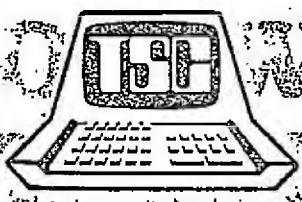


NPETE

RE SOFTWARE SOFTWARE
SOFTWARE SOFTWARE
WARE SOFTWARE SG
RES SOFTWARE TEXT
SOFTWARE Processing
FTWARE SYSTEM
RE SOFTWARE SC
SOFTWARE SOFTWARE
WARE SOFTWARE CO
ARE SOFTWARE SL68-29d SOFTWARE
SOFTWARE SOFTWARE
WARE SOFTWARE SO
RE SOFTWARE SCETI
SOFTWARE SOFTWARE
IT SOFTWARE SC



TECHNICAL SYSTEMS CONSULTANTS
SOFTWARE SOFTWARE

Contents

PREFACE	v
I. INTRODUCTION TO TEXT PROCESSING	1
II. COMMAND SUMMARY	13
III. REFERENCE MANUAL	17
IV. USING THE DISK VERSION	29
V. MACRO LIBRARY	33
VI. SYSTEM ADAPTIONS	41
VII. SOURCE LISTING	

Preface

The TSC Text Processing System is the most complex program, released by TSC to date. With this in mind, the following recommendations should be noted by the user.

Do not expect to master the system with one reading of the manual. The entire document should be read lightly the first time through, followed by a more rigorous reading. The "Reference Manual" section is very concise and contains detailed descriptions of all of the commands of the processor. This is the section which should be studied extensively.

Since the system is so complex, many results may occur which are contrary to the user's intentions. If strange output is encountered, reread the sections of the manual covering the commands being used. As more experience is gained, the system will become an invaluable tool, but as with any complex system, it takes time to learn its operation.

I. INTRODUCTION TO TEXT PROCESSING.

This world is producing millions of words of text each day. There are words in newspapers, magazines, books, catalogs, pamphlets, letters, documents, and manuals, and they all need to be organized before publication. It would certainly be a never ending task if all text formatting and organization were to be done manually. It simply would not get done. Thanks to computers and programs called text processors, text formatting (sometimes called word processing) becomes a fairly trivial task. The text processor allows for convenient and precise page formatting and organization. The final copy becomes extremely readable and neat, which are desirable features of any printed matter.

Just what can be done with text processors? The simplest functions perform exact page fitting. In other words, if the text page should have one inch margins with a page number centered at the bottom of each page, and perhaps a special title at the top of each, the processor will automatically provide these, given the appropriate commands. Line justification is another feature provided. Several types are available which include left-hand justification (left edge straight, right edge ragged), right-hand only justification (left ragged, right straight), left and right (both edges are straight), and center justification (both edges ragged but lines centered). An extensive text processor will provide features which will allow special operations such as footnote processing. The TSC Text Processing System supports all of the above features.

To gain some insight into the use of a text processor, several specific examples will be given using the TSC Text Processor's command set. The commands used by text processors vary from system to system but many are used in the same fashion. The TSC Text Processor uses an intermixed command and text method. To issue a particular command to the processor, it is necessary to start the command in column number one of a new line and begin the command with the control character, a period ('.'). This is the method used by most of the large scale system formatters including NROFF*, which is the system the TSC Text Processor has been modeled after.

Before any specific examples are shown, a description of the 'environment' will be given. The environment refers to the basic page and formatting features which will be in effect unless otherwise specified. The initial or default environment is very important. The TSC processor, without any command information, will perform left and right justification with a line length of 65 characters (the standard 8 1/2" page line width). Page length is initially defined to be 66 lines which is the standard for 11" paper and 6 lines/inch spacing. Other initial environment features provide for the passing of blank lines to output, and

*NROFF is a text formatting program written at Bell Laboratories. It runs on many large operating systems, including the UNIX Time Sharing System.

TSC Text Processor User's Manual
Version 2.3

for any line starting with a space or spaces to create a new line with the spaces now treated as unpaddable space characters*. With the environment initialized as above, it is possible to take any text file not having special command information embedded in it and still receive a sensibly formatted output. This is an important feature which is often overlooked by many processor designers. The environment may, of course, be changed or modified at any time by the use of special commands to allow for more personalized and detailed formatting jobs.

Let's take a look at some specific commands of the TSC processor. One of the simplest commands is the center lines command, .CE N, where N is the number of lines to be centered within the current line width. To use this command, as with any of the commands, it is only necessary to place the command right before the lines it is to affect. For example:

```
.CE 2
The Design of Text Processors
An Introduction
```

will cause the two lines listed to be neatly centered on the page. It can be seen that this is much easier than trying to manually calculate the correct spacing.

The initial environment was previously described. All of the parameters may be easily changed by the commands which directly affect them. One of the commands is .LN N and is used to set the current line length. To set the line length to 50, all that is necessary is a command line which reads as follows.

```
.LN 50
```

The length is now 50 columns. Another parameter easily set is the page length using the command .PL N, where N is the number of lines per page desired. Some other commands which change environment parameters include .FI and .NF which turn fill mode on and off (no fill) respectively. Fill means that as many words which will fit within the current line length are placed on each output line. This gives a straight left text edge and a slightly ragged right one. No fill simply copies the input lines directly to the output. It should be noted that 'fill' must be on for any justification to occur. The justification feature may be turned off using .NJ for 'no justification' or the type of justification may be set using .JU X. The X is the selection character and may be null, which turns justification on in the mode it was previously defined; it may be R for right hand, C for centered, or N for normal (left + right). Left justification is obtained by turning 'fill' on and justification off.

*Unpaddable spaces are characters which appear as spaces on the output but are not recognized as such by the processor. This means these spaces will not be 'spread out' by the justification routines.

Another environmental parameter is the capitalization mode. This is a special feature found only on the TSC Text Processing System and allows an upper case only terminal to be used for preparation of text which will later be output on a hardcopy device having lower case capabilities. The commands .CP and .NC turn this feature on and off respectively. If this mode is on, all letters are automatically converted to lower case unless preceded by a '@'. The '@' should be thought of as a typewriter shift key in its function. Another feature also enabled in this mode is similar to the 'shift and lock' on a typewriter. By typing a 'T' all characters following will be upper case until another 'T' is encountered.

It is often desirable, for readability, to use multiple spacing between lines. The TSC processor will allow this using the command .MS N where N is the space count desired and defaults to double spacing (N=2) if no value for N is given. The single space mode can be restored by either using .MS 1 or .SS for 'single space'.

Another group of commands deal with left margins and indentation. The left margin is normally set to 0 since the output device usually provides its own left margin (determined by paper positioning). Some applications require a wider margin at which time .LM N may be used to redefine it to be N spaces wide. Indent is similar to the left margin control with one difference. .LM N preserves the line length and simply moves the line to the right N spaces. .IN N, on the other hand, effectively reduces the line length by N columns in order to preserve the right hand margin. Setting the indent back to 0 will restore the full line length. Another form of indenting can be done by the use of the single indent command .SI N. Single indent is identical to indent except it is automatically restored to 0 after the line is output. It should be noted that the commands for left margin, indent, and single indent are additive in that if the following string of commands is issued:

```
.LM 10
.IN 8
.SI 5
```

the resultant output line would be preceded by 23 spaces, succeeding lines are preceded by 18 spaces assuming another .SI command was not used

A note of caution is necessary concerning a characteristic of several of the processor's commands. Most commands will perform only their specified function but some also cause a line 'break'. A break is the forcing of output of the line currently being collected in the line buffer. Normally a line is kept in the buffer until the specified line length has been reached, at which time justification may or may not occur, depending on the mode enabled (also assuming that 'fill' is turned on). The break will cause the partial line to be output without being filled,

TSC Text Processor User's Manual
Version 2.3

but the appropriate justification will be performed. This is useful for starting new paragraphs or new blocks of text. Some of the commands which cause a break are .CE, .FI, .NF, .IN, and .SI. Sometimes it is desirable that these commands do not cause a break. This can be done by using the 'no break' control character, ':'. So far, all commands have been shown preceeded by the normal control character, a period. To set an indent of 10 and not cause a break, the following should be used:

:IN 10

The colon may be used with any command, whether the command normally causes a break or not.

It is often necessary to produce a section of one or more blank lines. The space command, .SP N, can be used to output N blank lines. The space command also causes a break. If N is not specified, the processor will output 1 blank line. It may be required that the blank lines all be on the same page, maybe for later insertion of a photograph or illustration. The TSC Text Processor allows this by using the 'save space' command, .SV N, where N is the number of lines required. If there are not N lines remaining on the current page, no line is output but instead, printing continues and the count (N) is saved for later use. When the next page is reached, the 'output saved space' command may be used, .OS, which will then produce the remembered number of blank lines. A convenient method for using .OS will be given later. Another similar command is the 'need lines' command, .NL N, where N is a line count. This command says that there must be N lines remaining on the current page, and if there are not, eject to the next page. This is convenient for keeping special blocks of text together (keep them from being split by page boundaries), or for not starting a new paragraph at the bottom of a page if only 1 or 2 lines will fit.

The commands which have been described so far will allow very nice page formatting. If these were all that were available in a text processor, much time and effort would be saved. The TSC Text Processing System, however, offers many more commands and much more versatility. One feature often needed in documents or manuals is the page title. There are many different ways of providing titles but the TSC processor uses a title command which has the form:

.TL 'field1' 'field2' 'field3'

where field1 is left-adjusted, field2 is centered, and field3 is right-adjusted. Any one or all of the fields may be present. Another feature supported in the TSC processor is the ability to print the current page number in the text. Any place a percent sign (%) appears, it will be replaced by the page number. A few examples will clarify the use of the title command

```
. TL 'Main Title'
.TL ''Centered Title'Date'
.TL '--%--'
```

The first line will left adjust "Main Title" on the page. The second example causes "Centered Title" to be centered and "Date" to right adjusted. The final example will cause the current page number to be printed between two dashes.

Now it would be nice if there was some way of getting the title (and maybe a few other commands) to automatically execute at the top and/or bottom of each page of output. The TSC processor offers two advanced features to perform this task: macros and traps. A macro is a set of commands grouped together and given a name. When this name is later referenced, the entire group of commands will be executed. Essentially, what results is the ability to write special programs using the command set of the processor to do specific tasks such as headers, paragraphs, special titles, etc. The trap allows the user to specify a certain line on the output page where a specific macro is to be executed. To solve the title problem stated above it is convenient to define two macros, a header macro and a footer macro. The purpose of the header is to perform a sequence of commands to make the top of each new page appear the same. The footer macro works at the bottom of each page. Suppose it was required that the top of each page have three blank lines followed by a centered title and the bottom of each is to contain a centered page number between dashes. The following macros and trap placement would satisfy this requirement.

```
. DM HD
:SP 3
.TL ''Page Title''
:SP 3
..
.DM FT
:SP 3
.TL '--%--'
:PG
..
.AT 1 HD
.AT -? FT
```

The .DM command is used to define a macro and the first one listed in the example defines the header macro called HD. The header macro will space down 3 lines (without causing a break since the no break control character (';') was used), print a centered title, and finally print 3 more blank lines without causing a break. The last line of the header macro definition is '..' and is the command for terminating a macro definition. The second macro defined is FT and is used for the footer. Upon execution it will space down 3 lines (without a break), print a

TSC Text Processor User's Manual
Version 2.3

centered page number, and eject to the next page. The .AT commands were used to set the trap locations. .AT 1 HD causes the header macro to be executed at line 1 of every new page while .AT -7 FT causes the footer macro FT to be executed at the 7th line from the bottom of each page. The ability to specify trap locations and define macros makes titles and footers extremely simple and efficient.

One of the important aspects of using a text processor is the ability to make a few minor command changes and greatly change the final copy. As an example, suppose at the last minute it was decided that it would look better if there were four blank lines at the top of each page rather than three. If the document were being prepared by hand it would be necessary to retype the entire work to obtain the extra space. Using a small text processor it would only be necessary to go back and change the line count before each title. The TSC Text Processor and its ability to define macros means only one line in the entire text file needs to be changed. The second line of the header macro which is currently ':SP 3' would be changed to read ':SP 4'! One simple change and the desired result is obtained! It should be kept in mind that when preparing documents with a processor supporting macro capability, all of the often-used command strings should be defined in a macro so simple global changes may be easily performed if so desired.

There are more advanced features supported in the TSC Text Processing System. One of these is the ability to do conditional command execution. There are four forms of this command:

```
.IF O .XX
(IF E .XX
(IF N .XX
(IF !N .XX
```

where O and E stand for Odd and Even page number respectively, and N can be a number, a number register (to be explained shortly), or an expression containing numbers and number registers. The exclamation point is the 'NOT' operator and .XX is any command or macro name. The command works as follows; IF the condition is true (page is odd or even, or the number or expression is greater than zero) the command XX is executed, otherwise it is not. Preceding the expression by '!' will cause the command or macro to be executed only if the condition is not true (less than or equal to zero). The following special header macro definitions will illustrate the use of this command.

```
.DM HD
:SP 3
(IF O .TL "'Title'
(IF E .TL 'Title''
:SP 2
..
```

```
. DM HD
:SP 3
.IF %1 .TL "Title"
:SP 2
..
```

The first header defined causes the title to be right-adjusted on odd numbered pages and left-adjusted on even pages. The second definition will print a centered title on each page except page number one since the value of the expression will be zero when the page number is one (remember that the '%' represents the current page number).

Another feature contained in the TSC processor is the ability to use number registers. Two types exist, one which allows the user to read and access certain system parameters including the date, page number, current indent, left margin, current column position, current line on the page, and line length. The second type are user definable and can be used exactly as variables would be used in a program. Number registers are the single letters A-Z and the percent sign (%) already introduced. Several other number register features are supported by the TSC processor, including auto increment, assigning values to the registers, use in expressions (as seen in the .IF command), and the ability to print any register on the output in either Arabic, capital Roman, or small Roman numerals.

Some processors, including TSC's, allow communication between the processor and the operator during actual text processing. Three of these commands take on the following form:

```
.ST
.TM any string
.GI any string
```

The first command will stop the processing and print 'STOP' on the user's terminal. This may be desirable if special paper positioning is required or other special action is needed. When the processor has been stopped it may be restarted by typing any character on the terminal except an 'S' which will halt processing. The second command listed will send 'any string' to the terminal as a special message. It may be used before the 'STOP' command to issue special instructions to the operator. The last command will 'Get Input' from the terminal and insert it into the output stream. 'Any string' can be used for a prompt. An example where this command is quite useful is in the preparation of form letters. The processor may prompt the operator for names and addresses which are then typed in at the terminal and automatically inserted into the text!

One final command will be described in this introduction, the 'divert text' command. Sometimes it is desirable to save text currently encountered for later use. An example of this is when trying to do footnotes. It would be nice if immediately

TSC Text Processor User's Manual
Version 2.3

after the footnote reference was made, the actual footnote text could be typed, but saved for later insertion at the bottom of the page. The mechanism which allows this sort of operation is called a 'diversion' and is only available on the more complex text processors, such as TSC's. Two forms of the diversion usually exist:

```
.DI XX
.DA XX
```

where .DI instructs the processor to divert the following text into a diversion space named XX and .DA says to divert and append to the diversion space named XX. During diversion, all normal text processing still takes place, but rather than outputting the text to the printer, the text is written to a special place internal to the processor. The diversion process continues until the command for a divert is found without a name specifier. To recall the diverted text, it is only necessary to call it by name, exactly as macro calls are performed.

As an advanced exercise and demonstration of the diversion process (as well as many other processor commands) a complete set of macros for handling footnotes will be described. The reader should note that the following example is very complex and several readings will probably be required in order to fully understand its operation.

```
*          .NR B ?
.*         .DM HD
.*         :SP 2
.*         .IF %-1 .TL 'FOOTNOTE TEST'///
.*         :SP 2
.*         .AU 1
.*         .NR X 0
.*         .NR W 0-#B
.*         .IF #V .TR
.*         .NS
.*         ..
.*         .DM FO
.*         .NR V 0
.*         .IF #X .FT
.*         .CH FO -#B
.*         .PG
.*         ..
.*         .DM NM
.*         .TL '--%-'
.*         ..
.*         .DM BF
.*         .DA TX
.*         .EV 1
.*         .IF !#+X-1 .SA
.*         ..
```

- continued on next page -

```
. DM EF
. BR
. EV 0
. DI
. NR W -#Y
. CH FO #W
. IF #N-#P-#W . CH FO #N+1
.
. DM SA
-----
. BR
.
. DM TR
. BF
. NF
. FE
. FI
. EF
.
. DM FN
. DI FE
.
. DM FT
. EV 1
. NF
. TX
. RM TX
. DI
. FI
. EV 0
.
. AT 1 HD
. AT -#B FO
. AT -4 NM
. CH FO 70
. AT -#B FN .
. CH FO -#B
. EV 1
. RU 1
. LN 55
. EV 0
```

This example is quite similar to the one given in the "NROFF Users' Manual" written by J. Ossanna, of Bell Laboratories. To use these macros, merely insert their definitions at the beginning of the text file, and immediately after a footnote reference has been made, call macro BF. Following the call, simply type the footnote text and end it with a call to EF.

A description of the macros follows. The first line defines number register B and sets it equal to 7. Number register B is used to specify the size (in lines) of the bottom margin. A header macro definition follows (HD) and provides several functions. After spacing down two lines, the title is output

TSC Text Processor User's Manual
Version 2.3

unless it is page number one (the IF command). Two more lines are produced and the auto increment value is set to one. Number register X is cleared and it is later used to keep track of the number of footnotes on the current page. Next, W is set to the location of the bottom margin trap and will later be adjusted as necessary if footnotes are added. The IF #V command checks to see if there is any remaining footnote text from the previous page and if so they are reprocessed (number register V contains the line count of the last diversion). Finally, the 'no space' mode is turned on to suppress any spaces which might otherwise get printed needlessly at the top of the page.

The footer macro, FO, clears the diversion count, V, and checks the value of X. If X is not zero (meaning there were footnotes on the page), macro FT is invoked. The footer is then restored to its original location by using the Change command as defined by B. The last command does a page eject. Macro NM is used to print a centered page number at the bottom of each page.

The begin footnote macro, BF, starts with a divert append into the diversion space called TX. The environment* is switched, and if it is the first footnote on the page, macro SA is invoked which outputs a set of dashes as a simple footnote separator line. Diversion of the footnote text continues until macro EF is called. At this time a 'break' is executed and the original environment is restored. The diversion is stopped with the DI command. Number register W is updated by the number of diverted lines and the footer trap line is changed to compensate for the added footnote lines. Finally, if the number of diverted lines was great enough to move the footer trap up past the current line position, the trap is reset to the next line. TR is responsible for rediverting any lines of footnote text which will not fit on the page. It is very unusual for this to happen but this may occur if a footnote is very long and is referenced near the bottom of the page.

Macro FT is used for reading back the diverted text. It switches environments, sets the no fill mode, and calls TX, the actual footnote text. TX is then removed from the macro list, the fill mode is restored, and the environment switched. The last group of lines is used to define the trap locations of the various macros. The header is set to line one, and NM is set to execute four lines from the bottom of the page. The trap for the footer is planted at -#B, then moved past the bottom of the page while FN is also placed at -#B. FO is then moved back as originally placed so in effect both FO and FN are placed at the same line, but trap FN can only occur if the footer trap is moved up by the occurrence of a footnote. The last lines switch to environment one and initialize it for a line length of 55 and auto increment of one.

*Environment switching is a feature supported by many of the larger text processors (including TSC's) which allows all of the major environment parameters to change simultaneously.

As a final example of how a text processor can be used, a sample section of text will be given. The text is shown first with the commands and then as the text processor would output the final copy.

```
. CP
. SP 2
. CE 2
↑TEST OF SEVERAL↑
↑PROCESSOR COMMANDS↑
. SP
. SI 5
@THIS EXAMPLE SHOWS HOW COMMANDS AND TEXT CAN BE INTERMIXED
FOR LATER PROCESSING BY A TEXT PROCESSOR.
@THE EXAMPLE STARTED BY CENTERING TWO LINES FOLLOWED
BY A SINGLE INDENT TO SIGNIFY THE START OF A PARAGRAPH.
@THE CAPITALIZATION MODE IS ON AND THE UPPER CASE SHIFT
CHARACTERS ARE BEING USED.
. SP
. LM 10
. LN 45
. JU C
@THE ADJUST MODE WAS JUST CHANGED TO CENTERING
AS WELL AS A LINE LENGTH OF 45.
@THE LEFT MARGIN WAS SET TO 10 TO GIVE A NICELY
CENTERED NARROW LINE.
@SPECIAL EFFECTS LIKE THESE ARE EASILY ACCOMPLISHED.
. SP
. LM 0
. LN 65
. JU N
@THE PARAMETERS WERE JUST SWITCHED BACK SO THE
LINE APPEARANCE WILL BE RESTORED.
@THIS IS A SHORT EXAMPLE BUT SHOULD SHOW HOW THE
COMMANDS CAN BE INTEGRATED WITH THE TEXT.
```

This example appears in its expanded form on the next page.

This introduction to text processing is intended to be only that and is not a complete treatment of the subject. Many commands and features have been omitted. The ones included are the most general and the most used commands which offer the user a great deal of control and flexibility. Hopefully some eyes have been opened to the wide variety of applications of the text processor.

TSC Text Processor User's Manual
Version 2.3

EXPANDED EXAMPLE

TEST OF SEVERAL
PROCESSOR COMMANDS

This example shows how commands and text can be intermixed for later processing by a text processor. The example started by centering two lines followed by a single indent to signify the start of a paragraph. The capitalization mode is on and the upper case shift characters are being used.

The adjust mode was just changed to centering as well as a line length of 45. The left margin was set to 10 to give a nicely centered narrow line. Special effects like these are easily accomplished.

The parameters were just switched back so the line appearance will be restored. This is a short example but should show how the commands can be integrated with the text.

*NOTE: This entire user's manual was prepared using the TSC Text Editing System and the TSC Text Processing System.

Command Summary

Command Form	Initial Value	Default Argument	Cause Break*	Explanation
--------------	---------------	------------------	--------------	-------------

I. PAGE CONTROL

.PL +N	66 lines	66 lines	no	Page length.
.PG +N	N=1	-	yes	Eject to next page.
.PN +N	N=1	ignored	no	Page number.
.LM +N	N=0	previous	no	Left margin.
.NL N	-	N=1	no	Need N lines.

II. TEXT FILLING, ADJUSTING, AND CENTERING

.BR	-	-	yes	Break buffer.
.FI	fill	-	yes	Fill output lines.
.NF	fill	-	yes	No fill or justification.
.JU C	jst, norm	just.	no	Justify on.
.NJ	just,	-	no	No justification.
.CE +N	off	N=1	yes	Center N input lines.

III. VERTICAL SPACING

.MS N	prev	N=2	no	Multiple spacing.
.VS	single	-	no	Single space lines.
.SP N	-	N=1	yes	Space N lines.
.SV N	-	N=1	no	Save N lines.
.OS	-	-	no	Output saved lines.
.NS	space	-	no	No-space mode on.
.FS	-	-	no	Restore spacing.

IV. LINE LENGTH AND INDENTING

.LN +N	65	prev	no	Line length.
.IN +N	N=0	prev	yes	Indent.
.SI +N	-	N=1	yes	Single indent.
.PI ST	-	-	yes	Put string in indent.

V. MACROS, DIVERSIONS, AND LINE TRAPS

.DM XX	-	ignored	no	Define or redefine a macro.
.AM XX	-	ignored	no	Append to a macro.
.RM XX	-	ignored	no	Remove macro or diversion.
.DI XX	-	end	no	Divert out to macro "XX".
.DA XX	-	end	no	Divert and append to "XX".
.RT -N XX	-	-	no	Set trap at line N.
.CH -N -M	-	-	no	Change trap location. " " "
.CH XX -M	-	-	no	" " "
	-	-	no	End macro specification.

*The use of `:' as the control character (instead of `..') suppresses the break function.

TSC Text Processor User's Manual
Version 2.3

Command Form	Initial Value	Default Argument	Cause Break	Explanation
--------------	---------------	------------------	-------------	-------------

VI. NUMBER REGISTERS

. NR X +N	-	-	no	Number register.
. AU +N	0	prev	no	Set auto increment.
. AR	arabic	-	no	Arabic numbers.
. CR	arabic	-	no	Capital Roman numbers.
. SR	arabic	-	no	Small Roman numbers.

VII. TABS AND TAB CHARACTERS

. TA N...	none	none	no	Set tab columns.
. TF C	un. sp. *	un. sp. *	no	Set tab fill character.
. TC C	none	none	no	Set tab character.

VIII. THREE PART TITLES

. TL 'left'center'right'			no	Define title.
. LT +N	65	prev	no	Length of title.

IX. CONDITIONAL INPUT COMMANDS

. IF C COMMAND	-	-	no	If true, do command.
. IF !C COMMAND	-	-	no	"
. IF N COMMAND	-	-	no	"
. IF !N COMMAND	-	-	no	"

X. ENVIRONMENT SWITCHING

. EV N	N=0	N=0	no	Change environments.
--------	-----	-----	----	----------------------

XI. SPECIAL CONTROL COMMANDS

. CP	no caps	-	no	Capitals mode on.
. NC	no caps	-	no	No caps mode.
. ST	-	-	yes	Stop processing.
. EX	-	-	yes	Exit processor.
. PS	no pass	-	no	Pass text without proc.
. RP	-	-	yes	Repeat entire file.
. DH	-	-	yes	Double height line**.
. DW	-	-	yes	Double width line**.
. DB	-	-	yes	Double height and width**.

*Un. sp. = unpaddable space character.

**These commands require the output device
to support double dimensioned character printing

Command Form	Initial Value	Default Argument	Cause Break	Explanation
XII. EXTERNAL COMMUNICATION				
. TM ST	-	-	no	Send string to terminal.
. GI ST	-	-	no	Get line from terminal.
XIII. MISCELLANEOUS				
. *	-	-	no	Comment field.
XIV. UNDERLINE				
. UL	-	-	no	Underline next input line.
XV. DISK ORIENTED COMMANDS				
. IC C	">"	">"	no	Set item character.
. OF NAME	-	-	no	Open data file.
. CF	-	-	no	Close data file.
. RI S	-	-	no	Read item from file.
. NI N	-	N=1	no	Move to next item.
. NB N	-	N=1	no	Move to next block.

SPECIAL CHARACTER DEFINITIONS**Character Meaning**

- \ Standard escape character.
- @ Force capital letter.
- ↑ Set capital letter mode.
- # Number register specifier.
- : Basic control character.
- ⋮ No break control character.

TSC Text Processor User's Manual
Version 2.3

NUMBER REGISTERS

Register Meaning

A-B	User definable
C	Current column count
D	Day of the month
E	End of data file flag
F	User definable
G	.GI & .RI character count
H	User def.
I	Current indent
J-K	User def.
L	Current line length
M	Month
N	Line count on page
O	Current left margin
P	Current page length
Q-U	User def.
V	Last diversion line count
W-X	User def.
Y	Year (2 digits)
Z	User def.
Z	Page number.

Reference Manual

INTRODUCTION

All input lines to the processor which are to be interpreted as commands should be started with the control character (a '.', or ':') in column one followed immediately by the two letter command. If the characters are not system command names or user defined macros, the line will be ignored. The 'nobreak' control character ('::') may be used with any command to suppress normal line breakage during processing. Only a single command reference is permitted on any one line.

The following detailed command descriptions reference numerical arguments either as N, +N, or -N. N means any argument is taken as absolute and any previous value is simply replaced by the new value. +N is used when the argument may take any form of a number (either positive, negative, or absolute). Valid arguments of this form are +4, -10, and 3 where the old value would be incremented by 4, decremented by 10, and replaced by 3 respectively. Arguments of the form -N may use absolute values or negative values which are subtracted from the current page length (to reference N number of lines from the bottom of the page). When expressions are involved using the +N argument, the entire N is evaluated before the increment or decrement is applied (e.g. -6-3 will decrement the value by 3). Certain commands requiring arguments will keep the last argument assigned if the argument field is left empty when the command is called.

I. PAGE CONTROL

The page control commands are used to set the physical page parameters such as length, width, margins, numbering, etc. Top and bottom margins are not automatically provided and should be defined by the user with macros as described in a later section.

- .PL +N Set page length to N lines. Initial value is 66 lines and is reset to 66 if no argument is given. Does not cause a break. The maximum N is 255.
- .PG +N Eject to next page. If N is given the new page number will be adjusted accordingly. The page number is automatically incremented if no argument is given and the command does cause a break. Max N is 255.
- .PN +N Set the page number to +N. If .PN occurs before the first break or first text, it will be set for the first page. The value is initially 1 and the command does not cause a break. The maximum page number is 255.
- .LM +N Set the left margin according to +N. The entire output line will be offset to the right by the number of spaces the current LM is defined. Initially there is

TSC Text Processor User's Manual
Version 2.3

no margin ($N=0$) and no break occurs. Left margins should not exceed 100.

.NL N Need N lines on the page. If the distance to the next trap position or the bottom of the page is less than N, the paper is advanced to the next trap position (blank lines output). Otherwise no action takes place. No break occurs and the default argument is N=1.

II. TEXT FILLING, ADJUSTING, AND CENTERING

The following commands affect the appearance of individual lines of text. Two important parameters are referenced, Fill and Justify. The default fill mode is to fill output lines with as many words as possible without exceeding the set line length value. Any extra words are saved for output on the next line. A word is defined to be any string of characters separated by a space or spaces. If two words are to be separated by a space but are not to be split across line boundaries or separated by the justification routines, the unpaddable space character, "\ " (slash space) may be used. The default justification mode is left and right, giving straight margins on both sides. Filled lines which contain too few character positions to completely fill out the specified line length are padded with spaces until the correct length is achieved. The space filling or padding is done from alternate sides of the page as each line is justified to eliminate 'white rivers' which may otherwise occur in the text. No hyphenation is performed. It is important to note that fill must be on in order for the justification to be performed, but fill may be on by itself. If fill mode is off, characters are passed exactly as they appear on the input file.

.BR Break the line currently being filled in the buffer. The line is output after specified justification is done but no further filling or padding is attempted. Input lines beginning with spaces and empty text lines (blank lines) also cause a break.

.FI Fill mode is turned on and subsequent output lines are filled. This command causes a break.

.NF Turn off fill mode (nofill). Following input lines are neither filled or justified, but are copied to the output exactly as they appear on input, without regard to the current line length. Causes a line break.

