PRICE $2.00

THE 80 NOTEBOOK

OCTOBER 1980 ISSUE #7

NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation. THE 80 NOTEBOOK is not affiliated with Radio Shack or Tandy Corporation in any way.

MACHINE LANGUAGE AND ASSEMBLY LANGUAGE:

PROGRAMMING MADE AS SIMPLE AS BASIC

(continued from the September issue)

In last month's issue, we began our exploration of the world of machine and assembly language with a review of memory functions, the hardware architecture of the TRS-80, addressing and the instruction set our system will use. Now we will continue our study of this topic with a look at how to use our instructions to accomplish some tasks. The specific tasks we will examine involve making use of the I/O service routines in the BASIC ROM available to an assembly language application program.

First, at location address 16416 in memory, there is a DA field containing the address within the screen memory area of the cursor character on the display screen. The address value in the field may range from 15360 to 16383. Once the address value has been stored in the DA field, you will want to turn the cursor character on. This can be done by the following routine:

```
PUSH DE
PUSH IY
LD   A,14
CALL 9,51
POP  IY
POP  DE
```

To turn the cursor off at its current location, use the following routine:

```
PUSH DE
PUSH IY
LD   A,15
CALL 9,51
POP  IY
POP  DE
```

To display a character of text at the current cursor location on the screen, place the character of text in the A register and perform the following routine:

```
PUSH DE
PUSH IY
CALL 9,51
POP  IY
POP  DE
```

When an assembly language program wishes to obtain input data from the TRS-80 keyboard, it must scan the keyboard for a struck key, receive the representative character and then repeat that process for each character of text expected or until a termination character, such as a return key character, is received. As each character is received in the A register, it must be stored in an area of memory reserved in your program for keyboard input. If you are receiving a field of numeric data, you should remember that the numeric text as stored in your memory is in string form. If you wish to perform any arithmetic operations on that data, you will need to write an assembly language routine to convert the numeric string into an 8 or 16 bit machine language number format first. Even then you will still have to take signs and decimal points into consideration during any arithmetic manipulation of the converted numeric data. The following is an example of how to scan the keyboard until a character is received in the A register:

```
TAG  AGN
PUSH DE
PUSH IY
CALL 9,43
OR   A
JR   5,AGN
POP  IY
POP  DE
```

An assembly language program might also use data from or write data to cassette tape. Reading a record from a cassette is a four step operation. First you might define the cassette drive:

```
LD   A,Ø
CALL 9,53Ø
```

Then you must find the next sync byte on the tape:

```
CALL 9,662
```

Then you must read the record into memory one byte at a time. To read a 1ØØ character record into an area in memory labeled RECA:

```
LD   B,1ØØ
LDX  HL,ARECA
TAG  LBLA
PUSH BC
PUSH HL
CALL 9,565
POP  HL
POP  BC
LD   M,A
INCX HL
LD   A,B
```

```
DEC  A
LD   B,A
JR   5,LBLA
DS   1ØØ,RECA
DA   RECA,ARECA
```

Finally, you must turn the cassette drive off:

```
CALL 9,5Ø4
```

Likewise, to write a record to the cassette tape involves four similar steps. First, you must define the cassette drive. Then, you must write a leader and sync byte:

```
CALL 9,647
```

Then, you must write your record to tape one byte at a time. To write a 1ØØ byte record stored in a memory area labeled RECA to tape:

```
LD   B,1ØØ
LDX  HL,ARECA
TAG  LBLA
PUSH BC
PUSH HL
LD   A,M
CALL 9,612
POP  HL
POP  BC
LD   A,B
INCX HL
DEC  A
LD   B,A
JR   5,LBLA
DS   1ØØ,RECA
DA   RECA,ARECA
```

Finally, you must turn off the cassette drive.

If you have a stand-alone (not called by a user BASIC program) assembly language program, you should end your program by jumping to the BASIC command level ready state:

```
JP   9,6681
```

For those assembly language routines which are called by user written BASIC programs, the following conventions should be observed: To call an assembly language routine from BASIC, you must first identify the location of the routine and store it in a special area of memory at locations 16526 and 16527. For example, if your routine starts at location 32ØØØ, you convert this address into its 2 byte 16 bit format. In this form, byte 1 of the address would contain the value 2Ø8 and byte 2 the value Ø. These 2 bytes must be stored in reverse order because they represent an address. This can be done by the following:

```
1Ø POKE 16526,Ø:POKE 16527,2Ø8
```

You then call the routine using the USR command. If a signed 16 bit value is to be passed to your routine as an argument, it can be done through a numeric literal or field name argument expressed with the USR command. If no argument is to be passed, a dummy argument of zero is expressed with the USR command. The following is an example of a USR call with an argument contained in a field named N:

```
15 N=1295
2Ø A=USR(N)
```

If your routine is expecting an argument, the following instruction will retrieve the value and place it in the HL register pair:

```
CALL 9,2687
```

Remember, this value is a signed 16 bit value which means the possible values range from -32768 to +32767 because the first bit (leftmost, bit 1) represents a sign (a 1 bit represents a negative) rather than a numeric value position. At the end of your routine execution, if no argument value is being passed back to your BASIC program, you should end your routine with a RET instruction. If you are returning a signed 16 bit value (which would be placed in the A field of line 2Ø in our example), load the value into the HL register pair and execute a JP 9,2714 instruction at the end of your routine.

All assembly language programs and subroutines should begin by storing the current SP register value in a save area in your program and a new SP register value loaded based on a large enough stack area in your program to handle your program's needs. The saved SP register value must be re-loaded before you end your program or routine.

Now that we have a fundamental understanding of assembly language, we can discuss how to input and prepare a program or routine for use on your TRS-80. This process is divided into three parts: a program editor for inputting and modifying assembly language programs, a compiler for translating assembly language programs into machine language, and a relocating loader for loading a machine language program into a reserved section of memory and resolving its address usage to match the address available within its reserved section of memory.

