TSC 6800 Text Editing System

Technical Systems Consultants, Inc.
P.O. Box 2574
West Lafayette, Indiana 47906

INTRODUCTION

Contained in the following pages is a complete description and source listing of the TSC Text Editing System: This system is a content oriented text editor which is powerful, simple to use, and easy to learn. Particular attention should be paid to the section titled "Adapting to Your System".

As in all TSC software, a great effort has been put forth in testing to eliminate "bugs" in the code. This however is no guarantee of perfect code. If a suspected bug is spotted, please jot down the circumstances involved and send it to TSC. Errata sheets with all patches will be sent to owners if necessary.

GETTING THE SYSTEM STARTED

The general use of the disk version is completely described in the section of this manual titled 'Using the Disk Version'. The user should first read the Mini Tutorial section, then try to load and run the editor from disk.

MINI-TUTORIAL

The purpose of this section is to briefly introduce the reader to the use of the TSC Text Editing System. We will, therefore, illustrate its use with a number of examples. In order to make it more obvious what things are typed by the user and what things are displayed by the editor, we will subscribe to the convention that things underlined are user-typed and things not underlined are displayed by the editor.

When the editor is initially entered, the response is as shown above. At this time we will create our file by simply typing all lines until finished, terminating each line with a "carriage return".

```
NEW FILE:
    1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
    2.00 =TSC TEXT EDITING SYSTEM. A NUMBER OF
    3.00 =EXAMPLES WILL BE SHOWN TO ALLOW EASY AND
    4.00 =QUICK LEARNING OF ITS FEATURES.
    5.00 =FOLLOWING ARE SOME NONSENSE LINES:
    6.00 =ABCDEFGHIJKL
    7.00 =AAAAAAAA
    8.00 =TESTING 1234
    9.00 =THIS EDITOR IS FUN TO USE!
   10.00 =BBBBBBBB
   11.00 =
   12.00 =THIS IS THE END OF THIS FILE,
   13.00 =AT LEAST FOR NOW.
   14.00 =#
   13.00 =AT LEAST FOR NOW.
#
```

Notice it was necessary to type a pound sign (#) in column one to leave the buffered input mode. At this time, the system printed the last line and returned with the system prompt (a pound sign). The editor is now ready to accept commands.

Any time characters are being typed into the editor the following two characters have special meaning:

1. "control" H - Deletes the last character typed (backspace).
2. "control" X - Deletes entire current line being typed.

These are useful, when detected typing errors occur, for immediate correction.

Each line of text in the edit file is given or has a line number which is used by the editor to uniquely identify the line. Each line number is of the form "m.nn" where "m" is an integer and "n" represents any of the digits 0 through 9. To specify a line number, one has to specify only that portion of the line number to identify it uniquely. For example, 73, 73., 73.0, and 73.00 may be used to refer to line 73.00; 259.6 refers to line 259.60. The largest line number used with the editor is 9999.99. Let's

denote a specification of a line of text by the symbol "<line>". We will be using this symbol throughout this document.

An editor command tells the editor what action is to be performed and usually what line or block of lines are to be affected (if any). For each editing facility supported by the editor, there is a directive which is used in commands to indicate the desired action. For example, the editor can delete lines of text from a file, insert lines of text into the file, print lines contained in the file, and so on. Corresponding to each capability there is a directive; hence, there is a Delete directive, an Insert directive, a Print directive, and so on. If we define the symbol <directive> to mean any editor directive, the basic from of an edit command is

<line><directive>

For example, the command to display (Print) line 12.00 is

```
*12 P
   12.00 =THIS IS THE END OF THIS FILE.
*
```

where "12" is the <line> specification and "P" is the <directive> in this command. As can be seen in the example, this causes line number 12 to be printed on the terminal.

Now, let's learn how to use the insert directive. In normal usage of the word "insert" we say something like, "Insert this card after this other card". To use the Insert directive, we specify the line after which we want to insert new lines followed by an I:

<line>I

After typing the directive followed by a carriage return, the editor will select an appropriate line number and prompt for input by displaying the line number followed by an equal sign. After each line of text is entered and the carriage return is typed, the editor will prompt for the next line. To exit from the "Insert mode" one simply types a pound sign followed by an edit directive in response to a new line prompt.

Some examples of the use of Insert are

```
*8I
    8.10 =THIS IS AN INSERTED LINE.
    8.20 =SO IS THIS.
    8.30 =#11 I
   11.10 =ANOTHER INSERTED LINE.
   11.20 =#6 P
    6.00 =ABCDEFGHIJKL
```

It should be noted that the editor may renumber some lines following the inserted text. This occurs when enough lines are inserted such that the inserted line numbers overlap line numbers in the original text.

Next, let's learn how to use the Delete directive. With this directive we can delete one line or a block of lines with one directive. To delete only one line, we specify the <line> to be deleted followed by a D:

```
<line>D
```

When the carriage return is typed, the line disappears.

To delete more than one line we need to indicate not only the first line to delete but also the last line to be deleted. Let's call the last line the "target" line and denote its specification as "<target>". Although the editor supports fancier ways to specify the <target>, we'll just consider the two simplest: (1) <tartget> may be the number of lines to be deleted (counting both the first and last line of the block), or (2) <target> may be a pound sign followed immediately by the line number of the last line of the block to be deleted. Some example <target>s are: 3 (deletes three lines), 26 (delete 26 lines), and #26 (delete lines through line 26.00).

The syntax to Delete a block of lines is

```
<line>D <target>
```

where <line> indicates the first line to delete and <target> indicates the scope of the delete.

To illustrate the use of the Delete directive, let's assume we have a file containing 53 lines with integer line numbers (i.e., 1, 2, 3, ...,53). With the directives

```
#15D
#24D #31
#52D 2
BOTTOM OF FILE REACHED
#
```

we now have a file with lines 1 through 14, 16 through 23, and 32 through 51. The first directive deleted line 15. The second directive deleted lines 24 through 31. The third directive deleted two lines starting with line 52. Since it deleted the last line of the file, the editor displayed the message "BOTTOM OF FILE REACHED".

Before we discuss any more directives, we need to expand the definitions of <line> and <target>.

As editing operations are performed, the editor keeps track of the "current line" which usually is the line most recently affected by a successful edit directive. Upon entering the editor, the "current line" is the first line of the file. If, for example, we have just inserted three lines between lines 12.00 and 13.00, the current line will be 12.30. One should note that after a line or a block of lines have been Deleted, the line immediately following the last one deleted is made the current line (if the last line of the file was deleted, the new last line of the file will be the current line).

In our discussions above, we have implied that one has to explicitly indicate a <line> for each directive by specifying the line number of the line of interest. However, if <line> is not specified in a directive, the "current line" is used. For example, if one enters the directive

```
# D 2
#
```

the editor will delete two lines starting with the current line. In our example, since we were at line 6.00, the "D2" opertion deleted lines 6.00 and 7.00. As you will learn to appreciate, the "current line" default for <line> is extremely handy.

After performing all of the above operations, our file now looks like this:

```
 1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
 2.00 =TSC TEXT EDITING SYSTEM. A NUMBER OF
 3.00 =EXAMPLES WILL BE SHOWN TO ALLOW EASY AND
 4.00 =QUICK LEARNING OF ITS FEATURES.
 5.00 =FOLLOWING ARE SOME NONSENSE LINES:
 8.00 =TESTING 1234
 8.10 =THIS IS AN INSERTED LINE.
 8.20 =SO IS THIS.
 9.00 =THIS EDITOR IS FUN TO USE!
10.00 =BBBBBBBB
11.00 =
11.10 =ANOTHER INSERTED LINE.
12.00 =THIS IS THE END OF THIS FILE,
13.00 =AT LEAST FOR NOW.
```

We have seen that <line> may be specified by a line number or by default to the current line. There are also several other ways to specify <line>, or in other words, to move the pointer to a desired line prior to the execution of an edit directive. One may also specify <line> with a "+n" or "-n" (where n is an integer) meaning the next nth line in the file or the nth previous line in the file, respectively. Two other useful <line> designators are "^" ("↑" on some terminals) and "ϕ" (! on some terminals). The up arrow "↑" is used to designate the top or

first line in the file. The down arrow "↓" is used to move to the last line or bottom of file. These various <line> specifiers are shown in the example below with the PRINT directive.

```
*↑P
    1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
*+3 P
    4.00 =QUICK LEARNING OF ITS FEATURES.
*↓ P
   13.00 =AT LEAST FOR NOW.
*-2P
   11.10 =ANOTHER INSERTED LINE.
*
```

There may be times while editing a file when we know part of the contents of a line of interest but don't know its line number nor its displacement from the current line. In such a case we can use the "content-oriented" feature of the editor to find it. The syntax to specify <line> in this way is

/<string>/

where "/" is a character to delimit (enclose) the <string> which is a sequence of characters known to be in the line. When <line> is specified as "/<string>/", the editor will search for the current line through the file to find the next line containing the specified <string>. Some examples will help to clarify this: (1) /PRINT/ denotes the next line containing the character string "PRINT", and (2) /GO TO 35/ refers to the next line containing "GO TO 35". If the <string> is found in any subsequent line of the file, that line will be made the current line and the requested edit operation will be performed on it. If the <string> does not occur anywhere subsequent in the file, the editor will issue the message "NO SUCH LINE" and will not change the current line pointer. Note that the delimiter does not need to be a slash; it may be some other character such as a quote (') or a comma. For example, 'A/B' refers to the next line containing "A/B".

It is also possible to prefix the string designator with "-" (minus sign) to indicate a previous line containing that string. A few examples with our TEST FILE will show the use of "/<string>/" as a <line> designator.

```
*-/QUICK/P
    4.00 =QUICK LEARNING OF ITS FEATURES.
*;123; P
    8.00 =TESTING 1234
*+'END'P
   12.00 =THIS IS THE END OF THIS FILE.
*
```

To summarize, we have seen that <line> may be specified a number of ways, namely: (1) by default to the current line, (2) by typing a line number, (3) by "+n" denoting the nth subsequent line, (4) by "-n" referring to the nth previous line, (5) by /<string>/ denoting the next line in the file containing the indicated string of characters, (6) "-/<string>/" to denote the nearest previous line containing the specified character string, (7) "^" ("↑" on some terminals) to denote the first line of the file, and (8) "↓" ("!" on some terminals) to denote the last line of the file.

Now lets turn our attention to expanding the definition of <target>. As you may recall, a <target> is used in some directives to indicate the number of lines to be affected by the edit operation. We have already seen that a <target> may be specified by (1) an integer "n" indicating the number of lines to be affected, as P3, meaning print 3 lines, and (2) a line number preceded by a pound sign (#) indicating the line number of the last line to be affected, as P #6, meaning print all lines to and including line #6. The <target> is simply a designator telling how many lines the edit directive should operate on. In addition to the two mentioned forms of <target>, we also have, (3) if no <target> is specified in a command whose syntax includes one, a <target> of 1 is assumed, thereby affecting only one line. As with <line>, one may specify <target> by (4) "/<string>/" which indicates the next line in the file containing the specified character string, (5) " " to denote the top line in the file, and (6) " " to denote the bottom line in the file. A minus sign may be used to indicate that processing is to proceed backward through the file in the following two cases: (7) "-n" and (8) "-/<string>/".

With an understanding of <line> and <target> we can now discuss some more directives. The Print directive is used to display a line or a group of lines. Its syntax is

<line>P <target>

where "<line>" and "<target>" may be specified in any of the ways discussed above. To print just one line one needs to specify only <line> followed by a carriage return; therefore, the following two directives perform the same thing;:

<line>P

and

<line>

Going back to our test file, we can illustrate the various forms of <target> as used with the Print directive.

```
*2P
    2.00 =TSC TEXT EDITING SYSTEM. A NUMBER OF
*-1
    1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
```

```
#P /EASY/
     1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
     2.00 =TSC TEXT EDITING SYSTEM. A NUMBER OF
     3.00 =EXAMPLES WILL BE SHOWN TO ALLOW EASY AND
#! P -3
    13.00 =AT LEAST FOR NOW.
    12.00 =THIS IS THE END OF THIS FILE,
    11.10 =ANOTHER INSERTED LINE.
#-/BBB/ P -/123/
    10.00 =BBBBBBBB
     9.00 =THIS EDITOR IS FUN TO USE!
     8.20 =SO IS THIS.
     8.10 =THIS IS AN INSERTED LINE.
     8.00 =TESTING 1234
#12P!
    12.00 =THIS IS THE END OF THIS FILE,
    13.00 =AT LEAST FOR NOW.
#
```

The first directive displayed line 2.00 and made that line the current line. The second directive requested that the line immediately preceding the current line be displayed. The third directive displayed the block of lines from the current line down through the line containing the character string "EASY". The fourth directive printed 3 lines starting at the bottom of the file and ending at line 11.10, which becomes the current line. The fifth directive requested the previous line containing the character string "BBB" be found, and then starting with that line, display all lines going backwards through the file until a line containing the character string "123" has been displayed. This shows the extreme usefulness and power of the content-oriented characteristic of the editor. The last directive requested that all lines from line 12.00 to the end or bottom of file be displayed.

The next directive to discuss is Next which is used primarily to move the line pointer. Although it may be used otherwise, usually it is used only with the default <line>. Its syntax is

N<target>

This directive finds the line indicated by <target>, displays it, and makes it the current line. A few examples will illustrate its use.

```
*^P
    1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
*N
    2.00 =TSC TEXT EDITING SYSTEM. A NUMBER OF
*N 6
    8.20 =SO IS THIS.
*N -2
    8.00 =TESTING 1234
*
```

The following directive performs single-line replacements or inserts. Its syntax is

<line>=<text>

where "<line>" specifies the number of the line to be replaced or inserted and may, of course, default to the current line. "<text>" is the text to comprise the line. To illustrate this directive, let's continue our example series.

```
*=REPLACE CURRENT LINE HERE
*5.25=THIS LINE CREATED WITH 'EQUALS'.
*
```

The first directive changed the contents of line 8.00, the current line. The second example inserted a line with the line number 5.25.

The next directive to be discussed is the Change directive. It is used to change occurrences of one character string into another. Its syntax is

<line>C/< />1/<string>2/ <target> <occurrence>

where "/" is a delimiter character to separate the two character strings; "<string>1" is the character string to be replaced; "<string>2" is the string of character to replace them; "<target>" specifies the range of the changes; and "<occurrences>" specifies which occcurrence(s) of <string>1 should be replaced in the line(s). If <occurrence> is 1 or is not specified, then only the first occurrence of <string>1 in any line of the block will be changed - the second or subsequent occurrence of the string in such a line will not be affected. If 2 is specified for <occurrence>, then only the second occurrence of <string>1 in any line of the block will be changed. To change all occurrences of the indicated string in the block, use an asterisk (*) for <occurrence>. Let's illustrate the Change directive by continuing our example.

```
#4C /QUICK/FAST/
    4.00 =FAST LEARNING OF ITS FEATURES.
#8.1  C /THIS IS //
    8. 10=AN INSERTED LINE.
#-5C ;A;$; ;SOME; *
    3.00 =EX$MPLES WILL BE SHOWN TO $LLOW E$SY $ND
    4.00 =F$ST LE$RNING OF ITS FE$TURES.
    5.00 =FOLLOWING $RE SOME NONSENSE LINES:
#12 C /E/?/ -2 3
   12.00 =THIS IS THE END OF THIS FIL?,
   11.10 =ANOTHER INSERT?D LINE.
#
```

The first example replaced the string "QUICK" with the string "FAST" in line 4.00. The second example deleted the string "THIS IS" and a blank from line 8.1. The third example starts at the fifth previous line (line 3.00) and changes every occurrence of "A" to "$" down through all lines until the line containing the character string "SOME" (line 5.00) is reached. The last example changes the third occurrence of "E" to "?" in line 12.00 and then in line 11.10.

The last directive to be discussed is used to exit from the editor. This can be done several ways: STOP, S, or LOG. This will return you to your system monitor.

Now lets go back to our test file and illustrate some of the features and directives we have discussed.

```
#^P!
    1.00 =THIS IS AN EXAMPLE OF THE FANTASTICALLY USEFUL
    2.00 =TSC TEXT EDITING SYSTEM. A NUMBER OF
    3.00 =EX$MPLES WILL BE SHOWN TO $LLOW E$SY $ND
    4.00 =F$ST LE$RNING OF ITS FE$TURES.
    5.00 =FOLLOWING $RE SOME NONSENSE LINES:
    5.25 =THIS LINE CREATED WITH 'EQUALS'.
    8.00 =REPLACE CURRENT LINE HERE
    8.10 =AN INSERTED LINE.
    8.20 =SO IS THIS.
```

```
    9.00 =THIS EDITOR IS FUN TO USE!
   10.00 =BBBBBBBB
   11.00 =
   11.10 =ANOTHER INSERT?D LINE.
   12.00 =THIS IS THE END OF THIS FIL?,
   13.00 =AT LEAST FOR NOW.
*2C/C /C 6800 /
    2.00 =TSC 6800 TEXT EDITING SYSTEM. A NUMBER OF
*/BBB/
   10.00 =BBBBBBBB
*-;THIS IS; C 'E'XX' !
    1.00 =THIS IS AN XXXAMPLE OF THE FANTASTICALLY USEFUL
    2.00 =TSC 6800 TXXXT EDITING SYSTEM. A NUMBER OF
    3.00 =XXX$MPLES WILL BE SHOWN TO $LLOW E$SY $ND
    4.00 =F$ST LXX$RNING OF ITS FE$TURES.
    5.00 =FOLLOWING $RXX SOME NONSENSE LINES:
    5.25 =THIS LINXX CREATED WITH "EQUALS".
    8.00 =RXXPLACE CURRENT LINE HERE
    8.10 =AN INSXXRTED LINE.
    9.00 =THIS XXDITOR IS FUN TO USE!
   11.10 =ANOTHXXR INSERT?D LINE.
   12.00 =THIS IS THXX END OF THIS FIL?,
   13.00 =AT LXXAST FOR NOW.
*N -4
   10.00 =BBBBBBBB
*-1 I
    9.10 =TEST-TEST-TEST
    9.20 =1234567890
    9.30 =*D!
BOTTOM OF FILE REACHED
*^D!
BOTTOM OF FILE REACHED
*^P!
*S
```

The previous tutorial has been only a brief introduction to the TSC Text Editing System. The remainder of this manual contains a detailed description of each directive with examples, in the next section, followed by "How to Use Tape" and "Using the Disk Version". It is important to read and study the entire manual in order to fully understand and utilize all of the power and features of this editor. The source listing is the last section.

EDITOR DIRECTIVES

The following manual more explicitly describes all the editor commands, use of special features and adapting to your system. You would be well advised to first read the Mini-Tutorial preceding this section. It will give you an overall feel for what the editor can do, thus making the detailed descriptions more understandable. Before getting into the complete descriptions of the editor directives, a few general points will be covered.

USING STRINGS:

Several of the editor directives use character strings as arguments. These arguments are either matched against strings in the text, or replace a string in the text. A string argument begins after a delimiter character and continues as a sequence of any legal characters until the delimiter is again encountered. The delimiters are not considered part of the string to be used in the matching or replacement operations. Although the delimiters in the following descriptions are frequently represented as slashes, "/", any non-blank, non-alphanumeric character may be used as the delimiter such as: * / ( ) $ = , . [ ] : ' etc. Note that the following characters may not be used to enclose strings unless they are preceded by either a plus (+) or minus (-) sign: "↑" (denotes first line of file), "!" (denotes last line of file), "-" (denotes target is above current line), and the character denoted by LINO (normally a pound sign) which is used to flag line numbers. The delimiter character is redefined in each new request by its appearance before a string. If two strings exist in one directive (as in the CHANGE directive), the same delimiter character must be used for each string.

All of the editor directives use the <line> information preceding the directive to position the pointer prior to any directive action. The <line> parameter may of course be null, meaning leave the pointer at its current position. All of the following are valid <line> designators:

1.	Any number	references a specific line number
2.	+n	denotes the nth subsequent line
3.	-n	denotes the nth previous line
4.	/<string>/	refers to the next line in the file containing the indicated string of characters
5.	-/<string>/	refers to a previous line containing the indicated string

6. ↑	denotes the first line of the file
7. !	denotes the last line of the file
8. null	stay at the current line

Many of the editor directives require <target> information. This tells the editor to operate on the "current" line and all other lines in the file up to the line referenced by the <target>. In cases where a <target> is required, leaving it null will make the <target> default to one, meaning only the current line will be affected by the directive. All of the following are valid <target> designators:

1. an integer n	indicates that n lines should be affected by the edit operation
2. #n	denotes the line number of the last line to be affected
3. /<string>/	denotes the next line in the file containing the specified character string
4. -/<string>/	references the previous line containing the indicated string
5. ↑	denotes all lines up to the top of the file
6. !	denotes all lines to the bottom or last line of the file
7. +or- n	indicates that n lines should be affected and in which direction from the current line
8. (null)	defaults to 1 and only the current line is affected

As we have seen, the form <target> is used to specify a range of lines to which the directive will apply. The directive will be applied to each line, starting with the line specified by <line> and continuing until the target is reached.

If a string <target> is specified, the directive will apply to successive lines of text until a line containing the string is reached. Processing proceeds downward in the file unless the target is preceded by a "-" (minus sign), indicating that processing is to occur upward (toward the first line) in the file. Targets may also be preceded by a plus sign (indicating downard movement). If a line number target is specified,

processing begins at <line> and proceeds toward the target line number. Some examples of <target>s are:

```
2
+10
-3
/STRING/
+/STRING TARGET/
-/BACKWARD DISPLACEMENT TO A STRING/
+*ANY DELIMITER WILL WORK FOR STRING*
++EVEN PLUS SIGNS CAN WORK+
#23.00
```

SPECIFYING A COLUMN NUMBER:

Any "/<string>/" descriptor may be postfixed with a column number immediately after the second delimiter to indicate that the preceding string must begin in the column specified to be found. If the column specified is not in the range of the ZONE in effect, the request will be ignored. Some examples are:

```
/IDENT/11
/PROGRAM/77
*LABEL*2
$COMMENT$30
```

The COMMAND REPEAT CHARACTER:

A special "Command Repeat Character" has been set up in the editor to allow you to exactly repeat the last command in the input buffer. If a command causes an error or changes the contents of the input buffer, an ILLEGAL COMMAND will be reported upon subsequent use of the Repeat character until another repeatable command is entered. The repeat character is originally set to a CTRL R or 12 hex. Some examples of commands which may be useful to repeat are:

PRINT 15	To print a screen of lines at a time
NEXT	Allows you to single step thru the file with one key
↑CO!!	To quickly fill the workspace
FIND/SOME STRING/	If the first string found is not the one desired

USING THE EOL CHARACTER:

The editor supports an "EOL" or "End Of Line" character to allow multiple commands in a single line. INSERT and OVERLAY are exceptions in that they cannot be followed by other commands. The EOL character may be interactively changed using the SET command. An example of EOL use (with EOL set to "$") is:

```
↑D2$P10$T
```

This sequence will delete the first 2 lines of the file, then print the next 10 lines, and finally return the pointer to the top of the file.

USING TABS:

The user may interactively specify a tab character and up to 20 tab stops. The tab character may then be inserted into a line where it will be expanded when the end of the line is received. If tab stops or the tab character have not been previously set, but some character has been used throughout the file as a tab, it can still be expanded by setting it to be the tab character, setting up your tab stops and then using the EXPAND command on the file. Note that if the tab character has been set, subsequent uses of the INSERT or REPLACE commands will cause automatic tab expansion. However if a tab character is added to the file by the use of a CHANGE, APPEND or OVERLAY command, that character will remain intact in the file until the EXPAND command is invoked on the line containing that tab character.

EDITOR DIRECTIVES

There are five groups of editor directives: environment directives, system directives, "current line" movers, edit directives, and tape directives. A complete description of all directives in each group is given below. In the following descriptions, quantities enclosed in square brackets ([...]) are optional and may be omitted. A backslash (\) is used to separate options.

ENVIRONMENT DIRECTIVES:

H[EADER] [<count>]

MEANING:

A header line of <count> columns will be displayed. The heading is of the form "123456789012..." to indicate the column number. Columns for which tab stops are set will contain a minus character instead of the normal digit. If a column count is given, it becomes the default such that if just 'H' is subsequently typed, that number of columns will be printed.

EXAMPLES:

HEADER 72	Display column number headings for 72 columns
H 30	Display column numbers for 30 columns

NU[MBERS] [OFF\ON]

MEANING:

The line number flag is turned off or on. If the flag is off, then line numbers will never be printed. If neither "OFF" nor "ON" is specified, then the flag will be toggled from its current state.

EXAMPLES:

NUMBERS OFF	Turn line number printing off
NU ON	Turn it back on
NU	Toggle from on to off or from off to on

REN[UMBER]

MEANING:
The "renumber" directive will renumber all of the lines in the current edit file. Lines in the renumbered file will start with line number "1.00" and will have an increment of one. The line which was current before the command will still be the current line after the command (although its number will probably have been changed).

EXAMPLES:

RENUMBER	Renumber the lines in the current working file
REN	

SET <name> = '<char>'

MEANING:
SET is used to define certain special characters or symbols. The <name>s which may be set are:

TAB - the tab character
FILL - the tab fill character
EOL - the end of line character which may be used to separate several commands on a single line
LINO - the line number flag character which is used to indicate that a target is a specific line number

The default values are: TAB and EOL are "null"
FILL is "space"
LINO is "#"

EXAMPLES:

SET TAB='/'	Set the tab character to a slash
SET TAB=''	Disable tabbing by setting the tab character to a null
SET FILL=' '	Set tab fill character to a blank
SET EOL='$'	Set the EOL character to $
SET LINO='@'	Set the line number flag to at sign

TAB [<columns>]

MEANING:
Used to set the tab stops. All previous tab stops are cleared. If no columns are specified, then the only action is to clear all tab settings. Any tab characters occurring beyond the last tab stop are left in the text. The maximum number of tab stops allowed is 20. Tab stops MUST be entered in ascending order.

EXAMPLES:

TAB 11, 18, 30	Set tab stops at columns 11, 18, and 30
TAB 7 72	Set tab stops for a FORTRAN program
TAB	Clear all tab stops

V[ERIFY] [ON\OFF]

MEANING:
The verify flag is turned on or off. The verify flag is used by the directives CHANGE and NEXT (and several others) to display their results. If neither "ON" nor "OFF" is specified, then the flag will be toggled from its current state.

EXAMPLES:

VERIFY OFF	Turn verification off
V ON	Turn it back on

X

MEANING:
"X" is the cursor control command. Any time this command is entered, the editor will issue the 6 special character string previously set up. See "Adapting to Your System" for details.

EXAMPLES:

X	Output cursor control string

Z[ONE] [C1,C2]

MEANING:
ZONE is used to restrict all sub-string searches (FIND, CHANGE, <target>s, etc.) to columns "C1" to "C2" inclusive. Any substrings beginning outside those columns will not be detected. If C1 and C2 are not specified, then the zones will be reset to their defaults (columns 1 and 136).

EXAMPLES:

ZONE 11, 29	Restrict searches to columns 11 through 29
ZONE	Search columns 1 thru 136

SYSTEM DIRECTIVES:

LOG

MEANING:
Exit the editor.

EXAMPLES:

LOG

S[TOP]

MEANING:
Same as LOG.

EXAMPLES:

STOP

S

"CURRENT LINE" MOVERS:

B[OTTOM]

MEANING:
Move to the last line in the file and make it the current line.

EXAMPLES:

BOTTOM	Make the last line of the file the current line

B

F[IND] <target> [<occurrence>]

MEANING:
Move the current line pointer to the line specified by <target> and make it the current line. If the VERIFY flag is on (see VERIFY), the line will be printed. If <occurrence> is specified (an unsigned integer or an asterisk), the directive will be repeated <occurrence> times. If <occurrence> is an integer, it must not start in the first column following the second delimiter of a string <target>, as it would then appear to be a column specifier for that string. If no column is to be specified, insert a space after the second delimiter and before the <occurrence> as in the second example given below. An asterisk means all occurrences of the <target> will be found until the bottom or top of the file is reached. If the target is not found, the current line pointer will not be moved.

