

— CONTENTS —

<b>OUTLINE</b> .....	1
Activating the Editor .....	1
Editor Command Table .....	2
<b>CHARACTER POINTER AND DELIMITER</b> .....	3
Screen Edit .....	3
<b>TEXT EDITOR COMMANDS</b> .....	4
Input Commands .....	4
R (Read file) Command .....	4
A (Append file) Command .....	5
Output Command .....	6
W (Write) Command .....	6
Page Processing Commands .....	7
Type Command .....	10
T (Type) Command .....	10
CP Positioning Commands .....	11
B (Begin) Command .....	11
Z Command .....	11
J (Jump) Command .....	11
L (Line) Command .....	12
M (Move) Command .....	12
Correction Commands .....	13
C (Change) Command .....	13
Q (Queue) Command .....	13
I (Insert) Command .....	14
K (Kill) Command .....	15
D (Delete) Command .....	16
Search Command .....	17
S (Search) Command .....	17
Special Commands .....	18
\ Command .....	18
= Command .....	18
. Command .....	18
& Command .....	18
# Command .....	19
! Command .....	19
<b>ERROR MESSAGES</b> .....	20

1 ..... OUTLINE

1 ..... Activating the Editor

2 ..... Editor Command Table

3 ..... CHARACTER POINTER AND DELIMITER

3 ..... Screen Edit

4 ..... TEXT EDITOR COMMANDS

4 ..... Input Commands

4 ..... R (Read file) Command

5 ..... A (Append file) Command

6 ..... Output Command

6 ..... W (Write) Command

7 ..... Page Processing Commands

10 ..... Type Command

10 ..... T (Type) Command

11 ..... CP Positioning Commands

11 ..... B (Begin) Command

11 ..... Z Command

11 ..... J (Jump) Command

12 ..... L (Line) Command

12 ..... M (Move) Command

13 ..... Correction Commands

13 ..... C (Change) Command

13 ..... Q (Queue) Command

14 ..... I (Insert) Command

15 ..... K (Kill) Command

16 ..... D (Delete) Command

17 ..... Search Command

17 ..... S (Search) Command

18 ..... Special Commands

18 ..... ✓ Command

18 ..... = Command

18 ..... Command

18 ..... & Command

19 ..... # Command

19 ..... ! Command

20 ..... ERROR MESSAGES

# OUTLINE

The text editor generates, corrects and edits source files such as assembler source files and BASIC text files.

Data correction and edition are performed conversationally between the editor and programmer.

## —Activating the Editor—

The editor is activated by the following FDOS commands. (See the photograph at right.)

### 1. EDIT [CR]

The editor is activated and entry of an editor command is awaited when this FDOS command is executed.

### 2. EDIT <filename> [CR]

The editor is activated and the specified source file in the active disk drive is read when this FDOS command is executed, then the editor waits for entry of an editor command.

### 3. EDIT \$FD3; <filename> [CR]

The editor is activated and the specified source file in drive 3 is read when this FDOS command is executed.

The editor has the following correction and editing functions.

#### 1. Insertion

#### 2. Deletion

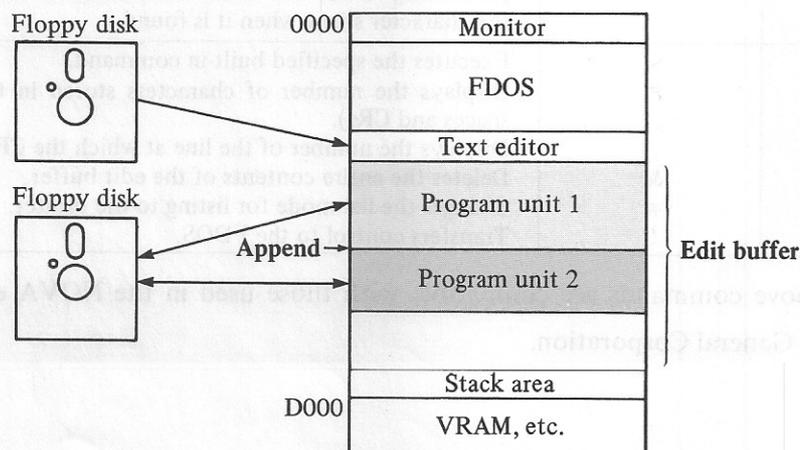
#### 3. Alteration

Each line of source text read into the edit buffer is assigned a sequential **line number**.

Positions to be corrected are specified with the **character pointer (CP)**.

Correction and edition can be performed in a units of lines or characters. Search and replacement can be performed in character string units.

The memory map is as shown below when the text editor is used.



Editor commands are listed in the following table. Commands separated from each other with the delimiter " → " and are executed when `[CR]` is entered.

Command type	Command name	Function
Input command	<b>R</b>	Clears the edit buffer and loads it with the input file indicated by the filename. The CP is positioned at the beginning of the edit buffer after execution of this command.
	<b>A</b>	Appends the input file indicated by the filename to the contents of the edit buffer. The CP position is not changed.
Output command	<b>W</b>	Writes the edit buffer contents to the output file specified by the filename in ASCII code.
Page processing command	<b>PR</b> <b>PA</b>	Loads the remainder of a file whose beginning has been loaded with the R or A command. The PR command clears the edit buffer before the data is loaded and the PA command adds the data to the current contents of the edit buffer.
	<b>PW</b>	Same as the W command, except that the maximum amount of data output is 1 page.
	<b>PC</b>	Terminates execution of the processing command. This command is required whenever a PR, PA or PW command is executed.
	<b>PK</b>	Kills the file opened by a page processing command.
Type command	<b>T</b>	Displays the entire contents of the edit buffer. The CP position is not changed.
	<b>nT</b>	Displays n lines starting at the CP position.
CP positioning command	<b>B</b>	Positions the CP at the beginning of the edit buffer.
	<b>nJ</b>	Positions the CP at the beginning of the line indicated by n.
	<b>nL</b>	Moves the CP to the beginning of the line n lines after the current CP position.
	<b>L</b>	Moves the CP to the beginning of the current line. This is the same as when n = 0 in the nL command.
	<b>nM</b>	Changes the CP position by n characters.
	<b>M</b> <b>Z</b>	Does not move the CP. This is the same as when n = 0 in the nM command. Moves the CP to the end of the text in the edit buffer.
Correction command	<b>C</b>	Searches for the specified character string and replaces it with another character string; the search starts at the current CP position and proceeds to the end of the edit buffer. The CP is repositioned to the end of the character string replaced.
	<b>Q</b>	Repeats the C command each time the specified character string is found until the end of the edit buffer is reached. The CP is repositioned to the end of the character string last replaced.
	<b>I</b>	Inserts the specified character string at the position of the CP. The CP is repositioned to the end of the character string inserted. Line numbers are updated when a line is inserted with this command.
	<b>nK</b>	Deletes the n lines following the CP. The CP position is not changed.
	<b>K</b>	Deletes all characters preceding the CP position until a <code>[CR]</code> code is detected. The <code>[CR]</code> code is not deleted.
	<b>nD</b> <b>D</b>	Deletes the n characters following the CP. No operation
Search command	<b>S</b>	Searches for the specified character string, starting at the CP position and proceeding to the end of the buffer. The CP is repositioned to the end of the character string when it is found.
Special command	<b>\</b>	Executes the specified built-in command.
	<b>=</b>	Displays the number of characters stored in the edit buffer (including spaces and CRs).
	<b>.</b>	Displays the number of the line at which the CP is located.
	<b>&amp;</b>	Deletes the entire contents of the edit buffer.
	<b>#</b> <b>!</b>	Changes the list mode for listing to the printer. Transfers control to the FDOS.