Once an assembly language program has been inputted through the program editor, it is stored on cassette tape for future modification or compilation. Should you wish to modify an exiting program, the editor will first read your program back into its buffer, accept your modifications and then rewrite your program back to tape. Here is the BASIC program that will serve as our editor:

```
5 INPUT "HOW MUCH STRING SPACE SHALL I CLEAR";C:CLEAR C
1Ø INPUT "ARE YOU 1) CREATING OR 2) MODIFYING A PROGRAM";A
15 INPUT "MAXIMUM NUMBER OF INSTRUCTIONS IN YOUR PROGRAM";B
2Ø DIM K$(B):N=Ø
25 FOR I=1 TO B:K$ (I)=" ":NEXTI
3Ø IF A=2 THEN 2ØØ
35 PRINT "ENTER YOUR PROGRAM ONE INSTRUCTION AT A TIME"
4Ø IF N=B THEN PRINT "BUFFER FULL":GOTO 6Ø
45 INPUT M$
5Ø IF M$="/*" THEN 6Ø
55 N=N+1:K$(N)=M$:GOTO 4Ø
```

```
6Ø PRINT "READY CASSETTE TO SAVE YOUR PROGRAM"
65 FOR I=1 TO N
75 PRINT #-1,CHR$(34)+K$(I)+CHR$(34)
8Ø NEXT I:PRINT #-1,CHR$(34)+"/*"+CHR$(34)
85 INPUT "DO YOU WISH A PROGRAM LISTING";M$
9Ø IF M$.NE."YES" THEN END
95 FOR I=1 TO N:LPRINT I;K$(I):PRINT I;K$(I):NEXT I:END
2ØØ PRINT "READY CASSETTE TO LOAD YOUR PROGRAM"
2Ø5 INPUT #-1,M$
21Ø IF M$="/*" THEN 22Ø
215 N=N+1:K$(N)=M$:IF N=B THEN PRINT "BUFFER FULL":GOTO 22Ø ELSE 2Ø5
22Ø PRINT "L FOR LIST, I FOR INSERT, C FOR CHANGE, E TO FINISH"
225 INPUT "WHAT FUNCTION";M$:IF M$="E" THEN 6Ø
23Ø IF M$="I" THEN 3ØØ
235 IF M$="C" THEN 4ØØ
24Ø IF M$.NE."L" THEN 225
245 INPUT "ENTER FROM, TO LINE #S";R,T
25Ø IF R.GT.T OR R.LT.1 OR R.GT.N OR T.GT.N THEN 245
255 FOR I=R TO T:PRINT I;K$(I):NEXT I:GOTO 22Ø
3ØØ INPUT "PLACE INSERTS AFTER LINE #";R:IF R.LT.1 OR R.GT.N THEN 3ØØ
3Ø5 INPUT "NUMBER OF INSERTED LINES";T:IF T.LT.1 THEN 3Ø5
31Ø IF N+T.GT.B THEN PRINT "NOT ENOUGH ROOM":GOTO 3ØØ
315 S=N:FOR I=1 TO N-R:K$(S+T)=K$(S):S=S-1:NEXT I:N=N+T
32Ø PRINT "ENTER INSERTED LINES ONE INSTRUCTION AT A TIME"
325 FOR I=1 TO T:INPUT M$:K$(R+I)=M$:NEXT I:GOTO 22Ø
4ØØ INPUT "LINE # BEING CHANGED";R:IF R.GT.N OR R.LT.1 THEN 4ØØ
4Ø5 PRINT "OLD LINE";K$(R)
41Ø INPUT "NEW LINE";M$:K$(R)=M$:GOTO 22Ø
```

Note that when initially creating a program, the last line must be a "/*". Each line of instruction as entered or changed with this editor would consist of an operation code followed by a blank followed by up to two operands (if required) separated by only a comma. Naturally, the blank and comma would not be needed for an operation code requiring no operands.

If you wish to have comments in your program for documentation purposes, you may include them by placing them on separate lines starting with a ";" character. These comment lines will be ignored by the compiler.

When modifying a program, you may list portions of your program on the screen by specifying starting and ending line numbers, insert additional lines of instructions immediatly following a specified line number or just replace a current line of instruction with a new one. As you can see, you can also obtain a listing of your program on your line printer.

When using this editor, you must have some idea how big your program will be in terms of lines of instructions. This information must be used to determine how much string space to clear since the editor's program buffer resides in BASIC's string space and must hold the whole program at one time. Normally, 2Ø to 3Ø characters per line is ample.

