Processor Technology Co.

2465 Fourth Street
Berkeley, Ca. 94710
(415) 549-0857

August 28, 1975

Dear Customer:

Enclosed you will find Software Package #1, our Self-contained Assembly Language Operating System. Over the past months we have found this System to be an extremely useful and poweful development tool. We are sure you will agree with us once you have tried it in your 8080 system.

Paper tapes or cassettes of the enclosed program listing will not be available to individuals but we have already sent paper tapes to several computer clubs around the country. We suggest you contact one of these clubs if you want a copy of the tape or need assistance.

We will be happy to send tape copies to any bona fide "amateur" computer club or society, so if you are a member of such a group, please let us know of your group's existence by sending us a copy of its latest newsletter.

In addition we are now preparing a manual describing the use of the System from the ground up. This will include a complete description, with examples, of every command, instructions on the use of all internal routines by other programs, and an overview of efficient file generation and handling. This loose leaf manual, complete with ring binder, is being sold for $25.00. Orders for the manual will be accepted only until November 1, 1975 with delivery beginning about Oct. 1st.

* * * * * * * * *

An expanded version of Package #1 will soon be available as a 4K Byte PROM module. The expanded version version allows dynamic Input/Output allocation, file area management by the executive, octal and/or hex data entry and many other capabilities not included in the original package #1. These PROMs will be sold with a module capable of holding 8K of PROM for $275.00 with everything needed to just plug into the computer and run. Why 8K???? Because we are leaving space for future program expansion. The first expansion is a powerful simulator that adds-on to the basic package.

SIMULATOR???? Yes, an Interpretive Simulator which runs 8080 programs on the same 8080 that contains the simulator!! Not just traps and breakpoints but simulated I/O, registers, flags, program counter and stack pointer. Any of these can be modified at all times, plus a single step mode that displays all registers, flags, stack pointer, program counter and memory contents after execution of each instruction. Take a look at the enclosed print out from the simulator program which demonstrates a small part of the true capabilities of the simulator. This Program must be run in conjunction with the 4K set and is being sold to purchasors of the set for $95.00.

Will paper tapes be available? We imagine tapes will appear but, using 8K of RAM to receive a paper tape program that takes the better part of a half hour to load each time a power glitch occurs just doesn't make for a real system. Using this module will save that expensive RAM for data and development programs as well as giving the true convinience of a "turn on the switch" system.

Speaking of expensive RAM we should explain how financing for our "FREE" software comes about. First we hope to sell a good number of PROM firmware modules with Interpretive Simulator and other nice expanded capabilities. Second, the full power of a computer or processor is directly related to the amount of memory available for storage of programs and data. Note that Package #1 occupies about 4K for the operating program and 2K for the System RAM and symbol table. In order to develop reasonably sized programs at least 8K more of RAM is required.

Now, since RAM is needed, Processor Technology makes the least expensive, fastest, lowest power and most reliable (every RAM IC is tested to Military MIL STD 883 specifications) 4K read/write memory module available today. But, bless our conniving little hearts, we have just lowered our price on RAM modules. The 4KRA-4 is now priced at $215.00 in kit form, and we have a special offer for users of Software Package #1.

* S P E C I A L *

All orders for 2 (two) 4KRA-4 modules received before October 1st, 1975 will receive a special software package containing, among other things, a 12 decimal digit floating point math package complete with instructions and handy pointers. Orders for this special must use the enclosed order form.

* * *

SPECIFICATIONS: ALS-8 Firmware module, SIM-1 Firmware module

Maximum capacity: 8192 8-bit words
Memory type: 4K static erasable Programmable ROM (up to 16 PROMs)
Access time: 1.0 sec maximum
Cycle time: 1.0 sec maximum
Power Requirements: +8VDC @ .4A maximum with SIM-1;-16VDC @ .3A max.
Operating range: +5 C to +60 C, to 90% humidity, non condensing
Dimensions: 5.3" x 10.0" (13.46cm x 25.4cm)

PROCESSOR TECHNOLOGY CORPORATION
SPECIAL ORDER FORM- SEPT. 1, 1975
KIT PRICES SHOWN

		Totals
☐	Send___ALS-8 PROM Firmware modules @ $275 ea.	$_______
☐	Send___SIM-1 Simulator Firmware modules @ $95 ea. (plugs into ALS-8)	_______
☐	Send___Assembly Language System Manual(s) only @ $25 (ALS-8 price includes manual)	_______
☐	Send___4KRA-4 Static Read/write memory modules @ $215	_______
☐	I ordered two or more 4KRA modules, so send me the special math package too!	_______
☐	I need even more hardware for my system, so send me the following goodies too!	
	___ 2KRO EPROM module for 1702A or 5203 PROMs @ $50.	_______
	___ 3P+S Input/Output Module (2 parallel, 1 serial port plus TTY and RS-232 interfaces) @ $125	_______
	___ MB-1 Mother Board, one piece, 16 slots @ $50 (with Bus Terminator included)	_______
	___ VDM Video Display Module, displays 16 lines of 64 characters each on any standard video monitor, comes with 1K of RAM on card and includes driving software. @ $160 ea.	_______
	Subtotal	_______
	Less % Discount	_______
	Taxable subtotal	_______
	California residents, 6% tax	_______
	TOTAL	_______

Discounts: Orders over $375 may subtract 5%; orders over $600 may subtract 10%.
ALL ORDERS POSTPAID IF FULL PAYMENT ACCOMPANIES ORDER!

We accept BankAmericard and Mastercharge.

PROCESSOR TECHNOLOGY CORP.
2465 Fourth Street
Berkeley, Calif
94710

SOFTWARE PACKAGE NO. 1

The Processor Technology Software Package #1 is a Self Contained Program Development System for any computer based on the Intel 8080 micro-processor. Included in the package is an executive to handle memory files, an assembler, and a line oriented editor.

To use the system 6K of memory must be available for use by the system. This memory is allocated as follows:

F000 - FFFF	Operating Program	~4K
D000 - D0FF	Special System RAM	2K
D100 - D7FF	Symbol Table (Assembler Only)	

In addition other memory must be available for source and object files necessary for the users programs.

I/O within the program interacts with I/O ports addressed as follows:

PORT	FUNCTION
0	Status Input Bit 6 indicates DAV Bit 7 indicates TBE
1	TTY Input
FF	Sense Switch Input Sense switch seven is used to control file listing.

EXECUTIVE COMMANDS:

CONTROL X	System reset and CR/LF
ENTR	Enter data to memory
DUMP	Display memory data
FILE	Create, assign or display file information
EXEC	Execute a program
ASSM	Assemble a source file to object code
PROG	Program a PROM (1702)
LIST	List File
DELT	Delete Lines of file.
1111	Any four numeric digits enters editor
PAGE	Move a page of data
CUST	Optional user command at location E000

The executive has one error message.....WHAT?.....indicating an improper command or an error on parameters following the command.

COMMAND FORMAT

ENTR AAAA ----Enter data to memory

This command is used to enter data to memory starting at address AAAA and continuing until a return command (/) is given. Data is entered in hexidecimal format.

Example:

```
ENTR 500
0 0A 30 44 FF FE/
```

DUMP AAAA BBBB ----Dump Contents of Memory

This command is used to examine the contents of memory. The values contained in memory from locations AAAA to BBBB are displayed in hexidecimal. Each line of display consists of an address followed by the contents of the next 16 memory locations. If BBBB is not specified only location AAAA will be displayed.

FILE /NAME/ AAAA

This command is used to enter, examine, or modify parameters of files created in the system. Up to six files can exist simultaneously with any one of the files called as "current". Depending on the form of the command and following parameters the following functions are performed.

FILE /NAME/ ADDR	Create a file with the name, NAME starting at address ADDR and make it current. If a file with the same name already exists output error message NO NO.
FILE /NAME/ 0	Delete file with name NAME and make no file current. Note no file can start at ADDR 0.
FILE /NAME/	Get file NAME and make it current. Save all parameters of existing current file.
FILE	Display parameters of the "current" file in the following format with AAAA and BBBB being the beginning of file and end of file addresses: NAME AAAA BBBB
FILES	Display the parameters of all files currently saved by the system.

EXEC AAAA Execute a program

This command is used to execute a program at address AAAA.

PROG AAAA BBBB Program PROM

This command is used to program a 1702A PROM. A programmer must be provided.

LIST N List file

This command is used to display the lines entered by the user into the file. The output consists of the lines in the file starting at line number N. If N is not specified the display starts at the beginning of the file. The user can terminate the display by using sense switch seven on the front panel.

DELT L1 L2 Delete line (s) from file

This command is used to delete lines entered by the user from the file. All lines starting at line L1 and continuing up to and including L2 are deleted from the file. If L2 is not specified only L1 is deleted.

PAGE AAAA BBBB Move page of data

This command is used to move one page (256 bytes) of data from address AAAA to BBBB.

CUST Optional user command at location E000

This command allows any routine to be placed at location E000 by the user. If the command is terminated by a RET and proper stack operations are used the system will return in an orderly manner.

ASSM (E) AAAA BBBB Assemble a source file to object code.

This command is used to assemble a source program written by the user and located in the file area. The assembler performs the assembly, assigning addresses to the object code starting at AAAA. On the second pass the object code is placed in memory starting at location BBBB. If BBBB is not specified it assumes the same value as AAAA. During pass one certain errors are displayed and during pass 2 a complete listing is produced. If the optional E is specified in the command only those lines which contain errors are listed.

************************* ****** *******

EDITOR

The editor is a line oriented editor which enables the user to easily create program files in the system. Each line is prefaced by a fixed line number which provides for stable line referencing. Since line numbers can range from 0000 to 9999 (Decimal) there are 10,000 lines that can exist in each file. (If enough storage exists.) As the user types lines on the input device they are entered into the file area. The editor places all line numbers in sequence and automatically overwrites an existing line in the file if a new line with the same line number is entered by the user. A feature of the editor is that the file area never contains any wasted space.

****The Editor ALWAYS operates on the current file.****

The editor does not automatically assign line numbers. The user must first, when entering a line of data, enter a decimal number which will be interpreted as being the line number. Valid line numbers must contain four digits...preceding zeros must be included. An entry to the editor is terminated by the carriage return key. No more than 80 characters may be input for one line.

All lines are ordered by the ascending numeric sequence of their line numbers. If the user wishes to insert lines after the initial entry is made it is suggested that he input the original lines with line numbers at least five units difference.

************************* ***************

ASSEMBLER

When the Assembler is given control by the executive it proceeds to translate the Symbolic 8080 Assembly Language (Source) program into 8080 machine (object) code. The Assembler is a two pass assembler which operates on the "current" file. Features of the Assembler include:

* free format source input
* symbolic addressing, including forward references and relative symbolic references.
* complex expressions may be used as arguments
* self defining constants
* multiple constant forms
* up to 256 five character symbols
* reserved names for 8080 registers
* ASCII character code generation
* 6 Pseudo Operations (assembler directives)

The assembler translates those lines contained in the current file into object code. The second character following the line number is considered to be the first source code character position. Hence the character immediately following the line number should normally be blank. Line numbers are not processed by the assembler they are merely reproduced on the listing

The assembler will assemble a source program file composed of STATEMENTS, COMMENTS, and PSEUDO OPERATIONS.

During Pass 1, the assembler allocates all storage necessary for the translated program and defines the values of all symbols used, by creating a symbol table. The storage allocated for the object code will begin at the first byte dictated by the 1st parameter in the original Executive ASSM command.

During pass 2, all expressions, symbols and ASCII constants are evaluated to absolute values and are placed in allocated memory in the appropriate locations. The listing, also produced during pass 2, indicates exactly what data is in each location of memory.

STATEMENTS may contain either symbolic 8080 machine instructions or pseudo-ops. The structure of such a statement is:

NAME   OPERATION   OPERAND   COMMENT

The name field, if present, must begin in assembler character position one. The symbol in the name field can contain as many characters as the user wants, however only the first 5 characters are used in the symbol table to uniquely define a symbol. All symbols in this field must begin with an alphabetic character and may contain no special characters.

The operation field, contains either a 8080 operation mnemonic or a system pseudo-operation code.

The operand field contains parameters pertaining to the operation in the operation field. If two arguments are present they must be separated by a comma. Example:

```
0015 FLOP  MOV M,B  COMMENT
0020 * COMMENT
0025       JMP  BEG
0030       CALL FLOP
0035 BEG   ADI  8+6-4
0040       MOV  A,B
```

All fields are separated and distinguished from one another by the presence of one or more blank characters. (Spaces)

The comment field is for explanatory remarks. It is reproduced on the listing without processing. See example 0015. Comment lines must start with an asterisk (*) in character position 1. See example 0020.

SYMBOLIC NAMES

To assign a symbolic name to a statement one merely places the symbol in the name field. To leave off the name field the user skips two or more spaces after the line number and begins the operation field. If a name is attached to a statement, the assembler assigns it the value of the current Location Counter. The Location Counter always holds the address of the next byte to be assembled. The only exception to this is the EQU pseudo-op. In this case a symbol in the name field is assigned a value which is contained in the operand field of the EQU pseudo-op statement. Example:

```
0057 POTTS EQU 128
```

assigns the value 128 to the name POTTS. This data can then be used elsewhere in the program as: eg ADI POTTS.

Names are defined when they appear in the name field. All defined names may be used as symbolic arguments in the argument filed. See examples 0015, 0025, 0030, 0035. field

In addition to user defined names, the assembler has reserved several symbols, the value of which is predetermined. These names may not be used by the user except in the operand field. They are (with their value in parenthesis):

A- the accumulator	(7)
B- Register B	(0)
C- Register C	(1)
D- Register D	(2)
E- Register E	(3)
H- Register H	(4)
L- Register L	(5)
M- Memory (through H,L)	(6)

In addition to the above reserved symbols, there is the single special character symbol ($). This symbol changes in value as the assembly progresses. It is always equated with the value of the program counter after the current instruction is assembled. It may only be used in the *operand* field. Examples:

```
JMP  $     means jump to the next location
MOV  A,B   after this instruction; i.e., the
           MOV instruction.
LDA  $+5
DB    0
DB    1    means load the data at the fifth location
DB    2    after this location. In this case the data has
DB    3    the value 5.
DB    4
DB    5
```

<u>RELATIVE SYMBOLIC ADDRESSING</u>

If the name of a particular location is known, a nearby location may be specified using the known name and a numeric offset. Example:

```
    JMP   BEG
    JPE   BEG+4
    CC    SUB
    CALL  $+48
BEG MOV   A,B
    HALT
    MVI   C, 'B'
    INR   B
```

In this example the instruction JMP BEG refers to the MOV A,B instruction. The instruction JPE BEG+4 refers to the INR B instruction. BEG+4 means the address BEG plus four bytes. This form of addressing can be used to locate several bytes before or after a named location.

CONSTANTS

The Assembler allows the user to write positive or negative numbers directly in a statement. They will be regarded as decimal constants and their binary equivalents will be used appropriately. All unsigned numbers are considered positive. Decimal constants can be defined using the descriptor "D" after the numeric value. (Not required, default is decimal)

Hexadecimal constants may be defined using the descriptor "H" after a numeric value. IE. +.0H, 10H, 3AH, 0F4H.

Note that a hexadecimal constant cannot start with the digits A-F. in this case a leading 0 must be included. This enables the assembler to differentiate between a numeric value and a symbol.

ASCII constants may be defined by enclosing the ASCII character within single quote marks, i.e., 'C'. For double word constants two characters may be defined within one quote string.

EXPRESSIONS

An expression is a sequence of one or more symbols, constants or other expressions separated by the arithmetic operators plus or minus.

```
PAM +3
ISAB-'A'+52
LOOP+32H-5
```

Expressions are calculated using 16 bit arithmetic. All arithmetic is done modulo 65536. Single byte data cannot contain a value greater than 255 or less than -256. Any value outside this range will result in an assembler error.

PSEUDO-OPERATIONS

The pseudo-operations are written as ordinary statements, but they direct the assembler to perform certain functions which do not always develop 8080 machine code. The following pages describe the pseudo-ops.

ORG--Set Program Origin:
label ORG expression Where the label is optional, but if present will be equated to the given expression.

END--End of Assembly; This pseudo-op informs the assembler that the last source statement has been read. The assembler will then start on pass 2 or terminate the assembly and pass control back to the executive. This pseudo-op is not needed when assembling from a memory file since the assembler will stop when an end of file indicator has been reached.

EQU--Equate Symbolic Value; The EQU is used to make two symbols equivalent in value; Label EQU expression

Where: Label- is a symbol the value of which will be determined from the expression.
Expression- is an expression which when evaluated will be assigned to the symbol given in the name field.

DS--Define Storage-- The DS causes the assembler to advance the Assembly Program Counter, effectively skipping past a given number of memory bytes.

label DS expression

DB--Define Byte; This pseudo-op is used to reserve one byte of storage. The content of the byte is specified in the argument field.

label DB expression

DW--Define Word; This pseudo-op is used to define two bytes of storage. The evaluated argument will be placed in the two bytes; high order 8 bits in the low order byte, and the low order 8 bits in the high order byte. This conforms to the Intel format for two byte addresses.

ASSEMBLER ERRORS

The following error flags are output on the assembler listing when the error occurs. Some of the errors are only output during pass 1.

O	Opcode Error
L	Label Error
D	Duplicate Label Error
M	Missing Label Error
V	Value Error
U	Undefined Symbol
S	Syntax Error
R	Register Error
A	Argument Error.

DEMONSTRATION PRINTOUT FROM SOFTWARE PACKAGE # 1
Page 1-3, Monitor, Editor and Assembler
Page 4-5, Interpretive Simulator demonstration

```
FILES                          COMMAND TO LIST ALL FILES
BLOP   2000   249B
TEST   4000   44A2
TARA   4500   4500

                                           CREATE FILE NAMED "SAMPL" STARTING AT ADDRESS 1000
FILE /SAMPL/ 1000
SAMPL 1000   1000  COMPUTER RESPONSE INDICATING NAME AND START & END ADDRESSES

FILES                                LIST ALL FILES
SAMPL  1000   1000
BLOP   2000   249B                    NEW FILE NOW SAVED AS SHOWN
TEST   4000   44A2
TARA   4500   4500


0000 * THIS IS A COMMENT LINE
0005 *
0009 * TEXT ENTERED TO SAMPL FILE
0015 * PROCESSOR TECHNOLOGY CORP.          LINES INPUT TO FILE BY LINE NUMBER
0020 * SOFTWARE PACKAGE #1
0025 *
0050 *
0050           MVI     A,'S'
0055 START     MOV     B,A
0060           OUT     1
0065           INR     A
0070           JMP     START
0075 * END OF SAMPL PROGRAM

FILE
SAMPL 1000   112[illegible]        FILE COMMAND LISTS FILE PARAMETERS

ASSM 0000          ASSEMBLE COMMAND

0000                          0000 * THIS IS A COMMENT LINE
0000                          0005 *
0000                          0009 * TEXT ENTERED TO SAMPL FILE
0000                          0015 * PROCESSOR TECHNOLOGY CORP.
0000                          0020 * SOFTWARE PACKAGE #1
0000                          0025 *
0000 3E 53                    0050           MVI     A,'S'
0002 47                       0055 START     MOV     B,A
0003 D3 01                    0060           OUT     1
0005 3C                       0065           INR     A
0006 C3 02 00                 0070           JMP     START
0009                          0075 * END OF SAMPL PROGRAM


FILE /BLOP/              CALL FILE BLOP AND MAKE IT CURRENT
BLOP   2000   249B        PARAMETERS LISTED
```

```
LIST         LIST COMMAND GIVEN

0000 *        <<  BLOP  >>
0001 *
0002 * HEX BOOTSTRAP LOADER
0003 *
0004 *    PROCESSOR TECHNOLOGY CORP.
0005 *    2465 FOURTH STREET
0006 *    BERKELEY, CA.
0007 *
0008 *
0010 *
0012 *
0015 BOOT   LXI D, ADDR
0020 NUM    MVI L,0     CLEAN HOUSE
0025 INSTAT IN  TTS     INPUT STATUS
0030        ANI IDR     IS DATA AVAILABLE?
0035        JZ  INSTAT
0037 *
0040        IN  TTI     GET CHARACTER
0045        OUT TTO     ECHO IT (OPTIONAL)
0050        SUI '0'     CONVERT TO BCD EQUIV
0051        JM   ONEWD
0052        CPI  10
0053        JC   DOIT
0054        ADI  0F9H  CONVERT A-F DOWN
0055 DOIT   DAD H
0060        DAD H
0065        DAD H
0070        DAD H
0075        ADD L
0080        MOV L,A
0085        JMP  INSTAT
0086 *
0087 * THIS ROUTINE ACCEPTS ONLY A SPACE
0088 * AS TERMINATOR.
0089 *
0090 ONEWD  CPI ' '-'0'  IS IT A SPACE?
0092        JNZ  NUM      IF NOT START OVER
0093 *
0095        MOV  A,L    GET CHR FROM L
0100        STAX D
0102        INX  D
0105        JMP  NUM
0110 *
0120 TTS EQU 0 *STATUS PORT
0125 TTI EQU 1 *TTY INPUT PORT
0130 TTO EQU 1 *TTO OUTPUT PORT
0135 IDR EQU 64 *BIT 6 FOR DAV TEST
0140 ADDR EQU 4096
```

```
FILE /SAMPL/0        DELETE FILE NAMED SAMPL

FILE /SAMPL/1000          CREATE NEW FILE NAMED SAMPL
SAMPL 1000 , 1000

0005  DW 'OS'
0010  DW 'TF'
0015  DW 'AW'
0020  DW 'ER'        INPUT PROGRAM BY LINE NUMBERS
0025  DW '# '
0030  DW '1_ 1'
0005 MESSG DW 'OS'     PUT IN NEW LINE 5
0050 *
0055  LXI H, MESSG
0055  LXI H, MESSG POINT TO MESSAGE
0060 START MOV A,M GET CHR TO REG A
0065 STAT  IN 0
0070  ANI 40H
0075  JZ A_ STAT       RUBOUT KEY GETS RID OF "A" ENTERED IN ERROR
0080  OUT 1
0085  INX H
0090  MOV A,H
0095  CPI 13
0100  JNZ START
0105  JMP 0F00CH GO BACK TO SYSTEM

0070  ANI 80H     PUT IN NEW LINE 70 TO CORRECT PROGRAM ERROR

LIST                 LIST FILE
0005 MESSG DW 'OS'
0010  DW 'TF'
0015  DW 'AW'
0020  DW 'ER'
0025  DW '# '
0030  DW ' 1'
0050 *
0055  LXI H, MESSG POINT TO MESSAGE
0060 START MOV A,M GET CHR TO REG A
0065 STAT  IN 0
0070  ANI 80H
0075  JZ  STAT
0080  OUT 1
0085  INX H
0090  MOV A,H
0095  CPI 13
0100  JNZ START
0105  JMP 0F00CH GO BACK TO SYSTEM

ASSME 0000  ASSEMBLE WITH NO LISTING TO SEE IF THERE ARE ASSEMBLY ERRORS

    NONE!!!!
```

```
0007                      0000 *   <<  SIMULATOR DEMONSTRATION  >>
0007                      0003 *
0007                      0005 *
0007                      0010 *       PROCESSOR TECHNOLOGY CORP.
0007                      0011 *       2465 FOURTH STREET
0007                      0012 *       BERKELEY, CALIF      947:0
0007 31 06 00             0015        LXI SP, 6
000A CD 1D 00             0020 START  CALL TTYIN   GET NUMBER
000D 47                   0025        MOV  B,A     STORE IT IN B
000E C6 0A                0030        ADI  10
0010 CD 1A 00             0035        CALL ADD
0013 80                   0040 FINI   ADD  B
0014 0D                   0045        DCR C
0015 D3 07                0050        OUT 7         OUTPUT TO PORT 7
0017 C3 0A 00             0055        JMP  START
001A 83                   0060 ADD    ADD  E
001B 2C                   0062        INR L
001C C9                   0065        RET
001D DB 01                0070 TTYIN  IN 1    GET CHR
001F D3 01                0075        OUT 1    ECHO IT
0021 C9                   0080        RET
0022                      0085 *
0022                      0090 SP EQU 6

SIMU ———— ENTER Simulator
*P 0000 ———— Set Program Counter
*P 0007 —— oops
*S A=0 B=0 C=0 D=0 E=125 L=0 —— Set Registers
*GO ———— START Simulation

AddR  FlAGS A   B  C  D  E   H  L  M   STACK
0007  01010 00  00 00 00 7D  0A 00 14  0006
000A  01010 00  00 00 00 7D  0A 00 14  0004 ← Call INST Uses STACK

001D  INPUT 01 =40H ← Simulated INPUT PoRT
001D  01010 40  00 00 00 7D  0A 00 14  0004
                         Simulated Output PoRT
001F  OUTPUT 01 =40 ←
001F  01010 40  00 00 00 7D  0A 00 14  0004 — RETURN
0021  01010 40  00 00 00 7D  0A 00 14  0006
000D  01010 40  40 00 00 7D  0A 00 14  0006 — Put A → B
000E  00000 4A  40 00 00 7D  0A 00 14  0006 — Add 10 — NoTE FlAGS
0010  00000 4A  40 00 00 7D  0A 00 14  0004 — Call INST
001A  01100 C7  40 00 00 7D  0A 00 14  0004 — Add E
001B  00000 C7  40 00 00 7D  0A 01 94  0004 — INCREMENT L
001C  00000 C7  40 00 00 7D  0A 01 94  0006 — RETURN
0013  10000 07  40 00 00 7D  0A 01 94  0006 — Add B
0014  11010 07  40 FF 00 7D  0A 01 94  0006 — DECREMENT C

0015  OUTPUT 07 =07 ← SIMULATED OUTPUT
0015  11010 07  40 FF 00 7D  0A 01 94  0006
0017  11010 07  40 FF 00 7D  0A 01 94  0006
000A  11010 07  40 FF 00 7D  0A 01 94  0004
*M 10 ———— Change Mode From Hex To decimal.
*S E=30 ———— Change Reg E
*GO
```

```
001D  INPUT 01 =50
001D  11010 050   064 255 000 030   010 001 148   J004

001F  OUTPUT 01 =32
001F  11010 050   064 255 000 030   010 001 148   0004
0021  11010 050   064 255 000 030   010 001 148   0006
000D  11010 050   050 255 000 030   010 001 148   0006
000E  00010 060   050 255 000 030   010 001 148   0006
0010  00010 060   050 255 000 030   010 001 148   0004
001A  00110 090   050 255 000 030   010 001 148   0004
001B  00000 090   050 255 000 030   010 002 009   0004
001C  00000 090   050 255 000 030   010 002 009   0006
0013  01000 140   050 255 000 030   010 002 009   0006
0014  01100 140   050 254 000 030   010 002 009   0006

0015  OUTPUT 07 =8C
0015  01100 140   050 254 000 030   010 002 009   0006
0017  01100 140   050 254 000 030   010 002 009   0006
*XIT
```