Most of the above commands are compatible with those used in the NOVA editor program manufactured by the Data General Corporation.

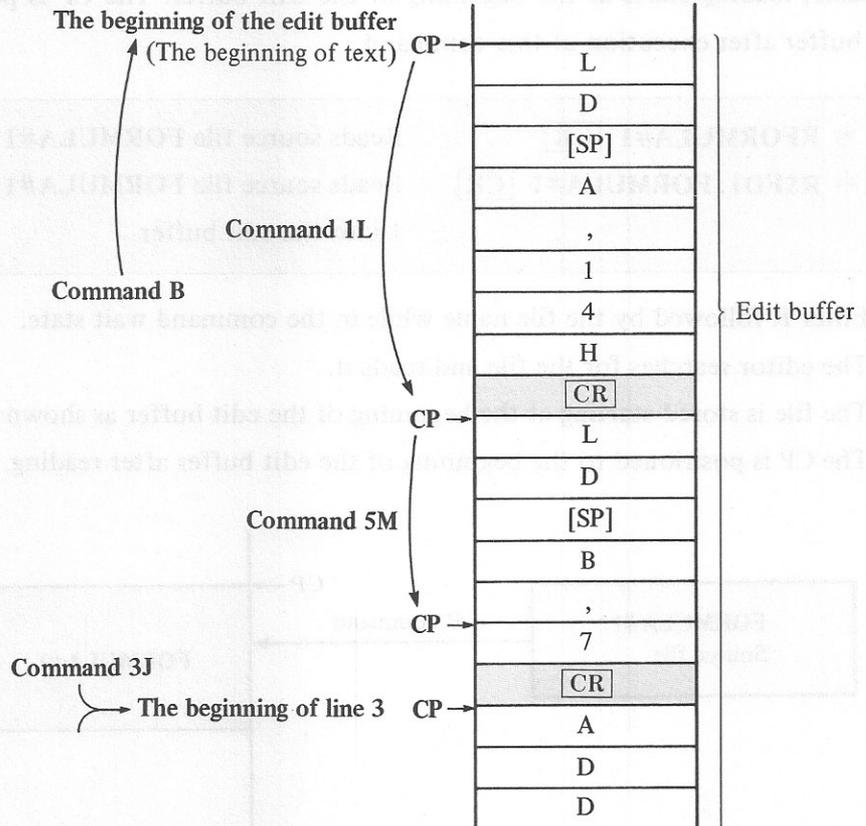
# CHARACTER POINTER AND DELIMITER

The character pointer (CP) is positioned at the boundary between two adjacent characters or the beginning or end of the text. It does not point directly at any character.

Movement of the CP is explained below based on the assumption that the following text is stored in the edit buffer.

```
1 LD A, 14H
2 LD B, 7
3 ADD A, B
4 DAA
```

(Line numbers are not stored in the edit buffer.)



The B command moves the CP to the beginning of the edit buffer, the J command moves it to the top of the specified line and the L command to the beginning of the nth line from the line in which the CP is currently located; the top of the specified line is the boundary following the `CR` code of a preceding line.

**Delimiters** are used as separators between editor commands. Entering several editor commands and separating them with delimiters allows them to be executed consecutively by pressing the `CR` key.

The delimiter " → " is entered when `CTRL` + `Z` are pressed.

The **I (Insert)** command must be followed by a delimiter because it uses `CR` codes as character codes for the source text.

The following example replaces ADD on line 3 in the above program with ADC.

`3J → 2M → 1D → IC → CR` or `B → CADD → ADC CR`

## —Screen Edit—

The line on which the cursor is located can be modified by entering new data from the keyboard and pressing the `CR` key after data has been displayed by executing the T, C, Q or S command. The CP is positioned at the end of the updated line.

Note that the line number may be changed when the I, D or K command is executed.

# TEXT EDITOR COMMANDS

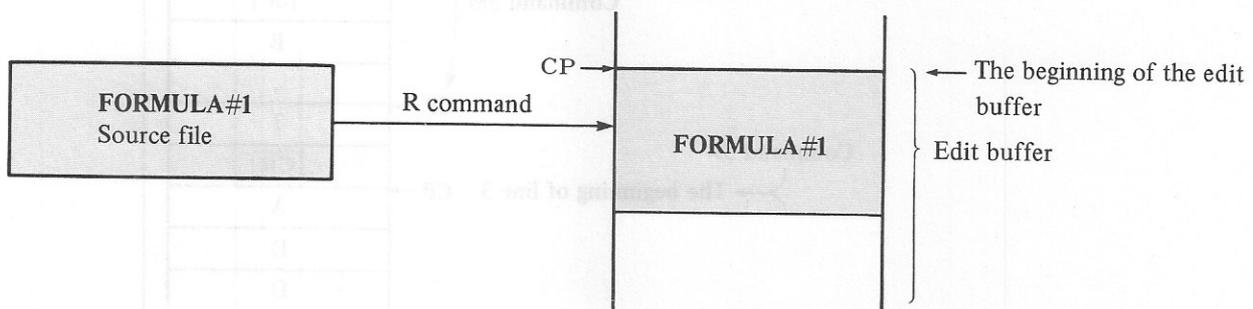
## — Input Commands —

### R (Read file) Command

This command clears the edit buffer area, then loads it with the source file (ASCII file) specified by the filename; loading starts at the beginning of the edit buffer. The CP is positioned at the beginning of the edit buffer after execution of this command.