The compiler program will read your assembly language program tape and translate it into a machine language program which will be outputted to tape for further processing and execution by the relocating loader program. Optionally, a line printer listing of the compilation may be obtained. Here is the BASIC program that will serve as our assembly language compiler:

```
1Ø INPUT "HOW MUCH STRING SPACE SHALL I CLEAR";C:CLEAR C
15 INPUT "MAXIMUM NUMBER OF INSTRUCTIONS IN YOUR PROGRAM";B
2Ø INPUT "MAXIMUM # OF TAGS IN YOUR PROGRAM";A
25 INPUT "MAXIMUM # OF BYTES IN YOUR MACHINE LANGUAGE PROGRAM";F
27 F=F+1Ø
3Ø DIM K$(B),J$(A),E%(A),H%(F),Z%(4)
35 G=(F/2)+5:DIM L%(G,2):N=Ø:P=Ø:Q=Ø:R=Ø:T=1:EC=Ø
4Ø INPUT "READY CASSETTE TO READ YOUR ASSEMBLY PROGRAM";YB$
45 INPUT #-1,M$:IF M$="/*" THEN 55
5Ø N=N+1:K$(N)=M$:IF N=B THEN PRINT "BUFFER FULL":GOTO 55 ELSE 45
55 FOR I=Ø TO A:J$(I)=" ":E%(I)=Ø:NEXT I
6Ø FOR I=Ø TO F:H%(I)=Ø:NEXT I
65 FOR I=Ø TO G:FOR V=Ø TO 2:L%(I,V)=Ø:NEXT V:NEXT I
7Ø INPUT "WANT A LINE PRINTER LISTING";YA$
75 PRINT "*LOC*NB*MACHINE CODE**  *ASSEMBLY TEXT*"
8Ø IF YA$="YES" THEN LPRINT "*LOC*NB*MACHINE CODE**  *ASSEMBLY TEXT*"
85 FOR I=1 TO N:Z%(1)=Ø:Z%(2)=Ø:Z%(3)=Ø:Z%(4)=Ø:T9=T:M$=K$(I):Z5=LEN(M$)
87 ER$=" ":OP$=" ":X1$=" ":X2$=" ":NB=Ø
9Ø IF LEFT$(M$,1)=";" THEN 9ØØØ
95 FOR Z6=1 TO Z5:IF MID$(M$,Z6,1)=" " THEN 1Ø5
1ØØ NEXT Z6:OP$=M$:GOTO 2ØØ
1Ø5 OP$=MID$(M$,1,Z6-1):Z6=Z6+1:IF Z6.GT.Z5 THEN 2ØØ
11Ø FOR Z7=Z6 TO Z5:IF MID$(M$,Z7,1)="," THEN 12Ø
115 NEXT Z7:X1$=MID$(M$,Z6):GOTO 2ØØ
12Ø X1$=MID$(M$,Z6,Z7-Z6):Z7=Z7+1:IF Z7.GT.Z5 THEN 2ØØ
125 X2$=MID$(M$,Z7)
2ØØ IF OP$.NE."TAG" THEN 23Ø
2Ø5 IF X1$=" " THEN 1225
2Ø7 LA$=X1$:GOSUB 21Ø:GOTO 9ØØØ
21Ø IF P=Ø THEN 225
215 FOR V=1 TO P:IF J$(V)=LA$ AND E%(V).GT.Ø THEN ER$="DUPLICATE LABEL":RET
    URN
217 IF J$(V)=LA$ THEN 227
22Ø NEXT V
225 P=P+1:V=P:IF P.GT.A THEN ER$="TOO MANY LABELS":RETURN
227 J$(V)=LA$:E%(V)=T:RETURN
23Ø IF OP$.NE."ORG" THEN 245
235 V=VAL(X1$):IF T+V.LT.Ø OR T+V.GT.F THEN ER$="INVALID ADDRESS":GOTO 9ØØØ
24Ø T=T+V:GOTO 9ØØØ
245 IF OP$.NE."DS" THEN 265
25Ø V=VAL(X1$):IF V.LT.1 OR V.GT.32767 OR T+V.GT.F THEN ER$="INVALID ALLOCA
    TION":GOTO 9ØØØ
255 IF X2$.NE." " THEN LA$=X2$:GOSUB 21Ø
26Ø IF ER$=" " THEN T=T+V
262 GOTO 9ØØØ
265 IF OP$.NE."DC" THEN 3Ø5
27Ø IF LEFT$(X1$,1).NE.CHR$(39) OR RIGHT$(X1$,1).NE.CHR$(39) THEN ER$="INVA
    LID SYNTAX":GOTO 9ØØØ
275 W=LEN(X1$)-2:IF W.LT.1 OR T+W.GT.F THEN ER$="INVALID SIZE":GOTO 9ØØØ
28Ø LA$=MID$(X1$,2,W)
285 FOR V=1 TO W:IF V.LT.5 THEN Z%(V)=ASC(MID$(LA$,V,1)):NB=NB+1
29Ø H%(V)=ASC(MID$(LA$,V,1)):NEXT V
295 IF X2$.NE." " THEN LA$=X2$:GOSUB 21Ø
3ØØ IF ER$=" " THEN T=T+W
3Ø2 GOTO 9ØØØ
3Ø5 IF OP$.NE."DB" THEN 35Ø
```

```
31Ø W=LEN(X18):IF W.NE.8 THEN 1225
315 V=128:ZA%=Ø:FOR Y=1 TO W
32Ø IF MID$(X1$,Y,1)="1" THEN ZA%=ZA%+V:GOTO 33Ø
325 IF MID$(X1$,Y,1).NE."Ø" THEN ER$="INVALID BIT":GOTO 9ØØØ
33Ø V=V/2:NEXT Y:H%(T)=ZA%:Z%(1)=ZA%:NB=1
332 IF T+1.GT.F THEN ER$="END OF MEMORY":GOTO 9ØØØ
335 IF X2$.NE." " THEN LA$=X2$:GOSUB 21Ø
34Ø IF ER$=" " THEN T=T+1
345 GOTO 9ØØØ
350 IF OP$.NE."DW" THEN 395
355 V=VAL(X1$):IF V.LT.Ø OR V.GT. 65535 THEN 1225
36Ø ZA%=INT(V/265):ZB%=V-(ZA%*256)
37Ø IF T+2.GT.F THEN ER$="END OF MEMORY":GOTO 9ØØØ
375 NB=NB+1:Z%(NB)=ZB%:NB=NB+1:Z%(NB)=ZA%:H%(T)=ZB%:H%(T+1)=ZA%
38Ø IF X2$.NE." " THEN LA$=X2$:GOSUB 21Ø
385 IF ER$=" " THEN T=T+2
39Ø GOTO 9ØØØ
395 IF OP$.NE."DN" THEN 42Ø
4ØØ V=VAL(X1$):IF V.LT.-128 OR V.GT.127 THEN 1225
4Ø5 IF V.LT.Ø THEN V=256-ABS(V)
415 NB=1:Z%(1)=V:H%(T)=V:GOTO 332
42Ø IF OP$.NE."DA" THEN 475 ELSE Y1=T
425 IF VAL(X1$).GT.Ø THEN 355 ELSE GOSUB 43Ø:GOTO 465
43Ø IF P=Ø THEN 445
435 FOR V=1 TO P:IF J$(V)=X1$ THEN 455
44Ø NEXT V
445 P=P+1:V=P:IF P.GT.A THEN ER$="TOO MANY LABELS":RETURN
45Ø J$(V)=X1$:E%(V)=Ø:Y5=V
455 Q=Q+1:IF Q.GT.G THEN ER$="TOO MANY RELOCATIONS":RETURN
46Ø L%(Q,1)=Y1:L%(Q,2)=V:RETURN
465 IF ER$.NE." " THEN 9ØØØ
47Ø V=E%(Y5):GOTO 36Ø
475 IF OP$.NE."DM" THEN 49Ø
48Ø V=VAL(X1$):IF V.LT.Ø OR V.GT.255 THEN 1225
485 GOTO 415
49Ø IF OP$.NE."RLC" THEN 6ØØ
495 LA$=X1$:GOSUB 5ØØ:GOTO 575
5ØØ V=-1:V1=-1:W=LEN(LA$):IF W.LT.2 THEN 53Ø
5Ø5 IF LEFT$(LA$,2)="IX" THEN NB=NB+1:Z%(NB)=221:GOTO 52Ø
51Ø IF LEFT$(LA$,2)="IY" THEN NB=NB+1:Z%(NB)=253:GOTO 52Ø
515 ER$="INVALID OPERAND":RETURN
52Ø V=6:IF W.GT.3 THEN 525 ELSE RETURN
525 V1=VAL(MID$(LA$,4)):RETURN
53Ø IF LA$="M" THEN V=6:RETURN
535 IF LA$="A" THEN V=7:RETURN
54Ø IF LA$="B" THEN V=Ø:RETURN
545 IF LA$="C" THEN V=1:RETURN
55Ø IF LA$="D" THEN V=2:RETURN
555 IF LA$="E" THEN V=3:RETURN
56Ø IF LA$="H" THEN V=4:RETURN
565 IF LA$="L" THEN V=5:RETURN
57Ø GOTO 515
575 IF V=-1 THEN 9ØØØ
577 NB=NB+1:Z%(NB)=2Ø3:NB=NB+1