NOTE Mode
IS Now DECIMAL

SOFTWARE PACKAGE #1

ASSME 0000

EXEC 0000

```
:1CF000002100D00E4E0F7720DC206F0318ED0CD25F0237EFE3ADA76F4CD0BF1
:1CF01C00CDC3F0CDA6F0C30CF021ACD0225CD01E02CD8EF078FE18C23CF0CDA6
:1CF03800F0C325F0FE0DC251F07DFEACCA25F0360D233601233EFFCD86F021AB
:1CF05400D073C9FE7F026CF03EACBDCA2DF02B1D065FCD9BF0C32DF0FE20DA2D
:1CF07000F0FE5BD22D047CD9BF0773EFDBDCA64F0231CC32DF0BDC8362023C3
:1CF08C0086F0DB00E60CA8EF0DB01E67F47C9DB00E680CA9BF078D301C9060D
:1CF0A800CD9BF0060AD9BF0067FCD9BF0CD9BF0C9CD93F2CDA6F02A66D0E911
:1CF0C40057F2060A3E43271D0CDD4F0C21AF4E92A50D03A71D04FCDEBF01A67
:1CF0E000131A6FC8135C2D4F004C91ABEC2F7F023130DC2EBF0C9130DC2F7F0
:1CF0FC000CC9AF1166000600C1B1205C204F1C9CD12F1DA1AF4C92100002268D0
:1CF118002252D0CDFE021ABD0237EFE203FD0C221F12272D0CDF7F83FD0FE2F
:1CF13400C25DF111520000E05237EFE2FCA4DF10DFA1AF4120013C33CF13E200D
:1CF15000FA58F1121334FF1CDFEF83FD0113AD0CD58FB78FE053FD8015AD0CD
:1CF16C00B4F1D822660215AD0CD71F5CDF7F33FD0115ED0CD58FB78FE053FD8
:1CF18800015ED0CDB4F1D82268D0215ED0CD71F5B7C92100000AB7C8545D2929
:1CF1A4001929D630FE0A3FD85F16001903C39DF12100000AB7C829292929CDCB
:1CF1C000F1FE103FD8356F03C3B7F1D630FE0AD8D607C9CD1FF22150D046CD9B
:1CF1DC00F02346CD9BF0C9CDD3F1CDF6F1C0CD3CF2CDD6F12346CD9BF0C90620
:1CF1F800CD9BF0C92A66D03A69D0BCC20EF23A68D0BDC20EF237232266D0C946
:1CF214003E0DB8C8CD9BF023C313F22150D0471F1F1F1FCD32F2772378CD32F2
:1CF2300077C9E60FC630FE3AD8C607C92150D00664CD4DF2060ACD4DF2C63077
:1CF24C00C9362F3490D24FF28023C944554L50F29B45584543F0B9454E5452F4
:1CF268003746494C45F2FF4C495354F5844454C54F5E14153534DF658504147
:1CF2840045F2D8505E4F4DF59D43555254E0003A5AD0B7CA1AF4C9CD93F23E10
:1CF2A000326AD0CDA6F03A67D0CDD3F13A66D0CDE3F13A6AD0326BD02A66D07D
:1CF2BC00D3077CFEF77EC2C7F2DB06CDE3F1CDFCF1D83A6BD03DC2B5F2C3A3F2
.1CF2D800CD93F23A5BD0B7CA1AF42A66D0EB2A68D006007BD3077AFEFF1AC2F7
:1CF2F400F2DB06772C1305C2EBF2C9CDA6F03A52D0B7CA79F3CDD8F3EBC224F3
:1CF310003A5AD0B7CA1DF43A59D0B7C239F3212CF4C320F43A5AD0B7CA4BF32A
:1CF32C0066D07CB5CA4BF3231F4C320F42A57D0EB2152D0D502057E12130DC3
:1CF34800B5FCD12100D00E0D1A467778121320DC250F33A5AD0B7CA83F32A66
:1CF36400D02205D02207D07D34CA72F33601AF3209D0C383F33AB0D0FE530E06
:1CF38000CA85F30E012100D0793259D0E5110500197EB7C2A3F323862 3C2A3F3
:1CF39C0033332323C38EF3E10E0546CD9BF00D23C2A6F3CDC4F3CDC4F3CDA6F0
:1CF3B800110400193A59D03DC289F3C9CDF6F1237E2BE5CDD3F1E17E2323E5CD
:1CF3D400E3F1E1C9AF3259D0060611 00D0215 2D00E05CDEBF0F5D51AB7C20EF4
:1CF3F000131AB7C20EF4EB11FAFF192257D07A3259D0E1F111080019EB05C8C3
:1CF40C00E1F3E1F1C204F411FBFF197AB7C9CDA6F02126F4CD13F2C31FF05748
:1CF4280041543F0D46554C40D4E4F204E4F0DCD93F2CD44F4DA1AF4CDA6F0C9
:1CF44400CDA6F0CD25F021ACD0272D0CDFEF0CDF7F8DA44F4FE2FC8CD58FB78
:1CF46000FE033FD8015AD0CDB4F1D87D2A66D077CD0EF2C350F40E0421ABD023
:1CF47C007EFE30DA1AF4FE3AD21AF40DC27BF42250D0110CD0CD56F5D2B3F423
:1CF49800CD46F5210C00CD4EF511ABD02A07D00E01CD34F5712207D0C31FF0CD
:1CF4B4000BF50E02CA3CF40D462B3602224ED03AABD00DCAD1F490CAF4F4DAE4
:1CF4D000F42A07D0545DCD2FF52207D00E02CD3DF5C3F4F42F3C545DCD2FF5EB
:1CF4EC00CD34F536012207D02A4ED0360D2311ABD00E01CD34F5C31FF0215DD0
:1CF508002250D02A050CD28F5EB2A50D0EB3E04CD2FF5CD56F5D8C87ECD2FF5
:1CF52400C30EF5233E11BEC0C31FF0856FD024C91A13B9C87723C334F51A1BB9
:1CF54000C8772BC33D754623 E2356235EC9732B722B712B70C906010E04B71A
:1CF55C009ECA61F504B2B0D025BF505C90E041AD601C35CF5CD46F5AFB8C8BB
:1CF57800C44EF5C05A5148063 0C377F5CDA6F0CD05F523CD13F2CDA6F0CD27F5
:1CF59400DBFFE680C023C38BF5CD93F2CDA6F03A66D0CDE3F116032A66D07DD3
:1CF5B000077ED3083F02D309CDD5F5AFD309DB06BECACEF5063FCD9BF015C2AE
:1CF5CC00F5C9CDFCF1D8C3A0F51E96AF3DC2D8F51DC2D7F5C9CD93F2CD05F522
:1CF5E8004ED02161D0 EB7C21 5F5215DD02250D0EB210CD0CD56F52A4ED0DA43
:1CF60400F62207D036 1EB2 05D0EB060D2B7D937C9A3E0DDA3AF6052BBEC212
:1CF62000F62B7D937C ADA3 F6BE2323CA30F623CD46F5210CD0CD4EF5C9B8EB
:1CF63C00C22FF63209 0C9C 0EF5CC20F5EB2A4ED00E01CD34F52207D03601C9
```

```
:1CF65800CD93F23A5ED0B7C268F62A66D02268D03AB0D0FE45C271F6AF326AD0
:1CF67400AF3274D03270D02A66D0226ED02A05D0224ED02A4ED0318ED07EFE01
:1CF69000CAE8F3EB13218ED03EA7CD36F00E0DCD34F571EB224ED03A70D0B7C2
:1CF6AC00B4F6CDD7F6C387F6CD90F7218ED0CDC0F6C387F63A6AD0B7C2CDF63A
:1CF6C800A0D0FE20C8218ED0CDA6F0CD13F2C9CDFEF03270D021ACD02272D07E
:1CF6E400FE20CA19F7FE2AC8CD0AFBDAC9FACAAAFCCD30F7C2C9FA0E05215AD0
:1CF700007E1213230DC200F7EB226CD03A6FD077233A6ED0772174D034CDFEF0
:1CF71C00CDF7F8DAF0FACD58FBFE20DA4FFAC2F0FAC34FFA2A72D07EFE20C8FE
:1CF738003AC0 32272D0C9CDF7F81AB7CA5BF7FA8DF7E270F7FE05DA85F7C2E8
:1CF75400F80E 2AFC3DFFACD7AFB3AA0D0FE20C0226ED03AACD0FE20C8C37BF7
:1CF77000CD7A B3AACD0FE20CA82FCEB2A6CD0722373C3C7F6CD7AFB444DC3E7
:1CF78C00F7C3 EFA218ED03A6FD0CD22F2233A6ED0CD22F2227AD0CDFEF021AC
:1CF7A800D022 2D07FFE20CA19F7FE2AC8CD0AFBDAA5FCCD30F7C2A5FCC319F7
:1CF7C4001AB7 AF4F7FAEBF7E0FE05DADBF7C2E8F8CDC7F8C355F7CD77FB444D
:1CF7E0002A68D0092268D0AFC3E2FACD86F8AF0E01C3DFFACD77FB3AA0D0FE20
:1CF7FC00C0EB2A6ED0EB226ED07D935F7C9A572A68D0192268D0C9CDD4F8C9CD
:1CF8180077FB 464FC7DB7CA3BF8FE02C464FCC33BF8CD77FBC464FC7D0FDC64
:1CF83400FC17 E08D464FC071717471A80FE76CC64FCC313F8CD77FBC464FC7D
:1CF85000FE08D464FC1AFE40CA67F8FEC77DCA3BF8FA3EF8C33BF82929298512
:1CF86C00CDA5 8CD7AFBC464FC7DFE08D464FCC33EF8FE06CC93F8CDD4F8CD77
:1CF88800FB3C E02D47DFC7DC313F8CD77FBC464FC7DFE08D464FC2929291A85
:1CF8A4005F2A 2D07EFE2C232272D0C26DFC7BC9FE01C2C4F8CD93F8E608C464
:1CF8C000FC7E 6F7CDD4F8CD77FB7D54CDD4F87AC313F8C92A68D077232268D0
:1CF8DC002A7AD02323CD22F2227AD0C93A70D0B7C21FF0CDA6F03E01C378F62A
:1CF8F80072D07EFE20C023227 2D0C3FAF8215BD02250D00602CD3AFAC94F5247
:1CF914000000 551550001444200 00FF445300000344570000 05454E44000600
:1CF93000484C5476524C43075252430F52414C175241521F524554C9434D412F
:1CF94C005354 337444141 27434D433F454900FB444900F34E4F50000584348
:1CF96800 47EF 385448 4CE35350484CF95043484CDF00535441 58024C4441580A
:1CF984000054 555348 5504F5000C1494E58000344435 8000B444144000 90049
:1CF9A0004E5 20444435 2054D4F564041444480414443885355429053424 29841
:1CF9BC004E4 A05852 1A84F5241B0434D50B85253 54C700414449C6414349CE
:1CF9D8005355 42D6534249DE414E49E6585249EE4F5249F6435049FE494E00DB
:1CF9F4004F55 54D34D 5649060 04A4D5000C343414C4CCD4C58490 0014C444100
:1CFA10003A53 5441003253484C44224C484C442A004E5A005A00084E43104300
:1CFA2C001850 4F205045285000304D0038002A50D01AB7CA4CFA48CDEBF01AC8
:1CFA480013C3 AFA3C13C9215AD02250D01111F90604CD3AFACAF8FA05CD3AFA
:1CFA6400CA6E A04CD3AFA2113F80E01CACBFACD3AFA2117F8CA6EFACD3AFA21
:1CFA80002AF8 A5EFA05CD3AFA2149F8CA6EFACD3AFA217EF80E02CACBFA04CD
:1CFA9C003AFA AC6FACD05F9C2F0FAC6C05706033A5AD04FFE527ACA6BFA7914
:1CFAB80014FE ACAC57AFE43C2F0FA14147A21B4F80E033279D03E5A805F3ED0
:1CFAD400CE00 71AB7 C2F0FA3A70D00600EB2A6ED009226ED0B7C83A79D0EBE9
:1CFAF0002190 C0E033 3DCFA215ED07EB7C2F0FA3A70D0B7CA3FF7C3C4F7FE41
:1CFB0C00D8FE B3FD8CD58FB215AD02250D005C22EFB041147FBCD3AFAC22EFB
:1CFB28006F26 0C341FB3A74D04711FFD0B7CA44FB3E053271D0CDD4F0373FC9
:1CFB44003CB7 94107420043014402450348044C054D06000600120478FE0BD0
:1CFB60001323 272D07EFE30D8FE3ADA5AFBFE41D8FE5BDA5AFBC9CDF7F82100
:1CFB7C000022 6D024 2277D02A72D02BCDFEF03275D0237EFE21DA36FCFE2CCA
:1CFB980036FC E2BCA 7FBFE2DC2B7FB3275D03A78D0FE02CA6DFC3E023278D0
:1CFBB400C38E B4F3A78D0B7CA6DFC79FE24C2CFFB232272D02A6ED0C30BFCFE
:1CFBD00027C2 BFB1100000E03232272D07EFE0DCA8BFCFE27C2F2FB232272D0
:1CFBEC007EFE 7C20CFC0DCA8BFC535FC3D9FBFE30DA8BFCFE3AD22AFCCD46FC
:1CFC0800DA8B CEB2A 6D0AF3278D03A75D0B7C221FC192276D0C384FB7D936F
:1CFC24007C9A 7C31BFCCD0AFBCA0BFCDA8BFCC378FC3A78D0B7C26DFC2A76D0
:1CFC40007C11 9D0B7C9CD58FB1B1A015AD0FE48CA5EFCFE44C25AFCAF12CD9A
:1CFC5C00F1C9 F12CDB4F1C93E5221000032A0D0C93E5332A0D0210000C340FC
:1CFC78003E55 36FFC3 56C366FC3E4D32A0D0CDCDF6C93E41C36FFC3E4F32A0
:1CFC9400D03A 0D0B7C30E03AFCDD4F80DC29CFCC93E4CC392FC3E4432A0D0CD
:0CFCB000CDF6 319F723C243F3C34AF3
:00
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                 MICROTEC MACRO ASSEMBLER VER 3.1           PAGE    1

                                    *
                                    *             ***  SELF CONTAINED SYSTEM  ***
                                    *       PROCESSOR TECHNOLOGY SOFTWARE PACKAGE #1
                                    *
                                    *
                                    *
                                    * THIS ROUTINE INITIALIZES THE FILE AREA FOR SUBSEQUENT PROCESSING
                                            ORG        H'F000'
      F000  21 00 D0                INITA   LXI        H,FILE0
      F003  0E 4E                           MVI        C,MAXFIL*FELEN
      F005  AF                              XRA        A
      F006  77                      INIT2   MOV        M,A
      F007  23                              INX        H
      F008  0D                              DCR        C
      F009  C2 06 F0                        JNZ        INIT2
                                    *
                                    * THIS IS THE STARTING POINT OF THE SELF CONTAINED SYSTEM ONCE
                                    * THE SYSTEM HAS BEEN INITIALIZED.  COMMANDS ARE READ FROM THE USER
                                    * EXECUTED AND CONTROL RETURNS BACK TO THIS POINT TO READ ANOTHER
                                    * USER COMMAND.
                                    *
      F00C  31 8E D0                SYS8    LXI        SP,AREA+18
      F00F  CD 25 F0                        CALL       READ                READ INPUT LINE
      F012  23                              INX        H
      F013  7E                              MOV        A,M                 FETCH FIRST CHARACTER
      F014  FE 3A                           CPI        A'9'+1              COMMAND OR LINE NUMBER
      F016  DA 76 F4                        JC         LINE                JUMP IF LINE FOR FILE
      F019  CD 08 F1                        CALL       VALC                GET COMMAND VALUES
      F01C  CD C3 F0                        CALL       COMM                CHECK LEGAL COMMANDS
      F01F  CD A6 F0                EOR     CALL       CRLF                GET HERE WHEN ROUTINE IS DONE
      F022  C3 0C F0                        JMP        SYS8
                                    *
                                    *
                                    *
                                    * THIS ROUTINE READS IN A LINE FROM THE TTY AND PLACES IT IN AN
                                    * INPUT BUFFER
                                    * THE FOLLOWING ARE SPECIAL CHARACTERS
                                    *    CR     -  TERMINATES READ ROUTINE
                                    *    LF     -  NOT RECOGNIZED BY ROUTINE
                                    *    CONTROL X DELETE CURRENT LINE
                                    *    DEL    -  DELETE CHARACTER
                                    * ALL DISPLAYABLE CHARACTERS BETWEEN BLANK-Z AND THE ABOVE
                                    * ARE RECOGNIZED BY THE READ ROUTINE ALL OTHER ARE SKIPPED
                                    * OVER.  THE ROUTINE WILL NOT ACCEPT MORE CHARACTERS THAN THE INPUT
                                    * BUFFER WILL HOLD
                                    *
      F025  21 AC D0                READ    LXI        H,IBUF              SET INPUT BUFFER ADDRESS
      F028  22 50 D0                        SHLD       ADDS                SAVE ADDRESS
      F02B  1E 02                           MVI        E,2                 INITIALIZE CHARACTER COUNT
      F02D  CD 8E F0                NEXT    CALL       IN8                 READ A LINE
      F030  78                              MOV        A,B
      F031  FE 18                           CPI        24                  CHECK FOR CONTROL X
      F033  C2 3C F0                        JNZ        CR
      F036  CD A6 F0                        CALL       CRLF                OUTPUT A CRLF
```

INITA LXI H, FILEØ FILEØ = DØØØ
MVI C, MAXFIL*FELEN DØØ4
XRA A (clear Acc) = 13D × 6D = 78D
INITB MOV M,A
INX H
DCR C
JNZ INITB

Read LXI H, IBUF (HL = DØAC)
SHLD ADDS (DØ5Ø)
MVI E, 2
Next CALL IN8
MOV A,B
CPI 24D (18H)
JNZ CR
CALL CRLF
JMP READ

```
LINE  ADDR  B1 B2 B3 B4 ERROR                 MICROTEC MACRO ASSEMBLER VER 3.1           PAGE    2

      F039  C3 25 F0                        JMP        READ
      F03C  FE 0D                   CR      CPI        ASCR                GET A ASCII CR
      F03E  C2 57 F0                        JNZ        DEL
      F041  7D                              MOV        A,L
      F042  FE AC                           CPI        >IBUF               CHECK FOR FIRST CHARACTER
      F044  CA 25 F0                        JZ         READ
      F047  36 0D                           MVI        M,ASCR              PLACE CR AT END OF LINE
      F049  23                              INX        H
      F04A  36 01                           MVI        M,1                 PLACE EOF INDICATOR IN LINE
      F04C  23                              INX        H
      F04D  3E FF                           MVI        A,>IBUF+83
      F04F  CD 86 F0                        CALL       CLER                CLEAR REMAINING BUFFER
      F052  21 AB D0                        LXI        H,IBUF-1
      F055  73                              MOV        M,E                 SAVE CHARACTER COUNT
      F056  C9                              RET
      F057  FE 7F                   DEL     CPI        127                 CHECK FOR DELETE CHARACTER
      F059  C2 6C F0                        JNZ        CHAR
      F05C  3E AC                           MVI        A,>IBUF
      F05E  BD                              CMP        L                   IS THIS 1ST CHARACTER
      F05F  CA 2D F0                        JZ         NEXT
      F062  2B                              DCX        H                   DECREMENT POINTER
      F063  1D                              DCR        E                   DECREMENT COUNT
      F064  06 5F                   BSPA    MVI        B,'5F'
      F066  CD 9E F0                        CALL       OUT8
      F069  C3 2D F0                        JMP        NEXT
      F06C  FE 20                   CHAR    CPI        A' '                CHECK FOR LEGAL CHARACTERS
      F06E  DA 2D F0                        JC         NEXT
      F071  FE 5B                           CPI        A'Z'+1
      F073  D2 2D F0                        JNC        NEXT
      F076  47                              MOV        B,A
      F077  CD 9E F0                        CALL       OUT8                ECHO CHARACTER
      F07A  77                              MOV        M,A
      F07B  3E FD                           MVI        A,>IBUF+81
      F07D  BD                              CMP        L                   CHECK FOR END OF LINE
      F07E  CA 64 F0                        JZ         BSPA
      F081  23                              INX        H
      F082  1C                              INR        E                   INCREMENT CHARACTER COUNT
      F083  C3 2D F0                        JMP        NEXT
                                    *
                                    *
                                    *
                                    * THIS ROUTINE IS USED TO BLANK OUT A PORTION OF MEMORY
                                    *
      F086  BD                      CLER    CMP        L
      F087  C8                              RZ
      F088  36 20                           MVI        M,A' '              PLACE BLANK IN MEMORY
      F08A  23                              INX        H
      F08B  C3 86 F0                        JMP        CLER
                                    *
                                    *
                                    * THIS SUBROUTINE IS USED TO READ A BYTE OF DATA FROM THE UART
                                    *
      F08E  DB 00                   IN8     IN         USTA                READ UART STATUS
      F090  E6 40                           ANI        H'40' -02
```

