

ELLIOTT

903

Volume 2: PROGRAMMING INFORMATION
Part 3: UTILITY PROGRAMS
Section 2: EDIT

Contents

	Page
Chapter 1: INTRODUCTION	
1.1 Purpose	1
1.2 Form of Distribution	1
1.3 Method of Use	1
1.4 Entry Points	1
Chapter 2: FUNCTIONS	
2.1 Character Sets	2
2.2 Basic Functions	2
2.2.1 FL Find Line	3
2.2.2 DL Delete up to Line	3
2.2.3 FC Find Characters in Succession	3
2.2.4 DC Delete up to Characters in Succession	3
2.2.5 FE Find End of Line	3
2.2.6 DE Delete up to End of Line	3
2.2.7 IS Insert String	3
2.2.8 IL Insert on New Line	3
2.2.9 IB Insert Block	4
2.2.10 RE Remainder of Tape	4
2.2.11 SH Stop on Halt	4
2.2.12 IH Insert Halt	4
2.2.13 ST Stop	4
2.2.14 CO Comment	4

	Page
2. 3 Example	5
Chapter 3: ERRORS	7
Chapter 4: OPERATING INSTRUCTIONS	8
Chapter 5: STORE USED	9
Chapter 6: TIME TAKEN	9

Chapter 1: INTRODUCTION

1.1 Purpose.

A modified copy of an eight hole paper tape is produced by means of deletion and insertion of strings of characters. It is suitable for editing a source program in SIR, ALGOL, FORTRAN, T2 etc., but not for editing a binary tape.

1.2 Form of Distribution.

903 EDIT is distributed as a mnemonic tape for input by SIR.

1.3 Method of Use.

A series of commands are read from a previously prepared tape and stored without checking. Each command is then read in succession from the stack and an appropriate amount of input and output tape is possessed.

EDIT may run at any program level and in any store module.

1.4 Entry Points.

- 0; Read correction tape
- 1; Read first input tape
- 2; Read subsequent input tapes
- 3; Re-entry after parity errors
- 4; Re-entry after stop instructions
- 5; Read output tape to check punching

Chapter 2: FUNCTIONS

2.1 Character Sets.

All tapes consist of a string of eight hole characters in 4100 code. The characters may be classified as:

- | | | | |
|-----|-------------------|---------------------------------------|---|
| (a) | Ignorable: | Carriage Return | Ⓞ |
| | | Erase | Ⓞ |
| (b) | Separators: | Space | Ⓞ |
| | | Tab | Ⓞ |
| (c) | Blanks or | | Ⓞ |
| | Runout | | |
| (d) | Newline | | Ⓞ |
| (e) | Halt: | End of tape | Ⓞ |
| | | character | |
| (f) | All odd | | |
| | parity characters | | |
| (g) | Alphanumeric: | All previously undefined, even parity | |
| | | characters. | |

2.2 Basic Functions.

Each tape consists of a succession of lines, each line being terminated by Ⓞ. Ignorable characters are always ignored on input.

The correction tape contains a series of commands each contained on one line (except for IB, see 2.2.9), the blank run-in being ignored. The first two alphanumeric characters on a line define the command. If the next character is a separator it is ignored, leading blanks are also ignored. The remainder of the line contains the command string (excluding Ⓞ at the end of the line).

The input tape also contains a number of lines. Blank run-in is not copied but the number of blanks is counted and when a similar number of blanks is found in sequence this is treated as the end of the tape. A halt code is not a terminator for this tape (except for SH, see 2.2.11). The remainder of the tape is treated according to the command read in on the command tape.

When Ⓞ is input, the sequence Ⓞ Ⓞ Ⓞ is always output. A check is made that if a blank follows the Ⓞ no further blanks are inserted.

The following commands are available.

2. 2. 1 FL Find Line

Blanks in the command string are ignored. Copy the input tape until a new line beginning with the command string is matched. Separators on the input tape are output but may be ignored in the matching, except that they must be present if in the command string. The last character output is the last character of the command string.

2. 2. 2 DL Delete up to Line

As for FL, but no output.

2. 2. 3 FC Find Characters in Succession

Blanks in the command string are ignored. If the characters in the command string are C_1, C_2, \dots, C_n , copy the input tape until C_1 has been copied, then copy until C_2 has been copied etc. It is not possible to match for \textcircled{L} or \textcircled{H} on the input tape. The last character output is C_n .

2. 2. 4 DC Delete up to Characters in Succession

As for FC, but no output.