EXAMPLES:

FIND /STRING/	Find the next line containing the string "STRING"
F/THREE LINES/ 3	Find the next three lines containing the string "THREE LINES"
F/ALL 'TIL BOTTOM/*	Find all following occurrences of the indicated string
F-/PROGRAM/7 *	Find all previous lines which have the word "PROGRAM" starting in column seven

N[EXT] [<target> [<occurrence>]]

MEANING:
The line specified by the target is made the current line. If the VERIFY flag is on, the line will be printed. If <occurrence> is specified, it must be an unsigned integer. It indicates which next occurrence of a line containing the target is to be made the current line. If the target is not reached, the current line pointer will be positioned at the bottom of the file (or top of the file for a negative <target>). If no target is specified, the next line will be made the current line.

EXAMPLES:

NEXT 5	Make the fifth following line the current line
N	Make the next line the current line
N-10	Make the 10th previous line current
N/STRING TARGET/	Make the next line containing "STRING TARGET" to be the current line
N/3RD OCCURRENCE/ 3	Make the third line containing the indicated string the current line

T[OP]

MEANING:
The first line of the file becomes the current line.

EXAMPLES:

TOP	Make the first line of the file the current line

EDIT DIRECTIVES:

A[PPEND] /<string>/ [<target>]

MEANING:
Append the specified <string> just beyond the last character of the current line (and to successive lines until the target is reached). If the string is postfixed with a column number, then append the string beginning at the specified column (rather than at the end of the line). Any characters previously in the line following the specified column will be lost.

EXAMPLES:

APPEND /./	Append a period to the end of the current line
A *HELLO* 2	Append the word "HELLO" to the end of the current line and to the end of the next line.
A/SEQUENCE/73 *END*7	Append the word "SEQUENCE" starting in column 73 of the current line and successive lines until a line containing the characters "END" begining in column seven is found

C[HANGE] /<string1>/<string2>/ [<target> [<occurrence>]]

MEANING:
Replace the string specified by <string1> with the string specified by <string2>. If no <target> is specified, only the current line is affected. The slashes represent any non-blank delimiter character. <occurrence> is used to specify which occurrence of <string1> is to be replaced in each line. It is either an unsigned integer or an asterisk ('*') signifying that all occurrences of the substring <string1> are to be replaced with <string2>. By default, only the first occurrence will be

changed. Note that if <occurrence> is specified, and if changes are to occur to the current line only, then the target should be a 1 (one).

EXAMPLES:

CHANGE /THIS/THAT/	Replace the first occurrence of "THIS" in the current line with "THAT"
C/A/B/ 1*	Change all occurrences of "A" in the current line to "B"
C /FIRST/LAST/10	Change the first occurrence of "FIRST" to "LAST" in the current line and also in the nine following lines
C /NEW/OLD/ /A TARGET/	Change the first occurrence of "NEW" to "OLD" in each line down through the line containing the string "A TARGET"
C ,A,, -10 *	Remove all "A"s in the current line and in the nine preceding lines
C*HELLO*	Delete the character string "HELLO" from the current line

CO[PY] [<destination-target> [<range-target>]]

MEANING:

The current line and successive lines until the <range-target> is reached are copied so that they follow the line specified by <destination-target>. The default <destination-target> is 1, thereby causing a copy of the current line to be placed after the next line. The default <range-target> is 1, thereby copying only one line. After the directive is executed, the current line pointer will be set to the new position of the last line copied. Some lines may be renumbered after a copy with no renumbering message issued.

EXAMPLES:

CO #18	Put a copy of the current line after line 18
COPY #3 4	Copy four lines beginning with the current line and place them after line 3

CO /CHECK/ +/RANGE/	After the next line which has the string "CHECK", place a copy of each line starting with the current line through the line containing "RANGE"

D[ELETE] [<target>]

MEANING:
The current line (and successive lines until the target is reached) is deleted. After the directive is executed, the current line will be the line following the last line deleted.

EXAMPLES:

DELETE 5	Delete five lines (the current line and the next four lines)
D	Delete the current line
D /STRING/	Delete lines from the current line through the next line that contains the string "STRING"

EXP[AND] [<target>]

MEANING:
The current tab character is expanded within all lines, beginning with the current line, continuing down to and including the line specified by <target>. Since tabs are normally expanded as lines are inserted into the file, this directive is primarily of use when one has forgotten to define a tab character.

EXAMPLES:

EXPAND 100	Expand 100 lines starting with the current line
EXP	Expand the current line

I[NSERT]

MEANING:
The editor will enter the buffered input mode, prompting with line numbers (unless line numbers have been disabled, see the NUMBERS directive) and insert the lines below the current line. Buffered input continues until a line beginning with the breakpoint character (a pound sign, "#") in column one is received. The characters

following the breakpoint character are treated as an editor directive. The editor will try to choose an insertion increment sufficient to insert at least 10 lines, or if that is not possible, the smallest increment possible. The current line pointer is positioned at the last line inserted. It should be noted that the editor may renumber text lines following the inserted text if the inserted line numbers overlap line numbers previously in the file.

EXAMPLES:

INSERT	Accept line input after the current line
I	Same

I[NSERT] <text>

MEANING:
The text (sequence of characters) which immediately follows the separator (or blank) after the directive name will be inserted as a separate line below the current line of the file. The line inserted becomes the current line. It should be noted that the editor may renumber text lines following the inserted text if the inserted line number overlaps line numbers previously in the file.

EXAMPLES:

I THIS BELOW THE CURRENT LINE OF THE FILE

INSERT EVERYTHING AFTER THE FIRST BLANK

MO[VE] [<destination-target> [<range-target>]]

MEANING:
The current line and successive lines until the <range-target> is reached are moved so that they follow the line specified by <destination-target>. The default <destination-target> is 1, thereby moving the current line after the next line in the file. The default <range-target> is 1, thereby moving only one line. After the directive is executed, the current line pointer will be set to the new position of the last line moved. Some lines may be renumbered after a move with no renumbering message issued.

EXAMPLES:

MOVE 3	Move the current line down three lines

MO #1 /TARGET STRING/	Move the current line and all lines down thru the line containing "TARGET STRING" after line 1
MO -/PROGRAM/ 5	Move five lines (including the current line) up within the file so that they follow a line containing the character string "PROGRAM"
MO #10 -5	Move the current line and the four previous lines below line number 10

O[VERLAY][<delimiter>]

MEANING:

The current line is printed, then a line of input is accepted from the terminal (the overlay line). The overlay line will be positioned directly beneath the line printed out. Each character of the overlay that is different from the <delimiter> character (which defaults to a blank) will replace the corresponding character in the current line. The overlaid line will be printed if verify is "ON".

EXAMPLES:

```
OVERLAY
  25.00=THIP IS THE CORRENT LUNE.
OVERLAY    S          U       I
  25.00=THIS IS THE CURRENT LINE.
```

O[VERLAY]<d><text>

MEANING:

This directive is similar to the previous form of the OVERLAY directive with these differences: (1) The current line is not printed. (2) The remainder of the directive line (after the delimiter character) is taken as the overlay text.

EXAMPLES:

```
OVERLAY---AT------------------- NUMBER.
  25.00=THAT IS THE CURRENT LINE NUMBER.
```

P[RINT] [<target>]

MEANING:
Beginning with the current line, lines are printed until the line specified by target is reached. By default, only the current line will be printed.

EXAMPLES:

P	Print the current line
PRINT 5	Print 5 lines starting with the current line
P -10	Print the current line and the nine previous lines
PRINT *STRING*	Print all lines down thru the next line containing "STRING"
P -/STRING/	Print all lines up through the next previous line containing "STRING"

R[EPLACE] [<target>]

MEANING:
A DELETE from the current line through the <target> line is performed. The editor then enters the buffered input mode, putting the new lines into the area vacated. It is not necessary to enter the same number of lines as were deleted. The line numbers of the lines inserted will probably not be the same as those deleted. The current line pointer will be positioned at the last line inserted. By default, only the current line will be deleted.

EXAMPLES:

R	Replace the current line
REPLACE 10	Replace 10 lines starting with the current line
R /TARGET STRING/	Replace all lines from the current line through the line containing "TARGET STRING"

=<text>

MEANING:
The "=" directive replaces the current line with the text supplied. The replacement text begins with the first character following the equals sign. The current line pointer is not moved.

EXAMPLES:

=THIS IS REPLACEMENT TEXT.

<null>

MEANING:
The null directive (i.e., just a carriage return) prints the current line.

TAPE DIRECTIVES:

GAP

MEANING:
Issue a string of 40 null characters to the tape unit.

EXAMPLES:

GAP	Puts leader or gap on tape

READ

MEANING:
The next file present on the tape will be loaded. All the lines read will be appended to the end of the current work file and the last line read will become the new current line.

EXAMPLES:

READ	Get the next file from tape

SAVE

MEANING:
Write the entire current file out to the tape unit. The tape is formatted as shown in the "USING TAPE" section of this manual. The file is terminated with an ASCII "control Z" character.

EXAMPLES:

SAVE	Puts the current file on tape

W[RITE] [<target>]

MEANING:
This directive is much like SAVE. The only difference being that SAVE puts the entire file on tape, while WRITE puts all lines from the current line through the target line onto tape. The same format as SAVE is used on the tape.

EXAMPLES:

WRITE	Write the current line to tape
WRITE #20	Write all lines from the current line thru line #20 out to the tape unit

USING TAPE:

The TSC TEXT EDITING SYSTEM contains four tape directives. These can be used with most types of tape devices including paper tape and Kansas City Standard cassette systems. When using SAVE or WRITE, the text is sent out to the tape, character at a time, in the following form:

TEXT. . . <C. R. >. . . TEXT. . . <C. R. >. . . TEXT. . . <C. R. ><CTRL Z>

The "CTRL Z" is the end of file marker. Note that there are no line numbers, line feeds, or null characters put on the tape, so the file is not suitable for displaying on a terminal in this form. When a tape is read back into the editor using the READ command, line numbers are automatically put back in. If there is more data on the tape than will fit in the workspace, the tape will continue to play, ignoring the excess characters. When the CTRL Z is reached, an error message will be issued.

The TSC TEXT EDITING SYSTEM provides delay after tape turn on for cassettes and also issues control characters for each "tape on", "tape off", "record on", and "record off". To set these characters to those needed by your particular tape system, see ADAPTING TO YOUR SYSTEM.

USING THE DISK EDITOR

The TSC Text Editing System for the FLEX Operating System is a content oriented text editor which is powerful, simple to use, and easy to learn. It is a great tool for creating or editing various types of text files such as files for the Assembler, Text Processor, and various language compilers and interpreters. The FLEX version is a 'disk to disk' type editor, meaning any size file which will fit on a single disk may be edited, regardless of the amount of user's RAM available (at least 12K is required).

DESCRIPTION

The general syntax of the EDIT utility is:

EDIT,<file spec 1>[,<file spec 2>]

The default extension is TXT and the default drive is the working drive. If only <file spec 1> is given, and the name specified does not exist on the disk, a new file with that name will be created. Creating new files in this manner will cause the editor to respond:

```
NEW FILE:
   1.00=
```

If the EDIT command line only has <file spec 1> specified and it is a name which does exist on the disk, that file will be loaded into the edit buffer, and the editor will issue a '#' as a prompt, signifying that the editor is ready to accept user commands. When the editing process is completed, the original file will now have the same name as before editing except the extension will now be BAK, which stands for 'backup'. If a file existed with the same name and an extension of BAK, the editor will ask:

DELETE BACKUP FILE?

Answering this with a Y will delete the old backup file and create a new one. Any other response will return control back to FLEX. The newly edited file will have the same name as the original, including the extension. The final form of EDIT is similar to the above but allows assigning the new file a specific name, different from the original. The original would then keep its original name. It should be noted that when editing an existing file, a new file is always created, and the original remains intact, even though its name may be changed.

Several examples will help clarify the above syntax. Suppose you want to create a file called TEST.TXT (no such name currently exists on the disk). The following command line should be typed:

```
EDIT,TEST
```

The editor would respond with 'NEW FILE' and be ready to accept lines of text.

Suppose you have created a file named TEST.TXT and you now wish to edit it (make changes to it). Typing:

```
EDIT,TEST
```

would now load the file TEST.TXT into memory and the editor would be ready to accept commands. When the editing process is completed, and control is returned to FLEX, the original file 'TEST.TXT' will now have the name 'TEST.BAK', but its contents will be unchanged. The file containing all of the changes made while in the editor will have the name 'TEST.TXT'.

If it is still necessary to make more changes to the new file, the same calling procedure may be used. Now, since there is a file called 'TEST.BAK' already on the disk, the editor will ask if you want the backup file deleted. If deleted, the same procedure as above will again take place, and you will end up with the old file having the name 'TEST.BAK' and the new one 'TEST.TXT'.

The final form of the EDIT utility is used if you desire to edit a file, but give the new file a new name. If the following was the command line:

```
EDIT,TEST,TEST2
```

the file TEST.TXT would be loaded into the editor, and the new file would have the name TEST2.TXT. This form of the command line should also be used if it is necessary to edit a file from one drive, and put the new file on a different drive. As an example:

```
EDIT,0.TEST,1.TEST
```

would edit the file TEST.TXT on drive 0, and put the new file, TEST.TXT on drive 1. The file TEST.TXT must not already exist on drive 1.

Once in the editor, all of the edit commands apply to the FLEX version of the editor with a few exceptions. These differences are stated below.

EXITING THE EDITOR

The STOP command (or 'S') should be used. The 'LOG' command may also be used. After typing STOP, LOG, or S, the editor will automatically finish any old to new disk file transfers. If editing a large file, this process may take a while, so do not expect FLEX to immediately issue the prompt after exiting the editor.

SAVE, READ, WRITE, AND GAP

These commands are still supported but may now also be used to transfer files (or parts of files) to and from the disk, as well as tape. The GAP command may only be used with tape. Upon entering one of the above commands, the editor will respond with:

TAPE OR DISK (T-D)?

A response if 'D' will then cause the editor to prompt for the disk file name. Any other response (other than 'D') will cause the editor to load from tape. Consult the full editor manual for further details in using this set of commands.

THE 'NEW' COMMAND

There is one additional command in the FLEX version of the editor, the 'NEW' command. This command aids the editor in handling text files larger than what will fit in memory at any one time. When editing a file, the editor will only load memory with as much of the file as will fit. The NEW command tells the editor that you are done editing that portion of the file and wish to load more of the text into memory. The NEW command works as follows: upon typing the command 'NEW', the editor will write everything from the top of the current work buffer (the first line currently in memory) down to but not including the 'current line' out to the new file on the disk. At this time, if there is any unread part of the original file, as much of it which can be placed in the unused remaining space of the work buffer (memory) will be read in off of the disk. Control will then be transfered back to the editor and the old 'current line' will now be the first line available to be edited. NEW can be used anytime during the edit session, but keep in mind that once it has been used, all parts of the file which were above the current line pointer will become inaccesible for the remainder of the editing session, since they have already been written out to disk. The editor can only operate on text in memory (the work buffer), therefore, global editor commands such as CHANGE and FIND will only be global with respect to the text in the buffer, and not the entire file, unless of course, the entire file will fit in the buffer. The NEW command may also be used when creating a new

file. While typing lines into the editor, it is possible to fill the buffer and a message will be issued stating 'NOT ENOUGH ROOM'. If this happens, typing NEW will cause the file from the top, down to the line pointer, to be written out to the disk, thus freeing up that much of the buffer space. Since no old file exists, nothing new will be read in from the disk.

BUFFER SIZE

The amount of buffer space available is directly proportional to the amount of memory installed in the computer. The more memory installed, the larger the edit buffer will be. The editor automatically adapts to the memory size.

ADAPTING TO YOUR SYSTEM:

The TSC TEXT EDITING SYSTEM has been assembled to run on a Motorola MIKBUG based 6800 system, such a the SWTPC 6800. In order to use the editor on your system, you must supply an input character routine and an output character routine as described below. Other adaptions can be easily made as explained in this section to customize the editor to your needs or desires. Please read these instructions carefully before making any changes.

1. INPUT CHARACTER ROUTINE - This routine is called by the editor to input a character from your keyboard into the A register and return. The parity bit is stripped off. No other registers may be altered except for flags. Put the address of your input routine at locations 0207 and 0208 hex.

2. OUTPUT CHARACTER ROUTINE - This routine is called by the editor to output a character from the A register to your display device and then return. Except for flags, no other registers may be affected. Place the address of your output routine at location 020A and 020B hex.

3. RETURN TO MONITOR ADDRESS - This is the address to which the CPU will jump upon the execution of a STOP or LOG command. Generally it should be set to the re-entry address of your system monitor. Place the address of your monitor at locations 0B56 and 0B57 hex.

4. TAPE INPUT CHARACTER ROUTINE - If your tape system is connected to a different I/O port than the terminal or uses a different routine to handle tape operations, you can set the address of this routine at locations 020D and 020E hex. The routine should read one character from the tape, place it in the A register with the parity bit stripped and return. Except for flags, no other registers may be affected.

5. TAPE OUTPUT CHARACTER ROUTINE - As explained above, if necessary to use a separate output routine for tape operations, set the address at locations 0210 and 0211 hex. The routine puts the contents of the A register onto tape and returns with no other registers affected except for the flags.

6. FULL DUPLEX - If your terminal requires software echo of typed characters and your input routine does not provide this, change the JMP (7E) at location 0206 hex to a JSR (BD). This change assumes your output character routine does not destroy the A register.

7. BEGIN POINTER - This pointer is the address of the first byte available in the edit workspace. It is presently setup pointing to the first byte following the editor. If necessary, the BEGIN POINTER may be changed by altering locations 035D and 035E hex as desired.

8. MEMORY END - This pointer is the address of the last byte available in the edit workspace. It is set to the FLEX Memend value.

9. SYSTEM CHARACTERS

   A) PROMPT CHARACTER - The prompt character is stored at location 0528 hex. It is presently a '#' or 23 hex.
   B) DELETE CHARACTER - The delete character is the FLEX Delete character.
   C) BACKSPACE CHARACTER - The backspace character is the FLEX Backspace character.
   D) BELL CHARACTER - When the input buffer is overflowed (more than 136 characters typed), the editor outputs a "bell" character. This is stored at location 065D hex and is presently set to a 'CTRL G' or 07 hex.
   E) REPEAT CHARACTER - The command repeat character is stored at 0530 hex. It is presently set to a 'CTRL R' or 12 hex.

10. TAPE TURN ON DELAY - The editor is assembled to delay approximately 2 seconds after tape turn on and before outputting data. This may be set as needed at location 00B6 hex. It is currently set to 06. Setting it to 0 is zero delay with larger values causing longer delays.

11. TAPE CONTROL CHARACTERS - The editor outputs special control characters to the tape system to control tape operations. These characters are sent for each operation and are presently set to nulls (for manual tape operation). These control characters may be set to suit your particular system and are located as follows:

TAPE ON (PLAY)	-	00B7
TAPE OFF (PLAY)	-	00B8
TAPE ON (RECORD)	-	00B9
TAPE OFF (RECORD)	-	00BA

12. CURSOR CONTROL CHARACTERS - The editor outputs a string of six control characters upon execution of the 'X' command. These can be set to special cursor control or other control characters. They are presently set to nulls (00). They may be changed as desired from 0B2B to 0B30 hex. Leave the 04 at the end of the string intact!

13. PROGRAM STACK - The editor uses an internal program stack for subroutine calls and temporary data storage. It is located at 01FF hex. If for some reason you wish to use a stack at another location, it can be accomplished by placing the desired address at 035A and 035B and again at 04FB and 04FC hex in normal 6800 form.

SYSTEM CHARACTERISTICS:

1. The maximum line number is 9999.99. If more than 9,999 lines are entered, the line number counter will turn over (go back to 0). the editor, therefore, should not be used with files of 10,000 lines or longer. (This is not really a limitation since 10,000 null lines (line number followed by a carriage return) uses up 40K of memory!)

2. When specifying a line number which is less than one, it is imperative that a leading zero be placed before the decimal point. This is so that the line number will be classified as a number rather than a delimiter.

3. The input buffer will hold 136 characters. If more than 136 characters are typed, they will be ignored and a "bell" character output to the terminal. To terminate the line, it is necessary to type the backspace character and then a carriage return.

4. Setting the "tab" character and the "fill" character the same will delete the TAB feature. There is no logical reason to do this.