CR CPI ASCR (0D)
JNZ DEL
MOV A,L
CPI >IBUF (AC)
JZ READ
MVI M,ASCR
INX H
MVI M,1 (EOF)
INX H
MVI A,>IBUF+83
CALL CLER
LXI H,IBUF-1
MOV M,E
RET

Rec'd B= input char

```
LINE  ADDR  B1 B2 B3 B4 ERROR                  MICROTEC MACR  ASSEMBLER VER 3.1              PAGE    3

 109  F092  CA 8E F0                        JZ        IN8
 110  F095  DB 01                           IN        UDAI                 READ DATA
 111  F097  E6 7F                           ANI       127                  STRIP OFF PARITY
 112  F099  47                              MOV       B,A
 113  F09A  C9                              RET
 114                                *
 115                                * THIS ROUTINE OUTPUTS A B TE OF DAT  TO THE UART
 116                                *
 117  F098  DB 00                   OUT8    IN        USTA                 READ STATUS
 118  F09D  E6 80                           ANI       H'80'
 119  F09F  CA 98 F0                        JZ        OUT8
 120  F0A2  78                      OK      MOV       A,B
 121  F0A3  D3 01                           OUT       UDAO                 TRANSMIT DATA
 122  F0A5  C9                              RET
 123                                *
 124                                * THIS ROUTINE WILL OUTPUT   CARRIAGE RETURN AND LINE FEED
 125                                * FOLLOWED BY TWO DELETE C ARACTERS W ICH PROVIDE TIME FOR
 126                                * A PRINT HEAD TO RETURN
 127                                *
 128  F0A6  06 0D                   CRLF    MVI       B,13                 CR
 129  F0A8  CD 98 F0                        CALL      OUT8
 130  F0AB  06 0A                   LF      MVI       B,10                 LF
 131  F0AD  CD 98 F0                        CALL      OUT8
 132  F0B0  06 7F                           MVI       B,127
 133  F0B2  CD 98 F0                        CALL      OUT8
 134  F0B5  CD 98 F0                        CALL      OUT8
 135  F0B8  C9                              RET
 136                                *
 137                                *
 138                                * THIS ROUTINE JUMPS TO A  OCATION IN MEMO Y GIVEN BY THE
 139                                * INPUT COMMAND AND BEGINS EXECUTION   PROGRAM THERE.
 140                                *
 141  F0B9  CD 93 F2                EXEC    CALL      VCHK                 CHECK FOR PARAMETER
 142  F0BC  CD A6 F0                        CALL      CRLF
 143  F0BF  2A 66 D0                        LHLD      BBUF                 FETCH ADDRESS
 144  F0C2  E9                              PCHL                           JUMP TO PROGRAM
 145                                *
 146                                *
 147                                *
 148                                * THIS ROUTINE CHECK THE I PUT COMMAND AG INST ALL LEGAL COMMANDS
 149                                * STORED IN A TABLE.  IF A LEGAL COMMA D IS FOUND A JUMP IS
 150                                * MADE TO THAT ROUTINE.  OTHERWISE AN ERROR MESSAGE IS OUTPUT
 151                                * TO THE USER
 152                                *
 153  F0C3  11 57 F2                COMM    LXI       D,CTA                COMMAND TABLE ADDRESS
 154  F0C6  06 0A                           MVI       B,NCO                NUMBER OF COMMANDS
 155  F0C8  3E 04                           MVI       A,4                  LENGTH OF COMMAND
 156  F0CA  32 71 D0                        STA       NCHR                 SAVE
 157  F0CD  CD D4 F0                        CALL      COMS                 SEARCH TABLE
 158  F0D0  C2 1A F4                        JNZ       WHAT                 JUMP IF ILLEGAL COMMAND
 159  F0D3  E9                              PCHL                           JUMP TO ROUTINE
 160                                *
 161                                *
 162                                *
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                  MICROTEC MACRO ASSEMBLER VER 3.1              PAGE    4

 163                                *
 164                                *
 165                                * THIS ROUTINE CHECKS IF A BASE CHARACTER STRING IS EQUAL TO
 166                                * ANY OF THE STRINGS CONTA NED IN A TA LE P INTED TO BY
 167                                * D,E.  THE LENGTH OF THE  TRINGS ARE  256.  ON RETURN IF THE
 168                                * ZERO FLAG IS SET A MATCH WAS FOUND.  F THE ZERO FLAG IS CLEAR (0)
 169                                * NO MATCH WAS FOUND.  REG STER B CONTAINS THE NUMBER OF
 170                                * STRINGS TO COMPARE
 171                                * THE TABLE CONSISTS OF AN  NUMBER OF CHAR. WITH 2 BYTES CONTAINING
 172                                * VALUES ASSOCIATED WITH I .  IT CAN BE USED TO SEARCH THROUGH
 173                                * A COMMAND TABLE OR SYMBOL TABLE.
 174                                * ON RETURN D,E POINT TO T E LAST BYTE ASSOCIATED WITH THE
 175                                * CHARACTER STRING IF A MA CH WAS FOUN .  IF NO MATCH WAS
 176                                * FOUND D,E POINT TO THE N XT LOCATION AFTER THE END OF THE TABLE.
 177                                *
 178  F0D4  2A 50 D0                COMS    LHLD      ADDS                 FETCH COMPARE ADDRESS
 179  F0D7  3A 71 D0                        LDA       NCHR                 GET LENGTH OF STRING
 180  F0DA  4F                              MOV       C,A
 181  F0DB  CD EB F0                        CALL      SEAR                 COMPARE STRINGS
 182  F0DE  1A                              LDAX      D                    FETCH VALUE
 183  F0DF  67                              MOV       H,A
 184  F0E0  13                              INX       D
 185  F0E1  1A                              LDAX      D                    FETCH VALUE
 186  F0E2  6F                              MOV       L,A
 187  F0E3  C8                              RZ
 188  F0E4  13                              INX       D                    SET TO NEXT STRING
 189  F0E5  05                              DCR       B                    DECREMENT COUNT
 190  F0E6  C2 D4 F0                        JNZ       COMS
 191  F0E9  04                              INR       B                    CLEAR ZERO FLAG
 192  F0EA  C9                              RET
 193                                *
 194                                *
 195                                *
 196                                * THIS ROUTINE CHECKS IF T   CHARACTER STRINGS CONTAINED IN MEMORY
 197                                * ARE EQUAL.  THE STRINGS A E POINTED    BY H,L AND D,E.
 198                                * ON RETURN THE ZERO FLAG S T INDICATES A MATCH.  REGISTER C
 199                                * INDICATES THE LENGTH OF T E STRINGS.   N RETURN THE POINTERS
 200                                * POINT TO THE NEXT ADDRES  AFTER THE CH RACTER STRINGS
 201                                *
 202  F0EB  1A                      SEAR    LDAX      D                    FETCH CHARACTER
 203  F0EC  BE                              CMP       M                    COMPARE CHARACTERS
 204  F0ED  C2 F7 F0                        JNZ       INCA
 205  F0F0  23                              INX       H
 206  F0F1  13                              INX       D
 207  F0F2  0D                              DCR       C                    DECREMENT CHARACTER COUNT
 208  F0F3  C2 EB F0                        JNZ       SEAR
 209  F0F6  C9                              RET
 210  F0F7  13                      INCA    INX       D
 211  F0F8  0D                              DCR       C
 212  F0F9  C2 F7 F0                        JNZ       INCA
 213  F0FC  0C                              INR       C                    CLEAR ZERO FLAG
 214  F0FD  C9                              RET
 215                                *
 216                                *
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                 MICROTEC MACRO ASSEMBLER VER 3.1             PAGE    5

 217                                  * THIS ROUTINE ZEROS OUT A BUFFER IN MEMORY WHICH IS THEN
 218                                  * USED BY OTHER SCANNING ROUTINES
 219                                  *
 220  F0FE  AF                        ZBUF      XRA       A                   GET A ZERO
 221  F0FF  11 66 D0                            LXI       D,ABUF+12           BUFFER ADDRESS
 222  F102  06 0C                               MVI       B,12                BUFFER LENGTH
 223  F104  1B                        ZBU1      DCX       D                   DECREMENT ADDRESS
 224  F105  12                                  STAX      D                   ZERO BUFFER
 225  F106  05                                  DCR       B
 226  F107  C2 04 F1                            JNZ       ZBU1
 227  F10A  C9                                  RET
 228                                  *
 229                                  * THIS ROUTINE CALL ETRA TO OBTAIN THE INPUT PARAMETER VALUES
 230                                  * AND CALLS AN ERROR ROUTINE IF AN ERROR OCCURRED IN THAT ROUTINE
 231                                  *
 232  F10B  CD 12 F1                  VALC      CALL      ETRA                GET INPUT PARAMETERS
 233  F10E  DA 1A F4                            JC        WHAT                JUMP IF ERROR
 234  F111  C9                                  RET
 235                                  *
 236                                  *
 237                                  * THIS ROUTINE EXTRACTS THE VALUES ASSOCIATED WITH A COMMAND
 238                                  * FROM THE INPUT STREAM AND PLACES THEM IN THE ASCII BUFFER (ABUF).
 239                                  * IT ALSO CALLS A ROUTINE TO CONVERT THE ASCII HEXADECIMAL TO BINARY
 240                                  * AND STORES THEM IN THE BINARY BUFFER (BBUF)
 241                                  * ON RETURN CARRY SET INDICATES AN ERROR IN INPUT PARAMETERS
 242                                  *
 243  F112  21 00 00                  ETRA      LXI       H,0                 GET A ZERO
 244  F115  22 68 D0                            SHLD      BBUF+2              ZERO VALUE
 245  F118  22 52 D0                            SHLD      FBUF                SET NO FILE NAME
 246  F11B  CD FE F0                            CALL      ZBUF                ZERO BUFFER
 247  F11E  21 AB D0                            LXI       H,IBUF-1            INPUT BUFFER ADDRESS
 248  F121  23                        VAL1      INX       H
 249  F122  7E                                  MOV       A,M                 FETCH INPUT CHARACTER
 250  F123  FE 20                               CPI       A' '                LOOK FOR FIRST BLANK
 251  F125  3F                                  CMC
 252  F126  D0                                  RNC                           RETURN IF CARRY
 253  F127  C2 21 F1                            JNZ       VAL1                JUMP IF NO BLANK
 254  F12A  22 72 D0                            SHLD      PNTR                SAVE POINTER
 255  F12D  CD F7 F8                            CALL      SBLK                SCAN TO FIRST PARAMETER
 256  F130  3F                                  CMC
 257  F131  D0                                  RNC                           RETURN IF CR
 258  F132  FE 2F                               CPI       A'/'
 259  F134  C2 5D F1                            JNZ       VAL5                NO FILE NAME
 260  F137  11 52 D0                            LXI       D,FBUF              NAME FOLLOWS PUT IN FBUF
 261  F13A  0E 05                               MVI       C,NMLEN
 262  F13C  23                        VAL2      INX       H
 263  F13D  7E                                  MOV       A,M
 264  F13E  FE 2F                               CPI       A'/'
 265  F140  CA 4D F1                            JZ        VAL3
 266  F143  0D                                  DCR       C
 267  F144  FA 1A F4                            JM        WHAT
 268  F147  12                                  STAX      D                   STORE FILE NAME
 269  F148  00                                  NOP
 270  F149  13                                  INX       D


LINE  ADDR  B1 B2 B3 B4 ERROR                 MICROTEC MACRO ASSEMBLER VER 3.1             PAGE    6

 271  F14A  C3 3C F1                            JMP       VAL2
 272  F14D  3E 20                     VAL3      MVI       A,A' '              GET AN ASCII SPACE
 273  F14F  0D                        VAL4      DCR       C
 274  F150  FA 5B F1                            JM        DONE
 275  F153  12                                  STAX      D                   FILL IN WITH SPACES
 276  F154  13                                  INX       D
 277  F155  C3 4F F1                            JMP       VAL4
 278  F158  CD FE F8                  DONE      CALL      SBL2
 279  F15B  3F                                  CMC
 280  F15C  D0                                  RNC
 281  F15D  11 5A D0                  VAL5      LXI       D,ABUF
 282  F160  CD 5B FB                            CALL      ALPS                PLACE PARAMETER IN BUFFER
 283  F163  78                                  MOV       A,B                 GET DIGIT COUNT
 284  F164  FE 05                               CPI       5                   CHECK NUMBER OF DIGITS
 285  F166  3F                                  CMC
 286  F167  D8                                  RC                            RETURN IF TOO MANY DIGITS
 287  F168  01 5A D0                            LXI       B,ABUF
 288  F16B  CD B4 F1                            CALL      AHEX                CONVERT VALUE
 289  F16E  D8                                  RC                            ILLEGAL CHARACTER
 290  F16F  22 66 D0                            SHLD      BBUF                SAVE IN BINARY BUFFER
 291  F172  21 5A D0                            LXI       H,ABUF
 292  F175  CD 71 F5                            CALL      NORM                NORMALIZE ASCII VALUE
 293  F178  CD F7 F8                            CALL      SBLK                SCAN TO NEXT PARAMETER
 294  F17B  3F                                  CMC
 295  F17C  D0                                  RNC                           RETURN IF CR
 296  F17D  11 5E D0                            LXI       D,ABUF+4
 297  F180  CD 5B FB                            CALL      ALPS                PLACE PARAMETER IN BUFFER
 298  F183  78                                  MOV       A,B                 GET DIGIT COUNT
 299  F184  FE 05                               CPI       5                   CHECK NUMBER OF DIGITS
 300  F186  3F                                  CMC
 301  F187  D8                                  RC                            RETURN IF TOO MANY DIGITS
 302  F188  01 5E D0                            LXI       B,ABUF+4
 303  F18B  CD B4 F1                            CALL      AHEX                CONVERT VALUE
 304  F18E  D8                                  RC                            ILLEGAL CHARACTER
 305  F18F  22 68 D0                            SHLD      BBUF+2              SAVE IN BINARY BUFFER
 306  F192  21 5E D0                            LXI       H,ABUF+4
 307  F195  CD 71 F5                            CALL      NORM                NORMALIZE ASCII VALUE
 308  F198  B7                                  ORA       A                   CLEAR CARRY
 309  F199  C9                                  RET
 310                                  *
 311                                  *
 312                                  * THIS ROUTINE FETCHES DIGITS FROM THE BUFFER ADDRESSED BY
 313                                  * REGISTER B,C AND CONVERTS THE ASCII DECIMAL DIGITS INTO
 314                                  * BINARY.  UP TO A 16 BIT VALUE CAN BE CONVERTED.  THE SCAN
 315                                  * STOPS WHEN A BINARY ZERO IS FOUND IN THE BUFFER
 316                                  *
 317  F19A  21 00 00                  ADEC      LXI       H,0                 GET A 16 BIT ZERO
 318  F19D  0A                        ADE1      LDAX      B                   FETCH ASCII DIGIT
 319  F19E  B7                                  ORA       A                   SET ZERO FLAG
 320  F19F  C8                                  RZ                            RETURN IF FINISHED
 321  F1A0  54                                  MOV       D,H                 SAVE CURRENT VALUE
 322  F1A1  5D                                  MOV       E,L                 SAVE CURRENT VALUE
 323  F1A2  29                                  DAD       H                   TIMES TWO
 324  F1A3  29                                  DAD       H                   TIMES TWO
```

```
LINE  ADDR  B  B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1              PAGE    7

      F1A4  19              19       DAD     D              ADD IN ORIGINAL VALUE
      F1A5  29              29       DAD     H              TIMES TWO
      F1A6  D6 30           D630     SUI     48             ASCII BIAS
      F1A8  FE 0A           FE 0A    CPI     10             CHECK FOR LEGAL VALUE
      F1AA  3F              3F       CMC
      F1AB  D8                       RC                     RETURN IF ERROR
      F1AC  5F                       MOV     E,A
      F1AD  16 00           1D8      MVI     D,0
      F1AF  19              .5F      DAD     D              ADD IN NEXT DIGIT
      F1B0  03                       INX     B              INCREMENT POINTER
      F1B1  C3 9D F1        1600     JMP     ADE1
                            19     *
                            03     *
                         C39DF1    * THIS ROUTINE FETCHES DIGITS FROM THE BUFFER ADDRESSED BY
                                   * REGISTERS B,C AND CONVERTS THE ASCII HEXADECIMAL DIGITS INTO
                                   * BINARY.  UP TO A 16 BIT VALUE CAN BE CONVERTED.  THE SCAN STOPS
                                   * WHEN A BINARY ZERO IS FOUND IN THE BUFFER
                                   *
      F1B4  21 00 00        21     AHEX     LXI     H,0            GET A 16 BIT ZERO
      F1B7  0A           0A        AHE1     LDAX    B              FETCH ASCII DIGIT
      F1B8  B7           C8                 ORA     A              SET ZERO FLAG
      F1B9  C8           29                 RZ                     RETURN IF ZERO
      F1BA  29           29                 DAD     H              LEFT SHIFT
      F1BB  29           29                 DAD     H              LEFT SHIFT
      F1BC  29           29                 DAD     H              LEFT SHIFT
      F1BD  29                              DAD     H              LEFT SHIFT
      F1BE  CD C8 F1                        CALL    AHS1           CONVERT TO BINARY
      F1C1  FE 10                           CPI     H'10'          CHECK FOR LEGAL VALUE
      F1C3  3F                              CMC
      F1C4  D8                              RC                     RETURN IF ERROR
      F1C5  85                              ADD     L
      F1C6  6F                              MOV     L,A
      F1C7  03                              INX     B              INCREMENT POINTER
      F1C8  C3 B7 F1                        JMP     AHE1
                                   *
                                   * THIS SUBROUTINE CONVERTS ASCII HEX DIGITS INTO BINARY
                                   *
      F1CB  D6 30                  AHS1     SUI     48             ASCII BIAS
      F1CD  FE 0A                           CPI     10             DIGIT 0-10
      F1CF  D8                              RC
      F1D0  D6 07                           SUI     7              ALPHA BIAS
      F1D2  C9                              RET
                                   *
                                   *
                                   * THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII HEXADECIMAL
                                   * AND OUTPUTS THE CHARACTERS TO THE TTY
                                   *
      F1D3  CD 1F F2               HOUT     CALL    BINH           CONVERT VALUE
      F1D6  21 50 D0                        LXI     H,HCON         CONVERSION AREA
      F1D9  46                     CHUT     MOV     B,M            FETCH OUTPUT CHARACTER
      F1DA  CD 98 F0                        CALL    OUT8           OUTPUT CHARACTER
      F1DD  23                              INX     H
      F1DE  46                              MOV     B,M            FETCH CHARACTER
      F1DF  CD 98 F0                        CALL    OUT8           OUTPUT CHARACTER
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1              PAGE    8

      F1E2  C9                              RET
                                   *
                                   * THIS ROUTINE DOES THE SAME AS ABOVE BUT OUTPUTS A BLANK
                                   * AFTER THE LAST CHARACTER
                                   *
      F1E3  CD D3 F1               HOTB     CALL    HOUT           CONVERT AND OUTPUT
      F1E6  CD F6 F1                        CALL    BLK1           OUTPUT A BLANK
      F1E9  C9                              RET
                                   *
                                   * THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII DECIMAL
                                   * DIGITS AND OUTPUTS THE CHARACTERS TO THE TTY
                                   *
      F1EA  CD 3C F2               DOUT     CALL    BIND           CONVERT VALUE
      F1ED  CD D6 F1                        CALL    HOUT+3         OUTPUT VALUE (2 DIGITS)
      F1F0  23                              INX     H
      F1F1  46                              MOV     B,M            GET LAST DIGIT
      F1F2  CD 98 F0                        CALL    OUT8           OUTPUT
      F1F5  C9                              RET
                                   *
                                   * THIS ROUTINE OUTPUTS A BLANK
                                   *
      F1F6  06 20                  BLK1     MVI     B,A' '         GET A BLANK
      F1F8  CD 98 F0                        CALL    OUT8
      F1FB  C9                              RET
                                   *
                                   *
                                   * THIS ROUTINE IS USED BY OTHER ROUTINES TO INCREMENT THE
                                   * STARTING ADDRESS IN A COMMAND AND COMPARE IT WITH THE FINAL
                                   * ADDRESS IN THE COMMAND.  ON RETURN THE CARRY FLAG SET
                                   * INDICATES THAT THE FINAL ADDRESS HAS BEEN REACHED
                                   *
      F1FC  2A 66 D0               ACHK     LHLD    BBUF           FETCH START ADDRESS
      F1FF  3A 69 D0                        LDA     BBUF+3         STOP ADDRESS (HIGH)
      F202  BC                              CMP     H              COMPARE ADDRESSES
      F203  C2 0E F2                        JNZ     ACH1
      F206  3A 68 D0                        LDA     BBUF+2         STOP ADDRESS (LOW)
      F209  BD                              CMP     L              COMPARE ADDRESSES
      F20A  C2 0E F2                        JNZ     ACH1
      F20D  37                              STC                    SET CARRY IF EQUAL
      F20E  23                     ACH1     INX     H              INCREMENT START ADDRESS
      F20F  22 66 D0                        SHLD    BBUF           STORE START ADDRESS
      F212  C9                              RET
                                   *
                                   *
                                   * THIS ROUTINE OUTPUTS CHARACTERS FROM A CHARACTER STRING UNTIL
                                   * A CARRIAGE RETURN IS FOUND
                                   *
      F213  46                     SCRN     MOV     B,M            FETCH CHARACTER
      F214  3E 0D                           MVI     A,13           CARRIAGE RETURN
      F216  B8                              CMP     B              CHARACTER = CR
      F217  C8                              RZ