58Ø IF V1.GT.-1 THEN Z%(NB)=V1:NB=NB+1
585 Z%(NB)=V
```

```
59Ø IF T+NB.GT.F THEN ER$="END OF MEMORY":GOTO 9ØØØ
595 FOR W=1 TO NB:H%(T)=Z%(W):T=T+1:NEXT W:GOTO 9ØØØ
6ØØ IF OP$.NE."RR" THEN 615
6Ø5 LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+24
61Ø GOTO 575
615 IF OP$.NE."RRC" THEN 63Ø
62Ø LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+8
625 GOTO 575
63Ø IF OP$.NE."SBC" THEN 66Ø
632 V=VAL(X1$):IF V.GT.Ø AND V.LT.256 THEN NB=2:Z%(1)=222:Z%(2)=V:GOTO 59Ø
635 LA$=X1$:GOSUB 5ØØ:W=152
64Ø IF V=-1 THEN 9ØØØ
645 V=V+W:NB=NB+1:Z%(NB)=V
65Ø IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
655 GOTO 59Ø
66Ø IF OP$.NE."SBCX" THEN 7Ø5
665 LA$=X1$:GOSUB 67Ø:GOTO 695
67Ø IF LA$="BC" THEN V=Ø:RETURN
675 IF LA$="DE" THEN V=16:RETURN
68Ø IF LA$="HL" THEN V=32:RETURN
685 IF LA$="SP" THEN V=48:RETURN
69Ø V=-1:ER$="INVALID OPERAND":RETURN
695 IF V=-1 THEN 9ØØØ
7ØØ NB=NB+1:Z%(NB)=237:NB=NB+1:Z%(NB)=V+66:GOTO 59Ø
7Ø5 IF OP$.NE."SCF" THEN 715
71Ø NB=1:Z%(1)=55:GOTO 59Ø
715 IF OP$.NE."SET" THEN 735 ELSE WA=192
72Ø W=VAL(X1$):IF W.LT.1 OR W.GT.8 THEN 1225
725 LA$=X2$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+WA+(ABS(W-8)*8)
73Ø GOTO 575
735 IF OP$.NE."SLA" THEN 75Ø
74Ø LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+32
745 GOTO 575
75Ø IF OP$.NE."SRA" THEN 765
755 LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+4Ø
76Ø GOTO 575
765 IF OP$.NE."SRL" THEN 78Ø
77Ø LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+56
775 GOTO 575
78Ø IF OP$.NE."SUB" THEN 795
785 W=VAL(X1$):IF W.GT.Ø AND W.LT.256 THEN NB=2:Z%(1)=214:Z%(2)=W:GOTO 59Ø
79Ø LA$=X1$:GOSUB 5ØØ:W=144:GOTO 64Ø
795 IF OP$.NE."XOR" THEN 81Ø
8ØØ W=VAL(X1$):IF W.GT.Ø AND W.LT.256 THEN NB=2:Z%(1)=238:Z%(2)=W:GOTO 59Ø
8Ø5 LA$=X1$:GOSUB 5ØØ:W=168:GOTO 64Ø
81Ø IF OP$.NE."IN" THEN 825 ELSE V=219
815 W=VAL(X1$):IF W.LT.Ø OR W.GT.255 THEN 1225
82Ø NB=2:Z%(1)=V:Z%(2)=W:GOTO 59Ø
825 IF OP$.NE."LDD" THEN 835
83Ø NB=2:Z%(1)=237:Z%(2)=168:GOTO 59Ø
835 IF OP$.NE."LDDR" THEN 845
84Ø NB=2:Z%(1)=237:Z%(2)=184:GOTO 59Ø
845 IF OP$.NE."LDI" THEN 855
85Ø NB=2:Z%(1)=237:Z%(2)=16Ø:GOTO 59Ø
855 IF OP$.NE."LDIR" THEN 865
86Ø NB=2:Z%(1)=237:Z%(2)=176:GOTO 59Ø
865 IF OP$.NE."NEG" THEN 875
87Ø NB=2:Z%(1)=237:Z%(2)=68:GOTO 59Ø
```

```
875 IF OP$.NE."NOP" THEN 885
88Ø NB=1:Z%(1)=Ø:GOTO 59Ø
885 IF OP$.NE."OR" THEN 9ØØ
89Ø W=VAL(X1$):IF W.GT.Ø AND W.LT.256 THEN NB=2:Z%(1)=246:Z%(2)=W:GOTO 59Ø
895 LA$=X1$:GOSUB 5ØØ:W=176:GOTO 64Ø
9ØØ IF OP$.NE."OUT" THEN 91Ø
9Ø5 V=211:GOTO 815
91Ø IF OP$.NE."POP" THEN 965
915 LA$=X1$:GOSUB 92Ø:GOTO 955
92Ø IF LA$="AF" THEN V=241:RETURN
925 IF LA$="BC" THEN V=193:RETURN
93Ø IF LA$="DE" THEN V=2Ø9:RETURN
935 IF LA$="HL" THEN V=225:RETURN
94Ø IF LA$="IX" THEN V=225:NB=NB+1:Z%(NB)=221:RETURN
945 IF LA$="IY" THEN V=225:NB=NB+1:Z%(NB)=253:RETURN
95Ø V=-1:ER$="INVALID OPERAND":RETURN
955 IF V=-1 THEN 9ØØØ
96Ø NB=NB+1:Z%(NB)=V:GOTO 59Ø
965 IF OP$.NE."PUSH" THEN 98Ø
97Ø LA$=X1$:GOSUB 92Ø:IF V.GT.-1 THEN V=V+4
975 GOTO 955
98Ø IF OP$.NE."RES" THEN 985 ELSE WA=128:GOTO 72Ø
985 IF OP$.NE."RL" THEN 1ØØØ
99Ø LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN V=V+16
995 GOTO 575
1ØØØ IF OP$.NE."INC" THEN 1Ø55
1ØØ5 LA$=X1$:GOSUB 5ØØ:IF V=-1 THEN 9ØØØ
1Ø1Ø IF V=Ø THEN W=4
1Ø15 IF V=1 THEN W=12
1Ø2Ø IF V=2 THEN W=2Ø
1Ø25 IF V=3 THEN W=28
1Ø3Ø IF V=4 THEN W=36
1Ø35 IF V=5 THEN W=44
1Ø4Ø IF V=6 THEN W=52
1Ø42 IF V=7 THEN W=6Ø
1Ø45 NB=NB+1:Z%(NB)=W:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
1Ø5Ø GOTO 59Ø
1Ø55 IF OP$.NE."INCX" THEN 1Ø85
1Ø57 IF X1$="SP" THEN W=51:GOTO 1Ø8Ø
1Ø6Ø LA$=X1$:GOSUB 925:IF V=-1 THEN 9ØØØ
1Ø65 IF V=193 THEN W=3
1Ø7Ø IF V=2Ø9 THEN W=19
1Ø75 IF V=225 THEN W=35
1Ø8Ø NB=NB+1:Z%(NB)=W:GOTO 59Ø
1Ø85 IF OP$.NE."JP" THEN 119Ø
1Ø9Ø W=VAL(X1$):IF W.LT.1 OR W.GT.9 THEN 1225
1Ø95 IF W=9 AND X2$="M" THEN NB=1:Z%(NB)=223:GOTO 59Ø
11ØØ IF W=9 AND X2$="IX" THEN NB=2:Z%(1)=221:Z%(2)=233:GOTO 59Ø
11Ø5 IF W=9 AND X2$="IY" THEN NB=2:Z%(1)=253:Z%(2)=233:GOTO 59Ø
111Ø IF W=1 THEN V=218
1115 IF W=2 THEN V=21Ø
112Ø IF W=3 THEN V=25Ø
1125 IF W=4 THEN V=242
113Ø IF W=5 THEN V=2Ø2
1135 IF W=6 THEN V=194
114Ø IF W=7 THEN V=234
1145 IF W=8 THEN V=226
```

```
115Ø IF W=9 THEN V=195
1155 NB=NB+1:Z%(NB)=V:V=VAL(X2$):Y1=T+1
116Ø IF V.GT.Ø AND V.LT.65536 THEN 1165 ELSE 118Ø
1165 ZA%=INT(V/256):V=V-(ZA%*256)
1175 NB=NB+1:Z%(NB)=V:NB=NB+1:Z%(NB)=ZA%:GOTO 59Ø
118Ø X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
1185 V=E%(Y5):GOTO 1165
119Ø IF OP$.NE."JR" THEN 125Ø
1195 W=VAL(X1$):IF W.LT.1 OR W.GT.9 THEN 1225
12ØØ IF W=1 THEN V=56:GOTO 123Ø
12Ø5 IF W=9 THEN V=24:GOTO 123Ø
121Ø IF W=2 THEN V=48:GOTO 123Ø
1215 IF W=6 THEN V=32:GOTO 123Ø
