	POP	BC	RESTORE POSITION
21E0	DABE1D	JP	C,1DBEH	STOP IF <BREAK>
21E3	23	INC	HL	NULL
21E4	7E	LD	A,(HL)	INPUT?
21E5	B7	OR	A	
21E6	2B	DEC	HL	
21E7	C5	PUSH	BC	SAVE POSITION

21E8	CA041F	JP	Z,1F04H	NO ACTION IF NULL INPUT
21EB	362C	LD	(HL),2CH	PLACE A COMMA BEFORE
21ED	1805	JR	21F4H	DATA & CONTINUE
<u>READ</u>				
21EF	E5	PUSH	HL	SAVE POSITION IN TEXT
21F0	2AFF40	LD	HL,(40FFH)	HL => CURRENT DATA LOCATION
21F3	F6AF	OR	0AFH	SET READ & SKIP TO 21F5H
21F4	AF	XOR	A	SET INPUT
21F5	32DE40	LD	(40DEH),A	SAVE READ/INPUT FLAG
21F8	E3	EX	(SP),HL	SAVE DATA POINTER. GET TEXT POINTER
21F9	1802	JR	21FDH	CONTINUE
21FB	CF	RST	08H	SYNTAX ERROR IF NEXT
21FC	2C	" , "		CHARACTER IS NOT A COMMA
21FD	CD0D26	CALL	260DH	FIND VARIABLE
2200	E3	EX	(SP),HL	GET DATA POINTER. SAVE TEXT POINTER
2201	D5	PUSH	DE	SAVE VARIABLE POINTER
2202	7E	LD	A,(HL)	GET DATA CHARACTER.
2203	FE2C	CP	2CH	COMMA?
2205	2826	JR	Z,222DH	SKIP IF SO.
2207	3ADE40	LD	A,(40DEH)	GO GET
220A	B7	OR	A	MORE DATA
220B	C29622	JP	NZ,2296H	IF READ
220E	3AA940	LD	A,(40A9H)	OD
2211	B7	OR	A	ERROR
2212	1E06	LD	E,06H	IF
2214	CAA219	JP	Z,19A2H	TAPE
2217	3E3F	LD	A,3FH	PRINT
2219	CD2A03	CALL	032AH	"??"
221C	CDB31B	CALL	1BB3H	AND INPUT MORE DATA
221F	D1	POP	DE	CLEAR
2220	C1	POP	BC	STACK
2221	DABE1D	JP	C,1DBEH	STOP IF <BREAK>
2224	23	INC	HL	WAS
2225	7E	LD	A,(HL)	INPUT
2226	B7	OR	A	LINE
2227	2B	DEC	HL	NULL?
2228	C5	PUSH	BC	FIX STACK
2229	CA041F	JP	Z,1F04H	IGNORE REST OF REQUEST IF SO
222C	D5	PUSH	DE	FIX STACK
222D	CDDC41	CALL	41DCH	DOS LINK
2230	E7	RST	20H	GET TYPE OF ACCUM
2231	F5	PUSH	AF	SAVE TYPE
2232	2019	JR	NZ,224DH	JUMP IF NOT STRING
2234	D7	RST	10H	GET NEXT CHARACTER.
2235	57	LD	D,A	DELIMITERS
2236	47	LD	B,A	ARE DOUBLE QUOTES
2237	FE22	CP	22H	DOUBLE QUOTES?
2239	2805	JR	Z,2240H	SKIP IF SO
223B	163A	LD	D,3AH	DELIMITER # 2 = ":"
223D	062C	LD	B,2CH	DELIMITER # 1 = ","
223F	2B	DEC	HL	BACK UP
2240	CD6928	CALL	2869H	GET STRING FROM DATA
2243	F1	POP	AF	RESTORE TYPE OF ACCUM
2244	EB	EX	DE,HL	SAVE DATA POINTER
2245	215A22	LD	HL,225AH	SET RETURN TO 225AH
2248	E3	EX	(SP),HL	AND RESTORE PROGRAM POINTER

2249	D5	PUSH	DE	SAVE DATA POINTER
224A	C3331F	JP	1F33H	ASSIGN DATA TO VARIABLE
224D	D7	RST	10H	GET NEXT CHARACTER
224E	F1	POP	AF	GET TYPE
224F	F5	PUSH	AF	OF ACCUM
2250	014322	LD	BC,2243H	SET RETURN
2253	C5	PUSH	BC	TO 2243H
2254	DA6C0E	JP	C,0E6CH	DECODE S.P. & INTEGERS
2257	D2650E	JP	NC,0E65H	DECODE D.P. NUMBERS
225A	2B	DEC	HL	BACK UP DATA POINTER
225B	D7	RST	10H	GET NEXT CHARACTER
225C	2805	JR	Z,2263H	SKIP IF END OF LINE
225E	FE2C	CP	2CH	COMMA?
2260	C27F21	JP	NZ,217FH	ERROR IF NOT
2263	E3	EX	(SP),HL	SAVE DATA POINTER. GET TEXT POINTER.
2264	2B	DEC	HL	BACKUP
2265	D7	RST	10H	GET CHARACTER.
2266	C2FB21	JP	NZ,21FBH	LOOP BACK IF MORE REQUESTED
2269	D1	POP	DE	GET DATA POINTER
226A	00	NOP		DELETED
226B	00	NOP		ROUTINE
226C	00	NOP		
226D	00	NOP		
226E	00	NOP		
226F	3ADE40	LD	A, (40DEH)	READ?
2272	B7	OR	A	
2273	EB	EX	DE,HL	HL => DATA
2274	C2961D	JP	NZ,1D96H	SAVE POINTER & RETURN IF SO
2277	D5	PUSH	DE	SAVE PROGRAM POINTER
2278	CDDF41	CALL	41DFH	DOS LINK
227B	B6	OR	(HL)	ANY MORE DATA?
227C	218622	LD	HL,2286H	PRINT "?EXTRA IGNORED"
227F	C4A728	CALL	NZ,28A7H	IF THERE WAS
2282	E1	POP	HL	RESTORE PROGRAM POINTER
2283	C36921	JP	2169H	CLEAN UP & RETURN
2286	3F45787472612069676E6F726564			"?EXTRA IGNORED"
2294	0D00			0D 00
2296	CD051F	CALL	1F05H	SCAN PAST STATEMENT
2299	B7	OR	A	END OF LINE?
229A	2012	JR	NZ,22AEH	SKIP IF NOT
229C	23	INC	HL	END OF
229D	7E	LD	A, (HL)	PROGRAM?
229E	23	INC	HL	
229F	B6	OR	(HL)	
22A0	1E06	LD	E,06H	OD ERROR
22A2	CAA219	JP	Z,19A2H	IF SO
22A5	23	INC	HL	GET
22A6	5E	LD	E, (HL)	CURRENT
22A7	23	INC	HL	DATA
22A8	56	LD	D, (HL)	LINE #
22A9	EB	EX	DE,HL	SAVE LINE #
22AA	22DA40	LD	(40DAH),HL	
22AD	EB	EX	DE,HL	
22AE	D7	RST	10H	GET TOKEN
22AF	FE88	CP	88H	DATA?
22B1	20E3	JR	NZ,2296H	LOOP IF NOT

22B3	C32D22	JP	222DH	CONTINUE.
				<u>NEXT</u>
22B6	110000	LD	DE,0000H	CLEAR DE (NO INDEX)
22B9	C40D26	CALL	NZ,260DH	GET PARAM, IF ANY
22BC	22DF40	LD	(40DFH),HL	SAVE POSITION IN TEXT
22BF	CD3619	CALL	1936H	SEARCH FOR FOR DATA
22C2	C29D19	JP	NZ,199DH	NF ERROR IF NOT FOUND
22C5	F9	LD	SP,HL	POSITION STACK
22C6	22E840	LD	(40E8H),HL	SAVE STACK POSITION
22C9	D5	PUSH	DE	SAVE COUNTER POINTER
22CA	7E	LD	A,(HL)	GET TYPE OF COUNTER
22CB	23	INC	HL	NEXT BYTE
22CC	F5	PUSH	AF	SAVE SIGN OF STEP
22CD	D5	PUSH	DE	SAVE COUNTER POINTER
22CE	7E	LD	A,(HL)	GET TYPE OF COUNTER
22CF	23	INC	HL	NEXT BYTE
22D0	B7	OR	A	TEST TYPE
22D1	FAEA22	JP	M,22EAH	JUMP IF INTEGER
22D4	CDB109	CALL	09B1H	ACCUM = STEP
22D7	E3	EX	(SP),HL	SAVE FOR DATA POINTER
22D8	E5	PUSH	HL	SAVE COUNTER POINTER
22D9	CD0B07	CALL	070BH	ACCUM = ACCUM + COUNTER
22DC	E1	POP	HL	GET COUNTER POINTER
22DD	CDCB09	CALL	09CBH	SAVE UPDATED COUNTER
22E0	E1	POP	HL	GET FOR DATA POINTER
22E1	CDC209	CALL	09C2H	BCDE = UPPER LIMIT (TO)
22E4	E5	PUSH	HL	SAVE FOR DATA POINTER
22E5	CD0C0A	CALL	0A0CH	COMPARE. DONE YET?
22E8	1829	JR	2313H	CONTINUE
22EA	23	INC	HL	LOAD
22EB	23	INC	HL	STEP
22EC	23	INC	HL	INTO
22ED	23	INC	HL	BC
22EE	4E	LD	C,(HL)	
22EF	23	INC	HL	
22F0	46	LD	B,(HL)	
22F1	23	INC	HL	
22F2	E3	EX	(SP),HL	LOAD
22F3	5E	LD	E,(HL)	COUNTER
22F4	23	INC	HL	INTO
22F5	56	LD	D,(HL)	DE
22F6	E5	PUSH	HL	SAVE VARIABLE & COUNTER POINTER
22F7	69	LD	L,C	HL =
22F8	60	LD	H,B	BC
22F9	CDD20B	CALL	0BD2H	HL = COUNTER + STEP
22FC	3AAF40	LD	A,(40AFH)	OV
22FF	FE04	CP	04H	ERROR
2301	CAB207	JP	Z,07B2H	IF ANSWER S.P.
2304	EB	EX	DE,HL	SAVE COUNTER IN DE
2305	E1	POP	HL	GET COUNTER POINTER
2306	72	LD	(HL),D	SAVE
2307	2B	DEC	HL	COUNTER
2308	73	LD	(HL),E	TO MEMORY
2309	E1	POP	HL	GET FOR DATA POINTER
230A	D5	PUSH	DE	SAVE COUNTER
230B	5E	LD	E,(HL)	GET

230C	23	INC	HL	TOO
230D	56	LD	D, (HL)	VALUE
230E	23	INC	HL	
230F	E3	EX	(SP), HL	SAVE FOR DATA POINTER. GET COUNTER
2310	CD390A	CALL	0A39H	COMPARE. DONE YET?
2313	E1	POP	HL	GET FOR DATA POINTER
2314	C1	POP	BC	GET SIGN OF STEP
2315	90	SUB	B	A = SIGN OF STEP
2316	CDC209	CALL	09C2H	CLEAR STACK AND GET LOOP START
2319	2809	JR	Z, 2324H	JUMP IF DONE
231B	EB	EX	DE, HL	SAVE LOOP
231C	22A240	LD	(40A2H), HL	STARTING LINE #
231F	69	LD	L, C	SET LOOP
2320	60	LD	H, B	STARTING POSITION
2321	C31A1D	JP	1D1AH	RE-EXECUTE LOOP
2324	F9	LD	SP, HL	CLEAR FOR DATA FROM STACK
2325	22E840	LD	(40E8H), HL	SAVE NEW STACK POSITION
2328	2ADF40	LD	HL, (40DFH)	GET POSITION IN TEXT
232B	7E	LD	A, (HL)	GET NEXT CHARACTER
232C	FE2C	CP	2CH	COMMA?
232E	C21E1D	JP	NZ, 1D1EH	NEXT STATEMENT IF NOT
2331	D7	RST	10H	GET NEXT CHARACTER
2332	CDB922	CALL	22B9H	DO ANOTHER NEXT.

EXPRESSION EVALUATION

2335	CF	RST	08H	SYNTAX ERROR
2336	28	" "		IF NEXT CHARACTER NOT " "
2337	2B	DEC	HL	BACKUP
2338	1600	LD	D, 00H	PRIORITY = 0
233A	D5	PUSH	DE	SAVE PRIORITY
233B	0E01	LD	C, 01H	REQUEST ONE BYTE
233D	CD6319	CALL	1963H	CHECK FOR ENOUGH SPACE
2340	CD9F24	CALL	249FH	EVALUATE ITEM
2343	22F340	LD	(40F3H), HL	SAVE POSITION
2346	2AF340	LD	HL, (40F3H)	GET POSITION
2349	C1	POP	BC	GET PRIORITY
234A	7E	LD	A, (HL)	GET OPERATOR
234B	1600	LD	D, 00H	GET OPERATOR PRIORITY = 0
234D	D6D4	SUB	0D4H	ADJUST TO RELATIONAL OPERATOR
234F	3813	JR	C, 2364H	JUMP IF NOT RELATIONAL
2351	FE03	CP	03H	CHECK AGAIN
2353	300F	JR	NC, 2364H	JUMP IF NOT RELATIONAL
2355	FE01	CP	01H	COMPUTE
2357	17	RLA		OPERATOR
2358	AA	XOR	D	PRIORITY
2359	BA	CP	D	
235A	57	LD	D, A	SAVE PRIORITY
235B	DA9719	JP	C, 1997H	SYNTAX ERROR IF BAD
235E	22D840	LD	(40D8H), HL	SAVE POSITION
2361	D7	RST	10H	GET NEXT CHARACTER
2362	18E9	JR	234DH	CHECK FOR SECOND CHARACTER.
2364	7A	LD	A, D	WAS THERE A
2365	B7	OR	A	RELATIONAL OPERATOR?
2366	C2EC23	JP	NZ, 23ECH	JUMP IF SO
2369	7E	LD	A, (HL)	GET CHARACTER.
236A	22D840	LD	(40D8H), HL	SAVE POSITION
236D	D6CD	SUB	0CDH	MATHEMATICAL OPERATOR?

236F	D8	RET	C	RETURN IF NOT
2370	FE07	CP	07H	CHECK AGAIN
2372	D0	RET	NC	RETURN IF NOT
2373	5F	LD	E,A	SAVE # OF OPERATOR
2374	3AAF40	LD	A, (40AFH)	GET TYPE OF ACCUM
2377	D603	SUB	03H	ADJUST TYPE
2379	B3	OR	E	COMBINE WITH # OF OPERATOR
237A	CA8F29	JP	Z,298FH	JUMP IF STRING ADDITION
237D	219A18	LD	HL,189AH	HL =>
2380	19	ADD	HL,DE	OPERATOR PRIORITY
2381	78	LD	A,B	GET PREVIOUS PRIORITY
2382	56	LD	D, (HL)	GET NEW PRIORITY
2383	BA	CP	D	COMPARE OLD & NEW
2384	D0	RET	NC	RETURN IF OLD HIGHER
2385	C5	PUSH	BC	SAVE PREVIOUS PRIORITY
2386	014623	LD	BC,2346H	SET RETURN
2389	C5	PUSH	BC	THROUGH EVALUATOR
238A	7A	LD	A,D	GET NEW PRIORITY
238B	FE7F	CP	7FH	EXPONENTIATION?
238D	CAD423	JP	Z,23D4H	JUMP IF SO
2390	FE51	CP	51H	LOGICAL OPERATOR?
2392	DAE123	JP	C,23E1H	JUMP IF SO
2395	212141	LD	HL,4121H	HL => ACCUM
2398	B7	OR	A	CLEAR FLAGS
2399	3AAF40	LD	A, (40AFH)	GET TYPE OF ACCUM
239C	3D	DEC	A	ADJUST
239D	3D	DEC	A	TYPE
239E	3D	DEC	A	
239F	CAF60A	JP	Z,0AF6H	TM ERROR IF STRING
23A2	4E	LD	C, (HL)	GET INTEGER PART
23A3	23	INC	HL	OF ACCUM
23A4	46	LD	B, (HL)	TO THE STACK
23A5	C5	PUSH	BC	
23A6	FAC523	JP	M,23C5H	JUMP IF INTEGER
23A9	23	INC	HL	GET TOP PART OF
23AA	4E	LD	C, (HL)	FLOATING POINT
23AB	23	INC	HL	NUMBER IN ACCUM
23AC	46	LD	B, (HL)	TO THE STACK
23AD	C5	PUSH	BC	
23AE	F5	PUSH	AF	SAVE STATUS
23AF	B7	OR	A	TEST IT
23B0	E2C423	JP	PO,23C4H	JUMP IF S.P.
23B3	F1	POP	AF	RESTORE STATUS
23B4	23	INC	HL	NEXT BYTE
23B5	3803	JR	C,23BAH	SKIP IF NOT D.P.
23B7	211D41	LD	HL,411DH	POSITION TO SAVE REST OF ACCUM
23BA	4E	LD	C, (HL)	SAVE
23BB	23	INC	HL	BOTTOM
23BC	46	LD	B, (HL)	FOUR
23BD	23	INC	HL	BYTES
23BE	C5	PUSH	BC	OF
23BF	4E	LD	C, (HL)	ACCUM
23C0	23	INC	HL	
23C1	46	LD	B, (HL)	
23C2	C5	PUSH	BC	
23C3	06			MASK TO 23C5H
23C4	F1	POP	AF	CLEAR STACK GET TYPE

23C5	C603	ADD	A,03H	ADJUST TYPE
23C7	4B	LD	C,E	SAVE PRIORITY
23C8	47	LD	B,A	& TYPE
23C9	C5	PUSH	BC	ON STACK
23CA	010624	LD	BC,2406H	SET RETURN THROUGH 2406H
23CD	C5	PUSH	BC	SAVE RETURN
23CE	2AD840	LD	HL,(40D8H)	GET POSITION
23D1	C33A23	JP	233AH	CONTINUE. EVALUATE EXPRESSION
23D4	CDB10A	CALL	0AB1H	CONVERT ACCUM TO S.P.
23D7	CDA409	CALL	09A4H	PUT ACCUM ON STACK
23DA	01F213	LD	BC,13F2H	RETURN THROUGH EXPONENTIATION
23DD	167F	LD	D,7FH	PRIORITY = 7FH
23DF	18EC	JR	23CDH	EVALUATE EXPONENT & RETURN
23E1	D5	PUSH	DE	SAVE PRIORITY
23E2	CD7F0A	CALL	0A7FH	HL = FIRST ARGUMENT
23E5	D1	POP	DE	RESTORE PRIORITY
23E6	E5	PUSH	HL	SAVE FIRST ARGUMENT
23E7	01E925	LD	BC,25E9H	RETURN THROUGH LOGIC
23EA	18E1	JR	23CDH	ROUTINE AFTER EVALUATION OF SECOND ARGUMENT
23EC	78	LD	A,B	GET OLD PRIORITY
23ED	FE64	CP	64H	IF IT WAS NOT
23EF	D0	RET	NC	RELATIONAL, EXIT.
23F0	C5	PUSH	BC	SAVE OLD PRIORITY
23F1	D5	PUSH	DE	SAVE NEW PRIORITY
23F2	110464	LD	DE,6404H	SET PRIORITY
23F5	21B825	LD	HL,25B8H	RETURN THROUGH
23F8	E5	PUSH	HL	RELATIONAL EVALUATE
23F9	E7	RST	20H	CHECK TYPE OF ACCUM
23FA	C29523	JP	NZ,2395H	JUMP IF NOT STRING
23FD	2A2141	LD	HL,(4121H)	GET VALUE OF ACCUM
2400	E5	PUSH	HL	SAVE FIRST ARGUMENT
2401	018C25	LD	BC,258CH	EVALUATE SECOND
2404	18C7	JR	23CDH	ARGUMENT & PROCESS
2406	C1	POP	BC	GET TYPE OF FIRST ARGUMENT
2407	79	LD	A,C	SAVE OPERATION #
2408	32B040	LD	(40B0H),A	
240B	78	LD	A,B	GET FIRST TYPE
240C	FE08	CP	08H	DOUBLE PRECISION?
240E	2828	JR	Z,2438H	JUMP IF SO
2410	3AAF40	LD	A,(40AFH)	GET SECOND TYPE
2413	FE08	CP	08H	DOUBLE PRECISION
2415	CA6024	JP	Z,2460H	JUMP IF SO
2418	57	LD	D,A	SAVE SECOND TYPE
2419	78	LD	A,B	GET FIRST TYPE
241A	FE04	CP	04H	SINGLE PRECISION?
241C	CA7224	JP	Z,2472H	JUMP IF SO
241F	7A	LD	A,D	GET SECOND TYPE
2420	FE03	CP	03H	STRING?
2422	CAF60A	JP	Z,0AF6H	TM ERROR IF SO
2425	D27C24	JP	NC,247CH	JUMP IF S.P.
2428	21BF18	LD	HL,18BFH	HL => INTEGER OPERATIONS TABLE
242B	0600	LD	B,00H	HL =>
242D	09	ADD	HL,BC	OPERATION
242E	09	ADD	HL,BC	ADDRESS
242F	4E	LD	C,(HL)	BC = OPERATION
2430	23	INC	HL	ADDRESS

2431	46	LD	B, (HL)	
2432	D1	POP	DE	GET FIRST ARGUMENT
2433	2A2141	LD	HL, (4121H)	GET SECOND ARGUMENT
2436	C5	PUSH	BC	SAVE OPERATION
2437	C9	RET		ADDRESS & GO TO IT
2438	CDDB0A	CALL	0ADBH	CONVERT SECOND ARGUMENT TO D.P.
243B	CDFC09	CALL	09FCH	MOVE IT TO ACCUM 2
243E	E1	POP	HL	LOAD
243F	221F41	LD	(411FH), HL	FIRST ARGUMENT
2442	E1	POP	HL	INTO ACCUM
2443	221D41	LD	(411DH), HL	FROM STACK
2446	C1	POP	BC	
2447	D1	POP	DE	
2448	CDB409	CALL	09B4H	
244B	CDDB0A	CALL	0ADBH	CONVERT ACCUM TO D.P.
244E	21AB18	LD	HL, 18ABH	HL => D.P. OPERATOR TABLE
2451	3AB040	LD	A, (40B0H)	GET OPERATOR #
2454	07	RLCA		TIMES 2
2455	C5	PUSH	BC	SAVE BC
2456	4F	LD	C, A	HL => OPERATOR
2457	0600	LD	B, 00H	ADDRESS
2459	09	ADD	HL, BC	
245A	C1	POP	BC	RESTORE BC
245B	7E	LD	A, (HL)	HL = OPERATOR
245C	23	INC	HL	ADDRESS
245D	66	LD	H, (HL)	
245E	6F	LD	L, A	
245F	E9	JP	(HL)	EXECUTE OPERATOR
2460	C5	PUSH	BC	SAVE SECOND ARGUMENT TYPE
2461	CDFC09	CALL	09FCH	ACCUM 2 = FIRST ARGUMENT
2464	F1	POP	AF	GET SECOND ARGUMENT TYPE
2465	32AF40	LD	(40AFH), A	SET TYPE OF ACCUM
2468	FE04	CP	04H	SINGLE PRECISION?
246A	28DA	JR	Z, 2446H	GET SECOND ARGUMENT & CONVERT IF SO
246C	E1	POP	HL	SAVE INTEGER ARGUMENT
246D	222141	LD	(4121H), HL	IN ACCUM
2470	18D9	JR	244BH	CONVERT & EXECUTE OPERATOR
2472	CDB10A	CALL	0AB1H	CONVERT FIRST ARGUMENT TO S.P.
2475	C1	POP	BC	GET SECOND
2476	D1	POP	DE	ARGUMENT
2477	21B518	LD	HL, 18B5H	HL => S.P.O. OPERATOR TABLE
247A	18D5	JR	2451H	EXECUTE OPERATOR
247C	E1	POP	HL	GET FIRST ARGUMENT
247D	CDA409	CALL	09A4H	SAVE SECOND ARGUMENT
2480	CDCF0A	CALL	0ACFH	ACCUM = CSNG (FIRST ARGUMENT)
2483	CDBF09	CALL	09BFH	BCDE = SECOND ARGUMENT
2486	E1	POP	HL	ACCUM =
2487	222341	LD	(4123H), HL	FIRST
248A	E1	POP	HL	ARGUMENT
248B	222141	LD	(4121H), HL	
248E	18E7	JR	2477H	EXECUTE OPERATION

INTEGER DIVISION ROUTINE

2490	E5	PUSH	HL	SAVE SECOND ARGUMENT
2491	EB	EX	DE, HL	ACCUM=
2492	CDCF0A	CALL	0ACFH	CSNG (FIRST ARGUMENT)
2495	E1	POP	HL	GET SECOND ARGUMENT

2496	CDA409	CALL	09A4H	SAVE FIRST ARGUMENT ON STACK
2499	CDCF0A	CALL	0ACFH	ACCUM = CSNG (SECOND ARGUMENT)
249C	C3A008	JP	08A0H	ACCUM = FIRST/SECOND
249F	D7	RST	10H	GET NEXT TOKEN
24A0	1E28	LD	E,28H	MO ERROR
24A2	CAA219	JP	Z,19A2H	IF END OF LINE
24A5	DA6C0E	JP	C,0E6CH	DECODE # IF NUMERIC
24A8	CD3D1E	CALL	1E3DH	A-Z?
24AB	D24025	JP	NC,2540H	JUMP IF A VARIABLE
24AE	FECF	CP	0CDH	+ TOKEN?
24B0	28ED	JR	Z,249FH	JUMP IF SO
24B2	FE2E	CP	2EH	DECIMAL POINT?
24B4	CA6C0E	JP	Z,0E6CH	DECODE # IF SO
24B7	FECE	CP	0CEH	- TOKEN?
24B9	CA3225	JP	Z,2532H	JUMP IF SO
24BC	FE22	CP	22H	DOUBLE QUOTES?
24BE	CA6628	JP	Z,2866H	JUMP IF SO
24C1	FECB	CP	0CBH	NOT TOKEN?
24C3	CAC425	JP	Z,25C4H	INVERT # IF SO
24C6	FE26	CP	26H	AMPERSAND?
24C8	CA9441	JP	Z,4194H	DECODE HEX NUMBER IF SO
24CB	FEC3	CP	0C3H	ERR TOKEN?
24CD	200A	JR	NZ,24D9H	JUMP IF NOT

