SYMBOLIC EDITOR PROGRAMMING MANUAL

PDP-8

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PDP-8 SYMBOLIC EDITOR PROGRAMMING MANUAL

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PREFACE

The PDP-8 comes to the user complete with an extensive selection of system programs and routines making the full data processing capability of the new computer immediately available to each user, eliminating many commonly experienced initial programming delays.

The programs described in these abstracts come from two sources, past programming effort on the PDP-5 computer, and present and continuing programming effort on the PDP-8. Thus the PDP-8 programming system takes advantage of the many man-years of program development and field testing by PDP-5 users.

Although in many cases PDP-8 programs originated as PDP-5 programs, all utility and functional program documentation is issued in a new, recursive format introduced with the PDP-8.

Programs written by users of either the PDP-5 or the PDP-8 and submitted to the users' library (DECUS - Digital Equipment Corporation Users' Society) are immediately available to PDP-8 users.

Consequently, users of either computer can take immediate advantage of the continuing program developments for the other.

CONTENTS

ILLUSTRATION

INTRODUCTION

The PDP-8 Symbolic Editor allows the user to prepare and edit symbolic tapes in ASCII code with the ASR-33 Teleprinter on line. The tedious task of correcting symbolic program tapes using the ASR-33 off line is thereby avoided. Proper use of the PDP-8 Symbolic Editor can substantially ease the labor and reduce the number of passes necessary to correct symbolic program tapes.

The Editor reads a page, or section, of symbolic tape into a buffer in core storage, where it is available for examination and correction. The page buffer occupies all of core not taken up by the Editor itself and has a capacity of approximately 6000 characters. When the Editor has finished reading a page, a bell rings to signal the user to commence editing. The user may then call for a listing of individual (numbered) lines, in any order, and insert desired changes and corrections. In addition, text may be added to the buffer, or inserted between specified lines. Groups of lines or individual lines may be deleted by a single command, or the entire page may be erased if desired. Upon command, the Editor will then either list or punch out the corrected lines or page on paper tape. The Editor can also be used to generate a new symbolic tape by typing input directly on the console keyboard.

CHAPTER 1

OPERATING FEATURES

By convention, paper tape information is organized into variable sized blocks, called pages and lines. Pages are separated on paper tape by ASCII form feed codes, and lines are separated by carriage return/line feed pairs. A page of text may contain up to 60 lines of heavily commented text; it can hold up to 240 lines without comments or formatting. Each line of text includes the terminating carriage return/line feed combination. All lines in the memory buffer are implicitly numbered in decimal notation starting with 1. This implicit enumeration is continually updated by the Editor to take account of line insertions and deletions. For editing and listing purposes, each line is referred to by its current implicit decimal number on the page.

Since the form feed code is not stored in core memory, there are no page divisions in the text buffer; the entire contents of the buffer are treated as a single page. However, the user may organize the contents of the text buffer into several pages if he so desires.

MODES OF OPERATION

To distinguish between editing commands and the actual text to be entered in the buffer, the Editor operates either in the Command Mode or Text Mode, respectively. In the command mode all input typed on the ASR-33 Teleprinter will be interpreted as commands to the Editor to perform some operation, or to allow the operator to perform some operation, on the text stored in the buffer. In the text mode, all typed input is interpreted as text to replace, be inserted into, or appended to the contents of the text buffer.

Transition Between Modes

After being loaded into core memory the Editor is in the command mode; that is, the program is waiting for a command. The user types the desired command code and terminates it by striking the carriage return (RETURN) key. This nonprinting character (hereafter represented by "↵")

tells the Editor to carry out the command--to either replace, insert, or append the data. The Editor then enters the requested text mode and responds with a line feed character (represented by ↓ as soon as it has processed the command and begun the operation.

After the Editor has entered the text mode, the user types the desired corrections or insertions. When finished, he strikes the FORM key with the CTRL key depressed. This special combination character generates the form feed code which tells the Editor to return to the command mode. A bell rings to indicate the transition back to the command mode.

