MICRO-80

DECEMBER 1979   Price $2.50

CONTENTS

MICRO-80

MICRO-80

EDITORIAL

Welcome to the first issue of MICRO-8∅.

MICRO-8∅ is the first and only Australian monthly magazine devoted entirely to the TRS-8∅ microcomputer. It is available by subscription - $24.∅∅ for 12 months or by mail order at $2.5∅ per copy. Special bulk purchase rates are also available to computer shops etc., - please use the form in this issue to order your copy or subscription.

The purpose of MICRO-8∅ is to publish software and other information to help you get the most from your TRS-8∅ computer and its peripherals. MICRO-8∅ is in no way connected with the TANDY organisation.

WE WILL BUY YOUR PROGRAMS!

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-8∅ to earn some extra income is included in every issue.

CONTENT

Each month we publish at least one applications program in Level 1 BASIC; one in Level 2 BASIC and one in Disk BASIC (or Disk compatible Level 2). We also publish Utility programs in Level 2 BASIC and Machine Language. At least every second issue we will have an article on hardware modiciations or constructional articles for useful peripherals. Starting next month, we will also be running a series of informative articles on Assembly Language Programming, new product reviews etc., In addition to all this, we are prepared to print news from TRS-8∅ Users clubs, letters to the Editor, display advertisements and classified ads.

ADVERTISING

We accept camera ready copy for display advertising at the following rates:
- FULL PAGE (19 cm wide x 28 cm high) $120
- 1/2 PAGE (19 "   "   x 14 "   ") $ 60
- 1/4 PAGE (19 "   "   x 7 "   ") $ 30

Classified Ads are $8.00 for up to 50 words.
These prices are valid for issues 2 & 3 only. Ads must be submitted by the 15th of each month in order to appear in the following month's issue. An official Company Order or payment must be included with the advert.

SPECIAL OFFER - ISSUE 2 ONLY
FOR ISSUE 2 ONLY, THE COST OF A FULL PAGE ADVERT. WILL BE REDUCED BY 1/3rd TO ONLY $80!!!!!!!!

TRS-8∅ USERS CLUB NEWS

We are prepared to print news of the activities of TRS-8∅ Users Clubs, up to a maximum of 200 words per club per month, space permitting. Copy must be TYPED with DOUBLE LINE SPACING and reach us NO LATER than the 15th of each month in order to appear in the following month's issue.

CASSETTE SOFTWARE

For those who don't get as much time at the keyboard as they would like, the
programs in each month's issue are available on a cassette for an additional $3.50.
Or you can order a 12 month subscription to the magazine PLUS cassette for $60.00,
that's only $5.00 per month!!!!

COPYRIGHT

-00000-

********** NEXT MONTH'S ISSUE **********

The January edition of MICRO-80 will be ready for posting about 14th January. It
will contain at least the following programs:

HANGMAN (L1)   - The traditional, educational word-game brought up to date,
                                complete with a picture of the hangman's victim, on the
                                screen.

FRUSTRATION (L2)   - Test your reactions as you try to beat the computer's
                                "clock".

GAME OF LIFE   - This is the well-known and fascinating illustration of
(L2+m/c language)   growth and decay of populations (civilizations?), now
                                written to run on your TRS-80. Machine language sub-
                                routines have been used to speed up execution. You can
                                observe up to one generation per second.

STOCK RECORDING SYSTEM -   This program, which will run in both Level 2 and Disk BASIC,
(Disk/L2)   enables the businessman to keep track of his stock, item by
                                item. It can also be used as a simple data base management
                                system.

MONITOR IN BASIC (L2)   - Written in Level 2 BASIC, this monitor program allows you to
                                examine and modify memory also to dump the contents of a
                                block of memory to the screen or printer. It is absolutely
                                essential if you want to enter our machine language programs
                                but don't have a monitor such as T BUG.

In addition to all these programs, January's issue will also contain the first in a
series of articles on Assembly language programming. If you are interested in this
fascinating and powerful method of programming but have not known where to start,
this series of articles is for you.

SOFTWARE SECTION

The listings for these programs are published in a separate section at the back of the magazine. This section explains what each program is all about and how to use it

SUPER MASTERMIND - L1/4K

If you are a Mastermind addict then this is the version you have been waiting for. Whereas Mastermind only has pegs of six different colours, Super Mastermind has up to nine (not colours on the TRS 80 of course, but numbers 1-9). With Mastermind you have four colours to find at any one time. With Super Mastermind you have up to 10! But the best part of all is that the Computer does all the tedious work for you. It chooses the numbers and it works out how many you have placed in the right position and how many are correct but in the wrong position. It also works out your score and gives you a progressive average score. All you have to do is use your powers of logic. If the unthinkable happens and you can't work out the answer, simply enter 999 and the computer tells you. But that ends the session and wipes out your score.

The listing at the back includes full instructions and takes 3822 bytes. That's about 240 more than a 4K system can handle. So, if you are entering this program into a 4K system we suggest you leave out some or all of the instructions. We have printed them separately so that you can read them without using the computer.

To reduce the program size to 2368 bytes:
Delete lines 4Ø, 5Ø, 81Ø and all higher lines.
Change line 3Ø to F. C=1 TO 15ØØ: N. C

LEVEL 2 - If you want to use this program in a Level 2 machine, enter it without abbreviations and with the following changes:

Add line 10 DIM A(3Ø)
Add line 12Ø RANDOM
Delete lines 95 and 97
Don't forget to use PRINT @ instead of PRINT AT.

-00000-

DIGITAL CLOCK - L1/4K

This is an extension of the clock program published in the Level 1 Manual. However, it prints the time in 4cm. high digits and indicates whether it is AM or PM. It is also more accurate because the timing loop runs for a full minute, rather than one second.

The timing variable is N in line 13Ø. To adjust the accuracy of the clock, note the starting time, let it run for about 12 hours then note the time on the clock and the correct time. Work out the number of minutes elapsed since the start:
A = elapsed minutes on computer
B = actual elapsed minutes
Then calculate a more accurate value of N:
new N = old N x A/B
Enter the new value of N into line 13Ø

As an exercise, why don't you write a small routine to do this calculation.
Another useful addition would be an alarm which could print a message on the screen,
turn on the cassette recorder or do both. When you have done this, write in and
we'll publish the best modification and send the winner a free copy of MICRO-80

-00000-

SNAKE - L2/4K

This program is a lot of fun for kids of all ages. It tests your reflexes and
generates a lot of excitement. It starts off by drawing a large box on the screen
inside which there is a short snake of ***'s moving across the screen. You use
the arrow keys to make it change direction but, if you turn it back on itself or
crash into the box, that's the end of you. At random intervals "mice" in the form
of graphics blocks suddenly appear. If you make the snake eat one before it
disappears then you score (you don't know how many points until after you've
eaten it). If it gets away (disappears) then its value is subtracted from your score.

As the game goes on, the snake gets longer and longer and moves faster and faster.

HAPPY HUNTING!

-00000-

MERGE - L2/4K or 16K

Have you ever wanted to add together two BASIC programs which you have stored on tape.
Normally, you would CLOAD the longer one then have to type in the shorter one. Never
again, the simple, 7 line program listed below, allows you to merge together two
BASIC programs in your TRS 80.

```
1  CLS
2  PRINT "16549=";PEEK(16549)
3  PRINT "16548=";PEEK(16548)
4  A= 17129
5  B=A: A=PEEK(B+1)*256+PEEK(B)
6  IF A > Ø THEN 5
7  POKE 16549,INT(B/256):POKE 16548, B-INT(B/256)*256: END
```

It works because Level 2 BASIC stores a "start of program" pointer at 4ØA4H and
4ØA5H (Decimal 16548 and 16549). This pointer is initialised to 42E9H (Decimal 17129)
during power-up. The start of program pointer is stored in RAM rather than ROM so
that it can be relocated to a higher address when DISC BASIC is loaded into RAM. The
above program uses PEEK and POKE statements to do the same thing when a Level 2
BASIC program is resident in memory. First, it prints out the existing value of the
start of program pointer, so that you can restore it later. Then it searches for the
first vacant position in RAM after the resident program and resets the start of
program pointer to that address. Now, when you CLOAD a program, it will commence
loading from this new address. Once it is loaded, you restore the pointer to its
original address and, lo and behold, the two programs are merged. You can do this
as often as you wish but make sure that the second program has higher line numbers
than the first, the third higher than the second, and so on.

To merge two programs, use the following procedure:

1. Key in or CLOAD the seven line program above.

2. Run it.

3. Write down the two addresses and their values which are printed on the screen (they should be 16549=66 and 16548=233)

4. CLOAD the BASIC program to be added to the existing program.

5. Using the keyboard, type in:
                              POKE 16549,66
                              POKE 16548,233
    (or whatever values were printed on the screen if you are not starting from the first program in memory)

6. LIST the combined program.

At this stage, you will have merged the seven line utility program with your BASIC program. To merge another BASIC program, repeat the above steps. When you have finished merging all your programs, kill lines 1-7.

If some of the programs you wish to merge have similar line numbers, you can still use MERGE with the assistance of Tandy's RENUM program. Load this into protected memory before loading MERGE, then use the following procedure:-

    Steps 1-4 as before.

5   Use RENUM to renumber the new BASIC program so that all line numbers are above those in the resident BASIC program.

6   POKE 16549,66
    POKE 16548,233

7   LIST the combined program

                              -00000-

LOADER L2/16K (4K)

Have you ever sat through a complete CLOAD of a long program, only to find that it hadn't loaded properly and you have to start all over again? This only has to happen once or twice and you never really trust that blinking asterisk again! Well, LOADER overcomes all that by printing out each byte of the BASIC program as it is loaded. You can see when the machine is loading rubbish, abort the LOAD and start again.

In order to give you a chance to read them, program lines are displayed one after the other, separated by a graphics block. Line numbers are not shown.

The program is at the back of the magazine. There are two listings in HEX. One is for a 4K machine, the other is for a 16K machine. To enter either listing you will need a monitor such as T BUG (or the monitor in BASIC we will publish next month). Follow the procedure below for the 16K version (4K shown in brackets).

1. Answer MEMORY SIZE? with 31880 (19600). This enables you to use the program more than once

2. Enter T BUG

3. Type M 7C90 xx 4C (4C90 xx)

4. Continue entering the values in the listing until you get to 7EE6 (4EE6) then enter "X": 7EE6 xx X (4EE6 xx X)

5. Enter J 7D33 (4D33)

If all is well, a copyright message will print out at the top of the screen and control will be automatically transferred to BASIC.