122Ø IF W=5 THEN V=4Ø:GOTO 123Ø
1225 ER$="INVALID OPERAND":GOTO 9ØØØ
123Ø W=VAL(X2$):IF W.LT.-128 OR W.GT.127 THEN 1225
1235 IF W.LT.Ø THEN W=256-ABS(W)
1245 NB=NB+1:Z%(NB)=V:NB=NB+1:Z%(NB)=W:GOTO 59Ø
125Ø IF OP$.NE."LD" THEN 1425 ELSE Y1=T+1
1255 IF X1$="@BC" AND X2$="A" THEN NB=1:Z%(1)=2:GOTO 59Ø
126Ø IF X1$="@DE" AND X2$="A" THEN NB=1:Z%(1)=18:GOTO 59Ø
1265 V=VAL(X1$):IF V.GT.Ø AND V.LT.65536 AND X2$="A" THEN NB=1:Z%(1)=5Ø:GOT
     O 1165
127Ø LA$=X1$:GOSUB 5ØØ:IF V.GT.-1 THEN 1282
1272 IF X2$.NE."A" THEN 9ØØØ
1275 ER$=" ":GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
128Ø V=E%(Y5):NB=1:Z%(1)=5Ø:GOTO 1165
1282 Y2=V:Y4=V1
1285 IF X1$="A" AND X2$="@BC" THEN NB=1:Z%(1)=1Ø:GOTO 59Ø
129Ø IF X1$="A" AND X2$="@DE" THEN NB=1:Z%(1)=26:GOTO 59Ø
1295 V=VAL(X2$):IF X1$="A" AND V.GT.255 AND V.LT.65536 THEN NB=1:Z%(1)=58:G
     OTO 1165
1296 V=Ø:W=Ø
1297 W=VAL(X2$):IF W.GT.Ø AND W.LT.256 THEN 132Ø
13ØØ LA$=X2$:GOSUB 5ØØ:IF V.GT.-1 THEN 132Ø
13Ø5 IF X1$.NE."A" THEN 9ØØØ
131Ø ER$=" ":GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
1315 V=E%(Y5):NB=1:Z%(1)=58:GOTO 1165
132Ø Y3=V:ON Y2+1 GOTO 1325, 1335, 1345, 1355, 1365, 1375, 1385, 1415
1325 IF W.GT.Ø THEN NB=2:Z%(1)=6:Z%(2)=W:GOTO 59Ø
133Ø NB=NB+1:Z%(NB)=64+Y3:GOTO 65Ø
1335 IF W.GT.Ø THEN NB=2:Z%(1)=14:Z%(2)=W:GOTO 59Ø
134Ø NB=NB+1:Z%(NB)=72+Y3:GOTO 65Ø
1345 IF W.GT.Ø THEN NB=2:Z%(1)=22:Z%(2)=W:GOTO 59Ø
135Ø NB=NB+1:Z%(NB)=8Ø+Y3:GOTO 65Ø
1355 IF W.GT.Ø THEN NB=2:Z%(1)=3Ø:Z%(2)=W:GOTO 59Ø
136Ø NB=NB+1:Z%(NB)=88+Y3:GOTO 65Ø
1365 IF W.GT.Ø THEN NB=2:Z%(1)=38:Z%(2)=W:GOTO 59Ø
137Ø NB=NB+1:Z%(NB)=96+Y3:GOTO 65Ø
1375 IF W.GT.Ø THEN NB=2:Z%(1)=46:Z%(2)=W:GOTO 59Ø
138Ø NB=NB+1:Z%(NB)=104+Y3:GOTO 65Ø
1385 IF Y4.GT.-1 THEN 14Ø5
139Ø IF W.GT.Ø THEN NB=2:Z%(1)=54:Z%(2)=W:GOTO 59Ø
1395 IF Y3=6 THEN 1225
14ØØ NB=1:Z%(1)=112+Y3:GOTO 65Ø
14Ø5 IF W.GT.Ø THEN NB=4:Z%(2)=54:Z%(3)=Y4:Z%(4)=W:GOTO 59Ø
```

```
141Ø NB=3:Z%(2)=112+Y3:Z%(3)=Y4:GOTO 59Ø
1415 IF W.GT.Ø THEN NB=2:Z%(1)=62:Z%(2)=W:GOTO 59Ø
142Ø NB=NB+1:Z%(NB)=12Ø+Y3:GOTO 65Ø
1425 IF OP$.NE."LDX" THEN 1655
143Ø W=VAL(X2$):V=VAL(X1$)
1435 IF V.GT.Ø AND V.LT.65536 THEN 144Ø ELSE 1475
144Ø IF X2$="BC" THEN NB=2:Z%(1)=237:Z%(2)=67:GOTO 1165
1445 IF X2$="DE" THEN NB=2:Z%(1)=237:Z%(2)=83:GOTO 1165
145Ø IF X2$="HL" THEN NB=1:Z%(1)=34:GOTO 1165
1455 IF X2$="IX" THEN NB=2:Z%(1)=221:Z%(2)=34:GOTO 1165
146Ø IF X2$="IY" THEN NB=2:Z%(1)=253:Z%(2)=34:GOTO 1165
1465 IF X2$="SP" THEN NB=2:Z%(1)=237:Z%(2)=115:GOTO 1165
147Ø GOTO 1225
1475 IF X1$.NE."BC" THEN 15ØØ
148Ø IF W.GT.Ø AND W.LT.65536 THEN NB=2:Z%(1)=237:Z%(2)=75:V=W:GOTO 1165
1485 IF LEFT$(X2$,1)="#" THEN V=VAL(MID$(X2$,2)):NB=1:Z%(1)=1:GOTO 1165
149Ø Y1=T+2:X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
1495 V=E%(Y5):NB=2:Z%(1)=237:Z%(2)=75:GOTO 1165
15ØØ IF X1$.NE."DE" THEN 1525
15Ø5 IF W.GT.Ø AND W.LT.65536 THEN NB=2:Z%(1)=237:Z%(2)=91:V=W:GOTO 1165
151Ø IF LEFT$(X2$,1)="#" THEN V=VAL(MID$(X2$,2)):NB=1:Z%(1)=17:GOTO 1165
1515 Y1=T+2:X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
152Ø V=E%(Y5):NB=2:Z%(1)=237:Z%(2)=91:GOTO 1165
1525 IF X1$.NE."HL" THEN 155Ø
153Ø IF W.GT.Ø AND W.LT.65536 THEN NB=1:Z%(1)=42:V=W:GOTO 1165
1535 IF LEFT$(X2$,1)="#" THEN V=VAL(MID$(X2$,2)):NB=1:Z%(1)=33:GOTO 1165
154Ø Y1=T+2:X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
1545 V=E%(Y5):NB=1:Z%(1)=42:GOTO 1165
155Ø IF X1$.NE."IX" THEN 1575
1555 IF W.GT.Ø AND W.LT.65536 THEN NB=2:Z%(1)=221:Z%(2)=42:V=W:GOTO 1165
156Ø IF LEFT$(X2$,1)="#" THEN V=VAL(MID$(X2$,2)):NB=2:Z%(1)=221:Z%(2)=33:GO
     TO 1165
1565 Y1=T+2:X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
157Ø V=E%(Y5):NB=2:Z%(1)=221:Z%(2)=42:GOTO 1165
1575 IF X1$.NE."IY" THEN 16ØØ
158Ø IF W.GT.Ø AND W.LT.65536 THEN NB=2:Z%(1)=253:Z%(2)=42:V=W:GOTO 1165
1585 IF LEFT$(X2$,1)="#" THEN NB=2:V=VAL(MID$(X2$,2)):Z%(1)=253:Z%(2)=33:GO
     TO 1165
159Ø Y1=T+2:X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
1595 V=E%(Y5):NB=2:Z%(1)=253:Z%(2)=42:GOTO 1165
16ØØ IF X1$.NE."SP" THEN 164Ø
16Ø5 IF X2$="HL" THEN NB=1:Z%(1)=249:GOTO 59Ø
161Ø IF X2$="IX" THEN NB=2:Z%(1)=221:Z%(2)=249:GOTO 59Ø
1615 IF X2$="IY" THEN NB=2:Z%(1)=253:Z%(2)=249:GOTO 59Ø
162ØIF W.GT.Ø AND W.LT.65536 THEN NB=2:Z%(1)=237:Z%(2)=123:V=W:GOTO 1165
1625 IF LEFT$(X2$,1)="#" THEN NB=1:V=VAL(MID$(X2$,2)):Z%(1)=49:GOTO 1165
163Ø Y1=T+2:X1$=X2$:GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
1635 V=E%(Y5):NB=2:Z%(1)=237:Z%(2)=123:GOTO 1165
164Ø Y1=T+2:IF X2$="HL" THEN Y1=T+1
1645 GOSUB 43Ø:IF ER$.NE." " THEN 9ØØØ
165Ø V=E%(Y5):GOTO 144Ø
1655 IF OP$.NE."CPIR" THEN 1665
166Ø NB=2:Z%(1)=237:Z%(2)=177:GOTO 59Ø
1665 IF OP$.NE."CPL" THEN 1675