ERR

24CF	D7	RST	10H	GET NEXT CHARACTER
24D0	3A9A40	LD	A,(409AH)	GET ERR CODE
24D3	E5	PUSH	HL	SAVE POSITION
24D4	CDF827	CALL	27F8H	RESULT TO INTEGER FORM
24D7	E1	POP	HL	RESTORE POSITION
24D8	C9	RET		DONE

ERL

24D9	FEC2	CP	0C2H	ERL TOKEN?
24DB	200A	JR	NZ,24E7H	JUMP IF NOT
24DD	D7	RST	10H	GET NEXT CHARACTER
24DE	E5	PUSH	HL	SAVE POSITION
24DF	2AEA40	LD	HL,(40EAH)	GET ERL
24E2	CD660C	CALL	0C66H	RETURN AS S.P. NUMBER
24E5	E1	POP	HL	RESTORE POSITION
24E6	C9	RET		DONE

VARPTR

24E7	FEC0	CP	0C0H	VARPTR TOKEN?
24E9	2014	JR	NZ,24FFH	JUMP IF NOT
24EB	D7	RST	10H	GET NEXT CHARACTER
24EC	CF	RST	08H	SYNTAX ERROR IF
24ED	28	" ("		NEXT CHARACTER NOT
24EE	CD0D26	CALL	260DH	GET VARIABLE POINTER
24F1	CF	RST	08H	SYNTAX ERROR IF
24F2	29	") "		NEXT CHARACTER NOT °) °
24F3	E5	PUSH	HL	SAVE POSITION
24F4	EB	EX	DE,HL	HL => VARIABLE
24F5	7C	LD	A,H	VARIABLE NOT FOUND?
24F6	B5	OR	L	
24F7	CA4A1E	JP	Z,1E4AH	FC ERROR IF SO
24FA	CD9A0A	CALL	0A9AH	RETURN POINTER AS INTEGER

24FD	E1	POP	HL	RESTORE POSITION
24FE	C9	RET		DONE
24FF	FEC1	CP	0C1H	USR TOKEN?
2501	CAFE27	JP	Z,27FEH	JUMP IF SO
2504	FEC5	CP	0C5H	INSTR TOKEN?
2506	CA9D41	JP	Z,419DH	JUMP IF SO
2509	FEC8	CP	0C8H	MEM TOKEN?
250B	CAC927	JP	Z,27C9H	JUMP IF SO
250E	FEC7	CP	0C7H	TIME\$ TOKEN?
2510	CA7641	JP	Z,4176H	JUMP IF SO
2513	FEC6	CP	0C6H	POINT TOKEN?
2515	CA3201	JP	Z,0132H	JUMP IF SO
2518	FEC9	CP	0C9H	INKEY\$ TOKEN?
251A	CA9D01	JP	Z,019DH	JUMP IF SO
251D	FEC4	CP	0C4H	STRING\$ TOKEN?
251F	CA2F2A	JP	Z,2A2FH	JUMP IF SO
2522	FEBE	CP	0BEH	FN TOKEN?
2524	CA5541	JP	Z,4155H	JUMP IF SO
2527	D6D7	SUB	0D7H	EMBEDDED FUNCTIONS?
2529	D24E25	JP	NC,254EH	JUMP IF SO
252C	CD3523	CALL	2335H	EVALUATE NESTED EXPRESSIONS
252F	CF	RST	08H	SYNTAX ERROR IF
2530	29) "		NEXT CHARACTER NOT ") "
2531	C9	RET		DONE
2532	167D	LD	D,7DH	PRIORITY = 7DH
2534	CD3A23	CALL	233AH	EVALUATE EXPRESSION ON RIGHT
2537	2AF340	LD	HL,(40F3H)	GET POSITION
253A	E5	PUSH	HL	SAVE POSITION
253B	CD7B09	CALL	097BH	INVERT SIGN OF ACCUM
253E	E1	POP	HL	RESTORE POSITION
253F	C9	RET		DONE
2540	CD0D26	CALL	260DH	GET VARIABLE POINTER
2543	E5	PUSH	HL	SAVE POSITION
2544	EB	EX	DE,HL	DE => VARIABLE
2545	222141	LD	(4121H),HL	ACCUM => VARIABLE
2548	E7	RST	20H	CHECK TYPE OF VARIABLE
2549	C4F709	CALL	NZ,09F7H	MOVE DATA TO ACCUM IF NOT STRING
254C	E1	POP	HL	RESTORE POSITION
254D	C9	RET		DONE
254E	0600	LD	B,00H	BC =
2550	07	RLCA		OFFSET
2551	4F	LD	C,A	IN FUNCTION TABLE
2552	C5	PUSH	BC	SAVE OFFSET
2553	D7	RST	10H	GET NEXT CHARACTER.
2554	79	LD	A,C	GET OFFSET
2555	FE41	CP	41H	SINGLE PARAMETER FUNCTION?
2557	3816	JR	C,256FH	JUMP IF SO
2559	CD3523	CALL	2335H	EVALUATE FIRST EXPRESSION
255C	CF	RST	08H	SYNTAX ERROR IF
255D	2C	","		NEXT CHARACTER NOT COMMA
255E	CD40A	CALL	0AF4H	TM ERROR IF NOT STRING
2561	EB	EX	DE,HL	SAVE POSITION
2562	2A2141	LD	HL,(4121H)	GET STRING POINTER
2565	E3	EX	(SP),HL	SAVE STRING POINTER
2566	E5	PUSH	HL	SAVE OFFSET IN TABLE
2567	EB	EX	DE,HL	RESTORE POSITION
2568	CD1C2B	CALL	2B1CH	EVALUATE SECOND PARAM

256B	EB	EX	DE,HL	SAVE SECOND PARAM
256C	E3	EX	(SP),HL	AND GET OFFSET IN TABLE
256D	1814	JR	2583H	CONTINUE
256F	CD2C25	CALL	252CH	EVALUATE EXPRESSION
2572	E3	EX	(SP),HL	GET OFFSET. SAVE POSITION
2573	7D	LD	A,L	PUT OFFSET IN A
2574	FE0C	CP	0CH	INTEGER TYPE?
2576	3807	JR	C,257FH	SKIP IF SO
2578	FE1B	CP	1BH	INTEGER TYPE?
257A	E5	PUSH	HL	SAVE POSITION
257B	DCB10A	CALL	C,0AB1H	MAKE TYPE S.P.
257E	E1	POP	HL	RESTORE POSITION
257F	113E25	LD	DE,253EH	SET RETURN ADDRESS
2582	D5	PUSH	DE	TO 253EH
2583	010816	LD	BC,1608H	BC => FUNCTION ADDRESS
2586	09	ADD	HL,BC	HL => FUNCTION ADDRESS
2587	4E	LD	C,(HL)	GET
2588	23	INC	HL	FUNCTION
2589	66	LD	H,(HL)	ADDRESS
258A	69	LD	L,C	
258B	E9	JP	(HL)	EXECUTE FUNCTION
258C	CDD729	CALL	29D7H	GET PARAM #1
258F	7E	LD	A,(HL)	A = LENGTH #1
2590	23	INC	HL	
2591	4E	LD	C,(HL)	BC => STRING #1
2592	23	INC	HL	
2593	46	LD	B,(HL)	
2594	D1	POP	DE	CLEAR STACK
2595	C5	PUSH	BC	SAVE POINTER #1
2596	F5	PUSH	AF	SAVE LENGTH #1
2597	CDDE29	CALL	29DEH	GET PARAM #2
259A	D1	POP	DE	D = LENGTH #1
259B	5E	LD	E,(HL)	E = LENGTH #2
259C	23	INC	HL	
259D	4E	LD	C,(HL)	BC => STRING #2
259E	23	INC	HL	
259F	46	LD	B,(HL)	
25A0	E1	POP	HL	HL => STRING #1
25A1	7B	LD	A,E	BOTH STRINGS
25A2	B2	OR	D	EQUAL LENGTH?
25A3	C8	RET	Z	RETURN MATCH IF SO
25A4	7A	LD	A,D	STRING #1
25A5	D601	SUB	01H	NULL?
25A7	D8	RET	C	RETURN #2 GREATER IF SO
25A8	AF	XOR	A	STRING #2
25A9	BB	CP	E	NULL?
25AA	3C	INC	A	
25AB	D0	RET	NC	RETURN #1 GREATER IF SO
25AC	15	DEC	D	ONE LESS
25AD	1D	DEC	E	CHARACTER TO CHECK
25AE	0A	LD	A,(BC)	COMPARE CHARACTERS
25AF	BE	CP	(HL)	
25B0	23	INC	HL	POINT TO
25B1	03	INC	BC	NEXT CHARACTERS
25B2	28ED	JR	Z,25A1H	LOOP IF MATCH
25B4	3F	CCF		INVERT COMPARE RESULT
25B5	C36009	JP	0960H	FINISH UP & RETURN

```

25B8 3C      INC      A          RELATIONAL
25B9 8F      ADC      A,A        COMPARE
25BA C1      POP      BC          RESULT:
25BB A0      AND      B          A = 0 FALSE
25BC C6FF    ADD      A,0FFH    A = -1 TRUE
25BE 9F      SBC      A,A
25BF CD8D09  CALL    098DH
25C2 1812    JR       25D6H    CONTINUE
    
```

NOT

```

25C4 165A    LD       D,5AH    SET PRIORITY
25C6 CD3A23  CALL    233AH    EVALUATE ARGUMENT
25C9 CD7F0A  CALL    0A7FH    HL = ACCUM
25CC 7D      LD       A,L      L
25CD 2F      CPL
25CE 6F      LD       L,A      NOT (L)
25CF 7C      LD       A,H      H
25D0 2F      CPL
25D1 67      LD       H,A      NOT (H)
25D2 222141 LD      (4121H),HL ACCUM = HL
25D5 C1      POP      BC      CLEAR STACK
25D6 C34623  JP       2346H    CONTINUE
    
```

RST 20H ROUTINE

```

25D9 3AAF40  LD       A,(40AFH) GET TYPE FLAG
25DC FE08    CP       08H      DOUBLE PRECISION?
25DE 3005    JR       NC,25E5H SKIP IF SO
25E0 D603    SUB     03H      ADJUST FLAG
25E2 B7      OR       A          SET STATUS
25E3 37      SCF
25E4 C9      RET
25E5 D603    SUB     03H      ADJUST FLAG
25E7 B7      OR       A          SET STATUS
25E8 C9      RET
25E9 C5      PUSH    BC          SAVE PRIORITY
25EA CD7F0A  CALL    0A7FH    HL = SECOND ARGUMENT
25ED F1      POP     AF          GET PRIORITY
25EE D1      POP     DE          GET FIRST ARGUMENT
25EF 01FA27 LD      BC,27FAH  SET RETURN
25F2 C5      PUSH    BC          THROUGH 27FAH
25F3 FE46    CP       46H      OR?
25F5 2006    JR       NZ,25FDH JUMP IF NOT
25F7 7B      LD      A,E      L
25F8 B5      OR      L
25F9 6F      LD      L,A      L OR E
25FA 7C      LD      A,H      A =
25FB B2      OR      D          H OR D
25FC C9      RET
25FD C9      RET
    
```

NET RESULT

TYPE	Z/NZ	FLAG		
		P/M	C/NC	PO/PE
INTEGER	NZ	M****	C	PE
STRING	Z****	P	C	PE
S.P.	NZ	P	C	PO****
D.P.	NZ	P	NC****	PE

```

25E9 C5      PUSH    BC          SAVE PRIORITY
25EA CD7F0A  CALL    0A7FH    HL = SECOND ARGUMENT
25ED F1      POP     AF          GET PRIORITY
25EE D1      POP     DE          GET FIRST ARGUMENT
25EF 01FA27 LD      BC,27FAH  SET RETURN
25F2 C5      PUSH    BC          THROUGH 27FAH
25F3 FE46    CP       46H      OR?
25F5 2006    JR       NZ,25FDH JUMP IF NOT
25F7 7B      LD      A,E      L
25F8 B5      OR      L
25F9 6F      LD      L,A      L OR E
25FA 7C      LD      A,H      A =
25FB B2      OR      D          H OR D
25FC C9      RET
25FD C9      RET
    
```

25FD 7B	LD	A, E	L
25FE A5	AND	L	=
25FF 6F	LD	L, A	LAND E
2600 7C	LD	A, H	A=
2601 A2	AND	D	HAND D
2602 C9	RET		FINISH UP & RETURN

DIM & LOCATE/CREATE VARIABLE

2603 2B	DEC	HL	BACK UP
2604 D7	RST	10H	GET NEXT CHARACTER
2605 C8	RET	Z	EXIT IF DONE
2606 CF	RST	08H	SYNTAX ERROR IF
2607 2C	" , "		NEXT CHARACTER NOT COMMA
2608 010326	LD	BC, 2603H	RETURN FOR MORE
260B C5	PUSH	BC	WHEN DONE WITH ONE
260C F6AF	OR	0AFH	SET CREATE. MASK TO 260EH
260D AF	XOR	A	SET LOCATE
260E 32AE40	LD	(40AEH), A	SAVE LOCATE/CREATE FLAG
2611 46	LD	B, (HL)	GET FIRST CHARACTER. OF NAME
2612 CD3D1E	CALL	1E3DH	A - Z?
2615 DA9719	JP	C, 1997H	SYNTAX ERROR IF NOT
2618 AF	XOR	A	SET SECOND
2619 4F	LD	C, A	CHARACTER AS NULL
261A D7	RST	10H	GET NEXT CHARACTER
261B 3805	JR	C, 2622H	GOOD IF NUMERIC JUMP
261D CD3D1E	CALL	1E3DH	A - Z?
2620 3809	JR	C, 262BH	NO GOOD IF NOT. JUMP
2622 4F	LD	C, A	SAVE SECOND CHARACTER
2623 D7	RST	10H	GET NEXT CHARACTER
2624 38FD	JR	C, 2623H	GET MORE IF NUMERIC
2626 CD3D1E	CALL	1E3DH	A - Z?
2629 30F8	JR	NC, 2623H	GET MORE IF SO
262B 115226	LD	DE, 2652H	RETURN
262E D5	PUSH	DE	THROUGH 2652H
262F 1602	LD	D, 02H	TYPE = INTEGER
2631 FE25	CP	25H	PERCENT SIGN?
2633 C8	RET	Z	CONTINUE IF SO
2634 14	INC	D	TYPE = STRING?
2635 FE24	CP	24H	DOLLAR'S SIGN
2637 C8	RET	Z	CONTINUE IF SO
2638 14	INC	D	TYPE S.P.
2639 FE21	CP	21H	EXCLAMATION POINT
263B C8	RET	Z	CONTINUE IF SO
263C 1608	LD	D, 08H	TYPE D.P.
263E FE23	CP	23H	POUND'S SIGN?
2640 C8	RET	Z	CONTINUE IF SO.
2641 78	LD	A, B	GET FIRST CHARACTER
2642 D641	SUB	41H	DE
2644 E67F	AND	7FH	=
2646 5F	LD	E, A	OFFSET IN
2647 1600	LD	D, 00H	TYPE TABLE
2649 E5	PUSH	HL	SAVE POSITION
264A 210141	LD	HL, 4101H	HL => TYPE TABLE
264D 19	ADD	HL, DE	HL => VARIABLE TYPE
264E 56	LD	D, (HL)	GET DEFAULT TYPE
264F E1	POP	HL	RESTORE POSITION
2650 2B	DEC	HL	BACK UP

2651	C9	RET		CONTINUE.
2652	7A	LD	A,D	GET TYPE OF VARIABLE
2653	32AF40	LD	(40AFH),A	SAVE VARIABLE TYPE
2656	D7	RST	10H	GET NEXT CHARACTER
2657	3ADC40	LD	A,(40DCH)	FOR IN
265A	B7	OR	A	PROGRESS?
265B	C26426	JP	NZ,2664H	SKIP IF SO
265E	7E	LD	A,(HL)	GET CHARACTER (SUBSCRIPTED
265F	D628	SUB	28H	"(" (VARIABLE?
2661	CAE926	JP	Z,26E9H	JUMP IF SO
2664	AF	XOR	A	CLEAR FOR
2665	32DC40	LD	(40DCH),A	FLAG
2668	E5	PUSH	HL	SAVE POSITION
2669	D5	PUSH	DE	SAVE TYPE
266A	2AF940	LD	HL,(40F9H)	DE => SIMPLE VARIABLE TABLE
266D	EB	EX	DE,HL	
266E	2AFB40	LD	HL,(40FBH)	HL => ARRAY TABLE
2671	DF	RST	18H	ARE THEY SAME?
2672	E1	POP	HL	GET TYPE
2673	2819	JR	Z,268EH	JUMP IF SO
2675	1A	LD	A,(DE)	GET TYPE SF TABLE ENTRY
2676	6F	LD	L,A	SAVE IN L
2677	BC	CP	H	COMPARE WITH REQUESTED TYPE
2678	13	INC	DE	POINT TO ENTRY NAME
2679	200B	JR	NZ,2686H	JUMP IF WRONG TYPE
267B	1A	LD	A,(DE)	GET FIRST CHARACTER OF ENTRY NAME
267C	B9	CP	C	SAME AS GIVEN?
267D	2007	JR	NZ,2686H	JUMP IF NOT
267F	13	INC	DE	POINT TO SECOND CHARACTER
2680	1A	LD	A,(DE)	GET SECOND CHARACTER OF ENTRY NAME
2681	B8	CP	B	SAME AS GIVEN?
2682	CACC26	JP	Z,26CCH	JUMP IF SO
2685	3E			MASK TO 2687H
2686	13	INC	DE	MOVE UP
2687	13	INC	DE	TO DATE
2688	E5	PUSH	HL	SAVE TYPE
2689	2600	LD	H,00H	MOVE PAST
268B	19	ADD	HL,DE	DATA
268C	18DF	JR	266DH	CONTINUE SEARCH
268E	7C	LD	A,H	GET TYPE
268F	E1	POP	HL	GET POSITION
2690	E3	EX	(SP),HL	SAVE POSITION & GET RETURN
2691	F5	PUSH	AF	SAVE TYPE
2692	D5	PUSH	DE	SAVE POSITION IN TABLE
2693	11F124	LD	DE,24F1H	CALLED BY
2696	DF	RST	18H	VARPTR?
2697	2836	JR	Z,26CFH	EXIT IF SO (NOT FOUND)
2699	114325	LD	DE,2543H	CALLED BY
269C	DF	RST	18H	EXPRESSION EVALUATOR?
269D	D1	POP	DE	GET POSITION IN TABLE
269E	2835	JR	Z,26D5H	JUMP IF SO
26A0	F1	POP	AF	GET TYPE
26A1	E3	EX	(SP),HL	SAVE RETURN & GET POSITION
26A2	E5	PUSH	HL	SAVE POSITION
26A3	C5	PUSH	BC	SAVE VARIABLE TO FIND
26A4	4F	LD	C,A	BC = TYPE
26A5	0600	LD	B,00H	

26A7	C5	PUSH	BC	SAVE TYPE
26A8	03	INC	BC	ADJUST
26A9	03	INC	BC	TYPE
26AA	03	INC	BC	
26AB	2AFD40	LD	HL, (40FDH)	HL => END OF ARRAYS
26AE	E5	PUSH	HL	SAVE END OF SOURCE
26AF	09	ADD	HL,BC	HL => NEW END OF ARRAY
26B0	C1	POP	BC	BC => END OF ARRAYS
26B1	E5	PUSH	HL	SAVE NEW END OF ARRAYS
26B2	CD5519	CALL	1955H	MOVE ARRAYS UP
26B5	E1	POP	HL	GET END OF ARRAYS
26B6	22FD40	LD	(40FDH),HL	SAVE END OF ARRAYS
26B9	60	LD	H,B	HL => END OF
26BA	69	LD	L,C	VARIABLES
26BB	22FB40	LD	(40FBH),HL	SAVE END OF VARIABLES
26BE	2B	DEC	HL	CLEAR TO OLD
26BF	3600	LD	(HL),00H	END OF VARIABLES
26C1	DF	RST	18H	DONE YET?
26C2	20FA	JR	NZ,26BEH	LOOP IF NOT
26C4	D1	POP	DE	GET TYPE
26C5	73	LD	(HL),E	SAVE TYPE IN NEW ENTRY
26C6	23	INC	HL	NEXT LOCATION
26C7	D1	POP	DE	GET VARIABLE NAME
26C8	73	LD	(HL),E	SAVE SECOND CHARACTER.
26C9	23	INC	HL	NEXT LOCATION
26CA	72	LD	(HL),D	SAVE FIRST CHARACTER
26CB	EB	EX	DE,HL	PUT LOCATION IN DE
26CC	13	INC	DE	DE => VARIABLE DATA
26CD	E1	POP	HL	RESTORE POSITION
26CE	C9	RET		DONE
26CF	57	LD	D,A	DE = 0 = VARIABLE
26D0	5F	LD	E,A	NOT FOUND
26D1	F1	POP	AF	CLEAR
26D2	F1	POP	AF	STACK
26D3	E3	EX	(SP),HL	GET POSITION & SAVE RETURN
26D4	C9	RET		RETURN
26D5	322441	LD	(4124H),A	CLEAR EXPONENT OF ACCUM
26D8	C1	POP	BC	CLEAR STACK
26D9	67	LD	H,A	CLEAR VALUE OF ACCUM
26DA	6F	LD	L,A	
26DB	222141	LD	(4121H),HL	
26DE	E7	RST	20H	CHECK TYPE OF ACCUM
26DF	2006	JR	NZ,26E7H	SKIP IF NOT STRING
26E1	212819	LD	HL,1928H	GET ACCUM
26E4	222141	LD	(4121H),HL	AS NULL STRING
26E7	E1	POP	HL	RESTORE POSITION
26E8	C9	RET		CONTINUE

SEARCH FOR ARRAY VARIABLE

26E9	E5	PUSH	HL	SAVE POSITION
26EA	2AAE40	LD	HL, (40AEH)	GET LOCATION/CREATE FLAG
26ED	E3	EX	(SP),HL	RESTORE POSITION. SAVE FLAG
26EE	57	LD	D,A	ZERO SUBSCRIPTIONS SO FAR
26EF	D5	PUSH	DE	SAVE # OF SUBSCRIPTS
26F0	C5	PUSH	BC	SAVE VARIABLE NAME
26F1	CD451E	CALL	1E45H	EVALUATE SUBSCRIPT
26F4	C1	POP	BC	RESTORE VARIABLE NAME

26F5	F1	POP	AF	RESTORE # SUBSCRIPTS
26F6	EB	EX	DE,HL	HL = SUBSCRIPT
26F7	E3	EX	(SP),HL	GET FLAG. SAVE SUBSCRIPT
26F8	E5	PUSH	HL	SAVE FLAG
26F9	EB	EX	DE,HL	HL = POSITION IN TEXT
26FA	3C	INC	A	INCREMENT #
26FB	57	LD	D,A	OF SUBSCRIPTS
26FC	7E	LD	A,(HL)	GET NEXT CHARACTER
26FD	FE2C	CP	2CH	COMMA?
26FF	28EE	JR	Z,26EFH	LOOP IF MORE SUBSCRIPTS
2701	CF	RST	08H	SYNTAX ERROR IF
2702	29)"		NEXT CHARACTER NOT ")"
2703	22F340	LD	(40F3H),HL	SAVE POSITION
2706	E1	POP	HL	RESTORE FLAG
2707	22AE40	LD	(40AEH),HL	SAVE FLAG
270A	D5	PUSH	DE	SAVE # OF SUBSCRIPTS
270B	2AFB40	LD	HL,(40FBH)	HL => ARRAY TABLE
270E	3E			MASK TO 2710H
270F	19	ADD	HL,DE	MOVE PAST ARRAY
2710	EB	EX	DE,HL	DE
2711	2AFD40	LD	HL,(40FDH)	=>
2714	EB	EX	DE,HL	FREE MEMORY
2715	DF	RST	18H	DONE CHECKING ARRAYS?
2716	3AAF40	LD	A,(40AFH)	GET TYPE OF ARRAY
2719	2827	JR	Z,2742H	JUMP IF NOT FOUND
271B	BE	CP	(HL)	ARRAY PROPER TYPE?
271C	23	INC	HL	NEXT BYTE
271D	2008	JR	NZ,2727H	SKIP IF NOT PROPER TYPE
271F	7E	LD	A,(HL)	GET FIRST CHARACTER OF ENTRY
2720	B9	CP	C	COMPARE WITH GIVEN
2721	23	INC	HL	POINT TO NEXT BYTE
2722	2004	JR	NZ,2728H	SKIP IF NO MATCH
2724	7E	LD	A,(HL)	GET SECOND CHARACTER
2725	B8	CP	B	COMPARE WITH GIVEN
2726	3E			MASK TO 2728H
2727	23	INC	HL	POINT TO NEXT BYTE
2728	23	INC	HL	POINT TO NEXT BYTE
2729	5E	LD	E,(HL)	DE = SIZE OF
272A	23	INC	HL	ARRAY DATA SPACE
272B	56	LD	D,(HL)	
272C	23	INC	HL	
272D	20E0	JR	NZ,270FH	LOOP IF NOT CORRECT NAME
272F	3AAE40	LD	A,(40AEH)	CREATE
2732	B7	OR	A	NEW VARIABLE?
2733	1E12	LD	E,12H	POSSIBLE DD ERROR
2735	C2A219	JP	NZ,19A2H	ERROR IF FOUND ON CREATE
2738	F1	POP	AF	GET # SUBSCRIPTS
2739	96	SUB	(HL)	COMPARE WITH TABLE ENTRY
273A	CA9527	JP	Z,2795H	JUMP IF MATCH
273D	1E10	LD	E,10H	BS ERROR
273F	C3A219	JP	19A2H	GO TO ERROR ROUTINE
2742	77	LD	(HL),A	SAVE ARRAY TYPE
2743	23	INC	HL	NEXT BYTE
2744	5F	LD	E,A	DE = TYPE = SIZE
2745	1600	LD	D,00H	
2747	F1	POP	AF	GET # SUBSCRIPTS
2748	71	LD	(HL),C	SAVE