.JU C Justification is enabled. If fill mode is off, adjusting will be deferred until it is back on. If the justify type character, "C", is present the justification type is set as follows: N sets for normal (default, left and right), R sets right only justify, and C will center lines (both margins ragged). If the type character is absent, justification is turned back on with the type previously used. No break is caused.

- . NJ Turn justification off. If fill is on, the resultant output line will have a straight left and a ragged right edge. No break is caused and the justify type remains unchanged.
- . CE +N Center the next N input lines. A break occurs before the command and then automatically after each line is output. If the resultant line is longer than the current line length, the output line will be left hand adjusted. The maximum count is 255.

III. VERTICAL SPACING

All line spacing defaults to standard single spacing. It may be set at any time by using the MS command. If the line spacing is N, N-1 blank lines are inserted after each output line. The occurrence of a trap will terminate any remaining spacing count. Contiguous space should be saved by using the SY and OS commands.

- . MS N Set multiple line spacing to N. N-1 blank lines are inserted after each output line. No break is caused and if N is not specified the value of 2 will be used (double spacing). Max value is 255.
- . SS Set single space mode. No blank lines are output after text lines and no break occurs.
- . SP N Space N lines. The number of output lines is limited to the distance to the nearest trap or bottom of the page. If nospace mode is on, no spaces are output. If no value for N is given, it defaults to 1. SP causes a break.
- . SV N Save N lines of space. If the distance to the next trap (or the bottom of the page) is greater than N, N lines are output, otherwise no lines are immediately output but the count (N) is saved for later output (see OS). Subsequent SY commands will overwrite any previously remembered N. Nospace mode has no effect. The command does not cause a break and the default value for N is 1.
- . OS Output saved space. This command is used to output any previously saved space from the SY request. The remembered count is cleared after calling OS and nospace mode has no effect. A break does not occur.
- . NS No-space mode is turned on. The no-space mode inhibits SP requests and PG requests without a next page number. This mode is automatically turned off after the output of a line of text. No break is caused.
- . RS Restore space mode. If the nospace mode is on, it is turned off with this command without causing a break.

TSC Text Processor User's Manual
Version 2.3

IV. LINE LENGTH AND INDENTING

Using the following set of commands, the user has complete control over the line length and various forms of indenting. The line length includes all indent spaces but does not include left margin spacing. As long as the fill mode is turned on, the resultant output line will be less than or equal to the current line length minus the indent. Line lengths of less than 6 columns are not permitted

- .LN +N Set line length. The initial value is 65 columns and the command does not cause a line break. Line lengths must be between 6 and 255 columns inclusive.
- .IN +N Set the line indent according to N. With a line length of L and an indent of N, N spaces are output before each line and the remaining text is restricted to a size of L-N. Initially the indent is 0 and the command causes a break.
- .SI +N Single indent N spaces. Only the next output line will be indented by the amount specified by N. Note that single indenting may be done backwards into an indent field. (e.g. if indent is 10, SI -4 would temporarily set the overall indent to 10-4 or 6). IN and SI counts are cumulative and the final value may not be negative! This command causes a line break.
- .PI ST Put string in indent field. The string represented by "ST" (leading spaces ignored), is inserted into the field normally filled with spaces by the indent count. If the string is longer than the indent count, the string will be truncated so it will not extend past the indent field.

V. MACROS, DIVERSIONS, AND LINE TRAPS

A macro is a set of commands and/or text which can be assigned a name and called by name at a later time. All macro names are two characters long and must be different from any names already in existence in the system command name table. Macros are defined or redefined by using the DM command or by using the output diversion command, DI. Macros already in existence may be appended to by using the AM or DA commands. If a macro is named XX, it may be invoked by an input line beginning with ".XX". A trap may also be placed at a specific vertical page placement to cause automatic macro execution at that point by using the AT command. During macro definition, number registers are not expanded into numeric values but are at the time the macro is executed. No other special character translation is done during macro definitions (e.g. tab expansion, etc.). Keep in mind that macros may be any combination of commands, macro calls, and text, but a macro may not define another macro (it may create a diversion).

A diversion is treated as a macro upon execution but is created in a different manner. Processed output may be diverted into a macro space for such purposes as footnote processing or vertical page size determination for conditional changing of page parameters (number register V contains the last diversion line count). All normal processing takes place during a diversion except left margins. It is standard practice to read back the diverted text in 'nofill' mode to suppress further line processing.

If at any time during macro definitions or diversion creation the macro space is overflowed, a system error will be generated and processing will be halted. None of the macro commands cause breaks in the line filling.

- . DM XX Define or redefine a macro with the character name XX. The actual macro begins with the next input line. The macro definition is copied until the termination character "... is found starting in column 1. Macros may not contain DM requests but may create diversions.
- . AM XX Append to the macro named XX. This command acts exactly like DM except the following input lines are appended to an existing macro rather than creating a new named space.
- . RM XX Remove macro or diversion. The macro named XX is removed from the name list and subsequent calls to this name will have no effect.
- . DI XX Divert output into the macro space named XX. The macro named XX is defined or redefined at this point. All normal text processing occurs during diversions except left margin page offsetting is not done. The diversion process is ended when another DI or DA is encountered. Diversions can not be nested! The count of the number of lines last diverted is kept in number register V for possible later reference.
- . DA XX Divert append version of DI. The same rules apply for both commands.
- . AT -N XX At line N invoke macro XX. Any macro previously planted at line -N is replaced by XX. N is measured from the top of the page (0 or 1 may be used to represent the top) and -N is measured from the bottom of the page (e.g. if the page length is 66, line -1 represents line 66). If no macro name is given with the command, the trap located at line -N, if any, is removed.
- . CH -N -M Change trap. See next.

TSC Text Processor User's Manual
Version 2.3

.CH XX -M Change the trap planted at line -N to occur instead at line -M. Alternately, change the location of the trap for macro XX to line -M. If there is not a trap set at -N, the request is ignored.

.. Terminate a macro definition.

VI. NUMBER REGISTERS

Number registers are a type of variable used during processing. There are two classifications, user definable and system. Number registers have single character names (A through Z and '%'). Number registers may be used any time a number is expected in a command and also may appear imbedded in text. There are two methods of referencing a number register:

#X
#+X

where '#' is the register designator character and X is the name of the register. When using '%' it should not be preceded by the '#'. The '+' in the second example specifies that the number register is to be auto incremented prior to its use and it will retain the new incremented value. The auto increment amount is set using the AU command. When a number register reference is encountered it is converted to decimal, lower case Roman, or upper case Roman, as determined by the mode set. Number registers appearing in macro definitions are not converted until the macro is actually executed. Number registers may also be used to construct expressions any time a number is expected in a command (expressions may not be imbedded in text). The expressions are evaluated left to right and may contain only the operators '+' and '-'.

.NR X +N Assign a value to number register X. This command should only be used to assign values to user definable number registers.

.AU +N Set the auto increment amount to +N. Any time a register is referenced as "#+X", the AU value will be added to it prior to its actual use.

.AR Arabic numbers. See below.

.CR Capital (upper case) Roman numbers. See below.

.SR Small (lower case) Roman numbers. Number registers will subsequently be converted into Arabic, capital Roman, or small Roman respectively. This mode is initially Arabic and also applies to the outputting of page numbers using the '%'

The following is a list of the system and user definable number register names.

Register Meaning

A-B	User definable
C	Current column count
D	Day of the month
E-F	User def.
G	Get input (.GI) character count
H	User def.
I	Current indent
J-K	User def.
L	Current line length
M	Month
N	Line count on page
O	Current left margin
P	Current page length
Q-U	User def.
V	Last diversion line count
W-X	User def.
Y	Year (2 digits)
Z	User def.
%	Page number

VII. TABS AND TAB CHARACTERS

The currently defined horizontal tab character is replaced by the required number of fill characters corresponding to the distance to the next defined tab stop column (on the line currently being filled). The fill character is normally the unpaddable space character but may be defined by using the TF command. Up to 20 tab stops may be defined and should be set in ascending order. Initially no tab stops are defined and the tab character is null. Any non alphanumeric character may be defined as the tab character. It should be noted that using tabs with the fill mode turned on can result in nonsensical output tab fields since the user may not know what the current output column is.

- .TA N... Tab stop settings. The default tab stops are all null (none) and a total of 20 may be defined. The stop values may be separated by spaces, commas, or any other nonnumerics, e.g. TA 10,20,25,40.
- .TF C Set the tab fill character. This is normally the unpaddable space character but may be defined to any nonnumeric printable character. If 'C' is not specified the fill defaults to the unpaddable space character.
- .TC C Define the tab character. Initially the tab character is null (none) but may be defined to any nonnumeric printable character. If 'C' is not specified the tab character again becomes null.

TSC Text Processor User's Manual
Version 2.3

VIII. THREE PART TITLES

Very convenient titling may be performed by using the TL command. Three fields may be used for left, centered, and right justification of titles. All 3 fields may be used or any combination of fields. The justification is done with respect to the title length which is independent of the defined line length. This length is initially 65 columns. The use of TL has no effect on current line accumulation (does not cause a break). .TL is usually used in header and footer macros. For example, .TL '~-%--' will print the page number in the center of the title length.

.TL 'LEFT'CENTER'RIGHT'

Place titles adjusted according to field. The strings represented by "LEFT", "CENTER", and "RIGHT" are respectively left adjusted, centered, and right adjusted within the current title length. Any of the fields may be empty and any nonnumeric printing character may be used in place of the field delimiter "/". The "%" character will be replaced by the current page number in Arabic or Roman representation.

.LT +N

Set title length. The lengths of titles and lines are separate parameters. Indents do not apply to titles but left margin adjustment does.

IX. CONDITIONAL INPUT COMMANDS

Input command and macro calls may be performed on a conditional bases. Chained conditionals are permitted as in: IF #A IF #B .XX.

.IF C COMMAND See next

.IF !C COMMAND "

.IF N COMMAND "

.IF !N COMMAND

IF is the conditional command. "COMMAND" can be any system command or macro name. "C" is a built in condition code and can be either O or E to represent Odd or Even page numbers respectively. "N" is any number and can be a number, a number register, or any combination of these in the form of an expression using addition and subtraction. If the condition is true (the built in condition is satisfied or the number is greater than zero), the command or macro named is executed, otherwise the command is ignored. If "C" or "N" are preceeded by a '!' (not), the command is executed if the condition is false or the number is less than or equal to zero.

X. ENVIRONMENT SWITCHING

There are a number of parameters which control the text processing and are grouped together and called the environment. These environment parameters may be changed all at once using the switch command. There are two environments, 0 and 1. They both have identical initial values for all parameters. Parameters within these environments are those associated with:

line length	vertical line spacing
indenting	centering count
adjusting	auto increment
filling	partially collected words
title length	partially collected lines

All other parameters are global, or in other words, they are not switched with the environment but remain unchanged. Examples of global values include left margin, page number, current line number, number registers, trap tables, and macro definitions. Since partially collected words and lines are kept with the environment, switching environments will not cause a break and will also preserve any left over words.

.EV N Change to environment N where N can be 0 or 1. If N is left null, environment 0 is assumed.

XI. SPECIAL CONTROL COMMANDS

The following commands control certain aspects of the processor. The double height and width commands are hardware dependent. You should refer to the "adaption" section of this manual for details.

- .CP Turn capital letter mode on. When enabled, this mode will allow the use of an upper case only terminal to prepare text for later output to a device which supports both upper and lower case. Each character is automatically converted to lower case unless it is immediately preceded by a '@' at which time that character remains upper case. Strings of characters may be kept in upper case by enclosing them between up arrows "↑". The "@" is like a shift key and the "↑" acts like a shift and lock key.
- .NC Turn off capitals mode. Initially this mode is off and the special capitalization characters ("@" and "↑") are ignored.
- .ST Stop causes processing to temporarily halt and the word "STOP" is output to the terminal. At this time, typing an "S" will cause all processing to be stopped and the processor will be exited. Typing any other character will cause processing to continue. The stop command does cause a line break.

TSC Text Processor User's Manual
Version 2.3

- . EX Exit the processor. Text processing is stopped just as if all input had been finished. This command is useful in conjunction with the IF command.
- . PS Pass all input to the output. This command is primarily intended as a debugging aid since it allows all input (including command lines) to be passed to the output. No command interpretation or processing is done and once in this mode, the remaining text will be passed until the end of the input file is reached.
- . RP Repeat processing on file. This command will cause the file to be 'rewound' and all processing to be repeated. This is useful for some form letter type applications.
- . DH Print the next line in double height characters. This feature requires special hardware on the output device. Consult "Adaptions" for details.
- . DW Print the next line in double width characters. Requires special hardware.
- . DB Print next line in both double height and double width characters. Requires special hardware.

XII. EXTERNAL COMMUNICATION

Two commands exist which allow for communication between the processor and the user during actual text processing. The TM command is useful for sending special instructions to the terminal such as paper adjustment or character font change information. The GI command can be used in form letter preparation or insertion of special text strings while processing is taking place.

- . TM ST Send a message to the terminal. ST may be any string of characters or words. The leading blanks are ignored. The message is simply output to the terminal and may be used before the Stop command to issue special instructions.
- . GI ST Get input from the terminal. If ST is present (any string), it is output to the terminal as a prompt message. Characters typed from the terminal following the execution of GI are automatically inserted into the input stream for text processing. This command can be used to get name and address information for form letter preparation. The 'get input' function is terminated by typing a carriage return, therefore, only one line of text may be entered with each GI command executed. After completion of the command, the number register G contains the character count of the string typed (not including the carriage return).

XIII. MISCELLANEOUS

The following describe some of the smaller features of the text processor.

- .* Comment field. This may be used to insert comments into the input text and will be ignored by the processor. No output is created with this command (the comment is not passed to the output).

Special Characters

- \ Standard escape character. This character is used to remove special meaning from a character. For example, if a percent sign ("%) is needed in the output it is necessary to precede it with the "\", otherwise it will be interpreted as the page number (e.g. \%). To print a backslash, "\\" must be used.
- @ Force upper case letter if in the capitals mode (CP). This acts similar to the 'shift' key on a typewriter. Example: "@test" will be output with an upper case "T" and lower case "est".
- ↑ Upper case string delimiter. This character acts similar to the 'shift and lock' key on a typewriter. As an example, ↑this is a test↑ would cause "this is a test" to be output in all upper case characters. The capitals mode must be on (CP).
- # Number register specifier. When an alphabetic character is immediately preceded by a "#" it will be interpreted as a number register. Example: "#A" refers to number register "A".
- .
- :
- %

The period is the basic command control character. If in column one, it specifies a two character command or macro name follows.

The colon is the no-break control character. It functions exactly like the period, but will suppress breaks caused by various commands.

Page number symbol. Any place the percent sign appears, it will automatically be replaced by the current page number.

Special notes

- A. Any time input is being typed into the processor, typing a 'control X' will delete that line and issue a "?" as a prompt.
- B. The processor automatically makes sure there are two spaces after ".", "!", or "?". This does not apply to punctuation immediately followed by another character.

FLEX Version Features

I. BRINGING UP THE SYSTEM

The disk processor command file name is "PR.CMD". The general syntax for the PR command is:

```
PR,<file spec>[,<list of file specs>]
```

The <file spec> designates which text file is to be processed, and defaults to a .TXT extension. If the text to be processed is divided among several files, each file spec may be listed separately on the calling line separated by commas. A special feature supported by PR is the ability to process files from any number of discs on systems containing a limited number of drives. Substituting a '*' for the <file spec> anywhere on the calling line where a <file spec> is expected will cause the processor to halt and output to the terminal:

CHANGE DISKS AND TYPE A KEY

At this time, insert the disk containing the continuation file(s) and type any key to restart processing. It should be noted that the ability to process multiple files with one calling line should only be used when the files are actual continuations of the same text. The processor treats them as if they were all part of the same file, continuing page numbers, indenting, page width, etc., just as if the first file had never ended.

One other feature supported by the processor is the ability to automatically process a macro definition file prior to processing any of the files specified. Upon the execution of PR, the 'working' drive disk is searched for a file named 'MACRO.TXT'. If none is found, the processor starts processing the first file specified. If a MACRO file is present, it is read in and processed, just as if it had been the first file specified in the calling line. This is useful for defining all often used macros in this file so it is not necessary to redefine them in each processor text file prepared.

A few examples will clarify the calling of PR:

```
PR,CHAPTER1  
PR,0.CHPTR1,1.CHPTR2,*,0.CHPTR3
```

The first example will process the file named CHAPTER1.TXT. The file MACRO.TXT will also be processed if it exists. The second example will first try to process the file MACRO.TXT, then process the files CHPTR1.TXT on drive 0 and CHPTR2.TXT on drive 1. The processor will then halt and output the 'CHANGE DISK' message to the terminal because of the '*' used as a file spec. After changing disks in drive 0 and typing a key, the processor will process the file named CHPTR3.TXT on drive 0.

TSC Text Processor User's Manual
Version 2.3

When the processor is called, the following message will be output to the terminal:

PAGE LIMITS?

and is used to specify a particular block of pages to be processed. Typing a carriage return will cause all pages to be processed and output. Typing two numbers separated by a space or a comma will cause only the pages between those numbers (inclusive) to be output. For example, typing:

10,16

will result in only pages numbered 10 through 16 to be output. If just one number is entered, the processor will start outputting at that page number and continue to the end of the file. It should be noted that the processor always starts numbering the first page as number one unless instructed otherwise. As the processor is working, it may be stopped at any time by typing a "control C" on the terminal. (This feature is only supported on computers using a serial type interface (MP-S) as the terminal interface port.) The processor will respond with:

.. BREAK ..

output to the terminal. At this time processing may be continued by typing any character except an "S" which will cause the processor to be exited.

II. GENERAL USE

There are several things to keep in mind while preparing text for the text processor. Remember that all commands must begin in column one. It is usually most convenient to begin each sentence on a new line for easy future editing. Macros should be used as often as possible. The reason for this is to keep global changes as simple as possible, e.g. change only one line in a macro as opposed to changing single commands scattered throughout the file. It is not necessary to understand how the macros provided in this manual work in order to use them. All that is necessary is to know how to use them which is thoroughly explained. As experience is gained with the processor, you will be able to create your own special purpose macros for easy formatting.

TSC Text Processor User's Manual
Version 2.3

III. DISK ORIENTED COMMANDS

Several additional commands exist in the disk version of the text processor. All of these commands deal with the use of a "data file". The data file is a set of "blocks", with each block being divided into "items". An item can be any set of text or processor commands followed by an "end of item" character. The "end of item" character is initially a '>' but may be redefined using the .IC command (see below). The end of a block is specified by a null or empty item (two successive end of item characters form a null item; e.g. End of block>>) There are processor commands which allow inserting items into text (see .RI), skipping items (see .NI), moving to a new block (.NB), and the ability to open and close data files. For a specific example of these commands, see the Form Letter example in the MACRO LIBRARY section.

- .IC C Set the end of item character. This character is initially a '>' but may be defined to any nonalphanumeric printable character. If 'C' is not specified, the character defaults back to a '>'.
- .OF NAME Open a data file. This command will prepare the specified file for reading. If 'NAME' is specified on the command line (it should follow standard file spec rules) that file will be opened if found on the disk. If 'NAME' is not specified on the command line, the processor will prompt the terminal with: "DATA FILE NAME?" at which time the desired file name should be entered. The default extension is .TXT. If a file is already open, the .OF command will be ignored by the processor. It is only possible to have one file open at any one time. Closing a file using .CF will allow another file to then be opened.
- .CF Close data file. If a data file is opened, it will be closed and not allow any more data to be read from it. If no file is open, the command has no affect.
- .RI Read item from input file. If a file has been opened, the RI command will cause input from the file until an "end of item" character is read. The end of item character will be returned as a space if in the fill mode, or a carriage return if in nofill mode. If an S appears on the calling command line (.RI S), no character will be returned for the end of item character. In other words, the character will be 'S'uppressed. If there are no items remaining in the current block, .RI will have no affect. The RI command will also be ignored if no file has been opened. After reading data with the RI command, number register G will contain a count of the number of characters just read in

TSC Text Processor User's Manual
Version 2.3

- .NI N Move to next item. Normally sequential items are read by using the RI command. It is often desirable to skip items while processing text from a data file. The .NI command is used to skip one or more items in a block. If N is present on the calling line, it should be a number (or number register) which specifies the number of items to be skipped. If N is not present, the default is one item to be skipped. NI will not move past the end of a block.
- .NB N Move to next block. The use of NI and RI commands cause the sequential reading of items and will never move into the next block. It is necessary to use the .NB command to advance to the next block. If N is specified (a number or number register), N-1 blocks will be skipped. (example: If .NB 2 were specified, the next block would be skipped over and the next data read would be from the block following). If N is not specified, it defaults to 1. If there are no more blocks left in the data file and the .NB command is used, number register E will be set to one to designate an End of file condition.

IV. ADDITIONAL COMMANDS AND FEATURES

The disk version of the processor contains one additional command. This command permits underlining of words but may only be used with printer devices which support single character backspace capability. Unpredictable results will occur when trying to use this command on printers not supporting backspace.

- .UL Underline the next input line. The following line of text (single or multiple words) will result in the output being underlined. Only alphanumeric characters are underlined.

Macro Library

The following macro descriptions range from simple header and footer macros to a very complex footnote macro. It is not necessary to understand how the macros work, just how to use them. Each macro includes a description of what it does and how it can be used.

I. HEADERS AND FOOTERS

These macros are used to define top and bottom margins and also specify the contents of these margins, such as page numbers, titles, etc. Almost all processing jobs will require some sort of header and footer. Usually the macro definitions are placed at the beginning of the file (they need to appear before they are called for execution). The "AT" command is used to set the trap location (the line at which the macro should automatically execute) of each of the macros. Headers are set to line 1 and footers to a specific distance from the bottom of the page. Once these macros have been defined and their trap locations set, they can be forgotten about since the processor will do all the rest of the work. The first macro is a simple header macro which provides two blank lines, a centered title, and two more blank lines at the top of each page.

```
. DM HD
:SP 2
.TL "CENTERED TITLE"
:SP 2
.NS
.OS
..
.AT 1 HD
```

All of the header macros will contain a NS and OS command. NS will suppress any unnecessary spacing which may occur due to the unpredicted appearance of a SP command. For example, if the start of a new paragraph just happens to start at the top of a new page, there is no reason for the paragraph macro to space down two lines, since we are at the top of the page. NS will keep this from happening. The OS command instructs the processor to output any 'saved space' from the previous page. The next header is a little fancier. It does everything the previous one does except the titling is done a little differently. Here, if the current page number is even, the title is left hand justified. If the page is odd, the title is right hand adjusted.

Version 2.3

```
. DM HD
:SP 2
.IF E .TL 'EVEN TITLE'
.IF O .TL 'ODD TITLE'
:SP 2
.NS
.OS
..
.RT 1 HD
```

Subtitles may be used by simply placing a second TL command which contains the subtitle. The last header example is for those using a printer which uses separate sheets of paper (as opposed to continuous fed). This macro will issue a message to the terminal which instructs the operator to insert a new sheet of paper, before each page of text is processed. The paper should be set up such that the first line printed will be the top line of the paper. The operator will have to type a character on the terminal after each stop to restart the processor. Remember that typing an "S" will halt the processor.

```
. DM HD
.TM INSERT NEW SHEET
:ST
:SP 2
.TL 'TITLE'
:SP 2
.NS
.OS
..
.RT 1 HD
```

Footer macros are similar to headers except they are set to execute at the bottom of a page. For example, specifying AT -6 F0 would cause the macro called F0 to automatically execute at the 6th line from the bottom of the page. The first footer gives a five line bottom margin with the page number between 2 dashes centered on the page, 3 lines from the bottom.

```
. DM F0
:SP 2
.TL '--%--'
:PG
..
.RT -5 F0
```

It is often desirable to have page numbers on every page except page number one. The following footer will do exactly that.

TSC Text Processor User's Manual
Version 2.3

```
. DM FO
:SP 2
. IF %-1 . TL /*-%-*/
:PG
..
. RT -5 FO
```

There are several other types of header and footer macros which can be created. Some of these appear in the macros which follow.

II. PARAGRAPHS AND HEADINGS

There are many forms of paragraphing. The TSC Text Processor does not restrict one to using one particular form. One type of paragraph is to produce one blank line and start the first line of the paragraph indented five spaces. The following macro does just that:

```
. DM PP
. SP
. SI 5
..
```

To use the paragraph macro, simply call it by name any time a new paragraph is desired (e.g. type ".PP" in column one). One little feature which may be added to the macro is a need lines command, NL. In the following example, NL 3 is used to tell the processor that we desire at least three lines be left on the page before a new paragraph is started. This will keep one or two lone lines from being placed at the bottom of the page.

```
. DM PP
. SP
. NL 3
. SI 5
..
```

Many other types of paragraph macros may be created along the same lines as those presented.

Another useful macro can be created for major heading creation. One type of major heading might have a centered title spaced two lines down from the last line of text. The macro to accomplish this may look as follows:

```
. DM MH
. SP 2
. CE
..
```

To use this macro, type " MH" when the heading is desired. The next line should contain the heading title. For example:

TSC Text Processor User's Manual
Version 2.3

```
Line of text.  
.MH  
Heading Title
```

The last two macro examples are quite simple, but show how even two or three lines of commands may be replaced by a single macro call. This is quite useful if these operations are going to be repeated many times throughout a document.

III. FOOTNOTES

The following set of macros is all that is required to do very efficient and easy footnote handling. A description of how they actually work is contained in the introduction of this manual. To use these macros, it is only necessary to include their descriptions at the beginning of your file. As soon after a footnote is referenced in the text, call the macro BF (begin footnote) to begin the footnote. Immediately following this call, type the contents of the footnote, followed by a call to the macro EF (end footnote). The following serves as an example:

```
Text here referencing a footnote*.  
.BF  
*Footnote contents typed here and  
may be several lines long.  
.EF
```

It should be noted that the footnote macros contain their own header and footer macros which may be modified as desired. These macros should be the first lines of a file.

```
. NR B 7  
. DM HD  
:SP 2  
. IF %-1 . TL 'FOOTNOTE TEST'//  
:SP 2  
. AU 1  
. NR X 0  

```

TSC Text Processor User's Manual
Version 2.3

```
. DM BF
.DA TX
.EV 1
.IF !#+X-1 . SR
.
.DM EF
.BR
.EV 0
.DI
.NR W -#V
.CH FO #W
.IF #N-#P-#W . CH FO #N+1
.
.DM SA
-----
.BR
.
.DM TR
.BF
.NF
.FE
.FI
.EF
.
.DM FN
.DI FE
.
.DM FT
.EV 1
.NF
.TX
.RM TX
.DI
.FI
.EV 0
.
.AT 1 HD
.AT -#B FO
.AT -4 NM
.CH FO 70
.AT -#B FN
.CH FO -#B
.EV 1
.AU 1
.LN 55
.EV 0
```

Please remember that it is not necessary to fully understand how these macros work as long as you know how to use them.

TSC Text Processor User's Manual
Version 2.3

IV. TWO COLUMN OUTPUT

The TSC processor does not support backward line feeds so it is necessary to use some operator intervention in order to produce two column output. The following set of macros will produce two column output, each column being 31 characters wide. When the text of the first column reaches the bottom of the page, the string "REPOSITION PAPER" will be output to the terminal and a "STOP" command is executed. At this time the operator should reposition the paper to the top of the page and then restart the processor by typing any key but "S".

```
.LN 31
.NR A 0
.DM HD
.IF #A .PA
:SP 2
.AU 1
.IF !#+A-1 .TL "'title'"
.IF #A-1 :SP
:SP 2
.IF #A-1 .LM 34
..
.DM FO
:SP 2
.LM 0
.IF #A-1 .TL "'-%-'"
.IF #A-1 .NR A 0
:PG
..
.DM PA
.TM REPOSITION PAPER
:ST
.PN %-1
..
.AT 1 HD
.AT -5 FO
.BR
```

It should be noted that these macros also contain their own special set of header and footer macros which may be modified as desired.

V. FORM LETTERS

The last set of macros and examples deal with form letters. These macros are shown with some sample text and make extensive use of disk data files. This example should be thoroughly studied before trying to make use of disk data file commands. The RP (repeat) command is used so that the file is repeated over and over, until the end of file has been reached in the data file (number register E is non zero). The macro creates a name and address header at the top of each page. Following is "Dear (persons name)" and the text of the letter. The sample program is shown below, followed by the sample data file, and then a sample of the output produced by the processor.

```
.OF
.JU N
.NF
.DI NM
.RI
.BR
.DI
.IF #E .EX
.SP 6
.NM
.RI
.RI
.SP 3
.FI
Dear
.NM
.SP
.SI 5
We are writing to you to inform you that your
.RI
Insurance policy is about to expire.
Your policy number is
.RI
and expires on
.RI S
\
If you desire renewal, please send payment by
the end of this month.
If payment is not received, your policy will be terminated.
.RI
Thank you for your attention to this matter.
.SP 2
.NF
Thank you
.SP 3
Agent
NB
.RP
```

TSC Text Processor User's Manual
Version 2.3

The sample data file is as follows:

```
John Doe>
1313 Riverside Ave.>
Akron, Ohio 44225>
Fire>
F3-4322-946>
March 15, 1975>>
Bill Jones>
1111 Crescent Street
Apartment #12>
Kingston, New York 10011>
Automobile>E5-4936-263>March 14, 1975>
This is your second and final notice!>>
Hiram Johnson>
RR #3>
Lotson, Virginia 32004>
Life>
B1-2234-123>
March 12, 1975>>
```

As can be seen in the above sample data file, items may be placed one per line, or multiples per line as desired. The following is the output obtained from the first block of the data file.

John Doe
1313 Riverside Ave
Akron, Ohio 44225

Dear John Doe

We are writing to you to inform you that your Fire Insurance policy is about to expire. Your policy number is F3-4322-946 and expires on March 15, 1975. If you desire renewal, please send payment by the end of this month. If payment is not received, your policy will be terminated. Thank you for your attention to this matter.

Thank you,

Agent

System Adoptions

There are really only two features which can be user adjusted. One of these is macro storage space, the other being the double height and width character feature. These are treated separately below.

I. MACRO STORAGE SPACE

The macro storage space is presently set to approximately 3.5K and resides at the top of the first 12K block of memory. In 99% of all applications, this space will be much more than sufficient. If more memory is available, and you are requiring more macro space, the size of this space can be expanded. The end of the space is referenced at location \$02A8 (LMACRO) and may be changed as needed.