Whenever you wish to use LOADER, set your cassette recorder to PLAY and then press the shift key and the down arrow at the same time. The cassette recorder will start running and the program will be displayed on the screen as it loads. To abort loading, just hit the BREAK key or use the RESET button (the BREAK key will only work while loading is in progress). If there is nothing on the tape, the RESET button will have to be used.

When loading has finished, LOADER will tell you the locations of the first and last bytes of the BASIC program just loaded, in HEX. Some BASIC commands or errors could cause the entry pointer for LOADER to be changed or overwritten. In this case, type X=USR (0) or, if this fails type "SYSTEM" and answer the *? with /32051 (/19763) This will reinitialise LOADER.

Once you have tested LOADER and proved that you have entered it correctly, you should make a tape of it. Load T BUG again then:

1. Set up cassette recorder to RECORD

2. Type P 7C90 7EE5 7D33 LOADER (ENTER) (P 4C90 4EE5 4D33 LOADER (ENTER))

T BUG will punch a tape which can be loaded under the SYSTEM command using the file name "LOADER"

LOADER can also be stored on disk using TAPEDISK. TAPEDISK parameters are "T(LOADER):0 7C90 7EE5 7D33". To use it under DOS type LOAD "LOADER" then type BASIC or BASIC 2 and use the SYSTEM command as described above, to initialise it.

LOADER will occasionally generate error messages. If this happens during a LOAD, it means that you have run out of memory. At other times, ignore them. Finally, don't be tempted to use the space between the top of LOADER (7EE5 or 4EE5) and the end of RAM, for another program. LOADER uses this area to set up a table and it would destroy any other program resident there.

WORDPROC: A WORD PROCESSOR IN BASIC

By: Hubert S. Howe Jr. Ph.D.
Reprinted from TRS-8Ø Monthly Newsletter
Published by H & E Computronics Inc.
Box 149 New City, New York 10956 USA.

INTRODUCTION

WORDPROC is a Level 2 BASIC word processing program that enables you to create files that represent documents to be printed on your TRS-8Ø. It's main features are:

1. It is line-oriented. This means that every line of text has a line number associated with it, whether or not the line number is printed. Line numbers are convenient references to the text. Lines can be moved, replaced, rearranged, etc.

2. It allows entry of upper or lower case letters from your keyboard without modification to the TRS-8Ø.

3. Lower case letters can be printed properly if you have a line printer with the lower-case character set. Otherwise, everything is printed as upper-case.

4. Text files can be saved and loaded either on cassette tape or disk.

5. Lines can be edited in a manner very similar to Level 2 BASIC.

6. Text lines can automatically be right justified.

7. There are provisions for adjustable page lengths, page numbering, line lengths and a special title to be printed at the top of each page.

8. The program itself is in Level 2 BASIC so that you can study it to learn how it works and possibly modify it if desired.

MODIFYING THE PROGRAM

The standard version of WORDPROC is written to fit into a 16K Level 2 TRS-8Ø. It has provision for 12Ø lines of text, which is sufficient for much business correspondence or short documents. If you have 32K or 48K, you will probably want to expand it into your larger memory. This is accomplished by changing line 11Ø, near the beginning of the program. It currently reads:

        11Ø CLEAR 72ØØ: NL=12Ø

With 32K, you can expand to at least 200 lines, and with 48K to at least 400 lines. The amount of string space that you must clear is 60 (or the number of characters per line) times the number of lines. Thus, for 32K you could change line 110 to:

        11Ø CLEAR 12ØØØ: NL=2ØØ

and for 48K it could be:

        11Ø CLEAR 24ØØØ: NL=4ØØ

RUNNING THE PROGRAM

There are basically two modes of operation in WORDPROC: the FILL mode and the
COMMAND mode. FILL is used when you are entering text into the computer, and
COMMAND at other times. When in FILL, you can return to COMMAND only by typing
"shift-@" twice in succession.

When in the COMMAND mode, the computer prints the word "COMMAND" and awaits your
reply. The replies, all discussed in detail below, are either a single letter or
an entire word. The computer executes the command, which may require additional
input from you, and returns for another one.

HELP

"HELP" or "H" is a command provided in case you forget any of the commands. It
prints the complete list of commands and their associated words in order to remind
you. These are shown below:

```
F    FILL       Add text to the buffer.
P    PRINT      Print text on video display or line printer.
C    CLEAR      Clear buffer (erase all text).
R    REPLACE    Replace text starting at existing line.
I    INSERT     Insert text starting at empty line.
D    DELETE     Delete lines.
M    MOVE       Move lines.
N    NUMBER     Number line, eliminating blanks.
E    EDIT       Edit line.
L    LOAD       Load text file from cassette or disk.
S    SAVE       Save text file on cassette or disk.
J    JUSTIFY    Right-justify text lines.
H    HELP       HELP! Print commands.
     SET        Set variables governing program options.
```

FILL

"FILL" or "F" enters the FILL mode, which you use to enter text into the computer.
The line number is displayed at the left, and then the cursor flashes on and off
over the current position in the line where you are entering text. If you are
starting at the beginning, FILL begins at the first space in line $\emptyset$ (zero). If
there is already text in the buffer, it starts after the last filled space, and
any preceding lines are printed on the video display. The "cursor" used throughout
the program is not the same cursor used in BASIC but a graphics block that covers
the entire space filled in by the letter.

If you are typing in a word that goes off the end of a line (by exceeding the line
length, which is set by default to 6$\emptyset$ or by the SET command), WORDPROC
automatically removes the word from the preceding line and places it at the
beginning of the next line so that the word will not be split. Later, the line can
be right justified by the JUSTIFY command.

During the FILL mode, the following keys have special functions:

LEFT ARROW	backspace
RIGHT ARROW	TAB five spaces to the right
DOWN ARROW	form feed (page eject) on line printer
ENTER	move to the next line (inserts no character)
SHIFT LEFT ARROW	backspace entire line
SHIFT RIGHT ARROW	fill to end of line with spaces
SHIFT-@ (TWICE)	return to COMMAND mode

The following points should be noted about the use of these special keys:

The RIGHT ARROW (TAB) key actually inserts five spaces into the line.

The DOWN ARROW key will cause a page eject on the line printer (if your line printer
responds this way to a form feed), but does not affect the printing of the page
number and title.

When ENTER is typed, it does not insert any character into the line. If ENTER is
typed at the beginning of a line, the line remains empty, but FILL proceeds
to the next line.

If you type SHIFT LEFT ARROW when the cursor is at the bottom of the screen, it
appears to scroll to the next line but actually continues to fill the same line.

The FILL mode is entered by both REPLACE and INSERT as well as FILL. The
difference is that REPLACE and INSERT can insert lines of text anywhere a line
number can exist, and they will stop as soon as existing text is reached. FILL
always adds text to the end of the file.

PRINT

PRINT or P is used to print the text in the buffer, optionally on the line printer
or video display. When the PRINT command is issued, the computer asks two
questions:

        PRINT LINE NUMBERS (Y/N)?
        LINE PRINTER OR VIDEO DISPLAY (P/V)?

You must reply "Y" in order for line numbers to be printed. Similarly, you must
reply "P" for the line printer to be selected. When you select the line printer,
the program automatically checks to see if the line printer is ready. If it
is not, it prints the message:
        LINE PRINTER NOT READY. ABORT (Y/N)?

If you reply "Y" a new command is requested. Otherwise, it asks you to press
ENTER when ready, and repeats the printer test. You can never use the printer
without first readying it; the program will never inadvertently "freeze up" as
it will if you do a LLIST or LPRINT in BASIC (without a ready printer).

The printing on the video display is identical to that on the line printer, with
the exception that only upper case letters are printed. In particular, the
number of lines on the page is the same' the default number is 50 lines per page.

Not all these lines can be displayed on the screen at the same time, although they fit comfortably onto a page. Therefore, it is useful to PAUSE during the printing process by typing "shift-@". (This is a normal feature of Level 2 BASIC).

The program pauses and asks you to press ENTER before continuing at the end of each line of a multi-page printing.

The TITLE and PAGE NUMBER are optionally printed at the top of each page. These options are controlled by the SET command.

If your TRS-80 is unmodified, holding down the SHIFT key while typing a letter produces a lower case letter: otherwise the letter is upper case. This is the reverse of a typewriter. Some printers (such as the SELECTRAPRINT) have been modified so that they convert upper case into lower case and vice versa. Whichever way your TRS-80 works, WORDPROC considers upper and lower case letters to be distinct. This means that "A" is different from "shift-A" even though both are displayed as "A" on the screen.

CLEAR

CLEAR or "C" erases everything in the buffer. In addition, it restores all SET variables to their default values.

It is not necessary to perform a CLEAR at the befinning of a run. It is only necessary if you want to clear everything out and start again.

LOADING a text file from cassette tape or disk also clears out any**thing in the** buffer and fills it with the material read in.

INSERT

INSERT or "I" is used to FILL in new lines starting from any legal line number, provided that no text currently occupies the starting line. INSERT will operate just like FILL until you come to a line that is occupied with text, at which point it will automatically return to the COMMAND mode.

REPLACE

REPLACE or "R" is used to FILL in new lines starting from any legal line number, provided that text DOES exist at the starting line. The text at the starting line is replaced entirely. Following that line, REPLACE operates exactly like INSERT and continues to FILL new lines until you come to a line occupied with text, at which point it will automatically return to the COMMAND mode.

DELETE

DELETE or "D" is used to delete a range of lines from the text. Deleted lines are empty, but the line marker used to FILL continues to point to where it did before the DELETE. Therefore, to insert new text into the deleted lines, INSERT should be used rather than FILL

MOVE

MOVE or "M" is used to move a block of lines from one position in the buffer to another. The only limitation is that the position where the text is moved must

consist of empty lines. If a new line is ever found that contains text, the
MOVE is terminated, and only the lines that were moved prior to the error are
moved. The lines from which the text is moved are replaced with empty spaces.

When a MOVE begins, the computer asks:

FIRST LINE TO BE MOVED?
LAST LINE TO MOVE
FIRST NEW LINE?

If you were to reply to these questions, in succession, 1, 4, and 6, line 1
would be moved to line 6, line 2 to line 7, line 3 to line 8, and line 4 to
line 9. Lines 1, 2, 3, and 4 would become empty lines. If, in performing the
move, the computer found that line 7 already had text in it, it would print:

LINE 7 ALREADY OCCUPIED

and stop the move. However, line 1 would have been moved to line 6.

If you want to insert one or more blank lines into existing text, first MOVE
the lines following the line where you want the blank line inserted down to a free
area in the buffer, and then MOVE them back to one or more lines following their
original position.