5. There are several methods to insert a line above the top of a file containing a line 0.01 which may not be immediately obvious. One is to renumber the file which will change line 0.01 to line 1.00 allowing you to insert at 0.10. Another method is to issue an "INSERT -<x>", where <x> is the number of lines which will place you one line above the top of the file. If for example, the current line is line 0.01, the command, "INSERT -1", will allow you to insert above that line.

```
                        *
                        * COPYRIGHT 1978 (C) BY
                        *
                        * TECHNICAL SYSTEMS CONSULTANTS
                        *   BOX 2574
                        *   W. LAFAYETTE, INDIANA 47906
                        *   (317) 423 5465
                        *

                        * EXTERNAL EQUATES

01FF                    STACK   EQU     $01FF
E0D0                    MIKBUG  EQU     $E0D0
0023                    PROMPT  EQU     $23
000D                    CRGRET  EQU     $D
0007                    BELL    EQU     $7


                        * DISK TEMP STORAGE

0020                            ORG     $20

0020                    DSKFLG  RMB     1
0021                    DRWFLG  RMB     1
0022                    TTERM   RMB     1
0023                    ACCT    RMB     1
0024                    LASTIN  RMB     1
0025                    INDEX9  RMB     2
0027                    SAVEIT  RMB     2
0029                    BUFEND  RMB     2
002B                    RBFEND  RMB     2
002D                    XXX1    RMB     2
002F                    XXX2    RMB     2

0040                            ORG     $40

                        * TEMPORARY STORAGE

0040                    TEMP    RMB     2
0042                    XSAVE   RMB     2
0044                    BUFPNT  RMB     2
0046                    BUFSAV  RMB     2
0048                    CURPOS  RMB     2
004A                    NEWPOS  RMB     2
004C                    SRCHPT  RMB     2
004E                    STRNGB  RMB     2
0050                    STRNGE  RMB     2
0052                    STRGB1  RMB     2
0054                    STRGE1  RMB     2
0056                    STRPNT  RMB     2
0058                    SPCPT1  RMB     2
005A                    SPCPT2  RMB     2
005C                    LASTNO  RMB     2
```

```
005E            ZONE1   RMB     2
0060            ZONE2   RMB     2
0062            ZONBUF  RMB     2
0064            CHGPNT  RMB     2
0066            CHGEND  RMB     2
0068            OCRTMP  RMB     2
006A            NUMFLG  RMB     1
006B            VERFLG  RMB     1
006C            REPEAT  RMB     1
006D            MSLFLG  RMB     1
006E            PSTZFL  RMB     1
006F            OCRCNT  RMB     2
0071            FNDFLG  RMB     1
0072            STRCN1  RMB     1
0073            OVRBEG  RMB     1
0074            OVREND  RMB     1
0075            NOCURL  RMB     1
0076            LINFLG  RMB     1
0077            NXTFLG  RMB     1
0078            ALLFLG  RMB     1
0079            OCRFLG  RMB     1
007A            CHGONF  RMB     1
007B            APPCOL  RMB     1
007C            STRCNT  RMB     1
007D            INCAMT  RMB     1
007E            BMPFLG  RMB     1
007F            EQUFLG  RMB     1
0080            INLMFL  RMB     1
0081            MOVFLG  RMB     1
0082            REPFLG  RMB     1
0083            TMPCHR  RMB     1
0084            CHKFLG  RMB     1
0085            SNGLIN  RMB     1
0086            CHGFLG  RMB     1
0087            STRCN2  RMB     1
0088            FNONFL  RMB     1
0089            LSTFLG  RMB     1
008A            DECCNT  RMB     1
008B            PRNFLG  RMB     1
008C            CPYDRC  RMB     1
008D            DRCTN   RMB     1
008E            CHRCNT  RMB     2
0090            INZFLG  RMB     1
0091            NUMBER  RMB     3
0094            TRGLIN  RMB     2
0096            DELIM   RMB     1
0097            HEDCNT  RMB     1
0098            FILBEG  RMB     2
009A            FILEND  RMB     2
009C            TABPNT  RMB     2
009E            TABBUF  RMB     20
00B2 00         TABEND  FCB     0
00B3 20         FILL    FCC     ' '
00B4 23         LINO    FCC     '#'
```

```
00B5 00          DCC      FCB     0
00B6 06          DELAY    FCB     6
00B7 00          TONCH    FCB     0
00B8 00          TOFCH    FCB     0
00B9 00          RONCH    FCB     0
00BA 00          ROFCH    FCB     0
00BB 00 00       MEMEND   FDB     0
00BD             BUFFER   RMB     136

                 * DISK SYSTEM EQUATES

                 * GLOBAL VARIABLES

A080             LINBUF   EQU     $A080
AC00             BSP      EQU     $AC00
AC01             DEL      EQU     $AC01
AC02             EOL      EQU     $AC02
0004             NFER     EQU     4
0003             FEER     EQU     3
0008             EFER     EQU     $8
A840             FCB      EQU     $A840
AC06             TABCH    EQU     $AC06
AC07             BSE      EQU     $AC07
AC0B             SASN     EQU     $AC0B
AC0C             WASN     EQU     $AC0C
AC0C             ASN      EQU     WASN
AC0D             SYSFLG   EQU     $AC0D
AC11             LSTTRM   EQU     $AC11
AC16             RETRNR   EQU     $AC16
AC18             CHAR     EQU     $AC18
AC19             PRVCHR   EQU     $AC19
AC2B             DMEND    EQU     $AC2B
AC14             DBFPNT   EQU     $AC14

                 * SYSTEM CONSTANTS

000D             CR       EQU     $D
000A             LF       EQU     $A
0020             SPC      EQU     $20
0008             NL       EQU     $8
0003             EL       EQU     3
0003             DN       EQU     3

                 * SYSTEM ROUTINE ADDRESSES

B403             FMSCLS   EQU     $B403
B406             FMS      EQU     $B406

AD03             WARMS    EQU     $AD03
AD06             RENTER   EQU     $AD06
AD15             GETCHR   EQU     $AD15
AD18             PUTCHR   EQU     $AD18
AD1B             INBUF    EQU     $AD1B
```

```
AD1E            DPSTRN  EQU     $AD1E
AD24            DPCRLF  EQU     $AD24
AD27            NXTCH   EQU     $AD27
AD2D            GETFIL  EQU     $AD2D
AD33            SETEXT  EQU     $AD33
AD36            ADDBX   EQU     $AD36
AD39            OUTDEC  EQU     $AD39
AD3C            OUTHEX  EQU     $AD3C
AD3F            RPTERR  EQU     $AD3F

0200                    ORG     $0200


                * PROGRAM STARTS HERE

0200 7E 03 59   START   JMP     INITLZ
0203 7E 04 F0   RESTRT  JMP     PEDIT

                * EXTERNAL I-O ROUTINES

0206 7E AD 15   INCH    JMP     $AD15
0209 7E AD 18   OUTCH   JMP     $AD18
020C 7E AD 15   TINCH   JMP     $AD15     TAPE INPUT ROUTINE
020F 7E AD 18   TOUCH   JMP     $AD18     TAPE OUTPUT ROUTINE


                * COMMAND TABLE

0212 41         TABLE   FCC     'APPEND'
0218 00                 FCB     0
0219 14 C1              FDB     APPEND
021B 41                 FCC     'A'
021C 00                 FCB     0
021D 14 C1              FDB     APPEND
021F 42                 FCC     'BOTTOM'
0225 00                 FCB     0
0226 0B 58              FDB     BOTTOM
0228 42                 FCC     'B'
0229 00                 FCB     0
022A 0B 58              FDB     BOTTOM
022C 43                 FCC     'CHANGE'
0232 00                 FCB     0
0233 0F 9F              FDB     CHANGE
0235 43                 FCC     'COPY'
0239 00                 FCB     0
023A 11 CA              FDB     COPY
023C 43                 FCC     'CO'
023E 00                 FCB     0
023F 11 CA              FDB     COPY
0241 43                 FCC     'C'
0242 00                 FCB     0
0243 0F 9F              FDB     CHANGE
0245 44                 FCC     'DELETE'
024B 00                 FCB     0
```

```
024C 0E 58              FDB     DELETE
024E 44                 FCC     'D'
024F 00                 FCB     0
0250 0E 58              FDB     DELETE
0252 45                 FCC     'EXPAND'
0258 00                 FCB     0
0259 14 3E              FDB     EXPAND
025B 45                 FCC     'EXP'
025E 00                 FCB     0
025F 14 3E              FDB     EXPAND
0261 46                 FCC     'FIND'
0265 00                 FCB     0
0266 0B 9B              FDB     CFIND
0268 46                 FCC     'F'
0269 00                 FCB     0
026A 0B 9B              FDB     CFIND
026C 47                 FCC     'GAP'
026F 00                 FCB     0
0270 16 03              FDB     GAP
0272 48                 FCC     'HEADER'
0278 00                 FCB     0
0279 12 F2              FDB     HEADER
027B 48                 FCC     'H'
027C 00                 FCB     0
027D 12 F2              FDB     HEADER
027F 49                 FCC     'INSERT'
0285 00                 FCB     0
0286 0C 94              FDB     INSERT
0288 49                 FCC     'I'
0289 00                 FCB     0
028A 0C 94              FDB     INSERT
028C 4C                 FCC     'LOG'
028F 00                 FCB     0
0290 0B 32              FDB     EXIT
0292 4D                 FCC     'MOVE'
0296 00                 FCB     0
0297 11 AB              FDB     MOVE
0299 4D                 FCC     'MO'
029B 00                 FCB     0
029C 11 AB              FDB     MOVE
029E 4E                 FCC     'NEW'
02A1 00                 FCB     0
02A2 17 EA              FDB     NEW
02A4 4E                 FCC     'NEXT'
02A8 00                 FCB     0
02A9 0B 98              FDB     NEXT
02AB 4E                 FCC     'NUMBERS'
02B2 00                 FCB     0
02B3 0A C6              FDB     NUMSET
02B5 4E                 FCC     'NU'
02B7 00                 FCB     0
02B8 0A C6              FDB     NUMSET
02BA 4E                 FCC     'N'
02BB 00                 FCB     0
```

```
02BC 0B 98              FDB     NEXT
02BE 4F                 FCC     'OVERLAY'
02C5 00                 FCB     0
02C6 10 F5              FDB     OVERLA
02C8 4F                 FCC     'O'
02C9 00                 FCB     0
02CA 10 F5              FDB     OVERLA
02CC 50                 FCC     'PRINT'
02D1 00                 FCB     0
02D2 0A 20              FDB     PRINT
02D4 50                 FCC     'P'
02D5 00                 FCB     0
02D6 0A 20              FDB     PRINT
02D8 52                 FCC     'READ'
02DC 00                 FCB     0
02DD 17 2A              FDB     READ
02DF 52                 FCC     'RENUMBER'
02E7 00                 FCB     0
02E8 08 42              FDB     RENUMB
02EA 52                 FCC     'REN'
02ED 00                 FCB     0
02EE 08 42              FDB     RENUMB
02F0 52                 FCC     'REPLACE'
02F7 00                 FCB     0
02F8 0E 55              FDB     REPLAC
02FA 52                 FCC     'R'
02FB 00                 FCB     0
02FC 0E 55              FDB     REPLAC
02FE 53                 FCC     'SAVE'
0302 00                 FCB     0
0303 15 81              FDB     SAVE
0305 53                 FCC     'SET'
0308 00                 FCB     0
0309 13 A3              FDB     SET
030B 53                 FCC     'STOP'
030F 00                 FCB     0
0310 0B 32              FDB     EXIT
0312 53                 FCC     'S'
0313 00                 FCB     0
0314 0B 32              FDB     EXIT
0316 54                 FCC     'TAB'
0319 00                 FCB     0
031A 12 A7              FDB     TAB
031C 54                 FCC     'TOP'
031F 00                 FCB     0
0320 0B 67              FDB     TOP
0322 54                 FCC     'T'
0323 00                 FCB     0
0324 0B 67              FDB     TOP
0326 56                 FCC     'VERIFY'
032C 00                 FCB     0
032D 0B 0C              FDB     VERSET
032F 56                 FCC     'V'
0330 00                 FCB     0
```

```
0331 0B 0C              FDB     VERSET
0333 57                 FCC     'WRITE'
0338 00                 FCB     0
0339 15 91              FDB     WRITE
033B 57                 FCC     'W'
033C 00                 FCB     0
033D 15 91              FDB     WRITE
033F 58                 FCC     'X'
0340 00                 FCB     0
0341 0B 21              FDB     XCNTRL
0343 5A                 FCC     'ZONE'
0347 00                 FCB     0
0348 13 55              FDB     SZONE
034A 5A                 FCC     'Z'
034B 00                 FCB     0
034C 13 55              FDB     SZONE
034E 00                 FCB     0

                * NEW FILE STRING

034F 4E         NWFSTR  FCC     'NEW FILE:'
0358 04                 FCB     4

                * DISK EDITOR ENTRY POINT


                * INITIALIZATION ROUTINE

0359 8E 01 FF   INITLZ  LDS     #STACK
035C CE 1B 1E           LDX     #BEGPNT
035F DF 98              STX     FILBEG
0361 DF 9A              STX     FILEND      SET END ALSO
0363 09                 DEX
0364 86 0D              LDA A   #$D         SET END
0366 A7 00              STA A   0,X
0368 CE 02 03           LDX     #RESTRT
036B FF A0 48           STX     $A048       SET RESTART ADDRESS
036E CE 00 01           LDX     #1          SET ZONES
0371 DF 5E              STX     ZONE1
0373 CE 01 36           LDX     #$0136
0376 DF 60              STX     ZONE2
0378 86 46              LDA A   #70         SET UP HEADER
037A 97 97              STA A   HEDCNT
037C 4F                 CLR A
037D 97 74              STA A   OVREND      CLEAR FLAG
037F 97 7D              STA A   INCAMT
0381 97 9E              STA A   TABBUF      FIX STORAGE
0383 4A                 DEC A
0384 97 6A              STA A   NUMFLG      TURN ON NUMBERS
0386 97 6B              STA A   VERFLG      ALSO VERIFICATION
0388 CE 04 FA           LDX     #EDIT
038B FF AC 16           STX     RETRNR
038E 4F                 CLR A               CLEAR FLAGS
```

```
038F 97 90              STA A  INZFLG
0391 97 24              STA A  LASTIN
0393 97 20              STA A  DSKFLG
0395 97 21              STA A  DRWFLG
0397 FE AC 2B           LDX    DMEND
039A DF BB              STX    MEMEND     SET END
039C 96 BB              LDA A  MEMEND     GET MSB
039E 81 77              CMP A  #$77
03A0 23 04              BLS    DEDIT2
03A2 86 77              LDA A  #$77
03A4 97 BB              STA A  MEMEND
03A6 D6 BC      DEDIT2  LDA B  MEMEND+1
03A8 D7 2A      DEDI35  STA B  BUFEND+1
03AA D7 2C              STA B  RBFEND+1
03AC 16                 TAB                CALCULATE BUFFER END
03AD 54                 LSR B
03AE 54                 LSR B
03AF 54                 LSR B
03B0 10                 SBA
03B1 97 29              STA A  BUFEND     SET END
03B3 96 BB              LDA A  MEMEND     GET END
03B5 4A                 DEC A             SET READ BUFFER
03B6 97 2B              STA A  RBFEND
03B8 CE 18 9D           LDX    #RFCB      POINT TO FCB
03BB BD AD 2D           JSR    GETFIL     GET NAME
03BE 25 54              BCS    DEDI42
03C0 86 01              LDA A  #1         TXT CODE
03C2 BD AD 33           JSR    SETEXT     SET DEFAULT
03C5 C6 0C              LDA B  #12
03C7 CE 18 A0           LDX    #RFCB+3    SET POINTER
03CA DF 2D              STX    XXX1
03CC CE 19 E0           LDX    #WFCB+3
03CF DF 2F              STX    XXX2
03D1 DE 2D      DEDI37  LDX    XXX1
03D3 A6 00              LDA A  0,X
03D5 08                 INX
03D6 DF 2D              STX    XXX1
03D8 DE 2F              LDX    XXX2
03DA A7 00              STA A  0,X
03DC 08                 INX
03DD DF 2F              STX    XXX2
03DF 5A                 DEC B
03E0 26 EF              BNE    DEDI37
03E2 CE 18 9D           LDX    #RFCB      SET TO FCB
03E5 86 01              LDA A  #1         OPEN FOR READ
03E7 A7 00              STA A  0,X
03E9 BD B4 06           JSR    FMS        CALL FMS
03EC 27 0B              BEQ    DEDIT4
03EE A6 01              LDA A  1,X        CHECK ERROR
03F0 81 04              CMP A  #NFER      NO FILE ERR?
03F2 26 43              BNE    DEDI55
03F4 7C 00 24           INC    LASTIN     SET FOR NEW
03F7 20 27              BRA    DEDIT5
03F9 B6 AC 11   DEDIT4  LDA A  LSTTRM     CHECK TERM
```

```
03FC 81 0D            CMP A  #$D       IS IT TERM?
03FE 27 3C            BEQ    DEDIT6
0400 B1 AC 02         CMP A  EOL
0403 27 37            BEQ    DEDIT6
0405 CE 19 DD         LDX    #WFCB     POINT TO FCB
0408 BD AD 2D         JSR    GETFIL    GET NAME
040B 25 07            BCS    DEDI42    ERROR?
040D 86 01            LDA A  #1        SET TXT EXT
040F BD AD 33         JSR    SETEXT
0412 20 0C            BRA    DEDIT5
0414 CE 18 32  DEDI42 LDX    #ILST     POINT TO STRNG
0417 BD 06 06  DEDI45 JSR    PSTRNG    OUTPUT IT
041A BD B4 03  DEDI47 JSR    FMSCLS    CLOSE FMS
041D 7E AD 03         JMP    WARMS
0420 CE 19 DD  DEDIT5 LDX    #WFCB     POINT TO FCB
0423 86 02            LDA A  #2        OPEN FOR WRITE
0425 A7 00            STA A  0,X
0427 BD B4 06         JSR    FMS
042A 27 7C            BEQ    DEDIT8    ERROR?
042C A6 01            LDA A  1,X       CHECK ERROR
042E 81 03            CMP A  #FEER
0430 26 05            BNE    DEDI55    SERIOUS?
0432 CE 18 44         LDX    #FEST     POINT TO STRING
0435 20 E0            BRA    DEDI45
0437 BD AD 3F  DEDI55 JSR    RPTERR    REPORT ERROR
043A 20 DE            BRA    DEDI47    EXIT
043C CE 18 9D  DEDIT6 LDX    #RFCB     POINT TO FCB
043F BD 16 5E         JSR    CLSFC1    CLOSE FILE
0442 CE 18 A1  DEDI61 LDX    #RFCB+4   POINT TO NAME
0445 C6 0B            LDA B  #11       SET COUNTER
0447 A6 00     DEDI62 LDA A  0,X       GET CHARACTER
0449 A7 31            STA A  49,X      MOVE TO FLR
044B 08               INX              BUMP TO NEXT
044C 5A               DEC B            DEC THE COUNT
044D 26 F8            BNE    DEDI62
044F CE 18 CE         LDX    #RFCB+49  SET POINTER
0452 6F 0C            CLR    12,X      CLEAR EXT
0454 86 05            LDA A  #5        BAK CODE
0456 BD AD 33         JSR    SETEXT    SET EXTENSION
0459 CE 18 9D         LDX    #RFCB     SET FCB
045C 86 0D            LDA A  #13       RENAME CODE
045E A7 00            STA A  0,X
0460 BD B4 06         JSR    FMS       DO RENAME
0463 27 34            BEQ    DEDIT7    ERRORS?
0465 A6 01            LDA A  1,X       CHECK ERROR
0467 81 03            CMP A  #FEER
0469 26 CC            BNE    DEDI55    SERIOUS?
046B CE 18 50         LDX    #DLST     POINT TO STRING
046E BD 06 06         JSR    PSTRNG    OUTPUT IT
0471 BD AD 15         JSR    GETCHR    GET RESPONSE
0474 81 5F            CMP A  #$5F      LOWER CASE?
0476 23 02            BLS    DEDI67
0478 80 20            SUB A  #$20      REMOVE BIAS
047A 81 59     DEDI67 CMP A  #'Y       IS IT YES?
```

```
047C 26 9C            BNE     DEDI47     EXIT?
047E CE 18 9D         LDX     #RFCB      SET FCB
0481 86 0C            LDA A   #12        DELETE CODE
0483 A7 00            STA A   0,X
0485 BD B4 06         JSR     FMS        GO DELETE
0488 26 AD            BNE     DEDI55     ERROR?
048A CE 18 A1         LDX     #RFCB+4    POINT TO NAME
048D C6 0B            LDA B   #11        SET COUNT
048F A6 31     DEDI68 LDA A   49,X       GET CHARACTER
0491 A7 00            STA A   0,X        MOVE IT
0493 08               INX                BUMP TO NEXT
0494 5A               DEC B              DEC THE COUNT
0495 26 F8            BNE     DEDI68
0497 20 A9            BRA     DEDI61     REPEAT
0499 CE 18 9D  DEDIT7 LDX     #RFCB      SET TO FCB
049C 86 01            LDA A   #1         OPEN FOR READ
049E A7 00            STA A   0,X
04A0 BD B4 06         JSR     FMS        CALL FMS
04A3 26 92            BNE     DEDI55     ERRORS?
04A5 7E 04 20         JMP     DEDIT5
04A8 B6 19 E0  DEDIT8 LDA A   WFCB+3     GET DR NUM
04AB CE A8 40         LDX     #FCB       SET POINTER
04AE A7 03            STA A   3,X        SET DRIVE
04B0 86 10            LDA A   #16        OPEN SIR
04B2 A7 00            STA A   0,X
04B4 BD B4 06         JSR     FMS        CALL FMS
04B7 25 1F            BCS     DEDI88
04B9 86 07            LDA A   #7         GET IR
04BB A7 00            STA A   0,X
04BD BD B4 06         JSR     FMS        CALL FMS
04C0 25 16            BCS     DEDI88     ERROR?
04C2 A6 15            LDA A   21,X       GET COUNT
04C4 E6 16            LDA B   22,X
04C6 B1 18 B2         CMP A   RFCB+21    CHECK HI
04C9 22 0D            BHI     DEDI88
04CB 25 05            BLO     DEDI85
04CD F1 18 B3         CMP B   RFCB+22    CHECK LO
04D0 24 06            BHS     DEDI88
04D2 CE 18 84  DEDI85 LDX     #LSST      POINT TO STRING
04D5 BD 06 06         JSR     PSTRNG     OUTPUT IT
04D8 96 24     DEDI88 LDA A   LASTIN     NEW FILE?
04DA 27 0B            BEQ     DEDIT9
04DC CE 03 4F         LDX     #NWFSTR    POINT TO STRING
04DF BD 06 06         JSR     PSTRNG     OUTPUT IT
04E2 7C 00 90         INC     INZFLG
04E5 20 06            BRA     DEDI95
04E7 7C 00 20  DEDIT9 INC     DSKFLG     SET FLAG
04EA BD 17 3E         JSR     READ1      FILL BUFFER
04ED 7F 00 20  DEDI95 CLR     DSKFLG

                      * RESTART ENTRY POINT

04F0 7F 00 6C  PEDIT  CLR     REPEAT     DISABLE COMMAND REPEAT
```

```
04F3 DE 98              LDX     FILBEG      POINT TO BEGIN
04F5 DF 48              STX     CURPOS
04F7 7F 00 6D           CLR     MSLFLG


                * MAIN EDIT LOOP

04FA 8E 01 FF  EDIT     LDS     #STACK      SETUP STACK POINTER
04FD DF 40              STX     TEMP        SAVE POINTER
04FF DE 48              LDX     CURPOS      SET CURRENT POSITION
0501 DF 4A              STX     NEWPOS      SAVE IT
0503 CE 00 6E           LDX     #PSTZFL
0506 4F                 CLR A               CLEAR ACC.
0507 A7 00      EDIT1   STA A   0,X          CLEAR OUT LOCATION
0509 08                 INX                 BUMP POINTER
050A 8C 00 8F           CPX     #CHRCNT+1
050D 26 F8              BNE     EDIT1
050F DE 40              LDX     TEMP        RESTORE POINTER
0511 96 90              LDA A   INZFLG      INITIALIZE?
0513 27 06              BEQ     EDIT2
0515 7F 00 90           CLR     INZFLG
0518 7E 0C B6           JMP     INSER4      GO INSERT LINES
051B 96 6D      EDIT2   LDA A   MSLFLG      MULTIPLE ST. PER LINE?
051D 26 3A              BNE     EDIT55
051F 97 8F              STA A   CHRCNT+1
0521 CE 00 BD           LDX     #BUFFER     SET POINTER
0524 BD 05 DF           JSR     PCRLF
0527 86 23              LDA A   #PROMPT     SETUP PROMPT
0529 BD 02 09           JSR     OUTCH       OUTPUT IT
052C BD 02 06  EDIT25   JSR     INCH        GET INPUT CHAR.
052F 81 12              CMP A   #$12        IS IT REPEAT KEY?
0531 27 05              BEQ     EDIT26      BRANCH IF SO
0533 BD 06 1F           JSR     INCHR1      CHECK CHARACTER
0536 20 0B              BRA     EDIT31
0538 7D 00 6C  EDIT26   TST     REPEAT      IS REPEAT ENABLED?
053B 26 13              BNE     EDIT5       BRANCH IF SO
053D 7E 05 CF           JMP     ERROR       ELSE, IT'S AN ERROR
0540 BD 06 1C  EDIT3    JSR     INCHAR
0543 27 B5      EDIT31  BEQ     EDIT
0545 A7 00      EDIT4   STA A   0,X          PUT CHAR IN BUFFER
0547 81 0D              CMP A   #CRGRET     IS IT A C.R.
0549 27 05              BEQ     EDIT5
054B BD 06 56           JSR     BUFLIM      IS BUFFER FULL?
054E 20 F0              BRA     EDIT3       REPEAT
0550 CE 00 BD  EDIT5    LDX     #BUFFER     RESTORE POINTER
0553 DF 44              STX     BUFPNT
0555 86 01              LDA A   #1
0557 97 6C              STA A   REPEAT      ENABLE COMMAND REPEAT
0559 7F 00 6D  EDIT55   CLR     MSLFLG
055C BD 06 6E           JSR     FINDL       PROCESS LINE INFO
055F 4F                 CLR A
0560 97 76              STA A   LINFLG      CLEAR FLAS
0562 97 8D              STA A   DRCTN
0564 DF 4A              STX     NEWPOS      SAVE POINTER
```

```
0566 DE 44          LDX   BUFPNT
0568 BD 06 15       JSR   SKIPSP    SKIP SPACES
056B DF 44          STX   BUFPNT    SAVE POINTER
056D 81 3D          CMP A #'=       IS IT AN '='?
056F 26 08          BNE   EDIT56
0571 08             INX             BUMP POINTER
0572 DF 44          STX   BUFPNT
0574 CE 0C 71       LDX   #EQUALS
0577 20 4A          BRA   EDIT85    GO TO IT
0579 BD 07 FF EDIT56 JSR  TSTEND    TEST END
057C 26 05          BNE   EDIT58
057E CE 0A 20       LDX   #PRINT    POINT TO PRINT
0581 20 40          BRA   EDIT85
0583 DF 40    EDIT58 STX  TEMP
0585 CE 02 12       LDX   #TABLE    POINT TO TABLE
0588 DF 9C    EDIT6  STX  TABPNT    SAVE IT
058A 6D 00          TST   0,X       IS IT NULL?
058C 27 29          BEQ   EDIT8
058E 81 61          CMP A #'a       CHECK LOWER CASE
0590 25 02          BLO   EDIT63
0592 80 20          SUB A #$20      REMOVE BIAS
0594 A1 00    EDIT63 CMP A 0,X      CHECK CHARACTER
0596 26 0C          BNE   EDIT7     ARE THEY EQUAL?
0598 DE 44          LDX   BUFPNT    RESTORE POINTER
059A 08             INX             BUMP IT
059B A6 00    EDIT65 LDA A 0,X      GET NEXT CHAR.
059D DF 44          STX   BUFPNT    SAVE POINTER
059F DE 9C          LDX   TABPNT
05A1 08             INX             BUMP THE TABLE PNTR
05A2 20 E4          BRA   EDIT6
05A4 08       EDIT7  INX            BUMP THE POINTER
05A5 6D 00          TST   0,X       IS IT NULL?
05A7 26 FB          BNE   EDIT7
05A9 08             INX             BUMP POINTER 3 TIMES
05AA 08             INX
05AB 08             INX
05AC 6D 00          TST   0,X       END OF TABLE?
05AE 27 1F          BEQ   ERROR     REPORT ERROR
05B0 09             DEX
05B1 DF 9C          STX   TABPNT    SAVE THE POINTER
05B3 DE 40          LDX   TEMP
05B5 20 E4          BRA   EDIT65    REPEAT
05B7 08       EDIT8  INX            BUMP THE POINTER
05B8 EE 00          LDX   0,X       GET ADDRESS
05BA 8C 0C 94       CPX   #INSERT   IS IT INSERT?
05BD 26 04          BNE   EDIT85
05BF 96 80          LDA A INLMFL
05C1 26 0A          BNE   EDIT88
05C3 BD 09 CA EDIT85 JSR  TSTOVR    LIMITS?
05C6 26 1A          BNE   NOTFND
05C8 4F             CLR A           CLEAR FLAGS
05C9 97 73          STA A OVRBEG
05CB 97 74          STA A OVREND
05CD 6E 00    EDIT88 JMP  0,X       GO TO IT
```

```
                    * ERROR ROUTINE

05CF CE 05 DD  ERROR   LDX    #ERRSTR   POINT TO STRING

                    * PRINT ERROR MESSAGE

05D2 8D 32     PREROR  BSR    PSTRNG
05D4 7F 00 6C          CLR    REPEAT    DISABLE COMMAND REPEAT
05D7 7F 00 6D          CLR    MSLFLG    CLEAR FLAG
05DA 7E 04 FA          JMP    EDIT      RETURN

05DD 3F        ERRSTR  FCC    '?'
05DE 04                FCB    4

                    * PRINT CARRIAGE RETURN & LINE FEED

05DF 7E AD 24  PCRLF   JMP    DPCRLF    DO CR & LF

                    * REPORT LINE NOT FOUND

05E2 CE 05 E7  NOTFND  LDX    #NOFSTR   POINT TO STRING
05E5 20 EB             BRA    PREROR

05E7 4E        NOFSTR  FCC    'NO SUCH LINE'
05F3 04                FCB    4

                    * REPORT SYNTAX ERROR

05F4 CE 05 F9  SYNERR  LDX    #SYNSTR   POINT TO STRING
05F7 20 D9             BRA    PREROR

05F9 53        SYNSTR  FCC    'SYNTAX ERROR'
0605 04                FCB    4

                    * PRINT STRING ROUTINE