      F218  CD 98 F0                        CALL    OUT8           OUTPUT CHARACTER
      F21B  23                              INX     H              INCREMENT ADDRESS
      F21C  C3 13 F2                        JMP     SCRN
```

```
                          *
                          *
                          * THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                          * ASCII HEXADECIMAL DIGITS AND STORES THEM IN MEMORY
                          *
F21F  21 58 D0            BINH    LXI     H,HCON         CONVERSION ADDRESS
F222  47                          MOV     B,A            SAVE VALUE
F223  1F                          RAR
F224  1F                          RAR
F225  1F                          RAR
F226  1F                          RAR
F227  CD 32 F2                    CALL    BIN1
F22A  77                          MOV     M,A
F22B  23                          INX     H
F22C  78                          MOV     A,B
F22D  CD 32 F2                    CALL    BIN1           CONVERT TO ASCII
F230  77                          MOV     M,A
F231  C9                          RET
                          *
                          * THIS ROUTINE CONVERTS A VALUE TO HEXADECIMAL
                          *
F232  E6 0F               BIN1    ANI     H'0F'          LOW FOUR BITS
F234  C6 30                       ADI     48             MODIFY FOR ASCII
F236  FE 3A                       CPI     58             DIGIT 0-9
F238  D8                          RC
F239  C6 07                       ADI     7              MODIFY FOR A-F
F23B  C9                          RET
                          *
                          *
                          * THIS ROUTINE CONVERTS THE BINARY VALUE IN REG A INTO
                          * ASCII DECIMAL DIGITS AND STORES THEM IN MEMORY
                          *
F23C  21 58 D0            BINC    LXI     H,HCON         CONVERSION ADDRESS
F23F  06 64                       MVI     B,100
F241  CD 4D F2                    CALL    BID1           CONVERT HUNDREDS DIGIT
F244  06 0A                       MVI     B,10
F246  CD 4D F2                    CALL    BID1           CONVERT TENS DIGIT
F249  C6 30                       ADI     A'0'           GET UNITS DIGIT
F24B  77                          MOV     M,A            STORE IN MEMORY
F24C  C9                          RET
                          *
                          * THIS ROUTINE CONVERTS A VALUE TO DECIMAL
                          *
F24D  36 2F               BID1    MVI     M,A'0'-1       INITIALIZE DIGIT COUNT
F24F  34                          INR     M
F250  90                          SUB     B              CHECK DIGIT
F251  D2 4F F2                    JNC     BID1+2
F254  80                          ADD     B              RESTORE VALUE
F255  23                          INX     H
F256  C9                          RET
                          *
                          *
                          * LEGAL COMMAND TABLE
                          *
```

```
F257  44 55 4D 50         CTAB    DATA    A'DUMP'        DUMP COMMAND
F25B  F2 98                       DDB     DUMP           COMMAND ADDRESS
F25D  45 58 45 43                 DATA    A'EXEC'        EXECUTE COMMAND
F261  F0 B9                       DDB     EXEC           COMMAND ADDRESS
F263  45 4E 54 52                 DATA    A'ENTR'        ENTER COMMAND
F267  F4 37                       DDB     ENTR           COMMAND ADDRESS
F269  46 49 4C 45                 DATA    A'FILE'        FILE COMMAND
F26D  F2 FF                       DDB     FILE           COMMAND ADDRESS
F26F  4C 49 53 54                 DATA    A'LIST'        LIST COMMAND
F273  F5 84                       DDB     LIST           COMMAND ADDRESS
F275  44 45 4C 54                 DATA    A'DELT'        DELETE COMMAND
F279  F5 E1                       DDB     DELL           COMMAND ADDRESS
F27B  41 53 53 4D                 DATA    A'ASSM'        ASSEMBLE COMMAND
F27F  F6 56                       DDB     ASSM           COMMAND ADDRESS
F281  50 41 47 45                 DATA    A'PAGE'        PAGE TRANSFER COMMAND
F285  F2 D8                       DDB     PAGE           COMMAND ADDRESS
F287  50 52 4F 4D                 DATA    A'PROM'        PROM PROGRAM COMMAND
F28B  F5 9D                       DDB     PROM           DEFINE COMMAND ADDRESS
F28D  43 55 53 54                 DATA    A'CUST'        CUST COMMAND
F291  E0 00                       DDB     H'E000'        COMMAND ADDRESS
                          *
                          *
                          * THIS ROUTINE CHECKS IF ANY PARAMETERS WERE ENTERED
                          * WITH THE COMMAND, IF NOT AN ERROR MESSAGE IS ISSUED
                          *
F293  3A 5A D0            VCHK    LDA     ABUF           FETCH PARAMETER BYTE
F296  B7                          ORA     A              SET FLAGS
F297  CA 1A F4                    JZ      WHAT           NO PARAMETER
F29A  C9                          RET
                          *
                          *
                          * THIS ROUTINE DUMPS OUT THE CONTENTS OF MEMORY FROM
                          * THE START TO FINAL ADDRESSES GIVEN IN THE COMMAND
                          *
F29B  CD 93 F2            DUMP    CALL    VCHK           CHECK FOR PARAMETERS
F29E  3E 10                       MVI     A,16           LOCATIONS PER LINE
F2A0  32 6A D0                    STA     SCNT           DUMP COUNTER
F2A3  CD A6 F0            DUM1    CALL    CRLF           START NEW LINE
F2A6  3A 67 D0                    LDA     BBUF+1         FETCH ADDRESS
F2A9  CD D3 F1                    CALL    HOUT           OUTPUT ADDRESS
F2AC  3A 66 D0                    LDA     BBUF
F2AF  CD E3 F1                    CALL    HOTB           OUTPUT ADDRESS
F2B2  3A 6A D0                    LDA     SCNT           FETCH LINE COUNTER
F2B5  32 6B D0            DUM3    STA     DCNT
F2B8  2A 66 D0                    LHLD    BBUF           FETCH MEMORY ADDRESS
F2BB  7D                          MOV     A,L            GET LOW ORDER ADDRESS
F2BC  D3 07                       OUT     PADO           SET PROM ADDRESS
F2BE  7C                          MOV     A,H
F2BF  FE EF                       CPI     H'EF'
F2C1  7E                          MOV     A,M
F2C2  C2 C7 F2                    JNZ     DUM2
F2C5  DB 06                       IN      PDAI           READ PROM DATA
F2C7  CD E3 F1            DUM2    CALL    HOTB           OUTPUT VALUE
F2CA  CD FC F1                    CALL    ACHK           CHECK ADDRESS
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR

      F2CD  D8                            RC                          RETURN IF FINISHED
      F2CE  3A 6B D0                      LDA      DCNT               FETCH COUNTER
      F2D1  3D                            DCR      A                  DECREMENT COUNTER
      F2D2  C2 B5 F2                      JNZ      DUM1
      F2D5  C3 A3 F2                      JMP      DUMS
                                    *
                                    *
                                    * THIS ROUTINE WILL MOVE 1 PAGE (256 BYTES) FROM 1ST ADDRESS GIVEN IN
                                    * COMMAND TO 2ND ADDRESS IN COMMAND
                                    *
      F2D8  CD 93 F2                PAGE  CALL     VCHK               CHECK FOR PARAMETER
      F2DB  3A 5E D0                      LDA      ABUF+4             FETCH 2ND PARAMETER
      F2DE  B7                            ORA      A                  DOES 2ND PARAMETER EXIST
      F2DF  CA 1A F4                      JZ       WHAT
      F2E2  2A 66 D0                      LHLD     BBUF               FETCH MOVE FROM ADDRESS
      F2E5  EB                            XCHG
      F2E6  2A 68 D0                      LHLD     BBUF+2             FETCH MOVE TO ADDRESS
      F2E9  06 00                         MVI      B,0                SET COUNTER
      F2EB  7B                      PAG1  MOV      A,E
      F2EC  D3 07                         OUT      PADO               SET PROM ADDRESS
      F2EE  7A                            MOV      A,D
      F2EF  FE FF                         CPI      H'FF'              CHECK FOR PROM ADDRESS
      F2F1  1A                            LDAX     D                  GET DATA
      F2F2  C2 F7 F2                      JNZ      PAG2
      F2F5  DB 06                         IN       PDAI               READ PROM DATA
      F2F7  77                      PAG2  MOV      M,A
      F2F8  23                            INX      H
      F2F9  13                            INX      D
      F2FA  05                            DCR      B                  DECREMENT COUNT
      F2FB  C2 EB F2                      JNZ      PAG1
      F2FE  C9                            RET
                                    *
                                    *
                                    *
                                    * THIS ROUTINE INITIALIZES THE BEGINNING OF FILE ADDRESS
                                    * AND END OF FILE ADDRESS AS WELL AS INITIALIZES THE FILE AREA
                                    * WHEN THE FILE COMMAND IS USED
                                    *
      F2FF  CD A6 F0                FILE  CALL     CRLF
                                    * CHECK FOR FILE PARAMETERS
      F302  3A 52 D0                      LDA      FBUF
      F305  B7                            ORA      A
      F306  CA 79 F3                      JZ       FOUT               NO - GO LIST
      F309  CD D8 F3                      CALL     FSEA               LOOK UP FILE
      F30C  EB                            XCHG                        PNTR IN DE
      F30D  C2 24 F3                      JNZ      TEST               IF FOUND
                                    * NO ENTRY
      F310  3A 5A D0                      LDA      ABUF               CHECK FOR PARAM
      F313  B7                            ORA      A
      F314  CA 1D F4                      JZ       WHA1               NOT? - GIVE EM HELL
                                    * CHECK FOR ROOM IN DIRECTORY
      F317  3A 59 D0                      LDA      FEF
      F31A  B7                            ORA      A
      F31B  C2 39 F3                      JNZ      ROOM
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR

      F31E  21 2C F4                      LXI      H,EMES1
      F321  C3 20 F4                      JMP      MESS
                                    * ENTRY FOUND ARE THESE PARAMETERS
      F324  3A 5A D0                TEST  LDA      ABUF
      F327  B7                            ORA      A
      F328  CA 4B F3                      JZ       SWAPS
      F32B  2A 66 D0                      LHLD     BBUF
      F32E  7C                            MOV      A,H
      F32F  B5                            ORA      L
      F330  CA 4B F3                      JZ       SWAPS
      F333  21 31 F4                      LXI      H,EMES2            NO-NO CAN'T DO
      F336  C3 20 F4                      JMP      MESS               IT - DELETE FIRST
                                    * MOVE FILE NAME TO BLOCK POINTED TO BY FREAD
      F339  2A 57 D0                ROOM  LHLD     FREAD
      F33C  EB                            XCHG                        DIRECT POINTER IN D,E
      F33D  21 52 D0                      LXI      H,FBUF             FILE NAME POINTER IN H,L
      F340  D5                            PUSH     D
      F341  0E 05                         MVI      C,NMLEN            NAME LENGTH COUNT
      F343  7E                      MOV23 MOV      A,M
      F344  12                            STAX     D
      F345  13                            INX      D
      F346  0D                            DCR      C                  TEST COUNT
      F347  C3 B5 FC                      JMP      PATCH #1
      F34A  D1                      P+3   POP      D                  RESTORE ENTRY PTR, MAKE CURRENT
                                    * MAKE FILE POINTED TO BY D,E CURRENT
      F34B  21 08 D0                SWAPS LXI      H,FILE0
      F34E  0E 0D                         MVI      C,FELEN            ENTRY LENGTH
      F350  1A                      SWAP  LDAX     D
      F351  46                            MOV      B,M
      F352  77                            MOV      M,A                EXCHANGE
      F353  78                            MOV      A,B
      F354  12                            STAX     D
      F355  13                            INX      D                  BUMP POINTERS
      F356  23                            INX      H
      F357  0D                            DCR      C                  TEST COUNT
      F358  C2 50 F3                      JNZ      SWAP
                                    * CHECK FOR 2ND PARAMETER, => INITIALIZE NEW
      F35B  3A 5A D0                      LDA      ABUF
      F35E  B7                            ORA      A
      F35F  CA 83 F3                      JZ       FOOT               NO SECOND PARAMETER
                                    * PROCESS 2ND PARAMETER
      F362  2A 66 D0                      LHLD     BBUF               GET ADDRESS
      F365  22 05 D0                      SHLD     BOFP               SET BEGIN
      F368  22 07 D0                      SHLD     EOFP               SET END
      F36B  7D                            MOV      A,L                IS ADDRESS ZERO
      F36C  B4                            ORA      H
      F36D  CA 72 F3                      JZ       FIL35              YES
      F370  36 01                   FIL30 MVI      M,1                NON ZERO - SET EOF
      F372  AF                      FIL35 XRA      A
      F373  32 09 D0                      STA      MAXL               AND MAX LINE #
      F376  C3 83 F3                      JMP      FOOT               OUTPUT PARAMETERS
      F379  3A 80 D0                FOUT  LDA      IBUF+4
      F37C  FE 53                         CPI      A'S'               IS COMMAND FILES
      F37E  0E 06                         MVI      C,MAXFIL
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                MICROTEC MACRO ASS-4BLER VER 3.1            PAGE   13

649   F380  CA 85 F3                         JZ        FOUL
650   F383  0E 01                   FOOT     MVI       C,1
651                                 * OUTPUT THE # OF ENTIRES IN C
652   F385  21 00 D0                FOUL     LXI       H,FILE0
653   F388  79                               MOV       A,C
654   F389  32 59 D0                FINE     STA       FOCNT              SA E COUNT
655   F38C  E5                               PUSH      H
656   F38D  11 05 00                         LXI       D,NMLEN
657   F390  19                               DAD       D
658   F391  7E                               MOV       A,M
659   F392  B7                               ORA       A
660   F393  C2 A3 F3                         JNZ       FOOD               NON ZERO, OK TO OUTPUT
661   F396  23                               INX       H
662   F397  86                               ADD       M
663   F398  23                               INX       H
664   F399  C2 A3 F3                         JNZ       FOOD
665   F39C  33                               INX       SP
666   F39D  33                               INX       SP
667   F39E  23                               INX       H
668   F39F  23                               INX       H
669   F3A0  C3 B8 F3                         JMP       FEET
670                                 * HAVE AN ENTRY TO OUTPUT
671   F3A3  E1                      FOOD     POP       H                  PTR
672   F3A4  0E 05                            MVI       C,NMLEN
673   F3A6  46                      FAST     MOV       B,M                LOAD CHARACTER TO B
674   F3A7  CD 9B F0                         CALL      OUTB               OUTPUT
675   F3AA  0D                               DCR       C
676   F3AB  23                               INX       H
677   F3AC  C2 A6 F3                         JNZ       FAST               DO THE REST
678                                 * NOW OUTPUT BEGIN-END PTRS
679   F3AF  CD C4 F3                         CALL      FOOL               OUTPUT BEGIN
680   F3B2  CD C4 F3                         CALL      FOOL               OUTPUT END
681   F3B5  CD A6 F0                         CALL      CRLF               AND C/R
682                                 * TEST COUNT, H,L POINTS PAST E FP
683   F3B8  11 04 00                FEET     LXI       D,FELEN-NM EN-4
684   F3BB  19                               DAD       D                  MOVE TO NEXT ENTRY
685   F3BC  3A 59 D0                         LDA       FOCNT
686   F3BF  3D                               DCR       A                  TEST COUNT
687   F3C0  C2 89 F3                         JNZ       FINE               MORE TO DO
688   F3C3  C9                               RET                          DONE!!!
689                                 * OUTPUT NUMBER POINTED TO BY H L
690                                 * ON RET, H,L POINT 2 WORDS LATER
691   F3C4  CD F6 F1                FOOL     CALL      BLK1               SPACE
692   F3C7  23                               INX       H
693   F3C8  7E                               MOV       A,M
694   F3C9  2B                               DCX       H
695   F3CA  E5                               PUSH      H
696   F3CB  CD D3 F1                         CALL      HOUT               OUTPUT
697   F3CE  E1                               POP       H
698   F3CF  7E                               MOV       A,M
699   F3D0  23                               INX       H
700   F3D1  23                               INX       H
701   F3D2  E5                               PUSH      H
702   F3D3  CD E3 F1                         CALL      HOTB               OUTPUT
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                MICROTEC MACRO A SEMBLER VER 3.1            PAGE   14

703   F3D6  E1                               POP       H                  RESTORE H,L
704   F3D7  C9                               RET
705                                 * SEARCH THE FILE DIRECTORY FOR THE FILE
706                                 * WHOSE NAME IS IN FBUF.
707                                 * RETURN IF FOUND, ZERO IS OFF, H,L POINT TO
708                                 * ENTRY WHILE SEARCHING, ON EN RY FOUND WITH ADDR
709                                 * ZERO, SET FEF TO > 0 AND FREA  TO THE ADDR OF ENTRY.
710                                 *
711   F3D8  AF                      FSEA     XRA       A
712   F3D9  32 59 D0                         STA       FEF                CLAIM NO FREE ENTRIES
713   F3DC  06 06                            MVI       B,MAXFIL           COUNT OF ENTRIES
714   F3DE  11 00 D0                         LXI       D,FILE0            TABLE ADDRESS
715   F3E1  21 52 D0                FSE10    LXI       H,FBUF
716   F3E4  0E 05                            MVI       C,NMLEN
717   F3E6  CD E8 F0                         CALL      SEAR               TEST STRINGS
718   F3E9  F5                               PUSH      PSW                SAVE FLAG
719   F3EA  D5                               PUSH      D
720   F3EB  1A                               LDAX      D                  GET BOFP
721   F3EC  B7                               ORA       A                   EMPTY ENTRY?
722   F3ED  C2 0E F4                         JNZ       FSE20
723   F3F0  13                               INX       D                  TEST OTHER WORD
724   F3F1  1A                               LDAX      D
725   F3F2  B7                               ORA       A
726   F3F3  C2 0E F4                         JNZ       FSE20              NOPE-GO TEST FOR MATCH
727   F3F6  EB                               XCHG                         H,L GET MIDDLE OF FREE ENTRY
728   F3F7  11 FA FF                         LXI       D,-NMLEN-1
729   F3FA  19                               DAD       D                  MOVE TO BEGINNING
730   F3FB  22 57 D0                         SHLD      FREAD              SAVE ADDR
731   F3FE  7A                               MOV       A,D
732   F3FF  32 59 D0                         STA       FEF                SET FREE ENTRY FOUND
733   F402  E1                               POP       H                  RESTORE INTERIM PTR
734   F403  F1                               POP       PSW                UNJUNK STACK
735                                 * MOVE TO NEXT ENTRY
736   F404  11 08 00                FSE15    LXI       D,FELEN-NM N
737   F407  19                               DAD       D
738   F408  EB                               XCHG      NEXT ENTRY ADDR IN DE
739   F409  05                               DCR       B                  TEST COUNT
740   F40A  C8                               RZ                           DONE---NOPE
741   F40B  C3 E1 F3                         JMP       FSE10              TRY  EXT
742                                 * ENTRY WASN'T FREE, TEST FOR M TCH
743   F40E  E1                      FSE20    POP       H
744   F40F  F1                               POP       PSW
745   F410  C2 04 F4                         JNZ       FSE15              IF ZERO CLEAR, NO MATCH
746                                 * ENTRY FOUND
747   F413  11 FB FF                         LXI       D,-NMLEN           BACKUP
748   F416  19                               DAD       D                  H,L POINTS TO ENTRY
749   F417  7A                               MOV       A,D
750   F418  B7                               ORA       A                  CLEAR ZERO
751   F419  C9                               RET                          THAT'S ALL
752                                 *
753                                 *
754                                 * OUTPUT ERROR MESSAGE FOR ILLEG L COMMAND
755                                 *
756   F41A  CD A6 F0                WHAT     CALL      CRLF               OU  CRLF
```

```
F41D  21 26 F4          WHA1    LXI     H,EMES      MESSAGE ADDRESS
F420  CD 13 F2          MESS    CALL    SCRN
F423  C3 1F F0                  JMP EOR
                        *
F426  57 48 41 54       EMES    DATA    A'WHAT?'
F42A  3F
F42B  0D                        DATA    13          CARRIAGE RETURN
F42C  46 55 4C 4C       EMES1   DATA    A'FULL',13
F430  0D
F431  4E 4F 20 4E       EMES2   DATA    A'NO NO',13
F435  4F 0D
                        *
                        *
                        * CALL ROUTINE TO ENTER DATA INTO MEMORY
                        * AND CHECK FOR ERROR ON RETURN
                        *
                        * THIS ROUTINE IS USED TO ENTER DATA VALUES INTO MEMORY.
                        * EACH VALUE IS ONE BYTE AND IS WRITTEN IN HEXADECIMAL
                        * VALUES GREATER THAN 255 WILL CAUSE CARRY TO BE SET
                        * AND RETURN MADE TO CALLING PROGRAM
                        *
F437  CD 93 F2          ENTR    CALL    VCHK        CHECDK FOR PARAMETERS
F43A  CD 44 F4                  CALL    ENTS
F43D  DA 1A F4                  JC      WHAT
F440  CD A6 F0                  CALL    CRLF
F443  C9                        RET
                        *
                        *
F444  00 2F             EEND    EQU     A'/'        TERMINATION CHARACTER
F444  CD A6 F0          ENTS    CALL    CRLF
F447  CD 25 F0                  CALL    READ        READ INPUT DATA
F44A  21 AC D0                  LXI     H,IBUF      SET LINE POINTER
F44D  22 72 D0                  SHLD    PNTR        SAVE POINTER
F450  CD FE F0          ENT1    CALL    ZBUF        CLEAR BUFFER
F453  CD F7 F8                  CALL    SBLK        SCAN TO FIRST VALUE
F456  DA 44 F4                  JC      ENTS        JUMP IF CR FOUND
F459  FE 2F                     CPI     EEND
F45B  C8                        RZ                  RETURN (CARRY IS ZERO)
F45C  CD 58 F8                  CALL    ALPS        PLACE VALUE IN BUFFER
F45F  78                        MOV     A,B         GET DIGIT COUNT
F460  FE 03                     CPI     3           CHECK NUMBER OF DIGITS
F462  3F                        CMC
F463  D8                        RC                  RETURN IF MORE THAN 2 DIGITS