2749	23	INC	HL	ARRAY
274A	70	LD	(HL), B	NAME
274B	23	INC	HL	
274C	4F	LD	C, A	SAVE # SUBSCRIPTS IN C
274D	CD6319	CALL	1963H	REQUEST MEMORY SPACE
2750	23	INC	HL	SKIP OVER
2751	23	INC	HL	SIZE WORD
2752	22D840	LD	(40D8H), HL	SAVE POSITION OF SIZE OF WORD
2755	71	LD	(HL), C	SAVE # OF SUBSCRIPTS
2756	23	INC	HL	MOVE TO NEXT BYTE
2757	3AAE40	LD	A, (40AEH)	SET CARRY IF
275A	17	RLA		CREATING ARRAY
275B	79	LD	A, C	GET # OF SUBSCRIPTS
275C	010B00	LD	BC, 000BH	DEFAULT DIMENSION
275F	3002	JR	NC, 2763H	SKIP IF LOCATING
2761	C1	POP	BC	GET DIMENSION
2762	03	INC	BC	INCREASE BY 1
2763	71	LD	(HL), C	SAVE
2764	23	INC	HL	DIMENSION
2765	70	LD	(HL), B	IN TABLE
2766	23	INC	HL	
2767	F5	PUSH	AF	SAVE FLAG
2768	CDAA0B	CALL	0BAAH	SIZE = SIZE * DIMENSION
276B	F1	POP	AF	RESTORE FLAG
276C	3D	DEC	A	ONE LESS DIMENSION TO DO
276D	20ED	JR	NZ, 275CH	LOOP UNTIL DONE
276F	F5	PUSH	AF	SAVE FLAG
2770	42	LD	B, D	BC = SIZE OF
2771	4B	LD	C, E	DATA SPACE
2772	EB	EX	DE, HL	
2773	19	ADD	HL, DE	HL => NEXT AVAILABLE SPACE
2774	38C7	JR	C, 273DH	BS ERROR IF OVERFLOW
2776	CD6C19	CALL	196CH	CHECK MEMORY SPACE
2779	22FD40	LD	(40FDH), HL	SAVE NEW FREE MEMORY POINTER
277C	2B	DEC	HL	CLEAR ARRAY
277D	3600	LD	(HL), 00H	TO ZEROS
277F	DF	RST	18H	
2780	20FA	JR	NZ, 277CH	
2782	03	INC	BC	INCREMENT SIZE
2783	57	LD	D, A	D = 0
2784	2AD840	LD	HL, (40D8H)	POSITION TO SIZE WORD
2787	5E	LD	E, (HL)	DE = # SUBSCRIPTS
2788	EB	EX	DE, HL	DE
2789	29	ADD	HL, HL	=
278A	09	ADD	HL, BC	# SUBSCRIPTS * 2
278B	EB	EX	DE, HL	+ DATA SPACE SIZE
278C	2B	DEC	HL	BACKUP TO
278D	2B	DEC	HL	START OF SIZE WORD
278E	73	LD	(HL), E	SAVE
278F	23	INC	HL	SIZE
2790	72	LD	(HL), D	OF
2791	23	INC	HL	DATA SPACE
2792	F1	POP	AF	GET FLAG
2793	3830	JR	C, 27C5H	JUMP IF CREATING ARRAY
2795	47	LD	B, A	OFFSET = 0
2796	4F	LD	C, A	
2797	7E	LD	A, (HL)	GET # OF SUBSCRIPTS

2798	23	INC	HL	POINT TO FIRST SUBSCRIPT
2799	16			MASK TO 279BH
279A	E1	POP	HL	GET POSITION IN SUBSCRIPT
279B	5E	LD	E, (HL)	DE
279C	23	INC	HL	=
279D	56	LD	D, (HL)	MAX SUBSCRIPT
279E	23	INC	HL	VALUE + 1
279F	E3	EX	(SP), HL	SAVE POSITION HL = REQUESTED VALUE
27A0	F5	PUSH	AF	SAVE # SUBSCRIPTS LEFT
27A1	DF	RST	18H	SUBSCRIPT TOO LARGE?
27A2	D23D27	JP	NC, 273DH	BS ERROR IF SO
27A5	CDAA0B	CALL	0BAAH	OFFSET = OFFSET * DIMENSION
27A8	19	ADD	HL, DE	OFFSET = OFFSET + SUBSCRIPT
27A9	F1	POP	AF	RESTORE # SUBSCRIPTS LEFT
27AA	3D	DEC	A	ONE LESS TO PROCESS
27AB	44	LD	B, H	SAVE
27AC	4D	LD	C, L	OFFSET
27AD	20EB	JR	NZ, 279AH	LOOP UNTIL ALL PROCESSED
27AF	3AAF40	LD	A, (40AFH)	GET SIZE OF DATA ELEMENTS
27B2	44	LD	B, H	SAVE
27B3	4D	LD	C, L	OFFSET
27B4	29	ADD	HL, HL	OFFSET = OFFSET * 2
27B5	D604	SUB	04H	ADJUST TYPE
27B7	3804	JR	C, 27BDH	SKIP IF INTEGER OR STRING
27B9	29	ADD	HL, HL	OFFSET = OFFSET*2
27BA	2806	JR	Z, 27C2H	JUMP IF S.P.
27BC	29	ADD	HL, HL	OFFSET = OFFSET*2
27BD	B7	OR	A	CHECK TYPE
27BE	E2C227	JP	PO, 27C2H	SKIP IF D.P. OR INTEGER
27C1	09	ADD	HL, BC	OFFSET = OFFSET * 2 + OFFSET
27C2	C1	POP	BC	GET ARRAY START
27C3	09	ADD	HL, BC	HL => ELEMENT DATA
27C4	EB	EX	DE, HL	DE => ELEMENT DATA
27C5	2AF340	LD	HL, (40F3H)	GET POSITION
27C8	C9	RET		DONE

MEM

27C9	AF	XOR	A	CLEAR STATUS
27CA	E5	PUSH	HL	SAVE POSITION
27CB	32AF40	LD	(40AFH), A	SET TYPE NUMERIC
27CE	CDD427	CALL	27D4H	CALL FRE(0)
27D1	E1	POP	HL	RESTORE POSITION
27D2	D7	RST	10H	GET NEXT CHARACTER
27D3	C9	RET		DONE

FRE

27D4	2AFD40	LD	HL, (40FDH)	
27D7	EB	EX	DE, HL	DE => START OF FREE MEMORY
27D8	210000	LD	HL, 0000H	HL => TOP OF STACK
27DB	39	ADD	HL, SP	
27DC	E7	RST	20H	CHECK ARGUMENT TYPE
27DD	200D	JR	NZ, 27ECH	JUMP IF NOT STRING
27DF	CDDA29	CALL	29DAH	
27E2	CDE628	CALL	28E6H	
27E5	2AA040	LD	HL, (40A0H)	
27E8	EB	EX	DE, HL	DE => START OF STRING SPACE
27E9	2AD640	LD	HL, (40D6H)	HL => NEXT AVAILABLE STRING LOC.

27EC 7D	LD	A, L	HL
27ED 93	SUB	E	=
27EE 6F	LD	L, A	HL
27EF 7C	LD	A, H	-
27F0 9A	SBC	A, D	DE
27F1 67	LD	H, A	
27F2 C3660C	JP	0C66H	RETURN HL AS RESULT

POS

27F5 3AA640	LD	A, (40A6H)	GET POS
27F8 6F	LD	L, A	HL = POS
27F9 AF	XOR	A	
27FA 67	LD	H, A	
27FB C39A0A	JP	0A9AH	RETURN RESULT INTEGER

USR

27FE CDA941	CALL	41A9H	DOS LINK
2801 D7	RST	10H	GET NEXT CHARACTER
2802 CD2C25	CALL	252CH	EVALUATE ARGUMENT
2805 E5	PUSH	HL	SAVE POSITION
2806 219008	LD	HL, 0890H	RETURN THROUGH
2809 E5	PUSH	HL	0890H
280A 3AAF40	LD	A, (40AFH)	GET TYPE OF ARGUMENT
280D F5	PUSH	AF	SAVE TYPE
280E FE03	CP	03H	STRING?
2810 CCDA29	CALL	Z, 29DAH	CLEAN WORKSPACE IF SO
2813 F1	POP	AF	RESTORE TYPE
2814 EB	EX	DE, HL	
2815 2A8E40	LD	HL, (408EH)	GET LOCATION OF SUBROUTINE
2818 E9	JP	(HL)	GO TO SUBROUTINE

CONVERT ACCUM TO GIVEN TYPE

2819 E5	PUSH	HL	SAVE POSITION
281A E607	AND	07H	ADJUST TYPE
281C 21A118	LD	HL, 18A1H	HL => CONVERTOR TABLE
281F 4F	LD	C, A	BC => OFFSET IN TABLE
2820 0600	LD	B, 00H	
2822 09	ADD	HL, BC	HL => CONVERSION ROUTINE
2823 CD8625	CALL	2586H	EXECUTE CONVERTOR
2826 E1	POP	HL	RESTORE POSITION
2827 C9	RET		DONE

CHECK FOR ILLEGAL DIRECT ERROR

2828 E5	PUSH	HL	SAVE POSITION
2829 2AA240	LD	HL, (40A2H)	GET CURRENT LINE #
282C 23	INC	HL	DIRECT
282D 7C	LD	A, H	MODE?
282E B5	OR	L	
282F E1	POP	HL	RESTORE POSITION
2830 C0	RET	NZ	RETURN IF NOT
2831 1E16	LD	E, 16H	ID ERROR
2833 C3A219	JP	19A2H	GO TO ERROR ROUTINE

STR\$

2836 CDBD0F	CALL	0FBDH	CONVERT ACCUM TO ASCII
2839 CD6528	CALL	2865H	CREATE STRING DESCRIPTOR
283C CDDA29	CALL	29DAH	CLEAN UP

283F	012B2A	LD	BC, 2A2BH	SET RETURN
2842	C5	PUSH	BC	THROUGH 2A2BH
2843	7E	LD	A, (HL)	GET STRING LENGTH
2844	23	INC	HL	NEXT BYTE
2845	E5	PUSH	HL	SAVE POSITION
2846	CDBF28	CALL	28BFH	ALLOCATE STRING
2849	E1	POP	HL	RESTORE POSITION
284A	4E	LD	C, (HL)	GET WORKSTRING
284B	23	INC	HL	LOCATION
284C	46	LD	B, (HL)	
284D	CD5A28	CALL	285AH	CREATE NEW DESCRIPTOR
2850	E5	PUSH	HL	SAVE LOCATION OF DESCRIPTOR
2851	6F	LD	L, A	L = LENGTH OF STRING
2852	CDCE29	CALL	29CEH	MOVE STRING TO STRING SPACE
2855	D1	POP	DE	RESTORE LOCATION OF DESCRIPTION
2856	C9	RET		CONTINUE

CREATE A STRING

2857	CDBF28	CALL	28BFH	ALLOCATE STRING SPACE
285A	21D340	LD	HL, 40D3H	HL => POINTER WORKSPACE
285D	E5	PUSH	HL	SAVE POINTER
285E	77	LD	(HL), A	SAVE LENGTH OF STRING
285F	23	INC	HL	
2860	73	LD	(HL), E	SAVE STRING ADDRESS
2861	23	INC	HL	
2862	72	LD	(HL), D	
2863	E1	POP	HL	RESTORE POINTER
2864	C9	RET		DONE

A= # CHARACTERS IN STRING

WHEN DONE:

HL = 40D3H
(HL) = STRING DESCRIPTOR
DE => STRING DATA

CREATE STRING DESCRIPTOR

2865	2B	DEC	HL	BACKUP BEFORE STRING
2866	0622	LD	B, 22H	B = DELIMITER # 1 = QUOTES
2868	50	LD	D, B	D = DELIMITER # 2 = QUOTES
2869	E5	PUSH	HL	SAVE START OF STRING
286A	0EFF	LD	C, 0FFH	C = -1 (NUMBER OF CHARACTERS)
286C	23	INC	HL	NEXT BYTE
286D	7E	LD	A, (HL)	GET BYTE
286E	0C	INC	C	INCREASE # OF CHARACTERS
286F	B7	OR	A	TEST CHARACTER
2870	2806	JR	Z, 2878H	SKIP IF END OF LINE
2872	BA	CP	D	DELIMITER # 2?
2873	2803	JR	Z, 2878H	SKIP IF SO.
2875	B8	CP	B	DELIMITER # 1?
2876	20F4	JR	NZ, 286CH	LOOP IF NOT
2878	FE22	CP	22H	DOUBLE QUOTES?
287A	CC781D	CALL	Z, 1D78H	SKIP OVER IF SO
287D	E3	EX	(SP), HL	GET STRING POINTER. SAVE POSITION
287E	23	INC	HL	
287F	EB	EX	DE, HL	DE => STRING DATA
2880	79	LD	A, C	A = STRING LENGTH

2881	CD5A28	CALL	285AH	SAVE DESCRIPTOR TEMPORARILY
2884	11D340	LD	DE,40D3H	MOVE FROM TEMPORARY SPACE
2887	3ED5	LD	A,0D5H	UNUSED DUMMY ENTRY
2889	2AB340	LD	HL,(40B3H)	MOVE TO WORKSPACE
288C	222141	LD	(4121H),HL	SAVE LOCATION IN ACCUM
288F	3E03	LD	A,03H	SET ACCUM TO
2891	32AF40	LD	(40AFH),A	STRING TYPE
2894	CDD309	CALL	09D3H	MOVE DATA
2897	11D640	LD	DE,40D6H	DE => END OF WORKSPACE
289A	DF	RST	18H	WORKSPACE FULL?
289B	22B340	LD	(40B3H),HL	SAVE NEW WORK POINTER
289E	E1	POP	HL	GET POSITION
289F	7E	LD	A,(HL)	GET FIRST CHARACTER OF STRING
28A0	C0	RET	NZ	RETURN IF NO ERROR
28A1	1E1E	LD	E,1EH	ST ERROR
28A3	C3A219	JP	19A2H	GO TO ERROR ROUTINE

DISPLAY MESSAGE

28A6	23	INC	HL	MOVE TO NEXT BYTE
28A7	CD6528	CALL	2865H	CREATE DESCRIPTOR FOR MESSAGE
28AA	CDDA29	CALL	29DAH	CLEAN UP WORKSPACE
28AD	CDC409	CALL	09C4H	GET STRING INFO
28B0	14	INC	D	LENGTH = LENGTH + 1
28B1	15	DEC	D	ONE LESS BYTE TO DO
28B2	C8	RET	Z	RETURN IF DONE
28B3	0A	LD	A,(BC)	GET CHARACTER
28B4	CD2A03	CALL	032AH	DISPLAY CHARACTER
28B7	FE0D	CP	0DH	CARRIAGE RETURN?
28B9	CC0321	CALL	Z,2103H	DOS LINK IF SO
28BC	03	INC	BC	NEXT BYTE
28BD	18F2	JR	28B1H	LOOP

ALLOCATE SPACE FOR STRING

28BF	B7	OR	A	SET STATUS
28C0	0E			MASK TO 28C2H
28C0	F1	POP	AF	RESTORE STATUS
28C2	F5	PUSH	AF	SAVE STATUS
28C3	2AA040	LD	HL,(40A0H)	DE => END OF STRING SPACE
28C6	EB	EX	DE,HL	
28C7	2AD640	LD	HL,(40D6H)	HL => NEXT AVAILABLE STRING
28CA	2F	CPL		BC =
28CB	4F	LD	C,A	- LENGTH OF STRING
28CC	06FF	LD	B,0FFH	
28CE	09	ADD	HL,BC	HL => NEW NEXT AVAILABLE
28CF	23	INC	HL	SPACE OVER NEXT AVAILABLE SPRING
28D0	DF	RST	18H	STRING SPACE FULL?
28D1	3807	JR	C,28DAH	JUMP TO PACK IF SO.
28D3	22D640	LD	(40D6H),HL	SAVE NEW NEXT AVAILABLE STRING.
28D6	23	INC	HL	SPACE BACK TO ALLOCATED SPACE.
28D7	EB	EX	DE,HL	DE => NEW STRING
28D8	F1	POP	AF	RESTORE STATUS
28D9	C9	RET		DONE

PACK STRING SPACE

28DA	F1	POP	AF	WAS PACK ALREADY DONE?
28DB	1E1A	LD	E,1AH	OS ERROR
28DD	CAA219	JP	Z,19A2H	IF SO

28E0	BF	CP	A	INDICATE THAT A PACK
28E1	F5	PUSH	AF	HAS BEEN DONE
28E2	01C128	LD	BC,28C1H	RETURN TO ALLOCATION
28E5	C5	PUSH	BC	ROUTINE WHEN DONE
28E6	2AB140	LD	HL,(40B1H)	HL => TOP OF STRING SPACE
28E9	22D640	LD	(40D6H),HL	SET NEXT AVAILABLE LOCATION
28EC	210000	LD	HL,0000H	SAVE A
28EF	E5	PUSH	HL	NULL MARKER
28F0	2AA040	LD	HL,(40A0H)	SAVE BOTTOM OF
28F3	E5	PUSH	HL	STRING SPACE
28F4	21B540	LD	HL,40B5H	HL => WORKSPACE TABLE
28F7	EB	EX	DE,HL	HL => POSITION IN WORK TABLE
28F8	2AB340	LD	HL,(40B3H)	DE => END OF WORK TABLE
28FB	EB	EX	DE,HL	
28FC	DF	RST	18H	DONE WITH WORK TABLE
28FD	01F728	LD	BC,28F7H	SET RETURN TO 2BF7H
2900	C24A29	JP	NZ,294AH	PACK STRING IF NOT DONE
2903	2AF940	LD	HL,(40F9H)	HL => VARIABLES
2906	EB	EX	DE,HL	HL => POSITION IN TABLE
2907	2AFB40	LD	HL,(40FBH)	DE => ARRAYS
290A	EB	EX	DE,HL	
290B	DF	RST	18H	DONE WITH VARIABLES?
290C	2813	JR	Z,2921H	SKIP OUT IF SO
290E	7E	LD	A,(HL)	GET VARIABLE TYPE
290F	23	INC	HL	SKIP
2910	23	INC	HL	PAST
2911	23	INC	HL	NAME
2912	FE03	CP	03H	STRING?
2914	2004	JR	NZ,291AH	SKIP IF SO
2916	CD4B29	CALL	294BH	PACK STRING
2919	AF	XOR	A	CLEAR SIZE OF UNTO
291A	5F	LD	E,A	DE = SIZE OF DATA
291B	1600	LD	D,00H	SPACE
291D	19	ADD	HL,DE	MOVE PAST DATA
291E	18E6	JR	2906H	LOOP THROUGH VARIABLES
2920	C1	POP	BC	CLEAR STACK
2921	EB	EX	DE,HL	HL => POSITION IN TABLE
2922	2AFD40	LD	HL,(40FDH)	DE => END OF TABLE
2925	EB	EX	DE,HL	
2926	DF	RST	18H	DONE WITH ARRAYS
2927	CA6B29	JP	Z,296BH	JUMP OUT IF SO?
292A	7E	LD	A,(HL)	GET ARRAY TYPE
292B	23	INC	HL	POINT TO NEXT BYTE
292C	CDC209	CALL	09C2H	BC = DATA SPACE LENGTH
292F	E5	PUSH	HL	SAVE POSITION
2930	09	ADD	HL,BC	MOVE PAST DATA
2931	FE03	CP	03H	STRING?
2933	20EB	JR	NZ,2920H	LOOP IF NOT
2935	22D840	LD	(40D8H),HL	SAVE END OF ARRAY
2938	E1	POP	HL	RESTORE POSITION IN ARRAY
2939	4E	LD	C,(HL)	BC = # DIMENSIONS
293A	0600	LD	B,00H	
293C	09	ADD	HL,BC	MOVE PAST
293D	09	ADD	HL,BC	DIMENSION DATA
293E	23	INC	HL	MOVE ONE MORE BYTE
293F	EB	EX	DE,HL	HL => POSITION IN ARRAY
2940	2AD840	LD	HL,(40D8H)	DE => END OF ARRAY

2943	EB	EX	DE,HL	
2944	DF	RST	18H	DONE WITH ARRAY?
2945	28DA	JR	Z,2921H	LOOP BACK IF SO
2947	013F29	LD	BC,293FH	SET RETURN TO 293FH
294A	C5	PUSH	BC	SAVE RETURN ADDRESS
294B	AF	XOR	A	GET STRING LENGTH
294C	B6	OR	(HL)	& SET STATUS
294D	23	INC	HL	
294E	5E	LD	E, (HL)	GET
294F	23	INC	HL	STRING
2950	56	LD	D, (HL)	ADDRESS
2951	23	INC	HL	IN DE
2952	C8	RET	Z	LOOP IF NULL STRING
2953	44	LD	B,H	SAVE POSITION
2954	4D	LD	C,L	IN BC
2955	2AD640	LD	HL, (40D6H)	HL => NEXT AVAILABLE STRING
2958	DF	RST	18H	STRING ALREADY PACKED?
2959	60	LD	H,B	RESTORE POSITION
295A	69	LD	L,C	FROM BC
295B	D8	RET	C	RETURN IF ALREADY PACKED
295C	E1	POP	HL	GET RETURN
295D	E3	EX	(SP),HL	SAVE RETURN & GET POINTER
295E	DF	RST	18H	STRING BEFORE POINTER?
295F	E3	EX	(SP),HL	SAVE POINTER & GET RETURN
2960	E5	PUSH	HL	SAVE RETURN
2961	60	LD	H,B	RESTORE POSITION
2962	69	LD	L,C	FROM BC
2963	D0	RET	NC	RETURN IF STRING BEFORE POINTER
2964	C1	POP	BC	GET RETURN
2965	F1	POP	AF	CLEAR STATUS
2966	F1	POP	AF	
2967	E5	PUSH	HL	SAVE POSITION OF POINTER
2968	D5	PUSH	DE	SAVE STRING POINTER
2969	C5	PUSH	BC	SAVE RETURN
296A	C9	RET		RETURN
296B	D1	POP	DE	DE => STRING
296C	E1	POP	HL	HL => STRING DESCRIPTOR
296D	7D	LD	A,L	NULL POINTER?
296E	B4	OR	H	(DONE?)
296F	C8	RET	Z	RETURN IF SO
2970	2B	DEC	HL	BC => OLD
2971	46	LD	B, (HL)	POSITION
2972	2B	DEC	HL	OF
2973	4E	LD	C, (HL)	STRING
2974	E5	PUSH	HL	SAVE POSITION OF POINTER
2975	2B	DEC	HL	HL =
2976	6E	LD	L, (HL)	LENGTH
2977	2600	LD	H,00H	OF STRING
2979	09	ADD	HL,BC	HL = END OF OLD STRING
297A	50	LD	D,B	DE = START OF OLD STRING
297B	59	LD	E,C	
297C	2B	DEC	HL	
297D	44	LD	B,H	BC = END OF OLD STRING
297E	4D	LD	C,L	
297F	2AD640	LD	HL, (40D6H)	HL = START OF NEW STRING
2982	CD5819	CALL	1958H	MOVE STRING
2985	E1	POP	HL	RESTORE POSITION OF POINTER

2986	71	LD	(HL),C	SAVE
2987	23	INC	HL	NEW
2988	70	LD	(HL),B	POINTER
2989	69	LD	L,C	HL = NEXT
298A	60	LD	H,B	AVAILABLE
298B	2B	DEC	HL	LOCATION
298C	C3E928	JP	28E9H	LOOP

STRING ADDITION

298F	C5	PUSH	BC	SAVE PRIORITY
2990	E5	PUSH	HL	SAVE POSITION
2991	2A2141	LD	HL,(4121H)	GET STRING POINTER # 1
2994	E3	EX	(SP),HL	SAVE POINTER. GET POSITION
2995	CD9F24	CALL	249FH	EVALUATE SECOND ARGUMENT
2998	E3	EX	(SP),HL	GET POINTER, SAVE POSITION
2999	CDF40A	CALL	0AF4H	TM ERROR IF NOT STRING
299C	7E	LD	A,(HL)	GET LEN (ARG1)
299D	E5	PUSH	HL	SAVE POINTER # 1
299E	2A2141	LD	HL,(4121H)	GET POINTER #2
29A1	E5	PUSH	HL	SAVE POINTER # 2
29A2	86	ADD	A,(HL)	COMPUTE LEN (ARG 1 + ARG 2)
29A3	1E1C	LD	E,1CH	POSSIBLE LS ERROR
29A5	DAA219	JP	C,19A2H	ERROR IF OVERFLOW
29A8	CD5728	CALL	2857H	ALLOCATE NEW STRING
29AB	D1	POP	DE	GET POINTER #2
29AC	CDDE29	CALL	29DEH	DELETE WORKSPACE ENTRY
29AF	E3	EX	(SP),HL	GET PRINTER #1
29B0	CDDD29	CALL	29DDH	DELETE WORKSPACE ENTRY
29B3	E5	PUSH	HL	SAVE POINTER # 1
29B4	2AD440	LD	HL,(40D4H)	DE => NEW STRING
29B7	EB	EX	DE,HL	AREA
29B8	CDC629	CALL	29C6H	MOVE STRING #1
29BB	CDC629	CALL	29C6H	MOVE STRING #2
29BE	214923	LD	HL,2349H	SET RETURN TO 2349H
29C1	E3	EX	(SP),HL	SAVE RETURN & GET POSITION
29C2	E5	PUSH	HL	SAVE POSITION
29C3	C38428	JP	2884H	CONTINUE