COMMAND STRUCTURE

A command directs the Editor to perform a desired operation. Each command consists of a single letter, preceded by zero, one, or two arguments. The command letter tells the editor what to do; the arguments usually specify which numbered line or lines of text are affected. (Some arguments consist of special characters, see below.) Commands to the Editor must take one of the following forms, where E represents any command letter and the symbol) represents the non-printing character for carriage return (RETURN key):

Type of Command	Command Format	Meaning
No Argument	E)	Perform operation "E"
One Argument	nE)	Perform operation "E" on the referenced line.
Two Arguments	n,mE)	Perform operation "E" on lines n through m, inclusive.

The arguments m and n, which refer to numbered lines in memory, must be positive and m must be greater than n.

Two arguments must be separated by a comma, but no comma is allowed between the argument(s) and the command. Note also that in order to be executed a command must be followed by a carriage return ( )). When the RETURN key is depressed during operation of the Editor, the line feed code (↓) will be automatically generated by the Editor.

COMMAND REPERTOIRE

Commands to the Editor are grouped under three general headings:

Input Commands
Editing Commands
Output Commands

A listing and explanation of the three types of commands is given in the following tables. Unless otherwise noted, the Editor returns to the command mode after completing each operation specified by a command.

The Editor will print an error message consisting of a question mark whenever the user has requested nonexistent information or used an inconsistent or incorrect format in typing a command. For example, if a command requires two arguments, and only one (or none) is provided, the Editor will print "?" and ignore the command as typed. Similarly, if an argument is provided for a command that does not require one, or if a nonexistent command character is typed, the error message "?" will be typed and the command will be ignored.

Input Commands

Command	Action and Explanation
R↓	READ a page of text from paper tape reader. The Editor will read information from the input tape until a form feed (FORM/CTRL key combination) is encountered. All incoming text except the form feed is appended to the contents of the text buffer. Information already in the buffer remains there.
A↓	APPEND the incoming text from the ASR-33 teleprinter keyboard to the information already in the buffer (the buffer may be empty initially). The Editor will enter the text mode upon receiving this command and the user may then type in any number of lines of text. The new text will be appended to the information already in the buffer, or entered into an empty buffer, until the form feed (FORM/CTRL) key combination is struck.

By giving the APPEND command with an empty buffer, a symbolic program tape may effectively be generated on-line by entering the program via the keyboard.

NOTE: In both of these commands, the Editor returns to the command mode after reading the form feed character.

Editing Commands

The following commands permit deletion, alteration, or expansion of text in the buffer.

Command	Action and Explanation
K↵	KILL the entire page in the buffer. The values of special characters "/" and "." are set to zero. (See below.) The Editor remains in command mode.
nD↵	DELETE line n. Line n is removed from the text buffer. The numbers of all succeeding lines are reduced by one, as is the line count.
n,mD↵	DELETE lines n through m, inclusive. The line following line m becomes the new line n. The Editor remains in command mode after all DELETE operations.
nI↵	INSERT the typed in text before line n, until a form feed (FORM/CTRL) is encountered. The Editor enters text mode to accept input. The first line typed becomes the new line n. The numbers of all lines following the insertion, as well as the line count, are increased by the number of lines inserted. To reenter the command mode, the FORM/CTRL key combination must be depressed. If this is not done, all subsequent commands will be interpreted erroneouslly as text to be entered in the program.
nC↵	CHANGE line n. Line n is deleted, and the Editor enters text mode to accept input. The user may now type in as many lines of text as he desires

in place of the deleted line. If more than one line is inserted, subsequent lines will be automatically renumbered. Return to the command mode is accomplished by depressing the FORM/CTRL key combination.

n,mC↵ — CHANGE lines n through m, inclusive. This routine calls the DELETE routine and lines n through m are deleted, allowing the user to type in any number of lines (or none at all) in their place. The FORM/CTRL keys must be depressed to return the Editor to the command mode.

When lines are changed or deleted the space previously occupied is not recovered. Thus, it is possible to overflow the buffer by changing and/or deleting lines. This possibility may be readily eliminated by logically segmenting a program into pages of 40 or 50 lines.

Output Commands

Output commands are subdivided into list and punch commands. List commands will cause the printout on the ASR-33 Teleprinter of all or any part of the contents of the text buffer to permit examination of the text. Punch commands provide for the output of corrected text or for the duplication of pages of an input tape. All punch commands halt the computer. List or punch commands do not affect the contents of the buffer.

List Commands

The following commands cause part or all of the contents of the text buffer to be listed on the ASR-33 Teleprinter.

