



Tired of trying to decipher Applesoft listings? Use this set of programs to produce formatted listings and flowcharts to help check your logic.



FEATURE ARTICLE

THE NAVIGATOR

by Lee Swoboda, Padapple Consulting, 1451 NE Paulson Rd., Poulsbo, WA 98370

Writing documentation can be the most difficult part of programming. At a minimum, documentation should include a well annotated listing, sample printouts if appropriate, and a flowchart. It can be a lot of work — but just try to modify an undocumented program after the trail is “cold.” That’s when I remember that solid documentation is very important.

Applesoft’s LIST command never seems quite adequate for the task. The 34-column format and uneven indentation make listings difficult to read. A POKE 33,33 expands the listing to 80 columns, but it still lacks a professional appearance: no margins, no pagination and no word wraparound. And I’d like to see FOR-NEXT loops indented à la Pascal. Now what if we take this imaginary listing formatter and make it print flowcharts as well?

The Navigator is a BASIC listing utility and flowchart formatter that includes all of these features. Since it reformats only one record at a time, Navigator will handle any size program. Unfortunately, string manipulation is the slowest process in Applesoft, so it takes an average of five seconds per line to print a flowchart. This is much faster than you can draw, but seems very slow by the standards of most computer functions.

PROGRAM OPERATION

To start a session with Navigator, run NAVIGATOR.1. Two other modules, NAVIGATOR.2 and NAVIGATOR.3, will automatically be run by NAVIGATOR.1. Be sure the write-protect tabs are removed from both the Navigator and program disks. In addition, there must be enough room on your Navigator disk for a text version of your program. This file will take up about 50% more space than the program, depending on the nature of the program. It is deleted by Navigator at the end of the printout. If you are using one drive, copy your program onto the Navigator disk.

Three sets of input must be provided at the beginning of the run:

1. In the Program Data menu, enter your name, the name of the program to be formatted by Navigator, and the date. The date may be any string of nine or fewer characters.

2. The Data Format menu displays default output values. The following values are recommended:

```
LEFT MARGIN      8
RIGHT MARGIN     72
LINES PER PAGE   66
BOTTOM MARGIN    5
PRINT REMARKS?   Y
NAVIGATOR IN DRIVE 1
PROGRAM IN DRIVE 2
PRINTER IN SLOT  1
```

3. At the Selection menu, select the desired printout — a formatted listing, a flowchart, or both.

If your system has two drives, place the disk containing Navigator in drive 1 and the disk containing the program to be formatted in drive 2. From here on, operation is automatic. If you have only one drive, just leave the disk containing Navigator and the copy of your program in the drive.

A "normal" Applesoft listing is shown in Figure 1. When formatted by Navigator, the same listing will appear as shown in Figures 2 and 3. Figure 3 shows only a few of the possible symbols. Figure 4 shows how other BASIC structures are handled, and Figure 5 shows how the I/O symbols used compare to standard flowchart symbols.

ENTERING THE PROGRAMS

Enter NAVIGATOR.1 as it is shown in Listing 1. Be careful to type lines 1910 and 2430 exactly as shown, paying particular attention to the placement of quote marks. For ProDOS, substitute CAT for CATALOG in line 400. Save the completed listing before you run it with the command:

SAVE NAVIGATOR.1

Enter NAVIGATOR.2 as it is shown in Listing 2. For ProDOS you must delete lines 1370-1380. Save the resulting program with:

SAVE NAVIGATOR.2

Enter NAVIGATOR.3 as it is shown in Listing 3. For ProDOS you must delete lines 4140-4150. Save the resulting program with:

SAVE NAVIGATOR.3

FIGURE 1: Applesoft Sample Listing

```
10 REM *****
20 REM * SAMPLE PROGRAM FOR *
30 REM * DEMONSTRATION OF *
40 REM * NAVIGATOR PROGRAM *
50 REM *****
60 PRINT "SQUARE PRINTER"
70 INPUT "HOW MANY NUMBERS? ";N
80 PRINT CHR$(4)"PR#1"
90 FOR I = 1 TO N
100 PRINT I;" SQUARED IS ";I * I
    ;"."
110 NEXT
120 PRINT CHR$(4)"PR#0"
140 PRINT CHR$(4)"OPEN SQUARES
: PRINT CHR$(4)"WRITE SQU
ARES": FOR I = 1 TO N: PRINT
I: PRINT I * I: NEXT : PRINT
CHR$(4)"CLOSE"
150 PRINT "DONE"
```

FIGURE 2: Formatted Listing

```
10 REM *****
20 REM * SAMPLE PROGRAM FOR *
30 REM * DEMONSTRATION OF *
40 REM * NAVIGATOR PROGRAM *
50 REM *****
60 PRINT "SQUARE PRINTER"
70 INPUT "HOW MANY NUMBERS? ";N
80 PRINT CHR$(4)"PR#1"
90 FOR I = 1 TO N
100 PRINT I;" SQUARED IS ";I * I; "."
110 NEXT
120 PRINT CHR$(4)"PR#0"
140 PRINT CHR$(4)"OPEN SQUARES"
: PRINT CHR$(4)"WRITE SQUARES"
: FOR I = 1 TO N
: PRINT I
: PRINT I * I
: NEXT
: PRINT CHR$(4)"CLOSE"
150 PRINT "DONE"
```

For help in entering *Nibble* listings, see "A Welcome to New *Nibble* Readers" at the beginning of this issue.