MOVE STRING

29C6	E1	POP	HL	GET POSITION OF
29C7	E3	EX	(SP),HL	STRING DESCRIPTOR. SAVE RETURN
29C8	7E	LD	A,(HL)	GET STRING LENGTH
29C9	23	INC	HL	
29CA	4E	LD	C,(HL)	BC => STRING DATA
29CB	23	INC	HL	
29CC	46	LD	B,(HL)	
29CD	6F	LD	L,A	L = LENGTH
29CE	2C	INC	L	LENGTH = LENGTH + 1
29CF	2D	DEC	L	ONE LESS BYTE TO MOVE
29D0	C8	RET	Z	RETURN IF DONE
29D1	0A	LD	A,(BC)	GET BYTE
29D2	12	LD	(DE),A	SAVE BYTE
29D3	03	INC	BC	MOVE TO
29D4	13	INC	DE	NEXT BYTES
29D5	18F8	JR	29CFH	LOOP

DELETE WORKSTRING IF NECESSARY

29D7	CDF40A	CALL	0AF4H	TM ERROR IF NOT STRING
29DA	2A2141	LD	HL, (4121H)	DE => WORKSPACE ENTRY
29DD	EB	EX	DE, HL	
29DE	CDF529	CALL	29F5H	DELETE WORKSPACE ENTRY
29E1	EB	EX	DE, HL	HL => WORKSPACE ENTRY
29E2	C0	RET	NZ	EXIT IF NOT LAST ENTRY
29E3	D5	PUSH	DE	SAVE POINTER TO DESCRIPTOR
29E4	50	LD	D, B	DE => STRING
29E5	59	LD	E, C	DATA
29E6	1B	DEC	DE	POSITION JUST BEFORE DATA
29E7	4E	LD	C, (HL)	GET LENGTH OF STRING
29E8	2AD640	LD	HL, (40D6H)	HL => NEXT AVAILABLE STRING
29EB	DF	RST	18H	LAST STRING IN STRING SPACE?
29EC	2005	JR	NZ, 29F3H	SKIP IF NOT
29EE	47	LD	B, A	BC = LENGTH OF STRING
29EF	09	ADD	HL, BC	ADJUST NEXT
29F0	22D640	LD	(40D6H), HL	AVAILABLE STRING LOCATION
29F3	E1	POP	HL	RESTORE POINTER TO DESCRIPTOR
29F4	C9	RET		DONE

DELETE WORKSPACE ENTRY IF LAST

29F5	2AB340	LD	HL, (40B3H)	HL => LAST ENTRY
29F8	2B	DEC	HL	
29F9	46	LD	B, (HL)	BC => STRING DATA
29FA	2B	DEC	HL	
29FB	4E	LD	C, (HL)	
29FC	2B	DEC	HL	
29FD	DF	RST	18H	LAST ENTRY SAME AS GIVEN?
29FE	C0	RET	NZ	RETURN IF NOT
29FF	22B340	LD	(40B3H), HL	DELETE IF LAST
2A02	C9	RET		DONE

LEN

2A03	01F827	LD	BC, 27F8H	RETURN A AS
2A06	C5	PUSH	BC	INTEGER RESULT
2A07	CDD729	CALL	29D7H	DELETE TEMPORARY STRING
2A0A	AF	XOR	A	CLEAR
2A0B	57	LD	D, A	D
2A0C	7E	LD	A, (HL)	GET STRING LENGTH
2A0D	B7	OR	A	CLEAR STATUS
2A0E	C9	RET		FINISH UP

ASC

2A0F	01F827	LD	BC, 27F8H	RETURN A AS
2A12	C5	PUSH	BC	INTEGER RESULT
2A13	CD072A	CALL	2A07H	GET LEN IN A
2A16	CA4A1E	JP	Z, 1E4AH	FC ERROR IF NULL STRING
2A19	23	INC	HL	DE =>
2A1A	5E	LD	E, (HL)	FIRST CHARACTER
2A1B	23	INC	HL	OF STRING
2A1C	56	LD	D, (HL)	
2A1D	1A	LD	A, (DE)	GET FIRST CHARACTER.
2A1E	C9	RET		RETURN AS ASCII VALUE

CHR\$

2A1F	3E01	LD	A, 01H	CREATE A 1-BYTE
2A21	CD5728	CALL	2857H	STRING

2A24	CD1F2B	CALL	2B1FH	CONVERT PARAM TO INTEGER
2A27	2AD440	LD	HL, (40D4H)	GET STRING POINTER
2A2A	73	LD	(HL), E	STORE CHARACTER IN STRING
2A2B	C1	POP	BC	CLEAR STACK
2A2C	C38428	JP	2884H	FINISH UP & RETURN

STRING\$

2A2F	D7	RST	10H	GET NEXT CHARACTER
2A30	CF	RST	08H	SYNTAX CHARACTER IF CHARACTER
2A31	28	" ("		NOT " ("
2A32	CD1C2B	CALL	2B1CH	EVALUATE # CHARACTERS
2A35	D5	PUSH	DE	SAVE # CHARACTERS
2A36	CF	RST	08H	SYNTAX ERROR IF
2A37	2C	", "		NEXT CHARACTER NOT COMMA
2A38	CD3723	CALL	2337H	EVALUATE SECOND PARAM
2A3B	CF	RST	08H	SYNTAX ERROR IF NEXT
2A3C	29) "		CHARACTER NOT) "
2A3D	E3	EX	(SP), HL	SAVE POSITION. GET # CHARACTERS
2A3E	E5	PUSH	HL	SAVE # CHARACTERS
2A3F	E7	RST	20H	GET TYPE OF ACCUM
2A40	2805	JR	Z, 2A47H	JUMP IF STRING
2A42	CD1F2B	CALL	2B1FH	CONVERT PARAM TO INTEGER
2A45	1803	JR	2A4AH	SKIP NEXT INSTRUCTION
2A47	CD132A	CALL	2A13H	GET ASC (PARMA
2A4A	D1	POP	DE	RESTORE # CHARACTERS
2A4B	F5	PUSH	AF	SAVE STATUS TWICE
2A4C	F5	PUSH	AF	
2A4D	7B	LD	A, E	CREATE THE
2A4E	CD5728	CALL	2857H	NECESSARY STRING
2A51	5F	LD	E, A	
2A52	F1	POP	AF	RESTORE STATUS
2A53	1C	INC	E	NULL STRING?
2A54	1D	DEC	E	
2A55	28D4	JR	Z, 2A2BH	EXIT IF NULL STRING
2A57	2AD440	LD	HL, (40D4H)	HL => STRING
2A5A	77	LD	(HL), A	SAVE CHARACTER
2A5B	23	INC	HL	NEXT BYTE
2A5C	1D	DEC	E	ONE LESS TO STORE
2A5D	20FB	JR	NZ, 2A5AH	LOOP UNTIL DONE
2A5F	18CA	JR	2A2BH	CLEAR STACK & FINISH

LEFT\$

2A61	CDDF2A	CALL	2ADFH	GET PARAM 2
2A64	AF	XOR	A	OFFSET =0
2A65	E3	EX	(SP), HL	SAVE POSITION HL => PARAM 1
2A66	4F	LD	C, A	C = OFFSET
2A67	3E			SKIP TO 2A69H
2A68	E5	PUSH	HL	SAVE STRING POINTER
2A69	E5	PUSH	HL	SAVE STRING POINTER
2A6A	7E	LD	A, (HL)	GET LEN (SPRING)
2A6B	B8	CP	B	COMPARE WITH PARAM 2
2A6C	3802	JR	C, 2A70H	SKIP IF LESS
2A6E	78	LD	A, B	A = PARAM 2
2A6F	110E00	LD	DE, 000EH	SKIP TO 2A72H
2A70	0E00	LD	C, 00H	USE ENTIRE STRING
2A72	C5	PUSH	BC	SAVE OFFSET & PARAM 2
2A73	CDBF28	CALL	28BFH	CREATE DESTINATION STRING

2A76	C1	POP	BC	GET OFFSET & PARAM 2
2A77	E1	POP	HL	HL => STRING POINTER
2A78	E5	PUSH	HL	
2A79	23	INC	HL	HL => STRING DATA
2A7A	46	LD	B, (HL)	
2A7B	23	INC	HL	
2A7C	66	LD	H, (HL)	
2A7D	68	LD	L, B	
2A7E	0600	LD	B, 00H	BC = OFFSET INTO STRING
2A80	09	ADD	HL, BC	HL => START OF RESULT STRING
2A81	44	LD	B, H	BC => START OF RESULT STRING
2A82	4D	LD	C, L	
2A83	CD5A28	CALL	285AH	CREATE STRING DESCRIPTOR
2A86	6F	LD	L, A	L = LENGTH OF NEW STRING
2A87	CDCE29	CALL	29CEH	MOVE DATA TO NEW STRING
2A8A	D1	POP	DE	DE => STRING POINTER
2A8B	CDDE29	CALL	29DEH	CLEAN UP WORKSPACE
2A8E	C38428	JP	2884H	FINISH UP & RETURN

RIGHT\$

2A91	CDDF2A	CALL	2ADFH	GET PARAM 2
2A94	D1	POP	DE	GET &
2A95	D5	PUSH	DE	SAVE PARAM 1 POINTER
2A96	1A	LD	A, (DE)	OFFSET=
2A97	90	SUB	B	LEN (PARAM 1) - PARAM 2
2A98	18CB	JR	2A65H	CONTINUE

MID\$

2A9A	EB	EX	DE, HL	GET POSITION
2A9B	7E	LD	A, (HL)	GET CHARACTER.
2A9C	CDE22A	CALL	2AE2H	GET PARAM 2
2A9F	04	INC	B	SET STATUS
2AA0	05	DEC	B	OF PARAM 2
2AA1	CA4A1E	JP	Z, 1E4AH	FC ERROR IF ZERO
2AA4	C5	PUSH	BC	SAVE PARAM 2
2AA5	1EFF	LD	E, 0FFH	DEFAULT PARAM 3
2AA7	FE29	CP	29H	IS CHARACTER A ")" "?"
2AA9	2805	JR	Z, 2AB0H	SKIP IF SO
2AAB	CF	RST	08H	SYNTAX ERROR IF
2AAC	2C	" , "		NEXT CHARACTER NOT COMMA
2AAD	CD1C2B	CALL	2B1CH	EVALUATE PARAM 3
2AB0	CF	RST	08H	SYNTAX ERROR IF
2AB1	29	") "		NEXT CHARACTER NOT 911
2AB2	F1	POP	AF	GET PARAM 2
2AB3	E3	EX	(SP), HL	SAVE POSITION. GET PARAM 1
2AB4	01692A	LD	BC, 2A69H	RETURN THROUGH 2A69H
2AB7	C5	PUSH	BC	
2AB8	3D	DEC	A	# CHARACTERS = 0 IF
2AB9	BE	CP	(HL)	PARAM 2 IS GREATER
2ABA	0600	LD	B, 00H	THAN LEN (PARAM 1)
2ABC	D0	RET	NC	RETURN IF SO
2ABD	4F	LD	C, A	C = PARAM 2 = OFFSET
2ABE	7E	LD	A, (HL)	A = LEN (PARAM 1) - PARAM 2
2ABF	91	SUB	C	
2AC0	BB	CP	E	COMPARE WITH PARAM 3
2AC1	47	LD	B, A	USE CHARACTERS LEFT IF LESS
2AC2	D8	RET	C	LEFT THAN PARAM 3

2AC3 43 LD B,E USE PARAM 3 CHARACTERS
2AC4 C9 RET OTHERWISE. FINISH UP

VAL

2AC5 CD072A CALL 2A07H CLEAN UP WORKSPACE
2AC8 CAF827 JP Z,27F8H EXIT 0 IF NULL STRING
2ACB 5F LD E,A E = STRING LENGTH
2ACC 23 INC HL HL => STRING DATA
2ACD 7E LD A,(HL)
2ACE 23 INC HL
2ACF 66 LD H,(HL)
2AD0 6F LD L,A
2AD1 E5 PUSH HL SAVE STRING POINTER
2AD2 19 ADD HL,DE HL => END OF STRING + 1
2AD3 46 LD B,(HL) B = END MARKER
2AD4 72 LD (HL),D PLACE NULL END MARKER
2AD5 E3 EX (SP),HL SAVE END. GET START
2AD6 C5 PUSH BC SAVE MARKET CHARACTER
2AD7 7E LD A,(HL) GET FIRST CHARACTER.
2AD8 CD650E CALL 0E65H DECODE NUMBER
2ADB C1 POP BC RESTORE MARKER
2ADC E1 POP HL RESTORE END OF STRING
2ADD 70 LD (HL),B SAVE OLD MARKER
2ADE C9 RET DONE
2ADF EB EX DE,HL GET POSITION IN TEXT
2AE0 CF RST 08H SYNTAX ERROR IF
2AE1 29 ") " NEXT CHARACTER NOT ") "
2AE2 C1 POP BC GET RETURN
2AE3 D1 POP DE GET PARAM 2
2AE4 C5 PUSH BC SAVE RETURN
2AE5 43 LD B,E B = PARAM 2
2AE6 C9 RET CONTINUE
2AE7 FE7A CP 7AH SN ERROR IF NOT MID\$
2AE9 C29719 JP NZ,1997H TOKEN
2AEC C3D941 JP 41D9H GO TO DOS OTHERWISE

INP

2AEF CD1F2B CALL 2B1FH EVALUATE PARAM
2AF2 329440 LD (4094H),A STORE PARAM
2AF5 CD9340 CALL 4093H GET INPUT
2AF8 C3F827 JP 27F8H SAVE AS INTEGER

OUT

2AFB CD0E2B CALL 2B0EH EVALUATE PARAMS
2AFE C39640 JP 4096H OUTPUT & RETURN

EVALUATE PARAM AS INTEGER

2B01 D7 RST 10H GET NEXT CHARACTER
2B02 CD3723 CALL 2337H EVALUATE EXPRESSION
2B05 E5 PUSH HL SAVE POSITION
2B06 CD7F0A CALL 0A7FH CONVERT RESULT TO INTEGER
2B09 EB EX DE,HL RESULT IN DE
2B0A E1 POP HL RESTORE POSITION
2B0B 7A LD A,D SET STATUS
2B0C B7 OR A OF MSB
2B0D C9 RET DONE

SET UP FOR OUT

2B0E	CD1C2B	CALL	2B1CH	EVALUATE PARAM 1
2B11	329440	LD	(4094H),A	STORE FOR INP
2B14	329740	LD	(4097H),A	& OUT
2B17	CF	RST	08H	SYNTAX ERROR IF
2B18	2C	" , "		NEXT CHARACTER NOT COMMA
2B19	1801	JR	2B1CH	SKIP NEXT INSTRUCTION
2B1B	D7	RST	10H	GET NEXT CHARACTER
2B1C	CD3723	CALL	2337H	EVALUATE PARAM 2
2B1F	CD052B	CALL	2B05H	CONVERT TO INTEGER
2B22	C24A1E	JP	NZ,1E4AH	FC ERROR IF TOO LARGE
2B25	2B	DEC	HL	BACKUP
2B26	D7	RST	10H	GET NEXT CHARACTER.
2B27	7B	LD	A,E	A = LSB
2B28	C9	RET		DONE

LLIST

2B29	3E01	LD	A,01H	SET ROUTE
2B2B	329C40	LD	(409CH),A	TO PRINTER

LIST

2B2E	C1	POP	BC	CLEAR STACK
2B2F	CD101B	CALL	1B10H	GET START & END LINES
2B32	C5	PUSH	BC	SAVE START LINE
2B33	21FFFF	LD	HL,0FFFFH	SET DIRECT MODE
2B36	22A240	LD	(40A2H),HL	
2B39	E1	POP	HL	HL = START LINE POSITION
2B3A	D1	POP	DE	DE = END LINE
2B3B	4E	LD	C, (HL)	BC = FORWARD
2B3C	23	INC	HL	POINTER
2B3D	46	LD	B, (HL)	
2B3E	23	INC	HL	
2B3F	78	LD	A,B	END OF PROGRAM?
2B40	B1	OR	C	
2B41	CA191A	JP	Z,1A19H	READY IF SO
2B44	CDDF41	CALL	41DFH	DOS LINK
2B47	CD9B1D	CALL	1D9BH	CHECK FOR SPACE PAUSE KEY
2B4A	C5	PUSH	BC	SAVE FORWARD POINTER
2B4B	4E	LD	C, (HL)	BC = LINE #
2B4C	23	INC	HL	
2B4D	46	LD	B, (HL)	
2B4E	23	INC	HL	
2B4F	C5	PUSH	BC	SAVE LINE #
2B50	E3	EX	(SP),HL	GET LINE #. SAVE POSITION
2B51	EB	EX	DE,HL	DONE WITH
2B52	DF	RST	18H	LISTING?
2B53	C1	POP	BC	GET POSITION
2B54	DA181A	JP	C,1A18H	EXIT IF DONE
2B57	E3	EX	(SP),HL	GET FORWARD POINTER, SAVE END LINE
2B58	E5	PUSH	HL	SAVE FORWARD POINTER
2B59	C5	PUSH	BC	SAVE POSITION
2B5A	EB	EX	DE,HL	HL = LINE #
2B5B	22EC40	LD	(40ECH),HL	MAKE IT CURRENT LINE
2B5E	CDAF0F	CALL	0FAFH	DISPLAY LINE #
2B61	3E20	LD	A,20H	DISPLAY SPACE
2B63	E1	POP	HL	GET POSITION
2B64	CD2A03	CALL	032AH	OUTPUT SPACE

2B67	CD7E2B	CALL	2B7EH	DECODE LINE TO BUFFER
2B6A	2AA740	LD	HL, (40A7H)	HL => DECODED LINE
2B6D	CD752B	CALL	2B75H	OUTPUT LINE
2B70	CDFE20	CALL	20FEH	NEW LINE
2B73	18BE	JR	2B33H	LOOP FOR MORE
2B75	7E	LD	A, (HL)	GET CHARACTER
2B76	B7	OR	A	NULL? (END OF LINE?)
2B77	C8	RET	Z	RETURN IF SO
2B78	CD2A03	CALL	032AH	DISPLAY (PRINT?) CHARACTER.
2B7B	23	INC	HL	NEXT CHARACTER.
2B7C	18F7	JR	2B75H	LOOP UNTIL DONE
2B7E	E5	PUSH	HL	SAVE POSITION
2B7F	2AA740	LD	HL, (40A7H)	HL => KEYBOARD BUFFER
2B82	44	LD	B, H	BC => KEYBOARD BUFFER
2B83	4D	LD	C, L	
2B84	E1	POP	HL	RESTORE POSITION
2B85	C39A06	JP	069AH	CONTINUE
2B88	00	NOP		UNUSED BYTE
2B89	03	INC	BC	MOVE UP IN BUFFER
2B8A	15	DEC	D	ONE LESS CHARACTER LEFT
2B8B	C8	RET	Z	RETURN IF BUFFER FULL
2B8C	23	INC	HL	MOVE FORWARD IN TEXT
2B8D	7E	LD	A, (HL)	GET CHARACTER
2B8E	B7	OR	A	END OF LINE?
2B8F	02	LD	(BC), A	STORE CHARACTER IN BUFFER
2B90	C8	RET	Z	RETURN IF END OF LINE
2B91	C32D30	JP	302DH	CONTINUE
2B94	FEFB	CP	0FBH	SINGLE QUOTE?
2B96	2008	JR	NZ, 2BA0H	SKIP IF NOT
2B98	0B	DEC	BC	BACK UP
2B99	0B	DEC	BC	4 BYTES
2B9A	0B	DEC	BC	
2B9B	0B	DEC	BC	
2B9C	14	INC	D	INCREASE SPACE
2B9D	14	INC	D	LEFT BY
2B9E	14	INC	D	4 BYTES
2B9F	14	INC	D	
2BA0	FE95	CP	95H	ELSE?
2BA2	CC240B	CALL	Z, 0B24H	BACK UP IF SO
2BA5	D67F	SUB	7FH	ADJUST TOKEN VALUE
2BA7	E5	PUSH	HL	SAVE POSITION IN TEXT
2BA8	5F	LD	E, A	E = TOKEN VALUE
2BA9	215016	LD	HL, 1650H	HL => RESERVED WORDS
2BAC	7E	LD	A, (HL)	GET CHARACTER
2BAD	B7	OR	A	TEST IT
2BAE	23	INC	HL	NEXT CHARACTER
2BAF	F2AC2B	JP	P, 2BACH	LOOP IF NOT NEW WORD
2BB2	1D	DEC	E	ONE LESS TO SKIP
2BB3	20F7	JR	NZ, 2BACH	TO NEXT WORD IF NOT DONE
2BB5	E67F	AND	7FH	MASK OFF BIT 7
2BB7	02	LD	(BC), A	SAVE CHARACTER IN BUFFER
2BB8	03	INC	BC	NEXT BUFFER LOCATION
2BB9	15	DEC	D	ONE LESS BYTE FREE
2BBA	CAD828	JP	Z, 28D8H	JUMP IF BUFFER FULL
2BBD	7E	LD	A, (HL)	GET NEXT CHARACTER
2BBE	23	INC	HL	NEXT BYTE
2BBF	B7	OR	A	TEST CHARACTER

2BC0	F2B72B	JP	P,2BB7H	LOOP UNTIL NEW WORD
2BC3	E1	POP	HL	RESTORE TEXT POSITION
2BC4	18C6	JR	2B8CH	CONTINUE

DELETE

2BC6	CD101B	CALL	1B10H	GET START & END LINES
2BC9	D1	POP	DE	GET END LINE
2BCA	C5	PUSH	BC	SAVE START POSITION
2BCB	C5	PUSH	BC	TWICE
2BCC	CD2C1B	CALL	1B2CH	LOOK FOR END LINE
2BCF	3005	JR	NC,2BD6H	SKIP IF NOT FOUND
2BD1	54	LD	D,H	DE => FIRST LINE
2BD2	5D	LD	E,L	AFTER DELETION
2BD3	E3	EX	(SP),HL	SAVE POINTER. GET START
2BD4	E5	PUSH	HL	SAVE START
2BD5	DF	RST	18H	LAST GREATER OR EQUAL TO FIRST?
2BD6	D24A1E	JP	NC,1E4AH	FC ERROR IF SO
2BD9	212919	LD	HL,1929H	DISPLAY
2BDC	CDA728	CALL	28A7H	"READY?"
2BDF	C1	POP	BC	BC => START OF AREA TO DELETE
2BE0	21E81A	LD	HL,1AE8H	RETURN THROUGH UPDATE
2BE3	E3	EX	(SP),HL	SAVE RETURN THROUGH UPDATE
2BE4	EB	EX	DE,HL	DE => AREA AFTER DELETION
2BE5	2AF940	LD	HL,(40F9H)	HL => END OF AREA
2BE8	1A	LD	A,(DE)	GET BYTE
2BE9	02	LD	(BC),A	SAVE BYTE
2BEA	03	INC	BC	NEXT
2BEB	13	INC	DE	BYTES.
2BEC	DF	RST	18H	DONE YET?
2BED	20F9	JR	NZ,2BE8H	LOOP IF NOT
2BEF	60	LD	H,B	HL => NEW END
2BF0	69	LD	L,C	PROGRAM
2BF1	22F940	LD	(40F9H),HL	SAVE NEW END OF PROGRAM
2BF4	C9	RET		DONE

CSAVE

2BF5	CD8402	CALL	0284H	TURN TAPE ON WRITE LEADER
2BF8	CD3723	CALL	2337H	EVALUATE FILENAME
2BFB	E5	PUSH	HL	SAVE POSITION
2BFC	CD132A	CALL	2A13H	DE => FILENAME
2BFF	3ED3	LD	A,0D3H	WRITE 0D3H
2C01	CD6402	CALL	0264H	BYTE AS MARKET
2C04	CD6102	CALL	0261H	WRITE IT TWICE MORE
2C07	1A	LD	A,(DE)	GET FILENAME
2C08	CD6402	CALL	0264H	WRITE FILENAME
2C0B	2AA440	LD	HL,(40A4H)	DE => START OF PROGRAM
2C0E	EB	EX	DE,HL	DE => END OF PROGRAM
2C0F	2AF940	LD	HL,(40F9H)	
2C12	1A	LD	A,(DE)	GET BYTE
2C13	13	INC	DE	NEXT BYTE
2C14	CD6402	CALL	0264H	SAVE TO TAPE
2C17	DF	RST	18H	DONE YET?
2C18	20F8	JR	NZ,2C12H	LOOP IF NOT
2C1A	CDF801	CALL	01F8H	TURN OFF TAPE
2C1D	E1	POP	HL	RESTORE POSITION
2C1E	C9	RET		DONE