Command	Action and Explanation
L↵	LIST the entire page. This causes the Editor to list the entire contents of the text buffer.
nL↵	LIST line n. This line will be typed out, followed by a carriage return and a line feed.

n,mL↵ — LIST lines n through m, inclusive. Lines n through m will be listed on the teleprinter. The editor remains in the command mode.

Punch Commands

The following commands will punch out corrected text or duplicate pages from the input tape.

Note that all punch commands halt the computer, allowing the user to turn on the paper tape punch.

Command	Action and Explanation
P↵	PUNCH out the entire contents of the text buffer. Note that the Editor has no provisions for generating leader/trailer code. See Operation, page 2-5, for the procedure to generate leader/trailer.
nP↵	PUNCH line n.
n,mP↵	PUNCH lines n through m, inclusive, on the paper tape punch.
F↵	Form feed. The Editor punches approximately 1 inch of 200 code, a form feed character and approximately 5 inches more of 200 code. The computer then halts, allowing the user to turn off the paper tape punch. Pressing the CONTINUE key restarts the Editor. This command is used to logically divide a program into pages for reediting. The assembler will ignore the form feed.

SPECIAL CHARACTERS AND FUNCTIONS

A number of keys have special operating functions for use by Editor control. These keys and their associated functions are listed below. The first three are nonprinting; the symbols in parentheses are used for purposes of illustration.

Carriage Return (↵)

In both command and text modes, striking the carriage return key (RETURN) signals the Editor to process the information just typed. In command mode, the specified operation will be

performed following this terminator. In text mode, the preceding line of text will be placed in the text buffer.

Line Feed (↓)

The lines on each page of text are separated by the carriage return/line feed (CR/LF) combination. When a line is terminated by striking the RETURN key, the Editor program will automatically generate the line feed code. Line feeds will be ignored on input, but are automatically regenerated after each carriage return on output.

Back Arrow (←)

The back arrow (←) is used for error recoveries in both text and command modes. When used in text mode, (←) cancels the entire line. The user continues typing on the same line. When typing commands, (←) cancels the line as it was written and the Editor then generates a carriage return/line feed (CR/LF) combination.

Rubout (RO) and Leader/Trailer Code (200)

When read from paper tape, rubouts, leader/trailer code (200), and blank tape are completely ignored.

Tabulation (→|) and Switch Register Bit 1 Option

When the TAB key is struck in combination with the CTRL key, the Editor produces a tabulation. TAB (→|) generates up to ten spaces to the next tab stop on input from either the ASR-33 keyboard or the paper tape reader. On output, the tab spaces are generated or suppressed in accordance with the setting of switch register (SR) bit position 1:

If SR bit 1 is in the down (0) position, spaces will be generated in place of tabs.

If SR bit 1 is in the up (1) position, tabs will be generated.

The Editor cannot replace spaces with tabs until a tab has been inputted.

Form Feed (FORM/CTRL Combination)

The form feed is generated by depressing the FORM/CTRL key combination. It directs the Editor to leave the text mode of operation and return to the command mode.

Slash (/)

The slash symbol (/) stands for the numerical (decimal) value of the last line of the current page stored in the text buffer. It may also be used as part of an argument. For example, the command /L) will cause the last line in the text buffer to be listed.

Period (.)

The period symbol (.) stands for the numerical value of the line currently being processed. The period may be used as part of an argument. For example, the command 15, .C) will allow the user to change line 15 to the current line by deleting the old data and replacing it with data entered via the ASR-33 keyboard.

After a delete command (D) has been executed, the period symbol stands for the number of the line preceding the first deleted line. For example, if lines 15 to 23 of the current page have been deleted by giving the command "15,23D)", the period will equal line 14 after execution of the command.

Equal Sign (=)

When the equal sign is typed in the command mode, the Editor will print out the decimal value of the argument that precedes it, followed by a carriage return. The symbol may be used to determine the number of lines of text in the buffer (indicated by "/") or the value of the current line (indicated by "."). For example,

/=59 indicates that there are 59 lines in the text buffer (i.e., the decimal value of the last line is 59).

and

.=27 indicates that the current line is numbered "27".

and

/-6=53

Minus Sign (-)

Arithmetic subtract: This may be used to combine symbols in arguments.