II. DOUBLE CHARACTERS

Three commands exist in the processor which require special printer hardware. These are double height (DH), double width (DW), and double both (DB). Some commercially available printers will print several lines of double size characters if a special control character is received prior to the line. The double height control character (\$12) is referenced at location \$0C4B. The double width control character (\$0E) is referenced at locations \$0C56 and \$0C5F. These may be changed as required.

```

*
*
*   TSC 6800 TEXT PROCESSING SYSTEM
*   FOR THE FLEX DISK OPERATING SYSTEM
*
*
*   COPYRIGHT 1978 BY
*
*   TECHNICAL SYSTEMS CONSULTANTS, INC.
*   BOX 2574
*   WEST LAFAYETTE, IN 47906
*

```

0030 ORG \$0030

* TEMPORARY STORAGE

* NUMBER REGISTERS

0030	NMREGS	RMB	2	A-B
0032	COLCNT	RMB	1	C
0033	DAY	RMB	1	D
0034	EOF	RMB	1	E
0035		RMB	1	F
0036	GCNT	RMB	1	G
0037		RMB	1	H
0038	IND	RMB	1	I
0039		RMB	2	J-K
003B	LLN	RMB	1	L
003C	MNTH	RMB	1	M
003D	LINCNT	RMB	1	N
003E	LFM	RMB	1	O
003F	PGL	RMB	1	P
0040		RMB	5	Q-U
0045	LDIV	RMB	1	V
0046		RMB	2	W-X
0048	YEAR	RMB	1	Y
0049		RMB	1	Z

* SPECIAL DISK STORAGE

004A	EOIF	RMB	1
004B	EORF	RMB	1
004C	ITEM	RMB	1
004D	FILOPN	RMB	1

* SINGLE STORAGE

004E	ULFLG	RMB	1
004F	GDNUM	RMB	1
0050	ADD	RMB	1

0051	SUB	RMB	1
0052	BNUM	RMB	1
0053	NPGN	RMB	1
0054	INC	RMB	1

* MACRO SAVE BLOCK

0055	NUMPNT	RMB	2
0057	EXCHR	RMB	1
0058	LSTNUM	RMB	2
005A	CMFLG	RMB	1
005B	MBFLG	RMB	1
005C	MBFPNT	RMB	2
005E	NOCR	RMB	1
005F	DONE	RMB	1
0060	FLBF	RMB	1
0061	RTFLG	RMB	1

* MORE SINGLE STORAGE

0062	LEFT	RMB	1
0063	TFILF	RMB	1
0064	NOFL	RMB	1
0065	INNUM	RMB	1
0066	NEG	RMB	1
0067	SIGN	RMB	1
0068	NSP	RMB	1
0069	PGN	RMB	1
006A	PASCHR	RMB	1
006B	SPSPF	RMB	1
006C	DOCAP	RMB	1
006D	DOCIM	RMB	1
006E	NOOUT	RMB	1
006F	TOUTL	RMB	1
0070	PTFL	RMB	1
0071	SIN	RMB	1
0072	MINDIS	RMB	1
0073	EV	RMB	1
0074	NOEXP	RMB	1
0075	NXTTAB	RMB	2
0077	TABFLG	RMB	1
0078	COLCN2	RMB	1
0079	IND2	RMB	1
007A	NXTTRP	RMB	2
007C	SVDSPC	RMB	1
007D	FINMAC	RMB	1
007E	NEGT	RMB	1
007F	IFFLG	RMB	1
0080	MACCNT	RMB	1
0081	PASFLG	RMB	1
0082	NONUMS	RMB	1
0083	DWFLG	RMB	1
0084	DFMFLG	RMB	1
0085	SPIFLG	RMB	1

0086	DIVFLG	RMB	1
0087	DIVFL2	RMB	1
0088	RIFLG	RMB	1
0089	CRSUP	RMB	1
008A	NCOUNT	RMB	1
008B	PSCNT	RMB	1
008C	INFCB	RMB	2
008E	XTEMP2	RMB	2
0090	PRNTR	RMB	1
0091	TLPP	RMB	1
0092	LOWPG	RMB	1
0093	HIPG	RMB	1
0094	FSTRAM	RMB	2
0096	LSTRAM	RMB	2
0098	NXTRAM	RMB	2
009A	JNKCNT	RMB	1
009B	SBFLG	RMB	1
009C	LLN2	RMB	1
009D	MACNAM	RMB	2
009F	MACTMP	RMB	2
00A1	LSTAVL	RMB	2
00A3	FSTRVL	RMB	2
00A5	STPOUT	RMB	2
00A7	TCPNT	RMB	2
00A9	NXTMAC	RMB	2
00AB	NXTOUT	RMB	2
00AD	XMAC	RMB	2
00AF	TSIN	RMB	1
00B0	TIND	RMB	1
00B1	TLLN	RMB	1
00B2	SUPL	RMB	1
00B3	SWRDF	RMB	1
00B4	CAP	RMB	1
00B5	SCAP	RMB	1
00B6	TPOS	RMB	1
00B7	DELIM	RMB	1
00B8	TCNT	RMB	1
00B9	MCNT	RMB	1
00BA	TTLPN	RMB	2
00BC	ENDLIN	RMB	1
00BD	TAB	RMB	1
00BE	TFILL	RMB	1

* ENVIRONMENT PARAMETERS

00BF	AUTO	RMB	2
00C1	RDM	RMB	2
00C3	WIDTH	RMB	2
00C5	FILFLG	RMB	2
00C7	PFLG	RMB	2
00C9	PCHAR	RMB	2
00CB	CNJ	RMB	2

00CD	RTJ	RMB	2
00CF	MSC	RMB	2
00D1	CNTFLG	RMB	2
00D3	JUST	RMB	2
00D5	TLN	RMB	2
00D7	BUFPNT	RMB	4
00DB	BUFEND	RMB	4
00DF	EBFEND	RMB	4
00E3	CMNPNT	RMB	2
00E5	SPCPPT1	RMB	2
00E7	SPCPPT2	RMB	2
00E9	TEMP	RMB	2
00EB	TEMP2	RMB	2
00ED	TEMP5	RMB	2
00EF	TEMP6	RMB	2
00F1	RETREG	RMB	2
00F3	INDEX	RMB	2
00F5	XTEMP	RMB	2
00F7	MACEND	RMB	2
00F9	CRF	RMB	1
0110	ORG		\$0110
0110	TAB\$	RMB	20
0124	TABEND	RMB	1
0125	NUM	RMB	12

* DISK ROUTINE EQUATES

7127	GETFIL	EQU	\$7127
712D	SETEXT	EQU	\$712D
7121	NXTCH	EQU	\$7121
7118	DPSTRN	EQU	\$7118
713C	RPTERR	EQU	\$713C
711E	DPCRLF	EQU	\$711E
7803	FMSCLS	EQU	\$7803
7806	FMS	EQU	\$7806

* DOS EQUATES

708E	DATE	EQU	\$708E
7091	LSTTRM	EQU	\$7091
708C	WASN	EQU	\$708C
7096	RETRNR	EQU	\$7096
7094	DBUFFP	EQU	\$7094
7080	DBSP	EQU	\$7090
7081	DDEL	EQU	\$7081
7082	DEOL	EQU	\$7082
70A3	OUTCHN	EQU	\$70A3
70A1	ACIAFL	EQU	\$70A1
0200	ORG		\$0200

```

        *
        * PROGRAM ENTRY POINT
        *
0200 7E 02 0C    START   JMP    INTRO

        * JUMP TABLE

0203 7E 71 12    OUTCH   JMP    $7112
0206 7E 71 0F    INCH    JMP    $710F
0209 7E 71 03    MON     JMP    $7103
01FF             STACK   EQU    $01FF

        * MAIN PROGRAM STARTS HERE

020C 8E 01 FF    INTRO   LDS    #STACK    *** SETUP STACK ***
020F BD 02 51    INTRO0  JSR    CLRSPC   GO CLEAR SPACE
0212 97 92      STA A   LOWPG  SET PAGE LIMITS
0214 97 91      STA A   TLPP
0216 4A          DEC A
0217 97 93      STA A   HIPG
0219 B6 70 8E    LDA A   DATE   GET DOS DATE
021C 97 3C      STA A   MNTH  SET MONTH
021E B6 70 8F    LDA A   DATE+1
0221 97 33      STA A   DAY   SET DAY
0223 B6 70 90    LDA A   DATE+2
0226 97 48      STA A   YEAR  SET YEAR
0228 CE 16 CC    INTRO5  LDX    #PGSTR  PRMPT FOR PAGES
022B BD 15 BA    JSR    PSTRNG
022E BD 15 59    JSR    GIBUF
0231 7C 00 5A    INC    CMFLG
0234 BD 12 53    JSR    CHKNUM
0237 24 0D      BCC    INTRO6
0239 96 65      LDA A   INNUM
023B 97 92      STA A   LOWPG
023D BD 12 53    JSR    CHKNUM
0240 24 04      BCC    INTRO6
0242 96 65      LDA A   INNUM
0244 97 93      STA A   HIPG
0246 BD 15 B0    JSR    CRLF
0249 4F          CLR A
024A CE 00 4F    LDX    #GDNUM
024D 8D 06      BSR    CLRSP2
024F 20 18      BRA    INIT   GO INITIALIZE

        * CLEAR TEMPORARY SPACE

0251 4F          CLRSPC CLR A
0252 CE 00 30    LDX    #NMREGS
0255 A7 00      CLRSP2 STA A 0,X
0257 08          INX
0258 8C 00 90    CPX    #PRNTR
025B 26 F8      BNE    CLRSP2
025D CE 00 9B    LDX    #SBFLG
0260 A7 00      CLRSP4 STA A 0,X

```

0262 08		INX	
0263 8C 00 D7		CPX	#BUFFPNT
0266 26 F8		BNE	CLRSP4
0268 39		RTS	RETURN

* INITIALIZATION AND SETUP

0269 CE 01 10	INIT	LDX	#TABS	SET POINTER
026C 4F		CLR A		
026D A7 00	INIT25	STA A	0,X	CLEAR TABS
026F 08		INX		
0270 8C 01 25		CPX	#NIJM	
0273 26 F8		BNE	INIT25	FINISHED?
0275 4C		INC A		
0276 97 D3		STA A	JUST	SET INITIAL PARAMS.
0278 97 D4		STA A	JUST+1	
027A 97 BC		STA A	ENDLIN	MARK END LINE
027C 97 F9		STA A	CRF	
027E 97 C5		STA A	FILFLG	SET FOR FILL
0280 97 C6		STA A	FILFLG+1	
0282 97 3D		STA A	LINCNT	INIT LINE COUNT
0284 97 32		STA A	COLCNT	
0286 97 78		STA A	COLCN2	SET COLUMN CNT
0288 97 69		STA A	PGN	SET PAGE
028A 86 41		LDA A	#65	
028C 97 C3		STA A	WIDTH	SET PAGE WIDTH
028E 97 C4		STA A	WIDTH+1	
0290 97 3B		STA A	LLN	AND LINE LENGTH
0292 97 9C		STA A	LLN2	
0294 97 D5		STA A	TLN	SET TITLE LENGTH
0296 97 D6		STA A	TLN+1	
0298 4C		INC A		
0299 97 3F		STA A	PGL	SET PAGE LENGTH
029B CE 1D 8A		LDX	#MACROS	
029E DF A9		STX	NXTMAC	INIT MACRO SPACE
02A0 DF A3		STX	FSTAVL	
02A2 86 FF		LDA A	#\$FF	
02A4 A7 00	INIT3	STA A	0,X	
02A6 08		INX		
02A7 8C 2E FD		CPX	#LMACRO	FINISHED?
02AA 26 F8		BNE	INIT3	
02AC DF A1		STX	LSTAVL	
02AE 09		DEX		
02AF 6F 00		CLR	0,X	SET END OF MACROS
02B1 6F 01		CLR	1,X	
02B3 6F 02		CLR	2,X	
02B5 86 0D		LDA A	#\$D	FIX BUFFER
02B7 B7 19 A0		STA A	CMNDBF	
02BA 86 A0		LDA A	#\$A0	SET FILL CHAR.
02BC 97 BE		STA A	TFILL	
02BE CE 19 6E		LDX	#TRAPS	
02C1 86 FF		LDA A	#\$FF	INIT TRAPS
02C3 A7 00	INIT4	STA A	0,X	
02C5 08		INX		

02C6 8C 19 9E		CPX	#TRPEND	FINISHED?
02C9 26 F8		BNE	INIT4	
02CB CE 17 7A		LDX	#LINBUF	
02CE DF D7		STX	BUFPNT	SET POINTER
02D0 DF D9		STX	BUFPNT+2	
02D2 BD 06 97		JSR	FIXBFE	FIX BUFFER END
02D5 DE DB		LDX	BUFEND	
02D7 DF DD		STX	BUFEND+2	
02D9 CE 18 15		LDX	#EXTBUF	
02DC DF E1		STX	EBFEND+2	
02DE CE 1C 88		LDX	#MACTBL	CLEAR MACRO TABLE
02E1 DF F7		STX	MACEND	

* MAIN PROCESSOR LOOP

02E3 CE 17 6F	DPROC	LDX	#MACST	POINT TO NAME
02E6 DF F5		STX	XTEMP	SAVE IT
02E8 CE 1B 08		LDX	#TFCB	POINT TO FCB
02EB DF 8E		STX	XTEMP2	SAVE IT
02ED 86 3E		LDA A	#'>	SET ITEM CHAR
02EF 97 4C		STA A	ITEM	
02F1 C6 0B		LDA B	#11	SET COUNTER
02F3 DE F5	DPROC1	LDX	XTEMP	GET POINTER
02F5 A6 00		LDA A	0,X	GET CHAR
02F7 08		INX		BUMP POINTER
02F8 DF F5		STX	XTEMP	
02FA DE 8E		LDX	XTEMP2	GET DESTINATION
02FC A7 04		STA A	4,X	PUT IN NAME
02FE 08		INX		
02FF DF 8E		STX	XTEMP2	SAVE
0301 5A		DEC B		DEC THE COUNT
0302 26 EF		BNE	DPROC1	
0304 CE 1B 08		LDX	#TFCB	POINT TO FCB
0307 B6 70 8C		LDR A	WASN	GET WORK DRIVE
030A A7 03		STA A	3,X	SET IN FCB
030C 86 01		LDA A	#1	OPEN FOR READ
030E A7 00		STA A	0,X	
0310 BD 78 06		JSR	FMS	CALL FMS
0313 27 65		BEQ	PROC	
0315 A6 01		LDA A	1,X	
0317 81 04		CMP A	#4	CHECK FOR NO FILE
0319 27 03		BEQ	DPROC2	
031B 7E 16 7A		JMP	DOFMS4	GO REPORT ERROR
031E B6 70 91	DPROC2	LDA A	LSTTRM	GET LAST TERM
0321 81 0D		CMP A	#\$D	IS IT CR?
0323 27 19		BEQ	DPRO24	
0325 B1 70 82		CMP A	DEOL	IS IT EOL CHAR?
0328 27 14		BEQ	DPRO24	
032A 7F 00 5F		CLR	DONE	
032D CE 1B 08		LDX	#TFCB	POINT TO FCB
0330 BD 71 27		JSR	GETFIL	GET FILE NAME
0333 24 30		BCC	DPROCS5	
0335 B6 70 91		LDA A	LSTTRM	
0338 81 2A		CMP A	#\$2A	IS CHAR A '*'?

033A 27 19		BEQ	DPROC4	
033C 20 0B		BRA	DPROC3	
033E 7D 00 8B	DPROC24	TST	PSCNT	FIRST NAME?
0341 27 06		BEQ	DPROC3	
0343 BD 78 03	DPROC25	JSR	FMSCLS	CLOSE FMS
0346 7E 09 84		JMP	FINIS4	FINISH PAGE
0349 CE 17 26	DPROC3	LDX	#ILFN	POINT TO STRING
034C BD 15 BA		JSR	PSTRNG	OUTPUT IT
034F BD 78 03		JSR	FMSCLS	CLOSE FMS
0352 7E 02 09		JMP	MON	EXIT
0355 CE 17 45	DPROC4	LDX	#CHST	POINT TO STRING
0358 BD 15 BA		JSR	PSTRNG	OUTPUT IT
035B BD 15 CF		JSR	EINCH	WAIT FOR CHAR
035E BD 71 21		JSR	NXTCH	GET NEXT CHAR
0361 24 E6		BCC	DPROC3	
0363 20 B9		BRA	DPROC2	
0365 7C 00 8B	DPROC5	INC	PSCNT	BUMP PASS COUNTER
0368 CE 1B 08		LDX	#TFCB	SET POINTER
036B 86 01		LDA A	#1	SETUP TXT CODE
036D BD 71 2D		JSR	SETEXT	SET EXTENSION
0370 CE 1B 08		LDX	#TFCB	POINT TO FCB
0373 86 01		LDA A	#1	
0375 A7 00		STA A	0,X	OPEN FOR READ
0377 BD 16 6B		JSR	DOFMS	CALL FMS
037A 96 69	PROC	LDA A	PGN	CHECK PAGE NUMBER
037C 91 92		CMP A	LOWPG	AGAINST LOW PAGE
037E 24 06		BHS	PROC3	
0380 C6 0F	PROC2	LDA B	#\$F	
0382 D7 6E		STA B	NOOUT	SET NO OUTPUT FLAG
0384 20 0A		BRA	PUNTST	
0386 91 93	PROC3	CMP A	HIPG	AGAINST HIGH PAGE
0388 23 03		BLS	PROC4	
038A 7E 09 84		JMP	FINIS4	IF PNST, FINISH
038D 7F 00 6E	PROC4	CLR	NOOUT	

* TEST FOR PUNCTUATION

0390 96 C7	PUNTST	LDA A	PFLG	TEST FLAG
0392 81 03		CMP A	#3	
0394 26 07		BNE	PUNTS3	
0396 96 C9		LDA A	PCHAR	GET SPARE CHAR.
0398 7F 00 C7	PUNTS2	CLR	PFLG	CLEAR PUNCT. FLAG
039B 20 37		BRA	JSTFY	
039D BD 06 CE	PUNTS3	JSR	GETCHR	GET NEXT CHAR.
03A0 D6 5F		LDA B	DONE	FINISHED?
03A2 27 03		BEQ	PUNT35	
03A4 7E 09 75		JMP	FINISH	
03A7 D6 C5	PUNT35	LDA B	FILFLG	FILL ON?
03A9 27 29		BEO	JSTFY	
03AB D6 C7		LDA B	PFLG	TEST PUNCT. FLAG
03AD C1 01		CMP B	#1	
03AF 22 19		BHI	PUNTS7	
03B1 27 11		BEQ	PUNTS6	

03B3 81 2E	CMP A #'	IS CHAR A '.'	
03B5 27 08	BEQ PUNTS4		
03B7 81 21	CMP A #'!	IS IT '!'?	
03B9 27 04	BEQ PUNTS4		
03BB 81 3F	CMP A #'?	IS IT '?'?	
03BD 26 03	BNE PUNTS5		
03BF 7C 00 C7	PUNTS4	INC PFLG	SET PUNCT. FLAG
03C2 20 10	PUNTS5	BRA JSTFY	
03C4 81 20	PUNTS6	CMP A #\$20	IS CHAR SPACE?
03C6 27 F7		BEQ PUNTS4	
03C8 20 CE		BRA PUNTS2	
03CA 81 20	PUNTS7	CMP A #\$20	CHECK FOR SPACE
03CC 27 CR		BEQ PUNTS2	
03CE 97 C9		STA A PCHAR	SAVE SPARE CHAR.
03D0 86 20		LDA A #\$20	SET FOR SPACE
03D2 20 EB		BRA PUNTS4	

* JUSTIFICATION LOOP

03D4 CE 18 15	JSTFY	LDX #EXTBUF	FIX EXTRA POINTERS
03D7 DF EB		STX TEMP2	
03D9 DF DF		STX EBFEND	
03DB DE D7		LDX BUFPNT	GET BUFFER POINTER
03DD 81 0D		CMP A #\$D	IS CHAR. A CR?
03DF 26 14		BNE JSTFY3	
03E1 D6 C5		LDA B FILFLG	FILL MODE?
03E3 26 05		BNE JSTFY2	
03E5 DF A5	JSTFY1	STX STROUT	MARK LAST BUF. POS.
03E7 7E 05 4E		JMP OUTLIN	OUTPUT LINE
03EA 86 20	JSTFY2	LDA A #\$20	GET A SPACE
03EC A7 00	JSTF25	STA A 0,X	SAVE IT
03EE 08		INX	BUMP POINTER
03EF 9C DB		CPX BUFEND	END OF BUFFER?
03F1 26 F9		BNE JSTF25	
03F3 20 21		BRA JSTFY6	
03F5 BD 05 EA	JSTFY3	JSR TSULN	
03F8 A7 00		STA A 0,X	SAVE CHARACTER
03FA 7C 00 32		INC COLCNT	BUMP COLUMN COUNT
03FD 08		INX	BUMP POINTER
03FE 9C DB		CPX BUFEND	END?
0400 26 06		BNE JSTFY4	
0402 D6 C5		LDA B FILFLG	FILL MODE?
0404 27 02		BEQ JSTFY4	
0406 20 0E		BRA JSTFY6	
0408 8C 18 15	JSTFY4	CPX #EXTBUF	BUFFER OVERFLOW?
040B 26 04		BNE JSTFY5	
040D 86 0D		LDA A #\$D	STUFF A C. R.
040F 20 D4		BRA JSTFY1	
0411 DF D7	JSTFY5	STX BUFPNT	SAVE BUF POINTER
0413 7E 03 7A	JSTF55	JMP PROC	REPEAT LOOP
0416 D6 C7	JSTFY6	LDA B PFLG	CHECK FLAG
0418 C1 03		CMP B #3	
041A 26 04		BNE JSTF63	
041C 96 C9		LDA A PCHAR	GET CHARACTER

041E 20 0B		BRA	JSTF65	
0420 81 20	JSTF63	CMP A	##\$20	IS CHAR = SPACE?
0422 27 51		BEQ	ADJSPC	
0424 BD 06 CE		JSR	GETCHR	GET NEXT CHARACTER
0427 81 20		CMP A	##\$20	IS IT SPACE?
0429 27 4A		BEQ	ADJSPC	
042B 36	JSTF65	PSH A		SAVE CHAR.
042C 86 20		LDA A	##\$20	
042E DE DB		LDX	BUFEND	GET TO END
0430 09	JSTFY7	DEX		
0431 8C 17 79		CPX	#LINBUF-1	LOOK FOR SPACES
0434 27 1C		BEQ	JSTFY9	
0436 A1 00		CMP A	0,X	
0438 26 F6		BNE	JSTFY7	
043A 08	JSTFY8	INX		BUMP POINTER
043B 9C DB		CPX	BUFEND	
043D 27 13		BEQ	JSTFY9	
043F A6 00		LDA A	0,X	PICK UP CHARACTER
0441 DF E9		STX	TEMP	SAVE X
0443 DE EB		LDX	TEMP2	
0445 A7 00		STA A	0,X	MOVE THE CHAR.
0447 08		INX		
0448 DF EB		STX	TEMP2	
044A DE E9		LDX	TEMP	RESTORE X
044C 86 20		LDA A	##\$20	SET WITH SPACE
044E A7 00		STA A	0,X	SAVE IT
0450 20 E8		BRA	JSTFY8	REPEAT
0452 32	JSTFY9	PUL A		RESTORE CHARACTER
0453 7F 00 77		CLR	TABFLG	CLEAR TABS
0456 CE 01 24		LDX	#TABEND	POINT TO TABS
0459 DF 75		STX	NXTTAB	SET NEXT TAB
045B DE EB		LDX	TEMP2	RESTORE X
045D BD 05 EA	JSTF95	JSR	TSULN	TEST UNDERLINE
0460 A7 00		STA A	0,X	SAVE LAST CHAR.
0462 08		INX		BUMP POINTER
0463 DF DF		STX	EBFEND	SET END
0465 81 20		CMP A	##\$20	WAS CHAR A SPACE?
0467 27 0C		BEQ	ADJSPC	
0469 8C 18 42		CPX	#LINBU2	BUFFER OVERFLOW?
046C 27 07		BEQ	ADJSPC	
046E BD 06 CE		JSR	GETCHR	GET NEXT CHAR.
0471 DE DF		LDX	EBFEND	GET POINTER
0473 20 E8		BRA	JSTF95	

* ADJUST BUFFER FOR SPACES

0475 5F	ADJSPC	CLR B		CLEAR COUNT
0476 CE 17 7A		LDX	#LINBUF	POINT TO BUF BEGIN
0479 DF E5		STX	SPCF1	
047B A6 00	ADJSP2	LDA A	0,X	LOOK FOR SPACES
047D 81 20		CMP A	##\$20	
047F 26 09		BNE	ADJS35	
0481 5C		INC B		INC THE COUNTER
0482 08		INX		BUMP POINTER

0483 9C DB		CPX	BUFEND	
0485 26 F4		BNE	ADJSP2	
0487 7E 05 4E	ADJSP3	JMP	OUTLIN	OUTPUT LINE
048A DF E7	ADJS35	STX	SPCPT2	SET END
048C BD 06 08		JSR	DELCHR	DELETE INIT SPACES
048F CE 17 7A		LDX	#LINBUF	POINT TO BEGIN
0492 86 20		LDA A	##\$20	CHECK MORE SPACES
0494 A1 00	ADJSP4	CMP A	0, X	
0496 27 07		BEQ	ADJSP5	
0498 08		INX		BUMP TIL FIND
0499 9C DB		CPX	BUFEND	END OF BUFFER?
049B 27 10		BEQ	ADJSP6	
049D 20 F5		BRA	ADJSP4	REPEAT
049F 08	ADJSP5	INX		BUMP POINTER
04A0 9C DB		CPX	BUFEND	FINISHED?
04A2 26 05		BNE	ADJSP5	
04A4 7C 00 B3		INC	SWRDF	SET SINGLE WORD
04A7 20 04		BRA	ADJSP6	
04A9 A1 00	ADJSP5	CMP A	0, X	CHECK NEXT CHAR.
04AB 27 F2		BEQ	ADJSP5	
04AD D6 D1	ADJSP6	LDA B	CNTFLG	CENTERING?
04AF 27 03		BEQ	ADJSP7	
04B1 7E 06 6A		JMP	CNTRIT	GO CENTER LINE
04B4 D6 D3	ADJSP7	LDA B	JUST	JUSTIFICATION?
04B6 27 CF		BEQ	ADJSP3	
04B8 D6 CD		LDA B	RTJ	RIGHT HAND?
04BA 27 03		BEQ	ADJSP8	
04BC 7E 05 F8		JMP	RIGHTJ	GO DO RIGHT
04BF D6 CB	ADJSP8	LDA B	CNJ	CENTER JUST. ?
04C1 27 03		BEQ	ADJSP9	
04C3 7E 06 03		JMP	CENTJ	GO CENTER
04C6 D6 B3	ADJSP9	LDA B	SWRDF	CHECK SINGLE
04C8 26 BD		BNE	ADJSP3	
04CA D6 60		LDA B	FLBF	FLUSHING BUFFER?
04CC 26 B9		BNE	ADJSP3	
04CE D6 62		LDA B	LEFT	WHICH SIDE
04D0 27 3A		BEQ	RINS	GO FROM RIGHT

* INSERT SPACES FROM LEFT

04D2 CE 17 7A	LINS	LDX	#LINBUF	SET POINTER
04D5 DF E9		STX	TEMP	SAVE IT
04D7 DE DB	LINS2	LDX	BUFEND	POINT TO END
04D9 09		DEX		DEC THE POINTER
04DA A6 00		LDA A	0, X	GET CHARACTER
04DC 81 20		CMP A	##\$20	IS IT A SPACE?
04DE 26 A7		BNE	ADJSP3	
04E0 DE E9		LDX	TEMP	RESTORE POINTER
04E2 A6 00	LINS3	LDA A	0, X	GET CHAR
04E4 81 20		CMP A	##\$20	IS IT SPACE?
04E6 27 07		BEQ	LINS4	
04E8 08		INX		BUMP POINTER
04E9 9C DB		CPX	BUFEND	END OF BUFFER
04EB 27 E5		BEQ	LINS	

04ED 20 F3		BRA	LINS3	REPEAT
04EF C6 01	LINS4	LDA B	#1	SET COUNT = 1
04F1 BD 06 2E		JSR	INSSPC	GO INSERT SPACE
04F4 D6 CB		LDA B	CNJ	CENTER JUST?
04F6 27 01		BEQ	LINS5	
04F8 39		RTS		RETURN
04F9 DE E9	LINS5	LDX	TEMP	RESTORE POINTER
04FB A6 00	LINS6	LDA A	0, X	GET CHARACTER
04FD 81 20		CMP A	#\$20	IS IT SPACE?
04FF 26 07		BNE	LINS7	
0501 08		INX		BUMP POINTER
0502 9C DB		CPX	BUFEND	END OF BUFFER?
0504 27 CC		BEQ	LINS	
0506 20 F3		BRA	LINS6	
0508 DF E9	LINS7	STX	TEMP	SAVE X
050A 20 CB		BRA	LINS2	REPEAT

* INSERT SPACES FROM RIGHT SIDE

050C DE DB	RINS	LDX	BUFEND	SET POINTER
050E 86 20		LDA A	#\$20	SET UP SPACE
0510 09	RINS2	DEX		
0511 A1 00		CMP A	0, X	IS CHAR A SPACE?
0513 27 FB		BEQ	RINS2	
0515 DF E9		STX	TEMP	SAVE POINTER
0517 DE DB	RINS3	LDX	BUFEND	GO TO END
0519 09		DEX		
051A A6 00		LDA A	0, X	GET CHAR.
051C 81 20		CMP A	#\$20	IS IT SPACE?
051E 26 2E		BNE	OUTLIN	
0520 DE E9		LDX	TEMP	RESTORE X
0522 A6 00	RINS4	LDA A	0, X	GET CHAR
0524 81 20		CMP A	#\$20	IS IT SPACE?
0526 27 08		BEQ	RINS5	
0528 09		DEX		DEC THE POINTER
0529 8C 17 79		CPX	#LINBUF-1	FINISHED?
052C 27 DE		BEQ	RINS	
052E 20 F2		BRA	RINS4	REPEAT
0530 C6 01	RINS5	LDA B	#1	SET COUNT = 1
0532 BD 06 2E		JSR	INSSPC	INSERT SPACE
0535 D6 CB		LDA B	CNJ	CENTER JUST?
0537 27 01		BEQ	RINS6	
0539 39		RTS		RETURN
053A DE E9	RINS6	LDX	TEMP	RESTORE POINTER
053C A6 00	RINS7	LDA A	0, X	GET CHARACTER
053E 81 20		CMP A	#\$20	SPACE?
0540 26 08		BNE	RINS8	
0542 09		DEX		
0543 8C 17 79		CPX	#LINBUF-1	FINISHED?
0546 27 C4		BEQ	RINS	
0548 20 F2		BRA	RINS7	REPEAT
054A DF E9	RINS8	STX	TEMP	SAVE POINTER
054C 20 C9		BRA	RINS3	