MOVE can also be used to rearrange the text lines, in order to change the order
of paragraphs or for other reasons.

NUMBER

NUMBER or "N" is used to delete all blank lines in a text and close up the text.
In addition, the lines are renumbered from zero and the line marker indicating
the first empty line is moved to the first line after any text lines.

BLANK lines (lines containing only spaces) are different from EMPTY lines (lines
containing nothing at all). Only EMPTY lines are eliminated by NUMBER. Therefore,
blank lines can still be kept in a text.

SAVE

SAVE or "S" saves the text file, including the TITLE, all SET variables, and the
complete text, on cassette tape or disk. When you type SAVE, the computer first asks

TAPE OR DISK)T/D)?

If you are using cassette tape, the computer prints PRESS ENTER WHEN READY TO
CONTINUE? and waits foryou to ready the tape before continuing. (If you want
to abort the SAVE command, type BREAK and GOTO 200.)

If you are using a disk, the computer asks FILESPEC? Here you must give the
complete file specification. If you specify a drive number (such as ":0") the
file specification must be enclosed in double quotation marks.

Because of the way that Level 2 BASIC handles string variables, commas, colons,
and blank spaces must be translated into special graphics characters so that the

text can be restored to its precise previous structure. Unfortunately, this translation process takes a significant amount of computer time to execute - possibly several minutes. When you save or load a text file, the computer may appear to be doing nothing, but if you interrupt execution and turn on the trace function you will see it going through a long loop.

After the file has been saved on tape or disk, WORDPROC then retranslates the file back to its original condition so that execution may continue and you may add new text to the file. The retranslation, of course, also takes a significant amount of computer time.

Text files are stored in the same way whether on tape or disk. A file can be loaded from cassette and saved on disk or vice versa.

LOAD

LOAD or "L" loads a previously-saved text file from cassette tape or disk. LOADing the file restores everything to the condition it was in when the file was saved: all SET variables, the TITLE, and all text lines are retrieved. Any text presently in the buffer is lost.

In other respects the operation of LOAD is the same as SAVE.

SET

SET (there is no single letter abbreviation) is used to set internal variables that control the line length, page length, spacing of text, page numbering, starting page number, and TITLE. When you type SET, the computer asks you a series of questions about setting each of these variables. If you do not want to change the present value, simply hit ENTER without typing a new value. Only the new values are inserted.

The variables are as follows:

LINE LENGTH? This indicates the number of characters in a line. The default is 60. LINE LENGTH can be any value up to 255. LINE LENGTH controls the manner in which words are separated on lines during FILL, and is used to set the right margin by JUSTIFY.

PAGE LENGTH? This indicates the number of lines on a page. The default value is 50. During printing, the computer waits for you to press ENTER following the completion of each page.

SPACING (1-SINGLE, 2-DOUBLE)? This controls the spacing between lines during the printing. 1 is for single spacing, 2 for double, and any number that you input here would be used as the number of spaces between lines.

PAGE NUMBERING (1=YES, 0=NO)? This variable controls the printing of both the TITLE and the page numbering. If you respond with zero, nothing is printed at the top of the page. But, if you reply with one, then both the TITLE (if present) and the page number are printed at the head of each page. This line is not counted as a line of text.

STARTING PAGE NUMBER? This indicates the number of the first page of text. If you have long documents prepared as several text files, then they can still be combined into one final document by setting the starting page number to the right value. Page numbers are printed so that they are right justified over the right margin of the text.

TITLE? The TITLE is a character string that is printed left-justified on the top line of each page of text, followed by the page number. It is printed only if the PAGE NUMBERING is on. The TITLE can appear to be centered if it begins with blank spaces. The same TITLE is printed on each page. If the TITLE contains any punctuation, it must be enclosed within double quotation marks when it is SET.

EDIT

EDIT or "E" is one of the most powerful commands in the word processor. It is used to edit a single line at a time in a manner very similar to Level 2 BASIC. When you type "E", the computer asks you the line number of the line you want to edit. If the line contains text, it is printed on the second line of the video screen with the cursor flashing over the first character. (If there is no text, an error message is produced.) What you then type at the keyboard determines how the line is edited. The following EDIT commands are recognized:

SPACE - Advances the cursor to the next character in the line, unless you are at the last character in the line.

ENTER - Terminates editing. ENTER MUST be typed at the conclusion of the editing process.

LEFT ARROW - Backspace, unless you are at the first character.

I - Insert characters. Following the I, any characters that you type are inserted BEFORE the current cursor position. You can exit from I by typing either ENTER, which terminates editing, or SHIFT UP ARROW, which allows you to continue editing the line. If the line length equals or exceeds the maximum, you cannot start an Insert, but you can continue it past the end of a line.

D - Delete character. Only ONE character is deleted, and all characters following that character are moved left one space.

H - Hack and Insert. All text following the current cursor position is deleted, and whatever you type is added to the line. Note that the cursor does not flash in this mode, indicating that there is no text in the line at the position where you are adding text.

X - End of line and insert. The cursor is moved to the first position following the last character in the line, and then the operation of X is identical to H.

C - Change character. Only ONE character is changed, to whatever character is typed following the "C".

S - Search character. The character following the "S" is searched for in the line and, if it is found, the cursor is moved to that position; otherwise, it is not moved.

During the EDIT process, upper and lower case letters are kept distinct,
although everything is printed as upper-case on the screen.

When text is being entered in the FILL mode, it is impossible to enter
characters into a line past the line length, but in EDIT, only during the
operation of "H" or "X". it is possible to go past the end of a line. This
is legal, although the program prints a reminder after exiting from EDIT.
If you want to delete the characters, you can re-edit the line.

Once you enter "H" or "X" in EDIT, it is impossible to return to EDIT by
SHIFT UP ARROW as during "I". If you want to Insert, you must use "I". It
is impossible to Insert characters past the end of the line.

JUSTIFY

JUSTIFY or :J: is used to right justify all lines of text. Nothing is printed
during the operation of JUSTIFY, although it can take a long time to execute
if the program contains a long text. For this reason, it is suggested that you
JUSTIFY each section as you enter it into the text, and check it as you go along.

JUSTIFY inserts additional spaces into the middle of a line, only where there
are already empty spaces, to make the right margin vertically aligned. If a
line contains no spaces at all, it will not be justified. It will also not
be justified if it ends in a period, question mark, exclamation point or
right parenthesis.

When a line contains only a few words, requiring many spaces to be inserted, it
may be necessary to execute JUSTIFY more than once to complete the process.

If a line ends with a blank space, it may appear not to be justified even though
the program would consider it to be. To avoid this problem end the line with
ENTER as you type it in, unless the word is split from one line and moved to the
next.

When you end a line with a SHIFT RIGHT ARROW, spaces are automatically inserted
out to the end of the line. This is used when you want text such as titles,
etc. to be centered on the page, or not to be justified for any reason.

JUSTIFY physically changes the text in the buffer, so that it is then stored
with the internal spaces.

PROBLEMS

There are several problems that must be taken into account when using WORDPROC.
Most of these are a direct consequence of the fact that the program is in BASIC,
and thus could be eliminated by translating the program into assembly language.

First, you will probably notice immediately that the speed at which you can type
characters into the file is not always fast enough, and some letters that you do
type are not inserted into the text. This is because of the time that it takes
BASIC to process each letter, which is very short, but sometimes not short enough.

More annoying than this, however, is that sometimes the entire computer will appear to "freeze up" for a moment. As the buffer gets fuller and fuller, the likeliness that this will happen and the length of time it will take both get longer and longer. The reason for this is that BASIC must sometimes reallocate the string space while a job is running, even though many of the locations set aside for string space are still free. Nothing is really wrong when this happens, but it can disturb your use of the program.

Another BASIC problem, which we mentioned above, was the necessity of translating special characters before executing a SAVE or after a LOAD. One reason why this was necessary was that, in designing WORDPROC, we wanted to use only Level 2 BASIC so that anyone with a Level 2 computer could use the program, whether or not they have a disk. Some features of the program could be implemented more efficiently in DISK BASIC.

Finally, another problem for which there is presently no solution is that WORDPROC does not allow you to underline or overprint characters. Such a feature would require a line printer that could execute a carriage return without a line feed (such as the DIABLO or SPINTERM). Such a feature could easily be implemented in WORDPROC in the future if line printers with these capabilities become available to the average user.

NOTES ON USING WORDPROC

When using WORDPROC, it is convenient, and recommended, that you type ENTER when nearly at the end of a line. This will prevent ending the line with a space, which may appear to be misaligned after a JUSTIFY.

In order to center titles or other materials, use the SHIFT RIGHT ARROW following the last character in the line. This will prevent the line from being justified, by filling it with spaces.

To enter information at the beginning of a new page, use the INSERT command at a multiple of the page length (default is 5Ø). That is, INSERT at 5Ø. 1ØØ. 15Ø. etc. If you use the form feed character to eject pages, turn off the page numbering (by using SET).

Finally, if the program ever aborts with any BASIC error, you can retrieve anything presently in the buffer and return to the COMMAND mode by a GOTO 2ØØ statement. If you get an error in data transmission from a cassette tape or disk, you will find that the text may contain graphics in place of certain characters, after a GOTO 2ØØ. To untranslate the text and retrieve the proper characters, BREAK the program, type GOSUB 24ØØ, and then CONT.

-00000-

HARDWARE SECTION

IMPROVE YOUR CTR-41 CASSETTE RECORDER

If you have a Level 2 machine and a CTR-41 cassette recorder, chances are that you have trouble loading programs. There are several modifications you can make to your cassette recorder to improve this situation.

1. Enable the speaker on both CLOAD and CSAVE so that you can hear what is going on and set the level by ear.

2. Convert the TONE switch to control the motor so that you can position the tape without removing plugs.

3. Correct a ground-loop which introduces hum into the system and degrades recorder performance.

You can do all this in just a few minutes with a soldering iron, a Stanley knife, a Phillips head screwdriver, some wire and a resistor. But, if you have never done this sort of thing before, find someone to help you who has. WARNING: CARRYING OUT THESE MODIFICATIONS WILL VOID THE TANDY WARRANTY ON YOUR CASSETTE RECORDER. If the warranty has already run out, then you really have very little to lose and a lot to gain. Follow the instructions below:

1. Remove all plugs from the recorder, including the black plastic Mic-jack dummy plug. Take off the battery cover and remove the batteries.

2. Turn the recorder upside down on a soft surface, such as a newspaper, with the battery compartment facing away from you.

3. Remove the three Phillips head screws from the bottom of the case and the two in the battery compartment.

4. Lift up the rear of the case bottom a little then slide it forward about 1cm to clear the volume and tone controls. Put it aside.