- |                    |      |  |
|--------------------|------|--|
| * RFORMULA#1       | [CR] | Reads source file FORMULA#1 into the edit buffer.                              |
| * R\$FD1;FORMULA#1 | [CR] | Reads source file FORMULA#1 from the diskette in drive 1 into the edit buffer. |

- Enter R followed by the file name while in the command wait state.
- The editor searches for the file and reads it.
- The file is stored starting at the beginning of the edit buffer as shown below.
- The CP is positioned to the beginning of the edit buffer after reading.



- The message "Full buffer" is displayed when the edit buffer becomes full. In this case, only part of the input file is stored in the edit buffer.
- The page processing mode is entered automatically when the buffer becomes full. Therefore, the remainder of the input file can be read by the PR command.

Note: "\*" is displayed to indicate that the system is in the command wait state.

## A (Append file) Command

This command appends the file specified by the filename to the contents of the edit buffer. The CP position is not changed.

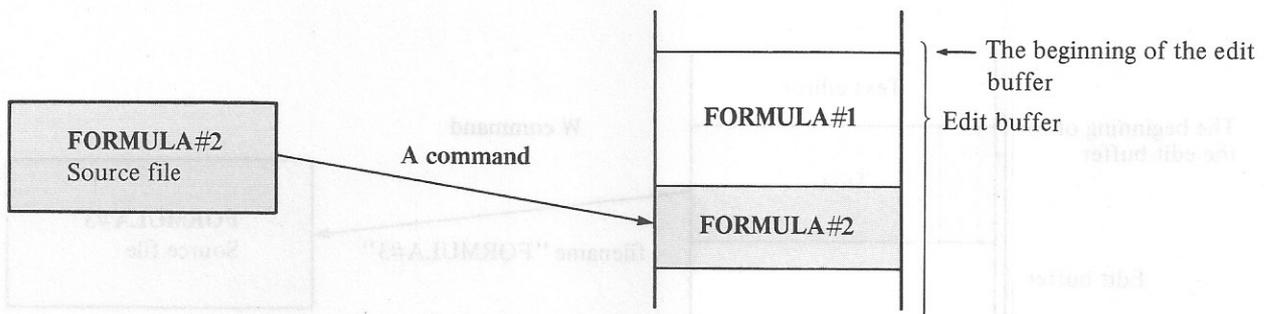
\* **A**FORMULA#2 **CR**

Appends source file **FORMULA#2** to the contents of the edit buffer.

\* **A\$FD2;FORMULA#2** **CR**

Appends source file **FORMULA#2** from the diskette in drive 2 to the contents of the edit buffer.

- Enter **A** followed by the file name while in the command wait state.
- The editor searches for the specified file and reads it.
- The file is stored in the area following the end of the last text in the edit buffer. The figure below shows a case in which the file **FORMULA#2** is appended to the file **FORMULA#1**.



- The message "Full buffer" is displayed when the edit buffer becomes full. In this case, only part of the specified file is stored in the edit buffer and the contents of the edit buffer must be reedited to store the entire file.
- The page processing mode is entered automatically when the buffer becomes full. Therefore, the remainder of the input file can be read by the **PR** command.

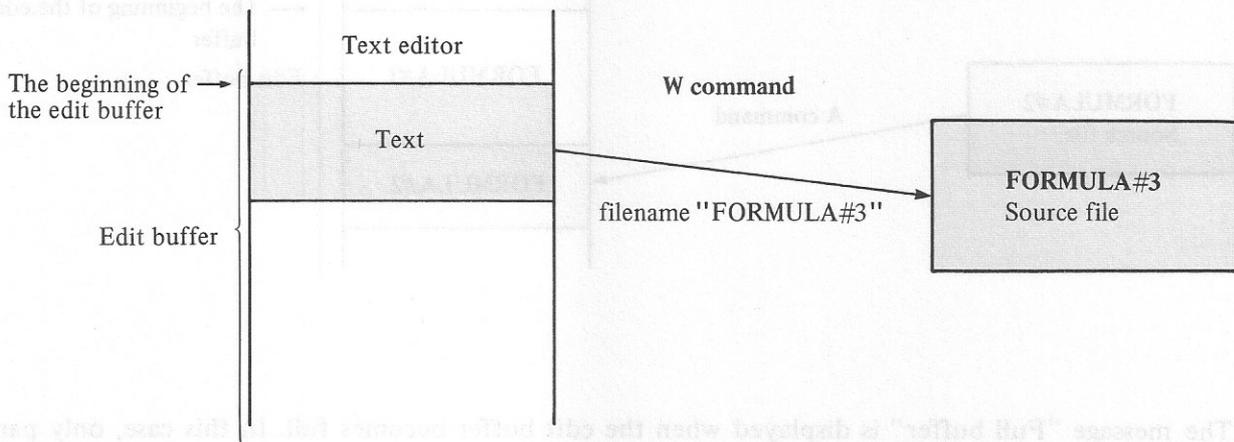
## —Output Command—

### W (Write) Command

This command outputs the entire contents of the edit buffer to the file specified by the filename regardless of the CP position.

- \* **WFORMULA#3** **CR** Outputs the contents of the edit buffer to file **FORMULA#3** in the active drive.
- \* **WSFD2;FORMULA#3** **CR** Outputs the contents of the edit buffer to file **FORMULA#3** in floppy disk drive 2 (\$FD2).

- Enter W followed by the filename while in the command wait state.
- The editor waits for entry of another command after the edit buffer contents have been output. The file generated is a source file.



- The CP position is not changed by execution of the W command.

## —Page Processing Commands—

These commands are used in cases where the total size of files to be edited exceeds the size of the edit buffer, as shown in the following examples.

If the diskette is replaced with a new one during page processing, the contents of the diskette may be destroyed. Be sure to terminate page processing before replacing the diskette.