* OUTPUT LINE FROM WORK BUFFER

054E 7F 00 B3	OUTLIN	CLR	SWRDF	CLR FLAG
0551 D6 3E		LDA B	LFM	LEFT MARGIN?
0553 7D 00 70		TST	PTFL	PUT IN INDENT?
0556 26 02		BNE	OUTLI1	
0558 DB 38		ADD B	IND	ADJUST LEFT
055A 7F 00 70	OUTLI1	CLR	PTFL	
055D DB 71		ADD B	SIN	ADD IN SINGLE IN.
055F 2B 0C		BMI	OUTLI3	
0561 27 0A		BEQ	OUTLI3	
0563 86 20	OUTLI2	LDA A	#\$20	SET UP SPACE
0565 37		PSH B		
0566 BD 15 D9		JSR	OUTCHR	OUTPUT SPACE
0569 33		PUL B		
056A 5A		DEC B		DEC COUNT
056B 26 F6		BNE	OUTLI2	
056D D6 C5	OUTLI3	LDA B	FILFLG	FILL MODE?
056F 27 11		BEQ	OUTLI5	
0571 86 20		LDA A	#\$20	SETUP SPACE
0573 DE DB		LDX	BUFEND	GO TO END
0575 8C 17 7A	OUTLI4	CPX	#LINBUF	EMPTY?
0578 27 19		BEQ	OUTLI6	
057A 09		DEX		DEC THE POINTER
057B A1 00		CMP A	0,X	IS IT SPACE?
057D 27 F6		BEQ	OUTLI4	
057F 08		INX		BUMP POINTER
0580 DF A5		STX	STPOUT	SET END
0582 CE 17 7A	OUTLI5	LDX	#LINBUF	
0585 9C A5		CPX	STPOUT	EMPTY?
0587 27 0A		BEQ	OUTLI6	
0589 A6 00	OUTL55	LDA A	0,X	GET CHARACTER
058B BD 15 D9		JSR	OUTCHR	OUTPUT IT
058E 08		INX		BUMP POINTER
058F 9C A5		CPX	STPOUT	FINISHED?
0591 26 F6		BNE	OUTL55	
0593 5F	OUTLI6	CLR B		CLEAR FLAGS
0594 D7 83		STA B	DWFLG	
0596 D7 64		STA B	NOFL	
0598 D7 C7		STA B	PFLG	
059A D7 68		STA B	NSP	
059C D7 71		STA B	SIN	
059E 73 00 62		COM	LEFT	SWITCH SP. SIDES
05A1 CE 17 7A		LDX	#LINBUF	SET POINTER
05A4 DF D7		STX	BUFPNT	
05A6 CE 01 10		LDX	#TABS	SET TABS
05A9 DF 75		STX	NXTTAB	
05AB BD 15 29		JSR	FIXWD	GO FIX WIDTH
05AE CE 18 15	OUTLI7	LDX	#EXTBUF	
05B1 9C DF	OUTL75	CPX	EBFEND	CHECK FOR EXTRA?
05B3 27 17		BEQ	OUTLI9	
05B5 A6 00		LDA A	0,X	GET CHARACTER
05B7 08		INX		
05B8 DF E9		STX	TEMP	

05BA DE D7	LDX	BUFFPNT	TRANSFER IT
05BC A7 00	STA A	0,X	
05BE 08	INX		BUMP POINTER
05BF 9C DB	CPX	BUFEND	CHECK END
05C1 27 09	BEQ	OUTLI8	OVERFLOW!
05C3 DF D7	STX	BUFFPNT	SAVE IT
05C5 DE E9	LDX	TEMP	
05C7 7C 00 32	INC	COLCNT	BUMP COLUMN COUNT
05CA 20 E5	BRA	OUTL75	REPEAT
05CC CE 18 15	OUTLI8	LDX #EXTBUF	FIX POINTER
05CF DF DF	STX	EBFEND	
05D1 BD 08 DE	JSR	PCRLF	OUTPUT CR & LF
05D4 96 CF	LDA A	MSC	MULTIPLE SPACE?
05D6 27 0A	BEQ	OUTL85	
05D8 4A	OUTL82	DEC A	
05D9 27 07	BEQ	OUTL85	
05DB 36	PSH A		OUTPUT EXTRA SPACE
05DC BD 08 DE	JSR	PCRLF	
05DF 32	PUL A		
05E0 20 F6	BRA	OUTL82	
05E2 96 60	OUTL85	LDA A	FLBF
05E4 27 01	BEQ	OUTLI9	
05E6 39	RTS		
05E7 7E 03 7A	OUTLI9	JMP	PROC
			GO PROCESS

* SET UNDERLINED CHARACTER

05EA 7D 00 4E	TSULN	TST	ULFLG	CHECK FLAG
05ED 27 08		BEQ	TSULN2	
05EF BD 12 1F		JSR	CLSFY	CLASS CHARACTER
05F2 5D		TST B		
05F3 27 02		BEQ	TSULN2	
05F5 8A 80		OPA A	#\$80	SET PARITY
05F7 39	TSULN2	RTS		RETURN

* RIGHT HAND JUSTIFY

05F8 BD 06 5C	RIGHTJ	JSR	CNTSPC	COUNT SPACES
05FB CE 17 79	RIGHT2	LDX	#LINBUF-1	
05FE 8D 2E		BSR	INSSPC	INSERT SPACES
0600 7E 05 4E		JMP	OUTLIN	OUTPUT LINE

* CENTER JUSTIFY

0603 8D 57	CENTJ	BSR	CNTSPC	COUNT SPACES
0605 57		ASR B		DIVIDE BY 2
0606 20 F3		BRA	RIGHT2	

* DELETE CHARACTERS

0608 DE E7	DELCHR	LDX	SPCPCT2	GET POINTER
060A 9C E5		CPX	SPCPCT1	EMPTY?
060C 27 1F		BEQ	DELCH4	
060E 9C DB		CPX	BUFEND	

0610 27 0E		BEQ	DELCH3	
0612 A6 00		LDA A	0,X	GET CHARACTER
0614 08		INX		BUMP THE POINTER
0615 DF E7		STX	SPCPT2	SAVE IT
0617 DE E5		LDX	SPCPT1	RESTORE
0619 A7 00		STA A	0,X	SAVE CHARACTER
061B 08		INX		BUMP POINTER
061C DF E5		STX	SPCPT1	
061E 20 E8		BRA	DELCHR	REPEAT
0620 DE E5	DELCH3	LDX	SPCPT1	GET POINTER
0622 86 20		LDA A	#\$20	SETUP SPACE
0624 9C DB	DELC35	CPX	BUFEND	
0626 27 05		BEQ	DELCH4	
0628 A7 00		STA A	0,X	PUT IN SPACE
062A 08		INX		BUMP POINTER
062B 20 F7		BRA	DELC35	
062D 39	DELCH4	RTS		

* INSERT SPACES

062E 5D		INSSPC	TST B	TEST COUNT
062F 27 2A			BEQ	IF NONE, RETURN
0631 37			PSH B	SAVE COUNT
0632 DF E9			STX	TEMP
0634 DE DB			LDX	BUFEND
0636 DF E5			STX	SPCPT1
0638 08		INSSP2	INX	SAVE
0639 5A			DEC B	
063A 26 FC			BNE	DEC THE COUNT
063C DF E7			STX	SPCPT2
063E DE E5	INSSP3		LDX	SPCPT1
0640 9C E9			CPX	TEMP
0642 27 0E			BEQ	FINISHED?
0644 A6 00			LDA A	GET CHARACTER
0646 09			DEX	DEC THE POINTER
0647 DF E5			STX	SAVE IT
0649 DE E7			LDX	SPCPT2
064B A7 00			STA A	0,X
064D 09			DEX	PUT CHARACTER
064E DF E7			STX	SPCPT2
0650 20 EC			BRA	REPEAT
0652 33	INSSP4		PUL B	RESTORE COUNT
0653 86 20			LDA A	SETUP SPACE
0655 08	INSS44		INX	BUMP THE POINTER
0656 A7 00			STA A	0,X
0658 5A			DEC B	STUFF SPACE
0659 26 FA			BNE	DEC THE COUNT
065B 39	INSSP5	RTS		RETURN

* COUNT SPACES

065C 5F	CNTSPC	CLR B		CLEAR COUNT
065D 86 20		LDA A	#\$20	SETUP SPACE
065F DE DB		LDX	BUFEND	SET POINTER

0661 09	CNTSP2	DEX		
0662 A1 00		CMP A 0,X	SPACE?	-
0664 26 03		BNE CNTSP3		
0666 5C		INC B	BUMP THE COUNT	
0667 20 F8		BRA CNTSP2		
0669 39	CNTSP3	RTS		

* CENTER LINE

066A 8D F0	CNTRIT	BSR CNTSPC	GO COUNT SPACES	
066C 96 83		LDA A DWFLG	DOUBLE WIDTH?	
066E 27 0E		BEQ CNTRI4		
0670 96 C3		LDA A WIDTH	GET WIDTH	
0672 10		SBA		
0673 48		ASL A	FIX FOR DOUBLE	
0674 91 C3		CMP A WIDTH		
0676 22 0C		BHI CNTRI5		
0678 16		TAB	SAVE	
0679 96 C3		LDA A WIDTH	SUB FROM WIDTH	
067B 10		SBA		
067C 16		TAB		
067D 57		ASR B	DIVIDE BY TWO	
067E 57	CNTRI4	ASR B		
067F CE 17 79		LDX #LINBUF-1	SET POINTER	
0682 8D AA		BSR INSSPC	GO INSERT SPACE	
0684 7A 00 D1	CNTRI5	DEC CNTFLG	DEC CENTER COUNT	
0687 26 0B		BNE CNTRI6		
0689 4F		CLR A		
068A 97 D1		STA A CNTFLG	CLEAR FLAG	
068C 96 63		LDA A TFILF	GET TEMP FILL	
068E 97 C5		STA A FILFLG	SET FILL	
0690 DE DB		LDX BUFEND	SET POINTER	
0692 DF A5		STX STPOUT	SET END	
0694 7E 05 4E	CNTRI6	JMP OUTLIN	OUTPUT LINE	

* FIX BUFFER END POINTER

0697 CE 17 7A	FIXBFE	LDX #LINBUF	SET POINTER	
0698 DF DB		STX BUFEND		
069C 96 3B		LDA A LLN	GET LINE LENGTH	
069E 90 C3		SUB A WIDTH	CALC. COLUMN NUM.	
06A0 4C		INC A		
06A1 97 32		STA A COLCNT	SAVE COUNT	
06A3 5F		CLR B		
06A4 96 C3		LDA A WIDTH	GET WIDTH	
06A6 9B DC		ADD A BUFEND+1	ADD TO BUFEND	
06A8 D9 DB		ADC B BUFEND		
06AA 97 DC		STA A BUFEND+1	SAVE RESULT	
06AC D7 DB		STA B BUFEND		
06AE 39		RTS	RETURN	

* RETURN FROM MACRO

06AF 7F 00 7D	RETMAC	CLR FINMAC	CLEAR FLAG	
---------------	--------	------------	------------	--

06B2 32		PUL A	FIX STACK
06B3 32		PUL A	
06B4 32		PUL A	
06B5 97 D1		STA A CNTFLG	RESTORE FLAG
06B7 CE 00 55		LDX #NUMPNT	
06B8 32	RETMA2	PUL A	RESTORE VALUES
06B9 A7 00		STA A 0, X	
06BD 08		INX	
06BE 8C 00 62		CPX #LEFT	FINISHED?
06C1 26 F7		BNE RETMA2	
06C3 7A 00 80		DEC MACCNT	DEC MACRO COUNTER
06C6 96 61		LDA A ATFLG	DOING AT?
06C8 27 0B		BEQ GETCH1	
06CA 39		RTS	RETURN

* CLEAR 'ENDLIN' AND GET CHARACTER

06CB 7F 00 BC	CLRGET	CLR	ENDLIN
---------------	--------	-----	--------

* GET NEXT CHARACTER

06CE BD 15 95	GETCHR	JSR TSTBRK	TEST FOR BREAK
06D1 96 7D		LDA A FINMAC	FINISH MACRO?
06D3 26 DA		BNE RETMAC	
06D5 96 57	GETCH1	LDA A EXCHR	GET EXTRA CHAR.
06D7 27 03		BEQ GETCH2	
06D9 7E 11 EF		JMP FTCHNM	GET NUMBER
06DC 96 5A	GETCH2	LDA A CMFLG	COMMAND?
06DE 27 0D		BEQ GETCH3	
06E0 DE E3	GETC22	LDX CMNPNT	SET POINTER
06E2 A6 00		LDA A 0, X	GET CHARACTER
06E4 81 0D		CMP A #\$D	C. R. ?
06E6 27 01		BEQ GETC25	
06E8 08		INX	BUMP THE POINTER
06E9 DF E3	GETC25	STX CMNPNT	SAVE IT
06EB 20 29		BRA FETCHR	
06ED 96 9B	GETCH3	LDA A SBFLG	SPECIAL BUFFER?
06EF 26 EF		BNE GETC22	
06F1 96 5B		LDA A MBFLG	MACRO BUFFER?
06F3 27 08		BEQ GETCH4	
06F5 BD 0F 1C		JSR INMAC	GET CHARACTER
06F8 26 1C		BNE FETCHR	
06FA 7E 0F B6		JMP MCEND	FINISH MACRO
06FD 96 85	GETCH4	LDA A SPIFLG	SPECIAL INPUT?
06FF 27 05		BEQ GETCH5	
0701 BD 15 CF		JSR EINCH	GET CHARACTER
0704 20 10		BRA FETCHR	
0706 96 84	GETCH5	LDA A DFMFLG	DEFINE MACRO?
0708 9A 5E		ORA A NOCR	
070A 26 87		BNE GETCH6	
070C 96 77		LDA A TABFLG	TABS?
070E 27 03		BEQ GETCH6	
0710 7E 0B 5D		JMP DOTAB	GO DO TAB
0713 BD 15 FC	GETCH6	JSR INCHR	GET CHARACTER

* FETCH AND CHECK CHARACTER

0716 81 1A	FETCHR	CMP A #\$\$1A	END OF FILE?
0718 26 05		BNE FETCH2	
071A 97 5F		STA A DONE	SET FLAG
071C 7E 09 75		JMP FINISH	
071F 81 0D	FETCH2	CMP A #\$\$D	C. R. ?
0721 26 2C		BNE FETCH3	
0723 7F 00 6B		CLR SPSPF	SPECIAL SPACE?
0726 D6 5E		LDA B NOCR	
0728 26 35		BNE FETC35	
072A D6 9B		LDA B SBFLG	CHECK FLAG
072C 26 07		BNE FETC22	
072E D6 BC		LDA B ENDLIN	END OF LINE?
0730 27 03		BEQ FETC22	
0732 BD 09 4E		JSR FLUSHB	FLUSH BUFFER
0735 97 BC	FETC22	STA A ENDLIN	SET FLAGS
0737 7F 00 9B		CLR SBFLG	
073A 7F 00 4E		CLR ULFLG	
073D D6 84		LDA B DFMFLG	TEST
073F DA D1		ORA B CNTFLG	
0741 26 08		BNE FETC25	
0743 D6 C5		LDA B FILFLG	TEST FILL
0745 27 04		BEQ FETC25	
0747 86 20		LDA A #\$\$20	SETUP SPACE
0749 20 19		BRA FETC36	
074B 86 0D	FETC25	LDA A #\$\$D	SETUP C. R.
074D 20 15		BRA FETC36	
074F D6 84	FETCH3	LDA B DFMFLG	GET FLAG
0751 DA 81		ORA B PASFLG	
0753 26 0A		BNE FETC35	
0755 D6 6A		LDA B PASCHR	PASS CHAR?
0757 27 0F		BEQ FETCH4	
0759 81 20		CMP A #\$\$20	IS IT A SPACE?
075B 26 02		BNE FETC35	
075D 8A 80		ORA A #\$\$80	SET PARITY
075F 5F	FETC35	CLR B	CLEAR FLAGS
0760 D7 BC		STA B ENDLIN	
0762 D7 6A		STA B PASCHR	
0764 7F 00 B4	FETC36	CLR CAP	
0767 39	FETC37	RTS	RETURN
0768 81 1F	FETCH4	CMP A #\$\$1F	CHECK CHAR
076A 22 03		BHI FETC45	
076C 7E 06 CE		JMP GETCHR	GO GET CHAR.
076F D6 BC	FETC45	LDA B ENDLIN	END OF LINE?
0771 27 1A		BEQ FETCH5	
0773 81 2E		CMP A #'.'	PERIOD?
0775 27 06		BEQ FETC47	
0777 81 3A		CMP A #'.'	COLON?
0779 26 05		BNE FETC48	
077B 97 64		STA A NOFL	SET NO FLUSH
077D 7E 08 24	FETC47	JMP COMMAND	DO COMMAND
0780 81 20	FETC48	CMP A #\$\$20	SPACE?

0782 26 09		BNE	FETCH5	
0784 97 6B		STA A	SPSPF	SET FLAG
0786 BD 09 4E		JSR	FLUSHB	FLUSH BUFFER
0789 86 A0	FETC49	LDA A	#\$A0	
078B 20 D2		BRA	FETC35	
078D D6 6B	FETCH5	LDA B	SPSPF	TEST FLAG
078F 27 07		BEQ	FETC55	
0791 81 20		CMP A	#\$20	IS IT SPACE?
0793 27 F4		BEQ	FETC49	
0795 7F 00 6B		CLR	SPSPF	CLEAR OUT
0798 D6 5A	FETC55	LDA B	CMFLG	COMMAND?
079A DA 5B		ORA B	MBFLG	
079C DA 82		ORA B	NONUMS	
079E DA 85		ORA B	SPIFLG	
07A0 DA 9B		ORA B	SBFLG	
07A2 26 1C		BNE	FETCH6	
07A4 91 BD		CMP A	TAB	CHECK IF TAB
07A6 26 18		BNE	FETCH6	
07A8 DE 75		LDX	NXTTAB	GET NEXT TAB
07AA D6 32		LDA B	COLCNT	GET COUNT
07AC 6D 00	FETC57	TST	0, X	CHECK
07AE 27 AF		BEQ	FETC35	
07B0 E1 00		CMP B	0, X	FINISHED?
07B2 25 03		BLO	FETC58	
07B4 08		INX		BUMP THE POINTER
07B5 20 F5		BRA	FETC57	
07B7 DF 75	FETC58	STX	NXTTRB	SAVE POINTER
07B9 96 BE		LDA A	TFILL	
07BB 97 77		STA A	TABFLG	SET FLAG
07BD 7E 07 5F	FETC59	JMP	FETC35	
07C0 D6 82	FETCH6	LDA B	NONUMS	NUMBERS?
07C2 26 1D		BNE	FETCH7	
07C4 81 23		CMP A	#'#	POUND SIGN?
07C6 27 04		BEQ	FETC65	
07C8 81 25		CMP A	#'%	PERCENT SIGN?
07CA 26 15		BNE	FETCH7	
07CC D6 5E	FETC65	LDA B	NOCR	DO C. R. ?
07CE 37		PSH B		
07CF 97 5E		STA A	NOCR	SAVE VALUES
07D1 97 74		STA A	NOEXP	
07D3 BD 11 D7		JSR	CLRNUM	CLEAR NUMBER
07D6 BD 12 C1		JSR	PRNU32	PROCESS NUMBER
07D9 33		PUL B		
07DA D7 5E		STA B	NOCR	RESTORE VALUES
07DC 24 DF		BCC	FETC59	
07DE 7E 06 CE		JMP	GETCHR	GET CHARACTER
07E1 81 5C	FETCH7	CMP A	#'\`	BACK SLASH?
07E3 26 05		BNE	FETC75	
07E5 97 6A		STA A	PASCHR	SET PASS CHAR.
07E7 7E 06 CB		JMP	CLRGET	GO GET IT
07EA 81 40	FETC75	CMP A	#'@	AT SIGN?
07EC 27 1C		BEQ	CAPIT	
07EE 81 5E		CMP A	#\$5E	UP ARROW?
07F0 27 21		BEQ	SETCRP	

07F2 D6 B4		LDA B	CRP	CHECK MODE
07F4 DA B5		ORA B	SCAP	
07F6 DA 5A		ORA B	CMFLG	
07F8 26 C3		BNE	FETC59	
07FA 81 41		CMP A	#'A	CHECK IF LETTER
07FC 25 BF		BLO	FETC59	
07FE 81 5A		CMP A	#'Z	
0800 22 BB		BHI	FETC59	
0802 D6 6C		LDA B	DOCAP	DO CAP?
0804 27 B7		BEQ	FETC59	
0806 8B 20		ADD A	#\$20	FORCE TO LOWER
0808 20 B3	FETCH8	BRA	FETC59	

* CAP SINGLE LETTER

080A D6 6C	CAPIT	LDA B	DOCAP	CHECK MODE
080C 27 FA		BEQ	FETCH8	
080E 97 B4		STA A	CAP	SET FLAG
0810 7E 06 CB	CAPIT2	JMP	CLRGET	

* CAP STRING OF LETTERS

0813 D6 6C	SETCAP	LDA B	DOCAP	CHECK MODE
0815 27 F1		BEQ	FETCH8	
0817 D6 B5		LDA B	SCAP	GET FLAG
0819 27 05		BEQ	SETCA2	
081B 7F 00 B5		CLR	SCAP	CLEAR IT
081E 20 F0		BRA	CAPIT2	
0820 97 B5	SETCA2	STA A	SCAP	SET FOR STRING
0822 20 EC		BRA	CAPIT2	

* COMMAND PROCESSOR

0824 7F 00 BC	COMAND	CLR	ENDLIN	CLEAR FLAG
0827 CE 19 9F		LDX	#CMNDBF-1	SET POINTER
082A 08	COMAN2	INX		BUMP IT
082B 7C 00 5E		INC	NOCR	SET NO C. R.
082E DF EF		STX	TEMP6	SAVE POINTER
0830 7C 00 82		INC	NONUMS	
0833 BD 06 CE		JSR	GETCHR	GET CHARACTER
0836 DE EF		LDX	TEMP6	RESTORE POINTER
0838 7F 00 5E		CLR	NOCR	CLEAR FLAG
083B 7F 00 82		CLR	NONUMS	
083E A7 00		STA R	0, X	PUT CHARACTER
0840 81 0D		CMP A	#\$D	WAS IT A C. R. ?
0842 26 E6		BNE	COMAN2	
0844 7F 00 BC		CLR	ENDLIN	RESET END LINE
0847 CE 19 A0		LDX	#CMNDBF	SET POINTER
084A A6 00	COMAN3	LDA A	0, X	GET CHARACTER
084C 08		INX		BUMP THE POINTER
084D E6 00		LDA B	0, X	GET NEXT CHAR
084F 08		INX		BUMP
0850 DF E3		STX	CMNPNT	SAVE THE POINTER
0852 81 5F		CMP A	#\$5F	LOWER CASE?

0854 23 04		BLS	COMAN4	
0856 80 20		SUB A	#\$20	SET TO UPPER
0858 C0 20		SUB B	#\$20	
085A CE 09 8D	COMAN4	LDX	#CMNDT	POINT TO TABLE
085D A1 00	COMAN5	CMP A	0, X	COMPARE FIRST
085F 26 0C		BNE	COMAN7	
0861 E1 01		CMP B	1, X	COMPARE SECOND
0863 26 08		BNE	COMAN7	
0865 97 5A		STA A	CMFLG	FOUND COMMAND
0867 EE 02		LDX	2, X	GET ADDRESS
0869 AD 00	COMAN6	JSR	0, X	GO DO IT
086B 20 2F		BRA	FINCM	FINISH COMMAND
086D 08	COMAN7	INX		BUMP POINTER
086E 08		INX		
086F 08		INX		
0870 08		INX		
0871 8C 0A 89		CPX	#TBLEND	TABLE END?
0874 26 E7		BNE	COMAN5	
0876 36		PSH A		
0877 96 80		LDA A	MACCNT	TEST MACRO NUMBER
0879 81 07		CMP A	#7	
087B 32		PUL A		
087C 24 15		BHS	MACOVF	OVERFLOW?
087E CE 1C 88		LDX	#MACTBL	POINT TO MACROS
0881 9C F7	COMAN8	CPX	MACEND	END?
0883 27 17		BEQ	FINCM	
0885 A1 00		CMP A	0, X	COMPARE FIRST
0887 26 04		BNE	COMAN9	
0889 E1 01		CMP B	1, X	COMPARE SECOND
088B 27 29		BEQ	CALMAC	
088D 08	COMAN9	INX		FIND NEXT ENTRY
088E 08		INX		
088F 08		INX		
0890 08		INX		
0891 20 EE		BRA	COMAN8	

* MACRO OVERFLOW ERROR

0893 CE 16 FB	MACOVF	LDX	#OVFSTR	POINT TO STRING
0896 BD 15 BA		JSR	PSTRNG	OUTPUT IT
0899 7E 02 09		JMP	MON	

* FINISH COMMAND

089C 96 7F	FINCM	LDA A	IFFLG	CHECK FOR IF
089E 27 0A		BEQ	FINCM1	
08A0 4F		CLR A		CLEAR FLAGS
08A1 97 BC		STA A	ENDLIN	
08A3 97 5A		STA A	CMFLG	
08A5 97 7F		STA A	IFFLG	
08A7 7E 08 4A		JMP	COMAN3	GO DO COMMAND
08AA 7F 00 64	FINCM1	CLR	NOFL	CLEAR FLAGS
08AD 7F 00 5A	FINCM2	CLR	CMFLG	
08B0 7C 00 BC	FINCM4	INC	ENDLIN	SET END LINE

08B3 7E 06 CE JMP GETCHR GO GET CHARACTER

* CALL MACRO

08B6 DF AD	CALMAC	STX	XMAC	SAVE POINTER
08B8 CE 00 61		LDX	#ATFLG	POINT TO VALUES
08BB A6 00	CALMA2	LDA A	0,X	GET VALUE
08BD 36		PSH A		PUT ON STACK
08BE 6F 00		CLR	0,X	CLEAR IT
08C0 09		DEX		
08C1 8C 00 54		CPX	#INC	FINISHED?
08C4 26 F5		BNE	CALMA2	
08C6 96 D1		LDA A	CNTFLG	SAVE CNT FLAG
08C8 36		PSH A		
08C9 7F 00 D1		CLR	CNTFLG	
08CC 7C 00 80		INC	MACCNT	BUMP COUNTER
08CF DE AD		LDX	XMAC	RESTORE COUNT
08D1 86 0F		LDA A	#\$F	
08D3 97 5B		STA A	MBFLG	SET FLAG
08D5 97 BC		STA A	ENDLIN	
08D7 EE 02		LDX	2,X	GET ADDRESS
08D9 DF 5C		STX	MBFPNT	SAVE AS POINTER
08DB 7E 03 7A		JMP	PROC	GO PROCESS

* PRINT C. R. AND L. F.