5. Remove the chrome carrying handle at the front of the recorder.

6. Speaker Mod - Solder a 100 Ohm resister from point A to point B on the printed circuit board (PCB). Stick a small piece of plastic insulating tape to the PCB under the resistor. Keep its leads short and make sure it lies flat on the insulating tape.

7/ Motor Mod. - Cut the two tracks on the PCB which are connected to terminals 2 and 3 of the TONE switch. Solder a 14cm piece of insulated wire to point C on the PCB. Run the other end forward, feeding it underneath the wires crossing the circuit board near its centre. Solder the other end to terminal 2 (the Centre) of the switch.

    Solder a 15cm long piece of insulated wire to point D on the PCB. There is a yellow wire here which should be left in place. Run the new wire alongside the one you have just installed and solder to terminal 1 of the switch (that's the one on the left which was previously unused).
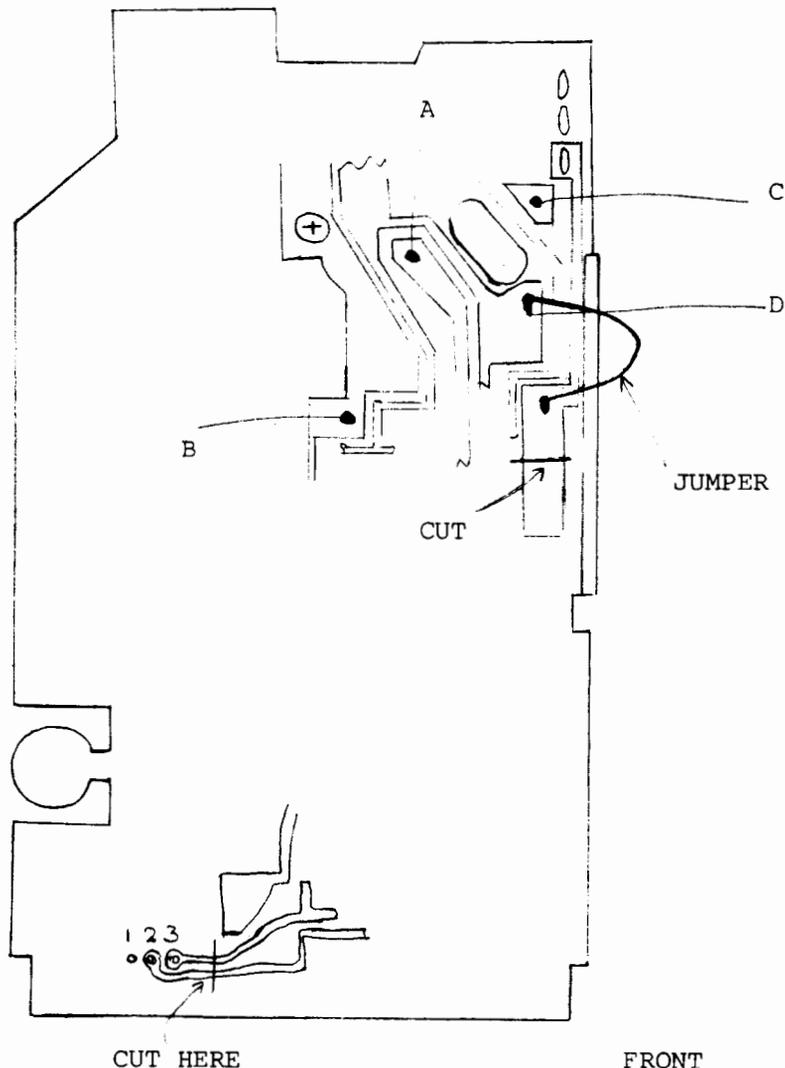
Now the recorder is permanently in the HIGH TONE mode and when the TONE
switch is set to LOW, the recorder will act as normal, under computer control.
However, when the switch is moved to the HIGH position, you can rewind, play
and use fast-forward without removing any plugs.

8. Ground-loop Mod - Cut the wide track on the PCB between the MIC and AUX jacks.

    Connect a 2.5cm long piece of insulated wire from the ground lug on the AUX
    jack to the ground lug on the EAR jack as shown in the diagram.

9. Replace the chrome carrying handle, bevelled edge to the top of the recorder

10. Reassemble the bottom case, remembering to slide it over the volume and tone
    controls before lowering it into position. Make sure that it fits correctly.

11. Replace the five screws. The two pointed ones go in the battery compartment
    holes.

CTR-41 CASSETTE MODIFICATION

LETTERS TO THE EDITOR

This space will be used in future issues to publish your letters. We'd like to
know how you feel about the magazine, the sort of articles and programs you want
us to publish, any useful tips you would like to pass on to other TRS-80 users and
requests for help from other readers to solve your problems. The space is here,
please make use of it.

-00000-

DO YOU HAVE SOFTWARE TO SELL?

MICRO-80 is interested in buying original programs for publication. We pay from
$15-$50 for a program, depending on our assessment of its appeal. As a guide, we
would pay the lower figure for games and simple utility programs. We would pay
more for business, educational, scientific and more comprehensive utility programs.

MICRO-80 also intends to become one of the leading suppliers of TRS-80 software,
both imported and produced locally. If you have written a program which you
believe is worth a lot more than $50 because of its wide appeal or because it would
offer very considerable financial benefits to users, then we could be interested in
distributing it for you, on tape or disk. We would take responsibility for all
production, advertising and distribution costs and would sign a royalty agreement
with you. If it really is good, then you could make thousands of dollars from it.
Some suggestions for programs which should sell:

- a machine language word processor which outperforms the Electric Pencil
  and sells for less

- a machine language monitor which outperforms T BUG and sells for less

- a set of programs which would teach the user to touch-type on the TRS-80
  keyboard. It needs to analyse mistakes and check typing speed

- a chess program which can beat Sargon and Micro Chess

- properly constructed educational programs to teach kids to read, do arithmetic etc.

Programs must be original, or have such a major component of originality that they
do not infringe any existing copyrights. You alone know that, so before we publish
a program, we will send you a form to sign in which you declare that the program is
your own original work and in which you give us permission to publish for the agreed
figure. This means that, should a third party claim a prior copyright to the program,
then it will be up to you to defend your copyright.

HOW TO SUBMIT A PROGRAM FOR CONSIDERATION

Send your program to us on CASSETTE or DISK, together with an application form from
your copy of MICRO-80. Make sure you include the following:

- YOUR NAME AND ADDRESS CLEARLY MARKED ON THE CASSETTE OR DISK

- The name of the program plus any special loading information, such as memory
  size, how to start a machine language program etc., clearly marked on the
  cassette or disk

- Whether it is Level 1, Level 2, Disk BASIC, System, T BUG (1 or 2) or editor/assembler format, clearly marked on the cassette or disk.

- A separate note giving useful information such as start, end and entry-point for machine language programs

- A clear expalantion of what the program does and how to use it. This will be published in MICRO-80 or used as the basis of the manual for a program to be published on tape or disk.

- A description of any changes that have to be made for different memory sizes etc.

- The size of memory and type of system needed to run it (eg. Level 1 4K, 32K with 2 disk drives, etc.)

- Any simple changes which can be made to make it more flexible or useful.

If your program is for publication in MICRO-80, then please avoid too many multi-statement lines unless they are essential to squeeze it into 4K. Remember, our readers have to key it into their machines! Please avoid using Level 1 abbreviations too, if you can. That way Level 2 owners can also use the program easily. Level 1 programs should be designed to run in 4K and Level 2 programs in 16K.

IMPORTANT: IF YOU WANT YOUR CASSETTE OR DISK RETURNED, ENCLOSE A STAMPED, SELF-ADDRESSED ENVELOPE.

-00000-

COMING SOON

Articles planned for the future include:

- Build your own light-pen for only a few dollars.
- Build an RS232 printer interface which works from your cassette port

Plus, of course, all those great programs you, our readers, are going to write for us.

APPLICATION FORM

To: MICRO-80, PO BOX 213, GOODWOOD, SA 5034. Phone: (08) 381 8542

Please send me:
- CURRENT ISSUE .... $ 2.50
- CURRENT ISSUE PLUS CASSETTE .... $ 6.00
- 12 MONTH SUBSCRIPTION, MAGAZINE ONLY $24.00
- 12 MONTH SUBSCRIPTION, MAGAZINE PLUS CASSETTE $60.00
- START SUBSCRIPTION FROM ....(Dec.79 is issue 1)
    (Make cheques payable to MICRO-8Ø)

NAME: ....

ADDRESS: ....