<u>CLOAD</u>				
2C1F D6B2	SUB	0B2H		CLOAD? (VERIFY)
2C21 2802	JR	Z,2C25H		JUMP IF SO
2C23 AF	XOR	A		RESET VERIFY FLAG
2C24 012F23	LD	BC,232FH		MASK TO 2C27H
2C25 2F	CPL			SET VERIFY FLAG
2C25 23	INC	HL		UPDATE POSITION
2C27 F5	PUSH	AF		SAVE VERIFY FLAG
2C28 7E	LD	A, (HL)		END OF
2C29 B7	OR	A		COMMAND?
2C2A 2807	JR	Z,2C33H		SKIP IF SO
2C2C CD3723	CALL	2337H		EVALUATE FILENAME
2C2F CD132A	CALL	2A13H		DE => FILENAME
2C32 1A	LD	A, (DE)		GET FILENAME
2C33 6F	LD	L,A		SAVE FILENAME IN L
2C34 F1	POP	AF		GET VERIFY FLAG
2C35 B7	OR	A		VERIFY?
2C36 67	LD	H,A		SAVE VERIFY FLAG
2C37 222141	LD	(4121H),HL		& FILENAME IN ACCUM
2C3A CC4D1B	CALL	Z,1B4DH		NEW IF NO VERIFY
2C3D 210000	LD	HL,0000H		THIS IS NOT PRINT #
2C40 CD9302	CALL	0293H		READ TAPE HEADER
2C43 2A2141	LD	HL, (4121H)		GET FLAG & FILENAME
2C46 EB	EX	DE,HL		DE = FLAG & FILENAME
2C47 0603	LD	B,03H		READ 3 MARKERS
2C49 CD3502	CALL	0235H		READ BYTE
2C4C D6D3	SUB	0D3H		MARKER?
2C4E 20F7	JR	NZ,2C47H		RESTART IF NOT
2C50 10F7	DJNZ	2C49H		LOOP FOR MORE MARKERS
2C52 CD3502	CALL	0235H		READ FILENAME
2C55 1C	INC	E		NULL
2C56 1D	DEC	E		FILENAME
2C57 2803	JR	Z,2C5CH		SKIP & LOAD IF SO
2C59 BB	CP	E		CORRECT FILENAME
2C5A 2037	JR	NZ,2C93H		NEXT FILE IF NOT
2C5C 2AA440	LD	HL, (40A4H)		POSITION TO PROGRAM START
2C5F 0603	LD	B,03H		EOF = 3 NULLS
2C61 CD3502	CALL	0235H		READ BYTE
2C64 5F	LD	E,A		SAVE IN E
2C65 96	SUB	(HL)		COMPARE WITH MEMORY
2C66 A2	AND	D		COMBINE WITH VERIFY FLAG
2C67 2021	JR	NZ,2C8AH		JUMP IF VERIFY & BAD
2C69 73	LD	(HL),E		SAVE BYTE
2C6A CD6C19	CALL	196CH		CHECK MEMORY SPACE
2C6D 7E	LD	A, (HL)		GET BYTE
2C6E B7	OR	A		END OF LINE?
2C6F 23	INC	HL		NEXT BYTE
2C70 20ED	JR	NZ,2C5FH		RESET EOF IF NOT
2C72 CD2C02	CALL	022CH		TWINKLE, TWINKLE, LITTLE *
2C75 10EA	DJNZ	2C61H		LOOP TO PROGRAM END
2C77 22F940	LD	(40F9H),HL		SAVE END OF PROGRAM
2C7A CDF801	CALL	01F8H		TURN OFF TAPE
2C7D 212919	LD	HL,1929H		PRINT
2C80 CDA728	CALL	28A7H		"READY"
2C83 2AA440	LD	HL, (40A4H)		POSITION TO PROGRAM START
2C86 E5	PUSH	HL		SAVE POSITION
2C87 C3E81A	JP	1AE8H		UPDATE POINTERS & RETURN

2C8A	CDBD31	CALL	31BDH	RESET TAPE I/O
2C8D	CDA728	CALL	28A7H	PRINT "BAD"
2C90	C3181A	JP	1A18H	GO TO READY
2C93	323E3C	LD	(3C3EH),A	DISPLAY FILENAME
2C96	0603	LD	B,03H	EOF = 3 MILES
2C98	CD3502	CALL	0235H	READ BYTE
2C9B	B7	OR	A	NULL?
2C9C	20F8	JR	NZ,2C96H	RESET IF NOT
2C9E	10F8	DJNZ	2C98H	LOOP TO END
2CA0	CD9602	CALL	0296H	READ NEW HEADER
2CA3	18A2	JR	2C47H	TRY AGAIN
2CA5	4241440D00			"BAD" 0D 00

PEEK

2CAA	CD7F0A	CALL	0A7FH	HL = ARGUMENT
2CAD	7E	LD	A, (HL)	GET BYTE
2CAE	C3F827	JP	27F8H	RETURN AS INTEGER

POKE

2CB1	CD022B	CALL	2B02H	EVALUATE ADDRESS
2CB4	D5	PUSH	DE	SAVE ADDRESS
2CB5	CF	RST	08H	SYNTAX ERROR IF
2CB6	2C		", "	NEXT CHARACTER NOT COMMA
2CB7	CD1C2B	CALL	2B1CH	EVALUATE VALUE TO STORE
2CBA	D1	POP	DE	RESTORE ADDRESS
2CBB	12	LD	(DE),A	STORE VALUE IN ADDRESS
2CBC	C9	RET		DONE

USING

2CBD	CD3823	CALL	2338H	EVALUATE FORMAT STRING
2CC0	CDF40A	CALL	0AF4H	TM ERROR IF NOT STRING
2CC3	CF	RST	08H	SYNTAX ERROR IF
2CC4	3B		"; "	NEXT CHARACTER NOT "; "
2CC5	EB	EX	DE,HL	SAVE POSITION IN DE
2CC6	2A2141	LD	HL, (4121H)	GET STRING POINTER
2CC9	1808	JR	2CD3H	CONTINUE
2CCB	3ADE40	LD	A, (40DEH)	GET DELIMITER
2CCE	B7	OR	A	NULL? (END OF LINE)
2CCF	280C	JR	Z,2CDDH	FC ERROR IF SO
2CD1	D1	POP	DE	GET STRING POINTER
2CD2	EB	EX	DE,HL	SAVE POSITION IN TEXT
2CD3	E5	PUSH	HL	SAVE POINTER TO FORMAT
2CD4	AF	XOR	A	CLEAR
2CD5	32DE40	LD	(40DEH),A	SEPARATOR CHARACTER.
2CD8	BA	CP	D	SET STATUS
2CD9	F5	PUSH	AF	SAVE SEPARATOR CHARACTER
2CDA	D5	PUSH	DE	SAVE POSITION IN TEXT
2CDB	46	LD	B, (HL)	GET LENGTH OF FORMAT
2CDC	B0	OR	B	TEST IT FOR NULL
2CDD	CA4A1E	JP	Z,1E4AH	FC ERROR IF SO
2CE0	23	INC	HL	
2CE1	4E	LD	C, (HL)	HL => FORMAT STRING
2CE2	23	INC	HL	
2CE3	66	LD	H, (HL)	
2CE4	69	LD	L, C	
2CE5	181C	JR	2D03H	CONTINUE
2CE7	58	LD	E, B	SAVE FORMAT SIZE LEFT

2CE8 E5	PUSH	HL	SAVE POSITION IN FORMAT
2CE9 0E02	LD	C, 02H	FIELD SIZE = 2
2CEB 7E	LD	A, (HL)	GET CHARACTER
2CEC 23	INC	HL	NEXT
2CED FE25	CP	25H	PERCENT SIGN?
2CEF CA172E	JP	Z, 2E17H	DISPLAY FIELD IF SO
2CF2 FE20	CP	20H	SPACE?
2CF4 2003	JR	NZ, 2CF9H	RESET IF NOT
2CF6 0C	INC	C	INCREASE FIELD SIZE
2CF7 10F2	DJNZ	2CEBH	LOOP TO END OF FORMAT
2CF9 E1	POP	HL	RESTORE POSITION IN FORMAT
2CFA 43	LD	B, E	RESTORE FORMAT SIZE LEFT
2CFB 3E25	LD	A, 25H	RESTORE CHARACTER TO DISPLAY
2CFD CD492E	CALL	2E49H	DISPLAY "+" IF REQUESTED
2D00 CD2A03	CALL	032AH	DISPLAY CHARACTER.
2D03 AF	XOR	A	CLEAR FLAGS
2D04 5F	LD	E, A	ZERO E AND D
2D05 57	LD	D, A	
2D06 CD492E	CALL	2E49H	PRINT LEADING + IF REQUIRED
2D09 57	LD	D, A	SAVE EDIT FLAG
2D0A 7E	LD	A, (HL)	GET FORMAT CHARACTER.
2D0B 23	INC	HL	NEXT BYTE
2D0C FE21	CP	21H	"! "?
2D0E CA142E	JP	Z, 2E14H	JUMP IF SO
2D11 FE23	CP	23H	"# "?
2D13 2837	JR	Z, 2D4CH	JUMP IF SO
2D15 05	DEC	B	ONE LESS CHARACTER IN FORMAT
2D16 CAFE2D	JP	Z, 2DFEH	JUMP IF DONE WITH FORMAT
2D19 FE2B	CP	2BH	"+" ?
2D1B 3E08	LD	A, 08H	INCLUDE SIGN IF SO
2D1D 28E7	JR	Z, 2D06H	LOOP BACK IF SO
2D1F 2B	DEC	HL	BACK UP AGAIN
2D20 7E	LD	A, (HL)	GET FORMAT CHARACTER
2D21 23	INC	HL	POINT TO NEXT CHARACTER
2D22 FE2E	CP	2EH	PERIOD?
2D24 2840	JR	Z, 2D66H	JUMP IF SO
2D26 FE25	CP	25H	PERCENT SIGN?
2D28 28BD	JR	Z, 2CE7H	JUMP IF SO
2D2A BE	CP	(HL)	SAME AS NEXT CHARACTER?
2D2B 20D0	JR	NZ, 2CFDH	JUMP IF NOT
2D2D FE24	CP	24H	DOLLAR'S SIGN?
2D2F 2814	JR	Z, 2D45H	JUMP IF SO
2D31 FE2A	CP	2AH	ASTERISK?
2D33 20C8	JR	NZ, 2CFDH	JUMP IF NOT
2D35 78	LD	A, B	ONLY ONE
2D36 FE02	CP	02H	CHARACTER LEFT IN FORMAT
2D38 23	INC	HL	MOVE FORWARD
2D39 3803	JR	C, 2D3EH	SKIP IF SO
2D3B 7E	LD	A, (HL)	GET CHARACTER
2D3C FE24	CP	24H	DOLLAR'S SIGN?
2D3E 3E20	LD	A, 20H	INCLUDE LEADING ASTERISKS
2D40 2007	JR	NZ, 2D49H	NO \$? SKIP IF SO
2D42 05	DEC	B	ONE LESS CHARACTER IN FORMAT
2D43 1C	INC	E	ONE MORE CHARACTER LEFT OF DECIMAL
2D44 FEAF	CP	0AFH	SKIP TO 2D46H
2D45 AF	XOR	A	CLEAR UPDATE NO LEADING ASTERISKS
2D46 C610	ADD	A, 10H	INCLUDE FLOATING DOLLAR'S SIGN

2D48	23	INC	HL	NEXT FORMAT CHARACTER.
2D49	1C	INC	E	ONE MORE CHARACTER LEFT OF DECIMAL
2D4A	82	ADD	A,D	UPDATE
2D4B	57	LD	D,A	EDIT FLAG
2D4C	1C	INC	E	ONE MORE CHARACTER LEFT OF DECIMAL
2D4D	0E00	LD	C,00H	NO CHARACTERS RIGHT OF DECIMAL
2D4F	05	DEC	B	ONE LESS CHARACTER IN FORMAT
2D50	2847	JR	Z,2D99H	JUMP IF FORMAT USED UP
2D52	7E	LD	A,(HL)	GET NEXT CHARACTER.
2D53	23	INC	HL	POINT TO NEXT BYTE
2D54	FE2E	CP	2EH	DECIMAL POINT?
2D56	2818	JR	Z,2D70H	JUMP IF SO
2D58	FE23	CP	23H	"#"?
2D5A	28F0	JR	Z,2D4CH	JUMP IF SO
2D5C	FE2C	CP	2CH	COMMA?
2D5E	201A	JR	NZ,2D7AH	JUMP IF NOT
2D60	7A	LD	A,D	GET EDIT FLAG
2D61	F640	OR	40H	INCLUDE COMMAS
2D63	57	LD	D,A	SAVE EDIT FLAG
2D64	18E6	JR	2D4CH	LOOP
2D66	7E	LD	A,(HL)	GET NEXT CHARACTER.
2D67	FE23	CP	23H	"#"?
2D69	3E2E	LD	A,2EH	RESTORE PERIOD
2D6B	2090	JR	NZ,2CFDH	JUMP IF NOT "#".
2D6D	0E01	LD	C,01H	ONE CHARACTER RIGHT OF DECIMAL
2D6F	23	INC	HL	MOVE FORWARD IN FORMAT
2D70	0C	INC	C	INCREASE CHARACTERS RIGHT OF DECIMAL
2D71	05	DEC	B	DECREASE CHARACTERS LEFT IN FORMAT
2D72	2825	JR	Z,2D99H	JUMP IF FORMAT USED UP
2D74	7E	LD	A,(HL)	GET CHARACTER FROM FORMAT
2D75	23	INC	HL	NEXT CHARACTER
2D76	FE23	CP	23H	"#"?
2D78	28F6	JR	Z,2D70H	LOOP FOR MORE IF SO
2D7A	D5	PUSH	DE	SAVE EDIT FLAG
2D7B	11972D	LD	DE,2D97H	SET RETURN
2D7E	D5	PUSH	DE	THROUGH 2D97H
2D7F	54	LD	D,H	DE = CURRENT FORMAT POSITION
2D80	5D	LD	E,L	
2D81	FE5B	CP	5BH	EXPONENTIAL NOTATION?
2D83	C0	RET	NZ	RETURN IF NOT
2D84	BE	CP	(HL)	CHECK FOR SECOND UP-ARROW.
2D85	C0	RET	NZ	RETURN IF NOT FOUND
2D86	23	INC	HL	CHECK
2D87	BE	CP	(HL)	FOR THIRD UP-ARROW.
2D88	C0	RET	NZ	RETURN IF NOT FOUND
2D89	23	INC	HL	CHECK
2D8A	BE	CP	(HL)	FOR FOURTH UP-ARROW.
2D8B	C0	RET	NZ	RETURN IF NOT FOUND
2D8C	23	INC	HL	NEXT BYTE
2D8D	78	LD	A,B	FORMAT SIZE REMAINING
2D8E	D604	SUB	04H	DECREASED BY 4
2D90	D8	RET	C	RETURN IF FORMAT USED UP
2D91	D1	POP	DE	CLEAR RETURN
2D92	D1	POP	DE	GET EDIT FLAG
2D93	47	LD	B,A	SAVE FORMAT SIZE REMAINING
2D94	14	INC	D	FLAG EXPONENTIAL NOTATION
2D95	23	INC	HL	NEXT BYTE

2D96	CAEBD1	JP	Z,0D1EBH	SKIP TO 2D99H MASK
2D97	EB	EX	DE,HL	RESTORE OLD POSITION
2D98	D1	POP	DE	RESTORE EDIT INFO
2D99	7A	LD	A,D	GET EDIT FLAG
2D9A	2B	DEC	HL	BACKUP
2D9B	1C	INC	E	EXPAND LEFT FOR SIGN
2D9C	E608	AND	08H	INCLUDE SIGN?
2D9E	2015	JR	NZ,2DB5H	JUMP IF SO
2DA0	1D	DEC	E	RESTORE LEFT SIZE
2DA1	78	LD	A,B	GET FORMAT SIZE LEFT
2DA2	B7	OR	A	NOTHING LEFT?
2DA3	2810	JR	Z,2DB5H	SKIP IF SO
2DA5	7E	LD	A,(HL)	GET CHARACTER.
2DA6	D62D	SUB	2DH	"-"?
2DA8	2806	JR	Z,2DB0H	SKIP OF SO
2DAA	FEFE	CP	0FEH	"+"?
2DAC	2007	JR	NZ,2DB5H	SKIP IF NOT
2DAE	3E08	LD	A,08H	INCLUDE SIGN
2DB0	C604	ADD	A,04H	TRAILING SIGN, IF ANY
2DB2	82	ADD	A,D	UPDATE
2DB3	57	LD	D,A	EDITING FLAG
2DB4	05	DEC	B	DECREASING FORMAT SIZE LEFT
2DB5	E1	POP	HL	RESTORE FORMAT SIZE LEFT
2DB6	F1	POP	AF	NO MORE DATA TO PRINT?
2DB7	2850	JR	Z,2E09H	JUMP IF NOT
2DB9	C5	PUSH	BC	SAVE FORMATTING INFO
2DBA	D5	PUSH	DE	
2DBB	CD3723	CALL	2337H	EVALUATE EXPRESSION
2DBE	D1	POP	DE	RESTORE
2DBF	C1	POP	BC	FORMATTING INFO
2DC0	C5	PUSH	BC	SAVE FORMAT SIZE LEFT
2DC1	E5	PUSH	HL	SAVE TEXT POSITION
2DC2	43	LD	B,E	B = # CHARACTERS LEFT OF DECIMAL
2DC3	78	LD	A,B	COMPUTE
2DC4	81	ADD	A,C	TOTAL FIELD SIZE
2DC5	FE19	CP	19H	25 OR MORE CHARACTERS?
2DC7	D24A1E	JP	NC,1E4AH	FC ERROR IF SO
2DCA	7A	LD	A,D	PUT EDIT FLAG IN A
2DCB	F680	OR	80H	COMMAND FORMATTING
2DCD	CDBE0F	CALL	0FBEH	CONVERT ACCUM TO ASCII
2DD0	CDA728	CALL	28A7H	OUTPUT FORMATTED NUMBER
2DD3	E1	POP	HL	RESTORE TEXT POSITION
2DD4	2B	DEC	HL	BACKUP
2DD5	D7	RST	10H	GET NEXT CHARACTER.
2DD6	37	SCF		REQUEST CARRIAGE RETURN
2DD7	280D	JR	Z,2DE6H	SKIP IF END OF STATEMENT
2DD9	32DE40	LD	(40DEH),A	SAVE CHARACTER
2DDC	FE3B	CP	3BH	SEMICOLON?
2DDE	2805	JR	Z,2DE5H	SKIP IF SO.
2DE0	FE2C	CP	2CH	COMMA?
2DE2	C29719	JP	NZ,1997H	SYNTAX ERROR IF NOT.
2DE5	D7	RST	10H	GET NEXT CHARACTER.
2DE6	C1	POP	BC	RESTORE FORMAT SIZE LEFT
2DE7	EB	EX	DE,HL	SAVE POSITION IN TEXT
2DE8	E1	POP	HL	RESTORE & SAVE
2DE9	E5	PUSH	HL	FORMAT STRING POINTER
2DEA	F5	PUSH	AF	SAVE TEXT CHARACTER

2DEB D5	PUSH	DE	SAVE POSITION IN TEXT
2DEC 7E	LD	A, (HL)	GET FORMAT STRING LENGTH
2DED 90	SUB	B	A = OFFSET INTO FORMAT
2DEE 23	INC	HL	HL
2DEF 4E	LD	C, (HL)	=>
2DF0 23	INC	HL	FORMAT
2DF1 66	LD	H, (HL)	STRING
2DF2 69	LD	L, C	
2DF3 1600	LD	D, 00H	DE =
2DF5 5F	LD	E, A	OFFSET INTO FORMAT
2DF6 19	ADD	HL, DE	HL => CURRENT POSITION IN FORMAT
2DF7 78	LD	A, B	ANYTHING LEFT TO USE
2DF8 B7	OR	A	IN FORMAT STRING?
2DF9 C2032D	JP	NZ, 2D03H	JUMP IF SO
2DFC 1806	JR	2E04H	SKIP IF NOT
2DFE CD492E	CALL	2E49H	DISPLAY "+" IF REQUESTED
2E01 CD2A03	CALL	032AH	DISPLAY CHARACTER
2E04 E1	POP	HL	GET POSITION IN TEXT
2E05 F1	POP	AF	GET CURRENT CHARACTER
2E06 C2CB2C	JP	NZ, 2CCBH	RE-USE FORMAT IF NOT DONE
2E09 DCFE20	CALL	C, 20FEH	CARRIAGE RETURN IF REQUESTED
2E0C E3	EX	(SP), HL	SAVE POSITION, GET SIRING POINTER
2E0D CDD29	CALL	29DDH	CLEAN UP WORKSPACE
2E10 E1	POP	HL	RESTORE POSITION IN TEXT
2E11 C36921	JP	2169H	RESET I/O & RETURN
2E14 0E01	LD	C, 01H	ONE CHARACTER OUTPUT FIELD
2E16 3EF1	LD	A, 0F1H	SKIP TO 2E18H
2E17 F1	POP	AF	CLEAR STACK
2E18 05	DEC	B	ONE LESS CHARACTER IN FORMAT
2E19 CD492E	CALL	2E49H	DISPLAY "+" IF REQUESTED
2E1C E1	POP	HL	RESTORE POSITION IN TEXT
2E1D F1	POP	AF	MORE DATA TO PRINT
2E1E 28E9	JR	Z, 2E09H	SKIP OUT IF NOT
2E20 C5	PUSH	BC	SAVE SIZE OF FORMAT LEFT
2E21 CD3723	CALL	2337H	EVALUATE EXPRESSION
2E24 CDF40A	CALL	0AF4H	TM ERROR IF NOT STRING
2E27 C1	POP	BC	RESTORE & SAVE SIZE
2E28 C5	PUSH	BC	OF FORMAT LEFT
2E29 E5	PUSH	HL	SAVE POSITION IN TEXT
2E2A 2A2141	LD	HL, (4121H)	HL => STRING DESCRIPTOR
2E2D 41	LD	B, C	B = # CHARACTERS IN FIELD
2E2E 0E00	LD	C, 00H	OFFSET = 0
2E30 C5	PUSH	BC	SAVE FIELD SIZE
2E31 CD682A	CALL	2A68H	GET LEFT\$
2E34 CDAA28	CALL	28AAH	OUTPUT FIELD
2E37 2A2141	LD	HL, (4121H)	HL => STRING DESCRIPTOR
2E3A F1	POP	AF	RESTORE FIELD SIZE
2E3B 96	SUB	(HL)	SUBTRACT PART ALREADY OUTPUT
2E3C 47	LD	B, A	B = # SPACES TO DISPLAY
2E3D 3E20	LD	A, 20H	CHAR = SPACE
2E3F 04	INC	B	INCREASE # SPACES BY ONE
2E40 05	DEC	B	ONE LESS SPACE TO DISPLAY
2E41 CAD32D	JP	Z, 2DD3H	JUMP IF DONE
2E44 CD2A03	CALL	032AH	DISPLAY
2E47 18F7	JR	2E40H	LOOP UNTIL DONE
2E49 F5	PUSH	AF	SAVE STATUS
2E4A 7A	LD	A, D	GET D

2E4B B7	OR	A	TEST IT
2E4C 3E2B	LD	A, 2BH	A = "+"
2E4E C42A03	CALL	NZ, 032AH	PRINT IF NON-ZERO
2E51 F1	POP	AF	RESTORE STATUS
2E52 C9	RET		DONE

RESULT:

PRINT "+" IF D<>0.

SYNTAX ERROR EDIT

2E53 329A40	LD	(409AH), A	ZERO ERR
2E56 2AEA40	LD	HL, (40EAH)	GET ERL
2E59 B4	OR	H	DIRECT
2E5A A5	AND	L	MODE?
2E5B 3C	INC	A	
2E5C EB	EX	DE, HL	DE = ERL
2E5D C8	RET	Z	RETURN IF DIRECT MODE
2E5E 1804	JR	2E64H	EDIT LINE

EDIT

2E60 CD4F1E	CALL	1E4FH	GET LINE # TO EDIT
2E63 C0	RET	NZ	RETURN IF MORE PARAMS
2E64 E1	POP	HL	CLEAR STACK
2E65 EB	EX	DE, HL	HL = LINE #
2E66 22EC40	LD	(40ECH), HL	SAVE LINE #
2E69 EB	EX	DE, HL	DE = LINE #
2E6A CD2C1B	CALL	1B2CH	FIND LINE POSITION
2E6D D2D91E	JP	NC, 1ED9H	UL ERROR IF NOT FOUND
2E70 60	LD	H, B	HL => LINE
2E71 69	LD	L, C	
2E72 23	INC	HL	SKIP PAST
2E73 23	INC	HL	FORWARD POINTER
2E74 4E	LD	C, (HL)	BC = LINE #
2E75 23	INC	HL	
2E76 46	LD	B, (HL)	
2E77 23	INC	HL	
2E78 C5	PUSH	BC	SAVE LINE #
2E79 CD7E2B	CALL	2B7EH	EXPAND LINE TO BUFFER
2E7C E1	POP	HL	GET & SAVE
2E7D E5	PUSH	HL	LINE #
2E7E CD4F0F	CALL	0FAFH	DISPLAY LINE #
2E81 3E20	LD	A, 20H	DISPLAY
2E83 CD2A03	CALL	032AH	A SPACE
2E86 2AA740	LD	HL, (40A7H)	HL => BUFFER
2E89 3E0E	LD	A, 0EH	TURN
2E8B CD2A03	CALL	032AH	ON CURSOR
2E8E E5	PUSH	HL	SAVE BUFFER START
2E8F 0EFF	LD	C, 0FFH	LENGTH = -1
2E91 0C	INC	C	LENGTH = LENGTH + 1
2E92 7E	LD	A, (HL)	END OF
2E93 B7	OR	A	LINE?
2E94 23	INC	HL	NEXT CHARACTER
2E95 20FA	JR	NZ, 2E91H	LOOP IF NOT
2E97 E1	POP	HL	GET BUFFER START
2E98 47	LD	B, A	COLUMN = 0
2E99 1600	LD	D, 00H	N = 0
2E9B CD8403	CALL	0384H	GET COMMAND

2E9E D630	SUB	30H	ADJUST VALUE
2EA0 380E	JR	C,2EB0H	JUMP IF NOT NUMERIC
2EA2 FE0A	CP	0AH	NUMERIC?
2EA4 300A	JR	NC,2EB0H	JUMP IF NOT NUMERIC
2EA6 5F	LD	E,A	N
2EA7 7A	LD	A,D	=
2EA8 07	RLCA		N
2EA9 07	RLCA		*
2EAA 82	ADD	A,D	10
2EAB 07	RLCA		+
2EAC 83	ADD	A,E	VALUE
2EAD 57	LD	D,A	SAVE N
2EAE 18EB	JR	2E9BH	LOOP
2EB0 E5	PUSH	HL	SET RETURN
2EB1 21992E	LD	HL,2E99H	THROUGH
2EB4 E3	EX	(SP),HL	2E99H
2EB5 15	DEC	D	N = 0?
2EB6 14	INC	D	
2EB7 C2BB2E	JP	NZ,2EBBH	SKIP IF NOT
2EBA 14	INC	D	N = 1
2EBB FED8	CP	0D8H	BACKSPACE?
2EBD CAD22F	JP	Z,2FD2H	JUMP IF SO
2EC0 FEDD	CP	0DDH	CARRIAGE RETURN?
2EC2 CAE02F	JP	Z,2FE0H	JUMP IF SO
2EC5 FEF0	CP	0F0H	SPACE?
2EC7 2841	JR	Z,2F0AH	JUMP IF SO
2EC9 FE31	CP	31H	UPPER CASE?
2ECB 3802	JR	C,2ECFH	SKIP IF SO
2ECD D620	SUB	20H	MAKE UPPER CASE
2ECF FE21	CP	21H	"O"?
2ED1 CAF62F	JP	Z,2FF6H	JUMP IF SO
2ED4 FE1C	CP	1CH	"L"?
2ED6 CA402F	JP	Z,2F40H	JUMP IF SO
2ED9 FE23	CP	23H	"S"?
2EDB 283F	JR	Z,2F1CH	JUMP IF SO
2EDD FE19	CP	19H	"I"?
2EDF CA7D2F	JP	Z,2F7DH	JUMP IF SO
2EE2 FE14	CP	14H	"D"?
2EE4 CA4A2F	JP	Z,2F4AH	JUMP IF SO
2EE7 FE13	CP	13H	"C"?
2EE9 CA652F	JP	Z,2F65H	JUMP IF SO
2EEC FE15	CP	15H	"E"?
2EEE CAE32F	JP	Z,2FE3H	JUMP IF SO
2EF1 FE28	CP	28H	"X"?
2EF3 CA782F	JP	Z,2F78H	JUMP IF SO
2EF6 FE1B	CP	1BH	"K"?
2EF8 281C	JR	Z,2F16H	JUMP IF SO
2EFA FE18	CP	18H	"H"?
2EFC CA752F	JP	Z,2F75H	JUMP IF SO
2EFF FE11	CP	11H	"A"?
2F01 C0	RET	NZ	RETURN IF NOT