0606 8D D7     PSTRNG  BSR    PCRLF     DO CR & LF
0608 A6 00     PDATA1  LDA A  0,X       GET CHARACTER
060A 81 04             CMP A  #4        IS IT TERM?
060C 27 0D             BEQ    SKIPS2
060E BD 02 09          JSR    OUTCH     OUTPUT IT
0611 08                INX
0612 20 F4             BRA    PDATA1

                    * SKIP ALL SPACES

0614 08        SKIPSA  INX
0615 A6 00     SKIPSP  LDA A  0,X       GET A CHAR.
```

```
0617 81 20              CMP A  #'         IS IT A SPACE?
0619 27 F9              BEQ    SKIPSA     REPEAT
061B 39         SKIPS2  RTS               RETURN

                * INPUT AND CHECK CHARACTER

061C BD 02 06   INCHAR  JSR    INCH       GET CHAR
061F B1 AC 00   INCHR1  CMP A  BSP        IS IT A BACKSPACE?
0622 26 1D              BNE    INCHR3
0624 8C 00 BD           CPX    #BUFFER    BUFFER BEGINNING?
0627 27 29              BEQ    INCHR4
0629 09                 DEX               DEC THE POINTER
062A 7A 00 8F           DEC    CHRCNT+1
062D B6 AC 07           LDA A  BSE        CHECK BSP ECHO
0630 81 08              CMP A  #8         IS IT ↑H?
0632 26 08              BNE    INCHR2
0634 86 20              LDA A  #$20       LOAD UP SPACE
0636 BD 02 09           JSR    OUTCH      OUTPUT IT
0639 B6 AC 07           LDA A  BSE
063C BD 02 09   INCHR2  JSR    OUTCH
063F 20 DB              BRA    INCHAR
0641 B1 AC 01   INCHR3  CMP A  DEL        IS IT A DELETE?
0644 27 0C              BEQ    INCHR4
0646 81 1F              CMP A  #$1F       IS IT CONTROL?
0648 22 04              BHI    INCH35
064A 81 0D              CMP A  #CRGRET    IS IT A C.R.?
064C 26 CE              BNE    INCHAR
064E 7C 00 8F   INCH35  INC    CHRCNT+1   INC CHAR. COUNT
0651 39                 RTS               RETURN
0652 4F         INCHR4  CLR A
0653 97 6C              STA A  REPEAT     DISABLE COMMAND REPEAT
0655 39         INCHR5  RTS               RETURN

                * CHECK FOR BUFFER OVERFLOW

0656 08         BUFLIM  INX               BUMP THE POINTER
0657 8C 01 45           CPX    #BUFFER+136
065A 26 F9              BNE    INCHR5
065C 86 07      OVER    LDA A  #BELL      LOAD UP BELL
065E BD 02 09           JSR    OUTCH      OUTPUT IT
0661 BD 02 06           JSR    INCH       GET NEW CHAR.
0664 B1 AC 00           CMP A  BSP        IS IT BACKSPACE?
0667 26 F3              BNE    OVER
0669 09                 DEX               DEC THE POINTER
066A 7A 00 8F           DEC    CHRCNT+1
066D 39                 RTS               RETURN

                * PROCESS LINE ROUTINE

066E 8D A5      FINDL   BSR    SKIPSP     SKIP SPACES
0670 81 3D              CMP A  #'=        IS IT '='?
```

```
0672 27 0A              BEQ     FINDL0
0674 7C 00 76           INC     LINFLG      SET FLAG
0677 BD 08 DC           JSR     CLASS       CLASSIFY CHAR.
067A C1 01              CMP B   #1          IS IT A LETTER
067C 23 03              BLS     FINDL2
067E DE 4A      FINDL0  LDX     NEWPOS      SET POINTER
0680 39         FINDL1  RTS                 RETURN
0681 27 23      FINDL2  BEQ     FIND1
0683 20 14              BRA     FIND

                * TARGET ENTRY POINT

0685 8D 03      FINDT   BSR     FINDT0
0687 DF 94              STX     TRGLIN      SAVE TARGET POS.
0689 39                 RTS                 RETURN
068A 7F 00 8D   FINDT0  CLR     DRCTN
068D BD 08 D9           JSR     SKPCLS
0690 C1 01              CMP B   #1          IS IT A LETTER?
0692 23 03              BLS     FINDT2
0694 7E 05 F4   FINDT1  JMP     SYNERR      REPORT SYNTAX ERROR
0697 27 7D      FINDT2  BEQ     FIND62
0699 7F 00 8D   FIND    CLR     DRCTN       CLEAR DIRECTION
069C 91 B4              CMP A   LINO        IS IT LINE NUMBER
069E 26 2E              BNE     FIND2
06A0 08                 INX                 BUMP THE POINTER
06A1 BD 06 15           JSR     SKIPSP
06A4 DF 44              STX     BUFPNT
06A6 BD 08 FE   FIND1   JSR     BCDCON
06A9 DF 40              STX     TEMP        SAVE POINTER
06AB DE 4A              LDX     NEWPOS
06AD 96 91              LDA A   NUMBER      GET NUMBER
06AF A1 00              CMP A   0,X         COMPARE IT
06B1 26 0A              BNE     FIND14
06B3 96 92              LDA A   NUMBER+1    GET NEXT NUM.
06B5 A1 01              CMP A   1,X         COMPARE
06B7 26 04              BNE     FIND14
06B9 96 93              LDA A   NUMBER+2
06BB A1 02              CMP A   2,X
06BD 24 03      FIND14  BCC     FIND16
06BF 7A 00 8D           DEC     DRCTN       SET DIRECTION
06C2 DE 40      FIND16  LDX     TEMP        RESTORE POINTER
06C4 BD 09 54           JSR     FNDNUM      FIND LINE NUMBER
06C7 27 B7              BEQ     FINDL1
06C9 D7 75              STA B   NOCURL
06CB 7E 09 99           JMP     BAKONE
06CE BD 07 FF   FIND2   JSR     TSTEND
06D1 26 0A              BNE     FIND3
06D3 7D 00 77           TST     NXTFLG      CHECK FLAG
06D6 27 A6              BEQ     FINDL0
06D8 DE 4A              LDX     NEWPOS      SET POINTER
06DA 7E 09 B8           JMP     UPONE       UP ONE LINE
06DD 81 21      FIND3   CMP A   #'!         IS IT A "!"?
06DF 26 08              BNE     FIND4
06E1 08                 INX                 BUMP THE POINTER
```

```
06E2 DF 44              STX     BUFPNT
06E4 DE 9A              LDX     FILEND      SET POINTER
06E6 7E 09 99           JMP     BAKONE      BACKUP ONE LINE
06E9 81 5E      FIND4   CMP A   #'↑         IS IT A "↑"?
06EB 26 09              BNE     FIND5
06ED 7A 00 8D           DEC     DRCTN       SET DIRECTION
06F0 08                 INX                 BUMP THE POINTER
06F1 DF 44              STX     BUFPNT
06F3 DE 98              LDX     FILBEG      SET POINTER TO BEGIN
06F5 39                 RTS                 RETURN
06F6 81 2B      FIND5   CMP A   #'+         IS IT A "+"?
06F8 27 07              BEQ     FIND6
06FA 81 2D              CMP A   #'-         IS IT A "-"?
06FC 26 47              BNE     FIND7
06FE 7A 00 8D           DEC     DRCTN       DEC DIRECTION
0701 08         FIND6   INX                 BUMP THE POINTER
0702 BD 08 D9           JSR     SKPCLS      SKIP SPACES
0705 C1 01              CMP B   #1          IS IT NUMBER?
0707 27 0D              BEQ     FIND62
0709 23 3A              BLS     FIND7
070B D6 76              LDA B   LINFLG      TEST FLAG
070D 27 85              BEQ     FINDT1
070F BD 09 4C           JSR     CLRNUM
0712 DE 4A              LDX     NEWPOS      SET POINTER
0714 20 1A              BRA     FIND66
0716 BD 08 FE   FIND62  JSR     BCDCON      CONVERT NUMBER
0719 DE 4A              LDX     NEWPOS      SET POINTER
071B 7D 00 76           TST     LINFLG      CHECK FLAG
071E 26 08              BNE     FIND65
0720 BD 09 E1   FIND63  JSR     TSTNUM      IS IT ZERO?
0723 27 1F              BEQ     FIND67
0725 BD 09 D3           JSR     DECNUM      DEC NUMBER
0728 BD 09 E1   FIND65  JSR     TSTNUM      TEST NUMBER
072B 27 17              BEQ     FIND67
072D BD 09 D3           JSR     DECNUM
0730 BD 08 60   FIND66  JSR     NXTLIN      GOTO NEXT LINE
0733 BD 09 CA           JSR     TSTOVR      CHECK LIMITS
0736 27 F0              BEQ     FIND65
0738 96 74              LDA A   OVREND      BEGINNING?
073A 26 08              BNE     FIND67
073C BD 09 E1           JSR     TSTNUM
073F 26 03              BNE     FIND67
0741 7C 00 80           INC     INLMFL
0744 39         FIND67  RTS                 RETURN
0745 8D 4F      FIND7   BSR     SETDEL      SET DELIMITER
0747 8D 6D      FIN702  BSR     ZONE        SET ZONE
0749 DE 4A              LDX     NEWPOS      SET POINTER
074B BD 08 60           JSR     NXTLIN      GO TO NEXT
074E 08         FIND71  INX                 BUMP POINTER 3 TIMES
074F 08                 INX
0750 08                 INX
0751 BD 07 F6   FIN711  JSR     FIXZON      CHECK FOR POSTZONE
0754 BD 09 D3   FIND72  JSR     DECNUM      DEC COLUMN COUNT
0757 27 22              BEQ     FIND75
```

```
0759 C6 0D              LDA B  #CRGRET
075B E1 00              CMP B  0,X      TEST FOR C.R.
075D 27 03              BEQ    FIND73
075F 08                 INX             BUMP THE POINTER
0760 20 F2              BRA    FIND72   REPEAT
0762 96 86    FIND73    LDA A  CHGFLG   CHECK IF CHANGE
0764 26 DE              BNE    FIND67
0766 08                 INX             BUMP THE POINTER
0767 9C 9A              CPX    FILEND   END OF FILE ?
0769 27 26              BEQ    FIND78
076B 09                 DEX
076C 96 86    FIND74    LDA A  CHGFLG
076E 26 D4              BNE    FIND67
0770 09                 DEX             DEC THE POINTER
0771 BD 08 60           JSR    NXTLIN
0774 BD 09 CA           JSR    TSTOVR   CHECK LIMITS
0777 26 17              BNE    FIND77
0779 20 D3              BRA    FIND71
077B BD 07 F6 FIND75    JSR    FIXZON   FIX UP ZONE
077E BD 08 6A           JSR    STRING   PROCESS STRING
0781 7D 00 71           TST    FNDFLG   FIND IT?
0784 27 E6              BEQ    FIND74
0786 5F                 CLR B
0787 96 86              LDA A  CHGFLG
0789 26 05              BNE    FIND77
078B BD 09 9F           JSR    BAKON2
078E DF 94              STX    TRGLIN   SAVE TARGET POINTER
0790 39       FIND77    RTS             RETURN
0791 86 01    FIND78    LDA A  #1       SET FLAG
0793 97 74              STA A  OVREND
0795 39                 RTS             RETURN

                * SET UP DELIMITERS

0796 97 96    SETDEL    STA A  DELIM    SAVE DELIMITER
0798 5F                 CLR B
0799 08                 INX             BUMP THE POINTER
079A DF 4E              STX    STRNGB   SAVE BEGINNING
079C A6 00    SETDE2    LDA A  0,X      GET A CHARACTER
079E 8D 5F              BSR    TSTEND
07A0 27 08              BEQ    SETDE4
07A2 91 96              CMP A  DELIM    IS IT A DELIMITER?
07A4 27 04              BEQ    SETDE4
07A6 08                 INX             BUMP THE POINTER
07A7 5C                 INC B           BUMB COUNTER
07A8 20 F2              BRA    SETDE2   REPEAT
07AA DF 50    SETDE4    STX    STRNGE   SAVE END OF STRING
07AC D7 7C              STA B  STRCNT
07AE 8D 4F              BSR    TSTEND
07B0 27 01              BEQ    SETDE5
07B2 08                 INX             BUMP THE POINTER
07B3 DF 44    SETDE5    STX    BUFPNT
07B5 39                 RTS             RETURN
```

```
                    * SET UP ZONE

07B6 BD 08 DC  ZONE     JSR     CLASS       GO CLASSIFY CHAR.
07B9 7F 00 6E           CLR     PSTZFL      CLEAR FLAG
07BC C1 01              CMP B   #1          IS IT A NUMBER
07BE 26 17              BNE     ZONE3
07C0 BD 08 FE           JSR     BCDCON      CONVERT NUMBER
07C3 8D 1B              BSR     CMPZN1      CHECK ZONE1
07C5 25 10              BCS     ZONE3
07C7 8D 22              BSR     CMPZN2      CHECK ZONE2
07C9 22 0C              BHI     ZONE3
07CB 7C 00 6E           INC     PSTZFL      SET FLAG
07CE 96 91              LDA A   NUMBER      PUT NUM IN ZONE BUF.
07D0 97 62              STA A   ZONBUF
07D2 96 92              LDA A   NUMBER+1
07D4 97 63              STA A   ZONBUF+1
07D6 39                 RTS                 RETURN
07D7 96 5E     ZONE3    LDA A   ZONE1       PUT ZONE1 IN BUF.
07D9 97 62              STA A   ZONBUF
07DB 96 5F              LDA A   ZONE1+1
07DD 97 63              STA A   ZONBUF+1
07DF 39                 RTS                 RETURN

                    * COMPARE ZONE1 TO NUMBER

07E0 D6 91     CMPZN1   LDA B   NUMBER      GET NUMBER
07E2 D1 5E              CMP B   ZONE1       CHECK
07E4 26 04              BNE     CMPZ14
07E6 D6 92              LDA B   NUMBER+1
07E8 D1 5F              CMP B   ZONE1+1
07EA 39        CMPZ14   RTS                 RETURN

                    * COMPARE ZONE2 TO NUMBER

07EB D6 91     CMPZN2   LDA B   NUMBER      GET NUMBER
07ED D1 60              CMP B   ZONE2       CHECK
07EF 26 04              BNE     CMPZ24
07F1 D6 92              LDA B   NUMBER+1
07F3 D1 61              CMP B   ZONE2+1
07F5 39        CMPZ24   RTS                 RETURN

                    * PUT CORRECT ZONE IN NUMBER

07F6 96 62     FIXZON   LDA A   ZONBUF      GET ZONE
07F8 97 91              STA A   NUMBER      PUT IN NUMBER
07FA 96 63              LDA A   ZONBUF+1
07FC 97 92              STA A   NUMBER+1
07FE 39                 RTS                 RETURN
```

```
                    * TEST TERMINATOR (C.R. OR EOL)

07FF 81 0D      TSTEND  CMP A  #CRGRET   IS IT C.R.?
0801 27 03              BEQ    TSTEN2
0803 B1 AC 02           CMP A  EOL
0806 39         TSTEN2  RTS              RETURN

                    * BUMP NUMBER BY 1, .1, OR .01

0807 86 01      BMPNUM  LDA A  #1
0809 D6 7D              LDA B  INCAMT    CHECK AMOUNT
080B 27 0C              BEQ    INCNUM
080D 2A 02              BPL    BMPNU4
080F 86 10              LDA A  #$10      SET BUMP
0811 9B 93      BMPNU4  ADD A  NUMBER+2  ADD IN
0813 19                 DAA              ADJUST IT
0814 97 93              STA A  NUMBER+2  SAVE
0816 25 01              BCS    INCNUM
0818 39                 RTS              RETURN

                    * INCREMENT NUMBER BY ONE

0819 86 01      INCNUM  LDA A  #1        SET UP ONE
081B 5F                 CLR B
081C 9B 92              ADD A  NUMBER+1  ADD IN ONE
081E 19                 DAA              ADJUST IT
081F 97 92              STA A  NUMBER+1  SAVE IT
0821 17                 TBA
0822 99 91              ADC A  NUMBER
0824 19                 DAA              ADJUST NUMBER
0825 97 91              STA A  NUMBER
0827 39                 RTS              RETURN

                    * PUT NUMBER AT X

0828 96 91      PUTNUM  LDA A  NUMBER    GET NUMBER
082A A7 00              STA A  0,X       SAVE IT
082C 96 92              LDA A  NUMBER+1
082E A7 01              STA A  1,X
0830 96 93              LDA A  NUMBER+2
0832 A7 02              STA A  2,X
0834 39                 RTS              RETURN

                    * GET NUMBER FROM X

0835 A6 00      GETNUM  LDA A  0,X       GET NUMBER
0837 97 91              STA A  NUMBER    SAVE IT
0839 A6 01              LDA A  1,X
083B 97 92              STA A  NUMBER+1
083D A6 02              LDA A  2,X
```

```
083F 97 93              STA A  NUMBER+2
0841 39                 RTS                 RETURN

                * RENUMBER FILE

0842 BD 0B 6E  RENUMB   JSR    TFORCR
0845 DE 98              LDX    FILBEG       SET POINTER
0847 7F 00 7D  RENUM1   CLR    INCAMT
084A BD 09 4C           JSR    CLRNUM       CLEAR NUMBER
084D 8D B8     RENUM2   BSR    BMPNUM       BUMP NUMBER
084F 8D D7              BSR    PUTNUM       SAVE IT
0851 BD 09 B8           JSR    UPONE
0854 96 74              LDA A  OVREND       HIT LIMIT?
0856 27 F5              BEQ    RENUM2       REPEAT
0858 96 84              LDA A  CHKFLG       CHECK FLAG
085A 27 01              BEQ    RENUM4
085C 39                 RTS                 RETURN
085D 7E 0A 4B  RENUM4   JMP    PRINT6       RETURN

                * GO TO NEXT LINE

0860 96 8D     NXTLIN   LDA A  DRCTN        CHECK DIRECTION
0862 2B 03              BMI    NXTLI2
0864 7E 09 B8           JMP    UPONE        MOVE UP ONE
0867 7E 09 99  NXTLI2   JMP    BAKONE       MOVE BACK ONE

                * PROCESS STRING ROUTINE

086A 7F 00 71  STRING   CLR    FNDFLG       CLEAR FLAG
086D D6 7C              LDA B  STRCNT
086F 26 06              BNE    STRIN1
0871 7C 00 71           INC    FNDFLG       FOUND NULL STRING
0874 DF 5C              STX    LASTNO       SAVE POINTER
0876 39                 RTS                 RETURN
0877 C6 0D     STRIN1   LDA B  #CRGRET
0879 DF 4C              STX    SRCHPT       SAVE POINTER
087B DF 5C              STX    LASTNO       SAVE POINTER
087D DE 4E              LDX    STRNGB       POINT TO BEGIN
087F DF 56     STRIN2   STX    STRPNT       SAVE POINTER
0881 A6 00              LDA A  0,X          GET A CHARACTER
0883 DE 4C              LDX    SRCHPT       RESTORE POINTER
0885 E1 00     STRIN3   CMP B  0,X          C. RET. ?
0887 27 21              BEQ    STRIN4
0889 91 B5              CMP A  DCC          CHECK DC CHAR
088B 27 21              BEQ    STRIN5       MAKE MATCH!
088D A1 00              CMP A  0,X          COMP. CHAR.
088F 27 1D              BEQ    STRIN5
0891 7D 00 6E           TST    PSTZFL       POST ZONE?
0894 26 14              BNE    STRIN4
0896 7D 00 71           TST    FNDFLG       FOUND?
0899 26 22              BNE    STRIN6
```

```
089B 08                    INX             BUMP THE POINTER
089C DF 5C                 STX     LASTNO  SAVE IT
089E 36                    PSH A           SAVE ACC.
089F 37                    PSH B
08A0 BD 08 19              JSR     INCNUM  INC NUMBER
08A3 BD 07 EB              JSR     CMPZN2  CHECK ZONE2
08A6 33                    PUL B           RESTORE ACC
08A7 32                    PUL A
08A8 23 DB                 BLS     STRIN3
08AA 7F 00 71  STRIN4      CLR     FNDFLG  CLEAR FLAG
08AD 39                    RTS             RETURN
08AE 08        STRIN5      INX
08AF DF 4C                 STX     SRCHPT  SAVE IT
08B1 7C 00 71              INC     FNDFLG  SET FLAG
08B4 DE 56                 LDX     STRPNT  POINT TO STRING
08B6 08                    INX             BUMP THE POINTER
08B7 9C 50                 CPX     STRNGE  END OF STRING?
08B9 27 0F                 BEQ     STRIN7
08BB 20 C2                 BRA     STRIN2
08BD DE 5C     STRIN6      LDX     LASTNO  RESTORE POINTER
08BF 08                    INX
08C0 BD 08 19              JSR     INCNUM  BUMP NUMBER
08C3 BD 07 EB              JSR     CMPZN2  CHECK ZONE
08C6 23 A2                 BLS     STRING
08C8 20 E0                 BRA     STRIN4
08CA D6 7C     STRIN7      LDA B   STRCNT  GET COUNT
08CC 27 08     STRIN8      BEQ     STRIN9
08CE 37                    PSH B           SAVE
08CF BD 08 19              JSR     INCNUM  FIX COL
08D2 33                    PUL B
08D3 5A                    DEC B           DEC COUNT
08D4 26 F6                 BNE     STRIN8
08D6 DE 5C     STRIN9      LDX     LASTNO
08D8 39                    RTS             RETURN

                * SKIP AND CLASSIFY

08D9 BD 06 15  SKPCLS      JSR     SKIPSP

                * CLASSIFY CHARACTER

08DC DF 44     CLASS       STX     BUFPNT  SAVE POINTER
08DE A6 00                 LDA A   0,X     GET CHARACTER
08E0 5F                    CLR B
08E1 81 2F                 CMP A   #$2F    CHECK IF NUMBER
08E3 23 18                 BLS     CLASS4
08E5 81 39                 CMP A   #'9
08E7 22 02                 BHI     CLASS2
08E9 5C                    INC B           SHOW NUMBER
08EA 39                    RTS             RETURN
08EB 81 40     CLASS2      CMP A   #$40    CHECK IF LETTER
08ED 23 0E                 BLS     CLASS4
```

```
08EF 81 5A              CMP A  #'Z
08F1 23 08              BLS    CLASS3
08F3 81 61              CMP A  #'a          CHECK LOWER CASE
08F5 25 06              BLO    CLASS4
08F7 81 7A              CMP A  #'z
08F9 22 02              BHI    CLASS4
08FB C6 02      CLASS3  LDA B  #2           SHOW LETTER
08FD 39         CLASS4  RTS                 RETURN


                * CONVERT ASCII TO BCD

08FE 8D 4C      BCDCON  BSR    CLRNUM       CLEAR NUMBER
0900 8D DA      BCDC01  BSR    CLASS        CLASSIFY CHAR.
0902 C1 01              CMP B  #1           IS IT A NUMBER?
0904 27 07              BEQ    BCDC02
0906 81 2E              CMP A  #'           IS IT A "."?
0908 27 17              BEQ    BCDC05
090A DF 44      BCDC15  STX    BUFPNT       SAVE POINTER
090C 39                 RTS                 RETURN
090D 08         BCDC02  INX                 BUMP THE POINTER
090E 84 0F              AND A  #$0F         MASK ASCII
0910 C6 04              LDA B  #4           SET COUNTER
0912 78 00 92   BCDC04  ASL    NUMBER+1
0915 79 00 91           ROL    NUMBER       SHIFT EVERYTHING LEFT
0918 5A                 DEC B               DEC THE COUNTER
0919 26 F7              BNE    BCDC04
091B 9B 92              ADD A  NUMBER+1     ADD IN NUMBER
091D 97 92              STA A  NUMBER+1
091F 20 DF              BRA    BCDC01
0921 C6 02      BCDC05  LDA B  #2           SET COUNTER
0923 D7 8A              STA B  DECCNT
0925 08         BCDC06  INX                 BUMP THE POINTER
0926 8D B4              BSR    CLASS        CLASSIFY CHAR.
0928 C1 01              CMP B  #1           IS IT NUMBER?
092A 27 04              BEQ    BCDC65
092C 4F                 CLR A
092D 09                 DEX                 DEC THE POINTER
092E 20 02              BRA    BCDC67
0930 84 0F      BCDC65  AND A  #$0F         MASK ASCII
0932 C6 04      BCDC67  LDA B  #4           SET COUNTER
0934 78 00 93   BCDC07  ASL    NUMBER+2
0937 5A                 DEC B
0938 26 FA              BNE    BCDC07
093A 9B 93              ADD A  NUMBER+2
093C 97 93              STA A  NUMBER+2
093E 7A 00 8A           DEC    DECCNT       DEC COUNTER
0941 26 E2              BNE    BCDC06
0943 08         BCDC08  INX                 BUMP THE POINTER
0944 8D 96              BSR    CLASS        CLASSIFY CHAR.
0946 C1 01              CMP B  #1           IS IT NUMBER?
0948 27 F9              BEQ    BCDC08
094A 20 BE              BRA    BCDC15
```

```
                    * CLEAR NUMBER ROUTINE

094C 4F          CLRNUM  CLR A               CLEAR ACC.
094D 97 91               STA A  NUMBER
094F 97 92               STA A  NUMBER+1     CLEAR ALL OUT
0951 97 93               STA A  NUMBER+2
0953 39                  RTS                 RETURN

                    * FIND NUMBERED LINE

0954 D6 91       FNDNUM  LDA B  NUMBER       GET DIGIT
0956 96 92               LDA A  NUMBER+1
0958 DE 98               LDX    FILBEG       SET POINTER TO BEGIN
095A 9C 9A       FNDNU1  CPX    FILEND       END OF FILE?
095C 26 05               BNE    FNDNU4
095E 7C 00 74            INC    OVREND       SET ERROR FLAG
0961 5C          FNDNU2  INC B
0962 39                  RTS                 RETURN
0963 E1 00       FNDNU4  CMP B  0,X          COMPARE DIGIT
0965 22 1C               BHI    FNDNU5
0967 26 F8               BNE    FNDNU2
0969 A1 01               CMP A  1,X          COMP NEXT DIGIT
096B 22 16               BHI    FNDNU5
096D 26 F2               BNE    FNDNU2
096F D6 93               LDA B  NUMBER+2     NEXT DIGIT
0971 E1 02               CMP B  2,X          CHECK DIGIT
0973 22 0E               BHI    FNDNU5
0975 26 EA               BNE    FNDNU2
0977 7D 00 85            TST    SNGLIN
097A 26 05               BNE    FNDN45
097C 7D 00 84            TST    CHKFLG
097F 26 E0               BNE    FNDNU2
0981 5F          FNDN45  CLR B
0982 39                  RTS                 RETURN
0983 7D 00 84    FNDNU5  TST    CHKFLG
0986 26 F9               BNE    FNDN45
0988 8D 05               BSR    FNDCRT       FIND C. R.
098A D6 91               LDA B  NUMBER       RESTORE NUM
098C 08                  INX                 BUMP THE POINTER
098D 20 CB               BRA    FNDNU1       REPEAT

                    * FIND THE NEXT CARRIAGE RETURN

098F 36          FNDCRT  PSH A               SAVE ACC.
0990 86 0D               LDA A  #CRGRET
0992 08          FNDCR2  INX                 BUMP THE POINTER
0993 A1 00               CMP A  0,X          CHECK FOR C. R.
0995 26 FB               BNE    FNDCR2
0997 32                  PUL A               RESTORE ACC.
0998 39                  RTS                 RETURN
```

```
                    * MOVE BACK ONE LINE

0999 9C 98      BAKONE  CPX     FILBEG
099B 27 17              BEQ     BAKON6
099D C6 01              LDA B   #1          SET COUNTER
099F 09         BAKON2  DEX                 DEC THE POINTER
09A0 9C 98              CPX     FILBEG      BEGINNING?
09A2 27 0D              BEQ     BAKON5
09A4 A6 00              LDA A   0,X         GET A CHAR.
09A6 81 0D              CMP A   #CRGRET     IS IT C.R.?
09A8 26 F5              BNE     BAKON2
09AA 5A                 DEC B               DEC THE COUNTER
09AB 2A F2              BPL     BAKON2
09AD 08                 INX                 BUMP THE POINTER
09AE C6 01              LDA B   #1
09B0 39         BAKON4  RTS                 RETURN
09B1 5D         BAKON5  TST B
09B2 27 FC              BEQ     BAKON4
09B4 7C 00 73   BAKON6  INC     OVRBEG      SET ERROR FLAG
09B7 39                 RTS                 RETURN

                    * MOVE UP ONE LINE

09B8 9C 9A      UPONE   CPX     FILEND      END OF FILE?
09BA 26 06              BNE     UPONE2
09BC C6 01      UPONE1  LDA B   #1          SET ERROR FLAG
09BE D7 74              STA B   OVREND
09C0 20 D7              BRA     BAKONE
09C2 8D CB      UPONE2  BSR     FNDCRT      FIND NEXT C.R.
09C4 08                 INX                 BUMP THE POINTER
09C5 9C 9A              CPX     FILEND      END?
09C7 27 F3              BEQ     UPONE1
09C9 39                 RTS                 RETURN

                    * TEST FOR OVER END LIMITS

09CA 7D 00 73   TSTOVR  TST     OVRBEG      BEGINNING?
09CD 26 03              BNE     TSTOV2
09CF 7D 00 74           TST     OVREND      END?
09D2 39         TSTOV2  RTS                 RETURN

                    * DECREMENT NUMBER BY ONE

09D3 86 99      DECNUM  LDA A   #$99
09D5 16                 TAB                 SET UP $9999
09D6 9B 92              ADD A   NUMBER+1    ADD IN
09D8 19                 DAA                 ADJUST IT
09D9 97 92              STA A   NUMBER+1    SAVE
09DB 17                 TBA
09DC 99 91              ADC A   NUMBER
```

```
09DE 19                  DAA
09DF 97 91               STA A  NUMBER

                 * TEST NUMBER FOR ZERO

09E1 96 91       TSTNUM  LDA A  NUMBER     CHECK IF ZERO
09E3 26 02               BNE    TSTNU2
09E5 96 92               LDA A  NUMBER+1
09E7 39          TSTNU2  RTS                RETURN

                 * VERIFY LINE ROUTINE

09E8 DF 4A       VERLIN  STX    NEWPOS      SAVE POINTER
09EA BD 09 8F            JSR    FNDCRT
09ED DF 5A               STX    SPCPT2      SAVE POSITION
09EF 4F                  CLR A
09F0 97 8E               STA A  CHRCNT
09F2 09          VERLI1  DEX                DEC POINTER
09F3 09                  DEX
09F4 09                  DEX
09F5 09          VERL12  DEX                DEC THE POINTER
09F6 E6 00               LDA B  0,X         CHECK CHAR
09F8 C1 0D               CMP B  #CRGRET     IS IT C.R.?
09FA 27 09               BEQ    VERL15
09FC E6 03               LDA B  3,X         CHECK
09FE C1 20               CMP B  #'          IS IT A SPACE?
0A00 26 03               BNE    VERL15
0A02 4C                  INC A
0A03 20 F0               BRA    VERL12
0A05 97 8F       VERL15  STA A  CHRCNT+1    SAVE COUNT
0A07 08                  INX
0A08 08                  INX
0A09 08                  INX                BUMP POINTER
0A0A 08                  INX
0A0B DF 58               STX    SPCPT1
0A0D BD 0F 17            JSR    DELCHR      DELETE SPACES
0A10 DE 4A               LDX    NEWPOS
0A12 96 6B               LDA A  VERFLG      CHECK FLAG
0A14 27 05               BEQ    VERLI2
0A16 8D 40               BSR    OUTLIN      OUTPUT LINE
0A18 BD 09 99            JSR    BAKONE      BACKUP ONE LINE