...................................POST CODE ....

```
20   CLS:PRINT@ 470,"SUPER MASTERMIND"
30   PRINT@ 842,"DO YOU NEED INSTRUCTIONS (1=YES, 2=NO)";:INPUTC
40   IF (C<>1)*(C<>2) THEN20
50   IF C=1 GOSUB 810
60   CLS:F=0:G=0:H=0
70   INPUT"HIGHEST NUMBER (2 TO 9)";A
75   IF (A<2)+(A>9) THEN70
80   INPUT"NUMBER OF COLUMNS (2 TO 10)";B
85   IF (B<2)+(B>10) THEN80
90   REM * GENERATE RANDOM NUMBERS *
95   INPUT "ENTER A NUMBER BETWEEN 1 AND 100";N
97   FOR C=1 TO N:J=RND(32767): NEXT C
100  CLS
110  FOR C=1 TO B
130  A(C-1)=RND(A):PRINT@ 4+2*C,"#";:NEXT C
140  PRINT TAB(27);"CORRECT POSITN   CORRECT VALUE"
150  REM * INPUT A SET OF NUMBERS *
160  J=0
165  FORE=1 TO 13
170  PRINT@ 64*E," ";:FORC=1 TO 4:PRINT"                    ",:NEXTC
180  PRINT@ 1+64*E,E+J;
190  FOR C=1 TO B
200  PRINT@ 4+64*E+2*C,". ";:A(9+C)=A(C-1)
210  NEXTC
220  FOR C=1 TO B
230  PRINT@ 940," ";
240  INPUT A(19+C)
245  REM PRINT 43 SPACES
250  PRINT@ 905,"                                           ";
260  IF A(19+C)=999 THEN340
270  IF (A(19+C)<1)+(A(19+C)>A) THEN290
280  GOTO300
290  PRINT@ 907,"REDO, VALUE MUST BE FROM 1 TO";A;"    ";:GOTO240
300  IF C=B THEN 315
310  PRINT@ 3+64*E+2*C, A(19+C);".";:: NEXTC
315  PRINT@ 3+64*E+2*C, A(19+C);
320  NEXTC
330  GOTO440
340  REM * PRINT HIDDEN NUMBERS *
350  C=B: PRINT@ 5+64*E,"THE CORRECT ANSWER IS:-"
360  FORC=1 TO B
370  PRINT@ 67+64*E+2*C,A(C-1);:NEXTC
380  PRINT:PRINT TAB(5);"THAT IS THE END OF THIS SESSION."
390  PRINT TAB(5);:INPUT"DO YOU WANT TO PLAY AGAIN (1=YES, 2=NO)";C
400  IF (C=1)+(C=2) THEN430
410  PRINT TAB(5);:INPUT"PLEASE ENTER 1 FOR YES, 2 FOR NO";C
420  GOTO400
430  ON C GOTO 60,700
440  REM * CHECK FOR CORRECT POSITION *
450  PRINT TAB(27);: FOR C=1 TO B
460  IF A(19+C)<>A(9+C) THEN480
470  PRINT"* ";: A(9+C)=0: A(19+C)=0
480  NEXTC
490  REM * CHECK FOR CORRECT SOLUTION *
500  FORC=1 TO B
510  IF A(9+C)<>0 THEN720
520  NEXTC
530  REM * SCORING ROUTINE *
```

```
540   F=F+E+J:G=G+1
550   H=F/G
560   PRINT:PRINT"CONGRATULATIONS MASTERMIND!  YOU WORKED IT OUT IN ONLY";
570   PRINTE+J;:IF E+J=1 THEN590
580   PRINT"TRIES.":GOTO600
590   PRINT"TRY."
600   E=1:C=0
610   PRINT:PRINT TAB(7);"YOU HAVE PLAYED";G;: IF G=1 THEN 630
620   PRINT"GAMES THIS SESSION": GOTO640
630   PRINT"GAME THIS SESSION"
640   PRINT TAB(7);"YOUR AVERAGE SCORE IS";INT(10*H)/10;:IFH=1THEN660
650   PRINT"TRIES PER GAME.": GOTO 670
660   PRINT"TRY PER GAME."
670   PRINT:PRINTTAB(7);"DO YOU WANT TO TRY AGAIN (1=YES, 2=NO)";:INPUTC
680   IF (C=1)+(C=2) THEN 690
685   INPUT"PLEASE ANSWER WITH 1 OR 2.";C
687   GOTO680
690   ON C GOTO 100,700
700   CLS:PRINT@ 465,"'BYE FROM SUPER MASTERMIND."
710   FOR C=1 TO 1500: NEXTC: CLS: END
720   REM * CHECK FOR CORRECT VALUE *
730   PRINT TAB(43);
740   FOR C=1 TO B: FOR D=1 TO B
750   IF A(19+C)=0 THEN 790
760   IF A(19+C)<>A(9+D) THEN780
770   PRINT"0 ";:A(9+D)=0: D=B
780   NEXT D
790   NEXT C
800   NEXTE
805   J=J+13:GOTO165
810   REM * INSTRUCTIONS *
820   CLS
830   PRINT"SUPER MASTERMIND ALLOWS YOU TO PIT YOUR WITS AGAINST THE"
840   PRINT"COMPUTER.":PRINT:PRINT"THE COMPUTER SELECTS A SERIES OF";
850   PRINT" NUMBERS AT RANDOM AND 'PRINTS' THEM INVISIBLY AT";
860   PRINT" THE TOP OF THE SCREEN.  YOUR TASK IS TO FIND"
870   PRINT"THE VALUE OF EACH NUMBER AND ITS CORRECT POSITION."
880   PRINT:PRINT"TO DO THIS, YOU ENTER NUMBERS ONE AT A TIME AND";
890   PRINT" THE COMPUTER":PRINT"PLACES THEM IN A ROW ON THE SCREEN";
900   PRINT", UNDER THE INVISIBLE ONES."
910   PRINT"IT THEN COMPARES THE TWO SETS OF NUMBERS.  FOR EACH";
920   PRINT" CORRECT":PRINT"NUMBER IN ITS CORRECT POSITION, THE";
930   PRINT" COMPUTER WILL PRINT AN *,":PRINT"FOR EACH OTHER CORRECT";
940   PRINT" NUMBER IN THE WRONG POSITION, THE"
950   PRINT"COMPUTER WILL PRINT AN 0.  THE GAME ENDS WHEN YOU";
960   PRINT" EXACTLY":PRINT"DUPLICATE THE INVISIBLE NUMBERS."
970   PRINT:PRINT TAB(18);"ENTER ANY NUMBER TO CONTINUE";:INPUTC
980   CLS:PRINT:PRINT
990   PRINT"YOU MAY CHOOSE ANY NUMBER OF COLUMNS, UP TO 10 AND";
1000  PRINT" THE MAXIMUM":PRINT"VALUE FOR THE NUMBERS, UP TO 9.";
1010  PRINT"  (0 IS NOT A VALID NUMBER).":PRINT"THE MORE";
1020  PRINT" COLUMNS YOU HAVE AND THE HIGHER THE MAXIMUM VALUE,";
1030  PRINT" THE":PRINT"HARDER THE GAME.":PRINT
1040  PRINT"AT ANY TIME, YOU MAY ENTER  999 , WHEN THE COMPUTER";
1050  PRINT" WILL":PRINT"DISPLAY THE INVISIBLE NUMBERS AND THE GAME";
1060  PRINT" WILL END.":PRINT:PRINT TAB(26);"GOOD LUCK":PRINT
1080  PRINT:PRINT TAB(18);"ENTER ANY NUMBER TO CONTINUE";:INPUTC
1090  CLS:RETURN
```

```
        * DIGITAL CLOCK *
5 DIM A(15)
10 CLS
20 INPUT "ENTER 1 FOR AM, 2 FOR PM";A
30 IF(A<>1)*(A<>2) THEN 10
40 A$="AM":B$="PM"
50 INPUT"THE HOUR IS";E
60 F=INT(E/10):E=E-(F*10)
70 INPUT"THE MINUTES ARE";C
75 CLS
80 D=INT(C/10):C=C-(D*10)
90 A(1)=C:A(2)=D:A(3)=E:A(4)=F
100 A(11)=C+1:A(12)=D+1:A(13)=E+1:A(14)=F+1
110 GOTO 330
120 A(11)=C:A(12)=D:A(13)=E:A(14)=F
130 N=14736
150 FOR M=1 TO N:NEXT M
160    C=C+1
170    IF C>9 GOTO190
180   GOTO 400
190    C=0
200    D=D+1
210    IF D>5 GOTO230
220    GOTO 400
230    D=0
240    E=E+1
250    IF E>9 GOTO270
260    GOTO290
270    E=0
280    F=F+1
285    IF (F=1)*(E=3) THEN 290
287    GOTO 400
290    C=0:D=0:E=1:F=0
300 IF A=1 THEN 320
310 A=1:GOTO 330
320 A=2
330 ON A GOTO 340,350
340 PRINT @ 502,A$:GOTO 400
350 PRINT @ 502,B$
400    G=92:H=12:IF C<>0 THEN 420
410    GOSUB950
420    ON C GOSUB500,540,600,650,700,750,800,850,900
425    IF D=A(12) THEN 130
430    G=74:IF D<>0 THEN 440
435    GOSUB950
440    ON D GOSUB500,540,600,650,700,750,800,850,900
445    IF E=A(13) THEN 120
450     G=42:IF E<>0 THEN 470
460    GOSUB950
470    ON E GOSUB500,540,600,650,700,750,800,850,900
475    IF F=A(14) THEN 120
477   G=24
480    ON F+1 GOSUB 950,500,540,600,650,700,750,800,850,900
499    GOTO 120
500    REM * PRINT 1 *
505    GOSUB 10000
510    GOSUB6000
515    RETURN
```

```
540      REM * PRINT 2 *
545      GOSUB 10000
550       GOSUB9000
560       GOSUB5000
570       GOSUB7000
575      RETURN
600      REM * PRINT 3 *
605      GOSUB 10000
610       GOSUB9000
620       GOSUB7000
630       GOSUB8000
635      RETURN
650      REM * PRINT 4 *
655      GOSUB 10000
660       GOSUB4000
670       GOSUB2000
680       GOSUB6000
685       RETURN
700      REM * PRINT 5 *
705      GOSUB 10000
710       GOSUB9000
720       GOSUB4000
730       GOSUB8000
740      RETURN
750      REM * PRINT 6 *
755      GOSUB 10000
760       GOSUB9000
770       GOSUB4000
780       GOSUB5000
790       GOSUB8000
795       RETURN
800      REM * PRINT 7 *
805      GOSUB 10000
810       GOSUB1000
820       GOSUB7000
830       GOSUB8000
840      RETURN
850      REM * PRINT 8 *
855      GOSUB 10000
860       GOSUB9000
870       GOSUB4000
875       GOSUB5000
880       GOSUB7000
885       GOSUB8000
890      RETURN
900      REM * PRINT 9 *
905     GOSUB 10000
910       GOSUB9000
920       GOSUB4000
930       GOSUB7000
940       GOSUB8000
945      RETURN
950      REM * PRINT 0 *
955      GOSUB 10000
960       GOSUB1000
965       GOSUB3000
970       GOSUB4000
975       GOSUB5000
```

```
'980      GOSUB7000
985       GOSUB8000:RETURN
1000      REM * TOP HOR. LINE *
1010       Y=H:FORX=G TO G+12:SET(X,Y):NEXTX
1020        RETURN
2000      REM * MIDDLE HOR. LINE *
2010       Y=H+5:FORX=G TO G+12:SET(X,Y):NEXTX
2020        RETURN
3000      REM * BOTTOM HOR. LINE *
3010       Y=H+10:FORX=G TO G+12:SET(X,Y):NEXTX
3020        RETURN
4000      REM * TOP LEFT VERT. LINE *
4010       X=G:FORY=H TO H+5:SET(X,Y):NEXTY
4020        RETURN
5000      REM * BOTTOM LEFT VERT.LINE *
5010       X=G:FORY=H+5 TO H+10:SET(X,Y):NEXTY
5020        RETURN
6000      REM * CENTER VERT. LINE *
6010       X=G+6:FORY=H TO H+10:SET(X,Y):NEXTY
6020        RETURN
7000      REM* TOP RIGHT VERT. LINE *
7010       X=G+12:FORY=H TO H+5:SET(X,Y):NEXTY
7020        RETURN
8000      REM * BOTTOM RIGHT VERT. LINE *
8010       X=G+12:FORY=H+5 TO H+10:SET(X,Y):NEXTY
7508        RETURN
8020     RETURN
9000      REM * ALL HOR. LINES *
9010     FORY=H TO H+10 STEP5:FORX=G TO G+12:SET(X,Y):NEXTX:NEXTY
9020        RETURN
10000 REM * CLEAR OLD NUMBER *
10010 FOR Y=H TO H+10 STEP 5:FOR X=G TO G+12:RESET(X,Y)
10020 NEXT X:NEXT Y
10030 FOR X=G TO G+12 STEP 6:FOR Y=H TO H+10:RESET(X,Y)
10040 NEXT Y:NEXT X:RETURN
```