THE PROGRAMMING APPROACH

An Applesoft program is normally stored starting at memory location \$801. It is not stored in the same text form that you see it on the screen. If it were, the program would take up more space, and would execute considerably slower. Instead, the program is stored in a tokenized form. Each keyword, such as PRINT, FOR, THEN,

Applesoft's LIST command never seems quite adequate. . .



or GOSUB, is stored as a single, unique number, called a token, that represents that keyword. For instance, \$97 represents the keyword HOME. In addition, the line numbers are represented by two hexadecimal bytes in low-high order. For example, line 1000 (hex 3E8) is represented by the bytes \$E8 \$03.

Two bytes that you don't see in a listing are added to each line. These are called a line link, and they point to the address of the next program line. Figure 6 shows how two consecutive program lines are stored in the addresses \$801-\$811. The lines are separated by zero bytes, and each line begins with the line link, followed by the line number, and the tokenized representation of the line.

The 6502 processor in the Apple cannot understand the tokenized code. It must run a built-in program called the Applesoft interpreter, which figures out what the tokenized program means and directs control to various built-in machine language routines as needed. For a more extensive discussion of how Applesoft lines are interpreted, see Sandy Mossberg's excellent explanation in "Disassembly Lines: Parse," *Nibble* Vol. 3/No. 7, and pp. 121-128 in the *Applesoft II BASIC Programming Reference Manual*.

There are two ways to format the tokenized code. The first is to replace the function of the Applesoft editor with a program of our own, written in either machine language or BASIC. (See *The Apple Orchard*, Winter 1980, for an example of this approach.)