/-6, /L↵ will list the last seven lines of the text buffer.

Space (␣)

Arithmetic addition: This may also be used as is the minus (-). For example;
.-1, .␣1L↵ will list the line preceding the current line, the current line itself, and the line following the current line.

CHAPTER 2

OPERATING PROCEDURE

This chapter describes the sequence of operations necessary to load, edit, and punch out a corrected symbolic program tape, and gives examples of the use of the Editor.

After the Editor has been loaded, it may be used to read into the text buffer a page of the symbolic program to be corrected. When the page has been read in and a form feed code has been encountered, the Editor strikes a bell to signal the user that the Editor is in the command mode. Corrections and additions may then be either typed in from the ASR-33 Teleprinter keyboard or inserted from paper tape via the reader. Individually numbered lines may be listed in any order to permit insertions, deletions, or changes. Text may be inserted between specified lines or appended to the end of a section. Individual lines, groups of lines, or an entire page may be deleted upon a single command. To ensure that a tape is correct, desired portions or an entire page may be prelisted before punching begins. Finally, the corrected lines, groups of lines, or the entire page may be listed and/or punched. The original text remains available in the core buffer in case further corrections are necessary.

The following paragraphs give the detailed procedure for loading the Editor and a symbolic tape, making required corrections, and punching the corrected symbolic tape.

LOADING SEQUENCE

Before editing can begin, the Symbolic Editor program must be loaded into core with the Binary Loader (Digital-8-2-U-Rim), and the symbolic program tape to be corrected must be read into the core text buffer. The loading of the symbolic tape is performed by the Editor itself under keyboard control.

Loading The Binary Loader

The Binary Loader is loaded into core by the Read-In Mode (RIM) Loader. The RIM Loader itself is initially placed into core memory by PDP-8 console keys and switches, as is explained in

Digital-8-1-U. When this has been done, the RIM Loader is used to load the Binary Loader into core by means of the following procedure:

1. Put the Binary Loader tape in reader.

2. Set the SWITCH REGISTER (SR) to 7756, the starting address of the RIM Loader.

3. Press the LOAD ADDRESS key and then the START key.

4. Turn on the reader and wait until the tape is completely read in. When the reader stops, the Binary Loader is in memory.

Loading The Symbolic Editor With The Binary Loader

To load the Symbolic Editor with the Binary Loader, place the Editor program tape in the reader, set the SR to 7777 (the starting address of the BIN Loader), press the LOAD ADDRESS key and then the START key, and turn on the tape reader. When the reader stops, the Editor is in core memory.

If the ACCUMULATOR (AC) does not contain zero when the reader halts, a checksum discrepancy exists which indicates that the Editor tape has been read in incorrectly. Load the tape again by repeating the procedure described above.

To start the Editor program initially set the SR to 176, press the LOAD ADDRESS key and then the START key. The loading of the symbolic tape to be edited and all subsequent operations are performed through the use of the Editor by giving appropriate commands from the Teletype keyboard.

Loading A Symbolic Tape Using The Editor

1. Place the symbolic tape of the program to be corrected in the paper tape reader.

2. At the ASR-33 keyboard, type the read command followed by a carriage return (i.e., type "R↵"). Turn on the reader. The symbolic tape will be read automatically into the text buffer.

3. The Editor will continue reading the tape until either the text buffer in core is almost full (full capacity is approximately 150 lines) or the form feed code is encountered at the end of the tape. If the tape contains no form feed code, strike the FORM/CTRL key combination after the tape has been read in. Upon recognizing the form feed character, the Editor enters the command mode and rings a bell to indicate that it is ready for the first command.

CAUTION

If the form feed code is encountered before the symbolic tape has completely read in (as indicated by the ringing of the bell), turn off the paper tape reader. Otherwise, characters on tape will be interpreted as commands to the Editor. The section of tape read in up to the form feed code should then be edited first before proceeding with the remainder of the tape.

EDITING A SYMBOLIC TAPE

The actual editing procedure depends, of course, on a particular user's requirements. The general procedure is illustrated in the example that follows. For input, editing, and output commands that may be given to the Editor, refer to the detailed explanation of the Command Structure, Command Repertoire and Special Characters and Functions under Operating Features (see pages 1-2 through 1-9) or see the corresponding summaries of commands and special characters in the appendix. Observe also the following operating notes and precautions:

1. Terminate each command to the Editor by striking the RETURN (carriage return) key. This directs the Editor to execute the command.