DUE TO A LAST MINUTE SYSTEM FAILURE, WE HAVE HAD TO CONVERT
SUPER MASTERMIND AND DIGITAL CLOCK INTO LEVEL 2 FORMAT IN
ORDER TO PRINT THEM. THE ONLY CHANGES LEVEL 1 USERS NEED
TO MAKE, ARE TO USE ABBREVIATIONS (ESSENTIAL TO FIT SUPER
MASTERMIND INTO A 4K MACHINE) AND TO TYPE "P. AT" INSTEAD
OF "PRINT AT".
LEVEL 2 USERS MUST STILL MAKE THE CHANGES EXPLAINED ON
PAGE 3.
WE APOLOGISE FOR ANY INCONVENIENCE THIS MIGHT CAUSE OUR
 READERS. THE CASSETTE IS, OF COURSE, UNAFFECTED AS IT
 CONTAINS BOTH LEVEL 1 AND LEVEL 2 VERSIONS.

```
~ COPYRIGHT 1979:PETER GRAHAM HARTLEY, 57 MAIN AVE.,FREWVILLE, SOUTH AUSTRALIA.
1 GOTO32
2 FORK=0TOKK:D$=INKEY$:IFD$=''GOTO4
3 S=ASC(D$):IFS<>91ANDS<>10ANDS<>9ANDS<>8GOTO4:ELSEZ=S
4 NEXT
5 IFZ=10Y=Y+1
6 IFZ=9X=X+2
7 IFZ=8X=X-2
8 IFZ=91Y=Y-1
9 M=15359+(Y)*64+X:C=PEEK(M):IFC=191GOSUB27
10 IFC=42ORC=35GOTO16
11 POKEM,42:KK=KK-.25
12 K1=K1+.125:IFK1>330THENK1=330
13 A1=A1+1:IFA1>K1THENA1=0
14 POKEA(A1),32:A(A1)=M:A2=RND(100):IFA2>94GOSUB29
15 IFA2<85GOSUB30:GOTO2:ELSEGOTO2
16 P=(Y*64+X):Z=RND(5):ONZGOSUB17,18,19,19,20,21:GOTO22
17 D$="CRUMP!":RETURN
18 D$="YAROOH!":RETURN
19 D$="BANG!":RETURN
20 D$="SMASH!":RETURN
21 D$="THUD!":RETURN
22 IFX>57THENX=X-9:GOTO16:ELSEPRINT@P,D$;:D$='':FORT=0TO5000:NEXT:CLS:PRINTCHR$(
23):PRINT"YOUR SCORE WAS";H:PRINT:PRINT"PLAY AGAIN?";
23 D$='':D$=INKEY$:IFD$=''23
24 IFD$="Y"THENRUN41
25 IFD$<>"N"23
26 CLS:PRINTCHR$(23):PRINT"BYE FROM SNAKE...":FORT=0TO5000:NEXT:CLS:END
27 A3=RND(20)*5:H=H+A3:PRINT@984," SCORE";H;"++++++++++++";:RETURN
28 X1=RND(30)*2+1:Y1=RND(11)+1:M=15359+X1*2+Y1*64+1:RETURN
29 GOSUB28:IFPEEK(M)=32THENPOKEM,191:RETURN
30 GOSUB28:IFPEEK(M)=191THENPOKEM,32:A2=RND(40)*5:H=H-A2:GOTO27
31 RETURN
32 CLS:PRINT@532,CHR$(23);"SNAKE":FORT=0TO2000:NEXT
33 PRINT:PRINT:D$='':PRINT"DO YOU REQUIRE INSTRUCTIONS?":PRINT
34 D$=INKEY$:IFD$<>"Y"ANDD$<>"N"34
35 IFD$="N"41
36 CLS:PRINT:PRINT"THIS IS A GAME OF SKILL - WITH JUST A LITTLE BIT OF CHANCE":P
RINT"THROWN IN FOR INTEREST...":PRINT:PRINT"THE GAME IS PLAYED WITHIN THE BOUNDA
RIES (DRAWN ON THE SCREEN)":PRINT"AND THE PLAYER IS REQUIRED TO CONTROL A SNAKE
OF ********* BY"
37 PRINT"SKILFULL USE OF THE FOUR KEYS MARKED WITH ARROWS...":PRINT:PRINT"IF THE
 HEAD OF THE SNAKE SHOULD COLLIDE WITH THE BOUNDARY,":PRINT"OR WITH ANY PART OF
THE SNAKE ITSELF... THE GAME ENDS.":PRINT
38 FORT=0TO8000:NEXT:INPUT"PRESS <<ENTER>> TO CONTINUE...";D$:CLS
39 PRINT:PRINT:PRINT"FROM TIME TO TIME TARGETS WILL APPEAR ON THE SCREEN...":PRI
NT"YOU ARE REQUIRED TO HIT THESE IN ORDER TO INCREASE YOUR SCORE.":PRINT"EACH TA
RGET HAS A LIMITED LIFE... IF IT ISN'T HIT BEFORE ITS":PRINT"LIFE EXPIRES... YOU
R SCORE IS DEBITED
40 PRINT"T W I C E  ITS VALUE.":PRINT"AS THE GAME PROGRESSES THE SNAKE WILL GRAD
UALLY INCREASE IN":PRINT"LENGTH, AND THE SPEED OF THE GAME WILL INCREASE...":PRI
NT:FORT=0TO7000:NEXT:PRINT"G O O D   L U C K !":FORT=0TO4000:NEXT:GOTO41
41 CLEAR200:RANDOM:CLS:POKE16553,255
42 B$=STRING$(63,'#')+' ':FORK=1TO15:PRINTB$;:NEXT:B$=STRING$(63,'#'):PRINTB$;:B
$='#'+STRING$(61,' ')+'# ':PRINT@64,'';:FORK=1TO13:PRINTB$;:NEXT
43 CLEAR10:DEFINTM,A,S,X,Y:DEFDBLH:SV=64:SH=2:DIMA(330):KK=50:X=RND(15)*2+15:Y=R
ND(3)+4:Z=RND(4):ONZGOSUB44,45,46,47:FORM=15423TO16383STEP64:POKEM,128:NEXT:GOTO
9
```

```
44 Z=91:RETURN
45 Z=10:RETURN
46 Z=9:RETURN
47 Z=8:RETURN
48 D$=INKEY$:IFD$=""48
49 Z=ASC(D$):PRINTD$;Z;"   ";:GOTO48
```

LOADER - 16K.

```
7C90:   4C 4F 41 44 45 52 20 32 2E 33 20 42 59 20 45 44
7CA0:   57 49 4E 20 50 41 41 59 20 28 43 29 20 20 31 39
7CB0:   37 39 20 2E 28 48 49 54 20 3C 53 48 49 46 54 3E
7CC0:   20 5C 20 54 4F 20 53 54 41 52 54 2E 29 42 41 53
7CD0:   49 43 20 50 52 4F 47 52 41 4D 20 4C 4F 43 41 54
7CE0:   49 4F 4E 53 20 3A 20 53 54 41 52 54 20 3D 20 58
7CF0:   58 58 58 48 20 2C 20 45 4E 44 20 3D 20 58 58 58
7D00:   58 48 20 20 4C 4F 41 44 45 52 20 32 2E 33 20 54
7D10:   59 50 45 20 3C 42 52 45 41 4B 3E 20 54 4F 20 53
7D20:   54 4F 50 20 57 48 49 4C 45 20 52 45 41 44 49 4E
7D30:   47 2E 20 CD 39 7D C3 19 1A CD C9 01 21 40 3C 22
7D40:   20 40 11 00 3C 21 90 7C 01 3D 00 ED B0 21 63 7D
7D50:   3E C3 32 33 40 22 34 40 AF 32 15 40 21 33 7D 22
7D60:   8E 40 C9 CD E3 03 FE 1A C2 DD 7E ED 73 8A 7C 31
7D70:   88 7C E5 D5 C5 F5 DD E5 FD E5 CD C9 01 21 04 7D
7D80:   11 00 3C 01 2E 00 ED B0 21 FA 3F 22 20 40 2A A4
7D90:   40 E5 DD E1 22 8C 7C CD 2E 7E F3 3E 00 CD 12 02
7DA0:   CD 96 02 CD 35 02 FE D3 28 F9 21 40 3C 06 04 CD
7DB0:   35 02 DD 77 00 DD 23 10 F6 3A 40 38 E6 04 20 44
7DC0:   CD 35 02 DD 77 00 DD 23 B7 28 15 FE 80 30 46 77
7DD0:   23 7C FE 3F 38 E3 7D FE F0 38 DE 21 40 3C 18 D9
7DE0:   3E BF 77 23 CD 35 02 DD 77 00 DD 23 DD E5 D1 CD
7DF0:   D0 7E 06 03 B7 20 B8 CD 35 02 DD 77 00 DD 23 06
7E00:   02 B7 20 AB DD 22 F9 40 DD 22 FB 40 DD 22 FD 40
7E10:   CD F8 01 18 36 E5 D6 80 87 26 7F 6F 5E 23 56 E1
7E20:   1A E6 7F 77 23 13 1A FE 5E 30 8E 77 18 F6 01 D0
7E30:   01 21 50 16 11 00 7F 7E E6 80 C4 44 7E 23 0B 78
7E40:   81 C8 18 F3 EB 73 23 72 23 EB C9 DD 2B DD 22 8E
7E50:   7C 21 CD 7C 11 C0 3F 01 37 00 ED B0 2A A4 40 11
7E60:   E2 3F CD 8D 7E 2A 8E 7C 11 F0 3F CD 8D 7E 2A A4
7E70:   40 E5 23 23 23 23 7E B7 20 FB 23 D1 EB 73 23 72
7E80:   EB 7E B7 20 EC 23 7E B7 28 37 2B 18 E4 7C CB 3F
7E90:   CB 3F CB 3F CB 3F CD B2 7E 7C E6 0F CD B2 7E 7D
7EA0:   CB 3F CB 3F CB 3F CB 3F CD B2 7E 7D E6 0F CD B2
7EB0:   7E C9 FE 0A 38 06 D6 09 F6 40 18 02 F6 30 12 13
7EC0:   C9 FD E1 DD E1 F1 C1 D1 E1 ED 7B 8A 7C C3 CC 06
7ED0:   E5 21 54 7C ED 52 E1 D0 3E 07 C3 F4 1F E5 F5 CD
7EE0:   4D 7D F1 E1 C9 FE
```