FIGURE 3: Formatted Flowchart

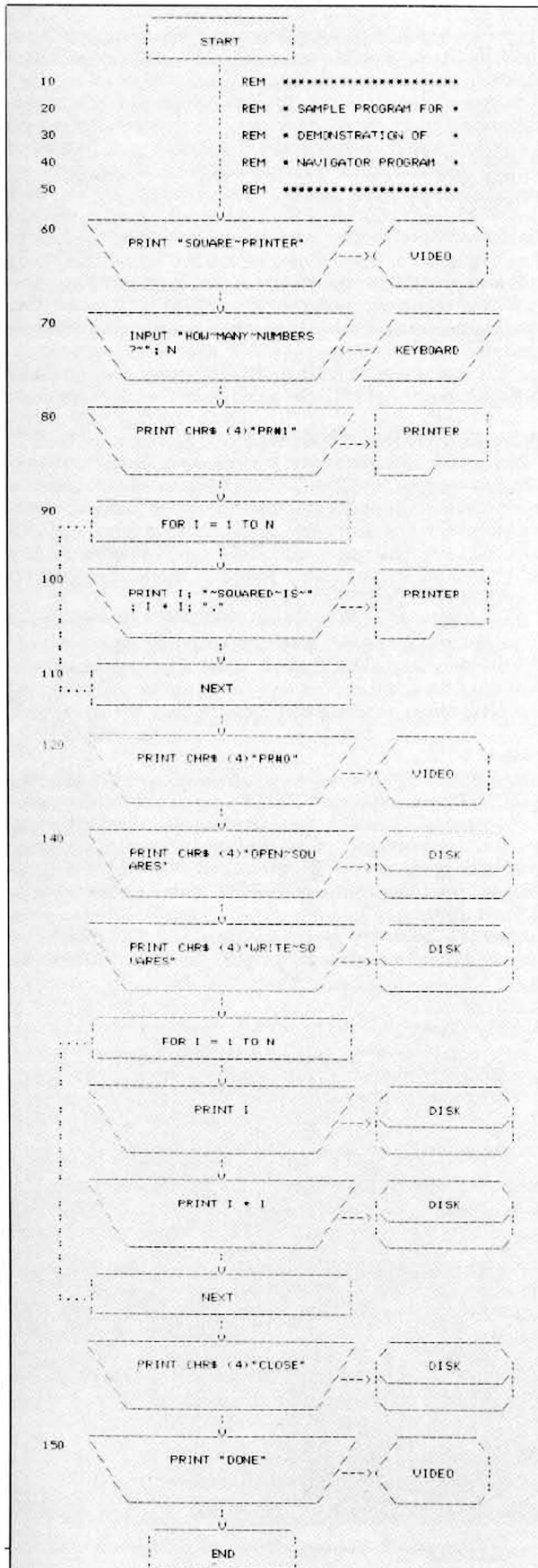


FIGURE 4: Program Flow Symbols

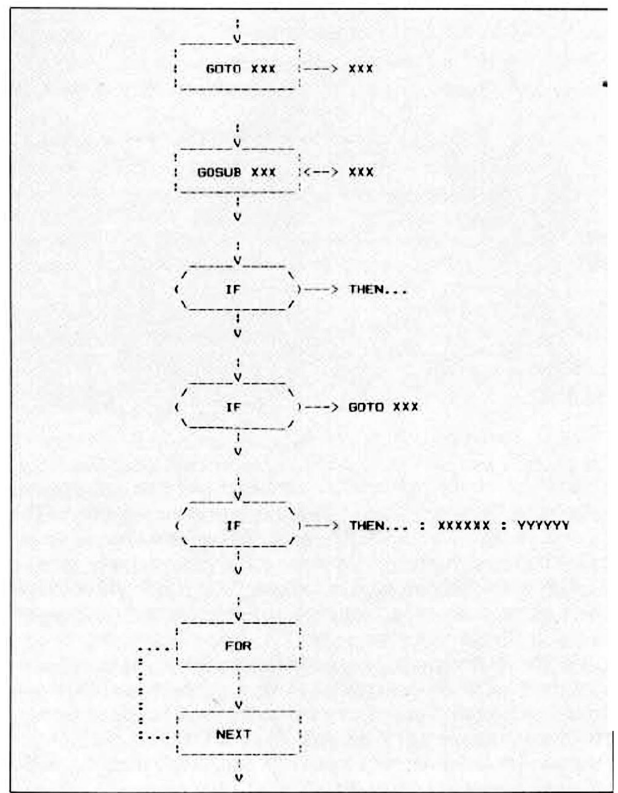


FIGURE 5: Flowchart Symbols

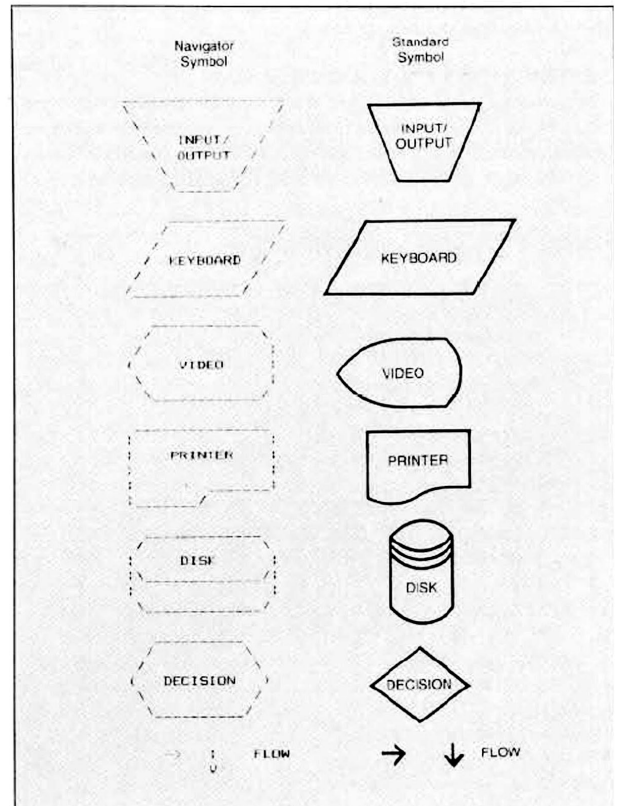


TABLE 1: Format Variables

Variables	Explanation
NAME\$	The name of the previous program that was processed
AUTH\$	The name of the programmer
DT\$	The date
SELECT	Output selection (i.e., listing, flowchart, or both)
COMMENTS	Suppress REMarks?
LMARGIN	Left margin
RMARGIN	Right margin (measured from the left-hand side of the page)
LPAGE	Lines per page
BMARGIN	Bottom margin
D1	Drive number that contains the Navigator disk
D2	Drive number that contains the program to be formatted
SLOT	Slot number of printer

This would require that both the formatter program and the program being formatted be resident in memory at the same time. The formatter program would PEEK its way through the other program, converting each Applesoft token to text in the same way as does the Applesoft LIST routine. The formatter program would be stored above the program to be formatted, and would therefore limit the size of the latter program.

The second method is the one used in Navigator. Applesoft converts the tokenized code to text whenever it performs the LIST command, but instead of printing the text to the video screen or printer, we have it print the text to the disk. Since the Applesoft editor terminates each listed line with a carriage return, this method is fully compatible with sequential text files, which separate records with a carriage return. The amount of overhead consumed by the formatter program is immaterial, since only one line of text is in memory at any one time. Processing text files is somewhat slower than PEEKING through memory, but we gain the advantage of being able to handle programs of any size.

RESTRICTIONS ON YOUR PROGRAM

Navigator is able to properly track the flow of nearly any program. However, a few restrictions apply. Navigator may have trouble with multiple NEXTs for the same FOR, and it can only handle a limited depth of FOR-NEXT nesting (depending on line length).

If you choose the "omit REM statements" option and you branch to REMs in your program, it will appear to branch to nonexistent lines.

Navigator recognizes three different methods of specifying disk commands:

1. PRINT CHR\$(4)"XXX"
2. PRINT D\$"XXX"
3. PRINT "<CTRL>DXXX"

D\$ is the only variable interpreted this way, and it should be reserved for DOS operations exclusively.