167Ø NB=1:Z%(1)=47:GOTO 59Ø
1675 IF OP$.NE."DEC" THEN 1735
168Ø LA$=X1$:GOSUB 5ØØ:IF V=-1 THEN 9ØØØ
```

```
1685 IF V=0 THEN W=5
1690 IF V=1 THEN W=13
1695 IF V=2 THEN W=21
1700 IF V=3 THEN W=29
1705 IF V=4 THEN W=37
1710 IF V=5 THEN W=45
1715 IF V=6 THEN W=53
1720 IF V=7 THEN W=61
1725 NB=NB+1:Z%(NB)=W:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
1730 GOTO 590
1735 IF OP$.NE."DECX" THEN 1770
1740 IF X1$="SP" THEN W=59:GOTO 1765
1745 LA$=X1$:GOSUB 925:IF V=-1 THEN 9000
1750 IF V=193 THEN W=11
1755 IF V=209 THEN W=27
1760 IF V=225 THEN W=43
1765 NB=NB+1:Z%(NB)=W:GOTO 590
1770 IF OP$.NE."DJNZ" THEN 1785
1775 W=VAL(X1$):IF W.LT.-128 OR W.GT.127 THEN 1225
1777 IF W.LT.0 THEN W=256-ABS(W)
1780 NB=2:Z%(1)=16:Z%(2)=W:GOTO 590
1785 IF OP$.NE."EXS" THEN 1810
1790 IF X1$="HL" THEN NB=1:Z%(1)=227:GOTO 590
1795 IF X1$="IX" THEN NB=2:Z%(1)=221:Z%(2)=227:GOTO 590
1800 IF X1$="IY" THEN NB=2:Z%(1)=253:Z%(2)=227:GOTO 590
1805 GOTO 1225
1810 IF OP$.NE."EXD" THEN 1820
1815 NB=1:Z%(1)=235:GOTO 590
1820 IF OP$.NE."EXA" THEN 1830
1825 NB=1:Z%(1)=8:GOTO 590
1830 IF OP$.NE."EXX" THEN 1840
1835 NB=1:Z%(1)=217:GOTO 590
1840 IF OP$.NE."BIT" THEN 1860
1845 W=VAL(X1$):IF W.LT.1 OR W.GT.8 THEN 1225
1850 LA$=X2$:GOSUB 500:IF V.GT.-1 THEN V=V+64+(ABS(W-8)*8)
1855 GOTO 575
1860 IF OP$.NE."CALL" THEN 1930 ELSE Y1=T+1
1862 W=VAL(X1$):IF W.LT.1 OR W.GT.9 THEN 1225
1865 V=VAL(X2$):IF V.GT.0 AND V.LT.65536 THEN 1880
1870 X1$=X2$:GOSUB 430:IF ER$.NE." " THEN 9000
1875 V=E%(Y5)
1880 IF W=1 THEN Y2=220
1885 IF W=2 THEN Y2=212
1890 IF W=3 THEN Y2=252
1895 IF W=4 THEN Y2=244
1900 IF W=5 THEN Y2=204
1905 IF W=6 THEN Y2=196
1910 IF W=7 THEN Y2=236
1915 IF W=8 THEN Y2=228
1920 IF W=9 THEN Y2=205
1925 NB=NB+1:Z%(NB)=Y2:GOTO 1165
1930 IF OP$.NE."CCF" THEN 1940
1935 NB=1:Z%(1)=63:GOTO 590
1940 IF OP$.NE."CP" THEN 1965
1945 W=VAL(X1$):IF W.GT.0 AND W.LT.256 THEN NB=2:Z%(1)=254:Z%(2)=W:GOTO 590
1950 LA$=X1$:GOSUB 500:IF V=-1 THEN 9000
1955 NB=NB+1:Z%(NB)=184+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
1960 GOTO 590
```

```
1965 IF OP$.NE."CPD" THEN 1975
197Ø NB=2:Z%(1)=237:Z%(2)=169:GOTO 59Ø
1975 IF OP$.NE."CPDR" THEN 1985
198Ø NB=2:Z%(1)=237:Z%(2)=185:GOTO 59Ø
1985 IF OP$.NE."CPI" THEN 1995
199Ø NB=2:Z%(1)=237:Z%(2)=161:GOTO 59Ø
1995 IF OP$.NE."ADC" THEN 2Ø2Ø
2ØØØ W=VAL(X1$):IF W.GT.Ø AND W.LT.256 THEN NB=2:Z%(1)=2Ø6:Z%(2)=W:GOTO 59Ø
2ØØ5 LA$=X1$:GOSUB 5ØØ:IF V=-1 THEN 9ØØØ
2Ø1Ø NB=NB+1:Z%(NB)=136+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
2Ø15 GOTO 59Ø
2Ø2Ø IF OP$.NE."ADCX" THEN 2Ø55
2Ø25 IF X1$="BC" THEN W=74:GOTO 2Ø5Ø
2Ø3Ø IF X1$="DE" THEN W=9Ø:GOTO 2Ø5Ø
2Ø35 IF X1$="HL" THEN W=1Ø6:GOTO 2Ø5Ø
2Ø4Ø IF X1$="SP" THEN W=122:GOTO 2Ø5Ø
2Ø45 GOTO 1225
2Ø5Ø NB=2:Z%(1)=237:Z%(2)=W:GOTO 59Ø
2Ø55 IF OP$.NE."ADD" THEN 2Ø8Ø
2Ø6Ø W=VAL(X1$):IF W.GT.Ø AND W.LT.256 THEN NB=2:Z%(1)=198:Z%(2)=W:GOTO 59Ø
2Ø65 LA$=X1$:GOSUB 5ØØ:IF V=-1 THEN 9ØØØ
2Ø7Ø NB=NB+1:Z%(NB)=128+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
2Ø75 GOTO 59Ø
2Ø8Ø IF OP$.NE."ADDX" THEN 2135
2Ø85 IF X1$.NE."HL" THEN 2115
2Ø87 NB=NB+1
2Ø9Ø IF X2$="BC" THEN Z%(NB)=9:GOTO 59Ø
2Ø95 IF X2$="DE" THEN Z%(NB)=25:GOTO 59Ø
21ØØ IF X2$="HL" THEN Z%(NB)=41:GOTO 59Ø
21Ø5 IF X2$="SP" THEN Z%(NB)=57:GOTO 59Ø
211Ø GOTO 1225
2115 IF X1$.NE."IX" THEN 2125
212Ø NB=NB+1:Z%(NB)=221:GOTO 2Ø87
2125 IF X1$.NE."IY" THEN 1225
213Ø NB=NB+1:Z%(NB)=253:GOTO 2Ø87
2135 IF OP$.NE."AND" THEN ER$="INVALID OP CODE":GOTO 9ØØØ
214Ø W=VAL(X1$):IF W.GT.Ø AND W.LT.256 THEN NB=2:Z%(1)=23Ø:Z%(2)=W:GOTO 59Ø
2145 LA$=X1$:GOSUB 5ØØ:IF V=-1 THEN 9ØØØ
215Ø NB=NB+1:Z%(NB)=16Ø+V:IF V1.GT.-1 THEN NB=NB+1:Z%(NB)=V1
2155 GOTO 59Ø
9ØØØ PRINT T9;NB;Z%(1);Z%(2);Z%(3);Z%(4);" ";K$(I)
9ØØ5 IF YA$="YES" THEN LPRINT T9;NB;Z%(1);Z%(2);Z%(3);Z%(4);" ";K$(I)
9Ø1Ø IF ER$.NE." " THEN PRINT ER$:PRINT:EC=EC+1
9Ø15 IF ER$.NE." " AND YA$="YES" THEN LPRINT ER$:LPRINT" "
9Ø2Ø NEXT I:IF P=Ø THEN 9Ø5Ø
9Ø25 FOR I=1 TO P:IF E%(I)=Ø THEN 9Ø3Ø ELSE 9Ø4Ø
9Ø3Ø PRINT "LABEL UNKNOWN-";J$(I):EC=EC+1
9Ø35 IF YA$="YES" THEN LPRINT "LABEL UNKNOWN-";J$(I)
9Ø4Ø NEXT I:FOR V=Ø TO Q:Z7=L%(V,2):L%(V,2)=E%(Z7):NEXT V
9Ø5Ø IF EC.GT.Ø THEN PRINT "ERRORS=";EC:END
9Ø55 INPUT "READY CASSETTE TO WRITE MACHINE LANGUAGE";YB$
9Ø6Ø PRINT #-1,T;Q:I=Ø
9Ø65 PRINT #-1,"T",H%(I),H%(I+1),H%(I+2),H%(I+3),H%(I+4),H%(I+5),H%(I+6),H%
     (I+7),H%(I+8),H%(I+9)
9Ø7Ø I=I+10:IF I.LE.T THEN 9Ø65
9Ø75 IF Q=Ø THEN 9Ø95
9Ø8Ø I=1
9Ø85 PRINT #-1,"Q",L%(I,1),L%(I,2),L%(I+1,1),L%(I+1,2),L%(I+2,1),L%(I+2,2),
      L%(I+3,1),L%(I+3,2),L%(I+4,1),L%(I+4,2)
9Ø9Ø I=I+5:IF I.LE.Q THEN 9Ø85
9Ø95 PRINT #-1,"E",Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø,Ø:END
```