### 1. When several files are to be edited and the resulting file is larger than the edit buffer:

① \* RABC CR Reads source file "ABC" into the edit buffer.

② \* ADEF CR Reads source file "DEF" and appends it to the contents of the edit buffer until the buffer is full. Message "Full buffer" is displayed.

Note: If \* RDEF CR is entered, the contents of the "ABC" stored in the edit buffer are cleared and file "DEF" is loaded in the edit buffer from its beginning.

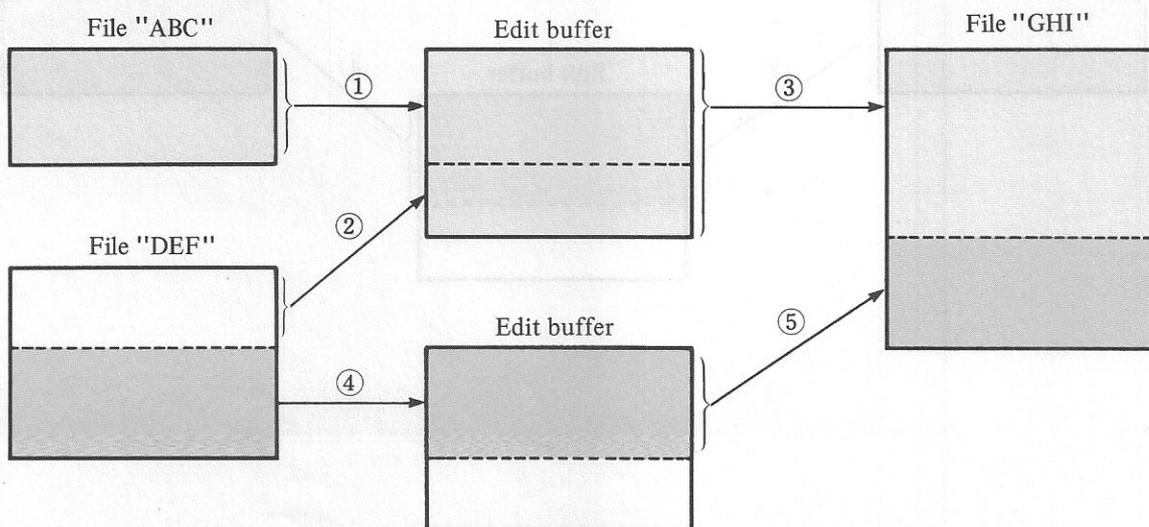
③ \* PWGHI CR Outputs the contents of the edit buffer to file "GHI" after editing is completed.

④ \* PR CR Reads the remainder of a file specified by a preceding PR command into the edit buffer. In this example, the command reads the remainder of file "DEF" into the edit buffer.

Note: File name "DEF" specified in step ② remains valid.

⑤ \* PW CR Outputs the contents of the edit buffer and appends it to the file specified by the preceding PW command after editing is completed. In this example, the command appends the contents of the edit buffer to file "GHI."

⑥ \* PC CR Terminates page processing. (This command is mandatory.)



2. When the file to be edited is larger than the edit buffer:

① \* RABC [CR]

Reads source file "ABC" into the edit buffer until it is full. Message "Full buffer" is displayed.

Note: Omission of the file name in the first page processing command will result in error.

② \* WDEF [CR]

Outputs the contents of the edit buffer to file "DEF" after editing is completed.

Note: \*PW [CR] results in an error. An error results when editing increases the size of file in the edit buffer so that it exceeds the size of the edit buffer.

③ \* PR [CR]

Reads the remainder of file "ABC" into the edit buffer.

Note: A file name specified in step ① remains valid until a new file name is specified.

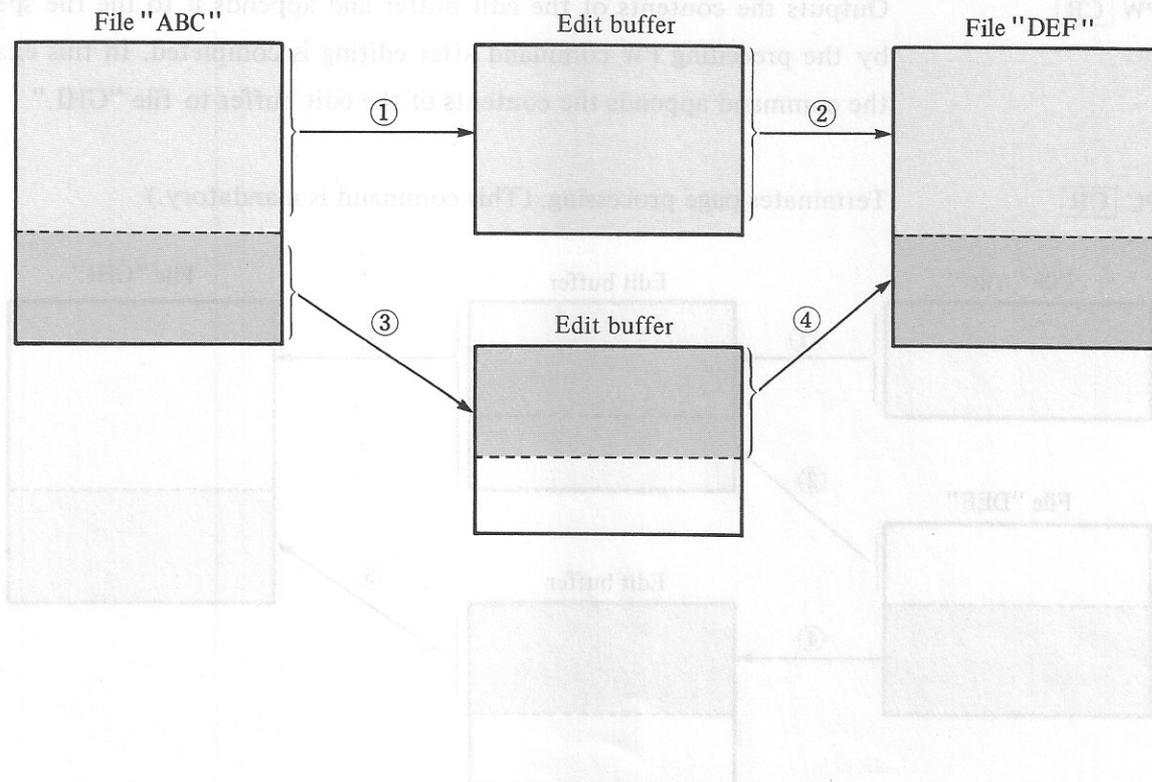
④ \* PW [CR]

Appends the contents of the edit buffer to file "DEF" after editing is completed.