08DE BD 14 26	PCRLF	JSR	PUSHX	SAVE X
08E1 8D 07	PCRLF2	BSR	SCRLF	DO CR AND LF
08E3 BD 15 95		JSR	TSTBRK	BREAK?
08E6 BD 14 38	PCRLF4	JSR	PULLX	RESTORE X
08E9 39		RTS		RETURN

* SPECIAL CARRIAGE RETURN LINE FEED

08EA 96 86	SCRLF	LDA A	DIVFLG	DIVERTING?
08EC 27 0F		BEQ	SCRLF2	
08EE CE 16 C5		LDX	#CRLFST	POINT TO STRING
08F1 A6 00	SCRLF1	LDA A	0,X	GET CHARACTER
08F3 81 04		CMP A	#4	IS IT TERM?
08F5 27 50		BEQ	SCRLF9	
08F7 BD 15 D9		JSR	OUTCHR	OUTPUT CHAR
08FA 08		INX		BUMP POINTER
08FB 20 F4		BRA	SCRLF1	
08FD 96 6E	SCRLF2	LDA A	NOOUT	CHECK OUTPUT
08FF 26 03		BNE	SCRLF4	
0901 BD 71 1E		JSR	DPCRLF	OUTPUT CR & LF
0904 7C 00 3D	SCRLF4	INC	LINCNT	BUMP LINE COUNTER
0907 96 3D	SCRLF5	LDA A	LINCNT	
0909 CE 19 6E		LDX	#TRAPS	POINT TO TRAPS
090C A1 00	SCRL55	CMP A	0,X	LINE = TRAP?
090E 27 24		BEQ	SCRLF8	
0910 08		INX		GET TO NEXT
0911 08		INX		
0912 08		INX		

0913 8C 19 9E		CPX	#TRPEND	END?
0916 26 F4		BNE	SCRL55	
0918 91 3F	SCRLF6	CMP A	PGL	BOTTOM OF PAGE?
091A 23 2B		BLS	SCRLF9	
091C 96 53		LDA A	NPGN	GET NEW PAGE NUM.
091E 27 07		BEQ	SCRLF7	
0920 7F 00 53		CLR	NPGN	
0923 97 69		STA A	PGN	SET PAGE NUMBER
0925 20 03		BRA	SCRL75	
0927 7C 00 69	SCRLF7	INC	PGN	BUMP BY ONE
092A 86 01	SCRL75	LDA A	#1	SET UP 1
092C 97 3D		STA A	LINCNT	SET LINE COUNT
092E 96 B2		LDA A	SUPL	CHECK FLAG
0930 26 15		BNE	SCRLF9	
0932 20 D3		BRA	SCRLF5	
0934 7C 00 61	SCRLF8	INC	ATFLG	BUMP AT COUNT
0937 96 BC		LDA A	ENDLIN	SAVE STATUS
0939 36		PSH A		
093A A6 01		LDA A	1,X	GET NAME
093C E6 02		LDA B	2,X	
093E BD 08 5R	SCRL85	JSR	COMAN4	GO PROCESS
0941 7A 00 61		DEC	ATFLG	DEC COUNT
0944 32		PUL A		
0945 97 BC		STA A	ENDLIN	RESTORE STATUS
0947 39	SCRLF9	RTS		RETURN

* BREAK FILLED BUFFER

0948 86 01	BRAK	LDA A	#1	SETUP 1
094A 91 3D		CMP A	LINCNT	TEST LINE COUNT
094C 27 B9		BEQ	SCRLF5	

+ FLUSH WORK BUFFER

094E 96 64	FLUSHB	LDA A	NOFL	NO FLUSH?
0950 26 1F		BNE	FLUSH5	
0952 86 20	FLUSH	LDA A	#\$20	SET UP SPACE
0954 DE D7		LDX	BUFPNT	SET POINTER
0956 8C 17 7A		CPX	#LINBUF	BEGINNING OF BUFFER?
0959 27 16		BEQ	FLUSH5	
095B DF A5		STX	STPOUT	SET END
095D 9C DB	FLUSH2	CPX	BUFEND	END?
095F 27 05		BEQ	FLUSH3	
0961 A7 00		STA A	0,X	SAVE CHARACTER
0963 08		INX		BUMP POINTER
0964 20 F7		BRA	FLUSH2	
0966 CE 17 7A	FLUSH3	LDX	#LINBUF	POINT TO BUFFER
0969 97 60		STA A	FLBF	SET FLAG
096B BD 04 75		JSR	ADJSPC	ADJUST SPACE
096E 7F 00 60		CLR	FLBF	
0971 7F 00 64	FLUSH5	CLR	NOFL	CLEAR FLRG
0974 39		RTS		RETURN

* FINISH AND CLEAN UP

0975 8D D7	FINISH	BSR	FLUSHB	FLUSH BUFFER
0977 CE 1B 08		LDX	#TFCB	POINT TO FCB
097A 86 04		LDA A	#4	SET FOR CLOSE
097C A7 00		STA A	0, X	
097E BD 16 6B		JSR	DOFMS	CALL FMS
0981 7E 03 1E		JMP	DPROC2	
0984 7C 00 B2	FINIS4	INC	SUPL	
0987 BD 0A B4		JSR	PAGE	GO PAGE
098A 7E 02 09		JMP	MON	EXIT

* COMMAND TABLE

098D 53	CMNDT	FCC	'SP'
098F 0A 91		FDB	SPACE
0991 50		FCC	'PG'
0993 0A B4		FDB	PAGE
0995 4D		FCC	'MS'
0997 0A D8		FDB	MULTS
0999 53		FCC	'SS'
099B 0A E6		FDB	SGLGS
099D 4E		FCC	'NJ'
099F 0A EA		FDB	NOJST
09A1 4A		FCC	'JU'
09A3 0A EE		FDB	JST
09A5 44		FCC	'DH'
09A7 0C 47		FDB	DUBH
09A9 44		FCC	'DW'
09AB 0C 52		FDB	DUBW
09AD 44		FCC	'DB'
09AF 0C 5B		FDB	DUBB
09B1 43		FCC	'CE'
09B3 0C 67		FDB	CENTER
09B5 42		FCC	'BR'
09B7 09 48		FDB	BRAK
09B9 2A		FCC	'*'
09BB 0A B3		FDB	SPACE6
09BD 46		FCC	'FI'
09BF 0C F5		FDB	FILL
09C1 4E		FCC	'NF'
09C3 0C EE		FDB	NOFILL
09C5 53		FCC	'SI'
09C7 0C 9C		FDB	SIND
09C9 50		FCC	'PI'
09CB 0C B9		FDB	PTIND
09CD 50		FCC	'PN'
09CF 0C 3A		FDB	PGNUM
09D1 4C		FCC	'LM'
09D3 0B 18		FDB	LEFTM
09D5 49		FCC	'IN'
09D7 0B 28		FDB	INDNT
09D9 4C		FCC	'LN'
09DB 0B 3E		FDB	LENTH
09DD 4E		FCC	'NS'

09DF 0B 70	FDB	NOSPC
09E1 52	FCC	'RS'
09E3 0B 73	FDB	RESPC
09E5 50	FCC	'PL'
09E7 0C 87	FDB	PAGEL
09E9 43	FCC	'CP'
09EB 0C B0	FDB	STCRP
09ED 4E	FCC	'NC'
09EF 0C B5	FDB	NOCRP
09F1 4E	FCC	'NL'
09F3 0D 1F	FDB	NEDL
09F5 53	FCC	'SY'
09F7 0D 66	FDB	SAYS
09F9 4F	FCC	'OS'
09FB 0D 82	FDB	OUTSY
09FD 41	FCC	'AT'
09FF 0D 8B	FDB	ATL
0A01 44	FCC	'DM'
0A03 0D D7	FDB	DEFMAC
0A05 41	FCC	'AM'
0A07 0E 15	FDB	APMAC
0A09 52	FCC	'RM'
0A0B 0E 1E	FDB	REMMAC
0A0D 44	FCC	'DI'
0A0F 0E 5E	FDB	DIVERT
0A11 44	FCC	'DA'
0A13 0E 7D	FDB	DIVAPP
0A15 53	FCC	'ST'
0A17 0D 08	FDB	STOP
0A19 54	FCC	'TL'
0A1B 0F FA	FDB	TITLE
0A1D 4C	FCC	'LT'
0A1F 0F ED	FDB	TLEN
0A21 43	FCC	'CH'
0A23 11 4B	FDB	CHNG
0A25 49	FCC	'IF'
0A27 10 DE	FDB	IF
0A29 4E	FCC	'NR'
0A2B 11 9E	FDB	NREG
0A2D 41	FCC	'AR'
0A2F 11 BD	FDB	ARB
0A31 53	FCC	'SR'
0A33 11 C1	FDB	SRDM
0A35 43	FCC	'CR'
0A37 11 C6	FDB	CROM
0A39 41	FCC	'AU'
0A3B 11 CA	FDB	SAUTO
0A3D 54	FCC	'TC'
0A3F 0B 77	FDB	TRBCH
0A41 54	FCC	'TF'
0A43 0B 82	FDB	TRBFIL
0A45 54	FCC	'TA'
0A47 0B 8E	FDB	STAB
0A49 45	FCC	EX'

0A4B 09 75	FDB	FINISH
0A4D 54	FCC	'TM'
0A4F 0B AD	FDB	TERM
0A51 47	FCC	'GI'
0A53 0B C2	FDB	GETIN
0A55 45	FCC	'EV'
0A57 06 D0	FDB	SENV
0A59 52	FCC	'RP'
0A5B 0C FC	FDB	RPT
0A5D 50	FCC	'PS'
0A5F 0A 8A	FDB	PASS
0A61 55	FCC	'UL'
0A63 14 4D	FDB	UNDL
0A65 52	FCC	'RI'
0A67 14 52	FDB	RDIT
0A69 49	FCC	'IC'
0A6B 14 6F	FDB	ITMCH
0A6D 4E	FCC	'NI'
0A6F 14 7B	FDB	NXTI
0A71 4E	FCC	'NB'
0A73 14 A1	FDB	NXTB
0A75 43	FCC	'CF'
0A77 14 CC	FDB	CLSFL
0A79 4F	FCC	'OF'
0A7B 14 DE	FDB	OPNF
0A7D 20	FCC	/ /
0A7F 0A B3	FDB	SPACE6
0A81 20	FCC	/ /
0A83 0A B3	FDB	SPACE6
0A85 20	FCC	/ /
0A87 0A B3	FDB	SPACE6
0A89 00	TBLEND	FCB
		0

* PASS FILE ROUTINE .PS

0A8A 7F 00 C5	PASS	CLR	FILFLG	FIX FLAGS
0A8D 7C 00 81		INC	PASFLG	
0A90 39		RTS		

* SPACE ROUTINE .SP N

0A91 BD 09 4E	SPACE	JSR	FLUSHB	FLUSH BUFFER
0A94 96 68		LDA A	NSP	NO SPACE?
0A96 26 1B		BNE	SPACE6	
0A98 BD 12 53		JSR	CHKNUM	CHECK FOR NUMBER
0A9B 96 65		LDA A	INNUM	GET NUMBER
0A9D 26 03		BNE	SPACE2	
0A9F 7C 00 65		INC	INNUM	INC BY ONE
0AA2 BD 0D 2E	SPACE2	JSR	FNTR	FIND TRAP
0AA5 91 65		CMP A	INNUM	EQUAL?
0AA7 25 02		BLO	SPACE4	
0AA9 96 65		LDA A	INNUM	GET NUMBER
0AAB 36	SPACE4	PSH A		
0AAC BD 08 DE		JSR	PCRLF	OUTPUT CR AND LF

0RAF 32		PUL A		
0AB0 4A		DEC A		DEC COUNT
0AB1 26 F8		BNE	SPACE4	
0AB3 39	SPACE6	RTS		RETURN

* PAGE ROUTINE . PG +N

0AB4 BD 12 53	PAGE	JSR	CHKNUM	CHECK FOR NUMBER
0AB7 24 07		BCC	PAGE2	
0AB9 96 69		LDA A	PGN	GET PAGE NUMBER
0ABB BD 12 42		JSR	FIXVAL	FIX VALUE
0ABE 20 08		BRA	PAGE4	
0AC0 96 68	PAGE2	LDA A	NSP	NO SPACE?
0AC2 26 13		BNE	PAGE6	
0AC4 96 53		LDA A	NPGN	GET NEW PAGE NUM.
0AC6 26 03		BNE	PAGE4	
0AC8 96 69		LDA A	PGN	
0ACA 4C		INC A		BUMP BY ONE
0ACB 97 53	PAGE4	STA A	NPGN	SAVE AS NEW
0ACD BD 09 4E		JSR	FLUSHS	FLUSH BUFFER
0AD0 BD 08 DE	PAGES	JSR	PCRLF	OUTPUT CR & LF
0AD3 96 53		LDA A	NPGN	GET NEW PAGE NUM.
0AD5 26 F9		BNE	PAGES	
0AD7 39	PAGE6	RTS		RETURN

* MULTIPLE SPACE ROUTINE MS +N

0AD8 BD 12 53	MULTS	JSR	CHKNUM	CHECK FOR NUMBER
0ADB 24 04		BCC	MULTS2	
0ADD 96 65		LDA A	INNUM	GET NUMBER
0ADF 20 02		BRA	MULTS3	
0AE1 86 02	MULTS2	LDA A	#2	DEFAULT IS 2
0AE3 97 CF	MULTS3	STA A	MSC	SET COUNT
0AE5 39		RTS		

* SINGLE SPACE ROUTINE SS

0AE6 7F 00 CF	SNGLS	CLR	MSC	CLEAR COUNT
0AE9 39		RTS		

* NO ADJUST ROUTINE NJ

0AEA 7F 00 D3	NOJST	CLR	JUST	CLEAR JUST FLAG
0AFD 39		RTS		

* SET JUSTIFICATION ROUTINE , JU C

0AEE 97 D3	JST	STA A	JUST	SET FLAG
0AF0 BD 12 11		JSR	LDNSKP	GET NEXT CHAR.
0AF3 BD 12 1F		JSR	CLSFY	CLASSIFY IT
0AF6 C1 02		CMP B	#2	
0AF8 26 09		BNE	JST15	
0AFA 81 4E		CMP A	#'N	NORMAL?
0AFC 26 06		BNE	JST2	

0FE 4F	JST1	CLR A	ADJUST FLRGS
0FF 97 CB		STA A CNJ	
B01 97 CD		STA A RTJ	
IB03 39	JST15	RTS	RETURN
IB04 81 52	JST2	CMP A #'R	RIGHT HAND?
IB06 26 06		BNE JST3	
IB08 7F 00 CB		CLR CNJ	FIX FLAGS
IB0B 97 CD		STA A RTJ	
IB0D 39		RTS	
IB0E 81 43	JST3	CMP A #'C	CENTERED?
IB10 26 EC		BNE JST1	
IB12 7F 00 CD		CLR RTJ	FIX FLAGS
IB15 97 CB		STA A CNJ	
IB17 39	JST4	RTS	RETURN

* SET LEFT MARGIN . LM +N

0B18 BD 12 53	LEFTM	JSR CHKNUM	CHECK FOR NUMBER
0B1B 24 0A		BCC LEFTM2	
0B1D 96 3E		LDA A LFM	GET MARGIN
0B1F BD 12 42		JSR FIXVAL	FIX VALUE
0B22 2A 01		BPL LEFTM1	
0B24 4F		CLR A	
0B25 97 3E	LEFTM1	STA A LFM	SET NEW VALUE
0B27 39	LEFTM2	RTS	RETURN

* SET INDENT . IN +N

0B28 BD 09 4E	INDNT	JSR FLUSHB	FLUSH BUFFER
0B2B BD 12 53		JSR CHKNUM	CHECK FOR NUMBER
0B2E 24 F7		BCC LEFTM2	
0B30 96 38		LDA A IND	GET INDENT
0B32 BD 12 42		JSR FIXVAL	FIX VALUE
0B35 2A 01		BPL INDNT2	
0B37 4F		CLR A	
0B38 90 38	INDNT2	SUB A IND	SET INDENT
0B3A 97 B0		STA A TIND	SAVE AS TEMP
0B3C 20 14		BRA LENT25	

* SET LENGTH OF LINE . LN +N

0B3E BD 12 53	LENTH	JSR CHKNUM	CHECK FOR NUMBER
0B41 24 19		BCC LENGTH5	
0B43 96 3B		LDA A LLN	GET LENGTH
0B45 BD 12 42		JSR FIXVAL	FIX VALUE
0B48 81 0E		CMP A #14	14 OR LESS?
0B4A 22 02		BHI LENGTH2	
0B4C 86 0F		LDA A #15	FORCE TO 15
0B4E 90 38	LENGTH2	SUB A LLN	SET NEW
0B50 97 B1		STA A TLLN	SAVE AS TEMP
0B52 DE D7	LENT25	LDX BUFPN	CHECK POINTER
0B54 8C 17 7A		CPX #LINBUF	
0B57 26 03		BNE LENGTH5	
0B59 7E 15 29		JMP FIXWD	GO FIX WIDTH

0B5C 39	LENGTH5	RTS	RETURN
* DO NECESSARY TABBING			
0B5D D6 32	DOTAB	LDA B COLCNT	GET COUNT
0B5F DE 75		LDX NXTTAB	POINT TO TAB
0B61 E1 00		CMP B 0, X	COMPARE
0B63 24 05		BHS DOTAB2	
0B65 96 BE		LDA A TFILL	GET FILL CHAR.
0B67 7E 07 4F		JMP FETCH3	
0B6A 7F 00 77	DOTAB2	CLR TABFLG	CLEAR FLAG
0B6D 7E 06 CE		JMP GETCHR	BACK TO GET CHAR.
* SET NO SPACE NS			
0B70 97 68	NOSPC	STA A NSP	SET FLAG
0B72 39		RTS	
* RESTORE SPACE MODE .RS			
0B73 7F 00 68	RESPC	CLR NSP	CLEAR FLAG
0B76 39		RTS	
* DEFINE TAB CHARACTER .TC C			
0B77 BD 12 11	TABCH	JSR LDNSKP	GET TO NEXT CHAR.
0B7A 81 0D		CMP A #\$D	IS IT A C. R. ?
0B7C 26 01		BNE TABCH2	
0B7E 4F		CLR A	CLEAR VALUE
0B7F 97 BD	TABCH2	STA A TAB	SAVE TAB CHAR.
0B81 39		RTS	RETURN
* DEFINE TAB FILL CHARACTER .TF C			
0B82 BD 12 11	TABFIL	JSR LDNSKP	GET TO NEXT CHAR.
0B85 81 0D		CMP A #\$D	IS IT C. R. ?
0B87 26 02		BNE TABFI2	
0B89 86 A0		LDA A #\$A0	SET UNPAD SPACE
0B8B 97 BE	TABFI2	STA A TFILL	SAVE CHAR.
0B8D 39		RTS	RETURN
* DEFINE TAB COLUMNS TA 1 2 3 4			
0B8E CE 01 10	STAB	LDX #TAB\$	POINT TO TABS
0B91 BD 14 26	STAB2	JSR PUSHX	SAVE X
0B94 BD 12 53		JSR CHKNUM	CHECK FOR NUMBER
0B97 24 0E		BCC STAB4	
0B99 BD 14 38		JSR PULLX	RESTORE
0B9C 96 65		LDA A INNUM	GET NUMBER
0B9E A7 00		STA A 0, X	SAVE IT
0BA0 08		INX	BUMP POINTER
0BA1 8C 01 24		CPX #TABEND	END OF TABLE?
0BA4 26 EB		BNE STAB2	
0BA6 39		PTS	RETURN

0BA7 BD 14 38	STAB4	JSR	PULLX	
0BAR 6F 00		CLR	0, X	CLEAR LAST
0BAC 39		RTS		

* OUTPUT STRING TO TERMINAL . TM STRING

0BAD BD 12 11	TERM	JSR	LDNSKP	GET TO NEXT CHAR.
0BB0 A6 00	TERM2	LDA A	0, X	GET CHAR.
0BB2 81 0D		CMP A	#\$D	IS IT C. R. ?
0BB4 27 03		BEQ	TERM4	
0BB6 08		INX		BUMP THE POINTER
0BB7 20 F7		BRA	TERM2	
0BB9 86 04	TERM4	LDA A	#4	SET UP 4
0BBB A7 00		STA A	0, X	SAVE IT
0BBD DE E3		LDX	CMNPNT	SET POINTER
0BBF 7E 15 BA		JMP	PSTRNG	GO PRINT STRING

* GET INPUT FROM TERMINAL . GI PROMPT

0BC2 8D E9	GETIN	BSR	TERM	GO PRINT PROMPT
0BC4 CE 16 E4		LDX	#QUSTR	POINT TO STR.
0BC7 BD 15 BC		JSR	PDATA	OUTPUT IT
0BCA BD 15 59		JSR	GIBUF	GET INPUT RESPONSE
0BCD 97 9B		STA A	SBFLG	SET FLAG
0BCF 39		RTS		RETURN

* SET NEW ENVIRONMENT . EV N

0BD0 BD 12 53	SENV	JSR	CHKNUM	CHECK FOR NUMBER
0BD3 24 08		BCC	SENY1	
0BD5 96 65		LDA A	INNUM	GET NUMBER
0BD7 27 05		BEQ	SENY2	
0BD9 86 01		LDA A	#1	SET UP 1
0BDB 20 01		BRA	SENY2	
0BDD 4F	SENY1	CLR A		CLEAR VALUE
0BDE 91 73	SENY2	CMP A	EV	PRESENT EV?
0BE0 26 01		BNE	SENY3	
0BE2 39		RTS		YES, RETURN
0BE3 97 73	SENY3	STA A	EV	SET NEW EV
0BE5 96 32		LDA A	COLCNT	SAVE COL COUNT
0BE7 D6 78		LDR B	COLCN2	
0BE9 D7 32		STA B	COLCNT	
0BEB 97 78		STA A	COLCN2	
0BED 96 38		LDA A	IND	FIX THE INDENT
0BEF D6 79		LDR B	IND2	
0BF1 97 79		STA A	IND2	
0BF3 D7 38		STA B	IND	
0BF5 96 3B		LDA A	LLN	DO LINE LENGTH
0BF7 D6 9C		LDA B	LLN2	
0BF9 97 9C		STA A	LLN2	
0BFB D7 3B		STA B	LLN	
0BFD CE 00 BF	SENY4	LDX	#AUTO	POINT TO BLOCK
0C00 A6 00		LDA A	0, X	GET VALUE
0C02 E6 01		LDA B	1, X	

0C04 A7 01		STA A	1,X	SWAP VALUE
0C06 E7 00		STA B	0,X	
0C08 08		INX		GO TO NEXT
0C09 08		INX		
0C0A 8C 00 D7		CPX	#BUFPNT	FINISHED?
0C0D 26 F1		BNE	SENV4	
0C0F A6 00	SENV6	LDA A	0,X	GET VALUE
0C11 E6 02		LDA B	2,X	
0C13 A7 02		STA A	2,X	SWAP
0C15 E7 00		STA B	0,X	
0C17 A6 01		LDA A	1,X	
0C19 E6 03		LDA B	3,X	
0C1B A7 03		STA A	3,X	
0C1D E7 01		STA B	1,X	
0C1F 08		INX		BUMP THE POINTER
0C20 08		INX		
0C21 08		INX		
0C22 08		INX		
0C23 8C 00 E3		CPX	#CMNPNT	FINISHED?
0C26 26 E7		BNE	SENV6	
0C28 CE 17 7A		LDX	#LINBUF	POINT TO BUFFER
0C2B A6 00	SENV8	LDA A	0,X	GET A CHAR.
0C2D E6 C8		LDA B	200,X	
0C2F A7 C8		STA A	200,X	SWAP FOR NEW
0C31 E7 00		STA B	0,X	
0C33 08		INX		BUMP TO NEXT
0C34 8C 18 42		CPX	#LINBU2	FINISHED?
0C37 26 F2		BNE	SENV8	
0C39 39		RTS		RETURN

* SET NEW PAGE NUMBER .PN +N

0C3A BD 12 53	PGNUM	JSR	CHKNUM	CHECK FOR NUMBER
0C3D 24 07		BCC	PGNUM4	
0C3F 96 69		LDA A	PGN	GET VALUE
0C41 BD 12 42		JSR	FIXVAL	GO FIX VALUE
0C44 97 69		STA A	PGN	SAVE NEW
0C46 39	PGNUM4	RTS		RETURN

* SET DOUBLE HEIGHT DH

0C47 BD 09 52	DUBH	JSR	FLUSH	FLUSH BUFFER
0C48 86 12	DUBH1	LDA A	#\$12	SET UP CODE
0C4C 7C 00 3D		INC	LINCNT	BUMP LINE COUNT
0C4F 7E 15 09	DUBH2	JMP	OUTCHP	OUTPUT CHARACTER

* SET DOUBLE WIDTH DW

0C52 BD 09 52	DUBW	JSR	FLUSH	FLUSH BUFFER
0C55 86 0E		LDA A	#\$0E	SET UP CODE
0C57 97 83		STA A	DWFLG	SET FLAG
0C59 20 F4		BRA	DUBH2	

* SET DOUBLE BUTH DB

0C5B BD 09 52	DUBB	JSR	FLUSH	FLUSH BUFFER
0C5E 86 0E		LDA A	#\$0E	SET UP CODE
0C60 97 83		STA A	DWFLG	SET FLAG
0C62 BD 15 D9		JSR	OUTCHR	OUTPUT CHARACTER
0C65 20 E3		BRA	DUBH1	

* CENTER N LINES . CE +N

0C67 BD 09 4E	CENTER	JSR	FLUSHB	FLUSH BUFFER
0C6A BD 12 53		JSR	CHKNUM	CHECK FOR NUMBER
0C6D 24 0B		BCC	CENTE2	
0C6F 96 D1		LDA A	CNTFLG	GET OLD COUNT
0C71 BD 12 42		JSR	FIXVAL	FIX VALUE
0C74 97 D1		STA A	CNTFLG	SAVE NEW
0C76 27 23		BEQ	PAGEL4	
0C78 20 04		BRA	CENTE4	
0C7A 86 01	CENTE2	LDA A	#1	DEFAULT TO 1
0C7C 97 D1		STA A	CNTFLG	SAVE COUNT
0C7E 96 C5	CENTE4	LDA A	FILFLG	GET FLAG
0C80 97 63		STA A	TFILF	SAVE AS TEMP
0C82 86 FF		LDA A	#\$FF	
0C84 97 C5		STA A	FILFLG	FORCE FILL MODE
0C86 39		RTS		RETURN

* SET PAGE LENGTH . PL +N

0C87 BD 12 53	PAGEL	JSR	CHKNUM	CHECK FOR NUMBER
0C8A 25 04		BCS	PAGEL1	
0C8C 86 42		LDA A	#66	DEFAULT TO 66
0C8E 20 09		BRA	PAGEL2	
0C90 96 3F	PAGEL1	LDA A	PGL	GET LAST VALUE
0C92 BD 12 42		JSR	FIXVAL	FIX VALUE
0C95 4D		TST A		
0C96 26 01		BNE	PAGEL2	
0C98 4C		INC A		BUMP BY ONE
0C99 97 3F	PAGEL2	STA A	PGL	SAVE NEW
0C9B 39	PAGEL4	RTS		RETURN

* SET SINGLE INDENT . SI +N

0C9C BD 09 4E	SIND	JSR	FLUSHB	FLUSH BUFFER
0C9F BD 12 53		JSR	CHKNUM	CHECK FOR NUMBER
0CA2 24 F7		BCC	PAGEL4	
0CA4 96 71		LDA A	SIN	GET OLD VALUE
0CA6 BD 12 42		JSR	FIXVAL	GO FIX VALUE
0CA9 90 71		SUB R	SIN	
0CBA 97 AF		STA A	TSIN	SAVE AS TEMP
0CAD 7E 0B 52		JMP	LENT25	

* SET CAPS MODE . CP

0CB0 86 0F	STCAP	LDA A	#\$F	SET FLRG
0CB2 97 6C		STA A	DOCAP	

0CB4 39

RTS

* CLEAR CAPS MODE . NC

0CB5 7F 00 6C	NOCAP	CLR	DOCAP	CLEAR FLAG
0CB8 39		RTS		

* PUT IN INDENT FIELD . PI STRING

0CB9 BD 09 4E	PTIND	JSR	FLUSHB	FLUSH BUFFER
0CBC BD 12 11		JSR	LDNSKP	GET TO NEXT CHAR.
0CBF D6 38		LDA B	IND	GET INDENT
0CC1 27 2A		BEQ	PTIND5	
0CC3 D7 5E		STA B	NOCR	SET FLAG
0CC5 5F		CLR B		
0CC6 37	PTIND2	PSH B		
0CC7 BD 06 CE		JSR	GETCHR	GO GET CHAR.
0CCA 33		PUL B		
0CCB 81 0D		CMP A	#\$D	CHECK IF C. R. ?
0CCD 27 0C		BEQ	PTIND3	
0CCF 37		PSH B		
0CD0 BD 15 D9		JSR	OUTCHR	GO OUTPUT CHAR.
0CD3 33		PUL B		
0CD4 5C		INC B		BUMP COUNT
0CD5 D1 38		CMP B	IND	FINISHED?
0CD7 24 0E		BHS	PTIND4	
0CD9 20 EB		BRA	PTIND2	
0CDB 86 20	PTIND3	LDA A	#\$20	SET UP SPACE
0CDD 37		PSH B		
0CDE BD 15 D9		JSR	OUTCHR	OUTPUT IT
0CE1 33		PUL B		
0CE2 5C		INC B		BUMP COUNT
0CE3 D1 38		CMP B	IND	FINISHED?
0CE5 25 F4		BLO	PTIND3	
0CE7 5C	PTIND4	INC B		BUMP COUNT
0CE8 D7 70		STA B	PTFL	SET FLAG
0CEA 7F 00 5E		CLR	NOCR	
0CED 39	PTIND5	RTS		RETURN

* SET NOFILL MODE . NF

0CEE BD 09 4E	NOFILL	JSR	FLUSHB	FLUSH BUFFER
0CF1 7F 00 C5		CLR	FILFLG	CLEAR FLAG
0CF4 39		RTS		

* SET FILL MODE . FI

0CF5 BD 09 4E	FILL	JSR	FLUSHB	FLUSH BUFFER
0CF8 7C 00 C5		INC	FILFLG	SET FLAG
0CFB 39		RTS		

* REPEAT COMMAND . RP

0FCF BD 09 4E	RPT	JSR	FLUSHB	FLUSH BUFFER
---------------	-----	-----	--------	--------------

0CFF 7C 00 B2	INC	SUPL	SET FLAG
0D02 BD 0A B4	JSR	PAGE	GO PAGE
0D05 7E 16 91	JMP	RWND	REWIND FILE

* STOP COMMAND . ST

0D08 BD 09 4E	STOP	JSR	FLUSHB	FLUSH BUFFER
0D0B CE 16 DA		LDX	#STPSTR	POINT TO STRING
0D0E BD 15 BA	STOP1	JSR	PSTRNG	OUTPUT IT
0D11 BD 15 CF		JSR	EINCH	GO GET CHAR.
0D14 81 53		CMP A	#'S	IS IT 'S'?
0D16 26 06		BNE	STOP2	
0D18 BD 78 03		JSR	FMSCLS	CLOSE FMS
0D1B 7E 09 84		JMP	FINIS4	GO FINISH
0D1E 39	STOP2	RTS		RETURN

* NEED N LINES . NL N

0D1F BD 12 53	NEDL	JSR	CHKNUM	CHECK FOR NUMBER
0D22 25 03		BCS	NEDL1	
0D24 7C 00 65		INC	INNUM	BUMP BY 1
0D27 8D 05	NEDL1	BSR	FNTR	GO FIND TRAP
0D29 91 65	NEDL2	CMP A	INNUM	COMPARE
0D2B 25 4A		BLO	SAYS25	
0D2D 39	NEDL4	RTS		

* FIND THE NEXT TRAP

0D2E 86 FF	FNTR	LDA A	#\$FF	SET MIN DISTANCE
0D30 97 72		STA A	MINDIS	
0D32 CE 19 6E		LDX	#TRAPS	POINT TO TRAPS
0D35 D6 3D	FNTR2	LDA B	LINCNT	GET COUNT
0D37 E1 00		CMP B	0, X	CHECK LOC.
0D39 24 0F		BHS	FNTR4	
0D3B A6 00		LDA A	0, X	GET DISTANCE
0D3D 81 FF		CMP A	#\$FF	
0D3F 27 11		BEQ	FNTR5	
0D41 10		SBA		SWAP REGISTERS
0D42 91 72		CMP A	MINDIS	MIN DISTANCE?
0D44 24 04		BHS	FNTR4	
0D46 97 72		STA A	MINDIS	SAVE NEW
0D48 DF 7A		STX	NXTTRP	SAVE POINTER
0D4A 08	FNTR4	INX		BUMP THE POINTER
0D4B 08		INX		
0D4C 08		INX		
0D4D 8C 19 9E		CPX	#TRPEND	FINISHED?
0D50 26 E3		BNE	FNTR2	
0D52 D6 72	FNTR5	LDA B	MINDIS	GET DISTANCE
0D54 C1 FF		CMP B	#\$FF	
0D56 26 07		BNE	FNTR6	
0D58 96 3F		LDA A	PGL	SET UP PAGE LEN.
0D5A 90 3D		SUB A	LINCNT	
0D5C 4C		INC A		FIX VALUE
0D5D 5F		CLR B		

0D5E 39		RTS	RETURN
0D5F E6 00	FNTR6	LDA B 0,X	
0D61 96 72		LDA A MINDIS	GET DISTANCE
0D63 DE 7A		LDX NXTTRP	POINT TO TRAP
0D65 39		RTS	RETURN