Don't use <CTRL>M directly in your program. If you need to use a carriage return within a string, use CHR\$(13) instead. Carriage returns are used to delimit program lines, so extras will cause unwanted results.

If I/O change takes place in the THEN portion of an IF-THEN statement, no change will take place in the flowchart symbols.

HOW THE PROGRAM WORKS

Navigator is a sophisticated text editor written in three modules. The first module, NAVIGATOR.1, creates a sequential text file in which each program line is a separate record. If you choose "listing only" or "flowchart only" in the Selection menu of NAVIGATOR.1, the program automatically runs NAVIGATOR.2 or NAVIGATOR.3. If you select "both", the computer runs NAVIGATOR.2 first, then NAVIGATOR.3.

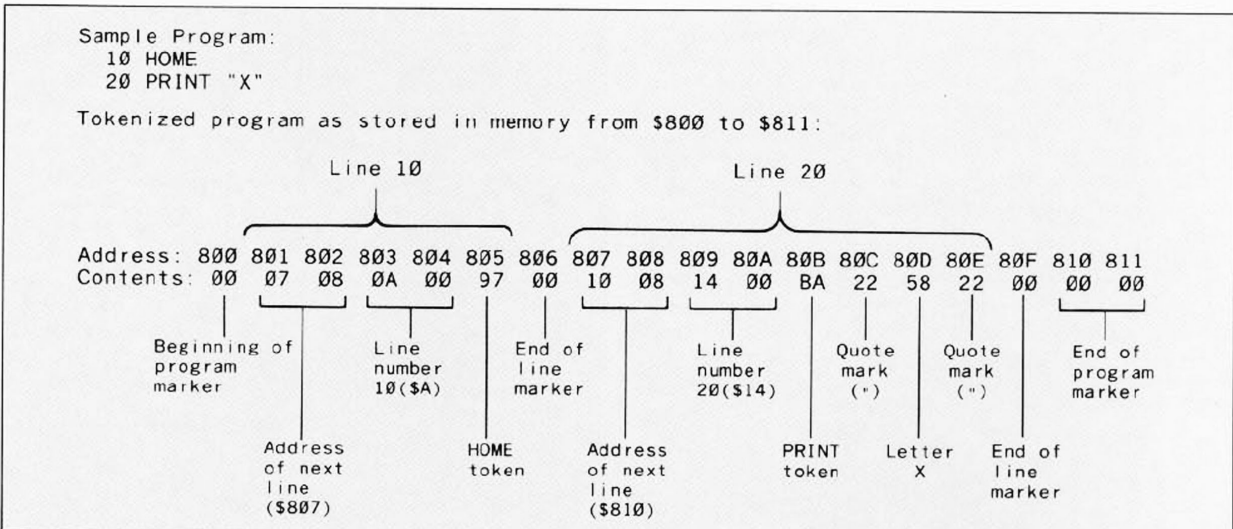
Both modules 2 and 3 read the text file, divide each record into substrings, and position each substring on a printed page. Both modules use the standard ASCII character set, so no special graphics capabilities are necessary. Likewise, all formatting is done by concatenating strings, so special printer control codes are not required.

Module 1

NAVIGATOR.1 first requests data such as name, date, and margins. It then creates and EXECs a sequential text file called EXE. When it is EXECed, EXE creates a second sequential text file, EXX, in which each record is a listing of one line of the program being processed. NAVIGATOR.2 and NAVIGATOR.3 read EXX and reformat each record/program line into a listing or flowchart, respectively.

Lines 130-260 read the text file NAVIGATOR.TXT, which contains the current formatting information (Table 1). If there is no

FIGURE 6: Tokenized Applesoft Program



NAVIGATOR.TXT file (indicated by an end-of-data error) a new one will automatically be set up. Lines 270-620 prompt the user for information about the program to be processed: the program name, programmer's name, and date. Lines 630-800 present the current values for eight specific format and system parameters.

Lines 810-850 offer the user the option to change any of these parameters. If the user elects to change a value, lines 890-900 identify which parameter will be changed. Lines 940-970 enter the new value, and lines 980-1160 screen the new value. After editing, the program returns to line 660 to allow additional editing, if desired.

Lines 1170-1370 require the user to select the desired output; lines 1360 and 1370 select the correct module for that output.

Lines 1420-1620 save the format and system parameters for later use by NAVIGATOR.2 and NAVIGATOR.3. Lines 1630-1660 verify that a program of the name given exists by attempting to rename it to the same name. The error trap allows the user to recover from entering a wrong file name, leaving the write-protect tab on, and other common errors. Lines 1670-1940 and lines 2290-2470 launch Navigator into action.