⑤ \* PC [CR]

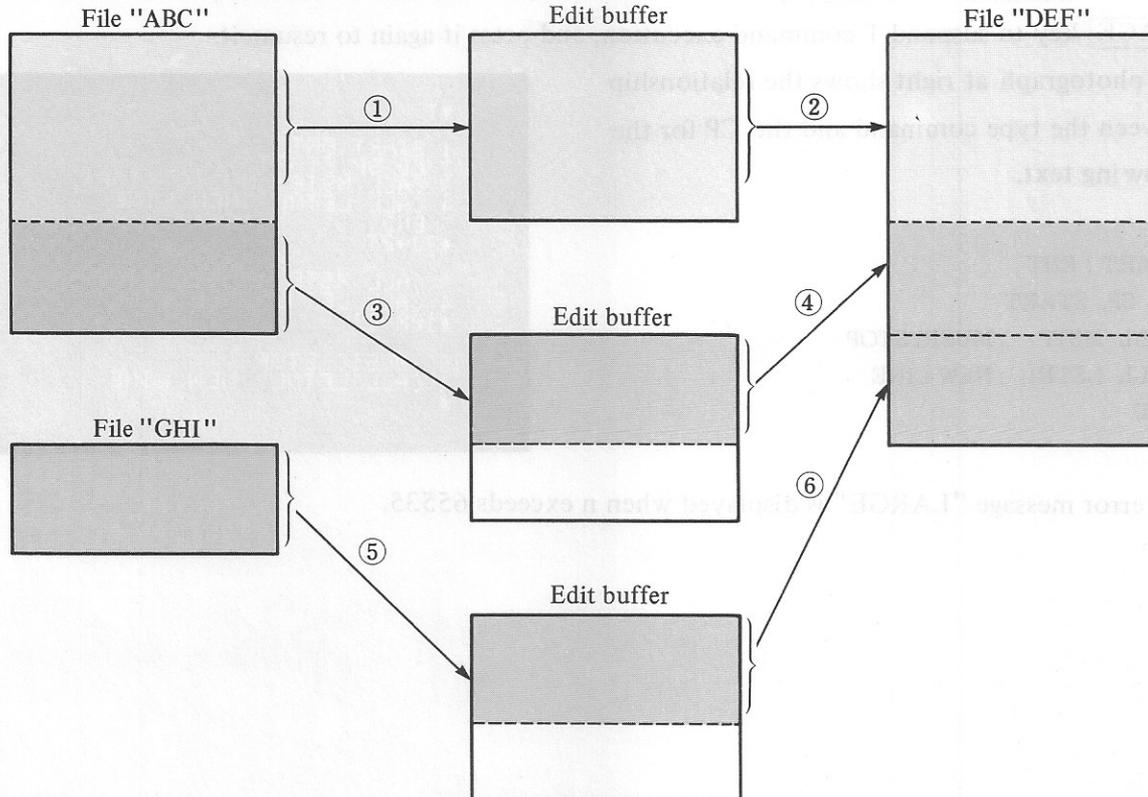
Terminates page processing. (This command is mandatory.)

⑥ Repeat steps ③ and ④ when file "ABC" is too large to be processed by performing steps ③ and ④ once.



3. When the file to be edited first is larger than the edit buffer and another file is to be edited and appended to the first edited file:

- ① \* RABC CR Reads file "ABC" into the edit buffer until the buffer is full.
- ② \* PWDEF CR Outputs the contents of the edit buffer to file "DEF" after editing is completed.
- ③ \* PR CR Reads the remainder of file "ABC" into the edit buffer.
- ④ \* PW CR Appends the contents of the edit buffer to file "DEF" after editing is completed.
- ⑤ \* RGHI CR Reads file "GHI" into the edit buffer.  
 Note: In this case, specifying \*PR CR will not be valid if the end of file "ABC" has been reached.
- ⑥ \* PW CR Appends the contents of the edit buffer to file "DEF" after editing is completed.
- ⑦ \* PC CR Terminates page processing. (This command is mandatory.)



4. When a file which was opened by a page processing command is to be killed:

- ① \* PK CR Kills the file opened by a page processing command.

## —Type Command—

### T (Type) Command

This command displays all or part of the contents of the edit buffer. The CP position is not changed.

- \* T **[CR]** Displays all of the contents of the edit buffer with line numbers attached.
- \* nT **[CR]** Displays the n lines following the CP. (Same as the above when n = 0.)

— Key in the number of lines, n followed by T (Type) while in the command wait state.

— Press **[CR]** to display the contents of the edit buffer.

— The following are special cases of nT.

n = 0 : the same as T

n < 0 : results in the error message "???"

n ≥ m, where m is the number of lines from the one at which the CP is located to the end of the buffer contents: only m lines are displayed.

— The current CP position can be determined with the nT command, since display starts with the character following the boundary at which the CP is located.

— Press the **[SHIFT]** and **[BREAK]** key simultaneously to terminate T command execution. Press the **[SPACE]** key to suspend T command execution, and press it again to resume it.

— The photograph at right shows the relationship between the type command and the CP for the following text.

```
START : ENT
LD SP, START
CALL MSTP ;MUSIC.STOP
CALL LETNL ;NEW LINE
END
```



```
*T
1 START:ENT
2 LD SP,START
3 CALL MSTP ;MUSIC STOP
4 CALL LETNL ;NEW LINE
5 END
*3J→2T
3 CALL MSTP ;MUSIC STOP
4 CALL LETNL ;NEW LINE
*10M→2T
3 ;MUSIC STOP
4 CALL LETNL ;NEW LINE
*
```

— The error message "LARGE" is displayed when n exceeds 65535.

## —CP Positioning Commands—

### B (Begin) Command

\* B  Positions the CP to the beginning of the edit buffer.

- Key in B while in the command wait state.
- Press .
- The B command is executed to position the CP to the beginning of the edit buffer.
- nB performs the same function.