"A" ROUTINE

2F02 C1	POP	BC	CLEAR RETURN
2F03 D1	POP	DE	GET LINE #
2F04 CDFE20	CALL	20FEH	NEW LINE ON DISPLAY
2F07 C3652E	JP	2E65H	DISPLAY LINE # & EDIT

SPACE ROUTINE

2F0A	7E	LD	A, (HL)	GET CHARACTER
2F0B	B7	OR	A	END OF LINE?
2F0C	C8	RET	Z	RETURN IF SO
2F0D	04	INC	B	NEXT POSITION IN LINE
2F0E	CD2A03	CALL	032AH	DISPLAY CHARACTER
2F11	23	INC	HL	MOVE UP IN LINE
2F12	15	DEC	D	ONE LESS SPACE TO DO
2F13	20F5	JR	NZ, 2F0AH	LOOP UNTIL DONE
2F15	C9	RET		DONE

"K" ROUTINE

2F16	E5	PUSH	HL	SET
2F17	215F2F	LD	HL, 2F5FH	RETURN
2F1A	E3	EX	(SP), HL	THROUGH 2F5FH
2F1B	37	SCF		SET DELETE FLAG

"S" ROUTINE

2F1C	F5	PUSH	AF	SAVE DELETE FLAG
2F1D	CD8403	CALL	0384H	GET SEARCH CHARACTER.
2F20	5F	LD	E, A	SAVE CHARACTER
2F21	F1	POP	AF	GET & SAVE
2F22	F5	PUSH	AF	DELETE FLAG
2F23	DC5F2F	CALL	C, 2F5FH	DISPLAY "!" IF DELETING
2F26	7E	LD	A, (HL)	GET NEXT CHARACTER.
2F27	B7	OR	A	END OF LINE?
2F28	CA3E2F	JP	Z, 2F3EH	JUMP IF SO
2F2B	CD2A03	CALL	032AH	DISPLAY CHARACTER
2F2E	F1	POP	AF	GET
2F2F	F5	PUSH	AF	DELETE FLAG
2F30	DCA12F	CALL	C, 2FA1H	DELETE CHARACTER IF NECESSARY
2F33	3802	JR	C, 2F37H	SKIP IF DELETING
2F35	23	INC	HL	MOVE FORWARD ONE BYTE
2F36	04	INC	B	INCREASE POSITION #
2F37	7E	LD	A, (HL)	GET CHARACTER
2F38	BB	CP	E	SEARCH CHARACTER?
2F39	20EB	JR	NZ, 2F26H	CONTINUE SEARCH IF NOT
2F3B	15	DEC	D	ONE LESS TO SEARCH FOR
2F3C	20E8	JR	NZ, 2F26H	SEARCH N TIMES
2F3E	F1	POP	AF	CLEAR STACK
2F3F	C9	RET		DONE

"L" ROUTINE

2F40	CD752B	CALL	2B75H	DISPLAY REST OF LINE
2F43	CDFE20	CALL	20FEH	NEW LINE ON DISPLAY
2F46	C1	POP	BC	CLEAR RETURN
2F47	C37C2E	JP	2E7CH	DISPLAY LINE # & CONTINUE

"D" ROUTINE

2F4A	7E	LD	A, (HL)	AT END OF
2F4B	B7	OR	A	OF LINE?
2F4C	C8	RET	Z	RETURN IF SO
2F4D	3E21	LD	A, 21H	DISPLAY"!"
2F4F	CD2A03	CALL	032AH	
2F52	7E	LD	A, (HL)	AT END OF
2F53	B7	OR	A	LINE ?

2F54	2809	JR	Z, 2F5FH	SKIP OUT IF SO
2F56	CD2A03	CALL	032AH	DISPLAY CHARACTER.
2F59	CDA12F	CALL	2FA1H	DELETE CHARACTER.
2F5C	15	DEC	D	ONE LESS REPEAT
2F5D	20F3	JR	NZ, 2F52H	REPEAT N TIMES
2F5F	3E21	LD	A, 21H	DISPLAY "!"
2F61	CD2A03	CALL	032AH	
2F64	C9	RET		DONE

"C" ROUTINE

2F65	7E	LD	A, (HL)	AT END
2F66	B7	OR	A	OF LINE
2F67	C8	RET	Z	RETURN IF SO
2F68	CD8403	CALL	0384H	GET NEW CHARACTER
2F6B	77	LD	(HL), A	SAVE NEW CHARACTER
2F6C	CD2A03	CALL	032AH	DISPLAY CHARACTER
2F6F	23	INC	HL	MOVE FORWARD IN LINE
2F70	04	INC	B	INCREASE COLUMN #
2F71	15	DEC	D	ONE LESS REPEAT
2F72	20F1	JR	NZ, 2F65H	REPEAT # TIMES
2F74	C9	RET		DONE

"H" ROUTINE

2F75	3600	LD	(HL), 00H	MARK END OF LINE
2F77	48	LD	C, B	SAVE NEW LINE LENGTH

"X" ROUTINE

2F78	16FF	LD	D, 0FFH	SET INDEFINITE REPEAT
2F7A	CD0A2F	CALL	2F0AH	SPACE TO END OF LINE

"I" ROUTINE

2F7D	CD8403	CALL	0384H	GET CHARACTER TO INSERT
2F80	B7	OR	A	NULL?
2F81	CA7D2F	JP	Z, 2F7DH	LOOP IF SO
2F84	FE08	CP	08H	BACKSPACE?
2F86	280A	JR	Z, 2F92H	JUMP IF SO
2F88	FE0D	CP	0DH	CARRIAGE RETURN?
2F8A	CAE02F	JP	Z, 2FE0H	JUMP IF SO
2F8D	FE1B	CP	1BH	ESCAPE?
2F8F	C8	RET	Z	RETURN IF SO
2F90	201E	JR	NZ, 2FB0H	JUMP IF ANY OTHER CHARACTER.
2F92	3E08	LD	A, 08H	BACKSPACE CHARACTER
2F94	05	DEC	B	AT BEGINNING
2F95	04	INC	B	OF LINE?
2F96	281F	JR	Z, 2FB7H	SKIP IF SO
2F98	CD2A03	CALL	032AH	BACKSPACE CURSOR
2F9B	2B	DEC	HL	MOVE BACK IN LINE
2F9C	05	DEC	B	DECREASE COLUMN #
2F9D	117D2F	LD	DE, 2F7DH	RETURN THROUGH
2FA0	D5	PUSH	DE	2F7DH (INSERT)
2FA1	E5	PUSH	HL	SAVE POSITION
2FA2	0D	DEC	C	DECREASE SIZE OF LINE
2FA3	7E	LD	A, (HL)	AT END
2FA4	B7	OR	A	OF LINE?
2FA5	37	SCF		RESTORE CARRY FLAG
2FA6	CA9008	JP	Z, 0890H	RETURN IF SO
2FA9	23	INC	HL	GET NEXT CHARACTER

2FAA 7E	LD	A, (HL)	
2FAB 2B	DEC	HL	SAVE AT
2FAC 77	LD	(HL), A	CURRENT POSITION
2FAD 23	INC	HL	MOVE UP
2FAE 18F3	JR	2FA3H	LOOP TO END OF LINE
2FB0 F5	PUSH	AF	SAVE STATUS
2FB1 79	LD	A, C	GET LINE LENGTH
2FB2 FEFF	CP	0FFH	LINE FULL?
2FB4 3803	JR	C, 2FB9H	JUMP IF NOT
2FB6 F1	POP	AF	RESTORE STATUS
2FB7 18C4	JR	2F7DH	LOOP BACK NO INSERT
2FB9 90	SUB	B	COMPUTE # BYTES TO MOVE
2FBA 0C	INC	C	INCREASE LINE SIZE
2FBB 04	INC	B	NEXT COLUMN
2FBC C5	PUSH	BC	SAVE LINE SIZE & COLUMN
2FBD EB	EX	DE, HL	SAVE POSITION IN LINE
2FBE 6F	LD	L, A	COMPUTE ENDING
2FBF 2600	LD	H, 00H	ADDRESS OF BLOCK MOVE
2FC1 19	ADD	HL, DE	HL => END OF DESTINATION BLOCK
2FC2 44	LD	B, H	DE => START OF SOURCE BLOCK
2FC3 4D	LD	C, L	BC => END OF SOURCE BLOCK
2FC4 23	INC	HL	
2FC5 CD5819	CALL	1958H	MOVE DATA BLOCK OVER
2FC8 C1	POP	BC	RESTORE SIZE & COLUMN
2FC9 F1	POP	AF	GET CHARACTER
2FCA 77	LD	(HL), A	SAVE CHARACTER
2FCB CD2A03	CALL	032AH	DISPLAY CHARACTER
2FCE 23	INC	HL	MOVE TO NEXT POSITION
2FCF C37D2F	JP	2F7DH	LOOP BACK

BACKSPACE ROUTINE

2FD2 78	LD	A, B	AT START
2FD3 B7	OR	A	OF LINE
2FD4 C8	RET	Z	RETURN IF SO
2FD5 05	DEC	B	DECREASE COLUMN #
2FD6 2B	DEC	HL	BACKUP IN BUFFER
2FD7 3E08	LD	A, 08H	BACKSPACE
2FD9 CD2A03	CALL	032AH	CURSOR
2FDC 15	DEC	D	ONE LESS REPEAT
2FDD 20F3	JR	NZ, 2FD2H	REPEAT N TIMES
2FDF C9	RET		DONE

CARRIAGE RETURN ROUTINE

2FE0 CD752B	CALL	2B75H	DISPLAY REMAINDER OF LINE
-------------	------	-------	---------------------------

"E" ROUTINE

2FE3 CDFE20	CALL	20FEH	NEW LINE ON DISPLAY
2FE6 C1	POP	BC	CLEAR RETURN
2FE7 D1	POP	DE	GET LINE #
2FE8 7A	LD	A, D	DIRECT
2FE9 A3	AND	E	MOVE ?
2FEA 3C	INC	A	
2FEB 2AA740	LD	HL, (40A7H)	HL => JUST BEFORE
2FEE 2B	DEC	HL	BUFFER
2FEF C8	RET	Z	RETURN IF DIRECT MODE
2FF0 37	SCF		SET CARRY FOR LATER
2FF1 23	INC	HL	HL => BUFFER

2FF2 F5	PUSH	AF	SAVE CARRY FLAG
2FF3 C3981A	JP	1A98H	SAVE LINE & EXIT

"O" ROUTINE

2FF6 C1	POP	BC	CLEAR STACK
2FF7 D1	POP	DE	GET LINE #
2FF8 C3191A	JP	1A19H	GO TO READY

2FFB DEC3C344B2			UNUSED GARBAGE.
-----------------	--	--	-----------------

SYSTEM ROM

ADDR

3000	C35E32	JP	325EH	SLOW TAPE HEADER WRITE.
3003	C39B32	JP	329BH	FAST TAPE HEADER WRITE.
3006	C37432	JP	3274H	SLOW TAPE HEADER READ.
3009	C3DA32	JP	32DAH	FAST TAPE HEADER READ.
300C	C3C031	JP	31C0H	CASSETTE OFF.
300F	C3D131	JP	31D1H	CASSETTE ON.
3012	C3AB34	JP	34ABH	WARM BOOT
3015	C35534	JP	3455H	BOOTSTRAP
3018	C3C235	JP	35C2H	MASKABLE INTERRUPT HANDLER
301B	C3FB35	JP	35FBH	RS-232 INITIALIZATION
301E	C35A36	JP	365AH	RS-232 INPUT
3021	C38036	JP	3680H	RS-232 OUTPUT
3024	C38E33	JP	338EH	KEYBOARD INPUT
3027	C33937	JP	3739H	I/O RE-ROUTER
302A	C3F731	JP	31F7H	PART OF CASSETTE HEADER ROUTINE
302D	C37B37	JP	377BH	
3030	C39937	JP	3799H	STRING=DATE\$+" "+TIME\$
3033	C3BB35	JP	35BBH	\$DATE
3036	C3A035	JP	35A0H	\$TIME
3039	DBE4	IN	A, (0E4H)	GET NON-MASKABLE INTERRUPT STATUS
303B	CB6F	BIT	05H, A	CHECK IT.
303D	C31C35	JP	351CH	GO TO NMI HANDLER
3040	18D3	JR	3015H	BOOTSTRAP
3042	C3B537	JP	37B5H	\$SETCAS
3045	406162636465666768696A6B6C6D6E			KEYBOARD ROWS 0-3 UNSHIFTED NO CAPS LOCK.
	----			6F707172737475767778797A
3060	3AEA37	LD	A, (37EAH)	CHECK COMPUTER VERSION.
3063	B7	OR	A	
3064	C9	RET		
3065	303132333435363738393A3B3C3D3E			KEYBOARD ROWS 4-6 UNSHIFTED NO CAPS LOCK.
	----			2F0D1F015B0A080920
307D	21DC05	LD	HL, 05DCH	
307E	22FF41	LD	(41FFH), HL	
3083	AF	XOR	A	NULL CHARACTER RETURN
3084	C9	RET		RETURN
3085	604142434445464748494A4B4C4D4E			KEYBOARD ROWS 0-3 SHIFTED NO CAPS LOCK.
	----			4F505152535455565758595A
30A0	77	LD	(HL), A	SAVE FLAG
30A1	AF	XOR	A	NULL CHARACTER
30A2	C9	RET		RETURN
30A3	AAAA			UNUSED GARBAGE
30A5	002122232425262728292A2B3C3D3E			KEYBOARD ROWS 4-6 SHIFTED NO CAPS LOCK
	----			3F0D1F011B1A181920
30BD	3E01	LD	A, 01H	TOGGLE CAPS LOCK
30BE	211940	LD	HL, 4019H	
30C2	AE	XOR	(HL)	
30C3	18DB	JR	30A0H	

30C5 404142434445464748494A4B4C4D4E KEYBOARD ROWS 0-3 UNSHIFTED CAPS LOCK.
---- 4F505152535455565758595A

30E0 CDD901 CALL 01D9H SCREEN PRINT
30E3 AF XOR A NULL CHARACTER
30E4 C9 RET RETURN

30E5 303132333435363738393A3B2C2D2E KEYBOARD ROWS 4-6 UNSHIFTED CAPS LOCK.
---- 2F0D1F015B0A080920

30FD 28E1 JR Z,30E0H SCREEN PRINT IF MATCH
30FF A6 AND (HL) BREAK?
3100 FE01 CP 01H
3102 C0 RET NZ RETURN IF NOT.
3103 EF RST 28H CALL BREAK HANDLER
3104 C9 RET DONE

3105 604142434445464748494A4B4C4D4E KEYBOARD ROWS 0-3 SHIFTED CAPS LOCK.
---- 4F505152535455565758595A

3120 14 INC D NEXT ROW NUMBER
3121 23 INC HL NEXT BUFFER LOCATION.
3122 CB01 RLC C NEXT ROW OF KEYS
3124 C9 RET CONTINUE.

3125 002122232425262728292A2B3C3D3E KEYBOARD ROWS 4-6 SHIFTED CAPS LOCK.
---- 3F0D1F011B1A181920

313D 3AFD41 LD A, (41FDH)
3140 6F LD L,A
3141 3AFE41 LD A, (41FEH)
3144 C9 RET

3145 202122232425262728292A2B2C2D2E TABLE OF CHARACTERS FOR PRINTER
---- 2F303132333435363738393A3B3C3D CODES 32 - 127
---- 3E3F404142434445464748494A4B4C
---- 4D4E4F505152535455565758595A5B
---- 5C5D5E5F406162636465666768696A
---- 6B6C6D6E6F70717273747576777879
---- 7A7B7C7D7E7F

31A5 3E01 LD A,01H 0.85 V TO TAPE
31A7 D3FF OUT (0FFH),A
31A9 060D LD B,0DH DELAY
31AB 10FE DJNZ 31ABH
31AD 3E02 LD A,02H 0 V TO TAPE
31AF D3FF OUT (0FFH),A
31B1 060D LD B,0DH DELAY
31B3 10FE DJNZ 31B3H
31B5 CDF331 CALL 31F3H 0.46 V TO TAPE
31B8 0678 LD B,78H DELAY
31BA 10FE DJNZ 31BAH
31BC C9 RET DONE
31BD 21A52C LD HL,2CA5H
31C0 3A1342 LD A,(4213H)
31C3 D3E0 OUT (0E0H),A

31C5	DBFF	IN	A, (0FFH)	
31C7	3A1042	LD	A, (4210H)	GET MASK FOR PORT EC.
31CA	E6FD	AND	0FDH	
31CC	CDED31	CALL	31EDH	
31CF	FB	EI		
31D0	C9	RET		
31D1	EB	EX	DE, HL	REMOVE RETURN ADDRESS,
31D2	E3	EX	(SP), HL	SAVE DE & BC,
31D3	C5	PUSH	BC	RESTORE RETURN ADDRESS
31D4	E5	PUSH	HL	WHEN DONE
31D5	EB	EX	DE, HL	
31D6	DBEC	IN	A, (0ECH)	
31D8	112020	LD	DE, 2020H	BLANK BLINKERS
31DB	ED533E3C	LD	(3C3EH), DE	
31DF	CDE831	CALL	31E8H	TURN ON CASSETTE.
31E2	01007D	LD	BC, 7D00H	DELAY COUNT
31E5	C36000	JP	0060H	WAIT A BIT & RETURN
31E8	3A1042	LD	A, (4210H)	GET MASK FOR EC.
31EB	F602	OR	02H	SET BIT 2
31ED	321042	LD	(4210H), A	SAVE MASK
31F0	D3EC	OUT	(0ECH), A	OUTPUT MASK
31F2	C9	RET		DONE
31F3	AF	XOR	A	RESET CASSETTE PORT.
31F4	D3FF	OUT	(0FFH), A	
31F6	C9	RET		
31F7	7E	LD	A, (HL)	SEE IF CALLER WAS
31F8	D623	SUB	23H	PRINT #
31FA	C25302	JP	NZ, 0253H	JUMP IF NOT
31FD	CD012B	CALL	2B01H	GET DEVICE NUMBER.
3200	CF	RST	08H	SYNTAX ERROR IF
3201	2C	" , "		NEXT CHARACTER NOT COMMA.
3202	C9	RET		DONE.
3203	0608	LD	B, 08H	READ
3205	CD2032	CALL	3220H	EIGHT BITS
3208	10FB	DJNZ	3205H	INTO D.
320A	3A1242	LD	A, (4212H)	INCREMENT BYTE COUNT
320D	3C	INC	A	
320E	E65F	AND	5FH	MASK TOP 3 BITS
3210	321242	LD	(4212H), A	SAVE NEW COUNT
3213	2008	JR	NZ, 321DH	SKIP IF COUNT <>0.
3215	3A3F3C	LD	A, (3C3FH)	TWINKLE,
3218	EE0A	XOR	0AH	TWINKLE,
321A	323F3C	LD	(3C3FH), A	LITTLE ASTERISK (!?)
321D	7A	LD	A, D	PUT BYTE IN A
321E	1878	JR	3298H	RESTORE REGISTERS & RETURN
3220	C5	PUSH	BC	SAVE BC
3221	DBFF	IN	A, (0FFH)	CHECK CASSETTE LEVEL
3223	17	RLA		PUT LEVEL IN CARRY
3224	3808	JR	C, 322EH	JUMP IF TIMING MARK FOUND.
3226	CD8D02	CALL	028DH	CHECK FOR <BREAK>.
3229	28F6	JR	Z, 3221H	LOOP IF NOT PRESSED.
322B	C35C33	JP	335CH	EXIT IF PRESSED.
322E	066E	LD	B, 6EH	DELAY: WAIT FOR END
3230	10FE	DJNZ	3230H	OF TIMING MARK.
3232	CDF331	CALL	31F3H	RESET CASSETTE PORT
3235	0698	LD	B, 98H	DELAY: WAIT FOR WHEN
3237	10FE	DJNZ	3237H	DATA PULSE SHOULD BE.

3239	DBFF	IN	A, (0FFH)	GET DATA LEVEL
323B	C1	POP	BC	RESTORE BC
323C	17	RLA		PUT DATA INTO CARRY.
323D	CB12	RL	D	SHIFT DATA INTO D.
323F	18B2	JR	31F3H	RESET PORT & RETURN.
3241	F5	PUSH	AF	SAVE REGISTERS
3242	C5	PUSH	BC	
3243	D5	PUSH	DE	
3244	0E08	LD	C, 08H	WRITE 8 BITS.
3246	57	LD	D, A	D=DATA BYTE
3247	CDA531	CALL	31A5H	OUTPUT TIMING MARK
324A	CB02	RLC	D	ROTATE BIT INTO CARRY
324C	300A	JR	NC, 3258H	NO PULSE IF 0.
324E	CDA531	CALL	31A5H	OUTPUT DATA PULSE
3251	0D	DEC	C	COUNT DOWN BITS LEFT.
3252	20F3	JR	NZ, 3247H	LOOP WHILE SOME LEFT.
3254	D1	POP	DE	RESTORE REGISTERS
3255	C1	POP	BC	
3256	F1	POP	AF	
3257	C9	RET		DONE
3258	069A	LD	B, 9AH	OUTPUT NO PULSE
325A	10FE	DJNZ	325AH	SIMPLY WAIT
325C	18F3	JR	3251H	
325E	E5	PUSH	HL	SAVE HL
325F	214132	LD	HL, 3241H	VECTOR TO SLOW WRITE
3262	220C42	LD	(420CH), HL	
3265	0653	LD	B, 53H	OUTPUT 63 ZEROES
3267	AF	XOR	A	
3268	CD4132	CALL	3241H	
326B	10FB	DJNZ	3268H	
326D	3EA5	LD	A, 0A5H	OUTPUT SYNC BYTE
326F	CD4132	CALL	3241H	
3272	1823	JR	3297H	RESTORE REGISTERS & RETURN
3274	E5	PUSH	HL	SAVE HL
3275	210332	LD	HL, 3203H	VECTOR TO SLOW READ
3278	220E42	LD	(420EH), HL	
327B	0640	LD	B, 40H	TRY TO FIND 64 ZEROES
327D	1600	LD	D, 00H	SET BYTE TO 0
327F	CD2032	CALL	3220H	READ BIT
3282	7A	LD	A, D	CHECK CURRENT BYTE.
3283	B7	OR	A	
3284	20F5	JR	NZ, 327BH	RE-START IF NON-ZERO
3286	10F7	DJNZ	327FH	CONTINUE UNTIL ENOUGH FOUND.
3288	CD2032	CALL	3220H	READ BIT
328B	7A	LD	A, D	CHECK CURRENT BYTE.
328C	FEA5	CP	0A5H	SYNC BYTE?
328E	20F8	JR	NZ, 3288H	LOOP IF NOT.
3290	212A2A	LD	HL, 2A2AH	DISPLAY "***"
3293	223E3C	LD	(3C3EH), HL	
3296	7C	LD	A, H	PUT "*" IN A
3297	E1	POP	HL	RESTORE REGISTERS
3298	C1	POP	BC	
3299	D1	POP	DE	
329A	C9	RET		DONE
329B	E5	PUSH	HL	SAVE HL
329C	21BA32	LD	HL, 32BAH	VECTOR TO FAST WRITE
329F	220C42	LD	(420CH), HL	