LOADER - 4K.

```
4C90:   4C 4F 41 44 45 52 20 32 2E 33 20 42 59 20 45 44
4CA0:   57 49 4E 20 50 41 41 59 20 28 43 29 20 20 31 39
4CB0:   37 39 20 2E 28 48 49 54 20 3C 53 48 49 46 54 3E
4CC0:   20 5C 20 54 4F 20 53 54 41 52 54 2E 29 42 41 53
4CD0:   49 43 20 50 52 4F 47 52 41 4D 20 4C 4F 43 41 54
4CE0:   49 4F 4E 53 20 3A 20 53 54 41 52 54 20 3D 20 58
4CF0:   58 58 58 48 20 2C 20 45 4E 44 20 3D 20 58 58 58
4D00:   58 48 20 20 4C 4F 41 44 45 52 20 32 2E 33 20 54
4D10:   59 50 45 20 3C 42 52 45 41 4B 3E 20 54 4F 20 53
4D20:   54 4F 50 20 57 48 49 4C 45 20 52 45 41 44 49 4E
4D30:   47 2E 20 CD 39 4D C3 19 1A CD C9 01 21 40 3C 22
4D40:   20 40 11 00 3C 21 90 4C 01 3D 00 ED B0 21 63 4D
4D50:   3E C3 32 33 40 22 34 40 AF 32 15 40 21 33 4D 22
4D60:   8E 40 C9 CD E3 03 FE 1A C2 DD 4E ED 73 8A 7C 31
4D70:   88 7C E5 D5 C5 F5 DD E5 FD E5 CD C9 01 21 04 4D
4D80:   11 00 3C 01 2E 00 ED B0 21 FA 3F 22 20 40 2A A4
4D90:   40 E5 DD E1 22 8C 7C CD 2E 4E F3 3E 00 CD 12 02
4DA0:   CD 96 02 CD 35 02 FE D3 28 F9 21 40 3C 06 04 CD
4DB0:   35 02 DD 77 00 DD 23 10 F6 3A 40 38 E6 04 20 44
4DC0:   CD 35 02 DD 77 00 DD 23 B7 28 15 FE 80 30 46 77
4DD0:   23 7C FE 3F 38 E3 7D FE F0 38 DE 21 40 3C 18 D9
4DE0:   3E BF 77 23 CD 35 02 DD 77 00 DD 23 DD E5 D1 CD
4DF0:   D0 4E 06 03 B7 20 B8 CD 35 02 DD 77 00 DD 23 06
4E00:   02 B7 20 AB DD 22 F9 40 DD 22 FB 40 DD 22 FD 40
4E10:   CD F8 01 18 36 E5 D6 80 87 26 7F 6F 5E 23 56 E1
4E20:   1A E6 7F 77 23 13 1A FE 5E 30 8E 77 18 F6 01 D0
4E30:   01 21 50 16 11 00 7F 7E E6 80 C4 44 4E 23 0B 78
4E40:   81 C8 18 F3 EB 73 23 72 23 EB C9 DD 2B DD 22 8E
4E50:   7C 21 CD 4C 11 C0 3F 01 37 00 ED B0 2A A4 40 11
4E60:   E2 3F CD 8D 4E 2A 8E 7C 11 F0 3F CD 8D 4E 2A A4
4E70:   40 E5 23 23 23 23 7E B7 20 FB 23 D1 EB 73 23 72
4E80:   EB 7E B7 20 EC 23 7E B7 28 37 2B 18 E4 7C CB 3F
4E90:   CB 3F CB 3F CB 3F CD B2 4E 7C E6 0F CD B2 4E 7D
4EA0:   CB 3F CB 3F CB 3F CB 3F CD B2 4E 7D E6 0F CD B2
4EB0:   4E C9 FE 0A 38 06 D6 09 FE 40 18 02 F6 30 12 13
4EC0:   C9 FD E1 DD E1 F1 C1 D1 E1 ED 7B 8A 7C C3 CC 06
4ED0:   E5 21 54 7C ED 52 E1 D0 3E 07 C3 F4 1F E5 F5 CD
4EE0:   4D 4D F1 E1 C9 00
```