0A1B DF 48       VERLI2  STX    CURPOS      SAVE POINTER
0A1D DF 4A               STX    NEWPOS
0A1F 39                  RTS                RETURN

                 * PRINT ROUTINE

0A20 8D 2F       PRINT   BSR    TSTEMP
0A22 DE 44               LDX    BUFPNT      SET POINTER
0A24 BD 06 85            JSR    FINDT       FIND TARGET
0A27 DE 4A       PRINT0  LDX    NEWPOS      SET POINTER
```

```
0A29 7C 00 8B           INC     PRNFLG      SET FLAG
0A2C DF 48              STX     CURPOS      SAVE IT
0A2E 9C 94      PRINT1  CPX     TRGLIN      TARGET LINE?
0A30 26 03              BNE     PRIN12
0A32 7F 00 8B           CLR     PRNFLG      CLEAR FLAG
0A35 8D 21      PRIN12  BSR     OUTLIN
0A37 96 8B              LDA A   PRNFLG      CHECK FLAG
0A39 27 0B              BEQ     PRINT5
0A3B 96 8D              LDA A   DRCTN       CHECK DIRECTION
0A3D 27 EF              BEQ     PRINT1
0A3F 09                 DEX                 DEC POINTER TWICE
0A40 09                 DEX
0A41 BD 09 99           JSR     BAKONE      MOVE BACK ONE
0A44 20 E8              BRA     PRINT1
0A46 BD 09 99   PRINT5  JSR     BAKONE      MOVE BACK ONE
0A49 DF 48              STX     CURPOS      SAVE POINTER
0A4B BD 0B 80   PRINT6  JSR     TSTMSL
0A4E 7E 04 FA           JMP     EDIT        RETURN

                * TEST IF FILE EMPTY

0A51 DE 98      TSTEMP  LDX     FILBEG
0A53 9C 9A              CPX     FILEND
0A55 27 F4              BEQ     PRINT6
0A57 39                 RTS

                * OUTPUT ONE LINE

0A58 BD 05 DF   OUTLIN  JSR     PCRLF
0A5B 96 6A              LDA A   NUMFLG
0A5D 26 06              BNE     OUTL15
0A5F 8D 15              BSR     OUTSPC      OUTPUT SPACE
0A61 08                 INX                 BUMP THE POINTER
0A62 08                 INX
0A63 20 03              BRA     OUTLI2
0A65 8D 16      OUTL15  BSR     OUTBCD      OUTPUT LINE NO.
0A67 09                 DEX
0A68 08         OUTLI2  INX
0A69 A6 00              LDA A   0,X         GET A CHAR.
0A6B 81 0D              CMP A   #CRGRET     IS IT C.R.?
0A6D 27 05              BEQ     OUTLI4
0A6F BD 02 09           JSR     OUTCH       OUTPUT IT
0A72 20 F4              BRA     OUTLI2      REPEAT
0A74 08         OUTLI4  INX                 BUMP THE POINTER
0A75 39                 RTS                 RETURN

                * OUTPUT A SPACE

0A76 86 20      OUTSPC  LDA A   #' '         LOAD UP SPACE
0A78 BD 02 09           JSR     OUTCH       OUTPUT IT
0A7B 0C                 CLC
```

```
0A7C 39                    RTS                 RETURN

                    * OUTPUT A BCD NUMBER

0A7D 96 6A         OUTBCD  LDA A  NUMFLG       CHECK FLAG
0A7F 27 2E                 BEQ    OUTB75
0A81 8D F3                 BSR    OUTSPC      OUTPUT A SPACE
0A83 C6 02                 LDA B  #2          SET COUNTER
0A85 0C                    CLC
0A86 A6 00         OUTBC2  LDA A  0,X          GET CHAR.
0A88 85 F0                 BIT A  #$F0         MASK
0A8A 25 02                 BCS    OUTBC3
0A8C 27 06                 BEQ    OUTB35
0A8E BD 0A BC      OUTBC3  JSR    OUTHL       OUTPUT DIGIT
0A91 0D                    SEC                SET FLAG
0A92 20 02                 BRA    OUTBC4
0A94 8D E0         OUTB35  BSR    OUTSPC
0A96 A6 00         OUTBC4  LDA A  0,X          GET DIGIT
0A98 C5 FE                 BIT B  #$FE         CHECK IF DONE
0A9A 27 06                 BEQ    OUTBC6
0A9C 85 0F                 BIT A  #$0F         MASK
0A9E 25 02                 BCS    OUTBC6
0AA0 27 05                 BEQ    OUTB65
0AA2 8D 1C         OUTBC6  BSR    OUTHR       OUTPUT DIGIT
0AA4 0D                    SEC
0AA5 20 02                 BRA    OUTBC7
0AA7 8D CD         OUTB65  BSR    OUTSPC
0AA9 08            OUTBC7  INX                BUMP THE POINTER
0AAA 5A                    DEC B              DEC THE COUNTER
0AAB 27 07                 BEQ    OUTBC8
0AAD 2A D7                 BPL    OUTBC2
0AAF 86 3D         OUTB75  LDA A  #'=
0AB1 7E 02 09      OUTB78  JMP    OUTCH       OUTPUT A "="
0AB4 86 2E         OUTBC8  LDA A  #'.          OUTPUT A "."
0AB6 BD 02 09              JSR    OUTCH
0AB9 0D                    SEC
0ABA 20 CA                 BRA    OUTBC2      GO FINISH

                    * OUTPUT DIGITS ROUTINE

0ABC 44            OUTHL   LSR A               SHIFT LEFT FOUR TIMES
0ABD 44                    LSR A
0ABE 44                    LSR A
0ABF 44                    LSR A
0AC0 84 0F         OUTHR   AND A  #$0F         MASK
0AC2 8B 30                 ADD A  #$30         MAKE ASCII
0AC4 20 EB                 BRA    OUTB78

                    * SET NUMBERS ON OR OFF

0AC6 8D 18         NUMSET  BSR    ONOFF       GET ON OFF
```

```
0AC8 27 07              BEQ     NUMSE2
0ACA 2B 0A              BMI     NUMSE4
0ACC 7F 00 6A           CLR     NUMFLG      CLEAR FLAG
0ACF 20 08              BRA     NUMSE6
0AD1 43          NUMSE2 COM A               COM. FLAG
0AD2 97 6A              STA A   NUMFLG      SAVE IN FLAG
0AD4 20 03              BRA     NUMSE6
0AD6 73 00 6A    NUMSE4 COM     NUMFLG      COM FLAG
0AD9 DE 4A       NUMSE6 LDX     NEWPOS
0ADB DF 48              STX     CURPOS      SET POINTER
0ADD 7E 0A 4B           JMP     PRINT6

                 * CHECK FOR ON OR OFF

0AE0 DE 44       ONOFF  LDX     BUFPNT      SET POINTER
0AE2 BD 06 15           JSR     SKIPSP      SKIP SPACES
0AE5 DF 44              STX     BUFPNT      SAVE POINTER
0AE7 DF 40              STX     TEMP
0AE9 CE 0A EF           LDX     #ONOFTB     POINT TO TABLE
0AEC 7E 05 88           JMP     EDIT6

                 * TABLE FOR ON OFF

0AEF 4F          ONOFTB FCC     'ON'
0AF1 00                 FCB     0
0AF2 0A FF              FDB     ON
0AF4 4F                 FCC     'OFF'
0AF7 00                 FCB     0
0AF8 0B 01              FDB     OFF
0AFA 0D                 FCB     CRGRET
0AFB 00                 FCB     0
0AFC 0B 04              FDB     TOGGLE
0AFE 00                 FCB     0

                 * ON OFF ROUTINES

0AFF 4F          ON     CLR A
0B00 39                 RTS                 RETURN
0B01 86 01       OFF    LDA A   #1          SET FLAG
0B03 39                 RTS                 RETURN
0B04 DE 44       TOGGLE LDX     BUFPNT
0B06 09                 DEX
0B07 DF 44              STX     BUFPNT      FIX
0B09 86 FF              LDA A   #$FF        SET FLAG
0B0B 39                 RTS                 RETURN

                 * SET VERIFY FLAG

0B0C 8D D2       VERSET BSR     ONOFF       CHECK ON OFF
0B0E 27 07              BEQ     VERSE2
```

```
0B10 2B 0A              BMI     VERSE4
0B12 7F 00 6B           CLR     VERFLG      CLEAR FLAG
0B15 20 08              BRA     VERSE6
0B17 43         VERSE2  COM A
0B18 97 6B              STA A   VERFLG
0B1A 20 03              BRA     VERSE6
0B1C 73 00 6B   VERSE4  COM     VERFLG      COM. FLAG
0B1F 20 B8      VERSE6  BRA     NUMSE6


                * CURSOR CONTROL COMMAND "X"

0B21 8D 4B      XCNTRL  BSR     TFORCR
0B23 CE 0B 2B           LDX     #CNRSTR     POINT TO STRING
0B26 BD 06 08           JSR     PDATA1      OUTPUT IT
0B29 20 AE              BRA     NUMSE6

0B2B 00         CNRSTR  FCB     0,0,0,0,0,0
0B31 04                 FCB     4           THIS 4 MUST REMAIN !!


                * EXIT ROUTINE

0B32 8D 3A      EXIT    BSR     TFORCR
0B34 7C 00 20   DEXIT   INC     DSKFLG      SET FLAG
0B37 DE 98      EXIT2   LDX     FILBEG      SET POINTER
0B39 DF 58              STX     SPCPT1
0B3B DE 9A              LDX     FILEND      GET END
0B3D DF 5A              STX     SPCPT2
0B3F BD 15 C5           JSR     RECOR1      GO WRITE IT
0B42 7D 00 24           TST     LASTIN      FINISHED?
0B45 26 05              BNE     EXIT5
0B47 BD 18 07           JSR     RNEW        READ NEW
0B4A 20 E8              BRA     DEXIT       REPEAT
0B4C CE 19 DD   EXIT5   LDX     #WFCB       POINT TO FCB
0B4F BD 16 5E           JSR     CLSFC1      CLOSE FILE
0B52 BD B4 03           JSR     FMSCLS      CLOSE FMS
0B55 7E AD 03           JMP     WARMS


                * SET POINTER TO BOTTOM

0B58 8D 14      BOTTOM  BSR     TFORCR
0B5A BD 0A 51   BOTTO1  JSR     TSTEMP
0B5D DE 9A              LDX     FILEND
0B5F BD 09 99           JSR     BAKONE      MOVE BACK ONE
0B62 DF 48      BOTTO2  STX     CURPOS      SAVE-POINTER
0B64 7E 0A 4B           JMP     PRINT6


                * SET POINTER TO TOP

0B67 8D 05      TOP     BSR     TFORCR
```

```
0B69 BD 0A 51           JSR     TSTEMP
0B6C 20 F4              BRA     BOTTO2

                * TEST OR C.R.

0B6E DE 44      TFORCR  LDX     BUFPNT      SET POINTER
0B70 BD 06 15           JSR     SKIPSP
0B73 81 0D              CMP A   #CRGRET     IS IT C.R.?
0B75 27 05              BEQ     TFORC2
0B77 B1 AC 02           CMP A   EOL
0B7A 26 01              BNE     TFORC3
0B7C 39         TFORC2  RTS                 RETURN
0B7D 7E 05 F4   TFORC3  JMP     SYNERR

                * TEST FOR MULTIPLE STATEMENTS PER LINE

0B80 DE 44      TSTMSL  LDX     BUFPNT      GET POINTER
0B82 86 0D              LDA A   #CRGRET     GET C.R.
0B84 F6 AC 02           LDA B   EOL         GET EOL CHAR.
0B87 A1 00      TSTMS2  CMP A   0,X          CHECK CHARACTER
0B89 27 0C              BEQ     TSTMS5
0B8B E1 00              CMP B   0,X
0B8D 27 03              BEQ     TSTMS4
0B8F 08                 INX                 BUMP POINTER ONE
0B90 20 F5              BRA     TSTMS2      REPEAT
0B92 08         TSTMS4  INX
0B93 DF 44              STX     BUFPNT      SAVE BUFFER POINT
0B95 97 6D              STA A   MSLFLG      SET FLAG
0B97 39         TSTMS5  RTS                 RETURN

                * PROCCESS THE NEXT COMMAND

0B98 7C 00 77   NEXT    INC     NXTFLG

                * FIND COMMAND

0B9B BD 0A 51   CFIND   JSR     TSTEMP
0B9E 7C 00 76           INC     LINFLG      SET FLAG
0BA1 8D 71              BSR     OCCURR      CHECK FOR OCCURRENCE
0BA3 DE 94              LDX     TRGLIN      SET POINTER
0BA5 BD 09 CA           JSR     TSTOVR      CHECK LIMITS
0BA8 27 2A              BEQ     CFIND2
0BAA D6 77              LDA B   NXTFLG      CHECK FLAG
0BAC 26 43              BNE     CFIND5
0BAE 20 04              BRA     CFIN12
0BB0 96 78      CFIND1  LDA A   ALLFLG      CHECK IF ALL
0BB2 26 40              BNE     CFIND6
0BB4 CE 0B FD   CFIN12  LDX     #CFNTST     POINT TO STRING
0BB7 BD 06 06           JSR     PSTRNG      OUTPUT IT
0BBA DE 5E              LDX     ZONE1       CHECK ZONES
```

```
0BBC 8C 00 01           CPX     #$0001
0BBF 26 07              BNE     CFIN13
0BC1 DE 60              LDX     ZONE2       CHECK ZONE 2
0BC3 8C 01 36           CPX     #$0136
0BC6 27 06              BEQ     CFIN14
0BC8 CE 0C 07  CFIN13   LDX     #ZOKSTR     POINT TO STRING
0BCB BD 06 08           JSR     PDATA1      OUTPUT IT
0BCE 7F 00 6D  CFIN14   CLR     MSLFLG
0BD1 7E 04 FA  CFIN15   JMP     EDIT        RETURN
0BD4 DE 94     CFIND2   LDX     TRGLIN      POINT TO TARGET
0BD6 9C 4A              CPX     NEWPOS      SAME ONE?
0BD8 27 1A              BEQ     CFIND6
0BDA DF 4A              STX     NEWPOS      SAVE IT
0BDC D6 79     CFIND3   LDA B   OCRFLG      CHECK FLAG
0BDE 27 11              BEQ     CFIND5
0BE0 D6 77              LDA B   NXTFLG      CHECK FLAG
0BE2 26 03              BNE     CFIND4
0BE4 BD 09 E8           JSR     VERLIN      VERIFY LINE
0BE7 BD 0C 4F  CFIND4   JSR     NXTOCR      CHECK NEXT OCCUR.
0BEA BD 09 CA           JSR     TSTOVR      CHECK LIMITS
0BED 27 E5              BEQ     CFIND2
0BEF 20 06              BRA     CFIND9
0BF1 BD 09 E8  CFIND5   JSR     VERLIN      VERIFY LINE
0BF4 7E 0A 4B  CFIND6   JMP     PRINT6
0BF7 D6 77     CFIND9   LDA B   NXTFLG      CHECK FLAG
0BF9 26 F6              BNE     CFIND5
0BFB 20 B3              BRA     CFIND1


0BFD 4E        CFNTST   FCC     'NOT FOUND'
0C06 04                 FCB     4

0C07 2E        ZOKSTR   FCC     '...ZONES OK?'
0C13 04                 FCB     4


               * CHECK FOR OCCURRENCE

0C14 DE 44     OCCURR   LDX     BUFPNT      SET POINTER
0C16 DF 46              STX     BUFSAV      SAVE IT
0C18 7F 00 78           CLR     ALLFLG
0C1B 7F 00 79           CLR     OCRFLG
0C1E BD 06 85           JSR     FINDT       FIND TARGET
0C21 DE 44              LDX     BUFPNT      RESTORE POINTER
0C23 BD 08 D9           JSR     SKPCLS
0C26 C1 01              CMP B   #1          IS IT NUMBER?
0C28 27 09              BEQ     OCCUR3
0C2A 81 2A              CMP A   #'*         IS IT A "*"?
0C2C 26 17              BNE     OCCUR5
0C2E 7C 00 78           INC     ALLFLG      SET FOR ALL OCCUR.
0C31 20 0F              BRA     OCCUR4
0C33 BD 08 FE  OCCUR3   JSR     BCDCON      GET NUMBER
0C36 BD 09 E1           JSR     TSTNUM      ZERO?
0C39 27 0A              BEQ     OCCUR5
```

```
0C3B BD 09 D3          JSR    DECNUM    DEC NUMBER
0C3E 27 05             BEQ    OCCUR5
0C40 8D 04             BSR    SAVOCR    SAVE OCCURRENCE
0C42 7C 00 79 OCCUR4   INC    OCRFLG    SET FLAG
0C45 39       OCCUR5   RTS              RETURN

                * SAVE PRESENT OCCURRENCE COUNT

0C46 96 91    SAVOCR   LDA A  NUMBER    GET NUMBER
0C48 97 6F             STA A  OCRCNT    SAVE IT
0C4A 96 92             LDA A  NUMBER+1
0C4C 97 70             STA A  OCRCNT+1
0C4E 39                RTS              RETURN

                * PROCCESS NEXT OCCURRENCE

0C4F 96 78    NXTOCR   LDA A  ALLFLG    CHECK FOR ALL
0C51 26 0F             BNE    NXTOC1
0C53 96 6F    NXTOC0   LDA A  OCRCNT    GET COUNT
0C55 97 91             STA A  NUMBER    PUT IN NUMBER
0C57 96 70             LDA A  OCRCNT+1
0C59 97 92             STA A  NUMBER+1
0C5B BD 09 D3          JSR    DECNUM    DEC THE COUNT
0C5E 27 0B             BEQ    NXTOC2
0C60 8D E4             BSR    SAVOCR    SAVE COUNT
0C62 96 86    NXTOC1   LDA A  CHGFLG
0C64 26 0A             BNE    NXTOC3
0C66 DE 46             LDX    BUFSAV    RESTORE POINTER
0C68 7E 06 85          JMP    FINDT     FIND TARGET AND RET
0C6B 7F 00 79 NXTOC2   CLR    OCRFLG    CLEAR FLAG
0C6E 20 F2             BRA    NXTOC1
0C70 39       NXTOC3   RTS              RETURN

                * EQUALS COMMAND

0C71 BD 0A 51 EQUALS   JSR    TSTEMP
0C74 DE 44             LDX    BUFPNT    SET POINTER
0C76 7F 00 7E          CLR    BMPFLG
0C79 7C 00 85          INC    SNGLIN
0C7C 7C 00 7F          INC    EQUFLG
0C7F 96 75             LDA A  NOCURL    CURRENT LINE?
0C81 26 20             BNE    INSER1
0C83 DE 4A             LDX    NEWPOS
0C85 DF 94             STX    TRGLIN    FIX TARGET
0C87 BD 08 35          JSR    GETNUM    GET NUMBER
0C8A 7C 00 82          INC    REPFLG    SET FLAG
0C8D 96 8F             LDA A  CHRCNT+1
0C8F 97 83             STA A  TMPCHR
0C91 7E 0E 8C          JMP    DELET0
```

```
                    * INSERT ROUTINE

0C94 DE 44      INSERT  LDX     BUFPNT      SET POINTER
0C96 7F 00 7E           CLR     BMPFLG      CLEAR FLAG
0C99 A6 00              LDA A   0,X         GET CHAR.
0C9B 81 0D              CMP A   #CRGRET
0C9D 27 17              BEQ     INSER4
0C9F 7C 00 85           INC     SNGLIN      SET FLAG
0CA2 08                 INX                 BUMP THE POINTER
0CA3 DF 44      INSER1  STX     BUFPNT      SAVE IT
0CA5 CE 00 BD           LDX     #BUFFER
0CA8 96 8F              LDA A   CHRCNT+1    GET COUNT
0CAA 9C 44      INSER2  CPX     BUFPNT      CHECK POINT
0CAC 27 04              BEQ     INSER3
0CAE 4A                 DEC A               DEC THE COUNTER
0CAF 08                 INX                 BUMP THE POINTER
0CB0 20 F8              BRA     INSER2
0CB2 8B 03      INSER3  ADD A   #3          FIX COUNT
0CB4 97 8F              STA A   CHRCNT+1
0CB6 7F 00 6C   INSER4  CLR     REPEAT      DISABLE COMMAND REPEAT
0CB9 DE 4A              LDX     NEWPOS      SET POINTER
0CBB DF 48              STX     CURPOS      SAVE POINTER
0CBD 96 7F              LDA A   EQUFLG
0CBF 27 06              BEQ     INSE42
0CC1 96 73              LDA A   OVRBEG      CHECK LIMIT
0CC3 27 14              BEQ     INSE43
0CC5 20 25              BRA     INSER5
0CC7 BD 08 35   INSE42  JSR     GETNUM
0CCA 96 80              LDA A   INLMFL      CHECK FLAG
0CCC 27 0B              BEQ     INSE43
0CCE 9C 9A              CPX     FILEND      EMPTY?
0CD0 27 07              BEQ     INSE43
0CD2 5F                 CLR B               CLEAR ACC.
0CD3 D7 91              STA B   NUMBER      SAVE IN NUMBER
0CD5 D7 92              STA B   NUMBER+1
0CD7 20 13              BRA     INSER5
0CD9 BD 09 B8   INSE43  JSR     UPONE       UP ONE LINE
0CDC E6 02              LDA B   2,X         GET DIGIT
0CDE 96 74              LDA A   OVREND      LIMIT?
0CE0 27 0A              BEQ     INSER5
0CE2 5F         INSE45  CLR B
0CE3 DE 9A              LDX     FILEND
0CE5 9C 98              CPX     FILBEG
0CE7 26 03              BNE     INSER5
0CE9 BD 09 4C           JSR     CLRNUM      CLEAR OUT NUMBER

                    * CALCULATE LINE NUMBER INCREMENT

0CEC DF 58      INSER5  STX     SPCPT1      SAVE POINTER
0CEE 96 7F              LDA A   EQUFLG
0CF0 26 2C              BNE     INSE60
0CF2 96 74              LDA A   OVREND      LIMIT?
0CF4 27 05              BEQ     INSE51
0CF6 7F 00 93           CLR     NUMBER+2
```

```
0CF9 20 1C              BRA     INSER6
0CFB 96 7E      INSE51  LDA A   BMPFLG
0CFD 26 18              BNE     INSER6
0CFF 96 93              LDA A   NUMBER+2  GET NUMBER
0D01 D7 83              STA B   TMPCHR
0D03 9A 83              ORA A   TMPCHR
0D05 27 0D              BEQ     INSE55
0D07 96 80              LDA A   INLMFL    CHECK FLAG
0D09 27 03              BEQ     INSE52
0D0B 7F 00 93           CLR     NUMBER+2
0D0E 86 01      INSE52  LDA A   #1
0D10 97 7D              STA A   INCAMT    SET AMOUNT
0D12 20 03              BRA     INSER6
0D14 7A 00 7D   INSE55  DEC     INCAMT
0D17 BD 08 07   INSER6  JSR     BMPNUM    BUMP NUMBER
0D1A 96 85              LDA A   SNGLIN    CHECK IF SINGLE IN
0D1C 27 04              BEQ     INSE61

                * ENTER BUFFERED INPUT MODE

0D1E DE 44      INSE60  LDX     BUFPNT
0D20 20 2D              BRA     INSE71
0D22 7F 00 80   INSE61  CLR     INLMFL
0D25 BD 05 DF           JSR     PCRLF
0D28 CE 00 91           LDX     #NUMBER   POINT TO NUMBER
0D2B BD 0A 7D           JSR     OUTBCD    OUTPUT IT
0D2E 7F 00 8E           CLR     CHRCNT
0D31 86 03              LDA A   #3        SET COUNTER
0D33 97 8F              STA A   CHRCNT+1
0D35 97 7E              STA A   BMPFLG    SET FLAG
0D37 CE 00 BD           LDX     #BUFFER   SET POINTER
0D3A BD 06 1C   INSE62  JSR     INCHAR    GET A CHARACTER
0D3D 27 E3              BEQ     INSE61
0D3F 81 0D              CMP A   #CRGRET   IS IT C.R.?
0D41 27 07              BEQ     INSER7
0D43 A7 00              STA A   0,X
0D45 BD 06 56           JSR     BUFLIM    CHECK LIMIT
0D48 20 F0              BRA     INSE62    REPEAT
0D4A A7 00      INSER7  STA A   0,X
0D4C CE 00 BD           LDX     #BUFFER   SET POINTER
0D4F DF 44      INSE71  STX     BUFPNT    SAVE IT
0D51 A6 00              LDA A   0,X       GET CHAR.
0D53 91 B4              CMP A   LINO      ESCAPE?
0D55 26 40              BNE     INSE72
0D57 96 8F              LDA A   CHRCNT+1
0D59 80 03              SUB A   #3        FIX COUNT
0D5B 97 8F              STA A   CHRCNT+1
0D5D 08                 INX               BUMP THE POINTER

                * CHECK IF RENUMBERING NECESSARY

0D5E DF 40      INS710  STX     TEMP      SAVE POINTER
0D60 DE 4A              LDX     NEWPOS
0D62 BD 09 B8           JSR     UPONE     UP ONE LINE
```

```
0D65 7D 00 74          TST    OVREND    LIMIT?
0D68 26 1A             BNE    INS711
0D6A 7C 00 84          INC    CHKFLG    SET FLAG
0D6D D6 91             LDA B  NUMBER    GET NUMBER
0D6F 96 92             LDA A  NUMBER+1
0D71 BD 09 63          JSR    FNDNU4    CHECK NUMBER
0D74 26 0E             BNE    INS711
0D76 4F                CLR A
0D77 97 7D             STA A  INCAMT    SET INC AMOUNT
0D79 97 93             STA A  NUMBER+2
0D7B BD 08 4D          JSR    RENUM2    RENUMBER FILE
0D7E CE 0D C1          LDX    #RENSTR   POINT TO STRING
0D81 BD 06 06          JSR    PSTRNG    OUTPUT IT
0D84 DE 40     INS711  LDX    TEMP      RESTORE POINTER
0D86 7D 00 85          TST    SNGLIN
0D89 27 06             BEQ    INS712
0D8B DE 4A             LDX    NEWPOS    FIX POINTER
0D8D DF 48             STX    CURPOS
0D8F 20 03             BRA    INS713
0D91 7C 00 6D  INS712  INC    MSLFLG    SET FLAG
0D94 7E 04 FA  INS713  JMP    EDIT

               * ACTUAL LINE INSERT

0D97 8D 3E     INSE72  BSR    MAKSPC    MAKE SOME SPACE
0D99 DE 40             LDX    TEMP      RESTORE POINTER
0D9B DF 4A             STX    NEWPOS
0D9D BD 08 28          JSR    PUTNUM    PUT NUMBER
0DA0 08                INX              BUMP 3 TIMES
0DA1 08                INX
0DA2 08                INX
0DA3 DF 40             STX    TEMP      SAVE POINTER
0DA5 DE 44     INSE75  LDX    BUFPNT
0DA7 A6 00             LDA A  0,X       GET CHAR.
0DA9 08                INX              BUMP THE POINTER
0DAA DF 44             STX    BUFPNT    SAVE IT
0DAC DE 40             LDX    TEMP
0DAE A7 00             STA A  0,X       PUT CHAR.
0DB0 08                INX              BUMP
0DB1 DF 40             STX    TEMP      SAVE
0DB3 81 0D             CMP A  #CRGRET
0DB5 26 EE             BNE    INSE75    REPEAT
0DB7 BD 14 67          JSR    EXPLIN    EXPAND TABS
0DBA 96 85             LDA A  SNGLIN
0DBC 26 A0             BNE    INS710
0DBE 7E 0C B6          JMP    INSER4


0DC1 53        RENSTR  FCC    'SOME LINES RENUMBERED'
0DD6 04                FCB    4

               * MAKE ROOM FOR INSERT
```

```
0DD7 7F 00 8A  MAKSPC  CLR     DECCNT     CLEAR COUNT
0DDA DE 58             LDX     SPCPT1     SET POINTER
0DDC DF 40             STX     TEMP       SAVE
0DDE 9C 9A             CPX     FILEND     END OF FILE?
0DE0 26 03             BNE     MAKSP1
0DE2 7C 00 8A          INC     DECCNT
0DE5 DE 9A     MAKSP1  LDX     FILEND     SET POINTER
0DE7 DF 58             STX     SPCPT1     SAVE
0DE9 D6 8E             LDA B   CHRCNT
0DEB 96 8F             LDA A   CHRCNT+1
0DED 26 03             BNE     MAKS21
0DEF 5D        MAKS18  TST B
0DF0 27 36     MAKSP2  BEQ     MAKSP4
0DF2 9C BB     MAKS21  CPX     MEMEND     END OF MEMORY?
0DF4 27 26             BEQ     MAKSP3
0DF6 08                INX                BUMP THE POINTER
0DF7 7D 00 8D          TST     DRCTN      WHICH DIRECTION?
0DFA 26 04             BNE     MAKS22
0DFC DF 42             STX     XSAVE      SAVE POINTER
0DFE 20 0C             BRA     MAK222
0E00 7D 00 8C  MAKS22  TST     CPYDRC
0E03 27 0E             BEQ     MAKS23
0E05 DF 42             STX     XSAVE      SAVE THE POINTER
0E07 DE 4A             LDX     NEWPOS     GET POSITION
0E09 08                INX                BUMP IT
0E0A DF 4A             STX     NEWPOS     SAVE IT
0E0C DE 94     MAK222  LDX     TRGLIN     GET TARGET
0E0E 08                INX                BUMP IT
0E0F DF 94             STX     TRGLIN
0E11 DE 42             LDX     XSAVE      RESTORE POINTER
0E13 4D        MAKS23  TST A              TEST THE ACC.
0E14 26 01             BNE     MAKS24
0E16 5A                DEC B              DEC THE COUNTER
0E17 4A        MAKS24  DEC A
0E18 26 D8             BNE     MAKS21
0E1A 20 D3             BRA     MAKS18     REPEAT
0E1C CE 0E 45  MAKSP3  LDX     #NORMST    POINT TO STRING
0E1F BD 06 06          JSR     PSTRNG     OUTPUT IT
0E22 7F 00 6D          CLR     MSLFLG
0E25 7E 04 FA          JMP     EDIT       RETURN
0E28 DF 9A     MAKSP4  STX     FILEND     SAVE POINTER
0E2A DF 5A             STX     SPCPT2     SAVE POINTER
0E2C 96 8A             LDA A   DECCNT     CHECK
0E2E 26 14             BNE     MAKSP6
0E30 DF 5A     MAKSP5  STX     SPCPT2     SAVE POINTER
0E32 DE 58     MAKS55  LDX     SPCPT1
0E34 9C 40             CPX     TEMP       DONE?
0E36 27 0C             BEQ     MAKSP6
0E38 09                DEX                DEC THE POINTER
0E39 A6 00             LDA A   0,X        GET CHAR.
0E3B DF 58             STX     SPCPT1     SAVE POINTER
0E3D DE 5A             LDX     SPCPT2
0E3F 09                DEX                DEC THE POINTER
0E40 A7 00             STA A   0,X        PUT THE CHAR.
```

```
0E42 20 EC              BRA     MAKSP5      REPEAT
0E44 39         MAKSP6  RTS                 RETURN

0E45 4E         NORMST  FCC     'NOT ENOUGH ROOM'
0E54 04                 FCB     4

                * REPLACE LINES ROUTINE

0E55 7C 00 82   REPLAC  INC     REPFLG      SET FLAG

                * DELETE LINES ROUTINE

0E58 DE 44      DELETE  LDX     BUFPNT      SET POINTER
0E5A BD 06 85           JSR     FINDT       FIND TARGET
0E5D BD 09 CA           JSR     TSTOVR      LIMITS?
0E60 27 2A              BEQ     DELET0
0E62 CE 0F 53   DELE02  LDX     #NTRCHS     POINT TO STRING
0E65 BD 06 06           JSR     PSTRNG      OUTPUT IT
0E68 7F 00 6C           CLR     REPEAT      DISABLE COMMAND REPEAT
0E6B CE 00 BD           LDX     #BUFFER     POINT TO BUFFER
0E6E BD 06 1C   DELE04  JSR     INCHAR      GET A CHARACTER
0E71 27 EF              BEQ     DELE02
0E73 A7 00              STA A   0,X         SAVE IT
0E75 08                 INX
0E76 81 0D              CMP A   #CRGRET     C.R.?
0E78 26 F4              BNE     DELE04      REPEAT
0E7A CE 00 BD           LDX     #BUFFER
0E7D BD 06 15           JSR     SKIPSP      SKIP SPACES
0E80 81 59              CMP A   #'Y         WAS IT 'Y'ES?
0E82 27 08              BEQ     DELET0
0E84 CE 0F 77           LDX     #NLDSTR     POINT TO STRING
0E87 BD 06 06           JSR     PSTRNG
0E8A 20 63              BRA     DELET5      RETURN
0E8C DE 4A      DELET0  LDX     NEWPOS      SET POINTER
0E8E 96 8D              LDA A   DRCTN       CHECK DIRECTION
0E90 27 15              BEQ     DELET1
0E92 BD 09 B8           JSR     UPONE       MOVE UP ONE
0E95 96 74              LDA A   OVREND      LIMIT?
0E97 27 02              BEQ     DELE15
0E99 DE 9A              LDX     FILEND
0E9B DF 5A      DELE15  STX     SPCPT2
0E9D DE 94              LDX     TRGLIN      GET TARGET
0E9F DF 48              STX     CURPOS      MAKE CURRENT
0EA1 DF 58              STX     SPCPT1
0EA3 DE 5A              LDX     SPCPT2      GET POINTER
0EA5 20 11              BRA     DELE25
0EA7 DF 58      DELET1  STX     SPCPT1      SAVE
0EA9 DF 48              STX     CURPOS
0EAB DE 94              LDX     TRGLIN      POINT TO TARGET
0EAD BD 09 B8           JSR     UPONE       MOVE UP ONE
0EB0 96 74              LDA A   OVREND      LIMIT?
```

```
0EB2 27 02              BEQ     DELET2
0EB4 DE 9A              LDX     FILEND      POINT TO END
0EB6 DF 5A      DELET2  STX     SPCPT2      SAVE POINTER
0EB8 4F         DELE25  CLR A
0EB9 5F                 CLR B
0EBA 9C 58      DELET3  CPX     SPCPT1
0EBC 27 07              BEQ     DELET4
0EBE 4C                 INC A
0EBF 26 01              BNE     DELE35
0EC1 5C                 INC B               BUMP THE COUNTER
0EC2 09         DELE35  DEX
0EC3 20 F5              BRA     DELET3