F464  01 5A D0                  LXI     B,ABUF      CONVERSION ADDRESS
F467  CD B4 F1                  CALL    AHEX        CONVERT VALUE
F46A  D8                        RC                  ERROR IN HEX CHARACTER
F46B  7D                        MOV     A,L
F46C  2A 66 D0                  LHLD    BBUF        FETCH MEMORY ADDRESS
F46F  77                        MOV     M,A         PUT IN MEMORY
F470  CD 0E F2                  CALL    ACH1        INCREMENT MEMORY LOCATION
F473  C3 50 F4                  JMP     ENT1
                        *
                        *
                        *
```

```
                        * THIS ROUTINE IS USED TO ENTER LINES INTO THE FILE
                        * AREA.  THE LINE NUMBER IS FIRST CHECKED TO SEE IF IT IS
                        * A VALID NUMBER (0000-9999).  NEXT IT IS CHECKED TO SEE IF IT IS
                        * GREATER THAN THE MAXIMUM CURRENT LINE NUMBER.  IF IT IS THE NEW
                        * LINE IS INSERTED AT THE END OF THE CURRENT FILE AND THE MAXIMUM
                        * LINE NUMBER IS UPDATED AS WELL AS THE END OF FILE POSITION
                        * LINE NUMBERS THAT ALREADY EXIST ARE INSERTED INTO THE FILE AREA
                        * AT THE APPROPRIATE PLACE AND ANY EXTRA CHARACTERS IN THE OLD
                        * LINE ARE DELETED
                        *
F476  0E 04             LINE    MVI     C,4         NO OF DIGITS TO CHECK
F478  21 AB D0                  LXI     H,IBUF-1    INITIALIZE ADDRESS
F47B  23                LICK    INX     H
F47C  7E                        MOV     A,M         FETCH LINE DIGIT
F47D  FE 30                     CPI     A'0'        CHECK FOR VALID NUMBER
F47F  DA 1A F4                  JC      WHAT
F482  FE 3A                     CPI     A'9'+1
F484  D2 1A F4                  JNC     WHAT
F487  0D                        DCR     C
F488  C2 7B F4                  JNZ     LICK
F48B  22 50 D0                  SHLD    ADDS        FIND ADDRESS
F48E  11 0C D0                  LXI     D,MAXL+3    SET ADDRESS
F491  CD 56 F5                  CALL    COMO
F494  D2 B3 F4                  JNC     INSR
                        * GET HERE IF NEW LINE IS GREATER THAN MAXIMUM LN
F497  23                        INX     H
F498  CD 46 F5                  CALL    LODM        GET NEW LINE NUMBER
F49B  21 0C D0                  LXI     H,MAXL+3
F49E  CD 4E F5                  CALL    STOM        MAKE IT MAXIMUM LINE NUMBER
F4A1  11 AB D0                  LXI     D,IBUF-1
F4A4  2A 07 D0                  LHLD    EOFP        END OF FILE POSITION
F4A7  0E 01                     MVI     C,1
F4A9  CD 34 F5                  CALL    LMOV        PLACE LINE IN FILE
F4AC  71                SEOF    MOV     M,C         END OF FILE INDICATOR
F4AD  22 07 D0                  SHLD    EOFP        END OF FILE ADDRESS
F4B0  C3 1F F0                  JMP     EOR
                        * GET HERE IF NEW LINE MUST BE INSERTED INTO ALREADY EXISTING
                        * FILE AREA
F4B3  CD 08 F5          INSR    CALL    FIN1        FIND LINE IN FILE
F4B6  0E 02                     MVI     C,2
F4B8  CA BC F4                  JZ      EQUL
F4BB  0D                        DCR     C           NEW LN NOT EQUAL TO SOME OLD LN
F4BC  46                EQUL    MOV     B,M
F4BD  2B                        DCX     H
F4BE  36 02                     MVI     M,2         MOVE LINE INDICATOR
F4C0  22 4E D0                  SHLD    INSP        INSERT LINE POSITION
F4C3  3A AB D0                  LDA     IBUF-1      NEW LN COUNT
F4C6  0D                        DCR     C
F4C7  CA D1 F4                  JZ      LT          NEW LN NOT = OLD LN
F4CA  90                        SUB     B           COUNT DIFFERENCE
F4CB  CA F4 F4                  JZ      ZERO        LINE LENGTHS EQUAL
F4CE  DA E4 F4                  JC      GT
                        * GET HERE IF NO OF CHARS IN OLD LINE > NO OF CHARS IN NEW LINE
                        * OR NEW LINE NUMBER WAS NOT EQUAL TO SOME OLD LINE NUMBER
```

```
F4D1  2A 07 D0        LT      LHLD    EOFP           END OF FILE ADDRESS
F4D4  54                      MOV     D,H
F4D5  5D                      MOV     E,L
F4D6  CD 2F F5                CALL    ADR            MOVE TO ADDRESS
F4D9  22 07 D0                SHLD    EOFP           NEW END OF FILE ADDRESS
F4DC  0E 02                   MVI     C,2
F4DE  CD 3D F5                CALL    RMOV           OPEN UP FILE AREA
F4E1  C3 F4 F4                JMP     ZERO
                      * GET HERE IF NO OF CHARS IN OLD LINE, NO OF CHARS IN NEW LINE
F4E4  2F              [illegible] CMA
F4E5  3C                      INR     A              COUNT DIFFERENCE
F4E6  54                      MOV     D,H
F4E7  5D                      MOV     E,L
F4E8  CD 2F F5                CALL    ADR
F4EB  EB                      XCHG
F4EC  CD 34 F5                CALL    LMOV           DELETE EXCESS CHARACTERS IN FILE
F4EF  36 01                   MVI     M,1            E-O-F INDICATOR
F4F1  22 07 D0                SHLD    EOFP           E-O-F ADDRESS
                      * GET HERE TO INSERT CURRENT LINE INTO FILE AREA
F4F4  2A 4E D0        ZERO    LHLD    INSP           INSERT ADDRESS
F4F7  36 0D                   MVI     M,ASCR
F4F9  23                      INX     H
F4FA  11 48 D0                LXI     D,IBUF-1       NEW LINE ADDRESS
F4FD  0E 01                   MVI     C,1            CHECK VALUE
F4FF  CD 34 F5                CALL    LMOV           PLACE LINE IN FILF
F502  C3 1F F0                JMP     EOR
                      *
                      *
                      * THIS ROUTINE IS USED TO FIND A LN IN THE FILF AREA
                      * WHICH IS GREATER THAN OR EQUAL TO THE CURRENT LINE NUMBER
                      *
F505  21 5D D0        FIND    LXI     H,ABUF+3       BUFFER ADDRESS
F508  22 5D D0                SHLD    ADDS           SAVE ADDRESS
F50B  2A 05 D0        FIN1    LHLD    BOFP           BEGIN FILE ADDRESS
F50E  CD 28 F5        FI1     CALL    EO1            CHECK FOR END OF FILE
F511  EB                      XCHG
F512  2A 5D D0                LHLD    ADDS           FETCH FIND ADDRESS
F515  EB                      XCHG
F516  3E 04                   MVI     A,4
F518  CD 2F F5                CALL    ADR            LN ADDRESS
F51B  CD 56 F5                CALL    COMO           COMPARE LINE NUMBERS
F51E  D8                      RC
F51F  C8                      RZ
F520  7E              FI2     MOV     A,M
F521  CD 2F F5                CALL    ADR            NEXT LINE ADDRESS
F524  C3 0E F5                JMP     FI1
                      *
                      *
                      * THIS ROUTINE CHECKS IF THE CURRENT ADDRESS
                      * WHEN SEARCHING THROUGH THE FILE AREA IS THE END OF FILE
                      *
F527  23              EOF     INX     H
F528  3E 01           EO1     MVI     A,1            E-O-F INDICATOR
F52A  BE                      CMP     M
```

```
F52B  C0                      RNZ
F52C  C3 1F F0                JMP     EOR
                      *
                      *
                      * THIS ROUTINE IS USED TO ADD A VALUE TO AN ADDRESS
                      * CONTAINED IN REGISTER H,L
                      *
F52F  85              ADR     ADD     L
F530  6F                      MOV     L,A
F531  D0                      RNC
F532  24                      INR     H
F533  C9                      RET
                      *
                      *
                      * THIS ROUTINE WILL MOVE CHARACTER STRINGS FROM ONE LOCATION
                      * OF MEMORY TO ANOTHER
                      * CHARACTERS ARE MOVED FROM LOCATION ADDRESSED BY D,E TO LOCATION
                      * ADDRESSED BY H,L.  ADDITIONAL CHARACTERS ARE MOVED BY
                      * INCREMENTING MEMORY UNTIL THE CHARACTER IN REGISTER C IS FETCHED
                      *
F534  1A              LMOV    LDAX    D              FETCH CHARACTER
F535  13                      INX     D              INCREMENT FETCH ADDRESS
F536  B9                      CMP     C              TERMINATION CHARACTER
F537  C8                      RZ
F538  77                      MOV     M,A            STORE CHARACTER
F539  23                      INX     H              INCREMENT STORE ADDRESS
F53A  C3 34 F5                JMP     LMOV
                      *
                      *
                      * THIS ROUTINE IS SMMILAR TO ABOVE EXCEPT THAT THE CHARACTER ADDRESS
                      * IS DECREMENTED AFTER EACH FETCH AND STORE
                      *
F53D  1A              RMOV    LDAX    D              FETCH CHARACTER
F53E  1B                      DCX     D              DECREMENT FETCH ADDRESS
F53F  B9                      CMP     C              TERMINATION CHARACTER
F540  C8                      RZ
F541  77                      MOV     M,A            STORE CHARACTER
F542  2B                      DCX     H              DECREMENT STORE ADDRESS
F543  C3 3D F5                JMP     RMOV
                      *
                      *
                      * THIS ROUTINE IS USED TO LOAD FOUR CHARACTERS FROM
                      * MEMORY INTO REGISTERS
                      *
F546  46              LODM    MOV     B,M            FETCH CHARACTER
F547  23                      INX     H
F548  4E                      MOV     C,M            FETCH CHARACTER
F549  23                      INX     H
F54A  56                      MOV     D,M            FETCH CHARACTER
F54B  23                      INX     H
F54C  5E                      MOV     E,M            FETCH CHARACTER
F54D  C9                      RET
                      *
                      *
```

```
                              * THIS ROUTINE STORES FOUR CHARACTERS FROM REGISTERS
                              * INTO MEMORY
                              *
       F54E  73               STOM     MOV       M,E             STORE CHARACTER
       F54F  2B                        DCX       H
       F550  72                        MOV       M,D             STORE CHARACTER
       F551  2B                        DCX       H
       F552  71                        MOV       M,C             STORE CHARACTER
       F553  2B                        DCX       H
       F554  70                        MOV       M,B             STORE CHARACTER
       F555  C9                        RET
                              *
                              *
                              * THIS ROUTINE IS USED TO COMPARE TWO CHARACTER STRINGS
                              * OF LENGTH 4.  ON RETURN ZERO FLAG SET MEANS BOTH
                              * STRINGS ARE EQUAL.  CARRY FLAG = 0 MEANS STRING ADDRESSED
                              * BY D,E WAS GREATER THAN OR EQUAL TO CHARACTER STRING
                              * ADDRESSED BY H,L
                              *
       F556  06 01            COM0     MVI       B,1             EQUAL COUNTER
       F558  0E 04                     MVI       C,4             STRING LENGTH
       F55A  B7                        ORA       A               CLEAR CARRY
       F55B  1A               CO1      LDAX      D               FETCH CHARACTER
       F55C  9E                        SBB       M               COMPARE CHARACTERS
       F55D  CA 61 F5                  JZ        CO2
       F560  04                        INR       B               INCREMENT EQUAL COUNTER
       F561  1B               CO2      DCX       D
       F562  2B                        DCX       H
       F563  0D                        DCR       C
       F564  C2 5B F5                  JNZ       CO1
       F567  05                        DCR       B
       F568  C9                        RET
                              *
                              * THIS ROUTINE IS SIMILAR TO THE ABOVE ROUTINE EXCEPT ON
                              * RETURN CARRY FLAG = 0 MEANS THAT CHARACTER STRING ADDRESSED
                              * BY D,E IS ONLY GREATER THAN STRING ADDRESSED BY H,L
                              *
       F569  0E 04            COM1     MVI       C,4             STRING LENGTH
       F56B  1A                        LDAX      D               FETCH CHARACTER
       F56C  D6 01                     SUI       1
       F56E  C3 5C F5                  JMP       CO1+1
                              *
                              *
                              * THIS ROUTINE WILL TAKE ASCII CHARACTERS AND ADD ANY
                              * NECESSARY ASCII ZEROS SO THE RESULT IS A 4 CHARACTER
                              * ASCII VALUE
                              *
       F571  CD 46 F5         NORM     CALL      LODM            LOAD CHARACTERS
       F574  AF                        XRA       A               FETCH A ZERO
       F575  B8                        CMP       B
       F576  C8                        RZ
       F577  BB               NOR1     CMP       E
       F578  C4 4E F5                  CNZ       STOM            STORE VALUES
       F57B  C0                        RNZ
```

```
       F57C  5A                        MOV       E,D             NORMALIZE VALUE
       F57D  51                        MOV       D,C
       F57E  48                        MOV       C,B
       F57F  06 30                     MVI       B,A'0'
       F581  C3 77 F5                  JMP       NOR1
                              *
                              *
                              * THIS ROUTINE IS USED TO LIST THE CONTENTS OF THE FILE
                              * AREA STARTING AT THE LINE NUMBER GIVEN IN THE COMMAND
                              *
       F584  CD A6 F0         LIST     CALL      CRLF
       F587  CD 05 F5                  CALL      FIND            FIND STARTING LN
       F58A  23                        INX       H
       F58B  CD 13 F2         LIS1     CALL      SCRN            OUTPUT LINE
       F58E  CD A6 F0                  CALL      CRLF
       F591  CD 27 F5                  CALL      EOF             END OF FILE
       F594  DB FF                     IN        SWCH            READ SWITCHES
       F596  E6 80                     ANI       H'80'
       F598  C0                        RNZ
       F599  23                        INX       H
       F59A  C3 8B F5                  JMP       LIS1
                              *
                              * THIS ROUTINE IS USED TO PROGRAM A 1702A PROM
                              *
       F59D  CD 93 F2         PROM     CALL      VCHK            CHECK FOR PARAMETER
       F5A0  CD A6 F0         PRO1     CALL      CRLF
       F5A3  3A 66 D0                  LDA       BBUF            GET ADDRESS (LOW)
       F5A6  CD E3 F1                  CALL      HOT8            OUTPUT ADDRESS
       F5A9  16 03                     MVI       D,3             NUMBER OF ATTEMPTS
       F5AB  2A 66 D0                  LHLD      BBUF            GET ADDRESS
       F5AE  7D               PRO2     MOV       A,L
       F5AF  D3 07                     OUT       PADO
       F5B1  7E                        MOV       A,M             GET DATA
       F5B2  D3 08                     OUT       PDAO            OUTPUT TO PROM
       F5B4  3E 02                     MVI       A,2
       F5B6  D3 09                     OUT       PCTO            ENABLE PROGRAMMER
       F5B8  CD D5 F5                  CALL      DLAY            500 MSEC DELAY
       F5BB  AF                        XRA       A               GET A ZERO
       F5BC  D3 09                     OUT       PCTO            DISABLE PROGRAMMER
       F5BE  DB 06                     IN        PDAI            READ DATA
       F5C0  BE                        CMP       M               COMPARE DATA
       F5C1  CA CE F5                  JZ        PRO3
       F5C4  06 3F                     MVI       B,A'?'          ERROR INDICATOR
       F5C6  CD 9B F0                  CALL      OUT8
       F5C9  15                        DCR       D               NUMBER OF ATTEMPTS
       F5CA  C2 AE F5                  JNZ       PRO2            TRY AGAIN
       F5CD  C9                        RET
       F5CE  CD FC F1         PRO3     CALL      ACHK            FINAL ADDRESS
       F5D1  D8                        RC                        RETURN IF FINISHED
       F5D2  C3 A0 F5                  JMP       PRO1            NEXT LOCATION
                              *
       F5D5  1E 96            DLAY     MVI       E,150
       F5D7  AF               DLA1     XRA       A               GET A ZERO (256)
       F5D8  3D               DLA2     DCR       A
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR

      F5D9  C2 D8 F5                      JNZ     DLA2
      F5DC  1D                            DCR     E
      F5DD  C2 D7 F5                      JNZ     DLA1
      F5E0  C9                            RET
                               *
                               *
                               * THIS ROUTINE IS USED TO DELETE LINES FROM THE FILE AREA
                               * THE REMAINING FILE AREA IS THEN MOVED IN MEMORY SO THAT
                               * THERE IS NO EXCESS SPACE IN MEMORY
                               *
      F5E1  CD 93 F2           DELL       CALL    VCHK           CHECK FOR PARAMETER
      F5E4  CD 05 F5                      CALL    FIND           FIND LINE IN FILE AREA
      F5E7  22 4E D0                      SHLD    DELP           SAVE DELETE POSITION
      F5EA  21 61 D0                      LXI     H,ABUF+7
      F5ED  7E                            MOV     A,M            CHECK FOR 2ND PARAMETER
      F5EE  B7                            ORA     A              SET FLAGS
      F5EF  C2 F5 F5                      JNZ     DEL1
      F5F2  21 5D D0                      LXI     H,ABUF+3       USE FIRST PARAMETER
      F5F5  22 50 D0           DEL1       SHLD    ADDS           SAVE FIND ADDRESS
      F5F8  EB                            XCHG
      F5F9  21 0C D0                      LXI     H,MAXL+3
      F5FC  CD 56 F5                      CALL    COMO           COMPARE LINE NUMBERS
      F5FF  2A 4E D0                      LHLD    DELP           LOAD DELETE POSITION
      F602  DA 43 F6                      JC      NOVR
                               * GET HERE IF DELETION DELETION INVOLVES END OF FILE
      F605  22 07 D0                      SHLD    EOFP           CHANGE E-O-F POSITION
      F608  36 01                         MVI     M,1            SET E-O-F INDICATOR
      F60A  EB                            XCHG
      F60B  2A 05 D0                      LHLD    BOFP           GET BEGIN FILE ADDRESS
      F60E  EB                            XCHG
      F60F  06 0D                         MVI     B,13           SET SCAN SWITCH
      F611  2B                            DCX     H              DECREMENT FILE ADDRESS
      F612  7D                 DEL2       MOV     A,L            CHECK FOR BOF
      F613  93                            SUB     E
      F614  7C                            MOV     A,H
      F615  9A                            SBB     D
      F616  3E 0D                         MVI     A,ASCR         LOOK FOR CR
      F619  DA 3A F6                      JC      DEL4           DECREMENTED PAST BOF
      F61B  05                            DCR     B
      F61C  2B                            DCX     H
      F61D  BE                            CMP     M              FIND NEW MAX LN
      F61E  C2 12 F6                      JNZ     DEL2
      F621  2B                            DCX     H
      F622  7D                            MOV     A,L
      F623  93                            SUB     E
      F624  7C                            MOV     A,H
      F625  9A                            SBB     D
      F626  DA 3B F6                      JC      DEL5
      F629  BE                            CMP     M              END OF PREVIOUS LINE
      F62A  23                            INX     H
      F62B  23                            INX     H
      F62C  CA 30 F6                      JZ      DEL3
      F62F  23                            INX     H
      F630  CD 46 F5           DEL3       CALL    LODM           LOAD NEW MAX LN
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR

      F633  21 0C D0                      LXI     H,MAXL+3       SET ADDRESS
      F636  CD 4E F5                      CALL    STOM           STORE NEW MAX LN
      F639  C9                            RET
      F63A  B8                 DEL4       CMP     B              CHECK SWITCH
      F63B  EB                 DEL5       XCHG
      F63C  C2 2F F6                      JNZ     DEL3-1
      F63F  32 09 D0                      STA     MAXL           MAKE MAX LN A SMALL NUMBER
      F642  C9                            RET
                               * GET HERE IF DELETION IS IN MIDDLE OF FILE AREA
      F643  CD 0E F5           NOVR       CALL    FI1            FIND END OF DELETE AREA
      F646  CC 20 F5                      CZ      FI2            NEXT LINE IF THIS LN IS EQUAL
      F649  EB                 NOV1       XCHG
      F64A  2A 4E D0                      LHLD    DELP           CHAR MOVE TO POSITION
      F64D  0E 01                         MVI     C,1            MOVE TERMINATOR
      F64F  CD 34 F5                      CALL    LMOV           COMPACT FILE AREA
      F652  22 07 D0                      SHLD    EOFP           SET EOF POSITION
      F655  36 01                         MVI     M,1            SET EOF INDICATOR
      F657  C9                            RET
                               *
                               *
                               * STARTING HERE IS THE SELF ASSEMBLER PROGRAM
                               * THIS PROGRAM ASSEMBLES PROGRAMS WHICH ARE
                               * IN THE FILE AREA
                               *
      F658  CD 93 F2           ASSM       CALL    VCHK           CHECK FOR PARAMETER
      F65B  3A 5E D0                      LDA     ABUF+4         GET 2ND PARAMETER
      F65E  B7                            ORA     A              CHECK FOR PARAMETERS
      F65F  C2 68 F6                      JNZ     ASM4
      F662  2A 66 D0                      LHLD    BBUF           FETCH 1ST PARAMETER
      F665  22 68 D0                      SHLD    BBUF+2         STORE INTO 2ND PARAMETER
      F668  3A 80 D0           ASM4       LDA     IBUF+4         FETCH INPUT CHARACTER
      F66B  FE 45                         CPI     'E'            ERROR ONLY INDICATOR
      F66D  C2 71 F6                      JNZ     ASM5
      F670  AF                            XRA     A              SET FOR ONLY ERRORS
      F671  32 6A D0           ASM5       STA     AERR           SET ERROR SWITCH
      F674  AF                            XRA     A              GET A ZERO
      F675  32 74 D0                      STA     NOLA           INITIALIZE LABEL COUNT
      F678  32 70 D0           ASM3       STA     PASI           SET PASS INDICATOR
      F67B  2A 66 D0                      LHLD    BBUF           FETCH ORIGIN
      F67E  22 6E D0                      SHLD    ASPC           INITIALIZE PC
      F681  2A 05 D0                      LHLD    BOFP           GET START OF FILE
      F684  22 4E D0                      SHLD    APNT           SAVE ADDRESS
      F687  2A 4E D0           ASM1       LHLD    APNT           FETCH LINE POINTER
      F68A  31 8E D0                      LXI     SP,AREA+18
      F68D  7E                            MOV     A,M            FETCH CHARACTER
      F68E  FE 01                         CPI     1              END OF FILE
      F690  CA EB F8                      JZ      EASS           JUMP IF END OF FILE
      F693  EB                            XCHG
      F694  13                            INX     D              INCREMENT ADDRESS
      F695  21 8E D0                      LXI     H,OBUF         BLANK START ADDRESS
      F698  3E A7                         MVI     A,>IBUF-5      BLANK END ADDRESS
      F69A  CD 86 F0                      CALL    CLER           BLANK OUT BUFFER
      F69D  0E 0D                         MVI     C,ASCR         STOP CHARACTER
      F69F  CD 34 F5                      CALL    LMOV           MOVE LINE INTO BUFFER
```