The amount of string space you must clear for a compilation relates to the compiler tables. In creating a machine language program from your assembly language program, the compiler must build and maintain several tables. The K$ table holds your assembly language programs. The J$ and E% tables hold the tags and field names and their location addresses within your program. The H% table holds your machine language program. The L% table holds the location of an address in your program requiring relocation when your program is loaded for execution, and the relative address of the tag or field name at the location requiring the program's load address to be added to it and placed in your machine code. The J$, E% and L% tables are maintained by two compiler subroutines starting at line 43Ø and at line 21Ø. If no errors are found during the compilation, your machine language program will be written to tape in a format compatable with the relocating loader program. During the compilation, the program will ask if a listing of the compilation will be required on the line printer. As you can see, each assembly language instruction is covered in a separate section of the compiler program. You can study the machine language coding of any assembly language instruction be examining the appropriate section. In this version of the compiler, the DJNZ and JR instructions only allow an immediate value operand as the displacement value from -128 to +127 in jumping from these instructions.

Here is the relocating loader program you will need to load and execute your machine language program.

```
5 INPUT "NUMBER OF RELOCATION ADDRESSES";K:IF K.LT.1 OR K.GT.32768 THEN 5
7 DIM L%(K,2)
1Ø INPUT "LOAD ADDRESS";A:IF A.LT.16384 OR A.GT.65535 THEN 1Ø
2Ø INPUT "MACHINE LANGUAGE PROGRAM AREA SIZE";B:IF B.LT.Ø OR K*2.GT.B OR B.
   GT.32767 OR A+B.GT.65535 THEN 2Ø
3Ø INPUT "READY CASSETTE TO READ MACHINE LANGUAGE PROGRAM";C$
4Ø INPUT #-1,T,Q:I=Ø:AX=A:J=1:IF T.GT.B THEN PRINT "TEXT SIZE ERROR":END
42 IF Q.GT.K THEN PRINT "RELOCATION TABLE SIZE ERROR":END
5Ø INPUT #-1,C$,H1,H2,H3,H4,H5,H6,H7,H8,H9,HA
55 IF C$.NE."T" THEN PRINT "LOAD TEXT ERROR":END
6Ø POKE A,H1:POKE A+1,H2:POKE A+2,H3:POKE A+3,H4
7Ø POKE A+4,H5:POKE A+5,H6:POKE A+6,H7:POKE A+7,H8
8Ø POKE A+8,H9:POKE A+9,HA:I=I+1Ø:IF I.LE.T THEN 5Ø
9Ø INPUT #-1,C$,H1,H2,H3,H4,H5,H6,H7,H8,H9,HA
1ØØ IF C$="E" THEN 16Ø
11Ø IF C$.NE."Q" THEN PRINT "RELOCATION TABLE ERROR":END
12Ø L%(J,1)=H1:L%(J,2)=H2:L%(J+1,1)=H3
13Ø L%(J+1,2)=H4:L%(J+2,1)=H5:L%(J+2,2)=H6
14Ø L%(J+3,1)=H7:L%(J+3,2)=H8:L%(J+4,1)=H9
15Ø L%(J+4,2)=HA:J=J+5:IF J.LE.Q THEN 9Ø
16Ø IF J=1 THEN 21Ø
17Ø FOR I=1 TO J:IF L%(I,1)=Ø THEN 2ØØ
18Ø Z%=A+L%(I,2):H1=L%(I,1):W%=INT(Z%/256):Z%=Z%-(W%*256)
19Ø POKE H1,Z%:POKE H1+1,W%
2ØØ NEXT I
21Ø INPUT "SHOULD I EXECUTE YOUR PROGRAM";C$
22Ø IF C$.NE."YES" THEN END
23Ø Z%=INT(A/256):W%=A-(Z%*256)
```

```
240 POKE 16526,W%:POKE 16527,Z%
250 X=USR(0):END
```