* SAVE SPACE ROUTINE .SV N

0D66 7F 00 7C	SAVS	CLR	SVDSPC	CLEAR COUNT
0D69 BD 12 53		JSR	CHKNUM	CHECK FOR NUMBER
0D6C 25 03		BCS	SAVS1	
0D6E 7C 00 65		INC	INNUM	GET COUNT
0D71 BD BB	SAVS1	BSR	FNTR	FIND TRAP
0D73 91 65	SAVS2	CMP A	INNUM	
0D75 25 06		BLO	SAVS4	
0D77 7F 00 68	SAVS25	CLR	NSP	CLEAR NO SPACE
0D7A 7E 0A AB		JMP	SPRACE4	GO DO SPACE
0D7D 96 65	SAVS4	LDA A	INNUM	GET COUNT
0D7F 97 7C		STA A	SVDSPC	SAVE COUNT
0D81 39	SAVS5	RTS		RETURN

* OUTPUT SAVED SPACE .OS

0D82 96 7C	OUTSV	LDA A	SVDSPC	GET REMEMBERED COUNT
0D84 27 FB		BEQ	SAVS5	
0D86 7F 00 7C		CLR	SVDSPC	CLEAR COUNT
0D89 20 EC		BRA	SAVS25	OUTPUT SPACE

* AT LINE N ROUTINE .AT -N

0D8B BD 12 53	ATL	JSR	CHKNUM	CHECK FOR NUMBER
0D8E 24 28		BCC	ATL35	
0D90 BD 11 3F		JSR	TSTNEG	IS IT NEGATIVE?
0D93 BD 0F C3		JSR	GTNAME	GET NAME
0D96 96 3F		LDA A	PGL	GET PAGE LEN.
0D98 4C		INC A		
0D99 BD 12 42		JSR	FIXVAL	FIX THE VALUE
0D9C 4D		TST A		
0D9D 26 01		BNE	ATL1	
0D9F 4C		INC A		BUMP BY ONE
0DA0 CE 19 6E	ATL1	LDX	#TRAPS	POINT TO TRAPS
0DA3 A1 00	ATL2	CMP A	0,X	COMPARE
0DA5 27 12		BEQ	ATL4	
0DA7 8D 27		BSR	INTRP	
0DA9 26 F8		BNE	ATL2	
0DAB CE 19 6E		LDX	#TRAPS	POINT TO TRAPS
0DAE C6 FF		LDA B	#\$FF	SET REFERENCE
0DB0 E1 00	ATL3	CMP B	0,X	
0DB2 27 14		BEQ	ATL5	
0DB4 8D 1A		BSR	INTRP	
0DB6 26 F8		BNE	ATL3	
0DB8 39	ATL35	RTS		RETURN
0DB9 D6 9D	ATL4	LDA B	MACNAME	GET NAME
0DBB 26 04		BNE	ATL45	

0DBD 5A		DEC B		DEC THE COUNT
0DBE E7 00		STA B	0,X	SAVE POSITION
0DC0 39		RTS		RETURN
0DC1 96 9E	ATL45	LDA A	MACNAM+1	GET NAME
0DC3 E7 01		STA B	1,X	SAVE CHAR.
0DC5 A7 02		STA A	2,X	
0DC7 39		RTS		RETURN
0DC8 D6 9D	ATL5	LDA B	MACNAM	GET NAME
0DCA 27 EC		BEQ	ATL35	
0DCC A7 00		STA A	0,X	SAVE CHARACTER
0DCE 20 F1		BRA	ATL45	

* INCREMENT TRAP POINTER

0DD0 08	INTRP	INX		FIX POINTER
0DD1 08		INX		
0DD2 08		INX		
0DD3 8C 19 9E		CPX	#TRPEND	FINISHED?
0DD6 39		RTS		

* DEFINE MACRO

0DD7 96 5B	DEFMAC	LDA A	MBFLG	CHECK DEF FLAG
0DD9 26 39		BNE	DEFMA5	
0DDB BD 0E 88		JSR	OPMAC	GO OPEN MACRO
0DDE 27 34	DEFMA2	BEQ	DEFMA5	
0DE0 7C 00 84		INC	DFMFLG	SET DEF FLAG
0DE3 7F 00 5A		CLR	CMFLG	CLEAR COMMAND
0DE6 BD 06 CB	DEFMA3	JSR	CLRGET	GO GET CHARACTER
0DE9 81 2E		CMP A	#',	IS IT A PERIOD?
0DEB 26 0E		BNE	DEFM35	
0DED BD 06 CB		JSR	CLRGET	GET NEXT CHAR.
0DF0 81 2E		CMP A	#',	IS IT A PERIOD?
0DF2 27 13		BEQ	DEFMA4	
0DF4 36		PSH A		SAVE CHAR
0DF5 86 2E		LDA A	#',	SET UP PERIOD
0DF7 BD 0E E7		JSR	OUTMAC	OUTPUT TO MACRO
0DFA 32		PUL A		RESTORE CHAR
0DFB BD 0E E7	DEFM35	JSR	OUTMAC	OUTPUT TO MACRO
0DFE 81 0D		CMP A	#\$D	IS IT A C.R.?
0E00 27 E4		BEQ	DEFMA3	
0E02 BD 06 CB		JSR	CLRGET	GET NEXT CHAR.
0E05 20 F4		BRA	DEFM35	
0E07 BD 0F 31	DEFMA4	JSR	CLSMAC	CLOSE MACRO
0E0A BD 06 CB	DEFM45	JSR	CLRGET	GET CHARACTER
0E0D 81 0D		CMP A	#\$D	IS IT A C.R.?
0E0F 26 F9		BNE	DEFM45	
0E11 7F 00 84		CLR	DFMFLG	CLEAR DEF FLAG
0E14 39	DEFMA5	RTS		RETURN

* APPEND TO A MACRO .AP XX

0E15 96 5B	RPMAC	LDA A	MBFLG	CHECK FLAG
0E17 26 FB		BNE	DEFMA5	

0E19 BD 0E BB	JSR	OPAPP	OPEN FOR APPEND
0E1C 20 C9	BRA	DEFMA2	

* REMOVE MACRO .RM XX

0E1E BD 0F C3	REMMRC	JSR	GTNAM	GO GET NAME
0E21 BD 0F 6E		JSR	FNDMAC	FIND MACRO
0E24 27 01		BEQ	REMMA4	
0E26 39		RTS		RETURN
0E27 DF 9F	REMMA4	STX	MACTMP	SAVE POINTER
0E29 EE 02		LDX	2,X	GET ADDRESS
0E2B BD 0F 9D		JSR	CHKLST	LAST MACRO?
0E2E 24 14		BCC	REMMA6	
0E30 DE 9F		LDX	MACTMP	GET POINTER
0E32 A6 02		LDA A	2,X	GET ADDRESS
0E34 E6 03		LDA B	3,X	
0E36 DE A1		LDX	LSTAVL	SET LAST AVAIL
0E38 A7 00		STA A	0,X	
0E3A E7 01		STA B	1,X	
0E3C DE A9		LDX	NXTMAC	SET UP NXT MAC
0E3E DF A1		STX	LSTAVL	SAVE AS LAST AVAIL
0E40 DE 9F		LDX	MACTMP	
0E42 20 0A		BRA	REMNAME	
0E44 DE 9F	REMMA6	LDX	MACTMP	SET UP POINTER
0E46 A6 02		LDA A	2,X	GET ADDRESS
0E48 E6 03		LDA B	3,X	
0E4A 97 A3		STA A	FSTAVL	SET FIRST AVAIL
0E4C D7 A4		STA B	FSTAVL+1	

* REMOVE MACRO NAME FROM TABLE

0E4E E6 04	REMNAME	LDA B	4,X	MOVE CHAR DOWN
0E50 E7 00		STA B	0,X	
0E52 08		INX		BUMP THE POINTER
0E53 9C F7		CPX	MACEND	FINISHED?
0E55 26 F7		BNE	REMNAME	
0E57 09		DEX		DEC THE POINTER
0E58 09		DEX		
0E59 09		DEX		
0E5A 09		DEX		
0E5B DF F7		STX	MACEND	SET NEW END
0E5D 39		RTS		RETURN

* DIVERT .DI XX

0E5E 96 86	DIVERT	LDA A	DIVFLG	CHECK DIV FLAG
0E60 27 0D		BEQ	DIVER2	
0E62 7C 00 87	DIVER0	INC	DIVFL2	SET MARKER
0E65 7E 0F 31		JMP	CLSMAC	CLOSE MACRO
0E68 7F 00 86	DIVER1	CLR	DIVFLG	CLEAR FLAGS
0E6B 7F 00 87		CLR	DIVFL2	
0E6E 39		RTS		RETURN
0E6F 7C 00 87	DIVER2	INC	DIVFL2	SET MARKER
0E72 8D 14		BSR	OPMAC	GO OPEN MACRO

0E74 27 F2	DIVER4	BEQ	DIVER1	
0E76 7C 00 86		INC	DIVFLG	SET FLAG
0E79 7F 00 45		CLR	LDIV	CLEAR COUNT
0E7C 39		RTS		RETURN

* DIVERT APPEND . DA XX

0E7D 96 86	DIVAPP	LDA A	DIVFLG	CHECK DIV FLAG
0E7F 26 E1		BNE	DIVER0	
0E81 7C 00 87		INC	DIVFL2	SET MARKER
0E84 8D 35		BSR	OPAPP	OPEN FOR APPEND
0E86 20 EC		BRA	DIVER4	

* OPEN A MACRO SPACE

0E88 BD 0F C3	OPMAC	JSR	GTNAM	GET MACRO NAME
0E8B 96 9D		LDA A	MACNAM	
0E8D 26 01		BNE	OPMAC2	PRESENT?
0E8F 39		RTS		
0E90 BD 0F 6E	OPMAC2	JSR	FNDMAC	LOOK FOR MACRO
0E93 26 04		BNE	OPMAC4	
0E95 8D 90		BSR	REMMA4	REMOVE OLD VERSION
0E97 20 F7		BRA	OPMAC2	OPEN MACRO
0E99 96 9D	OPMAC4	LDA A	MACNAM	GET NAME
0E9B D6 9E		LDA B	MACNAM+1	
0E9D 8C 1D 88		CPX	#MTEND	END OF TABLE?
0EA0 26 03		BNE	OPMAC5	
0EA2 7E 0F 19		JMP	SYSERR	REPORT ERROR
0EA5 A7 00	OPMAC5	STA A	0,X	SAVE NAME
0EA7 E7 01		STA B	1,X	
0EA9 96 A3		LDA A	FSTAVL	GET FIRST AVAIL
0EAB D6 A4		LDA B	FSTAVL+1	
0EAD A7 02		STA A	2,X	SAVE IN TABLE
0EBF E7 03		STA B	3,X	
0EB1 08		INX		BUMP THE POINTER
0EB2 08		INX		
0EB3 08		INX		
0EB4 08		INX		
0EB5 DF F7		STX	MACEND	SET NEW END
0EB7 DE A3		LDX	FSTAVL	GET LAST AVAIL
0EB9 20 1E		BRA	SAVSX	

* OPEN MACRO FOR APPEND

0EBB BD 0F C3	OPAPP	JSR	GTNAM	GET MACRO NAME
0EBE 96 9D		LDA A	MACNAM	
0EC0 26 01		BNE	OPAPP2	
0EC2 39		RTS		NO NAME
0EC3 BD 0F 6E	OPAPP2	JSR	FNDMAC	FIND MACRO
0EC6 26 D1		BNE	OPMAC4	
0EC8 EE 02	OPAPP4	LDX	2,X	GET LOCATION
0ECA BD 0F 9D		JSR	CHKLST	IS IT THE LAST ONE?
0ECD 24 0A		BCC	SAVSX	
0ECF 96 A3		LDA A	FSTAVL	GET FIRST AVAIL

0ED1 D6 A4	LDA B	FSTAVL+1	
0ED3 A7 00	STA A	0,X	SET NEW
0ED5 E7 01	STA B	1,X	
0ED7 DE A3	LDX	FSTAVL	

* SAVE SPECIAL INDEX

0ED9 7D 00 87	SAVSX	TST	DIVFL2	TEST MARKER
0EDC 27 06		BEQ	SAVSX2	
0EDE 7F 00 87		CLR	DIVFL2	CLEAR MARKER
0EE1 DF AB		STX	NXTOUT	SAVE POINTER
0EE3 39		RTS		RETURN
0EE4 DF A9	SAVSX2	STX	NXTMAC	SAVE POINTER
0EE6 39		RTS		

* OUTPUT TO MACRO SPACE

0EE7 DF AD	OUTMAC	STX	XMAC	SAVE POINTER
0EE9 7D 00 87		TST	DIVFL2	TEST MARKER
0EEC 27 04		BEQ	OUTMA0	
0EEE DE AB		LDX	NXTOUT	SET POINTER
0EF0 20 02		BRA	OUTMA1	
0EF2 DE A9	OUTMA0	LDX	NXTMAC	
0EF4 6D 00	OUTMA1	TST	0,X	TEST IF END
0EF6 27 18		BEQ	OUTMA4	
0EF8 81 1F		CMP A	#\$1F	IS IT \$1F?
0EFA 22 0C		BHI	OUTM18	
0EFC 81 0D		CMP A	#\$D	IS IT C.R.?
0EFE 26 0B		BNE	OUTMA3	
0F00 7D 00 87		TST	DIVFL2	TEST MARKER
0F03 27 03		BEQ	OUTM18	
0F05 7C 00 45		INC	LDIV	BUMP DIV LINE CNT
0F08 A7 00	OUTM18	STA A	0,X	PUT CHARACTER
0F0A 08	OUTMA2	INX		BUMP THE POINTER
0F0B 8D CC	OUTMA3	BSR	SAVSX	GO SAVE X
0F0D DE AD		LDX	XMAC	RESTORE POINTER
0F0F 39		RTS		
0F10 08	OUTMA4	INX		BUMP THE POINTER
0F11 9C A1		CPX	LSTAVL	LAST AVAIL?
0F13 27 04		BEQ	SYSERR	ERROR?
0F15 EE 00		LDX	0,X	GET POINTER
0F17 20 DB		BRA	OUTMA1	

* REPORT SYSTEM MACRO ERROR

0F19 7E 08 93	SYSERR	JMP	MACOVF	REPORT OVERFLOW
---------------	--------	-----	--------	-----------------

* INPUT TO MACRO SPACE

0F1C DE 5C	INMAC	LDX	MBFPNT	SET UP POINTER
0F1E A6 00	INMAC2	LDA A	0,X	GET THE CHARACTER
0F20 08		INX		BUMP THE POINTER
0F21 DF 5C		STX	MBFPNT	SAVE IT
0F23 4D		TST A		TEST THE CHAR.

0F24 26 06		BNE	INMAC4	
0F26 EE 00		LDX	0, X	GET LINK
0F28 26 F4		BNE	INMAC2	
0F2A 20 04		BRA	INMAC5	
0F2C 81 FF	INMAC4	CMP A	#\$FF	IS CHAR FF?
0F2E 27 EE		BEQ	INMAC2	
0F30 39	INMAC5	RTS		RETURN

* CLOSE MACRO SPACE

0F31 7D 00 87	CLSMAC	TST	DIVFL2	TEST MARKER
0F34 27 09		BEQ	CLSMA2	
0F36 4F		CLR A		
0F37 97 86		STA A	DIVFLG	CLEAR FLAG
0F39 97 87		STA A	DIVFL2	
0F3B DE AB		LDX	NXTOUT	SET POINTER
0F3D 20 02		BRA	CLSMA3	
0F3F DE A9	CLSMA2	LDX	NXTMAC	POINT TO NEXT MAC
0F41 6D 00	CLSMA3	TST	0, X	TEST CHARACTER
0F43 27 14		BEQ	CLSMA4	
0F45 6D 01		TST	1, X	TEST NEXT
0F47 27 17		BEQ	CLSMA5	
0F49 6D 02		TST	2, X	ONE MORE
0F4B 27 1A		BEQ	CLSMA6	
0F4D 6F 00		CLR	0, X	CLEAR OUT SPACE
0F4F 6F 01		CLR	1, X	
0F51 6F 02		CLR	2, X	
0F53 08		INX		FIX POINTER
0F54 08		INX		
0F55 08		INX		
0F56 DF A3		STX	FSTAVL	SET FIRST AVAIL
0F58 39		RTS		RETURN
0F59 EE 01	CLSMA4	LDX	1, X	GET LINK
0F5B 26 E4		BNE	CLSMA3	
0F5D 7E 0F 19		JMP	SYSERR	REPORT MACRO ERROR
0F60 86 FF	CLSMA5	LDA A	#\$FF	SET UP FF
0F62 A7 00		STA A	0, X	SAVE IT
0F64 08		INX		
0F65 20 F2		BRA	CLSMA4	
0F67 86 FF	CLSMA6	LDA A	#\$FF	SET UP FF
0F69 A7 00		STA A	0, X	SAVE IT
0F6B 08		INX		FIX POINTER
0F6C 20 F2		BRA	CLSMA5	

* FIND MACRO

0F6E 96 9D	FNDMAC	LDA A	MACNAM	CHECK NAME
0F70 27 17		BEQ	FNDMA4	
0F72 D6 9E		LDA B	MACNAM+1	GET NAME
0F74 CE 1C 88		LDX	#MRACTBL	POINT TO TABLE
0F77 9C F7	FNDMA1	CPX	MACEND	FINISHED?
0F79 27 0E		BEQ	FNDMA4	
0F7B A1 00		CMP A	0, X	TEST 1ST CHAR.
0F7D 26 04		BNE	FNDMA2	

TSC 6800 TEXT PROCESSOR

TSC MNEMONIC ASSEMBLER PAGE 41

0F7F E1 01		CMP B 1,X	TEST 2ND CHAR.
0F81 27 08		BEQ FNDMA6	
0F83 08	FNDMA2	INX	FIX POINTER
0F84 08		INX	
0F85 08		INX	
0F86 08		INX	
0F87 20 EE		BRA FNDMA1	REPEAT
0F89 DE F7	FNDMA4	LDX MACEND	SET POINTER
0F8B 39	FNDMA6	RTS	RETURN

* FIND LAST MACRO ENTRY

0F8C A6 00	FNDLST	LDA R 0,X	GET CHARACTER
0F8E 27 03		BEQ FNDLS2	IS IT ZERO?
0F90 08		INX	GO TO NEXT
0F91 20 F9		BRA FNDLST	
0F93 08	FNDLS2	INX	BUMP POINTER
0F94 DF A9		STX NXTMAC	SAVE POSITION
0F96 EE 00		LDX 0,X	PICK UP LINK
0F98 26 F2		BNE FNDLST	
0F9A DE A9		LDX NXTMAC	GET NEXT LOC.
0F9C 39		RTS	RETURN

* CHECK LAST MACRO ENTRY

0F9D 8D ED	CHKLST	BSR FNDLST	FIND LAST ENTRY
0F9F 08		INX	FIX POINTER
0FA0 08		INX	
0FA1 9C A3		CPX FSTAVL	IS IT FIRST?
0FA3 27 04		BEQ CHKLS2	
0FA5 DE A9		LDX NXTMAC	GET NEXT
0FA7 0D		SEC	
0FA8 39		RTS	RETURN
0FA9 09	CHKLS2	DEX	BACK UP
0FAA 09		DEX	
0FAB 09		DEX	
0FAC 86 FF		LDA R #\$FF	SET UP FF
0FAD A7 00		STA R 0,X	PUT CHARACTER
0FB0 A7 01		STA R 1,X	
0FB2 A7 02		STA R 2,X	
0FB4 0C		CLC	
0FB5 39		RTS	RETURN

* END MACRO EXECUTION

0FB6 96 80	MCEND	LDA R MACCNT	GET COUNT
0FB8 27 06		BEQ MCEND2	
0FB9 7F 00 5B		CLR MBFLG	CLEAR FLAG
0FBD 7C 00 7D		INC FINMAC	SET FINISHED
0FC0 7E 08 9C	MCEND2	JMP FINCM	GO FINISH

* GET TWO CHARACTER NAME

0FC3 BD 12 11	GTNAM	JSR LDNSKP	GET TO NEXT
---------------	-------	------------	-------------

0FC6 BD 12 1F		JSR	CLSFY	CLASSIFY IT
0FC9 C1 02		CMP B	#2	
0FCB 26 10		BNE	GTNA6	
0FCD 36		PSH A		SAVE CHARACTER
0FCE 08		INX		FIX THE POINTER
0FCF A6 00		LDA A	0, X	GET CHARACTER
0FD1 BD 12 1F		JSR	CLSFY	GO CLASSIFY
0FD4 C1 02		CMP B	#2	
0FD6 33		PUL B		RESTORE CHARACTER
0FD7 26 10		BNE	GTNA6	
0FD9 08		INX		ADJUST POINTER
0FDA DF E3		STX	CNPNT	SAVE IT
0FDC C1 5F		CMP B	#\$5F	LOWER CASE?
0FDE 23 04		BLS	GTNA4	
0FE0 80 20		SUB A	#\$20	MAKE UPPER
0FE2 C0 20		SUB B	#\$20	
0FE4 D7 9D	GTNA4	STA B	MACNAM	SAVE THE NAME
0FE6 97 9E		STA A	MACNAM+1	
0FE8 39		RTS		RETURN
0FE9 4F	GTNA6	CLR A		CLEAR OUT
0FEA 5F		CLR B		
0FEB 20 F7		BRA	GTNA4	

* SET TITLE LENGTH .LT +N

0FED BD 12 53	TLEN	JSR	CHKNUM	CHECK FOR NUMBER
0FF0 24 07		BCC	TLEN2	
0FF2 96 D5		LDA A	TLN	GET LENGTH
0FF4 BD 12 42		JSR	FIXVAL	GO FIX VALUE
0FF7 97 D5		STA A	TLN	SAVE NEW
0FF9 39	TLEN2	RTS		RETURN

* DO THREE PART TITLE .TL '1'2'3'

0FFA 7F 00 B6	TITLE	CLR	TPOS	CLEAR POSITION
0FFD BD 12 11		JSR	LDNSKP	GET TO NEXT
1000 81 0D		CMP A	#\$D	C. R. ?
1002 27 F5		BEQ	TLEN2	
1004 CE 1A 54		LDX	#TTLBUF	POINT TO BUFFER
1007 7C 00 5E		INC	NOCR	SET FLAG
100A DF BA	TITLE1	STX	TTLPT	SAVE POINTER
100C BD 06 CE		JSR	GETCHR	GO GET CHAR.
100F DE BA		LDX	TTLPT	RESTORE POINTER
1011 A7 00		STA A	0, X	SAVE THE CHAR.
1013 81 0D		CMP A	#\$D	FINISHED?
1015 27 03		BEQ	TITL12	
1017 08		INX		BUMP THE POINTER
1018 20 F0		BRA	TITLE1	
101A 7F 00 5E	TITL12	CLR	NOCR	CLEAR FLAG
101D CE 19 A0		LDX	#CMNDBF	POINT TO BUFFER
1020 A6 B4	TITL15	LDA A	TTLBUF-CMNDBF, X	
1022 A7 00		STA A	0, X	PUT CHAR.
1024 08		INX		GET TO NEXT
1025 81 0D		CMP A	#\$D	FINISHED?

1027 26 F7	BNE	TITL15	
1029 CE 19 A0	LDX	#CMNDBF	RESTORE POINTER
102C A6 00	LDR A	0,X	GET CHARACTER
102E 97 B7	STA R	DELIM	SAVE DELIMITER
1030 08	INX		BUMP THE POINTER
1031 DF E3	STX	CMNPNP	SAVE IT
1033 CE 1A 54	LDX	#TTLBUF	POINT TO BUFFER
1036 DF BA	STX	TTLPTN	
1038 86 20	LDA R	#\$20	SET UP SPACE
103A A7 00	STA R	0,X	SAVE IT
103C 08	INX		BUMP POINTER
103D 8C 1A A4	CPX	#TTLBUF+80	
1040 26 F8	BNE	TITLE2	
1042 6D 10 AB	JSR	CNTTTL	GO COUNT TITLE
1045 D7 B8	STA B	TCNT	SAVE COUNT
1047 BD 10 BF	JSR	XFRRTL	TRANSFER TITLE
104A BD 10 AB	JSR	CNTTTL	COUNT TITLE
104D 96 D5	LDA R	TLN	GET LENGTH
104F 10	SBA		
1050 47	ASR A		
1051 97 B9	STA A	MCNT	SAVE MIDDLE COUNT
1053 C6 20	LDR B	#\$20	GET SPACE
1055 91 B8	CMP R	TCNT	
1057 23 0F	BLS	TITLE5	
1059 90 B8	SUB A	TCNT	
105B DE BA	LDX	TTLPTN	RESTORE POINTER
105D E7 00	STA B	0,X	SAVE CHAR.
105F 08	INX		BUMP THE POINTER
1060 7C 00 B6	INC	TPOS	UPDATE POSITION
1063 4A	DEC A		
1064 26 F7	BNE	TITLE4	
1066 DF BA	STX	TTLPTN	SAVE POINTER
1068 BD 10 BF	JSR	XFRRTL	TRANSFER TITLE
106B BD 10 AB	JSR	CNTTTL	COUNT TITLE
106E 96 D5	LDA R	TLN	GET LENGTH
1070 90 B6	SUB A	TPOS	FIX POSITION
1072 11	CBA		
1073 23 0D	BLS	TITLE7	
1075 10	SBA		
1076 C6 20	LDA B	#\$20	SET UP SPACE
1078 DE BA	LDX	TTLPTN	SET POINTER
107A E7 00	STA B	0,X	PUT CHAR
107C 08	INX		BUMP POINTER
107D 4A	DEC A		DEC THE COUNT
107E 26 FA	BNE	TITLE65	
1080 DF BA	STX	TTLPTN	SAVE POINTER
1082 BD 10 BF	JSR	XFRRTL	TRANSFER TITLE
1085 96 D5	LDA A	TLN	GET LENGTH
1087 97 B6	STA A	TPOS	SAVE POSITION
1089 27 1C	BEQ	TITLE9	
108B DE 3E	LDA B	LFM	CHECK MARGIN
108D 27 0A	BEQ	TITL78	
108F 86 20	TITL75	LDA A	SETUP SPACE
1091 37		#\$20	
		PSH B	

1092 BD 15 D9	JSR	OUTCHR	OUTPUT SPACE
1095 33	PUL B		DEC COUNT
1096 5A	DEC B		
1097 26 F6	BNE	TITL75	
1099 CE 1A 54	TITL78	LDX #TTLBUF	POINT TO TITLE
109C A6 00	TITLE8	LDA A 0, X	GET A CHARACTER
109E BD 15 D9	JSR	OUTCHR	OUTPUT IT
10A1 08	INX		GO TO NEXT
10A2 7A 00 B6	DEC	TPOS	DEC COUNT
10A5 26 F5	BNE	TITLE8	REPEAT TIL DONE
10A7 BD 08 DE	TITLE9	PCRLF	OUTPUT CR & LF
10AA 39	RTS		RETURN

* COUNT CHARACTERS IN TITLE

10AB 5F	CNTTTL	CLR B	CLEAR COUNT
10AC DE E3		LDX CMNPNT	SET POINTER
10AE A6 00	CNTTT2	LDA A 0, X	GET CHARACTER
10B0 91 B7		CMP A DELIM	IS IT DELIMITER?
10B2 27 08		BEQ CNTTT3	
10B4 81 0D		CMP A #\$D	IS IT C. R. ?
10B6 27 04		BEQ CNTTT3	
10B8 08		INX	BUMP THE POINTER
10B9 5C		INC B	BUMP COUNT
10BA 20 F2		BRA CNTTT2	
10BC DF E5	CNTTT3	STX SPCPT1	SET POINTER
10BE 39		RTS	RETURN

* TRANSFER TITLE TO BUFFER

10BF DE E3	XFR TTL	LDX CMNPNT	SET POINTER
10C1 9C E5		CPX SPCPT1	FINISHED?
10C3 27 15		BEP BMPCP2	
10C5 A6 00		LDA A 0, X	GET CHARACTER
10C7 08		INX	BUMP TO NEXT
10C8 DF E3		STX CMNPNT	SAVE
10CA DE BA		LDX TTLPN	SET POINTER
10CC A7 00		STA A 0, X	PUT CHARACTER
10CE 08		INX	BUMP TO NEXT
10CF DF BA		STX TTLPN	SAVE
10D1 7C 00 B6		INC TPOS	BUMP POSITION
10D4 20 E9		BRA XFR TTL	REPEAT
10D6 20 02	XFR TT2	BRA BMPCP2	

* BUMP COMMAND POINTER

10D8 DE E3	BMP CP	LDX CMNPNT	GET POINTER
10DA 08	BMP CP2	INX	BUMP IT
10DB DF E3		STX CMNPNT	SAVE IT
10DD 39		RTS	RETURN

* IF COMMAND . IF CONDITION CM

10DE 7F 00 7E	IF	CLR	NEGT	CLEAR FLAG
---------------	----	-----	------	------------

10E1 BD 12 11	IF1	JSR	LDNSKP	FIND NEXT CHAR
10E4 81 21		CMP A	#'!'	IS IT A '!'
10E6 26 07		BNE	IF3	
10E8 73 00 7E		COM	NEGT	SET NEG FLAG
10EB 8D EB		BSR	BMPCP	BUMP POINTER
10ED 20 F2		BRA	IF1	
10EF 81 5F	IF3	CMP A	#\$5F	IS IT LOWER CASE?
10F1 23 02		BLS	IF35	
10F3 80 20		SUB A	#\$20	MAKE UPPER
10F5 81 4F	IF35	CMP A	#'0	CHECK IF ODD
10F7 26 07		BNE	IF4	
10F9 96 69		LDA A	PGN	GET PAGE NUMBER
10FB 46		ROR A		CHECK IF ODD
10FC 24 28		BCC	IFN	
10FE 20 09		BRA	IFY	
1100 81 45	IF4	CMP A	#'E	EVEN?
1102 26 27		BNE	IF6	
1104 96 69		LDA A	PGN	GET PAGE NUMBER
1106 46		ROR A		CHECK IF EVEN
1107 25 1D		BCS	IFN	
1109 96 7E	IFY	LDA A	NEGT	CHECK NEG
110B 26 31		BNE	IF8	
110D 8D C9	IF5	BSR	BMPCP	BUMP POINTER
110F BD 12 11		JSR	LDNSKP	GET NEXT CHAR
1112 7F 00 64		CLR	NOFL	CLEAR FLAG
1115 81 2E		CMP A	#'.'	IS IT PERIOD?
1117 27 06		BEO	IF55	
1119 81 3A		CMP A	#':	IS IT COLON?
111B 26 0D		BNE	IFN2	
111D 97 64		STA A	NOFL	SET NO FLUSH
111F 08	IF55	INX		FIX POINTER
1120 DF E3		STX	CMMNPNT	SAVE IT
1122 7C 00 7F		INC	IFFLG	SET IF FLAG
1125 39		RTS		RETURN
1126 96 7E	IFN	LDA A	NEGT	CHECK NEG
1128 26 E3		BNE	IFS	
112A 39	IFN2	RTS		RETURN
112B BD 12 53	IF6	JSR	CHKNUM	CHECK FOR NUMBER
112E 24 0E		BCC	IF8	
1130 DE E3		LDX	CMMNPNT	GET POINTER
1132 09		DEX		ADJUST
1133 09		DEX		
1134 DF E3		STX	CMMNPNT	SAVE
1136 96 65		LDA A	INNUM	GET NUMBER
1138 28 EC		BMI	IFN	
113A 27 EA		BEO	IFN	
113C 20 CB		BRA	IFY	
113E 39	IF8	RTS		RETURN