not There 00 0000

```
F6A2  71                MOV   M,C        PLACE CR IN BUFFER
F6A3  EB                XCHG
F6A4  22 4E D0          SHLD  APNT       SAVE ADDRESS
F6A7  3A 70 D0          LDA   PASI       FETCH PASS INDICATOR
F6AA  B7                ORA   A          SET FLAGS
F6AB  C2 B4 F6          JNZ   ASM2       JUMP IF PASS 2
F6AE  CD D7 F6          CALL  PAS1
F6B1  C3 87 F6          JMP   ASM1
F6B4  CD 98 F7    ASM2  CALL  PAS2
F6B7  21 8E D0          LXI   H,OBUF     OUTPUT BUFFER ADDRESS
F6BA  CD C0 F6          CALL  AOUT       OUTPUT LINE
F6BD  C3 87 F6          JMP   ASM1
                  *
                  * THIS ROUTINE IS USED TO OUTPUT THE LISTING FOR AN ASSEMBLY
                  * IT CHECKS WHETHER ALL LINES ARE PRINTED OR ONLY THOSE
                  * WITH ERRORS DEPENDING UPON THE ERROR SWITCH
                  *
F6C0  3A 6A D0    AOUT  LDA   AERR       FETCH ERROR SWITCH
F6C3  B7                ORA   A          SET FLAGS
F6C4  C2 CD F6          JNZ   AOU1       OUTPUT ALL LINES
F6C7  3A A0 D0    AOU2  LDA   OBUF+18    FETCH ERROR INDICATOR
F6CA  FE 20             CPI   A' '       CHECK FOR AN ERROR
F6CC  C8                RZ               RETURN IF NO ERROR
F6CD  21 8E D0    AOU1  LXI   H,OBUF     OUTPUT BUFFER ADDRESS
F6D0  CD A6 F0          CALL  CRLF
F6D3  CD 13 F2          CALL  SCRN       OUTPUT LINE
F6D6  C9                RET
                  *
                  * PASS 1 OF ASSEMBLER.  USED TO FORM SYMBOL TABLE
                  *
F6D7  CD FE F0    PAS1  CALL  ZBUF       CLEAR BUFFER
F6DA  32 70 D0          STA   PASI       SET FOR PASS 1
F6DD  21 AC D0          LXI   H,IBUF     INITIALIZE LINE POINTER
F6E0  22 72 D0          SHLD  PNTR       SAVE ADDRESS
F6E3  7E                MOV   A,M        FETCH CHARACTER
F6E4  FE 20             CPI   A' '       CHECK FOR A BLANK
F6E6  CA 19 F7          JZ    OPC        JUMP IF NO LABEL
F6E9  FE 2A             CPI   A'*'       CHECK FOR COMMENT
F6EB  C8                RZ               RETURN IF COMMENT
                  *
                  * PROCESS LABEL
                  *
F6EC  CD 0A FB          CALL  SLAB       GET AND CHECK LABEL
F6EF  DA C9 FA          JC    OPS        ERROR IN LABEL
F6F2  CA AA FC          JZ    ERRD       DUPLICATE LABEL
F6F5  CD 30 F7          CALL  LCHK       CHECK CHARACTER AFTER LABEL
F6F8  C2 C9 FA          JNZ   OPS        ERROR IF NO BLANK
F6FB  0E 05             MVI   C,LLAB     LENGTH OF LABELS
F6FD  21 5A D0          LXI   H,ABUF     SET BUFFER ADDRESS
F700  7E          MLAB  MOV   A,M        FETCH NEXT CHARACTER
F701  12                STAX  D          STORE IN SYMBOL TABLE
F702  13                INX   D
F703  23                INX   H
F704  0D                DCR   C          DECREMENT COUNT
```

```
F705  C2 00 F7          JNZ   MLAB
F708  EB                XCHG
F709  22 6C D0          SHLD  TABA       SAVE TABLE ADDRESS FOR EQU
F70C  3A 6F D0          LDA   ASPC+1     FETCH PC (HIGH)
F70F  77                MOV   M,A        STORE IN TABLE
F710  23                INX   H
F711  3A 6E D0          LDA   ASPC       FETCH PC (LOW)
F714  77                MOV   M,A        STORE IN TABLE
F715  21 74 D0          LXI   H,NOLA
F718  34                INR   M          INCREMENT NUMBER OF LABELS
                  *
                  * PROCESS OPCODE
                  *
F719  CD FE F0    OPC   CALL  ZBUF       ZERO WORKING BUFFER
F71C  CD F7 F8          CALL  SBLK       SCAN TO OPCODE
F71F  DA F0 FA          JC    OERR       FOUND CARRIAGE RETURN
F722  CD 58 FB          CALL  ALPS       PLACE OPCODE IN BUFFER
F725  FE 20             CPI   A' '       CHECK FOR BLANK AFTER OPCODE
F727  DA 4F FA          JC    OPCD       CR AFTER OPCODE
F72A  C2 F0 FA          JNZ   OERR       ERROR IF NO BLANK
F72D  C3 4F FA          JMP   OPCD       CHECK OPCODE
                  * THIS ROUTINE CHECKS THE CHARACTER AFTER A LABEL FOR A BLANK
                  * OR A COLON.
                  *
F730  2A 72 D0    LCHK  LHLD  PNTR
F733  7E                MOV   A,M        GET CHARACTER AFTER LABEL
F734  FE 20             CPI   A' '       CHECK FOR A BLANK
F736  C8                RZ               RETURN IF A BLANK
F737  FE 3A             CPI   A':'       CHECK FOR A COLON
F739  C0                RNZ
F73A  23                INX   H
F73B  22 72 D0          SHLD  PNTR       SAVE POINTER
F73E  C9                RET
                  *
                  * PROCESS ANY PSEUDO OPS THAT NEED TO BE IN PASS 1
                  *
F73F  CD F7 F8    PSU1  CALL  SBLK       SCAN TO OPERAND
F742  1A                LDAX  D          FETCH VALUE
F743  B7                ORA   A          SET FLAGS
F744  CA 5B F7          JZ    ORG1       ORG OPCODE
F747  FA 8D F7          JM    DAT1       DATA STATEMENT
F74A  E2 70 F7          JPO   EQU1       EQU OPCODE
F74D  FE 05             CPI   5
F74F  DA 85 F7          JC    RES1       RES OPCODE
F752  C2 E6 F8          JNZ   EASS       JUMP IF END
                  * DO DW PSEUDO/OP
F755  0E 02       ACO1  MVI   C,2        2 BYTE INSTRUCTION
F757  AF                XRA   A          GET A ZERO
F758  C3 DF FA          JMP   OCN1       ADD VALUE TO PROGRAM COUNTER
                  * DO ORG PSEUDO-OP
F75B  CD 7A FB    ORG1  CALL  ASCN       GET OPERAND
F75E  3A A0 D0          LDA   OBUF+18    FETCH ERROR INDICATOR
F761  FE 20             CPI   A' '       CHECK FOR AN ERROR
F763  C0                RNZ              IF ERROR DON'T CHANGE PC
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1          PAGE   25

      F764  22 6E D0                      SHLD      ASPC                STORE NEW ORIGIN
      F767  3A AC D0                      LDA       IBUF                GET FIRST CHARACTER
      F76A  FE 20                         CPI       A' '                CHECK FOR LABEL
      F76C  C8                            RZ                            NO LABEL
      F76D  C3 78 F7                      JMP       EQUS                CHANGE LABEL VALUE
                                * DO EQU PSEUDO-OP
      F770  CD 7A F8            EQU1      CALL      ASCN                GET OPERAND
      F773  3A AC D0                      LDA       IBUF                FETCH 1ST CHARACTER
      F776  FE 20                         CPI       A' '                CHECK FOR LABEL
      F778  CA 82 FC                      JZ        ERRM                MISSING LABEL
      F77B  EB                  EQUS      XCHG
      F77C  2A 6C D0                      LHLD      TABA                SYMBOL TABLE ADDRESS
      F77F  72                            MOV       M,D                 STORE LABEL VALUE
      F780  23                            INX       H
      F781  73                            MOV       M,E
      F782  C3 C7 F6                      JMP       AOU2                OUTPUT IF ERROR
                                * DO DS PSEUDO-OP
      F785  CD 7A F8            RES1      CALL      ASCN                GET OPERAND
      F788  44                            MOV       B,H
      F789  4D                            MOV       C,L
      F78A  C3 E7 F7                      JMP       RES21               ADD VALUE TO PROGRAM COUNTER
                                *
                                * DO DB PSEUDO-OP
      F78D  C3 6E FA            DAT1      JMP       OP2
                                *
                                * PERFORM PASS 2 OF THE ASSEMBLER
                                *
      F790  21 8E D0            PAS2      LXI       H,OBUF              SET OUTPUT BUFFER ADDRESS
      F793  3A 6F D0                      LDA       ASPC+1              FETCH PC (HIGH)
      F796  CD 22 F2                      CALL      BINH+3              CONVERT FOR OUTPUT
      F799  23                            INX       H
      F79A  3A 6E D0                      LDA       ASPC                FETCH PC (LOW)
      F79D  CD 22 F2                      CALL      BINH+3              CONVERT FOR OUTPUT
      F7A0  22 7A D0                      SHLD      OIND                SAVE OUTPUT ADDRESS
      F7A3  CD FE F0                      CALL      ZBUF                CLEAR BUFFER
      F7A6  21 AC D0                      LXI       H,IBUF              INITIALIZE LINE POINTER
      F7A9  22 72 D0            PABL      SHLD      PNTR                SAVE POINTER
      F7AC  7E                            MOV       A,M                 FETCH FIRST CHARACTER
      F7AD  FE 20                         CPI       A' '                CHECK FOR LABEL
      F7AF  CA 19 F7                      JZ        OPC                 GET OPCODE
      F7B2  FE 2A                         CPI       A'*'                CHECK FOR COMMENT
      F7B4  C8                            RZ                            RETURN IF COMMENT
      F7B5  CD 0A F8                      CALL      SLAB                SCAN OFF LABEL
      F7B8  DA A5 FC                      JC        ERRL                ERROR IN LABEL
      F7BB  CD 30 F7                      CALL      LCHK                CHECK FOR A BLANK OR COLON
      F7BE  C2 A5 FC                      JNZ       ERRL                ERROR IF NOT A BLANK
      F7C1  C3 19 F7                      JMP       OPC
                                *
                                *
                                *
                                * PROCESS PSEUDO OPS FOR PASS 2
      F7C4  1A                  PSU2      LDAX      D
      F7C5  B7                            ORA       A                   SET FLAGS
      F7C6  CA F4 F7                      JZ        ORG2                ORG OPCODE
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1          PAGE   26

      F7C9  FA EB F7                      JM        DAT2                DATA OPCODE
      F7CC  E0                            RPO                           RETURN IF EQU
      F7CD  FE 05                         CPI       5
      F7CF  DA DB F7                      JC        RES2                RES OPCODE
      F7D2  C2 E8 F8                      JNZ       EASS                END OPCODE
                                * DO DW PSEUDO-OP
      F7D5  CD C7 F8            ACO2      CALL      TYS6                GET VALUE
      F7D8  C3 55 F7                      JMP       ACO1
                                * DO DS PSEUDO-OP
      F7DB  CD 77 F8            RES2      CALL      ASBL                GET OPERAND
      F7DE  44                            MOV       B,H
      F7DF  4D                            MOV       C,L
      F7E0  2A 68 D0                      LHLD      BBUF+2              FETCH STORAGE COUNTER
      F7E3  09                            DAD       B                   ADD VALUE
      F7E4  22 68 D0                      SHLD      BBUF+2              SAVE
      F7E7  AF                  RES21     XRA       A                   GET A ZERO
      F7E8  C3 E2 FA                      JMP       OCN2
                                * DO DB PSEUDO-OP
      F7EB  CD 86 F8            DAT2      CALL      TYS5                GET OPERAND
      F7EE  AF                            XRA       A                   GET A ZERO
      F7EF  0E 01                         MVI       C,1                 BYTE COUNT
      F7F1  C3 DF FA                      JMP       OCN1
                                * DO ORG PSEUDO-OP
      F7F4  CD 77 F8            ORG2      CALL      ASBL                GET NEW ORIGIN
      F7F7  3A A0 D0                      LDA       OBUF+18             GET ERROR INDICATOR
      F7FA  FE 20                         CPI       A' '                CHECK FOR AN ERROR
      F7FC  C0                            RNZ                           DON'T MODIFY PC IF ERROR
      F7FD  EB                            XCHG
      F7FE  2A 6E D0                      LHLD      ASPC                FETCH PC
      F801  EB                            XCHG
      F802  22 6E D0                      SHLD      ASPC                STORE NEW PC
      F805  7D                            MOV       A,L
      F806  93                            SUB       E                   FORM DIFFERENCE OF ORIGINS
      F807  5F                            MOV       E,A
      F808  7C                            MOV       A,H
      F809  9A                            SBB       D
      F80A  57                            MOV       D,A
      F80B  2A 66 D0                      LHLD      BBUF+2              FETCH STORAGE POINTER
      F80E  19                            DAD       D                   MODIFY
      F80F  22 68 D0                      SHLD      BBUF+2              SAVE
      F812  C9                            RET
                                *
                                * PROCESS 1 BYTE INSTRUCTIONS WITHOUT OPERANDS
                                *
      F813  CD D4 F8            TYP1      CALL      ASTO                STORE VALUE IN MEMORY
      F816  C9                            RET
                                *
                                * PROCESS STAX AND LDAX INSTRUCTIONS
                                *
      F817  CD 77 F8            TYP2      CALL      ASBL                FETCH OPERAND
      F81A  C4 64 FC                      CNZ       ERRR                ILLEGAL REGISTER
      F81D  7D                            MOV       A,L                 GET LOW ORDER OPERAND
      F81E  B7                            ORA       A                   SET FLAGS
      F81F  CA 3B F8                      JZ        TY31                OPERAND = 0
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR              MICROTEC MACRO ASSEMBLER VER 3.1              PAGE  27

1405  F822  FE 02                               CPI     2              OPERAND = 2
1406  F824  C4 64 FC                            CNZ     ERRR           ILLEGAL REGISTER
1407  F827  C3 3B F8                            JMP     TY31
1408                                    *
1409                                    * PROCESS PUSH,POP,INX,DCX,DAD INSTRUCTIONS
1410                                    *
1411  F82A  CD 77 FB                    TYP3    CALL    ASBL           FETCH OPERAND
1412  F82D  C4 64 FC                            CNZ     ERRR           ILLEGAL REGISTER
1413  F830  7D                                  MOV     A,L            GET LOW ORDER OPERAND
1414  F831  0F                                  RRC                    CHECK LOW ORDER BIT
1415  F832  DC 64 FC                            CC      ERRR           ILLEGAL REGISTER
1416  F835  17                                  RAL                    RESTORE OPERAND
1417  F836  FE 08                               CPI     8
1418  F838  D4 64 FC                            CNC     ERRR           ILLEGAL REGISTER
1419  F83B  07                          TY31    RLC                    MULTIPLY BY 8
1420  F83C  17                                  RAL
1421  F83D  17                                  RAL
1422  F83E  47                          TY32    MOV     B,A
1423  F83F  1A                                  LDAX    D              FETCH OPCODE BASE
1424  F840  80                                  ADD     B              FORM OPCODE
1425  F841  FE 76                               CPI     118            CHECK FOR MOV M,M
1426  F843  CC 64 FC                            CZ      ERRR           ILLEGAL REGISTER
1427  F846  C3 13 F8                            JMP     TYP1
1428                                    *
1429                                    * PROCESS ACCUMULATOR, INR,DCR,MOV,RST INSTRUCTIONS
1430                                    *
1431  F849  CD 77 FB                    TYP4    CALL    ASBL           FETCH OPERAND
1432  F84C  C4 64 FC                            CNZ     ERRR           ILLEGAL REGISTER
1433  F84F  7D                                  MOV     A,L            GET LOW ORDER OPERAND
1434  F850  FE 08                               CPI     8
1435  F852  D4 64 FC                            CNC     ERRR           ILLEGAL REGISTER
1436  F855  1A                                  LDAX    D              FETCH OPCODE BASE
1437  F856  FE 40                               CPI     64             CHECK FOR MOV INSTRUCTION
1438  F858  CA 67 F8                            JZ      TY41
1439  F85B  FE C7                               CPI     199
1440  F85D  7D                                  MOV     A,L
1441  F85E  CA 3B F8                            JZ      TY31           RST INSTRUCTION
1442  F861  FA 3E F8                            JM      TY32           ACCUMULATOR INSTRUCTION
1443  F864  C3 3B F8                            JMP     TY31           INR,DCR
1444                                    * PROCESS MOV INSTRUCTION
1445  F867  29                          TY41    DAD     H              MULTIPLY OPERAND BY 8
1446  F868  29                                  DAD     H
1447  F869  29                                  DAD     H
1448  F86A  85                                  ADD     L              FORM OPCODE
1449  F86B  12                                  STAX    D              SAVE OPCODE
1450  F86C  CD A5 F8                            CALL    MPNT           INCREMENT POINTER
1451  F86F  CD 7A FB                            CALL    ASCN           GET NEXT OPERAND
1452  F872  C4 64 FC                            CNZ     ERRR           ILLEGAL REGISTER
1453  F875  7D                                  MOV     A,L            FETCH LOW ORDER OPERAND
1454  F876  FE 08                               CPI     8
1455  F878  D4 64 FC                            CNC     ERRR           ILLEGAL REGISTER
1456  F87B  C3 3E F8                            JMP     TY32
1457                                    *
1458                                    * PROCESS IMMEDIATE INSTRUCTIONS
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR              MICROTEC MACRO ASSEMBLER VER 3.1              PAGE  28

1459                                    * IMMEDIATE BYTE CAN BE BETWEEN -256 AND +255
1460                                    * MVI INSTRUCTION IS A SPECIAL CASE AND CONTAINS 2 ARGUMENTS
1461                                    * IN OPERAND
1462  F87E  FE 06                       TYP5    CPI     6              CHECK FOR MVI INSTRUCTION
1463  F880  CC 93 F8                            CZ      TY56
1464  F883  CD D4 F8                            CALL    ASTO           STORE OBJECT BYTE
1465  F886  CD 77 FB                    TY55    CALL    ASBL           GET IMMEDIATE ARGUMENT
1466  F889  3C                                  INR     A
1467  F88A  FE 02                               CPI     2              CHECK OPERAND FOR RANGE
1468  F88C  D4 7D FC                            CNC     ERRV           OPERAND OUT OF RANGE
1469  F88F  7D                                  MOV     A,L
1470  F890  C3 13 F8                            JMP     TYP1
1471                                    *
1472                                    * FETCH 1ST ARGUMENT FOR MVI AND LXI INSTRUCTIONS
1473                                    *
1474  F893  CD 77 FB                    TY56    CALL    ASBL           FETCH ARGUMENT
1475  F896  C4 64 FC                            CNZ     ERRR           ILLEGAL REGISTER
1476  F899  7D                                  MOV     A,L            GET LOW ORDER ARGUMENT
1477  F89A  FE 08                               CPI     8
1478  F89C  D4 64 FC                            CNC     ERRR           ILLEGAL REGISTER
1479  F89F  29                                  DAD     H              MULTIPLY BY 8
1480  F8A0  29                                  DAD     H
1481  F8A1  29                                  DAD     H
1482  F8A2  1A                                  LDAX    D              FETCH OPCODE BASE
1483  F8A3  85                                  ADD     L              FORM OPCODE
1484  F8A4  5F                                  MOV     E,A            SAVE OBJECT BYTE
1485  F8A5  2A 72 D0                    MPNT    LHLD    PNTR           FETCH POINTER
1486  F8A8  7E                                  MOV     A,M            FETCH CHARACTER
1487  F8A9  FE 2C                               CPI     ','            CHECK FOR COMMA
1488  F8AB  23                                  INX     H              INCREMENT POINTER
1489  F8AC  22 72 D0                            SHLD    PNTR
1490  F8AF  C2 6D FC                            JNZ     ERRS           SYNTAX ERROR IF NO COMMA
1491  F8B2  7B                                  MOV     A,E            GET OBJECT BYTE
1492  F8B3  C9                                  RET
1493                                    *
1494                                    * PROCESS 3 BYTE INSTRUCTIONS
1495                                    * LXI INSTRUCTION IS A SPECIAL CASE
1496                                    *
1497  F8B4  FE 01                       TYP6    CPI     1              CHECK FOR LXI INSTRUCTION
1498  F8B6  C2 C4 F8                            JNZ     TY6            JUMP IF NOT LXI
1499  F8B9  CD 93 F8                            CALL    TY56           GET REGISTER
1500  F8BC  E6 08                               ANI     H'08'          CHECK FOR ILLEGAL REGISTER
1501  F8BE  C4 64 FC                            CNZ     ERRR           REGISTER ERROR
1502  F8C1  7B                                  MOV     A,E            GET OPCODE
1503  F8C2  E6 F7                               ANI     H'F7'          CLEAR BIT IN ERROR
1504  F8C4  CD D4 F8                    TY6     CALL    ASTO           STORE OBJECT BYTE
1505  F8C7  CD 77 FB                    TY56    CALL    ASBL           FETCH OPERAND
1506  F8CA  7D                                  MOV     A,L
1507  F8CB  54                                  MOV     D,H
1508  F8CC  CD D4 F8                            CALL    ASTO           STORE 2ND BYTE
1509  F8CF  7A                                  MOV     A,D
1510  F8D0  C3 13 F8                            JMP     TYP1
1511  F8D3  C9                                  RET
1512                                    *
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR             MICROTEC MACRO ASSEMBLER VER 3.1            PAGE   29

1513                                  * THIS ROUTINE IS USED TO STORE OBJECT CODE PRODUCED
1514                                  * BY THE ASSEMBLER DURING PASS 2 INTO MEMORY
1515                                  *
1516  F8D4  2A 68 D0                  ASTO     LHLD      BBUF+2              FETCH STORAGE ADDRESS
1517  F8D7  77                                 MOV       M,A                 STORE OBJECT BYTE
1518  F8D8  23                                 INX       H                   INCREMENT LOCATION
1519  F8D9  22 68 D0                           SHLD      BBUF+2
1520  F8DC  2A 7A D0                           LHLD      OIND                FETCH OUTPUT ADDRESS
1521  F8DF  23                                 INX       H
1522  F8E0  23                                 INX       H
1523  F8E1  CD 22 F2                           CALL      BINH+3              CONVERT OBJECT BYTE
1524  F8E4  22 7A D0                           SHLD      OIND
1525  F8E7  C9                                 RET
1526                                  *
1527                                  * GET HERE WHEN END PSEUDO-OP IS FOUND OR WHEN END OF FILE
1528                                  * OCCURS IN SOURCE STATEMENTS.  CONTROL IS SET FOR EIGHER PASS 2
1529                                  * OR ASSEMBLEY TERMINATES IF FINISHED.
1530                                  *
1531  F8E8  3A 70 D0                  EASS     LDA       PASI                FETCH PASS INDICATOR
1532  F8EB  B7                                 ORA       A                   SET FLAGS
1533  F8EC  C2 1F F0                           JNZ       EOR                 JUMP IF FINISHED
1534  F8EF  CD A6 F0                           CALL      CRLF
1535  F8F2  3E 01                              MVI       A,1                 PASS INDICATOR FOR 2ND PASS
1536  F8F4  C3 78 F6                           JMP       ASM3                DO 2ND PASS
1537                                  *
1538                                  *
1539                                  * THIS ROUTINE SCANS THROUGH A CHARACTER STRING UNTIL
1540                                  * THE FIRST NON BLANK CHARACTER IS FOUND
1541                                  *
1542                                  * ON RETURN CARRY = 1 INDICATES A CR AS FIRST NON BLANK CHARACTER
1543                                  *
1544  F8F7  2A 72 D0                  SBLK     LHLD      PNTR                FETCH ADDRESS
1545  F8FA  7E                        SBL1     MOV       A,M                 FETCH CHARACTER
1546  F8FB  FE 20                              CPI       A' '                CHECK FOR A BLANK
1547  F8FD  C0                                 RNZ                           RETURN IF NON BLANK
1548  F8FE  23                        SBL2     INX       H                   INCREMENT
1549  F8FF  22 72 D0                           SHLD      PNTR                SAVE POINTER
1550  F902  C3 FA F8                           JMP       SBL1
1551                                  *
1552                                  *
1553                                  * THIS ROUTINE IS USED TO CHECK THE CONDITION CODE MNEMONICS
1554                                  * FOR CONDITIONAL JUMPS, CALLS, AND RETURNS.
1555                                  *
1556  F905  21 58 D0                  COND     LXI       H,ABUF+1
1557  F908  22 50 D0                           SHLD      ADDS
1558  F90B  06 02                              MVI       B,2                 2 CHARACTERS
1559  F90D  CD 3A FA                           CALL      COPC
1560  F910  C9                                 RET
1561                                  *
1562                                  *
1563                                  * THE FOLLOWING IS THE OPCODE TABLE
1564                                  *
1565                                  *
1566  F911  4F 52 47                  OTAB     DATA      A'ORG'
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR             MICROTEC MACRO ASSEMBLER VER 3.1            PAGE   30

1567  F914  00                                 DATA      0
1568  F915  00                                 DATA      0
1569  F916  45 51 55                           DATA      A'EQU'
1570  F919  00                                 DATA      0
1571  F91A  01                                 DATA      1
1572  F91B  44 42                              DATA      A'DB'
1573  F91D  00                                 DATA      0
1574  F91E  00                                 DATA      0
1575  F91F  FF                                 DATA      -1
1576  F920  44 53                              DATA      A'DS'
1577  F922  00                                 DATA      0
1578  F923  00                                 DATA      0
1579  F924  03                                 DATA      3
1580  F925  44 57                              DATA      A'DW'
1581  F927  00                                 DATA      0
1582  F928  00                                 DATA      0
1583  F929  05                                 DATA      5
1584  F92A  45 4E 44                           DATA      A'END'
1585  F92D  00                                 DATA      0
1586  F92E  06                                 DATA      6
1587  F92F  00                                 DATA      0
1588  F930  48 4C 54                           DATA      A'HLT'
1589  F933  76                                 DATA      118
1590  F934  52 4C 43                           DATA      A'RLC'
1591  F937  07                                 DATA      7
1592  F938  52 52 43                           DATA      A'RRC'
1593  F93B  0F                                 DATA      15
1594  F93C  52 41 4C                           DATA      A'RAL'
1595  F93F  17                                 DATA      23
1596  F940  52 41 52                           DATA      A'RAR'
1597  F943  1F                                 DATA      31
1598  F944  52 45 54                           DATA      A'RET'
1599  F947  C9                                 DATA      201
1600  F948  43 4D 41                           DATA      A'CMA'
1601  F94B  2F                                 DATA      47
1602  F94C  53 54 43                           DATA      A'STC'
1603  F94F  37                                 DATA      55
1604  F950  44 41 41                           DATA      A'DAA'
1605  F953  27                                 DATA      39
1606  F954  43 4D 43                           DATA      A'CMC'
1607  F957  3F                                 DATA      63
1608  F958  45 49                              DATA      A'EI'
1609  F95A  00                                 DATA      0
1610  F95B  FB                                 DATA      251
1611  F95C  44 49                              DATA      A'DI'
1612  F95E  00                                 DATA      0
1613  F95F  F3                                 DATA      243
1614  F960  4E 4F 50                           DATA      A'NOP'
1615  F963  00                                 DATA      0
1616  F964  00                                 DATA      0
1617  F965  58 43 48 47                        DATA      A'XCHG'
1618  F969  EB                                 DATA      235
1619  F96A  58 54 48 4C                        DATA      A'XTHL'
1620  F96E  E3                                 DATA      227
```