The EXEC command is a unique feature of Apple's disk operating systems. It causes records in a sequential text file to be executed as direct commands (see pp. 74-76 of *The DOS Manual* or pp. 133-

Line 750 converts control characters (ASCII 1-30), which are normally "invisible" in a listing, to two-character "visible" representations consisting of a caret (^) and a lower-case letter (e.g., <CTRL> D becomes ^d). Line 770 converts spaces within quotation marks to tildes (~), so there's no need to guess at the number of spaces (something I find particularly frustrating). Line 850 divides multiple-statement program lines into separate print lines.

To improve the appearance of the listing, Applesoft's spacing is adjusted. Applesoft adds spaces in places that seem awkward to me, so Navigator simply removes them. Line 870 removes double spaces. Line 880 removes spaces after a left parenthesis. Line 890 removes spaces before a left parenthesis. Line 900 adds a space after a comma; e.g., MID\$(A\$,7,3) becomes MID\$(A\$, 7, 3).

Lines 930-1200 print the new line, A\$(J). The subroutine at line 1540 is called to split long lines. Line 990 will skip REMARKS if you selected that option in the Data Format menu of NAVIGATOR 1. Lines 1000 and 1010 skip a line before and after REMARKS to highlight them in the listing. This occurs for "REMARKS only" lines; REMARKS that are added on at the end of other lines are not highlighted.

Line 960 skips to a new page and prints a header when the existing page is full. Line 1020 increments the line counter by one for each new line. Lines 1030 and 1130 indent and "unindent" FOR-NEXT loops. Line 1050 advances the print head to the left margin. Lines 1020-1080 print the follow-on statements in a multiple-statement line. Lines 1090-1110 print the line number, indentation (if any) and text. Line 1170 loops to print multiple-statement lines, and lines 1180-1190 re-initialize the REMARKS and THEN flags for the next round (line 1200).

Module 3

NAVIGATOR.3 reads each record from the text file EXX and reformats it in a manner similar to NAVIGATOR.2 (lines 550-990). Rather than simply printing the reformatted line, however, NAVIGATOR.3 reformats the string a second time, adding the appropriate characters to form the characteristic outlines shown in Figures 4 and 5. In order to provide the proper shape, the main program must analyze the function of the line being formatted:

1. Is it an input/output function?
2. Does it change control of input/output (I/O) devices?
3. If it is an I/O function, what is the source/destination of data?
4. Does it affect the flow of program logic?

Each of these situations is represented differently on a flowchart. In addition, the program must distinguish between conditional and unconditional transfers, and must identify other constructions such as FOR-NEXT loops (to multiple levels).

Once the program has analyzed the function of the line, it must format the line into a corresponding graphical representation.

Lines 1000-1370 look for changes in I/O control. In order to avoid disconnecting DOS, all such commands must be in the format PRINT D\$;"XXX", PRINT CHR\$(4)"XXX" or PRINT "<CTRL>DXXX". Lines 820, 830, and 960 test for these changes of control and set the flag DOS to 1. At line 1080 this flag is checked: if it is 1 then the following characters are checked to determine exactly how PRNT, SHAPE, and FLAG should be set; if it is 0 then the I/O checks are skipped. Unmodified input and output are directed to the keyboard and video screen, respectively.

Lines 1380-1450 assign a shape to each line, depending on its function. Most lines are assigned shape 1 (a rectangle), since this represents the most common function. Input/output functions are assigned shape 2 (a trapezoid), and decision functions, shape 3 (a diamond). Shapes are processed at line 2020, depending on the value of the variable SHAPE. Lines 2030-2300, 2310-2560 and 3030-3480 process shapes 1, 2 and 3, respectively.

Shape 1 functions are further analyzed for GOTO and GOSUB

The amount of overhead consumed by the formatter program is immaterial, since only one line of text is in memory at any one time.

142 of *BASIC Programming with ProDOS*). Thus, when line 1940 of NAVIGATOR.1 EXECs the text file EXE, the four records in EXE (created by lines 1860-1930) appear to the computer as if they were being entered directly from the keyboard.

The first record (line 1880) erases the previous program (NAVIGATOR.1) from memory. The second record (line 1890) loads the program to be formatted. The third record (line 1910) adds line 0 to the program. The fourth record (line 1920) runs the modified program. Now here's the trick — execution of the modified program never gets past line 0. Line 0 creates a new text file, EXX. It then lists the program into EXX, cleverly excluding itself from the listing, and runs either NAVIGATOR.2 or NAVIGATOR.3, as appropriate. All this is carried out automatically.

A nearly identical process takes place in lines 2290-2450 for a one-drive configuration. The EXEC file is called EX1, and the contents of the inserted line 0 are different.

Module 2

NAVIGATOR.2 edits and reformats each record in file EXX. Lines 450-980 are a one-pass editor. The computer scans the record once and constructs a new string or strings based on several criteria:

1. Is the record a REMARK or DATA statement? If so, the remainder of the string is concatenated without further adjustment.
2. Is the record a multiple-statement line? If so, it is concatenated as separate lines.
3. Is the record a conditional statement? If so, the portion from THEN on is concatenated as a separate line.
4. Is the record a string? If so, the portion between quotation marks is concatenated without further adjustment.