32A2	0600	LD	B,00H	OUTPUT 256 "55" BYTES
32A4	3E55	LD	A,55H	
32A6	CDB432	CALL	32B4H	
32A9	10F9	DJNZ	32A4H	
32AB	3E7F	LD	A,7FH	OUTPUT SYNC BYTE.
32AD	CDB432	CALL	32B4H	
32B0	3EA5	LD	A,0A5H	A = SLOW SYNC BYTE
32B2	18E3	JR	3297H	RESTORE REGISTERS & RETURN
32B4	F5	PUSH	AF	SAVE REGISTERS
32B5	C5	PUSH	BC	
32B6	D5	PUSH	DE	
32B7	4F	LD	C,A	PUT BYTE IN C.
32B8	1807	JR	32C1H	WRITE, NO START BIT.
32BA	F5	PUSH	AF	SAVE REGISTERS
32BB	C5	PUSH	BC	
32BC	D5	PUSH	DE	
32BD	4F	LD	C,A	PUT BYTE IN C.
32BE	CD3E33	CALL	333EH	OUTPUT START BIT
32C1	0608	LD	B,08H	8 BITS
32C3	CD3533	CALL	3335H	OUTPUT BIT
32C6	10FB	DJNZ	32C3H	LOOP UNTIL DONE.
32C8	188A	JR	3254H	RESTORE REGISTERS & RETURN
32CA	CD5033	CALL	3350H	READ START BIT
32CD	0608	LD	B,08H	READ EIGHT BITS
32CF	CD5033	CALL	3350H	READ BIT
32D2	CD7C33	CALL	337CH	CHECK FOR DATA ERROR
32D5	10F8	DJNZ	32CFH	LOOP FOR BYTE
32D7	C30A32	JP	320AH	CONTINUE
32DA	E5	PUSH	HL	SAVE HL
32DB	21CA32	LD	HL,32CAH	VECTOR TO
32DE	220E42	LD	(420EH),HL	FAST READ.
32E1	3E01	LD	A,01H	SET INTERRUPT
32E3	D3E0	OUT	(0E0H),A	VECTOR.
32E5	0680	LD	B,80H	TRY TO READ 128 BYTES
32E7	CD5033	CALL	3350H	READ BIT
32EA	79	LD	A,C	GET PULSE WIDTH
32EB	FE0F	CP	0FH	TOO SHORT?
32ED	38F6	JR	C,32E5H	LOOP BACK IF SO.
32EF	FE3E	CP	3EH	TOO LONG?
32F1	30F2	JR	NC,32E5H	LOOP BACK IF SO.
32F3	10F2	DJNZ	32E7H	TRY 128 TIMES.
32F5	210000	LD	HL,0000H	ZERO COUNTS
32F8	0640	LD	B,40H	READ 64 BITS
32FA	CD5033	CALL	3350H	READ BIT
32FD	CD5033	CALL	3350H	READ BIT
3300	51	LD	D,C	D = DELAY COUNT
3301	CD5033	CALL	3350H	READ BIT
3304	7A	LD	A,D	FIND DIFFERENCE
3305	91	SUB	C	IN DELAYS.
3306	3002	JR	NC,330AH	GET ABSOLUTE
3308	ED44	NEG		VALUE.
330A	FE0D	CP	0DH	ONE BIT?
330C	3805	JR	C,3313H	SKIP IF SO.
330E	24	INC	H	ONE MORE ZERO BIT.
330F	10E9	DJNZ	32FAH	LOOP FOR MORE
3311	1803	JR	3316H	SKIP IF DONE.
3313	2C	INC	L	ONE MORE ONE BIT.

```

3314 10E4    DJNZ   32FAH    LOOP FOR MORE
3316 3E40    LD     A,40H    64 ZERO
3318 BC      CP     H        BITS?
3319 280A    JR     Z,3325H  JUMP IF SO.
331B BD      CP     L        64 ONE BITS?
331C 20D7    JR     NZ,32F5H JUMP IF NOT.
331E 3E02    LD     A,02H    SET INTERRUPT
3320 D3E0    OUT   (0E0H),A VECTOR.
3322 CD5033  CALL  3350H    READ BIT.
3325 1600    LD     D,00H    ZERO BYTE
3327 CD5033  CALL  3350H    READ BIT INTO D.
332A CD7C33  CALL  337CH    CHECK FOR DATA ERROR
332D 7A      LD     A,D      GET BYTE
332E FE7F    CP     7FH     MARKER BYTE?
3330 20F5    JR     NZ,3327H JUMP IF NOT.
3332 C39032  JP     3290H    CONTINUE
3335 CB01    RLC    C       SHIFT BIT INTO CARRY
3337 3005    JR     NC,333EH SKIP IF ZERO BIT.
3339 111712  LD     DE,1217H DELAYS FOR 1 BIT.
333C 1803    JR     3341H
333E 112F2B  LD     DE,2B2FH DELAYS FOR 0 BIT.
3341 15      DEC    D       DELAY #1
3342 20FD    JR     NZ,3341H
3344 3E02    LD     A,02H   0 VOLTS TO TAPE
3346 D3FF    OUT   (0FFH),A
3348 1D      DEC    E       DELAY #2
3349 20FD    JR     NZ,3348H
334B 3E01    LD     A,01H   0.85 VOLTS TO TAPE
334D D3FF    OUT   (0FFH),A
334F C9      RET
3350 FB      EI
3351 0E00    LD     C,00H   COUNT =0
3353 0C      INC    C       COUNT = COUNT + 1
3354 3A4038  LD     A,(3840H) CHECK <BREAK>
3357 E604    AND   04H
3359 28F8    JR     Z,3353H LOOP IF NOT PRESSED
335B F3      DI
335C 21424B  LD     HL,4B42H DISPLAY "BK" OVER "***"
335F 223E3C  LD     (3C3EH),HL
3362 C30342  JP     4203H   EXIT.

```

THIS ROUTINE IS EXITED VIA AND INTERRUPT
 (MASKABLE>. EXIT IS TO 3365H OR 3369H,
 ACCORDING TO HOW PREVIOUSLY SET.

```

3365 1E01    LD     E,01H   BIT GOES HIGH.
3367 1802    JR     336BH
3369 1E00    LD     E,00H   BIT GOES LOW
336B 3E06    LD     A,06H   COUNT = COUNT + 6
336D 81      ADD   A,C
336E 4F      LD     C,A
336F DBFF    IN   A,(0FFH) GET CASSETTE LEVEL
3371 E601    AND   01H
3373 BB      CP     E       COMPARE WITH GIVEN LEVEL
3374 2003    JR     NZ,3379H SKIP IF NO MATCH.
3376 F1      POP   AF      RESTORE AF
3377 F1      POP   AF      REMOTE CALLER'S ADDRESS

```

3378	C9	RET		RETURN TO CALLER'S CALLER.
3379	F1	POP	AF	RESTORE AF
337A	FB	EI		RE-ENABLE INTERRUPTS
337B	C9	RET		RETURN TO LOOP
337C	79	LD	A,C	GET COUNT
337D	FE22	CP	22H	WAS THERE 0 OR 1? (CARRY)
337F	CB12	RL	D	PUT DATA BIT INTO D.
3381	FE0F	CP	0FH	TOO QUICK?
3383	3803	JR	C,3388H	DATA ERROR IF SO
3385	FE3E	CP	3EH	TO SLOW?
3387	D8	RET	C	RETURN IF NOT.
3388	3E44	LD	A,44H	DISPLAY "D"
338A	323E3C	LD	(3C3EH),A	FOR DATA ERROR.
338D	C9	RET		DONE
338E	CD6030	CALL	3060H	LOWER CASE INSTALLED?
3391	2010	JR	NZ,33A3H	SKIP IF NOT.
3393	018038	LD	BC,3880H	BC => SHIFT KEYS
3396	211840	LD	HL,4018H	HL => FLAG BYTE
3399	0A	LD	A,(BC)	GET IMAGE OF KEYS
339A	E602	AND	02H	MASK BIT 1 (RIGHT SHIFT)
339C	5F	LD	E,A	SAVE MASKED IMAGE
339D	AE	XOR	(HL)	TOGGLE AGAINST OLD IMAGE
339E	73	LD	(HL),E	SAVE NEW IMAGE
339F	A3	AND	E	MASK BIT 1
33A0	C2BD30	JP	NZ,30BDH	JUMP IF RIGHT SHIFT PRESSED.
33A3	3EFF	LD	A,0FFH	FLAG = 0FFH = NO CONTROL
33A5	214038	LD	HL,3840H	CHECK FOR
33A8	CB66	BIT	04H,(HL)	DOWN ARROW.
33AA	2808	JR	Z,33B4H	SKIP IF NOT PRESSED
33AC	CB25	SLA	L	CHECK FOR
33AE	CB46	BIT	00H,(HL)	LEFT SHIFT.
33B0	2802	JR	Z,33B4H	SKIP IF NOT PRESSED
33B2	3E1F	LD	A,1FH	FLAG = CONTROL KEY
33B4	322442	LD	(4224H),A	SAVE "CONTROL" FLAG
33B7	010138	LD	BC,3801H	BC => KEYBOARD ROW 0
33BA	213640	LD	HL,4036H	HL => BUFFER ROW 0
33BD	1600	LD	D,00H	D = ROW 0
33BF	0A	LD	A,(BC)	CHECK ROW
33C0	5F	LD	E,A	SAVE IN E
33C1	AE	XOR	(HL)	SET CHANGED BITS
33C2	73	LD	(HL),E	SAVE SCAN.
33C3	A3	AND	E	MASK RELEASED KEYS
33C4	2032	JR	NZ,33F8H	JUMP IF ANY KEYS PRESSED
33C6	CD2031	CALL	3120H	GO TO NEXT ROW
33C9	F2BF33	JP	P,33BFH	LOOP IF ANY ROWS LEFT
33CC	CD3D31	CALL	313DH	RESTORE SCANNING VARIABLES
33CF	A6	AND	(HL)	PREVIOUS KEYS STILL PRESSED?
33D0	2008	JR	NZ,33DAH	JUMP IF SO.
33D2	ED62	SBC	HL,HL	ZERO HL.
33D4	220142	LD	(4201H),HL	CLEAR REPEAT COUNTER
33D7	C37D30	JP	307DH	FINISH & RETURN NULL.
33DA	E5	PUSH	HL	SAVE HL
33DB	2A0142	LD	HL,(4201H)	INCREMENT
33DE	23	INC	HL	REPEAT
33DF	220142	LD	(4201H),HL	COUNTER.
33E2	ED5BFF41	LD	DE,(41FFH)	SCANNED
33E6	ED52	SBC	HL,DE	ENOUGH?

33E8	D1	POP	DE	RESTORE OLD HL TO DE
33E9	DAA130	JP	C,30A1H	RETURN NULL IF NOT.
33EC	AF	XOR	A	CLEAR MASK SO
33ED	12	LD	(DE),A	KEY WILL BE RE-READ.
33EE	220142	LD	(4201H),HL	SAVE COUNTER.
33F1	2E96	LD	L,96H	FAST REPEAT COUNT.
33F3	22FF41	LD	(41FFH),HL	SAVE NEW REPEAT COUNT.
33F6	18AB	JR	33A3H	RE-SCAN KEYBOARD
33F8	5F	LD	E,A	SAME KEY PATTERN IN E.
33F9	C5	PUSH	BC	SAVE BC
33FA	01C405	LD	BC,05C4H	1/50 SECOND DELAY
33FD	CD6000	CALL	0060H	FOR DE-BOUNCE.
3400	C1	POP	BC	RESTORE BC
3401	0A	LD	A,(BC)	RE-CHECK KEYBOARD
3402	A3	AND	E	SAME PATTERN
3403	C8	RET	Z	RETURN IF NOT
3404	32FE41	LD	(41FEH),A	SAVE
3407	7D	LD	A,L	SCAN POSITION
3408	32FD41	LD	(41FDH),A	AND VALUE.
340B	7A	LD	A,D	D = 8* ROW #
340C	17	RLA		
340D	17	RLA		
340E	17	RLA		
340F	57	LD	D,A	
3410	7B	LD	A,E	
3411	0F	RRCA		D = 8* ROW # + KEY #
3412	3803	JR	C,3417H	
3414	14	INC	D	
3415	18FA	JR	3411H	
3417	CD6030	CALL	3060H	DUAL SHIFTS?
341A	3A8038	LD	A,(3880H)	GET SHIFT(S).
341D	2002	JR	NZ,3421H	SKIP IF DUAL SHIFTS
341F	E601	AND	01H	MASK FOR SHIFTS
3421	E603	AND	03H	MASK FOR SHIFTS
3423	2802	JR	Z,3427H	SKIP IF NO SHIFTS
3425	CBF2	SET	06H,D	OFFSET D FOR SHIFTS
3427	3A1940	LD	A,(4019H)	CAPS LOCK?
342A	B7	OR	A	
342B	2802	JR	Z,342FH	SKIP OF NO CAPS LOCK
342D	CBFA	SET	07H,D	OFFSET D FOR CAPS LOCK.
342F	214530	LD	HL,3045H	HL => KEYBOARD TABLES.
3432	5A	LD	E,D	DE = OFFSET
3433	1600	LD	D,00H	
3435	19	ADD	HL,DE	HL = CHARACTER POSITION
3436	7E	LD	A,(HL)	GET CHARACTER
3437	FE1A	CP	1AH	SHIFT DOWN ARROW?
3439	CAA130	JP	Z,30A1H	NULL EXIT IF SO.
343C	47	LD	B,A	SAVE CHARACTER IN B.
343D	CD6030	CALL	3060H	
3440	78	LD	A,B	RESTORE CHARACTER
3441	2804	JR	Z,3447H	
3443	B7	OR	A	NULL CHARACTER?
3444	CABD30	JP	Z,30BDH	TOGGLE CAPS LOCK IF SO
3447	212442	LD	HL,4224H	HL => 'CONTROL' FLAG
344A	FE2A	CP	2AH	KEY = '*'
344C	2004	JR	NZ,3452H	SKIP IF NOT.
344E	3E1F	LD	A,1FH	CONTROL FLAG SET

3450	BE	CP	(HL)	
3451	78	LD	A,B	RESTORE CHARACTER
3452	C3FD30	JP	30FDH	CONTINUE
3455	ED56	IM	1	SET INTERRUPT MODE 1
3457	317D40	LD	SP,407DH	SET STACK
345A	D3E4	OUT	(0E4H),A	CLEAR NON-MASKABLE INTERRUPT STATUS
345C	F620	OR	20H	SET CONTROL BITS.
345E	D3EC	OUT	(0ECH),A	
3460	3E81	LD	A,81H	SELECT DRIVE 0, DOUBLE DENSITY
3462	D3F4	OUT	(0F4H),A	
3464	3ED0	LD	A,0D0H	RESET DISK CONTROLLER
3466	D3F0	OUT	(0F0H),A	
3468	CD1835	CALL	3518H	WAIT A BIT.
346B	3E04	LD	A,04H	SET MASKABLE INTERRUPT
346D	D3E0	OUT	(0E0H),A	VECTOR TO 4046H
346F	3E0B	LD	A,0BH	RESTORE TO TRACK 0
3471	D3F0	OUT	(0F0H),A	
3473	21AA36	LD	HL,36AAH	INITIALIZE SOME POINTERS
3476	110040	LD	DE,4000H	
3479	014C00	LD	BC,004CH	
347C	EDB0	LDIR		
347E	21F936	LD	HL,36F9H	
3481	11E541	LD	DE,41E5H	
3484	014000	LD	BC,0040H	
3487	EDB0	LDIR		
3489	CDC901	CALL	01C9H	CLEAR SCREEN
348C	CD8D02	CALL	028DH	CHECK <BREAK>
348F	C2AF37	JP	NZ,37AFH	NON-DISK IF PRESSED
3492	DBF0	IN	A,(0F0H)	DISK CONTROLLER ALIVE?
3494	3C	INC	A	
3495	CAAF37	JP	Z,37AFH	NON-DISK IF NOT
3498	010000	LD	BC,0000H	TRY 64K TIMES.
349B	0B	DEC	BC	ANOTHER TIME
349C	3E81	LD	A,81H	SELECT DRIVE 0,
349E	D3F4	OUT	(0F4H),A	DOUBLE DENSITY
34A0	78	LD	A,B	TEST IF READY TO GIVE UP
34A1	B1	OR	C	
34A2	CAAF37	JP	Z,37AFH	GIVE UP ? NON-DISK IF SO
34A5	DBF0	IN	A,(0F0H)	HOW ARE YOU DOING?
34A7	CB57	BIT	02H,A	FOUND TRACK 0 YET?
34A9	28F0	JR	Z,349BH	LOOP IF NOT
34AB	1E05	LD	E,05H	RE-CHECK 5 TIMES
34AD	010000	LD	BC,0000H	TRY UP TO 64K TIMES
34B0	DBF0	IN	A,(0F0H)	CHECK DISK CONTROLLER
34B2	CB4F	BIT	01H,A	INDEX MARK?
34B4	2011	JR	NZ,34C7H	SKIP OUT IF FOUND
34B6	0B	DEC	BC	ANOTHER TRY
34B7	3E81	LD	A,81H	SELECT DRIVE 0,
34B9	D3F4	OUT	(0F4H),A	DOUBLE DENSITY
34BB	78	LD	A,B	TRIED ENOUGH?
34BC	B1	OR	C	
34BD	20F1	JR	NZ,34B0H	LOOP IF NOT
34BF	217702	LD	HL,0277H	PRINT "DISKETTE?"
34C2	CD1B02	CALL	021BH	
34C5	18E4	JR	34ABH	TRY ALL OVER
34C7	1D	DEC	E	FIND INDEX MARK
34C8	20E3	JR	NZ,34ADH	5 TIMES, THEN EXIT

```

34CA 3E81      LD      A,81H          SELECT DRIVE 0,
34CC D3F4      OUT     (0F4H),A      DOUBLE DENSITY
34CE 210235    LD      HL,3502H      SET
34D1 224A40    LD      (404AH),HL    NON-MASKABLE
34D4 3EC3      LD      A,0C3H        INTERRUPT
34D6 324940    LD      (4049H),A     VECTOR (JP 3502H)
34D9 3E80      LD      A,80H        SET NMI
34DB D3E4      OUT     (0E4H),A     STATUS.
34DD 01F300    LD      BC,00F3H     READ FROM PORT 0F3H
34E0 210043    LD      HL,4300H     LOAD FROM 4300 UP.
34E3 3E01      LD      A,01H        READ SECTOR 1
34E5 D3F2      OUT     (0F2H),A
34E7 3E80      LD      A,80H        READ SINGLE RECORD, NON IBM
34E9 D3F0      OUT     (0F0H),A     FORMAT, NO DELAY
34EB CD1835    CALL   3518H        WAIT A BIT
34EE DBF0      IN      A,(0F0H)     DATA READY
34F0 E602      AND     02H
34F2 CAEE34    JP      Z,34EEH      LOOP IF NOT
34F5 EDA2      INI
34F7 3E81      LD      A,81H        SELECT DISK 0,
34F9 F640      OR      40H          DOUBLE DENSITY.
34FB D3F4      OUT     (0F4H),A
34FD EDA2      INI
34FF C3F734    JP      34F7H        LOOP
    
```

THE LOOP 34F7 IS EXITED ONLY BY AN INTERRUPT. THE DISK CONTROLLER GENERATES AN INTERRUPT WHEN IT IS DONE LOADING THE SECTOR. EXIT IS TO 3502H, BECAUSE THE DISK CONTROLLER GENERATES A NON-MASKABLE INTERRUPT.

```

3502 AF        XOR     A              CLEAR INTERRUPT STATUS
3503 D3E4      OUT     (0E4H),A
3505 21ED45    LD      HL,45EDH     SET NEW NON-MASKABLE INTERRUPT VECTOR.
3508 224940    LD      (4049H),HL
350B CD1835    CALL   3518H        WAIT A BIT
350E DBF0      IN      A,(0F0H)     CHECK STATUS
3510 E1        POP     HL           FIX STACK
3511 E61C      AND     1CH          ERROR?
3513 CA0043    JP      Z,4300H     OFF AND RUNNING IF NOT
3516 18B2      JR      34CAH       TRY AGAIN IF NOT.
3518 C5        PUSH    BC          SHORT WAIT FOR FDC
3519 C1        POP     BC
351A 00        NOP
351B C9        RET
351C C24940    JP      NZ,4049H    JUMP IF DISK INTERRUPT
351F DBE4      IN      A,(0E4H)    GET STATUS
3521 CB6F      BIT     05H,A       CHECK BIT 5
3523 28FA      JR      Z,351FH     LOOP UNTIL SET
3525 C30000    JP      0000H       RESET COMPUTER
3528 FF        UNUSED GARBAGE
3529 119135    LD      DE,3591H    SET RETURN ADDRESS
352C D5        PUSH    DE
352D DBEC      IN      A,(0ECH)
352F 3A2240    LD      A,(4022H)   CURSOR ON?
3532 B7        OR      A
3533 2822      JR      Z,3557H     SKIP IF NOT.
    
```

```

3535 3A1C40    LD      A, (401CH)    CURSOR BLINK?
3538 B7       OR      A
3539 201C     JR      NZ, 3557H    SKIP IF NOT
353B 211A40   LD      HL, 401AH    GET BLINKER COUNT
353E 35       DEC     (HL)         UPDATE IT.
353F 2016     JR      NZ, 3557H    SKIP IF NOT = ZERO
3541 3607     LD      (HL), 07H    SET COUNT TO 7
3543 23       INC     HL           POINT TO ON/OFF FLAG.
3544 7E       LD      A, (HL)     GET FLAG
3545 E601     AND     01H         MASK IT.
3547 EE01     XOR     01H         TOGGLE IT.
3549 77       LD      (HL), A     STORE IT
354A 2A2040   LD      HL, (4020H)  GET CURSOR POSITION
354D 2805     JR      Z, 3554H    SKIP IF OFF
354F 3A2340   LD      A, (4023H)  GET CURSOR CHARACTER.
3552 1802     JR      3556H
3554 3E20     LD      A, 20H      CURSOR = SPACE
3556 77       LD      (HL), A     DISPLAY CHARACTER.
3557 211642   LD      HL, 4216H   INCREMENT HEARTBEAT
355A 35       DEC     (HL)         COUNTER
355B C0       RET     NZ          SKIP IF NON-ZERO
355C 361E     LD      (HL), 1EH   SET COUNTER TO 1EH
355E 23       INC     HL           POINT TO SECONDS
355F 116602   LD      DE, 0266H   DE =>#SEC/MIN, #MIN/HR, #HR/DAY.
3562 0603     LD      B, 03H      SECONDS/MINUTES/HOURS.
3564 34       INC     (HL)         INCREMENT CURRENT TIME.
3565 1A       LD      A, (DE)     GET MAX # UNITS
3566 96       SUB     (HL)         COMPARE WITH CURRENT
3567 C0       RET     NZ          SKIP IF DONE
3568 77       LD      (HL), A     SAVE DIFFERENCE
3569 23       INC     HL           NEXT UNIT
356A 13       INC     DE          NEXT MAX
356B 10F7     DJNZ   3564H        LOOP UNTIL DONE
356D 23       INC     HL           INCREMENT DAY
356E 34       INC     (HL)
356F 23       INC     HL           GET MONTH
3570 7E       LD      A, (HL)
3571 2B       DEC     HL
3572 3D       DEC     A           DE => # DAYS IN MONTH
3573 83       ADD     A, E
3574 5F       LD      E, A
3575 1A       LD      A, (DE)     GET # DAYS IN MONTH.
3576 BE     CP      (HL)         DAY IN MONTH?
3577 D0       RET     NC          SKIP IF SO.
3578 7E       LD      A, (HL)     GET DAY.
3579 FE1E     CP      1EH         DAY ≥ 30?
357B 3006     JR      NC, 3583H   SKIP TO UPDATE MONTH
357D 2B       DEC     HL           GET YEAR
357E 7E       LD      A, (HL)
357F 23       INC     HL
3580 E603     AND     03H         LEAP YEAR?
3582 C8       RET     Z           SKIP IF SO.
3583 3601     LD      (HL), 01H   DAY = 1
3585 23       INC     HL           NEXT MONTH
3586 34       INC     (HL)
3587 7E       LD      A, (HL)     GET MONTH
3588 D60D     SUB     0DH         MONTH ≥ 13?

```

358A	D8	RET	C	SKIP IF NOT
358B	3601	LD	(HL), 01H	MONTH = 1
358D	2B	DEC	HL	NEXT YEAR
358E	2B	DEC	HL	
358F	34	INC	(HL)	
3590	C9	RET		SKIP OUT
3591	3A1042	LD	A, (4210H)	CLOCK ON?
3594	CB47	BIT	00H, A	
3596	C8	RET	Z	RETURN IF NOT
3597	3A1642	LD	A, (4216H)	CLOCK JUST UPDATED?
359A	FE1E	CP	1EH	
359C	C0	RET	NZ	RETURN IF NOT
359D	21353C	LD	HL, 3C35H	DISPLAY TO SCREEN

TIME\$

35A0	111942	LD	DE, 4219H	POINT TO TIME
35A3	0E3A	LD	C, 3AH	DELIMITER = ":"
35A5	0603	LD	B, 03H	CONVERT 3 NUMBERS
35A7	1A	LD	A, (DE)	GET NUMBER
35A8	1B	DEC	DE	POINT TO NEXT #
35A9	362F	LD	(HL), 2FH	FIRST DIGIT = "0" - 1
35AB	34	INC	(HL)	INCREMENT FIRST DIGIT
35AC	D60A	SUB	0AH	NUMBER = NUMBER - 10
35AE	30FB	JR	NC, 35ABH	LOOP IF NUMBER ≥ 0
35B0	C63A	ADD	A, 3AH	A = REMAINDER + "0"
35B2	23	INC	HL	STORE SECOND DIGIT
35B3	77	LD	(HL), A	
35B4	23	INC	HL	
35B5	05	DEC	B	DECREMENT # LEFT TO DO
35B6	C8	RET	Z	RETURN IF DONE.
35B7	71	LD	(HL), C	STORE DELIMITER.
35B8	23	INC	HL	
35B9	18EC	JR	35A7H	NEXT NUMBER

DATE\$

35BB	111C42	LD	DE, 421CH	POINT TO DATE
35BE	0E2F	LD	C, 2FH	DELIMITER = "/"
35C0	18E3	JR	35A5H	CONVERT DATE.

INTERRUPT HANDLER

35C2	F5	PUSH	AF	SAVE STATUS
35C3	DBE0	IN	A, (0E0H)	READ INTERRUPT LATCH.
35C5	1F	RRA		BIT 0 LOW?
35C6	D26533	JP	NC, 3365H	JUMP IF SO.
35C9	1F	RRA		BIT 1 LOW?
35CA	D26933	JP	NC, 3369H	JUMP IF SO.
35CD	C5	PUSH	BC	SAVE REGISTERS
35CE	D5	PUSH	DE	
35CF	E5	PUSH	HL	
35D0	DDE5	PUSH	IX	
35D2	FDE5	PUSH	IY	
35D4	21F135	LD	HL, 35F1H	SET RETURN ADDRESS
35D7	E5	PUSH	HL	
35D8	1F	RRA		BIT 2 LOW? (CLOCK INTERRUPT)
35D9	D24640	JP	NC, 4046H	JUMP IF SO.
35DC	1F	RRA		BIT 3 LOW?
35DD	D23D40	JP	NC, 403DH	JUMP IF SO.