4E memory problem

HNT?

43

RCL?

have been changed to lower case in hex listing

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1          PAGE   31

1621  F96F  53 50 48 4C                       DATA        A'SPHL'
1622  F973  F9                                DATA        249
1623  F974  50 43 48 4C                       DATA        A'PCHL'
1624  F978  DF         E9                     DATA        223       351
1625  F979  00                                DATA        0
1626  F97A  53 54 41 58                       DATA        A'STAX'
1627  F97E  02                                DATA        2
1628  F97F  4C 44 41 58                       DATA        A'LDAX'
1629  F983  0A                                DATA        10
1630  F984  00                                DATA        0
1631  F985  50 55 53 48                       DATA        A'PUSH'
1632  F989  C5                                DATA        197
1633  F98A  50 4F 50                          DATA        A'POP'
1634  F98D  00                                DATA        0
1635  F98E  C1                                DATA        193
1636  F98F  49 4E 58                          DATA        A'INX'
1637  F992  00                                DATA        0
1638  F993  03                                DATA        3
1639  F994  44 43 58                          DATA        A'DCX'
1640  F997  00                                DATA        0
1641  F998  0B                                DATA        11
1642  F999  44 41 44                          DATA        A'DAD'
1643  F99C  00                                DATA        0
1644  F99D  09                                DATA        9
1645  F99E  00                                DATA        0
1646  F99F  49 4E 52                          DATA        A'INR'
1647  F9A2  04                                DATA        4
1648  F9A3  44 43 52                          DATA        A'DCR'
1649  F9A6  05                                DATA        5
1650  F9A7  4D 4F 56                          DATA        A'MOV'
1651  F9AA  40                                DATA        64
1652  F9AB  41 44 44                          DATA        A'ADD'
1653  F9AE  80                                DATA        128
1654  F9AF  41 44 43                          DATA        A'ADC'
1655  F9B2  88                                DATA        136
1656  F9B3  53 55 42                          DATA        A'SUB'
1657  F9B6  90                                DATA        144
1658  F9B7  53 42 42                          DATA        A'SBB'
1659  F9BA  98                                DATA        152
1660  F9BB  41 4E 41                          DATA        A'ANA'
1661  F9BE  A0                                DATA        160
1662  F9BF  58 52 41                          DATA        A'XRA'
1663  F9C2  A8                                DATA        168
1664  F9C3  4F 52 41                          DATA        A'ORA'
1665  F9C6  B0                                DATA        176
1666  F9C7  43 4D 50                          DATA        A'CMP'
1667  F9CA  B8                                DATA        184
1668  F9CB  52 53 54                          DATA        A'RST'
1669  F9CE  C7                                DATA        199
1670  F9CF  00                                DATA        0
1671  F9D0  41 44 49                          DATA        A'ADI'
1672  F9D3  C6                                DATA        198
1673  F9D4  41 43 49                          DATA        A'ACI'
1674  F9D7  CE                                DATA        206
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1          PAGE   32

1675  F9D8  53 55 49                          DATA        A'SUI'
1676  F9DB  D6                                DATA        214
1677  F9DC  53 42 49                          DATA        A'SBI'
1678  F9DF  DE                                DATA        222
1679  F9E0  41 4E 49                          DATA        A'ANI'
1680  F9E3  E6                                DATA        230
1681  F9E4  58 52 49                          DATA        A'XRI'
1682  F9E7  EE                                DATA        238
1683  F9E8  4F 52 49                          DATA        A'ORI'
1684  F9EB  F6                                DATA        246
1685  F9EC  43 50 49                          DATA        A'CPI'
1686  F9EF  FE                                DATA        254
1687  F9F0  49 4E                             DATA        A'IN'
1688  F9F2  00                                DATA        0
1689  F9F3  DB                                DATA        219
1690  F9F4  4F 55 54                          DATA        A'OUT'
1691  F9F7  D3                                DATA        211
1692  F9F8  4D 56 49                          DATA        A'MVI'
1693  F9FB  06                                DATA        6
1694  F9FC  00                                DATA        0
1695  F9FD  4A 4D 50                          DATA        A'JMP'
1696  FA00  00                                DATA        0
1697  FA01  C3                                DATA        195
1698  FA02  43 41 4C 4C                       DATA        A'CALL'
1699  FA06  CD                                DATA        205
1700  FA07  4C 58 49                          DATA        A'LXI'
1701  FA0A  00                                DATA        0
1702  FA0B  01                                DATA        1
1703  FA0C  4C 44 41                          DATA        A'LDA'
1704  FA0F  00                                DATA        0
1705  FA10  3A                                DATA        58
1706  FA11  53 54 41                          DATA        A'STA'
1707  FA14  00                                DATA        0
1708  FA15  32                                DATA        50
1709  FA16  53 48 4C 44                       DATA        A'SHLD'
1710  FA1A  22                                DATA        34
1711  FA1B  4C 48 4C 44                       DATA        A'LHLD'
1712  FA1F  2A                                DATA        42
1713  FA20  00                                DATA        0
1714                              * CONDITION CODE TABLE
1715  FA21  4E 5A                             DATA        A'NZ'
1716  FA23  00                                DATA        0
1717  FA24  5A                                DATA        A'Z'
1718  FA25  00                                DATA        0
1719  FA26  08                                DATA        8
1720  FA27  4E 43                             DATA        A'NC'
1721  FA29  10                                DATA        16
1722  FA2A  43                                DATA        A'C'
1723  FA2B  00                                DATA        0
1724  FA2C  18                                DATA        24
1725  FA2D  50 4F                             DATA        A'PO'
1726  FA2F  20                                DATA        32
1727  FA30  50 45                             DATA        A'PE'
1728  FA32  28                                DATA        40
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1              PAGE   33

1729  FA33  50                              DATA      A'P'
1730  FA34  00                              DATA      0
1731  FA35  30                              DATA      48
1732  FA36  40                              DATA      A'H'
1733  FA37  00                              DATA      0
1734  FA38  38                              DATA      56
1735  FA39  00                              DATA      0
1736                                 *
1737                                 * THIS ROUTINE IS USED TO CHECK A GIVEN OPCODE AGAINST THE LEGAL
1738                                 * OPCODES CONTAINED IN THE OPCODE TABLE
1739                                 *
1740  FA3A  2A 50 D0                 COPC   LHLD      ADDS
1741  FA3D  1A                              LDAX      D                   FETCH CHARACTER
1742  FA3E  B7                              ORA       A                   SET FLAGS
1743  FA3F  CA 4C FA                        JZ        COP1                JUMP IF TERMINATION CHARACTER
1744  FA42  48                              MOV       C,B
1745  FA43  CD EB F0                        CALL      SEAR                COMPARE STRINGS
1746  FA46  1A                              LDAX      D
1747  FA47  C8                              RZ                            RETURN IF MATCH
1748  FA48  13                              INX       D                   NEXT STRING
1749  FA49  C3 3A FA                        JMP       COPC                CONTINUE SEARCH
1750  FA4C  3C                       COP1   INR       A                   CLEAR ZERO FLAG
1751  FA4D  13                              INX       D                   INCREMENT ADDRESS
1752  FA4E  C9                              RET
1753                                 *
1754                                 *
1755                                 * THIS ROUTINE CHECKS THE LEGAL OPCODES IN BOTH PASS 1 AND
1756                                 * PASS 2.  IN PASS 1 THE PROGRAM COUNTER IS INCREMENTED BY THE
1757                                 * CORRECT NUMBER OF BYTES.  AN ADDRESS IS ALSO SET SO THAT
1758                                 * AN INDEXED JUMP CAN BE MADE TO PROCESS THE OPCODE FOR
1759                                 * PASS2.
1760                                 *
1761                                 *
1762  FA4F  21 5A D0                 OPCD   LXI       H,ABUF              SET ADDRESS
1763  FA52  22 50 D0                        SHLD      ADDS
1764  FA55  11 11 F9                        LXI       D,OTAB              OPCODE TABLE ADDRESS
1765  FA58  06 04                           MVI       B,4                 CHARACTER COUNT
1766  FA5A  CD 3A FA                        CALL      COPC                CHECK OPCODES
1767  FA5D  CA F8 FA                        JZ        PSEU                JUMP IF A PSEUDO-OP
1768  FA60  05                              DCR       B                   3 CHARACTER OPCODES
1769  FA61  CD 3A FA                        CALL      COPC
1770  FA64  CA 68 FA                        JZ        OP1
1771  FA67  04                              INR       B                   4 CHAR OPCODES
1772  FA68  CD 3A FA                        CALL      COPC
1773  FA6B  21 13 F8                 OP1    LXI       H,TYP1              TYPE 1 INSTRUCTIONS
1774  FA6E  0E 01                    OP2    MVI       C,1                 1 BYTE INSTRUCTION
1775  FA70  CA CB FA                        JZ        OCNT
1776                                 *
1777  FA73  CD 3A FA                 OPC2   CALL      COPC                CHECK FOR STAX,LDAX
1778  FA76  21 17 F8                        LXI       H,TYP2
1779  FA79  CA 6E FA                        JZ        OP2
1780  FA7C  CD 3A FA                        CALL      COPC                CHECK FOR PUSH,POP,INX
1781                                 *                                    DCX AND DAD
1782  FA7F  21 2A F8                        LXI       H,TYP3
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                    MICROTEC MACRO ASSEMBLER VER 3.1              PAGE   34

1783  FA82  CA 6E FA                        JZ        OP2
1784  FA85  05                              DCR       B                   3 CHAR OPCODES
1785  FA86  CD 3A FA                        CALL      COPC                ACCUMULATOR INSTRUCTIONS.
1786                                 *                                    INR,DCR,MOV,RST
1787  FA89  21 49 F8                        LXI       H,TYP4
1788  FA8C  CA 6E FA                        JZ        OP2
1789                                 *
1790  FA8F  CD 3A FA                 OPC3   CALL      COPC                IMMEDIATE INSTRUCTIONS
1791  FA92  21 7E F8                        LXI       H,TYP5
1792  FA95  0E 02                           MVI       C,2                 2 BYTE INSTRUCTIONS
1793  FA97  CA CB FA                        JZ        OCNT
1794  FA9A  04                              INR       B                   4 CHAR OPCODES
1795  FA9B  CD 3A FA                        CALL      COPC                JMP,CALL,LXI,LDA,STA,
1796                                 *                                    LHLD,SHLD OPCODES
1797  FA9E  CA C6 FA                        JZ        OP4
1798  FAA1  CD 05 F9                        CALL      COND                CONDITIONAL INSTRUCTIONS
1799  FAA4  C2 F0 FA                        JNZ       OERR                ILLEGAL OPCODE
1800  FAA7  C6 C0                           ADI       192                 ADD BASE VALUE OF RETURN
1801  FAA9  57                              MOV       D,A
1802  FAAA  06 03                           MVI       B,3                 3 CHARACTER OPCODES
1803  FAAC  3A 5A D0                        LDA       ABUF                FETCH FIRST CHARACTER
1804  FAAF  4F                              MOV       C,A                 SAVE CHARACTER
1805  FAB0  FE 52                           CPI       A'R'                CONDITIONAL RETURN
1806  FAB2  7A                              MOV       A,D
1807  FAB3  CA 6B FA                        JZ        OP1
1808  FAB6  79                              MOV       A,C
1809  FAB7  14                              INR       D                   FORM CONDITIONAL JUMP
1810  FAB8  14                              INR       D
1811  FAB9  FE 4A                           CPI       A'J'                CONDITIONAL JUMP
1812  FABB  CA C5 FA                        JZ        OPAD
1813  FABE  FE 43                           CPI       A'C'                CONDITIONAL CALL
1814  FAC0  C2 F0 FA                        JNZ       OERR                ILLEGAL OPCODE
1815  FAC3  14                              INR       D                   FORM CONDITIONAL CALL
1816  FAC4  14                              INR       D
1817  FAC5  7A                       OPAD   MOV       A,D                 GET OPCODE
1818  FAC6  21 B4 F8                 OP4    LXI       H,TYP6
1819  FAC9  0E 03                    OP5    MVI       C,3                 3 BYTE INSTRUCTION
1820  FACB  32 79 D0                 OCNT   STA       TEMP                SAVE OPCODE
1821                                 * CHECK FOR OPCODE ONLY CONTAINING THE CORRECT NUMBER OF CHARACTERS
1822                                 * THUS SAY ADDQ WOULD GIVE AN ERROR
1823  FACE  3E 5A                           MVI       A,>ABUF             LOAD BUFFER ADDRESS
1824  FAD0  80                              ADD       B                   ADD LENGTH OF OPCODE
1825  FAD1  5F                              MOV       E,A
1826  FAD2  3E D0                           MVI       A,<ABUF             LOAD BUFFER ADDRESS
1827  FAD4  CE 00                           ACI       0                   GET HIGH ORDER ADDRESS
1828  FAD6  57                              MOV       D,A
1829  FAD7  1A                              LDAX      D                   FETCH CHARACTER AFTER OPCODE
1830  FAD8  B7                              ORA       A                   IT SHOULD BE ZERO
1831  FAD9  C2 F0 FA                        JNZ       OERR                OPCODE ERROR
1832  FADC  3A 70 D0                        LDA       PASI                FETCH PASS INDICATOR
1833  FADF  06 00                    OCN1   MVI       B,0
1834  FAE1  EB                              XCHG
1835  FAE2  2A 6E D0                 OCN2   LHLD      ASPC                FETCH PROGRAM COUNTER
1836  FAE5  09                              DAD       B                   ADD IN BYTE COUNT
```

```
ADDR  B1 B2 B3 B4 ERROR

FAE6  22 6E D0                SHLD    ASPC          STORE PC
FAE9  B7                      ORA     A             WHICH PASS
FAEA  C8                      RZ                    RETURN IF PASS 1
FAEB  3A 79 D0                LDA     TEMP          FETCH OPCODE
FAEE  EB                      XCHG
FAEF  E9                      PCHL
                      *
FAF0  21 99 FC        OERR    LXI     H,ERRO        SET ERROR ADDRESS
FAF3  0E 03                   MVI     C,3           LEAVE 3 BYTES FOR PATCH
FAF5  C3 DC FA                JMP     OCN1-3
                      *
FAF8  21 5E D0        PSEU    LXI     H,ABUF+4      SET BUFFER ADDRESS
FAFB  7E                      MOV     A,M           FETCH CHARACTER AFTER OPCODE
FAFC  B7                      ORA     A             SHOULD BE A ZERO
FAFD  C2 F0 FA                JNZ     OERR
FB00  3A 70 D0                LDA     PASI          FETCH PASS INDICATOR
FB03  B7                      ORA     A
FB04  CA 3F F7                JZ      PSU1
FB07  C3 C4 F7                JMP     PSU2
                      *
                      *
                      * THIS ROUTINE IS USED TO PROCESS LABELS
                      * IT CHECKS WHETHER A LABEL IS IN THE SYMBOL TABLE OR NOT
                      * ON RETURN Z=1 MEANS A MATCH WAS FOUND AND H,L CONTAIN THE VALUE
                      * ASSOCIATED WITH THE LABEL.  OTHERWISE D,E POINT TO THE NEXT AVAILABLE
                      * LOCATION IN THE TABLE.  THE REGISTER NAMES A,B,C,D,E,H,L,M
                      * ARE PREDEFINED BY THE SYSTEM AND NEED NOT BE ENTERED BY THE USER
                      * ON RETURN C=1 INDICATES A LABEL ERROR
FB0A  FE 41           SLAB    CPI     'A'           CHECK FOR LEGAL CHARACTER
FB0C  D8                      RC                    RETURN IF ILLEGAL CHARACTER
FB0D  FE 5B                   CPI     'Z'+1         CHECK FOR ILLEGAL CHARACTER
FB0F  3F                      CMC
FB10  D8                      RC                    RETURN IF ILLEGAL CHARACTER
FB11  CD 58 FB                CALL    ALPS          PLACE SYMBOL IN BUFFER
FB14  21 5A D0                LXI     H,ABUF        SET BUFFER ADDRESS
FB17  22 50 D0                SHLD    ADDS          SAVE ADDRESS
FB1A  05                      DCR     B             CHECK IF ONE CHARACTER
FB1B  C2 2E FB                JNZ     SLA1
                      * CHECK IF PREDEFINED REGISTER NAME
FB1E  04                      INR     B             SET B=1
FB1F  11 47 FB                LXI     D,RTAB        REGISTER TABLE ADDRESS
FB22  CD 3A FA                CALL    COPC          CHECK NAME OF REGISTER
FB25  C2 2E FB                JNZ     SLA1          NOT A PREDEFINED REGISTER
FB28  6F                      MOV     L,A
FB29  26 00                   MVI     H,0           SET VALUE (HIGH)
FB2B  C3 41 FB                JMP     SLA2
FB2E  3A 74 D0        SLA1    LDA     NOLA          FETCH SYMBOL COUNT
FB31  47                      MOV     B,A
FB32  11 FF D0                LXI     D,SYMT        SET SYMBOL TABLE ADDRESS
FB35  B7                      ORA     A             ARE THERE ANY LABELS
FB36  CA 44 FB                JZ      SLA3          JUMP IF NO LABELS
FB39  3E 05                   MVI     A,LLAB        FETCH LENGTH OF LABEL
FB3B  32 71 D0                STA     NCHR
FB3E  CD D4 F0                CALL    COMS          CHECK TABLE
```

```
ADDR  B1 B2 B3 B4 ERROR

FB41  37              SLA2    STC                   SET CARRY
FB42  3F                      CMC                   CLEAR CARRY
FB43  C9                      RET
FB44  3C              SLA3    INR     A             CLEAR ZERO FLAG
FB45  B7                      ORA     A             CLEAR CARRY
FB46  C9                      RET
                      *
                      * PREDEFINE REGISTER VALUES IN THIS TABLE
                      *
FB47  41              RTAB    DATA    'A'
FB48  07                      DATA    7
FB49  42                      DATA    'B'
FB4A  00                      DATA    0
FB4B  43                      DATA    'C'
FB4C  01                      DATA    1
FB4D  44                      DATA    'D'
FB4E  02                      DATA    2
FB4F  45                      DATA    'E'
FB50  03                      DATA    3
FB51  48                      DATA    'H'
FB52  04                      DATA    4
FB53  4C                      DATA    'L'
FB54  05                      DATA    5
FB55  4D                      DATA    'M'
FB56  06                      DATA    6
FB57  00                      DATA    0             END OF TABLE INDICATOR
                      *
                      *
                      * THIS ROUTINE SCANS THE INPUT LINE AND PLACES THE OPCODES AND
                      * LABELS IN THE BUFFER.  THE SCAN TERMINATES WHEN A CHARACTER
                      * OTHER THAN 0-9 OR A-Z IS FOUND
                      *
FB58  06 00           ALPS    MVI     B,0           SET COUNT
FB5A  12              ALP1    STAX    D             STORE CHARACTER IN BUFFER
FB5B  04                      INR     B             INCREMENT COUNT
FB5C  78                      MOV     A,B           FETCH COUNT
FB5D  FE 0B                   CPI     11            MAXIMUM BUFFER SIZE
FB5F  D0                      RNC                   RETURN IF BUFFER FILLED
FB60  13                      INX     D             INCREMENT BUFFER
FB61  23                      INX     H             INCREMENT INPUT ADDRESS
FB62  22 72 D0                SHLD    PNTR          SAVE LINE POINTER
FB65  7E                      MOV     A,M           FETCH CHARACTER
FB66  FE 30                   CPI     '0'           CHECK FOR LEGAL CHARACTERS
FB68  D8                      RC
FB69  FE 3A                   CPI     '9'+1
FB6B  DA 5A FB                JC      ALP1
FB6E  FE 41                   CPI     'A'
FB70  D8                      RC
FB71  FE 5B                   CPI     'Z'+1
FB73  DA 5A FB                JC      ALP1
FB76  C9                      RET
                      *
                      *
                      * THIS ROUTINE IS USED TO SCAN THROUGH THE INPUT LINE TO
```