2. 2. 5 FE Find End of Line

The command string is ignored. The input tape is copied until a \textcircled{L} is found, the \textcircled{L} is not punched but a switch is set so that if followed by a FL, FC or FE command the newline sequence is punched, otherwise it is ignored.

2. 2. 6 DE Delete up to End of Line

As for FE, but no output.

2. 2. 7 IS Insert String

Output the command string. The input tape is not affected. Runout may be output.

2. 2. 8 IL Insert on New Line

As for IS but a newline character sequence is output first.

2. 2. 9 IB Insert Block

Output the command string including \textcircled{L} characters. The command string terminates when \textcircled{L} is immediately followed by \uparrow . This enables a block of one or more lines to be inserted. The block to be inserted must always end with \textcircled{L} or $\textcircled{L} \textcircled{R}$. The terminating \uparrow and any characters up to and including the next L are not output. \uparrow may be output as part of the command string providing it does not immediately follow \textcircled{L} . The input tape is not affected. Runout may be output in the block.

2. 2. 10 RE Remainder of Tape

Copy the remainder of the input tape until a number of blanks is read equal to the number of blanks in the run-in, this runout is punched. The command string is ignored.

2. 2. 11 SH Stop on Halt

The input tape is copied until a \textcircled{H} is found, the program then halts. The \textcircled{H} is not copied.

2. 2. 12 IH Insert Halt

A \textcircled{H} is punched in the output tape. The command string is ignored.

2. 2. 13 ST Stop

The program halts. The command string is ignored.

2. 2. 14 CO Comment

The command is ignored.

The commands are obeyed in sequence until \textcircled{H} at the end of the correction tape is reached. The program then stops.

More than one input tape may be used. Blank run-in is always ignored although counted, so that the string of input characters is treated as if the first non-blank characters of the next tape immediately follow the characters at the end of the previous tape.

2.3 Example

Correction Tapes

FL (S) PL
 IS (S) U
 FC (S) C
 DE
 IS (S) OUT
 FL NODEC
 DE
 IS (S) (S) 10 (S) CT
 FC (S) 8
 IB
 (S) 11 (S) SPACE
 (S) 8 (S) SPACE +1
 ↑
 DC (S) T
 RE
 (H)

The IB command leaves a blank line in the output. Perhaps

IL (S) (S) 11 (S) SPACE
 IL (S) (S) 8 (S) SPACE+1
 intended?

ATHERBERT 02/04/2012

Input Tape

4 (S) OUTNO
 2 (S) +0
 PLS (S) 11 (S) DECAUT
 NODEC (S) 4 (S) MODE
 6 (S) +8
 7 (S) NOOCT
 8 (S) EXIT
 MODE (S) (S) +0
 (H)

Output Tape

4 (S) OUTNO
2 (S) +0
PLUS (S) 11 (S) DECOUT
NODEC (S) 10 (S) CT
6 (S) +8
11 (S) SPACE
8 (S) SPACE +1
8 (S) EXIT
MODE (S) (S) +0
(H)

Chapter 3: ERRORS

Whenever an error (other than a parity error) is detected, the punch outputs four inches of runout and the teleprinter prints an error message on a new line of the form

Ⓢ Ⓡ Ⓛ Ⓡ ERROR Ⓢ n Ⓢ X

where n is the error code and x the last character read from the input tape. A further two inches of blanks are then output by the punch and the program ends at a dynamic stop. If no teleprinter is present all messages are diverted to the punch.

The following errors are detected.

n=1 False command i. e. The characters defining the command are not one of the commands possible.

n=2 No command string on FL, DL, FC and DC commands.

n=3 Failure of the sum check when checking the edited tape for punching errors.

n=4 Ⓜ occurs within a command string, the data tape finishes before a command has been obeyed to completion.

Parity Errors give a dynamic stop but no output, re-entry at 3; will correct parity and continue.

Chapter 4: OPERATING INSTRUCTIONS

Load the EDIT program.

Load correction tape and enter program at 0; The correction tape is read without checking until \textcircled{H} is read, when the program ends at a dynamic stop.

Load first input tape and enter program at 1; Editing now commences with full checking.

If necessary, load continuation input tape and enter program at 2;

When editing is complete load the edited tape in the reader and enter program at 5; to check the tape produced. If the tape comes to a halt without any output, the punching is correct.

Chapter 5: STORE USED

The program uses 622 consecutive locations. The correction tape is stored in consecutive locations above the program.

Chapter 6: TIME TAKEN

EDIT operates at the speed of the tape punch.