0EC5 97 8F      DELET4  STA A   CHRCNT+1     SAVE COUNT
0EC7 D7 8E              STA B   CHRCNT
0EC9 8D 4C              BSR     DELCHR      DELETE CHARACTERS
0ECB 96 82              LDA A   REPFLG      REPLACE?
0ECD 27 20              BEQ     DELET5
0ECF DE 48              LDX     CURPOS      SET POINTER
0ED1 BD 09 99           JSR     BAKONE      BACKUP ONE LINE
0ED4 96 7F              LDA A   EQUFLG
0ED6 26 2F              BNE     DELET7
0ED8 96 73              LDA A   OVRBEG      CHECK LIMIT
0EDA 27 0B              BEQ     DELE45
0EDC 7F 00 6C           CLR     REPEAT      DISABLE COMMAND REPEAT
0EDF BD 09 4C           JSR     CLRNUM      CLEAR NUMBER
0EE2 DF 4A              STX     NEWPOS      SAVE NEW POSITION
0EE4 7E 0C EC           JMP     INSER5
0EE7 DF 4A      DELE45  STX     NEWPOS      SAVE
0EE9 7F 00 7E           CLR     BMPFLG
0EEC 7E 0C B6           JMP     INSER4      GO TO INSERT
0EEF DE 48      DELET5  LDX     CURPOS      CHECK POSITION
0EF1 9C 9A              CPX     FILEND      END?
0EF3 26 0D              BNE     DELET6
0EF5 BD 09 99           JSR     BAKONE      MOVE IT BACK
0EF8 DF 40              STX     TEMP        SAVE POINTER
0EFA CE 0F 88           LDX     #BFRSTR     POINT TO STRING
0EFD BD 06 06           JSR     PSTRNG      OUTPUT IT
0F00 DE 40              LDX     TEMP        RESTORE
0F02 DF 48      DELET6  STX     CURPOS
0F04 7E 0A 4B           JMP     PRINT6
0F07 DF 4A      DELET7  STX     NEWPOS      SAVE POINTER
0F09 96 83              LDA A   TMPCHR      GET CHAR COUNT
0F0B 97 8F              STA A   CHRCNT+1
0F0D 4F                 CLR A
0F0E 97 7E              STA A   BMPFLG      CLEAR FLAG
0F10 97 8E              STA A   CHRCNT
0F12 DE 44              LDX     BUFPNT      SET POINTER
0F14 7E 0C A3           JMP     INSER1      GO INSERT IT


                * DELETE CHARACTER BLOCK

0F17 DE 5A      DELCHR  LDX     SPCPT2      SET POINTER
0F19 9C 58              CPX     SPCPT1      EQUAL?
```

```
0F1B 27 35              BEQ     DELCH5
0F1D 9C 9A              CPX     FILEND      END OF FILE?
0F1F 27 0E              BEQ     DELCH2
0F21 A6 00              LDA A   0,X         GET A CHAR.
0F23 08                 INX                 BUMP THE POINTER
0F24 DF 5A              STX     SPCPT2      SAVE
0F26 DE 58              LDX     SPCPT1
0F28 A7 00              STA A   0,X         PUT CHAR.
0F2A 08                 INX
0F2B DF 58              STX     SPCPT1      SAVE POINTER
0F2D 20 E8              BRA     DELCHR      REPEAT
0F2F D6 8E      DELCH2  LDA B   CHRCNT      GET COUNT
0F31 96 8F              LDA A   CHRCNT+1
0F33 26 03              BNE     DELC31
0F35 5D         DELC21  TST B               CHECK COUNT
0F36 27 18      DELCH3  BEQ     DELCH4
0F38 09         DELC31  DEX                 DEC THE POINTER
0F39 7D 00 8D           TST     DRCTN       WHICH DIRECTION?
0F3C 26 09              BNE     DELC32
0F3E DF 42              STX     XSAVE
0F40 DE 94              LDX     TRGLIN      GET TARGET
0F42 09                 DEX                 DEC IT
0F43 DF 94              STX     TRGLIN      PUT IT BACK
0F45 DE 42              LDX     XSAVE       RESTORE POINTER
0F47 4D         DELC32  TST A               TEST COUNT
0F48 26 01              BNE     DELC34
0F4A 5A                 DEC B               DEC THE COUNTER
0F4B 4A         DELC34  DEC A
0F4C 26 EA              BNE     DELC31
0F4E 20 E5              BRA     DELC21
0F50 DF 9A      DELCH4  STX     FILEND      SET NEW END
0F52 39         DELCH5  RTS                 RETURN


0F53 54         NTRCHS  FCC     'TARGET NOT REACHED!'
0F66 0D                 FCB     $D,$A,0,0,0,0
0F6C 59                 FCC     'YOU SURE? '
0F76 04                 FCB     4

0F77 4E         NLDSTR  FCC     'NO LINES DELETED'
0F87 04                 FCB     4

0F88 42         BFRSTR  FCC     'BOTTOM OF FILE REACHED'
0F9E 04                 FCB     4


                * CHANGE COMMAND ROUTINE

0F9F BD 0A 51   CHANGE  JSR     TSTEMP
0FA2 DE 44              LDX     BUFPNT      POINT TO BUFFER
0FA4 BD 08 D9           JSR     SKPCLS
0FA7 BD 07 FF           JSR     TSTEND
0FAA 27 03              BEQ     CHAN12      ERROR
0FAC 5D         CHANG1  TST B
```

```
0FAD 27 03              BEQ     CHAN15      ERROR
0FAF 7E 10 D8  CHAN12   JMP     CHANG9
0FB2 7C 00 86  CHAN15   INC     CHGFLG      SET FLAG
0FB5 BD 07 96           JSR     SETDEL      SET DELIMITERS
0FB8 5F                 CLR B               CLEAR COUNT
0FB9 A6 00     CHANG2   LDA A   0,X         GET CHAR.
0FBB BD 07 FF           JSR     TSTEND
0FBE 27 08              BEQ     CHANG3
0FC0 91 96              CMP A   DELIM       IS IT DELIMITER?
0FC2 27 04              BEQ     CHANG3
0FC4 08                 INX                 BUMP THE POINTER
0FC5 5C                 INC B               BUMP THE COUNT
0FC6 20 F1              BRA     CHANG2
0FC8 DF 66     CHANG3   STX     CHGEND      SAVE POINTER
0FCA D7 87              STA B   STRCN2      SAVE COUNT
0FCC BD 07 FF           JSR     TSTEND
0FCF 27 01              BEQ     CHAN35
0FD1 08                 INX                 BUMP POINTER
0FD2 DF 44     CHAN35   STX     BUFPNT      SAVE IT
0FD4 DE 4A              LDX     NEWPOS
0FD6 08        CHAN37   INX                 BUMP THREE TIMES
0FD7 08                 INX
0FD8 08                 INX
0FD9 DF 64              STX     CHGPNT      SAVE POINTER
0FDB BD 10 DB           JSR     SVSTPT      SAVE STRING POINT
0FDE 7F 00 86           CLR     CHGFLG
0FE1 BD 0C 14           JSR     OCCURR      GET TARG & OCCUR.
0FE4 7C 00 86           INC     CHGFLG
0FE7 BD 10 E8           JSR     RSTSPT      RESTORE STRING
0FEA 7F 00 6E           CLR     PSTZFL
0FED DE 6F              LDX     OCRCNT      GET COUNT
0FEF DF 68              STX     OCRTMP      SAVE
0FF1 96 79              LDA A   OCRFLG
0FF3 27 07              BEQ     CHANG4
0FF5 96 7C              LDA A   STRCNT      CHECK COUNT
0FF7 26 03              BNE     CHANG4
0FF9 7E 10 D8           JMP     CHANG9
0FFC DE 4A     CHANG4   LDX     NEWPOS
0FFE 9C 94              CPX     TRGLIN      AT TARGET?
1000 26 03              BNE     CHANG5
1002 7C 00 89           INC     LSTFLG      SET FLAG IF SO
1005 BD 07 D7  CHANG5   JSR     ZONE3       SET ZONE
1008 7F 00 7A           CLR     CHGONF      CLEAR FLAG
100B DE 64              LDX     CHGPNT
100D BD 07 F6           JSR     FIXZON      SET ZONE
1010 BD 07 54           JSR     FIND72
1013 20 10              BRA     CHA510

                * LOOP THROUGH OCCURRENCES

1015 7F 00 7A  CHAN50   CLR     CHGONF
1018 DE 64     CHAN51   LDX     CHGPNT      SET POINTER
101A BD 07 F6           JSR     FIXZON
101D BD 07 EB           JSR     CMPZN2      CHECK ZONE
```

```
1020 22 7A              BHI     CHANG8
1022 BD 07 7B           JSR     FIND75
1025 96 91      CHA510  LDA A   NUMBER      GET NUMBER
1027 97 62              STA A   ZONBUF      PUT IN BUFFER
1029 96 92              LDA A   NUMBER+1
102B 97 63              STA A   ZONBUF+1
102D 5D                 TST B
102E 26 6C              BNE     CHANG8
1030 5C                 INC B               BUMP COUNTER
1031 D7 88              STA B   FNONFL      SET FLAG
1033 96 79      CHAN52  LDA A   OCRFLG      ANY OCCURR. ?
1035 27 0E              BEQ     CHANG6
1037 96 78              LDA A   ALLFLG      CHANGE ALL?
1039 26 0A              BNE     CHANG6
103B BD 0C 53           JSR     NXTOC0
103E DE 5C      CHAN55  LDX     LASTNO      SET POINTER
1040 08                 INX                 BUMP IT
1041 DF 64              STX     CHGPNT
1043 20 D0              BRA     CHAN50

                * DELETE STRING ONE

1045 7C 00 7A   CHANG6  INC     CHGONF      SET FLAG
1048 DE 68              LDX     OCRTMP      CHECK COUNT
104A 27 04              BEQ     CHAN61
104C 86 01              LDA A   #1
104E 97 79              STA A   OCRFLG
1050 DF 6F      CHAN61  STX     OCRCNT      FIX COUNT
1052 DE 5C              LDX     LASTNO      GET STR. LOCATION
1054 DF 58              STX     SPCPT1
1056 D6 7C              LDA B   STRCNT
1058 27 10              BEQ     CHAN66
105A 7F 00 8E           CLR     CHRCNT      CLEAR COUNT
105D D7 8F              STA B   CHRCNT+1
105F 27 04      CHAN62  BEQ     CHAN65
1061 08                 INX                 BUMP POINTER
1062 5A                 DEC B               DEC THE COUNTER
1063 20 FA              BRA     CHAN62      REPEAT
1065 DF 5A      CHAN65  STX     SPCPT2      SAVE POINTER
1067 BD 0F 17           JSR     DELCHR      GO DELETE

                * INSERT STRING TWO

106A D6 87      CHAN66  LDA B   STRCN2      GET COUNT
106C 27 23              BEQ     CHA675
106E 7F 00 8E           CLR     CHRCNT      CLEAR OUT COUNT
1071 D7 8F              STA B   CHRCNT+1    SET COUNTER
1073 DE 5C              LDX     LASTNO      SET POINTER
1075 DF 58              STX     SPCPT1
1077 BD 0D D7           JSR     MAKSPC      GO MAKE ROOM
107A D6 8F              LDA B   CHRCNT+1
107C DE 50              LDX     STRNGE      POINT TO STRING END
107E 08         CHAN67  INX                 BUMP IT
107F A6 00              LDA A   0,X         GET CHAR
```

```
1081 DF 40              STX     TEMP
1083 DE 5C              LDX     LASTNO
1085 A7 00              STA A   0,X         PUT CHARACTER
1087 08                 INX                 BUMP POINTER
1088 DF 5C              STX     LASTNO      SAVE
108A DF 64              STX     CHGPNT
108C DE 40              LDX     TEMP        RESTORE
108E 5A                 DEC B               DEC THE COUNTER
108F 26 ED              BNE     CHAN67
1091 96 78      CHA675  LDA A   ALLFLG      DO ALL?
1093 27 1C              BEQ     CHAN81
1095 DE 5C              LDX     LASTNO
1097 DF 64              STX     CHGPNT      SAVE POINTER
1099 7E 10 18           JMP     CHAN51      REPEAT

                * CHANGE CLEANUP AND FINISH

109C 7F 00 79   CHANG8  CLR     OCRFLG      CLEAR FLAG
109F DE 68              LDX     OCRTMP      GET COUNT
10A1 27 04              BEQ     CHAN80
10A3 86 01              LDA A   #1          SET FLAG
10A5 97 79              STA A   OCRFLG
10A7 DF 6F      CHAN80  STX     OCRCNT      SET OCCUR. COUNT
10A9 96 88              LDA A   FNONFL      CHECK FLAG
10AB 26 04              BNE     CHAN81
10AD 96 89              LDA A   LSTFLG
10AF 26 27              BNE     CHANG9
10B1 DE 4A      CHAN81  LDX     NEWPOS
10B3 96 7A              LDA A   CHGONF
10B5 27 03              BEQ     CHAN82
10B7 BD 09 E8           JSR     VERLIN      VERIFY CHANGE
10BA 96 89      CHAN82  LDA A   LSTFLG
10BC 27 06              BEQ     CHAN84
10BE 7F 00 7A           CLR     CHGONF      CLEAR FLAG
10C1 7E 0A 4B           JMP     PRINT6
10C4 BD 08 60   CHAN84  JSR     NXTLIN      FIND NEXT LINE
10C7 DF 4A              STX     NEWPOS      SAVE POINTER
10C9 9C 94              CPX     TRGLIN      TARGET LINE?
10CB 26 03              BNE     CHAN86
10CD 7C 00 89           INC     LSTFLG      SET LAST FLAG
10D0 08         CHAN86  INX                 BUMP 3 TIMES
10D1 08                 INX
10D2 08                 INX
10D3 DF 64              STX     CHGPNT      SAVE POINTER
10D5 7E 10 05           JMP     CHANG5      REPEAT
10D8 7E 05 CF   CHANG9  JMP     ERROR       REPORT ERROR

                * SAVE STRING POINTER INFO

10DB DE 4E      SVSTPT  LDX     STRNGB      GET POINTER
10DD DF 52              STX     STRGB1      SAVE IT
10DF DE 50              LDX     STRNGE
10E1 DF 54              STX     STRGE1
```

```
10E3 96 7C              LDA A  STRCNT    GET COUNT
10E5 97 72              STA A  STRCN1    SAVE IT
10E7 39                 RTS              RETURN

                * RESTORE STRING POINTER INFO

10E8 DE 52      RSTSPT  LDX    STRGB1    GET POINTER
10EA DF 4E              STX    STRNGB    RESTORE
10EC DE 54              LDX    STRGE1
10EE DF 50              STX    STRNGE
10F0 96 72              LDA A  STRCN1    GET COUNT
10F2 97 7C              STA A  STRCNT    RESTORE IT
10F4 39                 RTS              RETURN

                * OVERLAY ROUTINE

10F5 BD 0A 51   OVERLA  JSR    TSTEMP
10F8 86 20              LDA A  #$20      SETUP SPACE
10FA 97 96              STA A  DELIM     AS DELIMITER
10FC 7F 00 8D           CLR    DRCTN
10FF DE 44              LDX    BUFPNT    SET POINTER TO BUFFER
1101 A6 00              LDA A  0,X       GET A CHAR.
1103 81 0D              CMP A  #CRGRET
1105 27 12              BEQ    OVRLA1
1107 BD 08 DC           JSR    CLASS
110A 5D                 TST B
110B 27 03              BEQ    OVRLA0
110D 7E 05 F4           JMP    SYNERR    REPORT ERROR
1110 97 96      OVRLA0  STA A  DELIM     SET DELIMITER
1112 08                 INX
1113 A6 00              LDA A  0,X       GET CHARACTER
1115 81 0D              CMP A  #CRGRET
1117 26 2F              BNE    OVRL35
1119 7F 00 6C   OVRLA1  CLR    REPEAT    DISABLE COMMAND REPEAT
111C DE 4A              LDX    NEWPOS    SET POINTER
111E BD 0A 58   OVRL11  JSR    OUTLIN    OUTPUT CUR. LINE
1121 96 6A              LDA A  NUMFLG
1123 26 05              BNE    OVRL12
1125 CE 11 A9           LDX    #OVRLST+8 POINT TO STRING
1128 20 03              BRA    OVRL16
112A CE 11 A1   OVRL12  LDX    #OVRLST   POINT TO STRING
112D BD 06 06   OVRL16  JSR    PSTRNG    OUTPUT IT
1130 CE 00 BD           LDX    #BUFFER   POINT TO IN BUFFER
1133 BD 06 1C   OVRLA2  JSR    INCHAR    GET A CHAR.
1136 27 E1              BEQ    OVRLA1
1138 81 0D              CMP A  #CRGRET
113A 27 07              BEQ    OVRLA3
113C A7 00              STA A  0,X
113E BD 06 56           JSR    BUFLIM
1141 20 F0              BRA    OVRLA2
1143 A7 00      OVRLA3  STA A  0,X
1145 CE 00 BD           LDX    #BUFFER   POINT TO BUFFER
```

```
1148 DF 44      OVRL35  STX     BUFPNT
114A C6 0D              LDA B   #CRGRET
114C DE 4A              LDX     NEWPOS      POINT TO POSITION
114E 08                 INX
114F 08                 INX
1150 08                 INX
1151 DF 40              STX     TEMP        SAVE POINTER
1153 DE 44      OVRLA4  LDX     BUFPNT
1155 A6 00      OVRL41  LDA A   0,X         GET A CHAR.
1157 08                 INX                 BUMP POINTER
1158 DF 44              STX     BUFPNT      SAVE IT
115A 81 0D              CMP A   #CRGRET
115C 27 3B              BEQ     OVRLA7
115E DE 40              LDX     TEMP
1160 7D 00 8D           TST     DRCTN
1163 26 08              BNE     OVRL43
1165 E1 00              CMP B   0,X         CHECK IT
1167 27 0B              BEQ     OVRLA5
1169 91 96              CMP A   DELIM       IS IT DELIMITER?
116B 27 02              BEQ     OVRL45
116D A7 00      OVRL43  STA A   0,X         PUT CHARACTER
116F 08         OVRL45  INX                 BUMP POINTER
1170 DF 40              STX     TEMP
1172 20 DF              BRA     OVRLA4
1174 96 8D      OVRLA5  LDA A   DRCTN        CHECK DIRECTION
1176 26 21              BNE     OVRLA7
1178 4F                 CLR A
1179 97 8E              STA A   CHRCNT      CLEAR COUNT
117B DE 44              LDX     BUFPNT      GET POINTER
117D 4C         OVRL55  INC A
117E E1 00              CMP B   0,X         CHECK CHAR.
1180 27 03              BEQ     OVRLA6
1182 08                 INX                 BUMP THE POINTER
1183 20 F8              BRA     OVRL55      REPEAT
1185 97 8F      OVRLA6  STA A   CHRCNT+1    SAVE COUNT
1187 DE 40              LDX     TEMP
1189 DF 58              STX     SPCPT1      SET POINTER
118B 86 01              LDA A   #1
118D 97 8D              STA A   DRCTN       SET DIRECTION
118F BD 0D D7           JSR     MAKSPC      MAKE ROOM
1192 C6 0D              LDA B   #CRGRET
1194 DE 44              LDX     BUFPNT      GET POINTER
1196 09                 DEX
1197 20 BC              BRA     OVRL41
1199 DE 4A      OVRLA7  LDX     NEWPOS      GET POSITION
119B BD 09 E8           JSR     VERLIN      VERIFY LINE
119E 7E 04 FA   OVRLA8  JMP     EDIT        RETURN

11A1 20         OVRLST  FCC     ' OVERLAY '
11AA 04                 FCB     4

                * MOVE COMMAND
```

```
11AB 7C 00 81  MOVE    INC     MOVFLG    SET FLAG
11AE 8D 1A             BSR     COPY      GO DO COPY
11B0 96 8C             LDA A   CPYDRC    WHICH DIRECTION?
11B2 97 8D             STA A   DRCTN
11B4 DE 94             LDX     TRGLIN    GET TARGET
11B6 DF 5A             STX     SPCPT2
11B8 DE 4A             LDX     NEWPOS    GET POSITION
11BA DF 58             STX     SPCPT1
11BC DE 48             LDX     CURPOS    GET CURRENT POS.
11BE DF 94             STX     TRGLIN    MAKE IT TARGET
11C0 BD 0F 17          JSR     DELCHR    DELETE LINES
11C3 DE 94             LDX     TRGLIN
11C5 DF 48             STX     CURPOS    FIX POSITION
11C7 7E 0A 4B          JMP     PRINT6

                * COPY LINES COMMAND

11CA DE 44     COPY    LDX     BUFPNT    POINT TO BUFFER
11CC 7C 00 77          INC     NXTFLG    SET FLAG
11CF 7C 00 76          INC     LINFLG    SET FLAG
11D2 BD 06 85          JSR     FINDT     FIND TARGET
11D5 9C 9A             CPX     FILEND
11D7 27 1A             BEQ     COPY0
11D9 7F 00 76          CLR     LINFLG
11DC 7F 00 77          CLR     NXTFLG
11DF 9C 98             CPX     FILBEG    BEGINNING?
11E1 26 04             BNE     COPY02
11E3 9C 4A             CPX     NEWPOS
11E5 27 04             BEQ     COPY05
11E7 96 8D     COPY02  LDA A   DRCTN     FIX DIRECTION
11E9 97 8C             STA A   CPYDRC
11EB 7F 00 8D  COPY05  CLR     DRCTN
11EE BD 09 CA          JSR     TSTOVR    LIMITS?
11F1 27 03             BEQ     COPY1
11F3 7E 05 E2  COPY0   JMP     NOTFND    REPORT ERROR
11F6 BD 09 8F  COPY1   JSR     FNDCRT    FIND NEXT C. R.
11F9 08                INX               BUMP POINTER ONE
11FA DF 58             STX     SPCPT1
11FC DE 44             LDX     BUFPNT
11FE BD 06 85          JSR     FINDT     GO FIND TARGET
1201 BD 09 CA          JSR     TSTOVR    LIMITS?
1204 26 ED             BNE     COPY0
1206 7D 00 8D          TST     DRCTN     DIRECTION?
1209 26 08             BNE     COPY15
120B DE 4A             LDX     NEWPOS    GET POINTER
120D DF 40             STX     TEMP
120F DE 94             LDX     TRGLIN
1211 20 06             BRA     COPY18
1213 DE 94     COPY15  LDX     TRGLIN    GET TARGET
1215 DF 40             STX     TEMP      SAVE IT
1217 DE 4A             LDX     NEWPOS
1219 BD 09 8F  COPY18  JSR     FNDCRT    GET NEXT C. R.
121C 08                INX               BUMP POINTER
```

```
121D DF 94              STX     TRGLIN    SET TARGET
121F DE 40              LDX     TEMP
1221 DF 4A              STX     NEWPOS
1223 4F                 CLR A             CLEAR ACCUMULATORS
1224 5F                 CLR B
1225 08          COPY2  INX               BUMP THE POINTER
1226 4C                 INC A             BUMP COUNT
1227 26 01              BNE     COPY25
1229 5C                 INC B
122A 9C 94       COPY25 CPX     TRGLIN    FINISHED?
122C 27 10              BEQ     COPY3
122E 9C 58              CPX     SPCPT1    OVERLAP?
1230 26 F3              BNE     COPY2
1232 CE 12 8B           LDX     #OVLPST   POINT TO STRING
1235 BD 06 06           JSR     PSTRNG    OUTPUT IT
1238 7F 00 6D           CLR     MSLFLG
123B 7E 04 FA           JMP     EDIT      RETURN
123E 97 8F       COPY3  STA A   CHRCNT+1  SAVE COUNT
1240 D7 8E              STA B   CHRCNT
1242 86 01              LDA A   #1        SET DIRECTION
1244 97 8D              STA A   DRCTN
1246 BD 0D D7           JSR     MAKSPC    MAKE ROOM FOR LINES
1249 DE 5A              LDX     SPCPT2
124B DF 48              STX     CURPOS    SET CUR. POSITION
124D DE 58              LDX     SPCPT1
124F DF 42              STX     XSAVE
1251 DE 94              LDX     TRGLIN    GET TARGET
1253 DF 58              STX     SPCPT1
1255 DE 4A              LDX     NEWPOS
1257 DF 40              STX     TEMP      SET POINTER
1259 BD 0E 32           JSR     MAKS55    MOVE LINES
125C DE 42              LDX     XSAVE     RESTORE POINTER
125E 7F 00 73           CLR     OVRBEG
1261 BD 09 99           JSR     BAKONE    MOVE BACK ONE
1264 96 73              LDA A   OVRBEG    LIMIT?
1266 27 18              BEQ     COPY5
1268 BD 09 4C           JSR     CLRNUM    CLEAR NUMBER
126B 7C 00 84    COPY4  INC     CHKFLG    SET FLAG
126E BD 08 4D           JSR     RENUM2    GO RENUMBER
1271 DE 48              LDX     CURPOS    GET POSITION
1273 BD 09 99           JSR     BAKONE
1276 DF 48              STX     CURPOS    SET CUR. POSITION
1278 96 81              LDA A   MOVFLG    MOVE?
127A 27 01              BEQ     COPY45
127C 39                 RTS               RETURN
127D 7E 0A 4B    COPY45 JMP     PRINT6
1280 BD 08 35    COPY5  JSR     GETNUM    GET LINE NUMBER
1283 4F                 CLR A
1284 97 93              STA A   NUMBER+2
1286 BD 09 B8           JSR     UPONE
1289 20 E0              BRA     COPY4

128B 53          OVLPST FCC     'SOURCE OVERLAPS DESTINATION'
12A6 04                 FCB     4
```

```
                        * TAB SET COMMAND

12A7 CE 00 9E  TAB      LDX     #TABBUF   SET POINTER
12AA DF 9C              STX     TABPNT
12AC DE 44     TAB2     LDX     BUFPNT    POINT TO BUFFER
12AE BD 06 15           JSR     SKIPSP
12B1 DF 44              STX     BUFPNT    SAVE
12B3 BD 07 FF           JSR     TSTEND
12B6 27 2F              BEQ     TAB6
12B8 BD 08 DC           JSR     CLASS     CLASSIFY CHAR.
12BB C1 01              CMP B   #1        IS IT A NUMBER?
12BD 27 07              BEQ     TAB4
12BF 22 2E              BHI     TAB8
12C1 08                 INX               BUMP THE POINTER
12C2 DF 44              STX     BUFPNT    SAVE IT
12C4 20 E6              BRA     TAB2
12C6 BD 08 FE  TAB4     JSR     BCDCON    GET COLUMN
12C9 DF 44              STX     BUFPNT
12CB 5F                 CLR B
12CC BD 09 E1           JSR     TSTNUM    IS IT ZERO?
12CF 27 16              BEQ     TAB6
12D1 5C        TAB5     INC B             BUMP COUNT
12D2 37                 PSH B
12D3 BD 09 D3           JSR     DECNUM    DEC THE COUNT
12D6 33                 PUL B
12D7 26 F8              BNE     TAB5
12D9 DE 9C              LDX     TABPNT    POINT TO TABS
12DB E7 00              STA B   0,X       SAVE COUNT
12DD 08                 INX
12DE DF 9C              STX     TABPNT    FIX TAB POINTER
12E0 8C 00 B2           CPX     #TABEND
12E3 27 02              BEQ     TAB6
12E5 20 C5              BRA     TAB2
12E7 4F        TAB6     CLR A
12E8 DE 9C              LDX     TABPNT
12EA A7 00              STA A   0,X       CLEAR TAB
12EC 7E 0A D9           JMP     NUMSE6
12EF 7E 05 F4  TAB8     JMP     SYNERR    REPORT ERROR

                        * PRINT HEADER COMMAND

12F2 CE 00 9E  HEADER   LDX     #TABBUF   SET POINTER
12F5 DF 9C              STX     TABPNT
12F7 DE 44              LDX     BUFPNT
12F9 BD 06 15           JSR     SKIPSP    SKIP ALL SPACES
12FC BD 07 FF           JSR     TSTEND
12FF 27 1C              BEQ     HEAD42
1301 BD 08 DC  HEADE2   JSR     CLASS     CLASSIFY CHAR.
1304 C1 01              CMP B   #1        IS IT NUMBER?
1306 26 E7              BNE     TAB8      ERROR
1308 BD 08 FE           JSR     BCDCON    GET NUMBER COUNT
130B DF 44              STX     BUFPNT
```

```
130D BD 09 E1           JSR     TSTNUM      IS IT ZERO?
1310 27 40              BEQ     HEADE7
1312 5F                 CLR B
1313 5C         HEADE3  INC B               BUMP COUNTER
1314 37                 PSH B
1315 BD 09 D3           JSR     DECNUM      DEC NUMBER
1318 33                 PUL B
1319 26 F8              BNE     HEADE3
131B D7 97      HEADE4  STA B   HEDCNT      SAVE COUNT
131D BD 09 4C   HEAD42  JSR     CLRNUM      CLEAR NUMBER
1320 BD 05 DF           JSR     PCRLF       OUTPUT C.R. L.F.
1323 96 6A              LDA A   NUMFLG      LINE NUMBERS ON?
1325 27 08              BEQ     HEADE5
1327 C6 08              LDA B   #8          SET COUNT
1329 BD 0A 76   HEAD45  JSR     OUTSPC      OUT SPACE
132C 5A                 DEC B
132D 26 FA              BNE     HEAD45
132F BD 0A 76   HEADE5  JSR     OUTSPC
1332 5F                 CLR B               CLEAR COUNT
1333 37         HEAD55  PSH B
1334 BD 08 19           JSR     INCNUM      BUMP NUMBER
1337 33                 PUL B
1338 5C                 INC B               BUMP COUNT
1339 DE 9C              LDX     TABPNT      GET TAB COL.
133B E1 00              CMP B   0,X         THERE?
133D 26 0A              BNE     HEAD57
133F 86 2D              LDA A   #'-         SET UP '-'
1341 BD 02 09           JSR     OUTCH       OUTPUT IT
1344 08                 INX                 BUMP POINTER
1345 DF 9C              STX     TABPNT
1347 20 05              BRA     HEAD58
1349 96 92      HEAD57  LDA A   NUMBER+1    GET NUMBER
134B BD 0A C0           JSR     OUTHR       OUTPUT IT
134E D1 97      HEAD58  CMP B   HEDCNT
1350 26 E1              BNE     HEAD55      REPEAT TIL DONE
1352 7E 0A D9   HEADE7  JMP     NUMSE6

                * SET UP ZONE COLUMN COMMAND

1355 DE 44      SZONE   LDX     BUFPNT      POINT TO BUFFER
1357 BD 06 15           JSR     SKIPSP
135A BD 07 FF           JSR     TSTEND
135D 27 0A              BEQ     SZONE2
135F BD 08 DC           JSR     CLASS       CLASSIFY CHARACTER
1362 C1 01              CMP B   #1          IS IT A NUMBER?
1364 27 0A              BEQ     SZONE3
1366 22 38              BHI     SZONE8
1368 08                 INX
1369 DF 44      SZONE2  STX     BUFPNT      SAVE POINTER
136B CE 00 01           LDX     #$0001      SET COLUMN 1
136E 20 07              BRA     SZONE4
1370 BD 08 FE   SZONE3  JSR     BCDCON      GET NUMBER
1373 DF 44              STX     BUFPNT      SAVE POINTER
```

```
1375 DE 91              LDX     NUMBER
1377 DF 5E      SZONE4  STX     ZONE1       FIX ZONE1
1379 DE 44              LDX     BUFPNT
137B BD 06 15   SZONE5  JSR     SKIPSP      SKIP ALL SPACES
137E BD 07 FF           JSR     TSTEND
1381 27 0C              BEQ     SZONE6
1383 BD 08 DC           JSR     CLASS       GO CLASSIFY
1386 C1 01              CMP B   #1          IS IT A NUMBER?
1388 27 0A              BEQ     SZONE7
138A 22 14              BHI     SZONE8      ERROR
138C 08                 INX                 BUMP POINTER
138D 20 EC              BRA     SZONE5
138F CE 01 36   SZONE6  LDX     #$0136      SET COLUMN 136
1392 20 07              BRA     SZON75
1394 BD 08 FE   SZONE7  JSR     BCDCON      GET NUMBER
1397 DF 44              STX     BUFPNT      SAVE POINTER
1399 DE 91              LDX     NUMBER
139B DF 60      SZON75  STX     ZONE2       SET ZONE2
139D 7E 0A D9           JMP     NUMSE6
13A0 7E 05 CF   SZONE8  JMP     ERROR       REPORT ERROR

                * SET SPECIAL CHARATERS COMMAND

13A3 DE 44      SET     LDX     BUFPNT      SET POINTER
13A5 BD 06 15           JSR     SKIPSP
13A8 DF 44              STX     BUFPNT
13AA DF 40              STX     TEMP        SAVE POINTER
13AC CE 14 1D           LDX     #CHRTBL     POINT TO TABLE
13AF 7E 05 88           JMP     EDIT6       GO FIND NAME

                * SET SPECIALS HERE