```
1945                            * FETCH THE VALUE OF THE OPERAND FIELD.  ON RETURN THE VALUE OF THE
1946                            * OPERAND IS CONTAINED IN REGISTERS H,L.
1947                            *
1948  FB77  CD F7 F8            ASBL     CALL     SBLK          GET FIRST ARGUMENT
1949  FB7A  21 00 00            ASCN     LXI      H,0           GET A ZERO
1950  FB7D  22 76 D0                     SHLD     OPRD          INITIALIZE OPERAND
1951  FB80  24                           INR      H
1952  FB81  22 77 D0                     SHLD     OPRI-1        INITIALIZE OPERAND INDICATOR
1953  FB84  2A 72 D0            NXT1     LHLD     PNTR          FETCH SCAN POINTER
1954  FB87  2B                           DCX      H
1955  FB88  CD FE F0                     CALL     ZBUF          CLEAR BUFFER
1956  FB8B  32 75 D0                     STA      SIGN          ZERO SIGN INDICATOR
1957  FB8E  23                  NXT2     INX      H             INCREMENT POINTER
1958  FB8F  7E                           MOV      A,M           FETCH NEXT CHARACTER
1959  FB90  FE 21                        CPI      A' '+1
1960  FB92  DA 36 FC                     JC       SEND          JUMP IF CR OR BLANK
1961  FB95  FE 2C                        CPI      A','          FIELD SEPARATOR
1962  FB97  CA 36 FC                     JZ       SEND
1963                            * CHECK FOR OPERATORS
1964  FB9A  FE 2B                        CPI      A'+'          CHECK FOR PLUS
1965  FB9C  CA A7 FB                     JZ       ASC1
1966  FB9F  FE 2D                        CPI      A'-'          CHECK FOR MINUS
1967  FBA1  C2 B7 FB                     JNZ      ASC2
1968  FBA4  32 75 D0                     STA      SIGN
1969  FBA7  3A 78 D0            ASC1     LDA      OPRI          FETCH OPERAND INDICATOR
1970  FBAA  FE 02                        CPI      2             CHECK FOR TWO OPERATORS
1971  FBAC  CA 6D FC                     JZ       ERRS          SYNTAX ERROR
1972  FBAF  3E 02                        MVI      A,2
1973  FBB1  32 78 D0                     STA      OPRI          SET INDICATOR
1974  FBB4  C3 8E FB                     JMP      NXT2
1975                            * CHECK FOR OPERANDS
1976  FBB7  4F                  ASC2     MOV      C,A           SAVE CHARACTER
1977  FBB8  3A 78 D0                     LDA      OPRI          GET INDICATOR
1978  FBBB  B7                           ORA      A             CHECK FOR TWO OPERANDS
1979  FBBC  CA 6D FC                     JZ       ERRS          SYNTAX ERROR
1980  FBBF  79                           MOV      A,C
1981  FBC0  FE 24                        CPI      A'$'          LC EXPRESSION
1982  FBC2  C2 CF FB                     JNZ      ASC3
1983  FBC5  23                           INX      H             INCREMENT POINTER
1984  FBC6  22 72 D0                     SHLD     PNTR          SAVE POINTER
1985  FBC9  2A 6E D0                     LHLD     ASPC          FETCH LOCATION COUNTER
1986  FBCC  C3 0B FC                     JMP      AVAL
1987                            * CHECK FOR ASCII CHARACTERS
1988  FBCF  FE 27               ASC3     CPI      A''''         CHECK FOR SINGLE QUOTE
1989  FBD1  C2 FB FB                     JNZ      ASC5          JUMP IF NOT QUOTE
1990  FBD4  11 00 00                     LXI      D,0           GET A ZERO
1991  FBD7  0E 03                        MVI      C,3           CHARACTER COUNT
1992  FBD9  23                  ASC4     INX      H             INCREMENT POINTER
1993  FBDA  22 72 D0                     SHLD     PNTR          SAVE
1994  FBDD  7E                           MOV      A,M           FETCH NEXT CHARACTER
1995  FBDE  FE 0D                        CPI      ASCR          IS IT A CR
1996  FBE0  CA 8B FC                     JZ       ERRA          ARGUMENT ERROR
1997  FBE3  FE 27                        CPI      A''''         IS IT A QUOTE
1998  FBE5  C2 F2 FB                     JNZ      SSTR
```

```
1999  FBE8  23                           INX      H             INCREMENT POINTER
2000  FBE9  22 72 D0                     SHLD     PNTR          SAVE
2001  FBEC  7E                           MOV      A,M           FETCH NEXT CHARACTER
2002  FBED  FE 27                        CPI      A''''         CHECK FOR 2 QUOTES IN A ROW
2003  FBEF  C2 0C FC                     JNZ      AVAL+1        TERMINAL QUOTE
2004  FBF2  0D                  SSTR     DCR      C             CHECK COUNT
2005  FBF3  CA 8B FC                     JZ       ERRA          TOO MANY CHARACTERS
2006  FBF6  53                           MOV      D,E
2007  FBF7  5F                           MOV      E,A           SET CHARACTER IN BUFFER
2008  FBF8  C3 D9 FB                     JMP      ASC4
2009  FBFB  FE 30               ASC5     CPI      A'0'          CHECK FOR NUMERIC
2010  FBFD  DA 8B FC                     JC       ERRA          ILLEGAL CHARACTER
2011  FC00  FE 3A                        CPI      A'9'+1
2012  FC02  D2 2A FC                     JNC      ALAB
2013  FC05  CD 46 FC                     CALL     NUMS          GET NUMERIC VALUE
2014  FC08  DA 8B FC                     JC       ERRA          ARGUMENT ERROR
2015  FC0B  EB                  AVAL     XCHG
2016  FC0C  2A 76 D0                     LHLD     OPRD          FETCH OPERAND
2017  FC0F  AF                           XRA      A             GET A ZERO
2018  FC10  32 78 D0                     STA      OPRI          STORE IN OPERAND INDICATOR
2019  FC13  3A 75 D0                     LDA      SIGN          GET SIGN INDICATOR
2020  FC16  B7                           ORA      A             SET FLAGS
2021  FC17  C2 21 FC                     JNZ      ASUB
2022  FC1A  19                           DAD      D             FORM RESULT
2023  FC1B  22 76 D0            ASC7     SHLD     OPRD          SAVE RESULT
2024  FC1E  C3 84 FB                     JMP      NXT1
2025  FC21  7D                  ASUB     MOV      A,L
2026  FC22  93                           SUB      E
2027  FC23  6F                           MOV      L,A
2028  FC24  7C                           MOV      A,H
2029  FC25  9A                           SBB      D
2030  FC26  67                           MOV      H,A
2031  FC27  C3 1B FC                     JMP      ASC7
2032  FC2A  CD 0A FB            ALAB     CALL     SLAB
2033  FC2D  CA 0B FC                     JZ       AVAL
2034  FC30  DA 8B FC                     JC       ERRA          ILLEGAL SYMBOL
2035  FC33  C3 78 FC                     JMP      ERRU          UNDEFINED SYMBOL
2036                            *
2037                            * GET HERE WHEN TERMINATION CHARACTER IS FOUND.  BLANK,COMMA,CR.
2038                            * CHECK FOR LEADING FIELD SEPARATOR
2039  FC36  3A 78 D0            SEND     LDA      OPRI          FETCH OPERAND INDICATOR
2040  FC39  B7                           ORA      A             SET FLAGS
2041  FC3A  C2 6D FC                     JNZ      ERRS          SYNTAX ERROR
2042  FC3D  2A 76 D0                     LHLD     OPRD
2043  FC40  7C                  SEN1     MOV      A,H           GET HIGH ORDER BYTE
2044  FC41  11 79 D0                     LXI      D,TEMP        SET ADDRESS
2045  FC44  B7                           ORA      A             SET FLAGS
2046  FC45  C9                           RET
2047                            *
2048                            * GET A NUMERIC VALUE WHICH IS EIGHER HEXADECIMAL OR DECIMAL
2049                            * ON RETURN CARRY SET INDICATES AN ERROR
2050                            *
2051  FC46  CD 58 FB            NUMS     CALL     ALPS          GET NUMERIC
2052  FC49  1B                           DCX      D
```

```
LINE  ADDR  B1 B2 B3 B4 EF OR              MICROTEC MACRO ASSEMBLER VER 3.1            PAGE   39

      FC4A  1A                       LDAX    D               GET LAST CHARACTER
      FC4B  01 5A D0                 LXI     B,ABUF          SET BUFFER ADDRESS
      FC4E  FE 48                    CPI     A'H'            IS IT HEXADECIMAL
      FC50  CA 5E FC                 JZ      NUM2
      FC53  FE 44                    CPI     A'D'            IS IT DECIMAL
      FC55  C2 5A FC                 JNZ     NUM1
      FC58  AF                       XRA     A               GET A ZERO
      FC59  12                       STAX    D               CLEAR D FROM BUFFER
      FC5A  CD 9A F1         NUM1    CALL    ADEC            CONVERT DECIMAL VALUE
      FC5D  C9                       RET
      FC5E  AF               NUM2    XRA     A               GET A ZERO
      FC5F  12                       STAX    D               CLEAR H FROM BUFFER
      FC60  CD B4 F1                 CALL    AHEX            CONVERT HEX
      FC63  C9                       RET
                             *
                             * PROCESS REGISTER ERROR
      FC64  3E 52            ERRR    MVI     A,A'R'          GET INDICATOR
      FC66  21 00 00                 LXI     H,0             GET A 0
      FC69  32 A0 D0                 STA     OBUF+18         SET IN OUTPUT RUFFER
      FC6C  C9                       RET
                             * PROCESS SYNTAX ERROR
      FC6D  3E 53            ERRS    MVI     A,A'S'          GET INDICATOR
      FC6F  32 A0 D0                 STA     OBUF+18         STORE IN OUTPUT RUFFER
      FC72  21 00 00                 LXI     H,0             GET A ZERO
      FC75  C3 40 FC                 JMP     SEN1
                             * PROCESS UNDEFINED SYMBOL ERROR
      FC78  3E 55            ERRU    MVI     A,A'U'          GET INDICATOR
      FC7A  C3 6F FC                 JMP     ERRS+2
                             * PROCESS VALUE ERROR
      FC7D  3E 56            ERRV    MVI     A,A'V'          GET INDICATOR
      FC7F  C3 66 FC                 JMP     ERRR+2
                             * PROCESS MISSING LABEL ERROR
      FC82  3E 4D            ERRM    MVI     A,A'M'          GET INDICATOR
      FC84  32 A0 D0                 STA     OBUF+18         STORE IN OUTPUT RUFFER
      FC87  CD CD F6                 CALL    AOU1            DISPLAY ERROR
      FC8A  C9                       RET
                             * PROCESS ARGUMENT ERROR
      FC8B  3E 41            ERRA    MVI     A,A'A'          GET INDICATOR
      FC8D  C3 6F FC                 JMP     ERRS+2
                             * PROCESS OPCODE ERROR
                             * STORE 3 BYTES OF ZERO IN OBJECT CODE TO PROVIDE FOR A PATCH
      FC90  3E 4F            ERRO    MVI     A,A'O'          GET INDICATOR
      FC92  32 A0 D0                 STA     OBUF+18         STORE IN OUTPUT RUFFER
      FC95  3A 70 D0                 LDA     PASI            FETCH PASS INDICATOR
      FC98  B7                       ORA     A               WHICH PASS
      FC99  C8                       RZ                      RETURN IF PASS 1
      FC9A  0E 03                    MVI     C,3             NEED 3 RYTES
      FC9C  AF               ERO1    XRA     A               GET A ZERO
      FC9D  CD 04 F8                 CALL    ASTO            PUT IN LISTING AND MEMORY
      FCA0  0D                       DCR     C
      FCA1  C2 9C FC                 JNZ     ERO1
      FCA4  C9                       RET
                             * PROCESS LABEL ERROR
      FCA5  3E 4C            ERRL    MVI     A,A'L'          GET INDICATOR
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR              MICROTEC MACRO ASSEMBLER VER 3.1            PAGE   40

      FCA7  C3 92 FC                 JMP     ERRO+2
                             * PROCESS DUPLICATE LABEL ERROR
      FCAA  3E 44            ERRD    MVI     A,A'D'          GET ERROR INDICATOR
      FCAC  32 A0 D0                 STA     OBUF+18         STORE IN OUTPUT RUFFER
      FCAF  CD CD F6                 CALL    AOU1            DISPLAY ERROR
      FCB2  C3 19 F7                 JMP     OPC             PROCESS OPCODE
      FCB5  23               PATCH#1 INX     H
      FCB6  C2 44 F3 ←               JNZ     MOV 23          C2 43 F3    corrected
      FCB9  C3 4A F3                 JMP     P+3

                             * DEFINE INPUT AND OUTPUT PORTS
                             *
                             *
      D000  00 00            USTA    EQU     0   306₈        UART STATUS
      D000  00 01            UDAI    EQU     1   307₈        DATA IN
      D000  00 01            UDAO    EQU     1   207₈        DATA OUT
      D000  00 06            PDAI    EQU     6               PROM DATA IN
      D000  00 07            PADO    EQU     7               PROM ADDRESS OUT
      D000  00 08            PDAO    EQU     8               PROM DATA OUT
      D000  00 09            PCTO    EQU     9               PROM CONTROL OUT
      D000  00 FF            SWCH    EQU     H'FF'
                             *
                             * FILE AREA PARAMETERS
      D000  00 06            MAXFIL  EQU     6               MAX # OF FILES
      D000  00 05            NMLEN   EQU     5               NAME LENGTH
      D000  00 0D            FELEN   EQU     NMLEN+8         DIRECTORY ENTRY LENGTH
      D000                   FILE0   RES     NMLEN
      D005                   BOFP    RES     2
      D007                   EOFP    RES     2
      D009                   MAXL    RES     4
      D00D                   FILTB   RES     (MAXFIL-1)*FELEN
      D04E                   INSP    RES     2               INSERT LINE POSITION
      D050  D0 4E            DELP    EQU     INSP            DELETE LINE POSITION
      D050  00 0D            ASCR    EQU     13              ASCII CARRIAGE RETURN VALUE
      D050                   HCON    RES     2
      D052  D0 50            ADDS    EQU     HCON            FIND ADDRESS
      D052                   FBUF    RES     NMLEN           FILE NAME BUFFER
      D057                   FREAD   RES     2               FREE ADDRESS IN DIRECTORY
      D059                   FEF     RES     1               FREE ENTRY FOUND FLAG
      D05A  D0 59            FOCNT   EQU     FEF             OUTPUT COUNTER
      D05A                   ABUF    RES     12              ASCII RUFFER
      D066                   BBUF    RES     4               BINARY RUFFER
      D06A                   SCNT    RES     1
      D06B                   DCNT    RES     1               DUMP ROUTINE COUNTER
      D06C  00 0A            NCOM    EQU     10              NUMBER OF COMMANDS
      D06C                   TABA    RES     2               SYMNOL TABLE END ADDRESS
      D06E                   ASPC    RES     2               ASSEMBLER PROGRAM COUNTER
      D070                   PASI    RES     1               PASS INDICATOR
      D071                   NCHR    RES     1               LENGTH OF STRING FOR COMPARE
      D072                   PNTR    RES     2               LINE POINTER STORAGE
      D074                   NOLA    RES     1               NUMHER OF LABELS
      D075                   SIGN    RES     1               SIGN STORAGE FOR SCAN
      D076                   OPRD    RES     2               OPERAND STORAGE
      D078                   OPRI    RES     1               OPERAND FOUND INDICATOR
```

```
LINE  ADDR  B1 B2 B3 B4 ERROR                 MICROTEC MACRO ASSEMBLER VER 3.1          PAGE   41

2161  D079                           TEMP     RES        1
2162  D07A   D0 4E                   APNT     EQU        INSP         ASSEMBLE LINE POINTER
2163  D07A   D0 6A                   AERR     EQU        SCNT         ASSEMBLER ERROR PRINT SWITCH
2164  D07A                           OIND     RES        2            OUTPUT ADDRESS
2165  D07C   00 05                   LLAB     EQU        5            LENGTH OF LABELS
2166  D07C                           AREA     RES        18
2167  D08E                           OBUF     RES        25           OUTPUT BUFFER AREA
2168  D0A7                                    RES        5
2169  D0AC                           IBUF     RES        83
2170  D0FF   D0 FF                   SYMT     EQU        $            START OF SYMBOL TABLE
2171  D0FF                                    END

   TOTAL ASSEMBLER ERRORS =     0
```

1010 1100

128
32
12
+83
255

FF
FF
FD

```
                                               MICROTEC MACRO ASSEMBLER VER 3.1          PAGE   42

                                  SYMBOL TABLE

 *  1

A        0007      ABUF     D05A       ACH1     F20E      ACHK     F1FC
ACO1     F755      ACO2     F7D5       ADDS     D050      ADE1     F19D
ADEC     F19A      ADR      F52F       AERR     D06A      AHE1     F187
AHEX     F184      AHS1     F1CB       ALAB     FC2A      ALP1     FB5A
ALPS     FB58      AOU1     F6CD       AOU2     F6C7      AOUT     F6C0
APNT     D04E      AREA     D07C       ASBL     FB77      ASC1     FBA7
ASC2     FBB7      ASC3     FBCF       ASC4     FBD9      ASC5     FBF8
ASC7     FC1B      ASCN     FB7A       ASCR     000D      ASM1     F687
ASM2     F634      ASM3     F678       ASM4     F668      ASM5     F671
ASPC     D06E      ASSM     F658       ASTO     F8D4      ASUB     FC21
AVAL     FC0B      B        0000       BBUF     D066      BID1     F24D
BIN1     F232      BIND     F23C       BINH     F21F      BLK1     F1F6
BOFP     D005      BSPA     F064       C        0001      CHAR     F06C
CHOT     F1D9      CLER     F086       CO1      F558      CO2      F561
COM0     F556      COM1     F569       COMM     F0C3      COMS     F0D4
COND     F905      COP1     FA4C       COPC     FA3A      CR       F03C
CRLF     FCA6      CTAB     F257       D        0002      DAT1     F78D
DAT2     F7EB      DCNT     D06B       DEL      F057      DEL1     F5F5
DEL2     F612      DEL3     F630       DEL4     F63A      DEL5     F638
DELL     F5E1      DELP     D04E       DLA1     F5D7      DLA2     F5D8
DLAY     F5D5      DONE     F158       DOUT     F1EA      DUM1     F285
DUM2     F2[illegible]7      DUMP     F29B       DUMS     F2A3      E        0003
EASS     F6E8      EEND     002F       EMES     F426      EMES1    F42C
EMES2    F431      ENT1     F450       ENTR     F437      ENTS     F444
EO1      F5[illegible]8      EOF      F527       EOFP     D007      EOR      F01F
EQU1     F770      EQUL     F4BC       EQUS     F77B      ERO1     FC9C
ERRA     FC8B      ERRL     FCAA       ERRL     FCA5      ERRM     FC82
ERRO     FC50      ERRR     FC64       ERRS     FC6D      ERRU     FC78
ERRV     FC7D      ETRA     F112       EXEC     F0B9      FAST     F3A6
FBUF     D052      FEET     F388       FEF      D059      FELEN    000D
FI1      F50E      FI2      F520       FIL30    F370      FIL35    F372
FILE     F2FF      FILE0    D000       FILTB    D00D      FIN1     F50B
FIND     F5[illegible]      FINE     F389       FOCNT    D059      FOOD     F3A3
FOOL     F3[illegible]4      FOOT     F383       FOUL     F385      FOUT     F379
FREAD    D0[illegible]      FSE10    F3E1       FSE15    F404      FSE20    F40E
FSEA     F303      GT       F4E4       H        0004      HCON     D050
HOTB     F1E3      HOUT     F103       IBUF     D0AC      IN8      F08E
INCA     F0[illegible]7      INIT2    F006       INITA    F000      INSP     D04E
INSR     F403      L        0005       LCHK     F730      LF       F0AB
LICK     F4[illegible]      LINE     F476       LIS1     F588      LIST     F584
LLAB     0005      LMOV     F534       LODM     F546      LT       F4D1
M        0005      MAXFIL   0006       MAXL     D009      MESS     F420
MLAB     F7[illegible]0      MOV23    F343       MPNT     F8A5      NCHR     D071
NCOM     000[illegible]      NEXT     F02D       NMLEN    0005      NOLA     D074
NOR1     F5[illegible]      NORM     F571       NOV1     F649      NOVR     F643
NUM1     FC5[illegible]      NUM2     FC5E       NUMS     FC46      NXT1     FB84
NXT2     FB8[illegible]      OBUF     D08E       OCN1     FADF      OCN2     FAE2
OCNT     FAC[illegible]      OERF     FAF0       OIND     D07A      OX       F0A2
OP1      FA6[illegible]      OP2      FA6E       OP4      FAC6      OP5      FAC9
```

```
OPAD   FAC5          OPC    F719          OPC2   FA73          OPC3   FA8F
OPCD   FA4F          OPRD   D076          OPRI   D078          ORG1   F758
ORG2   F7F4          OTAB   F911          OUT8   F09B          PABL   F7A9
PADO   0007          PAG1   F2EB          PAG2   F2F7          PAGE   F2D8
PAS1   F6D7          PAS2   F790          PASI   D070          PCTO   0009
PDAI   0006          PDAO   0008          PNTR   D072          PRO1   F5A0
PRO2   F5AE          PRO3   F5CE          PROM   F59D          PSEU   FAF8
PSU1   F73F          PSU2   F7C4          PSW    0006          READ   F025
RES1   F785          RES2   F7DB          RES21  F7E7          RMOV   F53D
ROOM   F339          RTAB   FB47          SBL1   F8FA          SBL2   F8FE
SBLK   F8F7          SCNT   D06A          SCRN   F213          SEAR   F0EB
SEN1   FC40          SEND   FC36          SEOF   F4AC          SIGN   D075
SLA1   FB2E          SLA2   FB41          SLA3   FB44          SLAB   FB0A
SP     0006          SSTR   FBF2          STOM   F54E          SWAP   F350
SWAPS  F34B          SWCH   00FF          SYMT   D0FF          SYS8   F00C
TABA   D06C          TEMP   D079          TEST   F324          TY31   F83B
TY32   F83E          TY41   F867          TY56   F893          TY6    F8C4
TYP1   F813          TYP2   F817          TYP3   F82A          TYP4   F849
TYP5   F87E          TYP6   F884          TYS5   F886          TYS6   F8C7
UDAI   0001 ~207     UDAO   0001 ~207     USTA   0000 ~206     VAL1   F121
VAL2   F13C          VAL3   F14D          VAL4   F14F          VAL5   F15D
VALC   F10B          VCHK   F293          WHA1   F41D          WHAT   F41A
ZBU1   F104          ZBUF   F0FE          ZERO   F4F4
```

(octal)