### Z Command

\* Z  Positions the CP to the end of the contents of the edit buffer.

- Key in Z while in the command wait state.
- Press .
- The Z command is executed to position the CP to the end of the contents of the edit buffer.
- nZ performs the same function.

### J (Jump) Command

\* nJ  Positions the CP to the beginning of line n.

- Key in line number n and J while in the command wait state.
- Press .
- The nJ command is executed to position the CP to the beginning of line n.
- The following are special cases.

n = 0 or 1 or n is omitted: the command performs the same function as the B command.

n < 0 : results in the error message "???".

n ≥ m (where m is the number of lines of the edit buffer contents): the command performs the same function as the Z command.

## L (Line) Command

This command moves the CP forward or backward the specified number of lines. The CP is positioned at the beginning of the specified line after execution.

- \* **nL** **[CR]** Moves the CP to the beginning of the nth line from the line at which it is currently located.
- \* **L** **[CR]** Moves the CP to the beginning of the line at which it is currently located.

- Key in number of lines, **n** and **L** while in the command wait state.
- Press **[CR]**.
- The CP is positioned at the beginning of the specified line when the nL command is executed.
- The following are special cases:

- $n = 0$  : the command functions in the same manner as the L command.
- $n \geq m$  (where  $m$  is the number of lines from the line on which the CP is located to the end of the edit buffer contents): the command functions in the same manner as the Z command.
- $n < 0$ : the CP is moved  $|n|$  lines toward the beginning of the edit buffer.
  - $|n| \geq \ell - 1$  (where  $\ell$  is the number of the line at which the CP is currently located): the command functions in the same manner as the B command.

## M (Move) Command

This command moves the CP forward or backward by the specified number of characters. Spaces and carriage returns are counted as characters, but line numbers are not.

- \* **nM** **[CR]** Moves the CP to the position which is **n** characters from its current position.

- Key in number of characters, **n** and **M** while in the command wait state.
- Press **[CR]**.
- Executing the nM command moves the CP to the specified boundary between characters.
- When  $n < 0$ , the CP is moved backward by  $|n|$  characters.
- The CP position is not changed when  $n = 0$  or if it is omitted.

## —Correction Commands—

### C (Change) Command

This command replaces a string in the edit buffer with another string. The search for the specified string starts at the current CP position and proceeds toward the end of the edit buffer; the string is replaced when it is found and the CP is positioned at the end of the string replaced.

- \* Cstring 1 → string 2 **[CR]** Searches for the character string specified by **string 1**, starting at the current CP position and proceeding toward the end of the edit buffer; replaces the string with the one specified by **string 2** when it is found.
- \* Cstring 1 **[CR]** Deletes the character string specified by **string 1**.

- Key in C while in the command wait state.
- Key in the string to be located followed by a delimiter → ( **[CTRL]** + **[Z]** ).
- Key in the string which is to replace the one located.
- Press **[CR]** and a search is made for the first string. Only the first occurrence of the string is replaced. The line including the string replaced is displayed and the CP is positioned at the end of that string.
- The message "Not found" is displayed if the specified string is not found and the CP is positioned to the beginning of the edit buffer.

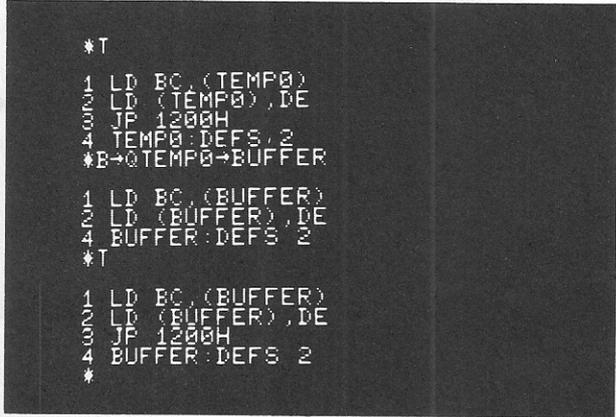
### Q (Queue) Command

This command repeats the function of the C command each time the specified character string is found until the end of the edit buffer is reached. The CP is repositioned to the end of the string last replaced.

- \* Qstring 1 → string 2 **[CR]** Causes the function of the C command to be executed repeatedly.
- \* Qstring 1 **[CR]** Deletes all occurrences of the character string specified by **string 1**.

- Key in Q while in the command wait state.
- The remainder of the operation is the same as for the C command.
- The photograph at right shows the result of execution of the Q command on the following text.

```
LD BC, (TEMPO)
LD (TEMPO), DE
JP 1200H
TEMPO : DEFS 2
```



```
*T
1 LD BC,(TEMPO)
2 LD (TEMPO),DE
3 JP 1200H
4 TEMPO:DEFS 2
*B->QTEMPO->BUFFER
1 LD BC,(BUFFER)
2 LD (BUFFER),DE
3 JP 1200H
4 BUFFER:DEFS 2
*T
1 LD BC,(BUFFER)
2 LD (BUFFER),DE
3 JP 1200H
4 BUFFER:DEFS 2
*
```

## I (Insert) Command

This command inserts the specified string at the CP position. A carriage return is performed on the CRT screen if one is included in the string.

Line numbers are updated automatically when a new line is inserted. The CP is repositioned to the end of the string inserted.

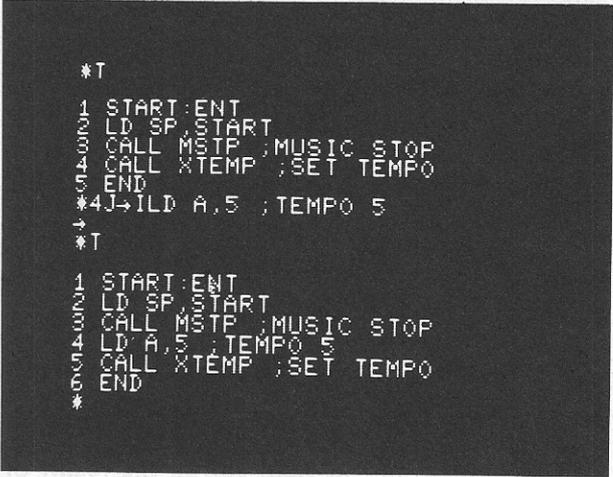
- \* Istring → **CR** Inserts the specified string at the CP position.
- \* Istring 1 **CR** Inserts the lines specified by string 1, string 2 and string 3 at the CP position.
- string 2 **CR**
- string 3 **CR**
- **CR** A **CR** is treated as a character by the I command. Therefore, a delimiter must be keyed in before **CR** is pressed to execute the command.