                * TAB
13B2 CE AC 06   STAB    LDX     #TABCH      POINT TO TAB CHAR
13B5 20 12              BRA     SETC

                * FILL
13B7 CE 00 B3   SFILL   LDX     #FILL       POINT TO FILL
13BA 20 0D              BRA     SETC

                * EOL
13BC CE AC 02   SEOL    LDX     #EOL        POINT TO EOL CHAR
13BF 20 08              BRA     SETC

                * DCC
13C1 CE 00 B5   SDCC    LDX     #DCC        POINT TO DCC
13C4 20 03              BRA     SETC

                * LINO
13C6 CE 00 B4   SLINO   LDX     #LINO       POINT TO IT

                * SET THE CHARACTER

13C9 DF 40      SETC    STX     TEMP        SAVE POINTER
```

```
13CB DE 44              LDX     BUFPNT      GO TO BUFFER
13CD BD 06 15           JSR     SKIPSP
13D0 81 3D              CMP A   #'=         IS IT =
13D2 26 3F              BNE     SETC8       ERROR
13D4 8D 40              BSR     CHFRQU
13D6 26 3B              BNE     SETC8       ERROR
13D8 8D 3C              BSR     CHFRQU
13DA 26 04              BNE     SETC2
13DC 4F                 CLR A               SET NULL CHAR.
13DD 36                 PSH A
13DE 20 0F              BRA     SETC4
13E0 BD 08 DC  SETC2    JSR     CLASS       GO CLASSIFY
13E3 5D                 TST B
13E4 26 2D              BNE     SETC8       ERROR?
13E6 81 0D              CMP A   #CRGRET
13E8 27 29              BEQ     SETC8
13EA 36                 PSH A               SAVE CHAR
13EB 8D 29              BSR     CHFRQU
13ED 26 24              BNE     SETC8       ERROR
13EF 08        SETC4    INX
13F0 DF 44              STX     BUFPNT      SAVE POSITION
13F2 BD 0B 6E           JSR     TFORCR      TEST END
13F5 32                 PUL A               GET CHAR
13F6 DE 40              LDX     TEMP        RESTORE POINTER
13F8 8C 00 B3           CPX     #FILL       IS IT FILL CHAR?
13FB 26 07              BNE     SETC5
13FD 4D                 TST A
13FE 26 0E              BNE     SETC6
1400 86 20              LDA A   #'          SETUP SPACE
1402 20 0A              BRA     SETC6
1404 8C 00 B4  SETC5    CPX     #LINO       IS IT LINO?
1407 26 05              BNE     SETC6
1409 4D                 TST A
140A 26 02              BNE     SETC6
140C 86 23              LDA A   #'#         SET IT
140E A7 00     SETC6    STA A   0,X
1410 7E 0A D9           JMP     NUMSE6      RETURN
1413 7E 05 F4  SETC8    JMP     SYNERR      REPORT ERROR

               * CHECK FOR QUOTE

1416 08        CHFRQU   INX                 BUMP POINTER
1417 BD 06 15           JSR     SKIPSP      SKIP SPACES
141A 81 27              CMP A   #''
141C 39                 RTS

               * SPECIAL CHARACTER TABLE

141D 54        CHRTBL   FCC     'TAB'
1420 00                 FCB     0
1421 13 B2              FDB     STAB
1423 46                 FCC     'FILL'
```

```
1427 00                  FCB    0
1428 13 B7               FDB    SFILL
142A 45                  FCC    'EOL'
142D 00                  FCB    0
142E 13 BC               FDB    SEOL
1430 4C                  FCC    'LINO'
1434 00                  FCB    0
1435 13 C6               FDB    SLINO
1437 44                  FCC    'DCC'
143A 00                  FCB    0
143B 13 C1               FDB    SDCC
143D 00                  FCB    0

                * EXPAND TABS COMMAND

143E BD 0A 51  EXPAND    JSR    TSTEMP
1441 DE 44               LDX    BUFPNT    GET POINTER
1443 BD 06 85            JSR    FINDT     FIND TARGET
1446 DE 4A               LDX    NEWPOS    SAVE IT
1448 86 01               LDA A  #1
144A 97 8B               STA A  PRNFLG    SET FLAG
144C DF 48     EXPAN1    STX    CURPOS    SET CURRENT
144E 9C 94               CPX    TRGLIN    LAST LINE?
1450 26 03               BNE    EXPAN2
1452 7F 00 8B            CLR    PRNFLG    CLEAR FLAG
1455 8D 10     EXPAN2    BSR    EXPLIN    GO DO LINE
1457 96 8B               LDA A  PRNFLG    DONE?
1459 27 09               BEQ    EXPAN5
145B DE 4A               LDX    NEWPOS    GET POINTER
145D BD 08 60            JSR    NXTLIN    FIND NEXT LINE
1460 DF 4A               STX    NEWPOS    SAVE
1462 20 E8               BRA    EXPAN1
1464 7E 0A 4B  EXPAN5    JMP    PRINT6

                * EXPAND TABS IN ONE LINE

1467 B6 AC 06  EXPLIN    LDA A  TABCH
146A 91 B3               CMP A  FILL      CHECK IF FILL=TAB
146C 27 52               BEQ    EXPLI7
146E CE 00 9E            LDX    #TABBUF   POINT TO TABS
1471 DF 9C               STX    TABPNT
1473 E6 00               LDA B  0,X       GET COLUMN
1475 27 49               BEQ    EXPLI7
1477 5F                  CLR B            CLEAR COUNT
1478 D7 8E               STA B  CHRCNT
147A DE 4A               LDX    NEWPOS    POINT TO LINE
147C 9C 9A               CPX    FILEND
147E 26 03               BNE    EXPLI1
1480 7E 04 FA            JMP    EDIT
1483 08        EXPLI1    INX              BUMP 3 TIMES
1484 08                  INX              PAST LINE NO.
1485 08                  INX
```

```
1486 5C         EXPLI2  INC B                   BUMP COUNTER
1487 A6 00              LDA A  0,X              CHECK FOR TAB
1489 81 0D              CMP A  #CRGRET
148B 27 33              BEQ    EXPLI7
148D B1 AC 06           CMP A  TABCH            IS IT TAB?
1490 27 03              BEQ    EXPLI3
1492 08                 INX                     BUMP THE POINTER
1493 20 F1              BRA    EXPLI2
1495 DF 40      EXPLI3  STX    TEMP             SAVE POSITION
1497 DE 9C              LDX    TABPNT
1499 E1 00      EXPL35  CMP B  0,X              CHECK COLUMN
149B 24 1E              BCC    EXPLI6
149D 86 FF              LDA A  #$FF             SET COUNT
149F 4C         EXPLI4  INC A
14A0 5C                 INC B
14A1 E1 00              CMP B  0,X              TAB COL. YET?
14A3 26 FA              BNE    EXPLI4
14A5 97 8F              STA A  CHRCNT+1         SAVE COUNT
14A7 DE 40              LDX    TEMP
14A9 DF 58              STX    SPCPT1           SET SPACE POINTER
14AB BD 0D D7           JSR    MAKSPC           GO MAKE ROOM
14AE D6 8F              LDA B  CHRCNT+1
14B0 5C                 INC B
14B1 96 B3              LDA A  FILL             GET FILL CHARACTER
14B3 A7 00      EXPLI5  STA A  0,X              PUT CHARACTER
14B5 08                 INX
14B6 5A                 DEC B                   DEC COUNT
14B7 26 FA              BNE    EXPLI5
14B9 20 AC              BRA    EXPLIN           REPEAT
14BB 08         EXPLI6  INX                     BUMP POINTER
14BC A6 00              LDA A  0,X
14BE 26 D9              BNE    EXPL35
14C0 39         EXPLI7  RTS                     RETURN


                * APPEND COMMAND

14C1 BD 0A 51   APPEND  JSR    TSTEMP
14C4 DE 44              LDX    BUFPNT           GET POINTER
14C6 BD 06 15           JSR    SKIPSP
14C9 BD 07 FF           JSR    TSTEND           ALL?
14CC 26 03              BNE    APPEN1
14CE 7E 05 F4   APPEN0  JMP    SYNERR
14D1 BD 08 DC   APPEN1  JSR    CLASS            GO CLASSIFY
14D4 5D                 TST B
14D5 26 F7              BNE    APPEN0
14D7 BD 07 96           JSR    SETDEL           SET DELIMITERS
14DA BD 08 DC           JSR    CLASS            CLASSIFY CHARACTER
14DD C1 01              CMP B  #1               IS IT NUMBER?
14DF 26 13              BNE    APPEN3
14E1 BD 08 FE           JSR    BCDCON           GET COLUMN NO.
14E4 BD 09 E1           JSR    TSTNUM           IS IT ZERO?
14E7 27 0B              BEQ    APPEN3
14E9 4F                 CLR A
```

```
14EA 4C          APPEN2  INC A                BUMP COUNTER
14EB 36                  PSH A
14EC BD 09 D3            JSR     DECNUM       DEC NUMBER
14EF 32                  PUL A                RESTORE COUNT
14F0 26 F8               BNE     APPEN2
14F2 97 7B               STA A   APPCOL       SAVE COUNT
14F4 BD 10 DB    APPEN3  JSR     SVSTPT       SAVE DEL INFO
14F7 DE 44               LDX     BUFPNT       GET POINTER
14F9 BD 06 85            JSR     FINDT        FIND TARGET
14FC BD 10 E8            JSR     RSTSPT       RESTORE DEL INFO
14FF 7F 00 8E            CLR     CHRCNT
1502 7C 00 8B            INC     PRNFLG       SET FLAG
1505 DE 4A               LDX     NEWPOS       SET POINTER
1507 9C 94       APPE35  CPX     TRGLIN       AT TARGET?
1509 26 03               BNE     APPEN4
150B 7F 00 8B            CLR     PRNFLG       CLEAR FLAG
150E 08          APPEN4  INX                  BUMP 3 TIMES
150F 08                  INX
1510 08                  INX
1511 96 7B               LDA A   APPCOL       GET COL. NUMBER
1513 26 06               BNE     APPEN5
1515 09                  DEX
1516 BD 09 8F            JSR     FNDCRT       GET TO C.R.
1519 20 37               BRA     APPEN7
151B 16          APPEN5  TAB
151C 5A          APPE53  DEC B                DEC COUNT
151D 27 1C               BEQ     APPE65
151F A6 00               LDA A   0,X          CHECK CHARACTER
1521 81 0D               CMP A   #CRGRET
1523 27 03               BEQ     APPEN6
1525 08                  INX                  BUMP POINTER
1526 20 F4               BRA     APPE53
1528 DF 58       APPEN6  STX     SPCPT1       SET POSITION
152A D7 8F               STA B   CHRCNT+1
152C 37                  PSH B
152D BD 0D D7            JSR     MAKSPC       GO MAKE MORE ROOM
1530 33                  PUL B
1531 86 20               LDA A   #' '         SET UP SPACE
1533 A7 00       APPE63  STA A   0,X          PUT IT
1535 08                  INX                  BUMP POINTER
1536 5A                  DEC B                DEC THE COUNT
1537 26 FA               BNE     APPE63
1539 20 17               BRA     APPEN7
153B DF 58       APPE65  STX     SPCPT1       SET POSITION
153D DF 40               STX     TEMP
153F A6 00       APPE66  LDA A   0,X          GET CHAR.
1541 81 0D               CMP A   #CRGRET      IS IT C.R.?
1543 27 04               BEQ     APPE67
1545 08                  INX
1546 5C                  INC B
1547 20 F6               BRA     APPE66
1549 D7 8F       APPE67  STA B   CHRCNT+1
154B DF 5A               STX     SPCPT2
154D BD 0F 17            JSR     DELCHR       DELETE REST OF LINE
```

```
1550 DE 40              LDX     TEMP        GET POINTER
1552 DF 58      APPEN7  STX     SPCPT1
1554 96 7C              LDA A   STRCNT      GET COUNT
1556 27 12              BEQ     APPE78
1558 97 8F      APPE72  STA A   CHRCNT+1
155A BD 0D D7           JSR     MAKSPC      GO MAKE ROOM
155D DE 50              LDX     STRNGE      POINT TO STRING
155F DF 58              STX     SPCPT1
1561 DE 4E              LDX     STRNGB
1563 DF 40              STX     TEMP        SET END
1565 BD 0E 32           JSR     MAKS55      PUT IN STRING
1568 DE 5A              LDX     SPCPT2      GET POINTER
156A 5F         APPE78  CLR B
156B BD 09 9F           JSR     BAKON2
156E BD 09 E8           JSR     VERLIN
1571 96 8B              LDA A   PRNFLG      DONE?
1573 27 07              BEQ     APPEN8
1575 BD 08 60           JSR     NXTLIN      FIND NEXT LINE
1578 DF 4A              STX     NEWPOS
157A 20 8B              BRA     APPE35      REPEAT
157C DF 48      APPEN8  STX     CURPOS
157E 7E 0A 4B   APPEN9  JMP     PRINT6

                * SAVE CURRENT FILE ON TAPE

1581 BD 0B 6E   SAVE    JSR     TFORCR
1584 DE 98              LDX     FILBEG      SET POINTER
1586 DF 58              STX     SPCPT1
1588 DE 9A              LDX     FILEND
158A DF 5A              STX     SPCPT2      SET END
158C 8D 29              BSR     RECORD      GO RECORD IT
158E 7E 0A D9   SAVE4   JMP     NUMSE6

                * WRITE PART OF FILE TO TAPE

1591 BD 0A 51   WRITE   JSR     TSTEMP
1594 DE 44              LDX     BUFPNT      SET POINTER
1596 BD 06 85           JSR     FINDT       FIND TARGET
1599 96 8D              LDA A   DRCTN       CHECK DIRECTION
159B 26 0C              BNE     WRITE2
159D BD 09 8F           JSR     FNDCRT
15A0 08                 INX
15A1 DF 5A              STX     SPCPT2      SET POINTER
15A3 DE 4A              LDX     NEWPOS
15A5 DF 58              STX     SPCPT1      SET BEGINNING
15A7 20 0A              BRA     WRITE4
15A9 DF 58      WRITE2  STX     SPCPT1
15AB DE 4A              LDX     NEWPOS
15AD BD 09 8F           JSR     FNDCRT
15B0 08                 INX
15B1 DF 5A              STX     SPCPT2      SET END
15B3 8D 02      WRITE4  BSR     RECORD      GO RECORD IT
```

```
15B5 20 D7      WRITE5  BRA     SAVE4


                * RECORD RECORD

15B7 86 02      RECORD  LDA A   #2          SET CODE
15B9 BD 16 6D           JSR     TORDSK      TAPE OR DISK?
15BC 27 07              BEQ     RECOR1
15BE 96 B9              LDA A   RONCH       READER ON
15C0 BD 02 0F           JSR     TOUCH
15C3 8D 30              BSR     TDELAY
15C5 DE 58      RECOR1  LDX     SPCPT1      GET POINTER
15C7 9C 5A      RECOR2  CPX     SPCPT2      FINISHED?
15C9 27 13              BEQ     RECOR4
15CB 08                 INX                 BUMP PAST LINE NUM
15CC 08                 INX
15CD 08                 INX
15CE A6 00      RECO25  LDA A   0,X         GET CHARACTER
15D0 81 0D              CMP A   #$D         IS IT CR?
15D2 27 05              BEQ     RECOR3
15D4 8D 45              BSR     SOUCH       OUTPUT IT
15D6 08                 INX                 BUMP THE POINTER
15D7 20 F5              BRA     RECO25
15D9 8D 40      RECOR3  BSR     SOUCH       OUTPUT CHARACTER
15DB 08                 INX
15DC 20 E9              BRA     RECOR2      REPEAT
15DE 96 20      RECOR4  LDA A   DSKFLG      CHECK MODE
15E0 27 03              BEQ     RECO45
15E2 7E 16 4F           JMP     CLSFCB      GO CLOSE FILE
15E5 86 1A      RECO45  LDA A   #$1A
15E7 BD 02 0F           JSR     TOUCH       OUTPUT EOF
15EA CE FF FF   RECOR5  LDX     #$FFFF      SET COUNT
15ED 09         RECOR6  DEX                 DEC THE COUNT
15EE 26 FD              BNE     RECOR6      LOOP TIL DONE
15F0 96 BA      RECOR7  LDA A   ROFCH       TURN OFF TAPE
15F2 BD 02 0F           JSR     TOUCH


                * DELAY FOR TAPE

15F5 96 B6      TDELAY  LDA A   DELAY       GET DELAY
15F7 27 09      DELAY1  BEQ     DELAY4      IS THERE ONE?
15F9 CE FF FF           LDX     #$FFFF      SET COUNT
15FC 09         DELAY2  DEX                 DEC THE COUNT
15FD 26 FD              BNE     DELAY2
15FF 4A                 DEC A               DEC COUNT
1600 20 F5              BRA     DELAY1      REPEAT
1602 39         DELAY4  RTS                 RETURN


                * PUT GAP ON TAPE (40 NULLS)

1603 BD 0B 6E   GAP     JSR     TFORCR      CHECK END
1606 96 B9              LDA A   RONCH       GET ON CHAR
```

```
1608 BD 02 0F          JSR     TOUCH
160B 8D E8             BSR     TDELAY      GO DELAY
160D C6 28             LDA B   #40         SETUP COUNT
160F 4F        GAP2    CLR A               SET NULL
1610 BD 02 0F          JSR     TOUCH       OUTPUT IT
1613 5A                DEC B               DEC THE COUNT
1614 26 F9             BNE     GAP2
1616 8D D8             BSR     RECOR7      FINISH UP
1618 7E 15 B5          JMP     WRITE5

               * SPECIAL OUTPUT ROUTINE

161B 7D 00 20  SOUCH   TST     DSKFLG      DISK OPERATION?
161E 27 29             BEQ     SOUCH6
1620 DF 25             STX     INDEX9      SAVE X REG
1622 7D 00 21          TST     DRWFLG      READ OR WRITE?
1625 27 05             BEQ     SOUCH2
1627 CE A8 40          LDX     #FCB        SET POINTER
162A 20 03             BRA     SOUCH3
162C CE 19 DD  SOUCH2  LDX     #WFCB       SET POINTER
162F BD B4 06  SOUCH3  JSR     FMS         CALL FMS
1632 27 18             BEQ     SOUCH8      ERRORS?
1634 BD AD 3F  SOUCH4  JSR     RPTERR      REPORT ERROR
1637 DE 25             LDX     INDEX9      RESTORE X
1639 7D 00 21          TST     DRWFLG
163C 26 06             BNE     SOUCH5
163E BD B4 03          JSR     FMSCLS      CLOSE FMS
1641 7E AD 03          JMP     WARMS       RETURN
1644 8D 09     SOUCH5  BSR     CLSFCB      CLOSE FILE
1646 7E 04 FA          JMP     EDIT
1649 7E 02 0F  SOUCH6  JMP     TOUCH
164C DE 25     SOUCH8  LDX     INDEX9      RESTORE X REG
164E 39                RTS                 RETURN

               * CLOSE FILE ON DISK

164F DF 25     CLSFCB  STX     INDEX9
1651 96 21             LDA A   DRWFLG      TEST FLAG
1653 27 15             BEQ     CLSFC2
1655 7F 00 21          CLR     DRWFLG      CLEAR FLAGS
1658 7F 00 20          CLR     DSKFLG
165B CE A8 40          LDX     #FCB        SET POINTER
165E 86 04     CLSFC1  LDA A   #4          SET FOR CLOSE
1660 A7 00             STA A   0,X
1662 BD B4 06          JSR     FMS         CALL FMS
1665 27 03             BEQ     CLSFC2
1667 BD AD 3F          JSR     RPTERR      REPORT ERROR
166A DE 25     CLSFC2  LDX     INDEX9      RESTORE X
166C 39                RTS                 RETURN

               * ASK IF TAPE OR DISK
```

```
166D 7F 00 20  TORDSK  CLR    DSKFLG    CLEAR FLAGS
1670 7F 00 21          CLR    DRWFLG
1673 97 23             STA A  ACCT      SAVE ACTION
1675 CE 16 F6          LDX    #TDST     POINT TO STRING
1678 BD 06 06          JSR    PSTRNG
167B BD AD 15          JSR    GETCHR    GET RESPONSE
167E 81 5F             CMP A  #$5F      IS IT LOWER?
1680 23 02             BLS    TORDS2
1682 80 20             SUB A  #$20      BIAS CHARACTER
1684 81 44     TORDS2  CMP A  #'D       IS IT DISK?
1686 26 62             BNE    TORDS4
1688 FE AC 14          LDX    DBFPNT    GET POINTER
168B DF 27             STX    SAVEIT    SAVE IT
168D B6 AC 11          LDA A  LSTTRM    GET TERM
1690 97 22             STA A  TTERM     SAVE IT
1692 CE 18 6B  TORDS3  LDX    #NMST     POINT TO STRING
1695 BD 06 06          JSR    PSTRNG    OUTPUT IT
1698 CE 00 BD          LDX    #BUFFER   POINT TO BUFFER
169B FF AC 14          STX    DBFPNT
169E BD 06 1C  TORD32  JSR    INCHAR    GET NAME
16A1 27 EF             BEQ    TORDS3    DELETE?
16A3 A7 00             STA A  0,X       PUT CHARACTER
16A5 08                INX              BUMP POINTER
16A6 81 0D             CMP A  #$D       IS IT CR?
16A8 26 F4             BNE    TORD32
16AA CE A8 40          LDX    #FCB      POINT TO FCB
16AD BD AD 2D          JSR    GETFIL    GET NAME
16B0 24 0A             BCC    TORD35    ERROR?
16B2 8D 57             BSR    RESTBF    RESTORE BUFFER
16B4 CE 18 32          LDX    #ILST     POINT TO STRING
16B7 BD 06 06  TORD33  JSR    PSTRNG    OUTPUT IT
16BA 20 32             BRA    TORDS6
16BC 8D 4D     TORD35  BSR    RESTBF    RESTORE BUFFER
16BE 86 01             LDA A  #1
16C0 97 20             STA A  DSKFLG
16C2 97 21             STA A  DRWFLG    SET FLAGS
16C4 CE A8 40          LDX    #FCB      SET POINTER
16C7 BD AD 33          JSR    SETEXT    SET DEFAULT EXT
16CA CE A8 40          LDX    #FCB
16CD 96 23             LDA A  ACCT      GET ACTION
16CF A7 00             STA A  0,X
16D1 BD B4 06          JSR    FMS       CALL FMS
16D4 27 14             BEQ    TORDS4    ERRORS?
16D6 A6 01             LDA A  1,X       CHECK ERROR
16D8 81 04             CMP A  #NFER     NO FILE?
16DA 27 09             BEQ    TORD37
16DC 81 03             CMP A  #FEER     FILE EXISTS?
16DE 26 0B             BNE    TORDS5
16E0 CE 18 44          LDX    #FEST     POINT TO STRING
16E3 20 D2             BRA    TORD33
16E5 CE 18 77  TORD37  LDX    #NFST     POINT TO STRING
16E8 20 CD             BRA    TORD33
16EA 39        TORDS4  RTS              RETURN
```

```
16EB BD AD 3F  TORDS5  JSR    RPTERR    REPORT ERROR
16EE 4F        TORDS6  CLR A            CLEAR FLAGS
16EF 97 20             STA A  DSKFLG
16F1 97 21             STA A  DRWFLG
16F3 7E 04 FA          JMP    EDIT

16F6 54        TDST    FCC    'TAPE OR DISK (T-D)? '
170A 04                FCB    4


               * RESTORE BUFFER

170B DE 27     RESTBF  LDX    SAVEIT    GET POINTER
170D FF AC 14          STX    DBFPNT
1710 96 22             LDA A  TTERM     GET TERM
1712 B7 AC 11          STA A  LSTTRM    RESTORE
1715 39                RTS              RETURN


               * READ OVER FLOW

1716 DF 25     ROVER   STX    INDEX9    SAVE X
1718 96 20             LDA A  DSKFLG    CHECK DISK IO
171A 27 04             BEQ    ROVER2
171C 96 21             LDA A  DRWFLG
171E 27 06             BEQ    ROVER4
1720 CE 0E 45  ROVER2  LDX    #NORMST   NO ROOM STRING
1723 BD 06 06          JSR    PSTRNG    PRINT STRING
1726 DE 25     ROVER4  LDX    INDEX9
1728 20 6A             BRA    READ5     JUMP AHEAD


               * READ ROUTINE

172A BD 0B 6E  READ    JSR    TFORCR    CHECK END
172D 86 01             LDA A  #1        SET CODE
172F BD 16 6D          JSR    TORDSK    TAPE OR DISK?
1732 27 0A             BEQ    READ1
1734 01                NOP
1735 0F                SEI
1736 96 B7             LDA A  TONCH     TAPE ON
1738 BD 02 0F          JSR    TOUCH
173B BD 15 F5          JSR    TDELAY    GO DELAY
173E BD 09 4C  READ1   JSR    CLRNUM    CLEAR NUMBER
1741 DE 9A             LDX    FILEND    GET END POINTER
1743 DF 40             STX    TEMP      SAVE VALUE
1745 9C 98             CPX    FILBEG    EMPTY FILE?
1747 27 08             BEQ    READ2
1749 BD 09 99          JSR    BAKONE    MOVE BACK ONE
174C BD 08 35          JSR    GETNUM    GET NUMBER
174F DE 9A             LDX    FILEND    GET POINTER
1751 DF 25     READ2   STX    INDEX9    SAVE POINTER
1753 96 25             LDA A  INDEX9
1755 7D 00 21          TST    DRWFLG    READ OR WRITE?
```

```
1758 26 06              BNE     READ25
175A 91 29              CMP A   BUFEND
175C 24 B8              BHS     ROVER       OVER END?
175E 20 04              BRA     READ27
1760 91 2B      READ25  CMP A   RBFEND
1762 24 B2              BHS     ROVER
1764 6F 00      READ27  CLR     0,X         CLEAR LINE NUMBER
1766 08                 INX
1767 6F 00              CLR     0,X
1769 08                 INX
176A 6F 00              CLR     0,X
176C 08                 INX
176D BD 17 C0   READ3   JSR     SINCH       GO GET CHARACTER
1770 81 0D              CMP A   #$D         IS IT CR?
1772 27 1B              BEQ     READ4
1774 81 1A              CMP A   #$1A        IS IT EOF?
1776 26 0E              BNE     READ35
1778 7D 00 20           TST     DSKFLG      DISK IO?
177B 27 17              BEQ     READ5
177D 7D 00 21           TST     DRWFLG      R OR W?
1780 26 12              BNE     READ5
1782 97 24              STA A   LASTIN      SET LAST IN
1784 20 0E              BRA     READ5
1786 81 1F      READ35  CMP A   #$1F        CONTROL CHARACTER?
1788 23 E3              BLS     READ3
178A A7 00              STA A   0,X         PUT CHARACTER
178C 08                 INX                 BUMP THE POINTER
178D 20 DE              BRA     READ3
178F A7 00      READ4   STA A   0,X         PUT CHARACTER
1791 08                 INX                 BUMP POINTER
1792 20 BD              BRA     READ2       REPEAT
1794 5F         READ5   CLR B
1795 BD 09 9F           JSR     BAKON2      MOVE BACK
1798 DF 9A      READ6   STX     FILEND      SET END
179A 7C 00 84           INC     CHKFLG      SET FLAG
179D 7D 00 20           TST     DSKFLG      DISK IO?
17A0 27 05              BEQ     READ65
17A2 BD 16 4F           JSR     CLSFCB      CLOSE FILE
17A5 20 07              BRA     READ7
17A7 96 B8      READ65  LDA A   TOFCH       TAPE OFF
17A9 BD 02 0F           JSR     TOUCH
17AC 01                 NOP
17AD 0E                 CLI
17AE 9C 40      READ7   CPX     TEMP        CHECK POINTER
17B0 27 05              BEQ     READ8
17B2 DE 40              LDX     TEMP        GET POINTER
17B4 BD 08 4D           JSR     RENUM2
17B7 7D 00 20   READ8   TST     DSKFLG      DISK IO?
17BA 27 01              BEQ     READ9
17BC 39                 RTS                 RETURN
17BD 7E 0B 5A   READ9   JMP     BOTTO1      RETURN

                * SPECIAL INPUT
```

```
17C0 7D 00 20  SINCH   TST     DSKFLG     DISK IO?
17C3 26 03             BNE     SINCH2
17C5 7E 02 0C          JMP     TINCH      TAPE INPUT
17C8 DF 25     SINCH2  STX     INDEX9     SAVE X REG
17CA 7D 00 21          TST     DRWFLG     CHECK FLAG
17CD 27 05             BEQ     SINCH4
17CF CE A8 40          LDX     #FCB       SET POINTER
17D2 20 03             BRA     SINCH5
17D4 CE 18 9D  SINCH4  LDX     #RFCB      SET FOR READ
17D7 BD B4 06  SINCH5  JSR     FMS        CALL FMS
17DA 27 0B             BEQ     SINCH7
17DC A6 01             LDA A   1,X        CHECK ERROR #
17DE 81 08             CMP A   #8         IS IT EOF?
17E0 27 03             BEQ     SINCH6
17E2 7E 16 34          JMP     SOUCH4
17E5 86 1A     SINCH6  LDA A   #$1A       SET EOF
17E7 7E 16 4C  SINCH7  JMP     SOUCH8     FINISH UP

               * NEW COMMAND

17EA BD 0B 6E  NEW     JSR     TFORCR
17ED DE 98             LDX     FILBEG     GET POINTER
17EF DF 58             STX     SPCPT1     SAVE IT
17F1 DE 48             LDX     CURPOS     GET CURRENT POS
17F3 DF 5A             STX     SPCPT2     SAVE IT
17F5 7C 00 20          INC     DSKFLG     SET FLAG
17F8 BD 15 C5          JSR     RECOR1     GO WRITE
17FB 7F 00 20          CLR     DSKFLG
17FE DE 98             LDX     FILBEG     POINT TO BEGIN
1800 DF 48             STX     CURPOS
1802 8D 03             BSR     RNEW       GO READ NEW
1804 7E 04 F0          JMP     PEDIT      RETURN

               * READ NEW TEXT FROM DISK

1807 DE 5A     RNEW    LDX     SPCPT2     GET POINTER
1809 86 01             LDA A   #1         SET FLAG
180B 97 20             STA A   DSKFLG
180D 4F                CLR A
180E 5F                CLR B
180F 9C 58     RNEW2   CPX     SPCPT1     CHECK LIMIT
1811 27 07             BEQ     RNEW4
1813 4C                INC A              BUMP COUNT
1814 26 01             BNE     RNEW3
1816 5C                INC B
1817 09        RNEW3   DEX                DEC THE POINTER
1818 20 F5             BRA     RNEW2      REPEAT
181A 97 8F     RNEW4   STA A   CHRCNT+1   SAVE COUNT
181C D7 8E             STA B   CHRCNT
181E BD 0F 17          JSR     DELCHR     DELETE BLOCK
1821 96 24             LDA A   LASTIN     FINISHED?
```

```
1823 26 09              BNE     RNEW5
1825 96 9A              LDA A   FILEND      GET POINTER
1827 91 29              CMP A   BUFEND      CHECK END
1829 24 03              BHS     RNEW5       END OF BUFFER?
182B BD 17 3E           JSR     READ1       GO READ IN
182E 7F 00 20  RNEW5    CLR     DSKFLG      CLEAR FLAG
1831 39                 RTS                 RETURN

                * STRINGS

1832 49         ILST    FCC     'ILLEGAL FILE NAME'
1843 04                 FCB     4
1844 46         FEST    FCC     'FILE EXISTS'
184F 04                 FCB     4
1850 44         DLST    FCC     'DELETE BACKUP FILE (Y-N)? '
186A 04                 FCB     4
186B 46         NMST    FCC     'FILE NAME? '
1876 04                 FCB     4
1877 4E         NFST    FCC     'NO SUCH FILE'
1883 04                 FCB     4
1884 57         LSST    FCC     'WARNING: LOW DISK SPACE!'
189C 04                 FCB     4

189D            RFCB    RMB     320
19DD            WFCB    RMB     321

1B1E            BEGPNT  EQU     *

                        END     START
```