* TEST FOR NEGATIVE NUMBER

113F 96 65	TSTNEG	LDA A	INNUM	GET NUMBER
1141 2A 07		BPL	TSTNE2	
1143 97 67		STH R	SIGN	SET SIGN

1145 97 66		STA A	NEG	SET NEG
1147 70 00 65		NEG	INNUM	NEGATE NUM.
114A 39	TSTNE2	RTS		RETURN
* CHANGE TRAP LOCATION . CH -M -N				
114B BD 12 53	CHNG	JSR	CHKNUM	CHECK FOR NUMBER
114E 24 19		BCC	CHNG3	
1150 8D ED		BSR	TSTNEG	NEGATIVE?
1152 96 3F		LDA A	PGL	GET PAGE LENGTH
1154 4C		INC A		
1155 BD 12 42		JSR	FIXVAL	FIX VALUE
1158 CE 19 6E		LDX	#TRAPS	POINT TO TRAPS
115B 4D		TST A		
115C 26 01		BNE	CHNG2	
115E 4C		INC A		BUMP IT
115F A1 00	CHNG2	CMP A	0, X	TEST LOCATION
1161 27 23		BEQ	CHNG5	
1163 BD 0D D0		JSR	INTRP	BUMP POS.
1166 26 F7		BNE	CHNG2	
1168 39	CHNG25	RTS		RETURN
1169 BD 0F C3	CHNG3	JSR	GTNAM	GO GET NAME
116C 96 9D		LDA A	MACNAM	
116E 27 F8		BEQ	CHNG25	
1170 D6 9E		LDA B	MACNAM+1	
1172 CE 19 6E		LDX	#TRAPS	POINT TO TRAPS
1175 A1 01	CHNG4	CMP A	1, X	CHECK CHAR.
1177 26 04		BNE	CHNG45	
1179 E1 02		CMP B	2, X	
117B 27 09		BEQ	CHNG5	
117D 08	CHNG45	INX		BUMP TO NEXT
117E 08		INX		
117F 08		INX		
1180 8C 19 9E		CPX	#TRPEND	END OF TABLE?
1183 26 F0		BNE	CHNG4	
1185 39		RTS		RETURN
1186 DF ED	CHNG5	STX	TEMP5	SAVE POINTER
1188 BD 12 53		JSR	CHKNUM	CHECK FOR NUMBER
118B 24 DB		BCC	CHNG25	
118D 8D B0		BSR	TSTNEG	IS IT NEG. ?
118F 96 3F		LDA A	PGL	GET PAGE LENGTH
1191 4C		INC A		
1192 BD 12 42		JSR	FIXVAL	FIX VALUE
1195 4D		TST A		
1196 26 01		BNE	CHNG6	
1198 4C		INC A		BUMP IT
1199 DE ED	CHNG6	LDX	TEMP5	RESTORE POINTER
119B A7 00		STA A	0, X	PUT CHAR
119D 39		RTS		RETURN

* SET NUMBER REGISTER . NR X N

119E BD 12 11	NREG	JSR	LDNSKF	GET TO NEXT
11A1 BD 12 1F		JSR	CLSFY	CLASSIFY IT

11A4 C1 02	CMP B	#2	
11A6 26 14	BNE	NREG4	
11A8 36	PSH A		SAVE
11A9 BD 10 D8	JSR	BMPCP	BUMP POINTER
11AC BD 12 53	JSR	CHKNUM	CHECK FOR NUMBER
11AF 32	PUL A		RESTORE
11B0 24 0A	BCC	NREG4	
11B2 BD 11 E1	JSR	FNDNUM	GO FIND NUMBER
11B5 A6 00	LDA A	0,X	GET CHARACTER
11B7 BD 12 42	JSR	FIXVAL	FIX VALUE
11BA A7 00	STA A	0,X	SAVE IT
11BC 39	NREG4	RTS	RETURN

* SET ARABIC MODE . AR

11BD 7F 00 C1	ARB	CLR	ROM	CLEAR ROMAN
11C0 39		RTS		RETURN

* SET FOR SMALL ROMAN . SR

11C1 86 80	SROM	LDA A	#\$80	
11C3 97 C1	ROM2	STA A	ROM	SET FLAG
11C5 39		RTS		

* SET FOR CAPITAL ROMAN . CR

11C6 86 0F	CROM	LDA A	#\$F	
11C8 20 F9		BRA	ROM2	SET FLAG

* SET AUTO INCREMENT . AU N

11CA BD 12 53	SAUTO	JSR	CHKNUM	CHECK FOR NUMBER
11CD 24 07		BCC	SAUTO4	
11CF 96 BF		LDA A	AUTO	GET OLD
11D1 BD 12 42		JSR	FIXVAL	FIX VALUE
11D4 97 BF		STA A	AUTO	SAVE NEW
11D6 39	SAUTO4	RTS		RETURN

* CLEAR NUMBER SPACE

11D7 5F	CLRNFM	CLR B		
11D8 D7 65		STA B	INNUM	CLEAR OUT NUM
11DA D7 54		STA B	INC	
11DC D7 4F		STA B	GDNFM	SET FLAGS
11DE D7 52		STA B	BNUM	
11E0 39		RTS		RETURN

* FIND NUMBER REGISTER

11E1 CE 00 30	FNDNUM	LDX	#NMREGS	SET POINTER
11E4 80 41		SUB A	#\$41	
11E6 DF 55		STX	NUMPNT	
11E8 9B 56		ADD A	NUMPNT+1	ADD OFFSET
11EA 97 56		STA A	NUMPNT+1	

11EC DE 55 11EE 39	LDX RTS	NUMPNT	GET POINTER RETURN
* FETCH NUMBER FROM BUFFER			
11EF DE 55 11F1 9C 58 11F3 27 16 11F5 A6 00 11F7 84 7F 11F9 08 11FA DF 55 11FC 81 0D 11FE 26 08 1200 7D 00 5E 1203 26 03 1205 7E 07 8D 1208 7E 07 4F 120B 7F 00 57 120E 7E 06 CE	FTCHNM CPX BEQ LDA R AND A INX STX CMP A BNE TST BNE JMP FTCHN1 FTCHN2 CLR JMP	LDX LSTNUM FTCHN2 0, X #\$7F NUMPNT #\$D FTCHN1 NOCR FTCHN1 FETCH5 FETCH3 EXCHR GETCHR	SET POINTER FINISHED? GET A CHAR. MASK IT BUMP THE POINTER SAVE IT C. R. ? TEST FLAG RETURN CLEAR EXTRA CHAR. GO GET CHAR
* LOAD POINTER AND SKIP SPACES			
1211 DE E3 1213 A6 00 1215 81 20 1217 26 03 1219 08 121A 20 F7 121C DF E3 121E 39	LDNSKP LDNSK2 LDA A CMP A BNE INX BRA LDNSK4 STX RTS	LDX LDR A #\$20 LDNSK4 LDNSK2 CMNPNT	SET POINTER GET CHARACTER IS IT SPACE? BUMP TO NEXT SAVE POSITION RETURN
* CLASSIFY CHARACTER			
121F 5F 1220 4D 1221 2B 1E 1223 81 5F 1225 23 06 1227 81 7F 1229 22 16 122B 80 20 122D 81 30 122F 25 10 1231 81 39 1233 22 02 1235 5C 1236 39 1237 81 41 1239 25 06 123B 81 5A 123D 22 02 123F C6 02 1241 39	CLSFY TST A BMI CMP A BLS CMP A BHI SUB A CMP A BLO CMP A BHI INC B RTS CLSFY2 BLO CMP A BHI LDA B CLSFY4	CLR B #\$20 CLSFY4 #\$5F CLSFY1 #\$7F CLSFY4 #\$20 #10 CLSFY4 #19 CLSFY2 #1A CLSFY4 #1Z CLSFY4 #2 RTS	CLEAR SPECIFIER TEST CHAR LOWER CASE? TEST FOR PARITY MAKE UPPER CASE CHAR A NUMBER? IF SO, SET RETURN IS CHAR A LETTER? IF SO, SET RETURN

* FIX NUMBER VALUE

1242 D6 65	FIXVAL	LDA B	INNUM	GET NUMBER
1244 7D 00 67		TST	SIGN	TEST SIGN
1247 27 03		BEQ	FIXVA4	
1249 7D 00 66		TST	NEG	TEST FOR NEG.
124C 27 01		BEQ	FIXVR3	
124E 50		NEG B		NEGATE NUM
124F 18	FIXVA3	RBA		FIX VALUE
1250 39		RTS		RETURN
1251 17	FIXVA4	TBA		
1252 39		RTS		

* CHECK FOR NUMBER

1253 4F	CHKNUM	CLR A		CLEAR FLAGS
1254 97 67		STA A	SIGN	
1256 97 66		STA A	NEG	
1258 BD 11 D7		JSR	CLRNUM	CLEAR NUMBER
125B 5C		INC B		
125C D7 5E		STA B	NOCR	SET FLAGS
125E BD 12 11		JSR	LDNSKP	GO TO NEXT
1261 81 2B		CMP A	#'+	IS IT A '+'?
1263 27 06		BEQ	CHKNU2	
1265 81 2D		CMP A	#'-	IS IT A '-'?
1267 26 0B		BNE	CHKNU4	
1269 97 66		STA A	NEG	SET NEG.
126B 08	CHKNU2	INX		BUMP THE POINTER
126C DF E3		STX	CMNPNT	SAVE IT
126E 8D 1F		BSR	PRNUM	PROCESS NUMBER
1270 24 0D		BCC	CHKNU6	
1272 20 07		BRA	CHKNU5	
1274 08	CHKNU4	INX		FIX POINTER
1275 DF E3		STX	CMNPNT	SAVE IT
1277 8D 32		BSR	PRNU28	PROCESS NUM.
1279 24 04		BCC	CHKNU6	
127B 8D 0B	CHKNU5	BSR	CLRTHM	CLEAR FLAGS
127D 0D		SEC		
127E 39		RTS		RETURN
127F 8D 07	CHKNU6	BSR	CLRTHM	CLEAR FLAGS
1281 DE E3		LDX	CMNPNT	SET POINTER
1283 09		DEX		
1284 DF E3		STX	CMNPNT	
1286 0C		CLC		
1287 39		RTS		RETURN

* CLEAR FLAGS

1288 7F 00 57	CLRTHM	CLR	EXCHR	CLEAR THEM
128B 7F 00 5E		CLR	NOCR	
128E 39		RTS		RETURN

* PROCESS NUMBER

128F 97 67	PRNUM	STA A	SIGN	CLEAR SIGN
1291 BD 11 D7	PRNUM2	JSR	CLRNUM	CLEAR NUMBER
1294 5C		INC B		
1295 D7 5E		STA B	NOCR	SET FLAGS
1297 7F 00 54	PRNU27	CLR	INC	
129A 7C 00 6A		INC	PASCHR	
129D BD 06 CE		JSR	GETCHR	GET NEXT CHAR.
12A0 7D 00 74		TST	NOEXP	DO EXPRESSIONS?
12A3 27 06		BEQ	PRNU29	
12A5 7F 00 74		CLR	NOEXP	
12A8 7E 13 52		JMP	PRNU82	JUMP AHEAD
12AB BD 12 1F	PRNU29	JSR	CLSFY	GO CLASSIFY
12AE C1 01		CMP B	#1	
12B0 25 05	PRNUM3	BLO	PRNU31	
12B2 27 51		BEQ	PRNUM5	
12B4 7E 13 40		JMP	PRNU73	
12B7 7D 00 65	PRNU31	TST	INNUM	TEST NUMBER
12BA 27 05		BEQ	PRNU32	
12BC 36		PSH A		
12BD 96 65		LDA A	INNUM	GET NUMBER
12BF 20 58		BRA	PRNUM6	
12C1 7F 00 65	PRNU32	CLR	INNUM	CLEAR NUMBER
12C4 81 23		CMP A	#'#	CHECK FOR '#'
12C6 27 18		BEQ	PRNUM4	
12C8 81 2B		CMP A	#'+	IS IT '+'?
12CA 26 04		BNE	PRNU35	
12CC 97 50		STA A	ADD	SET FOR ADD
12CE 20 C7		BRA	PRNU27	
12D0 81 2D	PRNU35	CMP A	#'-	IS IT '-'?
12D2 26 04		BNE	PRNU37	
12D4 97 51		STA A	SUB	SET FOR SUBTRACT
12D6 20 BF		BRA	PRNU27	
12D8 81 25	PRNU37	CMP A	#'%	IS IT '%'?
12DA 26 6C		BNE	PRNUM8	
12DC 96 69		LDA A	PGN	GET PAGE NUMBER
12DE 20 39		BRA	PRNUM6	
12E0 7C 00 6A	PRNUM4	INC	PASCHR	SET FLAG
12E3 BD 06 CE		JSR	GETCHR	GET CHARACTER
12E6 BD 12 1F		JSR	CLSFY	CLASSIFY IT
12E9 C1 02		CMP B	#2	
12EB 26 10		BNE	PRNU45	
12ED BD 11 E1		JSR	FNDNUM	GO FIND NUMBER
12F0 A6 00		LDA A	0, X	GET VALUE
12F2 7D 00 54		TST	INC	INCREMENT?
12F5 27 22		BEQ	PRNUM6	
12F7 98 BF		ADD A	AUTO	ADD IN AUTO
12F9 A7 00		STA A	0, X	SAVE NEW
12FB 20 1C		BRA	PRNUM6	
12FD 81 2B	PRNU45	CMP A	#'+	IS IT '+'?
12FF 26 47		BNE	PRNUM8	
1301 97 54		STA A	INC	SET INC.
1303 20 DB		BRA	PRNUM4	
1305 80 30	PRNUM5	SUB A	#\$30	BIAS NUMBER

1307 36		PSH R	
1308 D6 65		LDA B	INNUM GET NUM
130A 58		ASL B	ADJUST
130B 58		ASL B	
130C DB 65		ADD B	INNUM ADD IT IN
130E 58		ASL B	
130F 32		PUL R	RESTORE
1310 1B		ABA	
1311 97 65		STA A	INNUM SAVE NEW VALUE
1313 7C 00 4F		INC	GNUM SET GOOD
1316 7E 12 97		JMP	PRNU27 REPEAT
1319 D6 51	PRNUMS	LDA B	SUB SUBTRACT?
131B 27 06		BEQ	PRNU65
131D 16		TAB	DO SUBTRACT
131E 96 52		LDA A	BNUM
1320 10		SBA	
1321 20 06		BRA	PRNUM7
1323 D6 50	PRNU65	LDA B	ADD ADDITION?
1325 27 02		BEQ	PRNUM7
1327 9B 52		ADD A	BNUM DO ADD
1329 97 52	PRNUM7	STA A	BNUM SAVE NUMBER
132B 7F 00 50		CLR	ADD CLEAR FLAGS
132E 7F 00 51		CLR	SUB
1331 7C 00 4F		INC	GNUM SET GOOD
1334 7D 00 65		TST	INNUM TEST NUMBER
1337 26 03		BNE	PRNU72
1339 7E 12 97		JMP	PRNU27
133C 32	PRNU72	PUL R	RESTORE CHAR
133D 7E 12 C1		JMP	PRNU32
1340 7D 00 65	PRNU73	TST	INNUM TEST NUMBER
1343 27 03		BEQ	PRNUM8
1345 36		PSH A	
1346 20 D1		BRA	PRNUM6
1348 7F 00 74	PRNUM8	CLR	NOEXP CLEAR FLAG
134B 7D 00 4F		TST	GNUM TEST GOOD
134E 26 02		BNE	PRNU02
1350 0C		CLC	
1351 39		RTS	SET CONDITION
1352 97 57	PRNU82	STA A	EXCHR RETURN
1354 CE 01 25		LDX	#NUM SAVE EXTRA CHAR.
1357 96 52		LDA A	BNUM POINT TO NUMBER
1359 97 65		STA A	INNUM GET NUMBER
135B 27 04		BEQ	BTOD
135D D6 C1		LDA B	ROM ROM OR ARABIC?
135F 26 37		BNE	BTOROM

* BINARY TO ASCII ARABIC

1361 5F	BTOD	CLR B	
1362 81 64	BTOD1	CMP A	#100 NUM > 100?
1364 25 05		BLD	BTOD2
1366 80 64		SUB A	#100 SUB OFF 100
1368 5C		INC B	BUMP NUMBER
1369 20 F7		BRA	BTOD1

136B 5D		BTOD2	TST B	ANY YET?
136C 27 06			BEQ BTOD3	
136E CB 30			ADD B #\$\$30	SET HUNDREDS
1370 E7 00			STA B 0,X	SAVE
1372 08			INX	GO TO NEXT
1373 5F			CLR B	CLEAR REGISTER
1374 81 0A	BTOD3		CMP A #10	NUMBER > 10
1376 25 05			BLO BTOD4	
1378 80 0A			SUB A #10	SUB VALUE
137A 5C			INC B	BUMP NUMBER
137B 20 F7			BRA BTOD3	
137D 5D	BTOD4		TST B	ANY?
137E 27 05			BEQ BTOD45	
1380 CB 30			ADD B #\$\$30	ADD BIRS
1382 E7 00			STA B 0,X	SAVE TENS
1384 08			INX	BUMP TO NEXT
1385 BB 30	BTOD45		ADD A #\$\$30	ADD IN BIRS
1387 A7 00			STA A 0,X	SAVE ONES
1389 08			INX	BUMP POINTER
138A 96 57	BTOD5		LDA A EXCHR	GET EXTRA
138C A7 00			STA A 0,X	SAVE IT
138E 08			INX	BUMP TO NEXT
138F DF 58			STX LSTNUM	SAVE POSITION
1391 CE 01 25			LDX #NUM	POINT TO NUMBER
1394 DF 55			STX NUMPNT	
1396 0D			SEC	
1397 39			RTS	RETURN

* BINARY TO ASCII ROMAN

1398 C6 43		BTOROM	LDA B #'C	SET HUNDREDS
139A 81 64		BTOR01	CMP A #100	NUMBER > 100?
139C 25 07			BLO BTOR02	
139E 80 64			SUB A #100	SUBTRACT OFF
13A0 E7 00			STA B 0,X	SET 100
13A2 08			INX	BUMP TO NEXT
13A3 20 F5			BRA BTOR01	
13A5 81 5A	BTOR02		CMP A #90	CHECK FOR 90
13A7 25 0A			BLO BTOR03	
13A9 80 5A			SUB A #90	SUBTRACT OFF
13AB E7 01			STA B 1,X	PUT CHARACTER
13AD C6 58			LDA B #'X	SET TENS
13AF E7 00			STA B 0,X	SAVE IT
13B1 08			INX	BUMP TO NEXT
13B2 08			INX	
13B3 81 32	BTOR03		CMP A #50	CHECK FOR FIFTY
13B5 25 07			BLO BTOR04	
13B7 80 32			SUB A #50	SUBTRACT OFF
13B9 C6 4C			LDA B #'L	SET 'L'
13BB E7 08			STA B 0,X	SAVE IT
13BD 08			INX	BUMP THE POINTER
13BE 81 28	BTOR04		CMP A #40	CHECK FOR 40
13C0 25 0C			BLO BTOR05	
13C2 80 28			SUB A #40	SUBTRACT OFF

13C4 C6 58		LDA B #1X	SET TEN
13C6 E7 00		STA B 0,X	SAVE IT
13C8 C6 4C		LDA B #1L	SET 50
13CA E7 01		STA B 1,X	SAVE IT
13CC 08		INX	BUMP TO NEXT
13CD 08		INX	
13CE C6 58	BTOR05	LDA B #1X	SET UP 'X'
13D0 81 0A		CMP A #10	CHECK TENS
13D2 25 07		BLO BTOR06	
13D4 80 0A		SUB A #10	SUBTRACT OFF
13D6 E7 00		STR B 0,X	SAVE
13D8 08		INX	BUMP POINTER
13D9 20 F3		BRA BTOR05	
13DB 81 09	BTOR06	CMP A #9	CHECK IF 9
13DD 25 0A		BLO BTOR65	
13DF 80 09		SUB A #9	SUBTRACT 9
13E1 E7 01		STA B 1,X	SAVE CHARACTER
13E3 C6 49		LDA B #'I	
13E5 E7 00		STA B 0,X	
13E7 08		INX	GET TO NEXT
13E8 08		INX	
13E9 81 05	BTOR65	CMP A #5	CHECK FOR 5
13E8 25 07		BLO BTOR07	
13ED C6 56		LDA B #'V	SET UP 'V'
13EF E7 00		STA B 0,X	SAVE IT
13F1 08		INX	BUMP POINTER
13F2 80 05		SUB A #5	FIX VALUE
13F4 81 04	BTOR07	CMP A #4	CHECK FOR 4
13F6 25 0C		BLO BTOR08	
13F8 80 04		SUB A #4	SUBTRACT OFF
13FA C6 49		LDA B #'I	SET UP 'I'
13FC E7 00		STA B 0,X	SAVE CHARACTER
13FE C6 56		LDA B #'V	
1400 E7 01		STA B 1,X	SAVE 'V'
1402 08		INX	BUMP POINTER
1403 08		INX	
1404 C6 49	BTOR08	LDA B #'I	
1406 4D		TST A	TEST ONES
1407 27 06		BEQ BTOR09	
1409 E7 00		STA B 0,X	SAVE I'S
140B 08		INX	
140C 4A		DEC A	DONE?
140D 20 F5		BRA BTOR08	
140F DF 58	BTOR09	STX LSTNUM	SAVE POINTER
1411 96 C1		LDA A ROM	CHECK IF SMALL
1413 2A 0E		BPL BTODON	
1415 CE 01 25		LDX #NUM	RESET POINTER
1418 A6 00	BTOR92	LDA A 0,X	GET CHARACTER
141A 8B 20		ADD A #\$20	MAKE SMALL
141C A7 00		STA A 0,X	PUT BACK
141E 08		INX	BUMP TO NEXT
141F 9C 58		CPX LSTNUM	FINISHED?
1421 26 F5		BNE BTOR92	
1423 7E 13 8A	BTODON	JMP BTOD5	

* PUSH X ONTO STACK

1426 32	PUSHX	PUL A	GET RETURN ADR.
1427 33		PUL B	
1428 97 F1		STA A RETREG	SAVE IT
142A D7 F2		STA B RETREG+1	
142C DF F3		STX INDEX	SAVE X
142E 96 F3		LDA A INDEX	GET PART X
1430 D6 F4		LDA B INDEX+1	
1432 36		PSH A .	PUSH ON STACK
1433 37		PSH B	
1434 DE F1	PUSH4	LDX RETREG	GET RETURN
1436 6E 00		JMP 0, X	RETURN

* PULL X FROM STACK

1438 32	PULLX	PUL A	GET RETURN ADR.
1439 33		PUL B	
143A 97 F1		STA A RETREG	SAVE IT
143C D7 F2		STA B RETREG+1	
143E 33		PUL B	PULL X
143F 32		PUL A	
1440 97 F3		STA A INDEX	SAVE X
1442 D7 F4		STA B INDEX+1	
1444 96 F1		LDA A RETREG	GET RETURN ADR.
1446 D6 F2		LDA B RETREG+1	
1448 37		PSH B	PUSH BACK ON
1449 36		PSH A	
144A DE F3		LDX INDEX	LOAD UP X
144C 39		RTS	RETURN

* UNDERLINE COMMAND . UL

144D 86 01	UNDL	LDA A #1	SET UL FLAG
144F 97 4E		STA A ULFLG	
1451 39		RTS	RETURN

*

* DISK COMMANDS FOLLOW

* READ ITEM . RI [SI]

1452 96 4D	RDIT	LDA A FIOPEN	CHECK IF OPEN
1454 27 18		BEQ RDIT4	
1456 7F 00 89		CLR CRSUP	CLEAR SUP FLAG
1459 7F 00 36		CLR GCNT	CLEAR CHAR COUNT
145C 97 88		STA A RIFLG	SET FLAG
145E BD 12 11		JSR LDNSKP	SKIP JUNK
1461 BD 12 1F		JSR CLSFY	CHECK CHARACTER
1464 C1 02		CMP B #2	IS IT A LETTER?
1466 26 06		BNE RDIT4	
1468 81 53		CMP A #'S	IS IT AN 'S'?
146A 26 02		BNE RDIT4	

146C 97 89		STA A	CRSUP	SET SUP FLAG
146E 39	RDTIT4	RTS		RETURN

* SET ITEM CHARACTER . IC C

146F BD 12 11	ITMCH	JSR	LDNSKP	GET NEXT
1472 81 0D		CMP A	#\$D	END OF LINE?
1474 26 02		BNE	ITMCH2	
1476 86 3E		LDA A	#'>	
1478 97 4C	ITMCH2	STA A	ITEM	SET CHARACTER
147A 39		RTS		RETURN

* NEXT ITEM . NI N

147B 96 4D	NXTI	LDA A	FILOPN	FILE OPEN?
147D 27 21		BEQ	NXTI6	
147F BD 12 53		JSR	CHKNUM	LOOK FOR NUMBER
1482 96 65		LDA A	INNUM	
1484 26 01		BNE	NXTI2	
1486 4C		INC A		SET UP ONE
1487 97 8A	NXTI2	STA A	NCOUNT	SET ITEM COUNT
1489 96 4B	NXTI3	LDA A	EORF	CHECK IF EOR
148B 26 13		BNE	NXTI6	
148D 4C		INC A		SET NON ZERO
148E 97 88		STA A	RIFLG	SET FLAG
1490 BD 15 FC	NXTI4	JSR	INCHR	GET CHARACTER
1493 96 34		LDA A	EOFF	EOF?
1495 26 09		BNE	NXTI6	
1497 96 4A		LDA A	EOIF	EOI?
1499 27 F5		BEQ	NXTI4	REPRT TIL FOUND
149B 7A 00 8A		DEC	NCOUNT	DEC THE COUNT
149E 26 E9		BNE	NXTI3	
14A0 39	NXTI6	RTS		RETURN

* NEXT BLOCK . NB N

14A1 96 4D	NXTB	LDA A	FILOPN	FILE OPEN?
14A3 27 26		BEQ	NXTB6	
14A5 BD 12 53		JSR	CHKNUM	LOOK FOR NUMBER
14A8 96 65		LDA A	INNUM	
14AA 26 01		BNE	NXTB2	
14AC 4C		INC A		SET DEFAULT
14AD 97 8A	NXTB2	STA A	NCOUNT	SET COUNTER
14AF 7D 00 4B		TST	EORF	CHECK FOR EOR
14B2 26 0F		BNE	NXTB5	
14B4 86 01	NXTB4	LDA A	#1	SET FLAG
14B6 97 88		STA A	RIFLG	
14B8 BD 15 FC		JSR	INCHR	GET CHARACTER
14BB 96 34		LDA A	EOFF	CHECK EOF
14BD 26 0C		BNE	NXTB6	
14BF 96 4B		LDA A	EORF	CHECK EOR
14C1 27 F1		BEQ	NXTB4	
14C3 7F 00 4B	NXTB5	CLR	EORF	CLEAR FLAG

14C6 7A 00 8A		DEC	NCOUNT	DEC THE COUNT
14C9 26 E9		BNE	NXTB4	REPEAT TIL DONE
14CB 39	NXTB6	RTS		RETURN
 * CLOSE FILE .CF				
14CC 96 4D	CLSFL	LDA A	FILOPN	CHECK IF OPEN
14CE 27 0D		BEQ	CLSFL4	
14D0 CE 1B C8	CLSFL2	LDX	#DFCB	
14D3 86 04		LDA A	#4	SET FOR READ CLOSE
14D5 A7 00		STA A	0,X	
14D7 BD 16 6B		JSR	DOFMS	CALL FMS
14DA 7F 00 4D		CLR	FILOPN	CLEAR STATUS
14DD 39	CLSFL4	RTS		
 * OPEN FILE .OF [NAME]				
*				
14DE 96 4D	OPNF	LDA A	FILOPN	CHECK IF OPEN
14E0 26 43		BNE	OPNF8	
14E2 FE 70 94		LDX	DBUFP	GET BUFFER POINTER
14E5 DF 8E		STX	XTEMP2	SAVE POINTER
14E7 BD 12 11		JSR	LDNSKP	GET NEXY
14ER 81 0D		CMP A	#\$D	IS IT CR?
14EC 27 05		BEQ	OPNF4	
14EE FF 70 94		STX	DBUFP	SET POINTER
14F1 20 0F		BRA	OPNF5	
14F3 CE 17 5E	OPNF4	LDX	#NMST	POINT TO STRING
14F6 BD 15 BA		JSR	PSTRNG	OUTPUT IT
14F9 BD 15 59		JSR	GIBUF	GET RESPONSE
14FC CE 19 0A		LDX	#SBUF	GET POINTER
14FF FF 70 94		STX	DBUFP	SET POINTER
1502 CE 1B C8	OPNF5	LDX	#DFCB	POINT TO FCB
1505 BD 71 27		JSR	GETFIL	GET FILE NAME
1508 25 1C		BCS	OPNF6	ERROR?
150A DE 8E		LDX	XTEMP2	RESTORE POINTER
150C FF 70 94		STX	DBUFP	RESTORE BUFFER PNTR
150F CE 1B C8		LDX	#DFCB	POINT TO FCB
1512 86 01		LDA A	#1	TXT CODE
1514 BD 71 2D		JSR	SETEXT	SET EXTENSION
1517 CE 1B C8		LDX	#DFCB	POINT TO FCB
151A 86 01		LDA A	#1	OPEN FOR READ
151C A7 00		STA A	0,X	
151E BD 16 6B		JSR	DOFMS	CALL FMS
1521 86 01		LDA A	#1	
1523 97 4D		STA A	FILOPN	SET FLAG
1525 39	OPNF8	RTS		RETURN
1526 7E 03 49	OPNF6	JMP	DPROC3	
 * FIX WIDTH				
1529 96 B1	FIXWD	LDA A	TLLN	GET TEMP LENGTH
152B 9B 3B		ADD A	LLN	ADD TO LENGTH