- Key in I while in the command wait state.
- Key in the string to be inserted.
- Strings keyed in are inserted at the CP position and the contents of the edit buffer following the CP are automatically shifted toward the end of the edit buffer.
- When a **CR** is keyed in, it is inserted as a carriage return code.
- Key in a delimiter → ( **CTRL** + **Z** ) after all the strings have been keyed in.
- Press **CR** key to execute the I command.
- The photograph at right shows an example of using the I command.

Text:

```
1 START : ENT
2 LD SP, START
3 CALL TIMST ;TIMER SET
4 CALL XTEMP ;SET TEMPO
5 END
```

LD A, 5 ;TEMPO 5 is inserted between lines 3 and 4 of the above text.



```
*T
1 START:ENT
2 LD SP,START
3 CALL MSTP ;MUSIC STOP
4 CALL XTEMP ;SET TEMPO
5 END
*4J→ILD A,5 ;TEMPO 5
→
*T
1 START:ENT
2 LD SP,START
3 CALL MSTP ;MUSIC STOP
4 LD A,5 ;TEMPO 5
5 CALL XTEMP ;SET TEMPO
6 END
*
```

## K (Kill) Command

This command deletes the  $n$  lines preceding or following the CP from the edit buffer.

- \* **nK** `[CR]` Deletes the  $n$  lines preceding or following the CP from the edit buffer. A line is not deleted if the CP is located within it, since characters preceding or following the CP are not deleted.
- \* **K** `[CR]` Deletes characters preceding the CP position until a `[CR]` is detected. The `[CR]` is not deleted.

— Key in the number of lines,  $n$  and **K** while in the command wait state.

— Press `[CR]` to execute the K command.

— Operation differs according to the value of  $n$  as follows.

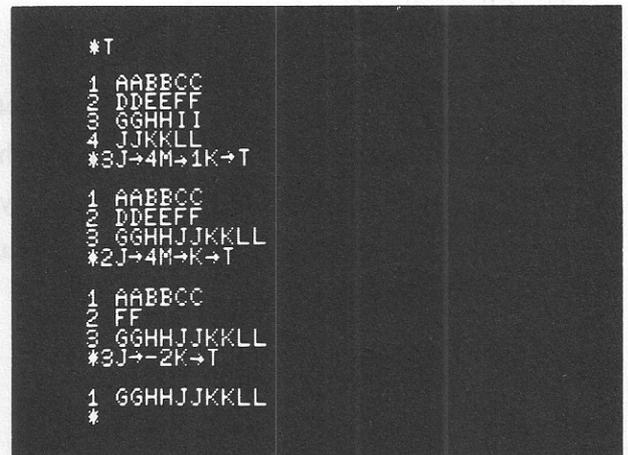
- $n > 0$  : Deletes all characters following the CP until  $n$  `[CR]` codes are detected. `[CR]` codes detected are also deleted. Command execution ends after the last `[CR]` code has been deleted.
- $n < 0$  : Deletes all characters preceding the CP until  $|n| + 1$  `[CR]` codes are detected. The  $(|n| + 1)$ th `[CR]` code is not deleted.
- $n = 0$  or not specified : Deletes all characters preceding the CP until a `[CR]` code is detected. That is, deletes the part of the line in front of the CP. The `[CR]` code detected is not deleted.

— Line numbers are automatically updated after deletion.

— The CP position is not changed.

— The photograph at right shows an example of the result of execution of the K command with the following text. (This text is presented only for the purpose of illustration; it has no meanings in assembly language.)

```
1 AABBC
2 DDEEFF
3 GGHHII
4 JJKLL
```



## D (Delete) Command

This command deletes the specified number of characters from the edit buffer, starting at the CP position.

- \* **nD** CR Deletes the specified number of characters from the edit buffer, starting at the CP position.  
A CR code is counted as a character.
- \* **D** CR (No operation results.)

— Key in the number of character **n** and **D**.

— Press CR to execute the command.

— Operation differs according to the value of **n** as follows.

**n** > 0 : Deletes the **n** characters following the CP from the edit buffer.

A CR code is counted as a character.

**n** < 0 : Deletes the |**n**| characters preceding the CP from the edit buffer.

A CR code is counted as a character.

**n** = 0 or No operation results.

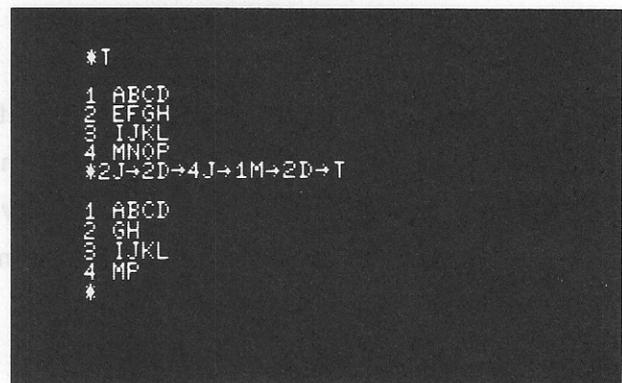
not specified

— Line numbers are automatically updated if necessary.

— The CP position is not changed.

— The photograph at right shows an example of the result of execution of the D command with the following text. (This text is presented only for the purpose of this illustration; it has no meaning in assembly language.)

```
1 ABCD
2 EFGH
3 IJKL
4 MNOP
```



```
*T
1 ABCD
2 EFGH
3 IJKL
4 MNOP
*TJ-2D-4J+1M+2D-T
1 ABCD
2 GH
3 IJKL
4 MP
*
```

## —Search Command—

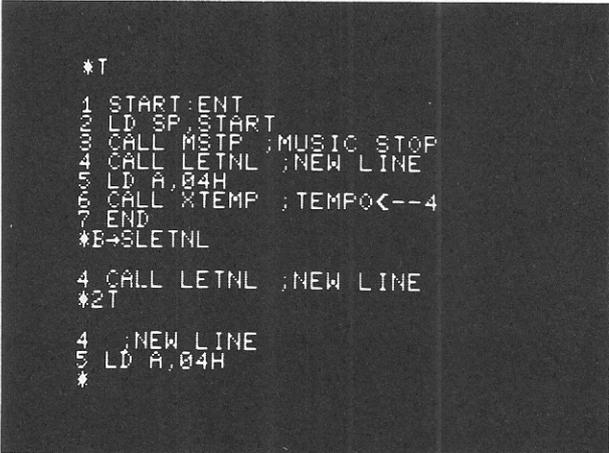
### S (Search) Command

This command searches for the specified character string in the contents of the edit buffer.