NO ERROR(S) DETECTED

SYMBOL TABLE:

ACCT	0023	ADDBX	AD36	ALLFLG	0078	APPCOL	007B	APPE35	1507
APPE53	151C	APPE63	1533	APPE65	153B	APPE66	153F	APPE67	1549
APPE72	1558	APPE78	156A	APPEN0	14CE	APPEN1	14D1	APPEN2	14EA
APPEN3	14F4	APPEN4	150E	APPEN5	151B	APPEN6	1528	APPEN7	1552
APPEN8	157C	APPEN9	157E	APPEND	14C1	ASN	AC0C	BAKON2	099F
BAKON4	09B0	BAKON5	09B1	BAKON6	09B4	BAKONE	0999	BCDC15	090A
BCDC65	0930	BCDC67	0932	BCDC01	0900	BCDC02	090D	BCDC04	0912
BCDC05	0921	BCDC06	0925	BCDC07	0934	BCDC08	0943	BCDCON	08FE
BEGPNT	1B1E	BELL	0007	BFRSTR	0F88	BMPFLG	007E	BMPNU4	0811
BMPNUM	0807	BOTT01	0B5A	BOTT02	0B62	BOTTOM	0B58	BSE	AC07
BSP	AC00	BUFEND	0029	BUFFER	00BD	BUFLIM	0656	BUFPNT	0044
BUFSAV	0046	CFIN12	0BB4	CFIN13	0BC8	CFIN14	0BCE	CFIN15	0BD1
CFIND	0B9B	CFIND1	0BB0	CFIND2	0BD4	CFIND3	0BDC	CFIND4	0BE7
CFIND5	0BF1	CFIND6	0BF4	CFIND9	0BF7	CFNTST	0BFD	CHA510	1025
CHA675	1091	CHAN12	0FAF	CHAN15	0FB2	CHAN35	0FD2	CHAN37	0FD6
CHAN50	1015	CHAN51	1018	CHAN52	1033	CHAN55	103E	CHAN61	1050
CHAN62	105F	CHAN65	1065	CHAN66	106A	CHAN67	107E	CHAN80	10A7
CHAN81	10B1	CHAN82	10BA	CHAN84	10C4	CHAN86	10D0	CHANG1	0FAC
CHANG2	0FB9	CHANG3	0FC8	CHANG4	0FFC	CHANG5	1005	CHANG6	1045
CHANG8	109C	CHANG9	10D8	CHANGE	0F9F	CHAR	AC18	CHFRQU	1416
CHGEND	0066	CHGFLG	0086	CHGONF	007A	CHGPNT	0064	CHKFLG	0084
CHRCNT	008E	CHRTBL	141D	CLASS	08DC	CLASS2	08EB	CLASS3	08FB
CLASS4	08FD	CLRNUM	094C	CLSFC1	165E	CLSFC2	166A	CLSFCB	164F
CMPZ14	07EA	CMPZ24	07F5	CMPZN1	07E0	CMPZN2	07EB	CNRSTR	0B2B
COPY	11CA	COPY0	11F3	COPY02	11E7	COPY05	11EB	COPY1	11F6
COPY15	1213	COPY18	1219	COPY2	1225	COPY25	122A	COPY3	123E
COPY4	126B	COPY45	127D	COPY5	1280	CPYDRC	008C	CR	000D
CRGRET	000D	CURPOS	0048	DBFPNT	AC14	DCC	00B5	DECCNT	008A
DECNUM	09D3	DEDI35	03A8	DEDI37	03D1	DEDI42	0414	DEDI45	0417
DEDI47	041A	DEDI55	0437	DEDI61	0442	DEDI62	0447	DEDI67	047A
DEDI68	048F	DEDI85	04D2	DEDI88	04D8	DEDI95	04ED	DEDIT2	03A6
DEDIT4	03F9	DEDIT5	0420	DEDIT6	043C	DEDIT7	0499	DEDIT8	04A8
DEDIT9	04E7	DEL	AC01	DELAY	00B6	DELAY1	15F7	DELAY2	15FC
DELAY4	1602	DELC21	0F35	DELC31	0F38	DELC32	0F47	DELC34	0F4B
DELCH2	0F2F	DELCH3	0F36	DELCH4	0F50	DELCH5	0F52	DELCHR	0F17
DELE02	0E62	DELE04	0E6E	DELE15	0E9B	DELE25	0EB8	DELE35	0EC2
DELE45	0EE7	DELET0	0E8C	DELET1	0EA7	DELET2	0EB6	DELET3	0EBA
DELET4	0EC5	DELET5	0EEF	DELET6	0F02	DELET7	0F07	DELETE	0E58
DELIM	0096	DEXIT	0B34	DLST	1850	DMEND	AC2B	DN	0003
DPCRLF	AD24	DPSTRN	AD1E	DRCTN	008D	DRWFLG	0021	DSKFLG	0020
EDIT	04FA	EDIT1	0507	EDIT2	051B	EDIT25	052C	EDIT26	0538
EDIT3	0540	EDIT31	0543	EDIT4	0545	EDIT5	0550	EDIT55	0559
EDIT56	0579	EDIT58	0583	EDIT6	0588	EDIT63	0594	EDIT65	059B
EDIT7	05A4	EDIT8	05B7	EDIT85	05C3	EDIT88	05CD	EFER	0008
EL	0003	EOL	AC02	EQUALS	0C71	EQUFLG	007F	ERROR	05CF
ERRSTR	05DD	EXIT	0B32	EXIT2	0B37	EXIT5	0B4C	EXPAN1	144C
EXPAN2	1455	EXPAN5	1464	EXPAND	143E	EXPL35	1499	EXPLI1	1483
EXPLI2	1486	EXPLI3	1495	EXPLI4	149F	EXPLI5	14B3	EXPLI6	14BB
EXPLI7	14C0	EXPLIN	1467	FCB	A840	FEER	0003	FEST	1844
FILBEG	0098	FILEND	009A	FILL	00B3	FIN702	0747	FIN711	0751
FIND	0699	FIND1	06A6	FIND14	06BD	FIND16	06C2	FIND2	06CE

FIND3	06DD	FIND4	06E9	FIND5	06F6	FIND6	0701	FIND62	0716
FIND63	0720	FIND65	0728	FIND66	0730	FIND67	0744	FIND7	0745
FIND71	074E	FIND72	0754	FIND73	0762	FIND74	076C	FIND75	077B
FIND77	0790	FIND78	0791	FINDL	066E	FINDL0	067E	FINDL1	0680
FINDL2	0681	FINDT	0685	FINDT0	068A	FINDT1	0694	FINDT2	0697
FIXZON	07F6	FMS	B406	FMSCLS	B403	FNDCR2	0992	FNDCRT	098F
FNDFLG	0071	FNDN45	0981	FNDNU1	095A	FNDNU2	0961	FNDNU4	0963
FNDNU5	0983	FNDNUM	0954	FNONFL	0088	GAP	1603	GAP2	160F
GETCHR	AD15	GETFIL	AD2D	GETNUM	0835	HEAD42	131D	HEAD45	1329
HEAD55	1333	HEAD57	1349	HEAD58	134E	HEADE2	1301	HEADE3	1313
HEADE4	131B	HEADE5	132F	HEADE7	1352	HEADER	12F2	HEDCNT	0097
ILST	1832	INBUF	AD1B	INCAMT	007D	INCH	0206	INCH35	064E
INCHAR	061C	INCHR1	061F	INCHR2	063C	INCHR3	0641	INCHR4	0652
INCHR5	0655	INCNUM	0819	INDEX9	0025	INITLZ	0359	INLMFL	0080
INS710	0D5E	INS711	0D84	INS712	0D91	INS713	0D94	INSE42	0CC7
INSE43	0CD9	INSE45	0CE2	INSE51	0CFB	INSE52	0D0E	INSE55	0D14
INSE60	0D1E	INSE61	0D22	INSE62	0D3A	INSE71	0D4F	INSE72	0D97
INSE75	0DA5	INSER1	0CA3	INSER2	0CAA	INSER3	0CB2	INSER4	0CB6
INSER5	0CEC	INSER6	0D17	INSER7	0D4A	INSERT	0C94	INZFLG	0090
LASTIN	0024	LASTNO	005C	LF	000A	LINBUF	A080	LINFLG	0076
LINO	00B4	LSST	1884	LSTFLG	0089	LSTTRM	AC11	MAK222	0E0C
MAKS18	0DEF	MAKS21	0DF2	MAKS22	0E00	MAKS23	0E13	MAKS24	0E17
MAKS55	0E32	MAKSP1	0DE5	MAKSP2	0DF0	MAKSP3	0E1C	MAKSP4	0E28
MAKSP5	0E30	MAKSP6	0E44	MAKSPC	0DD7	MEMEND	00BB	MIKBUG	E0D0
MOVE	11AB	MOVFLG	0081	MSLFLG	006D	NEW	17EA	NEWPOS	004A
NEXT	0B98	NFER	0004	NFST	1877	NL	0008	NLDSTR	0F77
NMST	186B	NOCURL	0075	NOFSTR	05E7	NORMST	0E45	NOTFND	05E2
NTRCHS	0F53	NUMBER	0091	NUMFLG	006A	NUMSE2	0AD1	NUMSE4	0AD6
NUMSE6	0AD9	NUMSET	0AC6	NWFSTR	034F	NXTCH	AD27	NXTFLG	0077
NXTLI2	0867	NXTLIN	0860	NXTOC0	0C53	NXTOC1	0C62	NXTOC2	0C6B
NXTOC3	0C70	NXTOCR	0C4F	OCCUR3	0C33	OCCUR4	0C42	OCCUR5	0C45
OCCURR	0C14	OCRCNT	006F	OCRFLG	0079	OCRTMP	0068	OFF	0B01
ON	0AFF	ONOFF	0AE0	ONOFTB	0AEF	OUTB35	0A94	OUTB65	0AA7
OUTB75	0AAF	OUTB78	0AB1	OUTBC2	0A86	OUTBC3	0A8E	OUTBC4	0A96
OUTBC6	0AA2	OUTBC7	0AA9	OUTBC8	0AB4	OUTBCD	0A7D	OUTCH	0209
OUTDEC	AD39	OUTHEX	AD3C	OUTHL	0ABC	OUTHR	0AC0	OUTL15	0A65
OUTLI2	0A68	OUTLI4	0A74	OUTLIN	0A58	OUTSPC	0A76	OVER	065C
OVERLA	10F5	OVLPST	128B	OVRBEG	0073	OVREND	0074	OVRL11	111E
OVRL12	112A	OVRL16	112D	OVRL35	1148	OVRL41	1155	OVRL43	116D
OVRL45	116F	OVRL55	117D	OVRLA0	1110	OVRLA1	1119	OVRLA2	1133
OVRLA3	1143	OVRLA4	1153	OVRLA5	1174	OVRLA6	1185	OVRLA7	1199
OVRLA8	119E	OVRLST	11A1	PCRLF	05DF	PDATA1	0608	PEDIT	04F0
PREROR	05D2	PRIN12	0A35	PRINT	0A20	PRINT0	0A27	PRINT1	0A2E
PRINT5	0A46	PRINT6	0A4B	PRNFLG	008B	PROMPT	0023	PRVCHR	AC19
PSTRNG	0606	PSTZFL	006E	PUTCHR	AD18	PUTNUM	0828	RBFEND	002B
READ	172A	READ1	173E	READ2	1751	READ25	1760	READ27	1764
READ3	176D	READ35	1786	READ4	178F	READ5	1794	READ6	1798
READ65	17A7	READ7	17AE	READ8	17B7	READ9	17BD	RECO25	15CE
RECO45	15E5	RECOR1	15C5	RECOR2	15C7	RECOR3	15D9	RECOR4	15DE
RECOR5	15EA	RECOR6	15ED	RECOR7	15F0	RECORD	15B7	RENSTR	0DC1
RENTER	AD06	RENUM1	0847	RENUM2	084D	RENUM4	085D	RENUMB	0842
REPEAT	006C	REPFLG	0082	REPLAC	0E55	RESTBF	170B	RESTRT	0203
RETRNR	AC16	RFCB	189D	RNEW	1807	RNEW2	180F	RNEW3	1817
RNEW4	181A	RNEW5	182E	ROFCH	00BA	RONCH	00B9	ROVER	1716

ROVER2	1720	ROVER4	1726	RPTERR	AD3F	RSTSPT	10E8
SAVE	1581	SAVE4	158E	SAVEIT	0027	SAVOCR	0C46
SEOL	13BC	SET	13A3	SETC	13C9	SETC2	13E0
SETC5	1404	SETC6	140E	SETC8	1413	SETDE2	079C
SETDE5	07B3	SETDEL	0796	SETEXT	AD33	SFILL	13B7
SINCH2	17C8	SINCH4	17D4	SINCH5	17D7	SINCH6	17E5
SKIPS2	061B	SKIPSA	0614	SKIPSP	0615	SKPCLS	08D9
SNGLIN	0085	SOUCH	161B	SOUCH2	162C	SOUCH3	162F
SOUCH5	1644	SOUCH6	1649	SOUCH8	164C	SPC	0020
SPCPT2	005A	SRCHPT	004C	STAB	13B2	STACK	01FF
STRCN1	0072	STRCN2	0087	STRCNT	007C	STRGB1	0052
STRIN1	0877	STRIN2	087F	STRIN3	0885	STRIN4	08AA
STRIN6	08BD	STRIN7	08CA	STRIN8	08CC	STRIN9	08D6
STRNGB	004E	STRNGE	0050	STRPNT	0056	SVSTPT	10DB
SYNSTR	05F9	SYSFLG	AC0D	SZON75	139B	SZONE	1355
SZONE3	1370	SZONE4	1377	SZONE5	137B	SZONE6	138F
SZONE8	13A0	TAB	12A7	TAB2	12AC	TAB4	12C6
TAB6	12E7	TAB8	12EF	TABBUF	009E	TABCH	AC06
TABLE	0212	TABPNT	009C	TDELAY	15F5	TDST	16F6
TFORC2	0B7C	TFORC3	0B7D	TFORCR	0B6E	TINCH	020C
TOFCH	00B8	TOGGLE	0B04	TONCH	00B7	TOP	0B67
TORD33	16B7	TORD35	16BC	TORD37	16E5	TORDS2	1684
TORDS4	16EA	TORDS5	16EB	TORDS6	16EE	TORDSK	166D
TRGLIN	0094	TSTEMP	0A51	TSTEN2	0806	TSTEND	07FF
TSTMS4	0B92	TSTMS5	0B97	TSTMSL	0B80	TSTNU2	09E7
TSTOV2	09D2	TSTOVR	09CA	TTERM	0022	UPONE	09B8
UPONE2	09C2	VERFLG	006B	VERL12	09F5	VERL15	0A05
VERLI2	0A1B	VERLIN	09E8	VERSE2	0B17	VERSE4	0B1C
VERSET	0B0C	WARMS	AD03	WASN	AC0C	WFCB	19DD
WRITE2	15A9	WRITE4	15B3	WRITE5	15B5	XCNTRL	0B21
XXX1	002D	XXX2	002F	ZOKSTR	0C07	ZONBUF	0062
ZONE1	005E	ZONE2	0060	ZONE3	07D7		

SASN	AC0B		
SDCC	13C1		
SETC4	13EF		
SETDE4	07AA		
SINCH	17C0		
SINCH7	17E7		
SLINO	13C6		
SOUCH4	1634		
SPCPT1	0058		
START	0200		
STRGE1	0054		
STRIN5	08AE		
STRING	086A		
SYNERR	05F4		
SZONE2	1369		
SZONE7	1394		
TAB5	12D1		
TABEND	00B2		
TEMP	0040		
TMPCHR	0083		
TORD32	169E		
TORDS3	1692		
TOUCH	020F		
TSTMS2	0B87		
TSTNUM	09E1		
UPONE1	09BC		
VERLI1	09F2		
VERSE6	0B1F		
WRITE	1591		
XSAVE	0042		
ZONE	07B6		