Lines 610-920 read through each record one character at a time and build a new record, A\$(J). Lines 720-920 examine each character of the record and place it in the new string, A\$(J).

statements (lines 2140-2150). Note that GOTO interrupts the main logic flow. Shape 3 functions are also analyzed for logic flow (lines 1710-1840), but since IF-THEN statements are always conditional transfers of control, the main logic flow is not interrupted.

Lines 1090-1370 assign a second shape to all I/O functions, depending on the source or destination of the data. These additional shapes are added at line 2530, depending on the value of the variable FLAG. The following symbols are added:

Function	Symbol	Lines
Screen output	Video display	2570-2670
Keyboard input	Parallelogram	2680-2780
Printer output	Document	2790-2890
Disk input/output	Disk	2900-3020

An arrow shows the direction of data flow (<--- or --->) for input and output).

Lines 1560-1700 trap the REMarks and DATA statements and print them immediately to the right of the main logic flow line.

Lines 1710-1840 determine whether the record contains a conditional statement (IF-THEN). It assigns the appropriate flow symbol, depending on whether transfer of control is conditional or unconditional (<--> or ---> respectively), and separates the THEN portion.

Since the execution of any additional statements in a multiple-statement condition depends on whether the IF statement is true, line 5140 formats all statements following the THEN or GOTO as a single entity, rather than as separate lines. This ensures that the resultant flowchart will more accurately represent program logic. Figure 4 shows how Navigator represents various changes in logic flow.

Lines 1860-2010 break the text portion of the line into substrings that will fit inside the symbol. I have taken the meat cleaver approach to dividing text — splitting the line into even lengths regardless of where the division falls. I experimented with word wraparound, but the disadvantage of an increased number of lines of text (thus more paper) seemed to outweigh the gain in readability. Text is padded with blanks (line 1970) and short lines are centered (line 2010). Each line is then formatted into a shape (line 2020).

After formatting a set of lines, NAVIGATOR.3 prints each symbol (lines 3490-3970). Lines 3530-3870 trap loops and add a dotted flow line that links the FOR and NEXT commands for the loop. Multiple flow lines are added for multiple nested loops. Lines 3700-3750 trap the FOR command and initialize the flow line; lines 3790-3870 trap the NEXT command and terminate the flow line. Finally, lines 3910-3950 print the shape. Line 3960 repeats the process for all statements in a multiple statement line, and line 3970 returns to get another record.

Figure 5 shows the symbols generated by Navigator. It differs from ANSI standards in that I/O is represented by a trapezoid, whereas ANSI specifies a parallelogram. The proportions of the symbols have been modified to more easily allow text to be placed inside, and to allow symbols to be generated with standard ASCII characters. I have developed my own standard for placement of line numbers, REMarks, DATA statements, and for marking FOR-NEXT loops.

Main logic flow is designated by an arrow formed from an ASCII 124 (|) with an ASCII 86 (v) in the same column on the line immediately below. Changes in logic flow are represented by arrows formed with a minus sign (-) and the symbols for less than (<) and greater than (>) to show direction.

CUSTOMIZING THE PROGRAM

One of the advantages of writing programs in high level languages is that they can be easily modified to suit new situations. I leave it to the user to decide what additional features are worth adding.

EXAMPLE 1: Modification for an Output Device

```

1121 IF MID$(B$,1,4) = "PR#7" THEN PRNT =
7:SHAPE = 1:FLAG = 6: GOTO 1860
1411 IF MID$(B$,7,5) = "PRINT" AND PRNT =
7 THEN SHAPE = 2
1541 IF MID$(B$,7,5) = "PRINT" AND PRNT =
7 THEN FLAG = 6
2530 ON FLAG GOSUB 2570,2680,2790,2900,2900,
4490
4490 REM -----
4500 REM DEVICE
4510 REM -----
4520 K = INT ((J + 3) / 2) + 1
4530 B$(K - 3) = B$(K - 3) + " " + LEFT$(U
NDERLNS,13): REM 2 SPACES
4540 B$(K - 2) = B$(K - 2) + " " + VS + "
" + VS: REM 2/13 SPACES
4550 B$(K - 1) = B$(K - 1) + " " + VS + "
" + VS: REM 3/13 SPACES
4560 B$(K) = B$(K) + "----" + VS + " DEVICE
" + VS: REM 3/4 SPACES
4570 B$(K + 1) = B$(K + 1) + " " + VS + "
" + VS: REM 5/13 SPACES
4580 B$(K + 2) = B$(K + 2) + " " + VS +
LEFT$(UNDERLNS,13) + VS: REM 6 SPACES
4590 RETURN

```

My own additions include a print font menu in NAVIGATOR.1 to allow me to take advantage of the Epson's panoply of print styles.

Users who have I/O devices in addition to the usual disk and printer (e.g., a modem), will need to customize NAVIGATOR.3 appropriately. Example 1 shows a modification to accommodate an output device in slot 7.

Navigator converts all spaces that appear within quotes to tilde characters (ASCII 126). If you don't care for this feature, you can change the tilde to a different character in line 830 of NAVIGATOR.2 and line 860 of NAVIGATOR.3, or you can delete these lines to keep spaces as spaces.