35E0	1F	RRA		BIT 4 LOW?
35E1	D20642	JP	NC,4206H	JUMP IF SO.
35E4	1F	RRA		BIT 5 LOT?
35E5	D20942	JP	NC,4209H	JUMP IF SO.
35E8	1F	RRA		BIT 6 LOW?
35E9	D24040	JP	NC,4040H	JUMP IF SO.
35EC	1F	RRA		
35ED	D24340	JP	NC,4043H	BIT 7 LOW? JUMP IF SO.
35F0	E1	POP	HL	DISCARD RETURN ADDRESS
35F1	FDE1	POP	IY	RESTORE REGISTERS
35F3	DDE1	POP	IX	
35F5	E1	POP	HL	
35F6	D1	POP	DE	
35F7	C1	POP	BC	
35F8	F1	POP	AF	
35F9	FB	EI		RE-ENABLE INTERRUPTS.
35FA	C9	RET		DONE.

RS-232 INITIALIZATION ROUTINE

35FB	F3	DI		DISABLE INTERRUPTS
35FC	DBEA	IN	A, (0EAH)	CHECK RS-232 STATUS
35FE	FEFF	CP	0FFH	NON-EXISTENT?
3600	2838	JR	Z,363AH	SKIP IF SO.
3602	AF	XOR	A	RESET RS-232
3603	D3E8	OUT	(0E8H),A	
3605	DD7E03	LD	A, (IX+03H)	GET BAUD RATE CODE.
3608	D3E9	OUT	(0E9H),A	SET BAUD RATES
360A	DD7E04	LD	A, (IX+04H)	GET CONFIGURATION
360D	B7	OR	A	NULL?
360E	282A	JR	Z,363AH	SKIP IF SO.
3610	D3EA	OUT	(0EAH),A	SET CONFIGURATION
3612	FD21E541	LD	IY,41E5H	IY => RS-232 INPUT DCB.
3616	CD4436	CALL	3644H	CLEAR OPTIONS
3619	DD7E05	LD	A, (IX+05H)	GET WAIT SWITCH
361C	B7	OR	A	
361D	2804	JR	Z,3623H	SKIP IF NOT SET
361F	FDCB04CE	SET	01H, (IY+04H)	SET WAIT FLAG.
3623	FDCB04D6	SET	02H, (IY+04H)	SET ACTIVE FLAG
3627	FD21ED41	LD	IY,41EDH	IY => RS-232 OUTPUT DCB
362B	B7	OR	A	CHECK WITH FLAG
362C	2804	JR	Z,3632H	SKIP IF NO WAIT
362E	FDCB04CE	SET	01H, (IY+04H)	SET WAIT FLAG.
3632	FDCB04D6	SET	02H, (IY+04H)	SET ACTIVE FLAG.
3636	DBE8	IN	A, (0E8H)	GET STATUS
3638	FB	EI		LISTEN FOR INTERRUPTS
3639	C9	RET		DONE
363A	AF	XOR	A	CLEAR A
363B	0604	LD	B,04H	4 PORTS
363D	0EE8	LD	C,0E8H	SMARTING AT 0E8
363F	ED79	OUT	(C),A	CLEAR PORT
3641	0C	INC	C	NEXT PORT
3642	10FB	DJNZ	363FH	LOOP UNTIL DONE
3644	21E841	LD	HL,41E8H	CLEAR INPUT OPTIONS
3647	0603	LD	B,03H	3 BYTES
3649	3600	LD	(HL),00H	CLEAR BYTE
364B	23	INC	HL	NEXT BYTE
364C	10FB	DJNZ	3649H	LOOP UNTIL DONE

```

364E 21F041 LD HL,41F0H CLEAR OUTPUT OPTIONS
3651 0603 LD B,03H 3 BYTES
3653 3600 LD (HL),00H CLEAR BYTES
3655 23 INC HL NEXT BYTE
3656 10FB DJNZ 3653H LOOP UNTIL DONE
3658 18DC JR 3636H RETURN
    
```

RS-232, INPUT ROUTINE

```

365A DD21E541 LD IX,41E5H IX => DCB FOR RS-232 INPUT
365E AF XOR A CLEAR INPUT
365F DD7703 LD (IX+03H),A CLEAR BUFFER
3662 DDCB0456 BIT 02H,(IX+04H) RS-232 ACTIVE?
3666 C8 RET Z RETURN IF NOT
3667 DBEA IN A,(0EAH) GET BIT 1,(IX+4) WAIT
3669 CB7F BIT 07H,A
366B 200D JR NZ,367AH
366D DDCB044E BIT 01H,(IX+04H)
3671 C8 RET Z RETURN IF NOT.
3672 CD8D02 CALL 028DH CHECK <BREAK>.
3675 28F0 JR Z,3667H WAIT IF NOT PRESSED.
3677 C30342 JP 4203H EXIT IF PRESSED.
367A DBEB IN A,(0EBH) GET DATA
367C DD7703 LD (IX+03H),A SAVE IN BUFFER
367F C9 RET DONE
    
```

RS-232 OUTPUT ROUTINE

```

3680 DD21ED41 LD IX,41EDH IX => DCB FOR RS-232 OUTPUT
3684 DDCB0456 BIT 02H,(IX+04H) RS-232 ACTIVE?
3688 C8 RET Z RETURN IF NOT
3689 DBEA IN A,(0EAH) GET STATUS
368B CB77 BIT 06H,A READY TO SEND?
368D 200D JR NZ,369CH SKIP IF SO.
368F DDCB044E BIT 01H,(IX+04H) WAIT?
3693 C8 RET Z RETURN IF NOT
3694 CD8D02 CALL 028DH CHECK <BREAK>
3697 28F0 JR Z,3689H WAIT IF NOT PRESSED
3699 C30342 JP 4203H EXIT IF PRESSED
369C DD7E03 LD A,(IX+03H) GET BUFFER CHAR.
369F B7 OR A NULL?
36A0 2001 JR NZ,36A3H SKIP IF CHAR IN BUFFER.
36A2 79 LD A,C GET CHAR FROM DISPATCHER
36A3 D3EB OUT (0EBH),A SEND CHAR.
36A5 DD360300 LD (IX+03H),00H ZERO BUFFER
36A9 C9 RET DONE
    
```

```

36AA C3961CC3781DC3901CC3D925C90000 INITIAL VECTORS AND DCBS FOR
---- C90000C31830012430000107000007 RAM 4000H-404BH. TAPE & DISK
---- 7304003C00B00006C203430100FF52
---- C30050C70000AFC900AAAAAAAAAAAAAA
---- AAC3FA35C3FA35C3FA35C32935C700
36F5 00000000 UNUSED
    
```

```

36F9 011E30000000524902213000000052 INITIAL VECTORS AND DCBS FOR
---- 4F021B30556CFF524E0000FFFF0000 RAM 41E5H-4224H TAPE & DISK.
---- C32E02C3FA35C3FA35413203322803
---- 3C0400001E331C0C000E0202393700
---- 000000FF
    
```

I/O RE-ROUTER

3739	DD7E03	LD	A, (IX+03H)	GET 1ST CHARACTER OF DESTINATION.
373C	FE52	CP	52H	IS IT A "R"?
373E	2003	JR	NZ, 3743H	SKIP IF NOT.
3740	DD7E04	LD	A, (IX+04H)	GET 2ND CHARACTER OF DESTINATION.
3743	CD5E37	CALL	375EH	GET DCB ADDRESS.
3746	C0	RET	NZ	RETURN IF NOT FOUND
3747	E5	PUSH	HL	SAVE IT.
3748	DD7E05	LD	A, (IX+05H)	GET 1ST CHARACTER OF SOURCE.
374B	FE52	CP	52H	IS IT A "R"?
374D	2003	JR	NZ, 3752H	SKIP IF NOT
374F	DD7E06	LD	A, (IX+06H)	GET 2ND CHARACTER OF SOURCE.
3752	CD5E37	CALL	375EH	GET DCB ADDRESS.
3755	EB	EX	DE, HL	MOVE TO SOURCE DCB.
3756	E1	POP	HL	FROM DESTINATION DCB
3757	C0	RET	NZ	RETURN IF NOT FOUND
3758	010300	LD	BC, 0003H	MOVE 3 BYTES
375B	EDB0	LDIR		
375D	C9	RET		DONE
375E	216C37	LD	HL, 376CH	POINT TO TABLE
3761	010F00	LD	BC, 000FH	WHICH IS 15 BYTES LONG
3764	EDB1	CPIR		SEARCH FOR GIVEN CHARACTER.
3766	C0	RET	NZ	RETURN IF NOT FOUND
3767	7E	LD	A, (HL)	LOAD DCB ADDRESS
3768	23	INC	HL	
3769	66	LD	H, (HL)	
376A	6F	LD	L, A	
376B	C9	RET		DONE
376C	4B1540	"K"	4015H	DEVICE LABELS
376F	441D40	"D"	401DH	& DCB ADDRESSES.
3772	502540	"P"	4025	
3775	49E541	"I"	41E5	
3778	4FED41	"O"	41ED	
377B	FE22	CP	22H	
377D	200A	JR	NZ, 3789H	
377F	3A9F40	LD	A, (409FH)	
3782	EE01	XOR	01H	
3784	329F40	LD	(409FH), A	
3787	3E22	LD	A, 22H	
3789	FE3A	CP	3AH	
378B	C2AA06	JP	NZ, 06AAH	
378E	3A9F40	LD	A, (409FH)	
3791	1F	RRA		
3792	DAA806	JP	C, 06A8H	
3795	17	RLA		
3796	C3A306	JP	06A3H	

BASIC TIMES (DATE\$+" "+TIME\$)

3799	D7	RST	10H	GET NEXT CHARACTER
379A	E5	PUSH	HL	SAVE POSITION
379B	3E11	LD	A, 11H	CREATE A 17 BYTE
379D	CD5728	CALL	2857H	STRING
37A0	2AD440	LD	HL, (40D4H)	GET POINTER
37A3	CDBB35	CALL	35BBH	LOAD DATE\$
37A6	3620	LD	(HL), 20H	SPACE APART
37A8	23	INC	HL	MOVE OVER
37A9	CDA035	CALL	35A0H	LOAD TIME\$

37AC C38428 JP 2884H FINISH UP & RETURN
37AF CDB537 CALL 37B5H SET CASSETTE SPEED
37B2 C37500 JP 0075H GO TO NON-DISK INITIALIZATION

SET CASSETTE BAUD RATE (\$SETCAS)

37B5 FB EI ENABLE INTERRUPTS
37B6 CDD737 CALL 37D7H NEW LINE
37B9 21F637 LD HL,37F6H PRINT "CASS?"
37BC CD1B02 CALL 021BH
37BF CD4900 CALL 0049H LISTEN FOR RESPONSE
37C2 FE0D CP 0DH CARRIAGE RETURN
37C4 280E JR Z,37D4H DEFAULT HIGH IF SO.
37C6 F5 PUSH AF SAVE AF.
37C7 CD3300 CALL 0033H DISPLAY RESPONSE
37CA F1 POP AF RESTORE AF.
37CB FE48 CP 48H "H"
37CD 2805 JR Z,37D4H HIGH SPEED IF SO.
37CF FE4C CP 4CH "L"
37D1 20E2 JR NZ,37B5H RETRY IF NOT
37D3 AF XOR A LOW SPEED IF SO.
37D4 321142 LD (4211H),A SAVE SPEED BYTE
37D7 3E0D LD A,0DH NEXT LINE
37D9 C33300 JP 0033H DONE
37DC 213030 LD HL,3030H SET TIME\$
37DF 227741 LD (4177H),HL VECTOR.
37E2 C32E02 JP 022EH CONTINUE
37E5 AAAAAA3F3F UNUSED
37EA 01 COMPUTER VERSION
37EB CD1B02 CALL 021BH PRINT MESSAGE
37EE 210202 LD HL,0202H PRINT "(C) 80 TANDY"
37F1 CD1B02 CALL 021BH
37F4 18E6 JR 37DCH CONTINUE
37F6 0E TURN ON CURSOR
37F6 436173733F20 "CASS? " MESSAGE
37FC 03 KEEP CURSOR FIXED
37FE AAAA UNUSED

***** END OF READ-ONLY MEMORY *****

TAPE RAM

4000 JP 1C96H RST 08H SYNTAX CHECK
 4003 JP 1D78H EST 10H GET NEXT CHARACTER
 4006 JP 1C90H RST 18H CP, HL, DE
 4009 JP 25D9H RST 20H GET CURRENT TYPE
 400C RET RST 28H UNUSED
 400E RET RST 30H UNUSED
 4012 JP 3018H RST 38H INTERRUPT VECTOR

KEYBOARD DCB

4015 TYPE = 1 = READ ONLY
 4016 DRIVER ADDRESS (3024H)
 4018 RIGHT SHIFT TOGGLE
 4019 CAPS LOCK TOGGLE
 401A CURSOR BLINK COUNT
 401B CURSOR BLINK STATUS (Z = OFF NZ = ON)
 401C CURSOR BLINK SWITCH (Z = BLINK NZ = NO BLINK)

VIDEO DCB

401D TYPE = 7 = READ/WRITE
 401E DRIVER ADDRESS (0473H)
 4020 CURSOR POSITION
 4022 CURSOR ON/OFF FLAG (Z = OFF NZ = CHARACTER UNDERCORSOR)
 4023 CURSOR CHARACTER (DEFAULT 0B0H)
 4024 TABS/SPECIAL CHARACTERS SWITCH (Z = TABS)

PRINTER DCB

4025 TYPE = 6 = WRITE
 4026 DRIVER ADDRESS (03C2H)
 4028 # LINES/PAGE
 4029 # LINES PRINTED + 1
 402A # CHARACTERS PRINTED + 1
 402B PRINTED WIDTH - 2 (255 = INFINITE)
 402C
 402D JP 5000H UNUSED DOS VECTOR
 4030 RST 0
 4033 XOR A
 RET
 4036 KEYBOARD BIT IMAGE (7 BYTES)
 403D JP 35FAH INTERRUPT VECTOR 3 (UNUSED)
 4040 JP 35FAH INTERRUPT VECTOR 6 (UNUSED)
 4043 JP 35FAH INTERRUPT VECTOR 7 (UNUSED)
 4046 JP 35A9H INTERRUPT VECTOR 2 (CLOCK)
 4049 RST 0 NMI VECTOR (RESET IF SO)
 404C UNUSED (36 BYTES)
 STACK SPACE DURING BOOTSTRAP

DIVISION SUPPORT ROUTINE

4080 SUB (4081H) L = L - LSB
 LD L, A
 LD A, H H = H - NMSB - CARRY
 SBC (4085H)
 LD H, A
 LD A, B B = B - NSB - CARRY
 SBC (4089H)
 LD B, A
 LD A, (408CH) A = OVERFLOW COUNTER

TAPE RAM

RET DONE
 NET RESULT: BHL = BHL - (ACCUM)

408E ADDRESS OF USR SUBROUTINE
 4090 RANDOM NUMBER SEED 2
 4093 IN A, (4094H) INPUT SUPPORT
 RET ROUTINE
 4096 OUT (4097H), A OUT SUPPORT
 RET ROUTINE
 4099 INKEY\$ STORAGE
 409A ERR STORAGE
 409B PRINTER COLUMN POSITION
 409C DEVICE ROUTING FLAG (-1=TAPE, 0=VIDEO, +1=PRINTER)
 409D DISPLAY LINE LENGTH
 409E VIDEO DISPLAY WIDTH
 40A0 TOP OF FREE MEMORY
 40A2 CURRENT LINE NUMBER
 40A4 DATA POINTER
 40A6 POS STORAGE
 40A7 ADDRESS OF KEYBOARD BUFFER
 40A9 CASSETTE INPUT FLAG (Z=TAPE INPUT)
 40AA RANDOM NUMBER SEED 1

ACCUM

	<u>INTEGER</u>	<u>STRING</u>	<u>S.P.</u>	<u>D.P.</u>
411C	UNUSED	UNUSED	UNUSED	UNUSED
411D	UNUSED	UNUSED	UNUSED	LSB
	UNUSED	UNUSED	UNUSED	NMSB
	UNUSED	UNUSED	UNUSED	NMSB
	UNUSED	UNUSED	BIT BUCKET	NMSB
4121	LSB	LENGTH	LSB	NMSB
	UNUSED	LSB	NMSB	NMSB
	UNUSED	MSB	MSB	MSB
4124	UNUSED	UNUSED	EXP	EXP
4125	SIGN	UNUSED	SIGN	SIGN

NOTE: IN SOME STRING OPERATIONS,
 4121 - 4122 POINTS TO AN ENTRY
 IN THE STRING POINTER WORKSPACE

ACCUM2

	<u>INTEGER</u>	<u>S.P.</u>	<u>D.P.</u>
4126	UNUSED	UNUSED	BIT BUCKET
4127	LSB	LSB	LSB
	MSB	NMSB	NMSB
		MSB	NMSB
		EXP	NMSB
			NMSB
			NMSB
			MSB
412F			EXP
4130	ASCII CONVERSION WORKSPACE (25 BYTES)		
414A	WORKSPACE FOR D.P. DIVISION (8 BYTES)		

TAPE RAMDISK BASIC VECTOR

4152	JP 012DH	CVI	L3 ERROR
4155	JP 012DH	FN	L3 ERROR
4158	JP 012DH	CVS	L3 ERROR
415B	JP 012DH	DEF	L3 ERROR
415E	JP 012DH	CVD	L3 ERROR
4161	JP 012DH	EOF	L3 ERROR
4164	JP 012DH	LOC	L3 ERROR
4167	JP 012DH	LOF	L3 ERROR
416A	JP 012DH	MKI\$	L3 ERROR
416D	JP 012DH	MKS\$	L3 ERROR
4170	JP 012DH	MKD\$	L3 ERROR
4173	JP 012DH	CMD	L3 ERROR
4176	JP 3030H	TIMES\$	NO ERROR
4179	JP 012DH	OPEN	L3 ERROR
417C	JP 012DH	FIELD	L3 ERROR
417F	JP 012DH	GET	L3 ERROR
4182	JP 012DH	PUT	L3 ERROR
4185	JP 012DH	CLOSE	L3 ERROR

DISK BASIC VECTORS (CONT)

4188	JP 012DH	LOAD	L3 ERROR
418B	JP 012DH	MERGE	L3 ERROR
418E	JP 012DH	NAME	L3 ERROR
4191	JP 012DH	KILL	L3 ERROR
4194	JP 012DH	&	L3 ERROR
4197	JP 012DH	LSET	L3 ERROR
419A	JP 012DH	RSET	L3 ERROR
419D	JP 012DH	INSTR	L3 ERROR
41A0	JP 012DH	SAVE	L3 ERROR
41A3	JP 012DH	LINE	L3 ERROR

DOS LINKS

41A6	RET	LINK FOR ERROR MESSAGE
41A9	RET	LINK FOR USR N
41AC	RET	LINK FOR READY
41AF	RET	LINK FOR INKEY\$
41B2	RET	LINK AFTER LINE ENCODE
41B5	RET	LINK AFTER PROGRAM UPDATE
41B8	RET	LINK AFTER PROGRAM CLEAR
41BB	RET	LINK DURING NEW + END
41BE	RET	LINK DURING I/O RESET
41C1	RET	LINK DURING SYSTEM OUTPUT
41C4	RET	LINK DURING KEYBOARD WAIT
41C7	RET	LINK FOR RUN EXP
41CA	RET	LINK FOR SEQUENTIAL FILE OUTPUT
41CD	RET	LINK BETWEEN PRINT ITEMS
41D0	RET	LINK DURING NEW LINE ON VIDEO
41D3	RET	LINK DURING PRINT OUTPUTTING
41D6	RET	LINK AT START OF INPUT
41D9	RET	LINK FOR MID\$ ON LEFT OF "="
41DC	RET	LINK AT START OF READ SCAN
41E2	RET	LINK FOR SYSTEM AUTO-START

TAPE RAMRS-232 INPUT DCB

41E5 TYPE = 1 = READ ONLY
41E6 DRIVER ADDRESS (301EH)
41E8 INPUT BUFFER (1 CHARACTER)
41E9 BIT 2 = DRIVER ON/OFF BIT 1 = WAIT/NO WAIT

RS-232 OUTPUT DCB

41ED TYPE = 2 = WRITE ONLY
41EE DRIVER ADDRESS (3021H)
41F0 OUTPUT BUFFER (1 CHARACTER)
41F1 BIT 2 = DRIVER ON/OFF BIT 1 = WAIT/NO WAIT

RS-232 INITIALIZATION DCB

41F5 TYPE = 2 = WRITE ONLY
41F6 DRIVER ADDRESS (301BH)
41F8 BAUD RATE CODE
41F9 CONFIGURATION CODE
41FA WAIT/NO WAIT SNITCH

KEYBOARD SCAN DATA

41FD SAVED POSITION IN SCAN
41FE SAVED IMAGE AT POSITION
41FF REPEAT DELAY COUNT
4201 REPEAT DELAY COUNTER

CASSETTE INFO

4203 JP 022EH BREAK VECTOR FOR TAPE/RS-232
4206 JP 35FAH INTERRUPT VECTOR # 4
4209 JP 35FAH INTERRUPT VECTOR # 5
420C TAPE WRITE VECTOR
420E TAPE READ VECTOR
4210 BIT MASK FOR PORT 0ECH
4211 CASSETTE BAUD RATE SELECT (NZ = 1500 BAUD)
4212 CASSETTE BLINKER COUNTER
4213 DEFAULT INTERRUPT VECTOR SETTING
4214 # VIDEO LINES TO PROTECT
4215 UNUSED IN TAPE SYSTEM

CLOCK DATA

4216 HEARTBEAT COUNTER
4217 SECOND
4218 MINUTE
4219 HOUR
421A YEAR
421B DAY
4210 MONTH

I/O ROUTER DCB

421D TYPE = 2 = WRITE ONLY
421E DRIVER ADDRESS (3739H)
4220 DESTINATION DEVICE NAME
4222 SOURCE DEVICE NAME
4224 UNUSED BY ROUTER, CONTROL KEY FLAG.

THE FOLLOWING INFORMATION IS EXTRA INPUT FROM MANY PEOPLE AND HAS NOT BEEN COMPLETED YET, AND WILL NOT BE UPDATED TO ANY USER, BUT WE THOUGHT THAT IT MAY HELP SOMEONE TO HAVE THIS INFORMATION AS IT STANDS.

E0 MASKABLE INTERRUPT LATCH. DIRECTS JUMPS

BITS

0 => 3365 CASSETTE ROUTINES
 1 => 3369 CASSETTE ROUTINES
 2 => 4046 CURSOR BLINK & CHECK
 3 => 403D
 4 => 4206
 5 => 4209
 6 => 4040
 7 => 4043

E4 NON-MASKABLE INTERRUPT LATCH

BITS

0
 1
 2
 3
 4
 5
 6
 7

E8 RS-232 STATUS REGISTER & MASTER RESET

OUT: ANY BYTE RESETS INTERFACE

IN:

BITS

0 UNUSED
 1
 2 UNUSED
 3 UNUSED
 4 RI RING INDICATOR
 5 CD CARRIER DETECT
 6 DSR DATA SET READY
 7 CTS CLEAR TO SEND

E9 RS-232 BAUD RATE SELECT & SENSE SWITCHES

IN: SENSE SWITCHES

BITS

7 PARITY 0 = ODD 1 = EVEN
 6 WORD LENGTH SELECT IN BITS
 5 00 = 5 01 = 6 10 = 7 11 = 8
 4 STOP BITS 0 = BIT 1 = 2 BITS
 3 PARITY 0 = ENABLED 1 = DISABLED
 2 BAUD RATE SELECT
 1 BAUD RATE SELECT
 0 BAUD RATE SELECT
 OUT: BAUD RATE SELECT
 BITS 0 - 3 SELECT RECEIVE RATE
 4 - 7 SELECT TRANSMIT RATE

EA RS-232C UART CONTROL REGISTER & STATUS REGISTER

IN: STATUS REGISTER

BITS

7	DATA READY	1 = TRUE
6	DATA SENT	1 = TRUE
5	OVERRUN	1 = TRUE
4	FRAMING ERROR	1 = TRUE
3	PARITY ERROR	1 = TRUE
2	UNUSED	
1	UNUSED	
0	UNUSED	

OUT: CONTROL REGISTER

BITS

7	PARITY	0 = ODD	1 = EVEN
6	WORD LENGTH		
5	SELECT		
4	STOP BITS		
3	PARITY ENABLE		
2	BREAK		
1	REQUEST TO END		
0	DATA TERMINAL READY		

EB RS-232C DATA REGISTER

IN: RECEIVED DATA

OUT: TRANSMIT DATA

EC MISCELLANEOUS CONTROLS (IMAGE 4210H)

BITS OUT ONLY

0				
1	CASSETTE MOTOR	0 = OFF	1 = ON	
2	DUAL SIZE VIDEO	0 = OFF	1 = ON	
3	SPECIAL CHARACTER SELECT		0 = KANA	1 = MISC.
4				
5				
6				
7				

F0 FDC STATUS REGISTER SEE FLOPPY DISK

F1 FDC TRACK REGISTER CONTROLLER MANUAL

F2 FDC SECTOR REGISTER

F3 FDC DATA REGISTER

F4 DISK DRIVE/DENSITY SELECT OUT ONLY

BITS

0	DRIVE 0 SELECT	
1	DRIVE 1 SELECT	
2	DRIVE 2 SELECT	
3	DRIVE 3 SELECT	
4		
5		
6		
7	DENSITY SELECT	0 = SINGLE 1 = DOUBLE

F8 PRINTER PORT
IN: PRINTER STATUS
BITS
0 UNUSED
1 UNUSED
2 UNUSED
3 UNUSED
4 NOT PRINTER FAULT
5 DEVICE SELECT
6 NOT OUT OF PAPER
7 NOT BUSY
OUT: DATA TO PRINT

FF CASSETTE PORT
IN: READ STATUS
BITS
0
1
2
3
4
5
6
7 DATA BIT 0 = LOW 1 = HIGH (LATCHED)
OUT: RECORD LEVEL
BITS
0 LEVEL SELECT 00 = 0.85V 10 = 0.0V
1 01 = 0.46V
2
3
4
5
6
7