\* S string **[CR]** Searches for the specified character string, starting at the current CP position; the CP is repositioned to the end of the character string when it is found.

- Key in S.
- Key in the string to be located.
- Press **[CR]** to execute the S command.
- The search starts at the current CP position and proceeds toward the end of the buffer.
- When the specified string is found, the line which includes it is displayed and the CP is positioned to the end of the character string.
- If the specified string cannot be found, the message "Not found" is displayed and the CP is repositioned to the beginning of the edit buffer.
- The photograph at right shows the result of a search for the character string "LETNL" in the following text. The line including "LETNL" is displayed following the S command. The 2T command indicates that the CP is positioned to the end of the string.

```
1 START : ENT
2 LD SP, START
3 CALL MSTP ; MUSIC STOP
4 CALL LETNL ; NEW LINE
5 LD A, 04H
6 CALL XTEMP ; TEMPO < -- 4
7 END
```



```
*T
1 START:ENT
2 LD SP,START
3 CALL MSTP ;MUSIC STOP
4 CALL LETNL ;NEW LINE
5 LD A,04H
6 CALL XTEMP ;TEMPO<--4
7 END
*B->SLETNL
4 CALL LETNL ;NEW LINE
*2T
4 ;NEW LINE
5 LD A,04H
*
```

## —Special Commands—

### \ Command

This command executes the specified built-in command. "\*" is displayed to indicate that command entry is awaited after execution.

\*\DELETE ABC  Deletes file ABC.ASC.

- Enter the \command when "\*" appears to indicate that command entry is awaited.
- Specify the built-in FDOS command and press the  key; the command is then executed.
- The XFER and EXEC commands cannot be executed. The RUN command cannot be executed when the program is too long to be executed.
- No built-in FDOS commands can be executed in the page processing mode. If an attempt is made to execute one in the page processing mode, the message "Page opened!" is output. In this case, reset the page processing mode with the PC or PK command.

### = Command

\* =  Displays the total number of characters (including spaces and CRs) stored in the edit buffer.

- Key in "=" (equal) while in the command wait state.
- Press ; the total number of characters stored in the edit buffer is displayed.

### . Command

\* .  Displays the number of the line on which the CP is located.

- Key in . (period) while in the command wait state.
- Press ; the line number on which the CP is located is displayed.

### & Command

\* &  Clears the contents of the edit buffer.

- Key in & (ampersand) while in the command wait state.
- Press ; the contents of the edit buffer are then cleared.

## # Command

\* # **CR** Changes the printer list mode.

- Key in # (sharp symbol) while in the command wait state.
- Press **CR**; the printer list mode is then changed.
- The printer list mode is disabled when the text editor is started. It is enabled when the # command is executed once; executing it again disables it, and so on.
- The following shows a listing obtained by executing the T command when the printer list mode is enabled.

```
1 ; TYPE COMMAND
2 ;
3 .TYPE:ENT
4 LD DE,SWTBL; DE:=SWITCH TABLE
5 CALL ?GSW; CHECK GLOBAL SWITCH
6 RET C
7 CALL C&L1 ; SELECT CRT OR LPT
8 CALL ?SEP ; CHECK SEPARATER
9 RET C
10 CP 2CH ; SEPARATER="," ?
11 LD A,3 ; 3 IS ERR CODE
12 SCF
13 RET NZ ; NO, ERR RETURN
14 TYPEO:CALL ?LSW ; CHECK LOCAL SWITCH
15 RET C
16 LD A,8 ; 8 IS ERR CODE
17 SCF
18 RET NZ ; ERROR, LSW EXIST
19 LD C,128 ; LU#:=128
20 EXX
21 LD B,4; DEFAULT MODE=ASC
22 EXX
23 CALL ROPEN ; READ-OPEN
24 RET C
25 CALL %NL
26 JR C,TYPEER
27 CALL TESW ; TEST GLOBAL SWITCH
28 DEFB 88H; /P
29 CCF
30 CALL C,PPAGE; LPT PAGING
31 JR C,TYPEER
32 CALL MODECK ; FILEMODE CHECK
33 DEFB 4 ; .ASC ?
```

## ! Command

\* ! **CR** Returns control to FDOS.

- Key in ! (exclamation mark) while in the command wait state.
- Press **CR**; control is then returned to FDOS.

# ERROR MESSAGES

Error message	Explanation	Related commands
<b>Full buffer</b>	The edit buffer is full.	R, A, PR, PA
<b>???</b>	$n < 0$ in the nT or nJ command.	T, J
<b>Large</b>	n greater than 65535 was specified.	T, J, L, M, K, D, B, Z
<b>Not found</b>	The string specified in the command was not found.	S, C, Q
<b>Invalid</b>	Other than an editor command was entered or an incorrect format was used. Ex) *H <span style="border: 1px solid black; padding: 0 2px;">CR</span> : There is no H command. *S <span style="border: 1px solid black; padding: 0 2px;">CR</span> : A string should be specified.	Any case
<b>Page opened ?</b>	The file to be subjected to page processing is not defined (or is not opened).	PR, PA, PW
<b>Page opened !</b>	An attempt to execute the ! or \command was made on a file which was subjected to page processing, but which was not closed or killed by the PC or PK command.	!, \
<b>No file is saved after edition End of job?</b>	These messages are displayed where an attempt is made to execute a ! command after the edit buffer contents have been corrected without first executing a W or PW command. Pressing the <span style="border: 1px solid black; padding: 0 2px;">Y</span> key in this case executes the ! command. Pressing the <span style="border: 1px solid black; padding: 0 2px;">N</span> key causes the system to await entry of a new editor command.	!

**Note:** Refer to the System Error Messages in the System Command manual for the system errors. Display of the message "Already opened" during execution of a W command indicates that there is a PW command which has not been closed.