Note that the load address, machine program size, and relocation address table size must match or exceed the capacities of those parameters as defined during the compilation of your program. The loader first processes the text records of machine code read from the tape and then reads all the address relocation records for further processing. Once the address relocation has been accomplished, the program may be executed. If the program is a subroutine to be called from one of your BASIC programs, the loader ends so that you may load and execute your BASIC program.

I think you will agree that this system for using assembly language and machine language programs is both simple and beneficial. Now all you need is some imagination and you can develope many interesting and fast running programs with features beyond BASIC's capabilities.

....

CYBERMATE AND THE 80 NOTEBOOK GOING OUT OF BUSINESS

Due to overwhelming competition (trying to get a piece of the market) from the over 3000 small TRS-80 products firms, sources of cost effective advertising (since the CIE S-80 Bulletin went out of business in August), very poor circulation (presently about 350) and poor sales this past summer, we are forced to go out of business. We appreciate your interest in our company's products and publications and wish to leave you on a friendly basis.

Due to our current poor financial condition, we are offering the following: for those of you with one year subscriptions, we are including a copy of our 41 program listing package with this last issue of The 80 Notebook. Our package is an ideal substitute for the remainder of a one year subscription and it represents more material than was planned for the remainder of the first subscription year. If you already own one of our 41 program packages, keep this second copy and you may request any two programs on cassette from our product line of 41 programs.

Our two year subscribers can request 5 programs on cassette as a substitute to their second year of The 80 Notebook.

Our three year subscribers can request 10 programs on cassette as a substitute to their second and third years of The 80 Notebook.

We hope this will be satisfactory as we have done our best to be fair with the resources we have left.

GOOD-BYE AND GOOD LUCK!