2. After a command to insert, change, or append text to the symbolic program has been executed, the Editor remains in the text mode until the operator

hits the FORM/CTRL key combination on the teletypewriter. This combination generates the form feed code character, which tells the Editor to return to the command mode.

3. If a great deal of text is added during the course of editing, the text buffer may overflow (full capacity is approximately 150 lines). If this occurs, the Editor rings the bell and returns to the command mode. Text entered after the buffer is full is ignored, and the contents of the buffer are not disturbed.

4. The Editor may be stopped at any time by pressing the STOP key; to continue press the CONTINUE key. If it is desired to restart the Editor without disturbing the buffer, place octal address 177 in the SR and press the LOAD ADDRESS and the START keys. If it is desirable to clear the buffer and then restart, place octal address 176 in the SR and press the START key.

Error Messages

The proper rules for giving commands containing from zero to two arguments must be observed during editing, as is explained under Operating Features (page 1-2). If commands are given in an incorrect format, or if arguments are either missing or extraneous, the Editor will respond with an error message by typing out a question mark. Notice that some commands can legitimately take from zero to two arguments. In general, if an argument is either missing or extraneous, the Editor types out "?" and ignores the command. Similarly, if a negative argument is encountered or an illegitmate command again is typed, the Editor again responds with the error message.

PUNCHING THE CORRECTED SYMBOLIC TAPE

The procedure for punching out the corrected symbolic tape depends to some extent on the user's requirements. The general sequence of steps is given below:

1. As desired, give the output command to punch out either line n of the text (nP↓), lines n through m (n,mP↓), the entire text (P↓), or form feeds (F↓).

2. After the carriage return (↵) following the punch command has been received, the computer will halt. Turn on the punch.

> NOTE: The Editor has no provisions for generating leader/trailer code. If leader/trailer code is to be punched, switch the ASR-33 Teleprinter to LOCAL (off-line), generate as much leader/trailer (code 200) as needed, and then switch the teleprinter back to LINE (on-line). To produce the code 200 leader, simultaneously press and hold the CTRL and SHIFT keys with the left hand; press and hold the REPT key; press and release the @ key. When the required amount of leader has been punched, release all keys. To produce the 377 code, simultaneously press and hold both the REPT and RUB OUT keys until a sufficient amount of leader has been punched. After the Editor has finished the punch operation, more leader/trailer code may be generated in the same way.

3. Press the CONTINUE key at the console to start punching.

If a tabulation has been produced by giving the TAB (⟶|) command, the tab spaces will automatically be generated on output. If the tab spaces are to be suppressed during punchout, set SR bit 1 at the console to the up (1) position.

4. After a tape has been punched, turn off the punch before typing any further commands. If this is not done, the control codes typed in will be punched on the symbolic tape.

5. Punching the symbolic program does not delete it from memory. The page remains in the text buffer in core until the KILL command (K ↵) is given to erase it. As the tape is punched, a listing of the actual tape is simultaneously printed. Thus, last minute errors may be spotted and corrected by further editing. If this is necessary, the affected lines may be recalled from the buffer by use of the Editor. If there are no errors, the correct listing of the symbolic tape is immediately available. If it is desired to read another tape into the buffer, the user must first delete the entire page of text (K ↵).

Remember that the recommended page length, as delimited by the form feed, is 60 lines or less. However, the Editor will accept more text if necessary.

EXAMPLE OF USE

The following detailed example of the editing of a page of text is intended to familarize the user with the basic operations of the Symbolic Editor. Where details of the loading sequence and operating procedures are not shown, it is assumed that the user has followed the correct procedures previously explained.