Most printers keep track of the top of form and will advance to it when they receive ASCII character 12. Navigator performs this function by outputting a series of linefeeds. You may want to substitute a "PRINT CHR\$(12);" statement for lines 1580-1600 in NAVIGATOR.2 and lines 4090-4110 in NAVIGATOR.3.

Navigator, Nibble Maestro, and Lightning Copy are available on diskette for an introductory price of \$19.95 plus \$1.50 shipping/handling (\$2.50 outside the U.S.) from Nibble, 45 Winthrop St., Concord, MA 01742. Introductory price expires 9/31/85.

LISTING 1: NAVIGATOR.1

For ProDOS change 'CATALOG' to 'CAT' in line 400.

```

10 REM *****
20 REM * NAVIGATOR.1 *
30 REM * BY LEE SWOBODA *
40 REM * COPYRIGHT (C) 1985 *
50 REM * BY MICROSPARC, INC *
60 REM * CONCORD, MA. 01742 *
70 REM *****
80 REM

```

```

90 REM
100 BLANK$ = "
      ": REM 39 SPACES
110 Q$ = CHR$ (34)
120 D$ = CHR$ (4)
130 REM -----
140 REM READ FORMAT PARAMETERS
150 REM -----
160 EF = 1: ONERR GOTO 2460
170 PRINT D$"OPEN NAVIGATOR.TXT"
180 PRINT D$"READ NAVIGATOR.TXT"
190 INPUT NAMES,AUTH$,DT$,SELECT,COMMENTS,LM
    ARGIN,RMARGIN,LPAGE,BMARGIN,D1,D2,SLOT
200 PRINT D$"CLOSE"
210 POKE 216,0
220 IF LPAGE = 0 THEN LPAGE = 66
230 IF RMARGIN = 0 THEN RMARGIN = 79
240 IF D1 = 0 THEN D1 = 1
250 IF D2 = 0 THEN D2 = 2
260 IF SLOT = 0 THEN SLOT = 1
270 REM -----
280 REM PROGRAM DATA MENU
290 REM -----
300 HEADERS$ = "PROGRAM DATA"
310 GOSUB 2000
320 FOOTERS$ = ""
330 GOSUB 2170
340 REM ENTER PROGRAM NAME
350 VTAB 10
360 PRINT "PROGRAM NAME (1 OR 2 FOR CATALOG)
      "
370 VTAB 11: CALL - 868: INPUT "": ".NAMES"
380 NF$ = LEFT$(NAMES,1)
390 EF = 4: ONERR GOTO 2460
400 IF NF$ = "1" OR NF$ = "2" THEN HOME : PRINT
    D$"CATALOGD":(NF$ = "2") + 1: GET Z$: PRINT
    Z$: GOTO 300
410 POKE 216,0
420 IF NAMES = "" OR NF$ < "A" OR NF$ > "Z" GOTO
    350
430 REM ENTER AUTHOR'S NAME
440 VTAB 12
450 INPUT "AUTHOR = ",AUTH$
460 IF AUTH$ = "" GOTO 440
470 RFM CONVERT TO LOWER CASE
480 I$ = LEFT$(AUTH$,1)
490 IF LEN (AUTH$) < 2 GOTO 560
500 FOR I = 2 TO LEN (AUTH$)
510 J = ASC ( MID$( AUTH$,I,1))
520 IF MID$( AUTH$,I - 1,1) < > " " AND J >
    64 AND J < 91 THEN J = J + 32
530 I$ = I$ + CHR$ (J)
540 NEXT I
550 AUTH$ = I$
560 REM ENTER DATE
570 VTAB 14
580 CALL - 868
590 INPUT "DATE (9 CHARS):",IN$
600 IF LEN (IN$) > 9 THEN 570
610 PRINT IN$: INPUT "OK? ";YS: IF Y$ < > "
    Y" GOTO 570
620 DT$ = IN$
630 REM -----
640 REM DATA FORMAT MENU
650 REM -----
660 HEADERS$ = "FORMAT DATA"
670 GOSUB 2000
680 FOOTERS$ = "NUMBER TO CHANGE OR N"
690 GOSUB 2170
700 VTAB 10
710 PRINT "1) LEFT MARGIN      = "LMARGIN
720 PRINT "2) RIGHT MARGIN     = "RMARGIN
730 PRINT "3) LINES PER PAGE    = "LPAGE
740 PRINT "4) BOTTOM MARGIN     = "BMARGIN
750 PRINT "5) PRINT REMARKS?   = ";
760 IF COMMENTS = 0 THEN PRINT "N"
770 IF COMMENTS = 1 THEN PRINT "Y"
780 PRINT "6) NAVIGATOR IN DRIVE "D1
790 PRINT "7) PROGRAM IN DRIVE  "D2
800 PRINT "8) PRINTER IN SLOT   "SLOT
810 VTAB 21: HTAB 25
820 PRINT "?";
830 GET A$