152D 97 3B		STA A	LLN	SAVE NEW
152F 96 B0		LDA A	TIND	GET TEMP IND.
1531 9B 38		ADD A	IND	ADD TO INDENT
1533 97 38		STA A	IND	SAVE NEW
1535 96 AF		LDA A	TSIN	GET TEMP SIND.
1537 9B 71		ADD A	SIN	ADD TO SIND.
1539 97 71		STA A	SIN	SAVE NEW
153B 4F		CLR A		CLEAR OLD VALUES
153C 97 B1		STA A	TLLN	
153E 97 B0		STA A	TIND	
1540 97 AF		STA A	TSIN	
1542 96 3B		LDA A	LLN	GET LINE LENGTH
1544 90 38		SUB A	IND	SUB INDENT
1546 90 71		SUB A	SIN	SUB S IND.
1548 81 0E		CMP A	#14	LESS THAN 15?
154A 22 02		BHI	FIXWD2	
154C 86 0F	FIXWD1	LDA A	#15	FORCE TO 15
154E 81 96	FIXWD2	CMP A	#150	>150?
1550 23 02		BLS	FIXWD3	
1552 86 96		LDA A	#150	
1554 97 C3	FIXWD3	STA A	WIDTH	SAVE NEW WIDTH
1556 7E 06 97		JMP	FIXBFE	GO FIX

* GET INPUT CHARACTERS

1559 CE 19 0A	GIBUF	LDX	#\$BUF	POINT TO BUFFER
155C 5F		CLR B		CLEAR COUNT
155D 37	GIBUF2	PSH B		
155E BD 15 CF		JSR	EINCH	GET CHARACTER
1561 33		PUL B		
1562 B1 70 81		CMP A	DDEL	IS IT DEL CHR?
1565 27 26		BEQ	GIBUF6	
1567 81 0D		CMP A	#\$D	C. R. ?
1569 27 0F		BEQ	GIBUF4	
156B B1 70 80		CMP A	DBSP	IS IT BACKSPACE?
156E 27 14		BEQ	GIBUF5	
1570 81 1F		CMP A	#\$1F	CONTROL CHAR. ?
1572 23 E9		BLS	GIBUF2	
1574 5C		INC B		BUMP THE COUNT
1575 A7 00		STA A	0, X	PUT CHARACTER
1577 08		INX		BUMP THE POINTER
1578 20 E3		BRA	GIBUF2	REPEAT
157A A7 00	GIBUF4	STA A	0, X	PUT CHARACTER
157C CE 19 0A		LDX	#\$BUF	FIX POINTER
157F DF E3		STX	CMNPNT	SAVE IT
1581 D7 36		STA B	GCNT	SAVE COUNT
1583 39		RTS		RETURN
1584 8C 19 0A	GIBUF5	CPX	#\$BUF	BUFFER BEGIN?
1587 27 04		BEQ	GIBUF6	
1589 09		DEX		DEC THE POINTER
158A 5A		DEC B		DEC THE COUNT
158B 20 D0		BRA	GIBUF2	REPEAT
158D CE 16 E4	GIBUF6	LDX	#QUSTR	POINT TO STRING
1590 BD 15 BA		JSR	PSTRNG	OUTPUT IT

1593 20 C4	BRA	GIBUF	
* TEST FOR BREAK			
1595 B6 70 A1	TSTBRK	LDA A ACIAFL	CHECK IF ACIA
1598 27 06		BEQ TSTBR2	
159A B6 80 04		LDA A \$8004	GET STATUS
159D 44		LSR A	CHECK
159E 25 01		BCS TSTBR4	
15A0 39	TSTBR2	RTS	RETURN
15A1 B6 80 05	TSTBR4	LDA A \$8005	GET CHARACTER
15A4 84 7F		AND A #\$7F	MASK CHAR
15A6 81 03		CMP A #3	IS IT $\dagger C$?
15A8 26 F6		BNE TSTBR2	
15AA CE 17 14		LDX #BRKSTR	POINT TO STRING
15AD 7E 0D 0E		JMP STOP1	OUTPUT IT
* OUTPUT A C. R. AND L. F.			
15B0 7C 70 A3	CRLF	INC OUTCHN	SET OUT CHAN
15B3 BD 71 1E		JSR DPCRLF	DO CR & LF
15B6 7F 70 A3		CLR OUTCHN	CLEAR OUT CHN
15B9 39	CRLF5	RTS	RETURN
* PRINT STRING			
15BA 8D F4	PSTRNG	BSR CRLF	OUTPUT CR & LF
* PRINT DATA			
15BC 7C 70 A3	PDATA	INC OUTCHN	SET OUT
15BF A6 00	PDATA1	LDA A 0,X	GET A CHARACTER
15C1 81 04		CMP A #4	IS IT TERM?
15C3 27 06		BEQ PDATA2	
15C5 BD 02 03		JSR OUTCH	OUTPUT IT
15C8 88		INX	MOVE TO NEXT
15C9 20 F1		BRA PDATA	REPEAT
15CB 7F 70 A3	PDATA2	CLR OUTCHN	
15CE 39		RTS	RETURN
* ECHO INPUT CHARACTER			
15CF 7C 70 A3	EINCH	INC OUTCHN	SET CHAN
15D2 BD 02 06		JSR INCH	GET CHARACTER
15D5 7F 70 A3		CLR OUTCHN	CLEAR CHAN
15D8 39		RTS	RETURN
* OUTPUT CHARACTER			
15D9 D6 86	OUTCHR	LDA B DIVFLG	DIVERTING?
15DB 27 06		BEQ OUTCH2	
15DD 7C 00 87		INC DIVFL2	SET FLAG
15E0 7E 0E E7		JMP OUTMAC	OUT TO MACRO
15E3 D6 6E	OUTCH2	LDA B NOOUT	DO OUTPUT?

15E5 27 01		BEQ	OUTCH3	
15E7 39		RTS		RETURN
15E8 40	OUTCH3	TST A		CHECK PARITY
15E9 2A 0C		BPL	DOOUT	
15EB 81 A0		CMP A	#\$A0	IS IT SPACE
15ED 27 08		BEQ	DOOUT	
15EF 8D 06		BSR	DOOUT	OUTPUT CHAR
15F1 86 08		LDA A	#\$8	SET UP BACKSPACE
15F3 8D 02		BSR	DOOUT	OUTPUT IT
15F5 86 5F		LDA A	#\$`	SETUP UNDER LINE
15F7 84 ?F	DOOUT	AND A	#\$?F	MASK CHAR.
15F9 7E 02 03		JMP	OUTCH	OUTPUT CHAR

* INPUT A CHARACTER

15FC 37	INCHR	PSH B		
15FD DF F5		STX. XTEMP		
15FF 7D 00 88		TST RIFLG		
1602 26 0C		BNE DATIN		
1604 CE 1B 00		LDX #TFCB		POINT TO TEXT FCB
1607 DF 8C		STX INFBCB		SET INPUT FCB
1609 BD 16 60		JSR DREAD		DO DISK READ
160C DE F5	INCHR2	LDX XTEMP		RESTORE X
160E 33		PUL B		RESTORE B
160F 39		RTS		RETURN

* DATA IN FROM DISK

1610 CE 1B C8	DATIN	LDX #DFCB		
1613 DF 8C		STX INFBCB		SET DATA FCB
1615 96 48		LDA A EORF		CHECK FOR EOR
1617 27 08		BEQ DATIN3		
1619 7F 00 88	DATIN2	CLR RIFLG		CLEAR MODE
161C 33	DATI25	PUL B		RESTORE REGS
161D DE F5		LDX XTEMP		
161F 20 DB		BRA INCHR		DO IN CHAR
1621 BD 16 60	DATIN3	JSR DREAD		DO DISK READ
1624 26 00		BNE DATI35		EOF?
1626 91 4C		CMP A ITEM		IS IT ITEM CHAR?
1628 26 24		BNE DATIN6		
162A 7D 00 4A		TST EOIF		TST EOI FLAG
162D 27 09		BEQ DATIN4		
162F 86 01		LDA A #1		SET EOP FLAG
1631 97 4B		STA A EORF		
1633 4F	DATI35	CLR A		
1634 97 89		STA A RIFLG		CLEAR MODE
1636 20 D4		BRA INCHR2		RETURN
1638 86 01	DATIN4	LDA A #1		
163A 97 4A		STA A EOIF		SET EOI FLAG
163C 96 89		LDA A CRSUP		SUP ON?
163E 27 07		BEQ DATIN5		
1640 97 BC		STA A ENDLIN		SET END LINE
1642 7F 00 89		CLR CRSUP		CLEAR FLAG
1645 20 D2		BRA DATIN2		

1647 86 0D	DATIN5	LDA A	#\$D	SETUP CR
1649 7F 00 88		CLR	RIFLG	CLEAR MODE
164C 20 0E		BRA	DATIN8	
164E D6 4A	DATIN6	LDA B	EOIF	CHECK EOI
1650 27 07		BEQ	DATIN7	
1652 81 0D		CMP A	#\$D	IS IT CR?
1654 27 C6		BEQ	DATI25	
1656 7F 00 4A		CLR	EOIF	
1659 7C 00 36	DATIN7	INC	GCNT	BUMP CHAR COUNTER
165C 33	DATIN8	PUL B		RESTORE B
165D DE F5		LDX	XTEMP	RESTORE X
165F 39		RTS		RETURN

* DISK READ CHARACTER

1660 DE 8C	DREAD	LDX	INFCB	SET FCB
1662 8D 07		BSR	DOFMS	CALL FMS
1664 27 04		BEQ	DREAD2	ERRORS?
1666 C6 01		LDA B	#1	SET FLAG
1668 D7 34		STA B	EOF	SET EOF
166A 39	DREAD2	RTS		RETURN

* DO FMS CALL

166B BD 78 06	DOFMS	JSR	FMS	CALL FMS
166E 26 01		BNE	DOFMS2	ERROR?
1670 39		RTS		RETURN
1671 A6 01	DOFMS2	LDA A	1,X	
1673 81 08		CMP A	#8	IS IT EOF?
1675 26 03		BNE	DOFMS4	
1677 86 1A		LDA A	#\$1A	SET EOF CHAR
1679 39		RTS		RETURN
167A A6 01	DOFMS4	LDA A	1,X	GET ERROR CODE
167C 81 04		CMP A	#4	NO FILE?
167E 26 08		BNE	DOFMS6	
1680 CE 17 38		LDX	#NFST	POINT TO STRING
1683 BD 15 BA		JSR	PSTRNG	OUTPUT IT
1686 20 03		BRA	DOFMS7	
1688 BD 71 3C	DOFMS6	JSR	RPTERR	REPORT ERROR
168B BD 78 03	DOFMS7	JSR	FMSCLS	CLOSE FMS
168E 7E 02 09		JMP	MON	

* REWIND FILE

1691 CE 1B 08	RWND	LDX	#TFCB	POINT TO FCB
1694 86 05		LDA A	#5	REWIND FILE
1696 A7 00		STA A	0,X	
1698 BD 16 6B		JSR	DOFMS	CALL FMS
169B 39		RTS		RETURN

* STRINGS

169C 44	DATSTR	FCC	'DATE (MM:DD:YY)? '	
16AD 04		FCB	4	

16E8 54	PRQU	FCC	'TYPE P FOR PRINTER... '
16C4 04		FCB	4
16C5 0D	CRLFST	FCB	\$D, \$A, 0, 0, 0, 0, 4
16CC 50	PGSTR	FCC	'PAGE LIMITS? '
16D9 04		FCB	4
16DA 07	STPSTR	FCB	7
16DB 53		FCC	'STOP... '
16E2 07		FCB	7, 4
16E4 3F	QUSTR	FCC	'? '
16E6 07		FCB	7, 4
16E8 4C	LPPSTR	FCC	'LINES PER SCREEN? '
16FA 04		FCB	4
16FB 2A	OVFSTR	FCC	'***** MACRO OVERFLOW *****'
1713 04		FCB	4
1714 2A	BRKSTR	FCC	'* PROGRAM BREAK *'
1725 04		FCB	4
1726 49	ILFN	FCC	'ILLEGAL FILE NAME'
1737 04		FCB	4
1738 4E	NFST	FCC	'NO SUCH FILE'
1744 04		FCB	4
1745 43	CHST	FCC	'CHANGE DISKS AND HIT KEY'
175D 04		FCB	4
175E 44	NMST	FCC	'DATA FILE NAME? '
176E 04		FCB	4
176F 4D	MACST	FCC	'MACRO'
1774 00		FCB	0, 0, 0
1777 54		FCC	'TXT'

* BUFFER STORAGE AREA

1778	LINBUF	RMB	155
1815	EXTBUF	RMB	45
1842	LINBU2	RMB	200
190A	SBUF	RMB	100
196E	TRAPS	RMB	48
199E	TRPEND	RMB	2
19A0	CMNDBF	RMB	180
1A54	TTLBUF	RMB	180
1B08	TFCB	RMB	192
1BC8	DFCB	RMB	192
1C88	MACTBL	RMB	256
1D88	MTEND	RMB	2
1D8A	MACROS	EQU	*
2EFD	LMACRO	EQU	\$2EFD
2EFF	LAST	EQU	\$2EFF
2F00	USER	EQU,	\$2F00
	END	START	

NO ERROR(S) DETECTED

SYMBOL TABLE:

ACIAFL	70A1	ADD	0050	ADJS35	048A	ADJS55	04R9	ADJSPI2	047B
ADJSP3	0487	ADJSP4	0494	ADJSP5	049F	ADJSP6	04AD	ADJSP7	04B4
ADJSP8	04BF	ADJSP9	04C6	ADJSPC	0475	APMAC	0E15	ARB	11BD
ATFLG	0061	ATL	0D8B	ATL1	0DA0	ATL2	0DA3	ATL3	0DB0
ATL35	0DB8	ATL4	0DB9	ATL45	0DC1	ATL5	0DC8	AUTO	00BF
BMPCP	10D8	BMPCP2	10DA	BNUM	0052	BRAK	0948	BRKSTR	1714
BTOD	1361	BTOD1	1362	BTOD2	136B	BTOD3	1374	BTOD4	137D
BTOD45	1385	BTOD5	138A	BTODON	1423	BTOR65	13E9	BTOR92	1418
BTOR01	139A	BTOR02	13A5	BTOR03	13B3	BTOR04	13BE	BTOR05	13CE
BTOR06	13DB	BTOR07	13F4	BTOR08	1404	BTOR09	140F	BTOR0M	1398
BUFEND	00DB	BUFPNT	00D7	CALMA2	08BB	CALMAC	08B6	CAP	00B4
CAPIT	080A	CAPIT2	0810	CENTE2	0C7A	CENTE4	0C7E	CENTER	0C67
CENTJ	0603	CHKLS2	0FA9	CHKLST	0F9D	CHKNU2	126B	CHKNU4	1274
CHKNU5	127B	CHKNU6	127F	CHKNUM	1253	CHNG	114B	CHNG2	115F
CHNG25	1168	CHNG3	1169	CHNG4	1175	CHNG45	117D	CHNG5	1186
CHNG6	1199	CHST	1745	CLRGET	06CB	CLRNUM	11D7	CLRSP2	0255
CLRSP4	0260	CLRSPC	0251	CLRTHM	1288	CLSFL	14CC	CLSFL2	14D0
CLSFL4	14DD	CLSFY	121F	CLSFY1	122D	CLSFY2	1237	CLSFY4	1241
CLSMA2	0F3F	CLSMA3	0F41	CLSMA4	0F59	CLSMA5	0F60	CLSMA6	0F67
CLSMAC	0F31	CMFLG	005A	CMNDBF	19A0	CMNDT	098D	CMMNPNT	00E3
CNJ	00CB	CNTFLG	00D1	CNTRI4	067E	CNTRI5	0684	CNTRI6	0694
CNTRIT	066A	CNTSP2	0661	CNTSP3	0669	CNTSPC	065C	CNTTT2	10AE
CNTTT3	10BC	CNTTTL	10AB	COLCN2	0078	COLCNT	0032	COMAN2	082A
COMAN3	084A	COMAN4	085A	COMAN5	085D	COMAN6	0869	COMAN7	086D
COMAN8	0881	COMAN9	088D	COMAND	0824	CRF	00F9	CRLF	15B0
CRLF5	15B9	CRLFST	16C5	CROM	11C6	CRSUP	0089	DATE	708E
DATI25	161C	DATI35	1633	DATIN	1610	DATIN2	1619	DATIN3	1621
DATIN4	1638	DATIN5	1647	DATING6	164E	DATINT7	1659	DATINS	165C
DATSTR	169C	DAY	0033	DBSP	7080	DBUFP	7094	DDEL	7081
DEFM35	0DFB	DEFM45	0E0A	DEFMA2	0DDE	DEFMA3	0DE6	DEFMA4	0E07
DEFMAS5	0E14	DEFMAC	0DD7	DELC35	0624	DELCH3	0620	DELCH4	062D
DELCHR	0608	DELIM	00B7	DEOL	7082	DFCB	1BC8	DFMFLG	0084
DIYAPP	0E7D	DIYER0	0E62	DIVER1	0E68	DIYER2	0E6F	DIYER4	0E74
DIVERT	0E5E	DIVFL2	0087	DIYFLG	0086	DOCAP	006C	DOCIM	006D
DOFMS	166B	DOFMS2	1671	DOFMS4	167A	DOFMS6	1688	DOFMS7	168B
DONE	005F	DOOOUT	15F7	DOTAB	0B5D	DOTAB2	9B6A	DPCRLF	711E
DPR024	033E	DPR025	0343	DPROC	02E3	DPROC1	02F3	DPROC2	031E
DPROC3	0349	DPROC4	0355	DPROC5	0365	DPSTRN	7118	DREAD	1660
DREAD2	166A	DUBB	0C5B	DUBH	0C47	DUBH1	0C4A	DUBH2	0C4F
DUBW	0C52	DWFLG	0083	EBFEND	00DF	EINCH	15CF	ENDLIN	00BC
EOFF	0034	EOIF	004A	EORF	004B	EY	0073	EXCHR	0057
EXTBUF	1815	FETC22	0735	FETC25	074B	FETC35	075F	FETC36	0764
FETC37	0767	FETC45	076F	FETC47	077D	FETC48	0780	FETC49	0789
FETC55	0798	FETC57	07AC	FETC58	07B7	FETC59	07BD	FETC65	07CC
FETC75	07EA	FETCH2	071F	FETCH3	074F	FETCH4	0768	FETCH5	078D
FETCH6	07C0	FETCH7	07E1	FETCH8	0808	FETCHR	0716	FILFLG	00C5
FILL	0CF5	FILOPN	004D	FINCM	089C	FINCM1	08AA	FINCM2	08AD
FINCM4	08B0	FINIS4	0984	FINISH	0975	FINMAC	007D	FIXBFE	0697
FIXYR3	124F	FIXYR4	1251	FIXVAL	1242	FIXWD	1529	FIXWD1	154C
FIXWD2	154E	FIXWD3	1554	FLBF	0060	FLUSH	0952	FLUSH2	095D
FLUSH3	0966	FLUSH5	0971	FLUSHB	094E	FMS	7806	FMSCLS	7803
FNDLS2	0F93	FNDLST	0F8C	FNDMA1	0F77	FNDMA2	0F83	FNDMA4	0F89
FNDMA6	0F8B	FNDMAC	0F6E	FNDNUM	11E1	FNTR	0D2E	FNTR2	0D35

FNTR4	0D4A	FNTR5	0D52	FNTR6	0D5F	FSTRVL	00A3	FSTRAM	0094
FTCHN1	1208	FTCHN2	120B	FTCHNM	11EF	GCNT	0036	GDNUM	004F
GETC22	06E0	GETC25	06E9	GETCH1	06D5	GETCH2	06DC	GETCH3	06ED
GETCH4	06FD	GETCH5	0706	GETCH6	0713	GETCHR	06CE	GETFIL	7127
GETIN	0BC2	GIBUF	1559	GIBUF2	155D	GIBUF4	157A	GIBUF5	1584
GIBUF6	158D	GTNA4	0FE4	GTNA6	0FE9	GTNAM	0FC3	HIPG	0093
IF	10DE	IF1	10E1	IF3	10EF	IF35	10F5	IF4	1100
IF5	1100	IF55	111F	IF6	112B	IF8	113E	IFFLG	007F
IFN	1126	IFN2	112A	IFY	1109	ILFN	1726	INC	0054
INCH	0206	INCHR	15FC	INCHR2	160C	IND	0038	IND2	0079
INDEX	00F3	INDNT	0B28	INDNT2	0B38	INFCB	008C	INIT	0269
INIT25	026D	INIT3	02A4	INIT4	02C3	INMAC	0F1C	INMAC2	0F1E
INMAC4	0F2C	INMAC5	0F30	INNUM	0065	INSS44	0655	INSSP2	0638
INSSP3	063E	INSSP4	0652	INSSP5	065B	INSSPC	062E	INTRO	020C
INTRO0	020F	INTRO5	0228	INTRO6	0246	INTRP	0DD0	ITEM	004C
ITMCH	146F	ITMCH2	1478	JNKCNT	009A	JST	0AEE	JST1	0AFE
JST15	0B03	JST2	0B04	JST3	0B0E	JST4	0B17	JSTF25	03EC
JSTF55	0413	JSTF63	0420	JSTF65	042B	JSTF95	045D	JSTFY	03D4
JSTFY1	03E5	JSTFY2	03EA	JSTFY3	03F5	JSTFY4	0408	JSTFY5	0411
JSTFY6	0416	JSTFY7	0430	JSTFY8	043A	JSTFY9	0452	JUST	00D3
LAST	2EFF	LDIV	0045	LDNSK2	1213	LDNSK4	121C	LDNSKP	1211
LEFT	0062	LEFTM	0B18	LEFTM1	0B25	LEFTM2	0B27	LENT25	0B52
LENTH	0B3E	LENTH2	0B4E	LENTH5	0B5C	LFM	003E	LINBU2	1842
LINBUF	177A	LINCNT	003D	LINS	04D2	LINS2	04D7	LINS3	04E2
LINS4	04EF	LINS5	04F9	LINS6	04FB	LINS7	0508	LLN	003B
LLN2	009C	LMACRO	2EFD	LOWPG	0092	LPPSTR	16E8	LSTRVL	00A1
LSTNUM	0058	LSTRAM	0096	LSTTRM	7091	MACCNT	0098	MACEND	00F7
MACNAM	009D	MACOVF	0893	MACROS	1D8A	MRCST	176F	MACTBL	1C88
MACTMP	009F	MBFLG	005B	MBFPNT	005C	MCEND	0FB6	MCEND2	0FC0
MCNT	00B9	MINDIS	0072	MNTH	003C	MON	0209	MSC	00CF
MTEND	1D88	MULTS	0AD8	MULTS2	0AE1	MULTS3	0AE3	NCOUNT	008A
NEDL	0D1F	NEDL1	0D27	NEDL2	0D29	NEDL4	0D2D	NEG	0066
NEGT	007E	NFST	1738	NMREGS	0030	NMST	175E	NOCAP	0CB5
NOCR	005E	NOEXP	0074	NOFILL	0CEE	NOFL	0064	NOJST	0AEA
NONUMS	0082	NOOUT	006E	NOSPC	0B70	NPGN	0053	NREG	119E
NREG4	11BC	NSP	0069	NUM	0125	NUMPNT	0055	NXTB	14A1
NXTB2	14AD	NXTB4	14B4	NXTB5	14C3	NXTB6	14CB	NXTCH	7121
NXTI	147B	NXTI2	1487	NXTI3	1489	NXTI4	1490	NXTI6	14A0
NXTMAC	00R9	NXTOUT	00AB	NXTRAM	0092	NXTTAB	0075	NXTTRP	007A
OPAPP	0EBB	OPAPP2	0EC3	OPAPP4	0EC8	OPMAC	0E88	OPMAC2	0E90
OPMAC4	0E99	OPMAC5	0EA5	OPNF	14DE	OPNF4	14F3	OPNF5	1502
OPNF6	1526	OPNF8	1525	OUTCH	0203	OUTCH2	15E3	OUTCH3	15E8
OUTCHN	70A3	OUTCHR	15D9	OUTL55	0589	OUTL75	05B1	OUTL82	05D8
OUTL85	05E2	OUTLI1	055A	OUTLI2	0563	OUTLI3	056D	OUTLI4	0575
OUTLI5	0582	OUTLI6	0593	OUTLI7	05AE	OUTLI8	05CC	OUTLI9	05E7
OUTLIN	054E	OUTM18	0F08	OUTMA0	0EF2	OUTMA1	0EF4	OUTMA2	0F0A
OUTMA3	0F0B	OUTMA4	0F10	OUTMAC	0EE7	OUTSY	0D82	OVFSTR	16FB
PAGE	0AB4	PAGE2	0AC0	PAGE4	0ACB	PAGES5	0AD0	PAGE6	0AD7
PAGEL	0C87	PAGEL1	0C90	PAGEL2	0C99	PAGEL4	0C9B	PASCHR	006A
PASFLG	0081	PASS	0A8A	PCHAR	00C9	PCRLF	08DE	PCRLF2	08E1
PCRLF4	08E6	PDATA1	15BC	PDATA1	15BF	PDATA2	15CB	PFLG	00C7
PGL	003F	PGN	0069	PGNUM	0C3A	PGNUM4	0C46	PGSTR	16CC
PRNTR	0090	PRNU27	1297	PRNU28	12RB	PRNU31	12B7	PRNU32	12C1
PRNU35	12D0	PRNU37	12D8	PRNU45	12FD	PRNU65	1323	PRNU72	133C

PRNU73 1340	PRNU82 1352	PRNUM 128F	PRNUM2 1291	PRNUM3 12B0
PRNUM4 12E0	PRNUM5 1305	PRNUM6 1319	PRNUM7 1329	PRNUM8 1348
PROC 037A	PROC2 0380	PROC3 0386	PROC4 038D	PRQU 16AE
PSCNT 008B	PSTRNG 15BA	PTFL 0070	PTIND 0CB9	PTIND2 0CC6
PTIND3 0CDB	PTIND4 0CE7	PTIND5 0CED	PULLX 1438	PUNT35 03A7
PUNTS2 0398	PUNTS3 039D	PUNTS4 03BF	PUNTS5 03C2	PUNTS6 03C4
PUNTS7 03CA	PUNTST 0390	PUSH4 1434	PUSHX 1426	QUSTR 16E4
RDIT 1452	RDIT4 146E	REMMA4 0E27	REMMA6 0E44	REMMAC 0E1E
REMNAME 0E4E	RESPC 0B73	RETMA2 06BA	RETMAC 06AF	RETREG 00F1
RETRNR 7096	RIFLG 0088	RIGHT2 05FB	RIGHTJ 05F8	RINS 050C
RINS2 0510	RINS3 0517	RINS4 0522	RINS5 0530	RINS6 053A
RINS7 053C	RINS8 054A	ROM 00C1	ROM2 11C3	RPT 0CFC
RPTERR 713C	RTJ 00CD	RWND 1691	SAUTO 11CA	SAUT04 11D6
SAYS 0D66	SAYS1 0D71	SAYS2 0D73	SAYS25 0D77	SAYS4 0D7D
SAYS5 0D81	SAYSX 0ED9	SAYSX2 0EE4	SBFLG 0098	SBUF 190A
SCAP 00B5	SCRL55 090C	SCRL75 092A	SCRL85 093E	SCRLF 08EA
SCRLF1 08F1	SCRLF2 08FD	SCRLF4 0904	SCRLF5 0907	SCRLF6 0918
SCRLF7 0927	SCRLF8 0934	SCRLF9 0947	SENV 0BD0	SENV1 0BDD
SENV2 0BDE	SENV3 0BE3	SENV4 0C00	SENV6 0C0F	SENV8 0C28
SETCA2 0B20	SETCAP 0813	SETEXT 712D	SIGN 0067	SIN 0071
SIND 0C9C	SNGLS 0AE6	SPACE 0A91	SPACE2 0AA2	SPACE4 0AAB
SPACE6 0AB3	SPCP1 00E5	SPCP2 00E7	SPIFLG 0085	SPSPF 006B
SROM 11C1	STAB 0B9E	STAB2 0B91	STAB4 0BA7	STACK 01FF
START 0200	STCAP 0CB0	STOP 0D08	STOP1 0D0E	STOP2 0D1E
STPOUT 00A5	STPSTR 16DA	SUB 0051	SUPL 00B2	SVDSPC 007C
SWRDF 00B3	SYSERR 0F19	TAB 00BD	TABCH 0877	TABCH2 0B7F
TABEND 0124	TABFI2 0B8B	TABFIL 0B82	TABFLG 0077	TABS 0110
TBLEND 0A89	TCNT 00B8	TCPNT 00A7	TEMP 00E9	TEMP2 00EB
TEMP5 00ED	TEMP6 00EF	TERM 0BAD	TERM2 0BB0	TERM4 0BB9
TFCB 1B08	TFILF 0063	TFILL 008E	TIND 00B0	TITL12 101A
TITL15 1020	TITL65 107A	TITL75 108F	TITL78 1099	TITLE 0FFA
TITLE1 100A	TITLE2 103A	TITLE4 105D	TITLE5 1060	TITLE6 106B
TITLE7 1082	TITLE8 109C	TITLE9 10A7	TLEN 0FED	TLEN2 0FF9
TLLN 00B1	TLN 00D5	TLPP 0091	TOUTL 006F	TPOS 0086
TRAPS 196E	TRPEND 199E	TSIN 00AF	TSTBR2 15A0	TSTBR4 15A1
TSTBRK 1595	TSTNE2 114A	TSTNEG 113F	TSULN 05EA	TSULN2 05F7
TTLBUF 1A54	TTLPTN 00BA	ULFLG 004E	UNDL 144D	USER 2F00
WASN 708C	WIDTH 00C3	XFRRT2 10D6	XFRRTL 10BF	XMAC 00AD
XTEMP 00F5	XTEMP2 008E	YEAR 0048		