This example concerns a program for adding up numbers stored in locations 200_8 through 207_8 of the computer, with the answer to be stored in location 410_8. The program is to start in location 600. The program listing is shown in Figure 1.

```
/ADD UP NUMBERS
*600
BEGN,      HLT
/TO START THE PROGRAM. HIT "CONTINUE" ON CONSOLE
/
/THE NEXT FIVE INSTRUCTIONS INITIALIZE THE ROUTINE
           CLA          /CLEAR THE ACCUMULATOR
           TAD M10      /LOAD AC WITH THE NUMBER -10
           DCA COUNTR /PUT INTO COUNTER
           TAD TWOHUN /LOAD AC WITH FIRST ADDRESS
           DCA POINTR /PUT IT INTO POINTER
/
/THE NEXT SEVEN INSTRUCTIONS ARE THE PROGRAM ITSELF
BEGN,      TAD I POINTR            /ADD NEXT NUMBER
           ISZ POINTR              /INDEX POINTER
           ISZ COUNTR              /INDEX COUNTER, IS IT ZERO?
           JMP BEGN                /NO; CONTINUE ADDING
           DCA I ANSWER            /YES; STORE ANSWER
           HLT                     /HALT
           JMP BEGN+1

/THE NEXT THREE REGISTERS CONTAIN THE CONSTANTS
M10,       -10                     /NEGATIVE TALLY NUMBER
TWOHUN,    200                     FIRST ADDRESS IN BUFFER
ANSWER     410

/THE NEXT TWO REGISTERS ARE RESERVED FOR VARIABLES
COUNTR,    0
POINTR,    0

$
```

Figure 1 Program Listing of Addition Routine

Let us assume that we have attempted to assemble this program using the PDP-8 Symbolic Assembler (PAL III). On PASS1, however, the Assembler typed the following:

```
DT  BEGN     AT  0606

UA  ADDRES   AT  0616

UA  ANSWER   AT  0612
BEGN     0600

UA  BUFFER   AT  0616
COUNTR   0620

UA  FIRST    AT  0616

UA  IN       AT  0616
M10      0615
POINTR   0621
TWOHUN   0616
```

The message DT GEGN AT 0606 signifies that the programmer has mistakenly used identical tags to specify two different addresses. An inspection of the program listing (Figure 1) shows that the tag "BEGN" has, indeed, been duplicated. It appears in line 3 of the listing as "BEGN, HLT," then in line 14, starting with "BEGN, TAD I POINTR." (Since the line numbers are implicit only, they are not shown in the example; they may be obtained by counting from the top down in Figure 1.) To correct the situation, the Symbolic Editor was read in with the Binary Loader, as explained under Loading Sequence in this chapter. The symbolic tape to be corrected was then loaded through the Editor by means of the READ (R) command. A series of commands were given: (the lines typed by the Editor have been underlined for clarity).

```
R

14L
BEGN,       TAD I POINTR          /ADD NEXT NUMBER
14C
ADDR,       TAD I POINTR          /ADD NEXT NUMBER
17L
            JMP BEGN              /NO; CONTINUE ADDING
17C
            JMP ADDR              /NO; CONTINUE
13,18L
/THE NEXT SEVEN INSTRUCTIONS ARE THE PROGRAM ITSELF
ADDR,       TAD I POINTR          /ADD NEXT NUMBER
            ISZ POINTR            /INDEX POINTER
            ISZ COUNTR            /INDEX COUNTER, IS IT ZERO?
            JMP ADDR              /NO; CONTINUE
            DCA I ANSWER          /YES; STORE ANSWER
```

```
25L
ANSWER      410
25C
ANSWER,     410
24L
TWOHUN,     200                    FIRST ADDRESS IN BUFFER
24C
TWOHUN,     200                    /FIRST ADDRESS IN BUFFER
.-1,.2L
M10,        -10                    /NEGATIVE TALLY NUMBER
TWOHUN,     200                    /FIRST ADDRESS IN BUFFER
ANSWER,     410
```