CONTINUATION OF 'WORDPROC'.

```
3090 IFL>=LATHEN200ELSE3010
3100 L$=LEFT$(A$(L),I):R$=RIGHT$(A$(L),LEN(A$(L))-I)
3110 I=I-1:GOTO3130
3120 L$=LEFT$(A$(L),J):R$=RIGHT$(A$(L),LEN(A$(L))-J):J=J+1
3130 A$(L)=L$+B$+R$:RETURN
3200 GOSUB3250
3210 M$=CHR$(M-128):GOTO3230
3220 GOSUB3250:M$=CHR$(ASC(M$)+128)
3230 A$(L)=L$+M$+R$:RETURN
3250 L$="":IFI>1THENL$=LEFT$(A$(L),I-1)
3260 R$="":IFI<LEN(A$(L))THENR$=RIGHT$(A$(L),LEN(A$(L))-I)
3270 RETURN
```

```
10 CLS:PRINT@340,"BASIC WORD PROCESSOR"
20 REM COPYRIGHT <C> 1979 BY HUBERT S. HOWE, JR.
100 DEFINTA-Z
110 CLEAR7200:NL=120
120 DIMA$(NL)
130 C=0:S=1:L=0:LA=0:SP=1:PN=1:PG=1:PL=50:LL=60:T$=""
140 B$=" ":BB$="  ":CC$=CHR$(143):BR$=CHR$(96):LF$=CHR$(11)
190 CLS
200 INPUT"COMMAND";C$
210 IFC$="F"ORC$="FILL"THENL=LA:GOTO450
220 IFC$="C"ORC$="CLEAR"THEN110
230 IFC$="D"ORC$="DELETE"THEN1200
240 IFC$="R"ORC$="REPLACE"THEN1300
250 IFC$="I"ORC$="INSERT"THEN1400
260 IFC$="P"ORC$="PRINT"THEN1000
270 IFC$="N"ORC$="NUM"THEN1500
280 IFC$="M"ORC$="MOVE"THEN1600
290 IFC$="H"ORC$="HELP"THEN1700
300 IFC$="SET"THEN1800
310 IFC$="E"ORC$="EDIT"THEN1900
330 IFC$="L"ORC$="LOAD"THEN2100
340 IFC$="S"ORC$="SAVE"THEN2200
350 IFC$="J"ORC$="JUSTIFY"THEN3000
390 GOTO200
450 CLS:IFL<1THEN500:REM FILL
460 FORJ=0TOL-1:C=J*64:IFC>958THENPRINT@960,B$:C=896
470 PRINT@C,J;:IFLEN(A$(J))<1THEN490
480 PRINTTAB(4);A$(J);
490 PRINT:NEXTJ
500 GOSUB700:PRINT@C,L;TAB(4);:C=C+4
505 IFLEN(A$(L))>0THENS=LEN(A$(L)):PRINT@C,A$(L):C=C+S
510 PRINT@C,CC$;:A$=INKEY$:IFA$=""THENPRINT@C,B$;:GOTO510
520 A=ASC(A$)
530 IFA=13THENA$=B$:GOTO750
540 IFA=8THEN800
550 IFA=9THEN900
560 IFA=24THEN850
570 IFA=10THEN740
580 IFA$=BR$THENPRINT:GOTO200
590 IFA=25THEN950
600 PRINT@C,A$;:A$(L)=A$(L)+A$:S=S+1:C=C+1:IFS<=LLTHEN510
610 S=1:L=L+1
620 IFL>NLTHENL=NL:PRINT"BUFFER FULL":GOTO200
625 IFLEN(A$(L))<>0THENPRINT:GOTO200
630 IFL>LATHENLA=L
635 IFA$=B$THEN500
640 FORI=LLTO0STEP-1:IFMID$(A$(L-1),I,1)=B$THEN660
650 C=C-1:NEXTI:GOTO500
660 IFI=LLTHEN500ELSEFORJ=I+1TOLL:PRINT@C,B$;
665 A$(L)=A$(L)+MID$(A$(L-1),J,1)
670 S=S+1:C=C+1:NEXTJ:A$(L-1)=LEFT$(A$(L-1),I-1):GOTO500
700 C=L*64:IFC<959THENRETURNELSEPRINT@960,B$:C=896:RETURN
740 A$(L)=A$(L)+LF$:A$=B$:REM DOWN ARROW
750 PRINT@C,A$;:GOTO610:REM ENTER
800 IFS<2THEN510:REM BACKSPACE
810 C=C-1:PRINT@C,BB$;:S=S-1:A$(L)=LEFT$(A$(L),S-1):GOTO510
850 IFS<2THEN510:REM SHIFT LEFT ARROW
860 A$(L)="":S=1:GOSUB700:C=C+4
```

```
870 PRINT@C,CHR$(30):GOTO510
900 IFS>LL-6THEN510:REM RIGHT ARROW
910 FORI=STOS+4:A$(L)=A$(L)+B$:PRINT@C,B$;:C=C+1:NEXTI
920 S=S+5:GOTO510
950 IFS>=LLTHEN610:REM SHIFT-RIGHT ARROW
960 PRINT@C,B$:FORI=STOLL:A$(L)=A$(L)+B$:NEXTI:GOTO510
990 INPUT"PRESS ENTER WHEN READY TO CONTINUE";EN$:RETURN
1000 CLS:INPUT"PRINT LINE NUMBERS (Y/N)";N$:REM PRINT
1010 INPUT"LINE PRINTER OR VIDEO DISPLAY (P/V)";P$
1020 IFP$="P"THEN1100
1030 CLS:PP=PG-1:FORK=0TOLASTEPPL:PP=PP+1
1040 IFPN<1THEN1050ELSEPRINTT$;
1045 PRINTTAB(LL-7);"PAGE";:PRINTUSING"###";PP
1050 FORJ=KTOK+PL-1:IFJ>LATHEN1090
1055 IFSP>0THENFORM=1TOSP:PRINT:NEXTM
1060 IFN$="Y"THENPRINTJ;
1070 PRINTA$(J);
1080 NEXTJ:GOSUB990
1090 PRINTCHR$(30):NEXTK:GOTO200
1100 IFPEEK(14312)<128THEN1130
1110 INPUT"LINE PRINTER NOT READY.  ABORT? (Y/N)";A$
1120 IFA$="Y"THEN200ELSEGOSUB990:GOTO1100
1130 PP=PG-1:FORK=0TOLASTEPPL:PP=PP+1
1140 IFPN<1THEN1150ELSEPRINTT$
1145 LPRINTTAB(LL-7);"PAGE";:LPRINTUSING"###";PP
1150 FORJ=KTOK+PL-1:IFJ>LATHEN1190
1155 IFSP>0THENFORM=1TOSP:LPRINT:NEXTM
1160 IFN$="Y"THENLPRINTJ;
1170 LPRINTA$(J);
1180 NEXTJ:GOSUB990
1190 LPRINTLF$:NEXTK:GOTO200
1200 INPUT"FIRST LINE TO DELETE";N1:REM DELETE
1210 INPUT"LAST LINE TO DELETE";N2:IFN1>NLTHENN1=NL
1220 IFN1<0THENN1=0
1230 IFN2>NLTHENN2=NL
1240 FORI=N1TON2:A$(I)="":NEXTI:GOTO200
1300 INPUT"LINE NUMBER";L:REM REPLACE
1310 IFA$(L)=""THENPRINT"CAN'T REPLACE LINE";L:GOTO200
1320 A$(L)="":GOTO450
1400 INPUT"STARTING LINE";L:REM INSERT
1410 IFLEN(A$(L))=0THEN450
1420 PRINT"CAN'T INSERT AT LINE";L:GOTO200
1500 K=0:REM NUMBER
1510 IFK>=LATHEN200
1520 IFLEN(A$(K))<>0THENK=K+1:GOTO1510
1530 FORJ=KTOLA-1:A$(J)=A$(J+1):NEXTJ
1540 A$(LA)="":LA=LA-1:GOTO1510
1600 INPUT"FIRST LINE TO MOVE2;N1:REM MOVE
1610 INPUT"LAST LINE TO MOVE";N2
1620 INPUT"FIRST NEW LINE";N3
1630 FORJ=N1TON2
1640 IFLEN(A$(N3))SQR0THEN1650
1645 PRINT"LINE";N3;"ALREADY OCCUPIED.":+200
1650 A$(N3)=A$(J):A$(J)=""
1660 N3=N3+1:IFN3>LATHENLA=N3
1670 NEXTJ:GOTO200
1700 CLS:PRINT"LEGAL cOMMANDS ARE AS FOLLOWS:"
1710 PRINT"F  FILL","P  PRINT","C  CLEAR"
```

```
1720 PRINT"R  REPLACE","I  INSERT","D  DELETE"
1730 PRINT"N  NUMBER","M  MOVE","E  EDIT"
1740 PRINT"L  LOAD","S  SAVE","H  HELP"
1750 PRINT"J  JUSTIFY","  SET"
1760 PRINT"TYPE 'SHIFT-@' TWICE TO RETURN TO COMMAND MODE."
1770 GOTO200
1800 CLS:INPUT"LINE LENGTH";LL
1810 INPUT"PAGE LENGTH";PL
1820 INPUT"SPACING (1=SINGLE, 2=DOUBLE)";SP
1830 INPUT"PAGE NUMBERING (1=YES, 0=NO)";PN
1840 IFPN>0THENINPUT"STARTING PAGE NUMBER";PG
1850 INPUT"TITLE";T$
1890 GOTO200
1900 CLS:INPUT"LINE NUMBER";L:REM EDIT
1910 IFLEN(A$(L))<1THENPRINT"CAN'T EDIT LINE";L:GOTO200
1920 PRINTL;TAB(4);A$(L);
1930 C=68:S=1:IFL>99THENC=69
1940 GOSUB2520
1950 IFA$=" "THEN2500
1960 IFASC(A$)=8THENGOSUB2510:GOTO1940
1970 IFASC(A$)=13THEN2900
1980 IFA$="I"THEN2600
1990 IFA$="D"THEN2650
2000 IFA$="H"THEN2750
2010 IFA$="X"THEN2800
2020 IFA$="C"THEN2850
2040 IFA$="S"THEN2950
2050 GOTO1940
2100 GOSUB2300:IFA$="D"THEN2150:REM LOAD
2110 INPUT#-1,LA,SP,PN,PG,PL,LL,T
2115 IFT>0THENINPUT#-1,T$
2120 FORL=0TOLA:INPUT#-1,A$(L)
2130 NEXTL:GOSUB2400:GOTO200
2150 OPEN"I",1,F$
2160 INPUT#1,LA,SP,PN,PG,PL,LL,T
2165 IFT>0THENINPUT#1,T$
2170 FORL=0TOLA:INPUT#1,A$(L)
2180 NEXTL:CLOSE:GOSUB2400:GOTO200
2200 GOSUB2300:GOSUB2450:IFA$="D"THEN2250:REM SAVE
2210 PRINT#-1,LA,SP,PN,PG,PL,LL,LEN(T$)
2215 IFLEN(T$)>0THENPRINT#-1,T$
2220 FORL=0TOLA:PRINT#-1,A$(L)
2230 NEXTL:GOSUB2400:GOTO200
2250 OPEN"O",1,F$
2260 PRINT#1,LA,SP,PN,PG,PL,LL,LEN(T$)
2265 IFLEN(T$)>0THENPRINT#1,T$
2270 FORL=0TOLA:PRINT#1,A$(L)
2280 NEXTL:CLOSE:GOSUB2400:GOTO200
2300 CLS:INPUT"TAPE OR DISK (T/D)";A$
2310 IFA$<>"D"THEN2330
2320 INPUT"FILESPEC";F$:RETURN
2330 PRINT"READY CASSETTE":GOSUB990:RETURN
2400 FORL=0TOLA:IFLEN(A$(L))<1THEN2440:UNTRANSLATE
2410 FORI=1TOLEN(A$(L)):M=ASC(MID$(A$(L),I,1))
2420 IFM>128THENGOSUB3200
2430 NEXTI
2440 NEXTL:RETURN
2450 FORL=0TOLA:IFLEN(A$(L))<1THEN2490:TRANSLATE
```

```
2460 FORI=1TOLEN(A$(L)):M$=MID$(A$(L),I,1)
2470 IFM$=","ORM$=":"ORM$=" "ORM$=CHR$(34)THENGOSUB3220
2480 NEXTI
2490 NEXTL:RETURN
2500 IFS>=LEN(A$(L))THEN1940
2505 PRINT@C,MID$(A$(L),S,1);:S=S+1:C=C+1:GOTO1940
2510 IFS<2THENRETURN
2515 PRINT@C,MID$(A$(L),S,1);:S=S-1:C=C-1:RETURN
2520 PRINT@C,CC$;:A$=INKEY$:IFA$<>""THEN2540
2530 PRINT@C,MID$(A$(L),S,1);:GOTO2520
2540 IFASC(A$)=13THEN2900ELSEIFASC(A$)=8THENGOSUB2510:GOTO2520
2545 RETURN
2550 IFLEN(A$(L))>=LLTHEN1940ELSERETURN
2560 IFLEN(A$(L))<1THEN1940ELSERETURN
2570 PRINTCHR$(8);
2575 IFS>=LEN(A$(L))THEN1940ELSERETURN
2580 L$="":IFS>1THENL$=LEFT$(A$(L),S-1):RETURN
2590 R$="":IFS<LEN(A$(L))THENR$=RIGHT$(A$(L),LEN(A$(L))-S+1)
2595 RETURN
2600 GOSUB2550:REM INSERT
2610 GOSUB2520:PRINT@C,A$;:IFASC(A$)=13THEN2900
2615 IFASC(A$)=27THEN1940ELSEIFASC(A$)=8THENGOSUB2510:GOTO2610
2620 GOSUB2580:GOSUB2590
2630 A$(L)=L$+A$+R$:S=S+1:C=C+1:PRINTR$;:GOTO2610
2650 GOSUB2560:GOSUB2570:PRINTCHR$(30);:REM DELETE
2660 GOSUB2580:S=S+1:GOSUB2590:S=S-1
2670 A$(L)=L$+R$:PRINTR$:GOTO1940
2750 PRINTCHR$(30):REM HACK
2760 IFS>1THENA$(L)=LEFT$(A$(L),S-1)
2770 GOSUB2520:PRINT@C,A$;:A$(L)=A$(L)+A$
2780 C=C+1:S=S+1:GOTO2770
2800 GOSUB2550:GOSUB2570:REM X
2810 GOSUB2590:PRINTR$;:C=C+LEN(A$(L))-S+1
2820 S=LEN(A$(L))+1:GOTO2750
2850 GOSUB2520:REM CHANGE
2860 PRINT@C,A$
2870 GOSUB2580:S=S+1:GOSUB2590:S=S-1
2880 A$(L)=L$+A$+R$:S=S+1:C=C+1:GOTO1940
2900 IFLEN(A$(L))>=LLTHEN2920:REM ENTER
2910 FORI=STOLEN(A$(L)):PRINT@C,MID$(A$(L),I,1);:C=C+1:NEXTI
2920 PRINT:S=1:IFLEN(A$(L))<=LLTHEN200
2930 PRINT"LINE";L;"CONTAINS";LEN(A$(L));"CHARACTERS.":GOTO200
2950 GOSUB2575:GOSUB2520:PRINT@C,MID$(A$(L),S,1);:REM SEARCH
2960 CS=C+1:FORI=S+1TOLEN(A$(L)):IFMID$(A$(L),I,1)<>A$THEN2980
2970 S=I:C=CS:GOTO1940
2980 CS=CS+1:NEXTI:GOTO1940
3000 L=-1:REM JUSTIFY
3010 L=L+1:IFLEN(A$(L))<10ORLEN(A$(L))>=LLTHEN3090
3015 E$=RIGHT$(A$(L),1)
3016 IFE$="."ORE$="!"ORE$="?"ORE$=")"ORE$=":"ORE$=LF$THEN3090
3020 FORI=1TOLEN(A$(L)):IFMID$(A$(L),I,1)<>B$THENNEXTI:GOTO3090
3040 I=FIX(LEN(A$(L))/2):J=I+1
3050 IFMID$(A$(L),I,1)=B$THENGOSUB3100
3060 I=I-1:IFI<2THEN3010
3065 IFLEN(A$(L))>=LLTHEN3090
3070 IFMID$(A$(L),J,1)=B$THENGOSUB3120
3080 J=J+1:IFJ>=LLTHEN3010
3085 IFLEN(A$(L))<LLTHEN3050
```

CONTINUED BENEATH 'LOADER - 4K'.