Having made the desired corrections, the programmer finally asks the Editor to punch out the entire text on tape by giving the punch (P) command, but still withholds the KILL command (K), pending final corrections. As the tape is punched out, a new program listing is simultaneously printed out and the entire text is preserved in the buffer.

```
/ADD UP NUMBERS
*600
BEGN,       HLT
/TO START THE PROGRAM. HIT "CONTINUE" ON CONSOLE
/
/THE NEXT FIVE INSTRUCTIONS INITIALIZE THE ROUTINE
            CLA          /CLEAR THE ACCUMULATOR
            TAD M10      /LOAD AC WITH THE NUMBER -10
            DCA COUNTR   /PUT INTO COUNTER
            TAD TWOHUN   /LOAD AC WITH FIRST ADDRESS
            DCA POINTR   /PUT IT INTO POINTER
/
/THE NEXT SEVEN INSTRUCTIONS ARE THE PROGRAM ITSELF
ADDR,       TAD I POINTR             /ADD NEXT NUMBER
            ISZ POINTR               /INDEX POINTER
            ISZ COUNTR               /INDEX COUNTER, IS IT ZERO?
            JMP ADDR                 /NO; CONTINUE
            DCA I ANSWER             /YES; STORE ANSWER
            HLT                      /HALT
            JMP BEGN+1

/THE NEXT THREE REGISTERS CONTAIN THE CONSTANTS
M10,        -10                      /NEGATIVE TALLY NUMBER
TWOHUN,     200                      /FIRST ADDRESS IN BUFFER
ANSWER,     410

/THE NEXT TWO REGISTERS ARE RESERVED FOR VARIABLES
COUNTR,     0
POINTR,     0

$
```

The PASS1 result of assembling this program is:

```
ADDR    0606
ANSWER  0617
BEGN    0600
COUNTR  0620
M10     0615
POINTR  0621
TWOHUN  0616
```

APPENDIX 1

SUMMARY OF SYMBOLIC EDITOR OPERATIONS

Special Key Functions

carriage return (RETURN key)	Complete specified action.
form feed (FORM/CTRL key combination)	Leave text mode.
back arrow (←—)	Text mode: cancels entire line.
	Command mode: cancels line as written; Editor generates carriage return/line feed (CR/LF) combination.
tabulation (TAB/CTRL combination)	On input generates up to ten tab spaces to next tab stop. Tabulation may be suppressed on output by placing SR bit 1 to up (1) position.

Editor Command Summary

Command	No. of Arguments	Function
R	0	Read incoming text from reader and append to text buffer.
A	0	Append incoming text from teleprinter.
K	0	Kill page of text in text buffer.
nD	1	Delete line n of text.
n,mD	2	Delete lines n through m, inclusive.

nI	1	Insert text before line n.
nC	1	Change line n of text.
n,mC	2	Change lines n through m, inclusive.
L	0	List entire page of text.
nL	1	List (print) line n of text.
n,mL	2	List lines n through m, inclusive.
P	0	Punch the entire contents of the buffer.
nP	1	Punch line n of text in buffer.
n,mP	2	Punch lines n through m, inclusive.
F	0	Punch form feed code.

Special Arguments

Argument	Meaning
Slash (/)	Decimal value of last line of current page
Period (.)	Decimal value of current line being processed
Equal sign (=)	When typed in command mode, decimal value of argument preceding it will be printed out. For example, /=58 indicates that last line is 58; and .=24 indicates that current line is 24.

APPENDIX 2

HIGH-SPEED VERSION OF SYMBOLIC EDITOR

There are several differences in the Editor command structure when using the high-speed version. These arise because of the independence of the keyboard/high-speed reader and teleprinter/high-speed punch.

Command summary (only those commands which are different are listed):

R↵ — Read a page of text from the high-speed reader and put into the core text buffer. The Editor will ring the bell and return to the command mode when either

1. a form feed character was read (signifying end of the logical page)
2. the tape ran off the end (did not have a form feed)
3. the buffer has overflowed

P↵, nP↵, n,mP↵ — Punch onto the high-speed punch. Does not halt the computer. The Editor returns to the command mode when done. The text in the buffer is unchanged.

T↵ — Punch leader/trailer code (blank) on the high-speed punch. Editor remains in the command mode.

F↵ — Punch some blank tape, a form feed, and some blank tape on the high-speed punch. The Editor remains in the command mode.

nN↵ — Punch and read in pages of text. This is equivalent to

P↵
F↵
K↵
R↵

repeated n times. The Editor returns to the command mode when done.

alt mode	This is equivalent to .+1L) It examines the next line.
	This is equivalent to .-1L) It examines the previous line.
<	

digital

Washington, D. C. • Parsippany, N. J. • Los Angeles • Palo Alto • Chicago • Ann Arbor Pittsburgh • Denver • Huntsville • Orlando Carleton Place, Ont. • Reading, England • Paris, France • Munich, Germany • Sydney, Australia

PRINTED IN U.S.A.

20-8/65