```

```

840 PRINT
850 IF A$ = "N" GOTO 1170
860 REM -----
870 REM EDIT DATA
880 REM -----
890 A = VAL (A$)
900 IF A < 1 OR A > 8 GOTO 810
910 PRINT
920 FOOTERS$ = ""
930 GOSUB 2170
940 VTAB 9 + A: HTAB 23
950 CALL - 868
960 INPUT "":AS
970 B = VAL (A$)
980 ON A GOTO 990,1010,1030,1050,1070,1110,1
    130,1150
990 IF B < 1 OR B > 20 GOTO 940
1000 LMARGIN = B: GOTO 660
1010 IF B < 60 OR B > 79 GOTO 940
1020 RMARGIN = B: GOTO 660
1030 IF B < 50 OR B > 88 GOTO 940
1040 LPAGE = B: GOTO 660
1050 IF B < 1 OR B > 10 GOTO 940
1060 BMARGIN = B: GOTO 660
1070 IF A$ < > "Y" AND A$ < > "N" GOTO 940
1080 COMMENTS = 0
1090 IF A$ = "Y" THEN COMMENTS = 1
1100 GOTO 660
1110 IF B < 1 OR B > 2 GOTO 940
1120 D1 = B: GOTO 660
1130 IF B < 1 OR B > 2 GOTO 940
1140 D2 = B: GOTO 660
1150 IF B < 1 OR B > 7 GOTO 940
1160 SLOT = B: GOTO 660
1170 REM -----
1180 REM SELECTION MENU
1190 REM -----
1200 HEADERS$ = "LISTINGS"
1210 GOSUB 2000
1220 FOOTERS$ = "SELECT"
1230 GOSUB 2170
1240 VTAB 10
1250 PRINT "1) LISTING ONLY"
1260 VTAB 12
1270 PRINT "2) FLOW CHART ONLY"
1280 VTAB 14
1290 PRINT "3) LISTING & FLOW CHART"
1300 VTAB 21: HTAB 10
1310 PRINT "?";
1320 GET A$
1330 A = VAL (A$)
1340 IF A < 1 OR A > 3 GOTO 1300
1350 SELECT = A
1360 NR = 2
1370 IF SELECT = 2 THEN NR = 3
1380 PRINT
1390 HOME
1400 GOSUB 1420
1410 GOTO 1670
1420 REM -----
1430 REM SAVE FORMAT DATA
1440 REM -----
1450 EF = 2: ONERR GOTO 2460
1460 PRINT D$"OPEN NAVIGATOR.TXT,D"D1
1470 PRINT D$"WRITE NAVIGATOR.TXT"
1480 PRINT NAMES$
1490 PRINT AUTH$
1500 PRINT DT$
1510 PRINT SELECT
1520 PRINT COMMENTS
1530 PRINT LMARGIN
1540 PRINT RMARGIN
1550 PRINT LPAGE
1560 PRINT BMARGIN
1570 PRINT D1

```



```

1580 PRINT D2
1590 PRINT SLOT
1600 PRINT D$"CLOSE": PRINT D$"OPEN EXX": PRINT
D$"CLOSE EXX": PRINT D$"DELETE EXX"
1610 POKE 216,0
1620 RETURN
1630 EF = 3: ONERR GOTO 2460
1640 PRINT D$"RENAME"NAME$, "NAME$, D"D2
1650 POKE 216,0
1660 RETURN
1670 IF D1 = D2 GOTO 2290
1680 REM -----
1690 REM FORM TEXT FILE
1700 REM -----
1710 HOME
1720 VTAB 10: POKE - 16368,0
1730 PRINT "PLACE NAVIGATOR DISK IN DRIVE #"
D1
1740 PRINT
1750 PRINT "PLACE THE DISK CONTAINING"
1760 PRINT ""NAME$"
1770 PRINT "IN DRIVE #"D2", "
1780 PRINT : PRINT "BE SURE PRINTER IS ON LI
NE,": PRINT : PRINT "AND PRESS ANY KEY";
1790 CALL - 756
1800 HOME
1810 VTAB 12: HTAB 13
1820 PRINT "THE NAVIGATOR"
1830 D$ = CHR$(4)
1840 Q$ = CHR$(34)
1850 GOSUB 1630
1860 PRINT D$"OPEN EXE,D"D1
1870 PRINT D$"WRITE EXE"
1880 PRINT "NEW"
1890 PRINT "LOAD "NAME$,D"D2
1900 F$ = "RUN NAVIGATOR." + STR$(NR) + ",D
" + STR$(D1)
1910 PRINT "0 D$=CHR$(4):?D$?Q$"OPENEXX,D"(D
1)Q$":?D$?Q$"WRITEEXX"Q$":LIST1-:?D$?Q$
"CLOSE"Q$":?D$?Q$F$Q$
1920 PRINT "RUN"
1930 PRINT D$"CLOSE"
1940 PRINT D$"EXEC EXE,D"D1
1950 END
1960 REM =====
1970 REM SUBROUTINES
1980 REM =====
1990 REM
2000 REM -----
2010 REM MENU HEADER
2020 REM -----
2030 HOME
2040 PRINT
2050 PRINT TAB(12)"THE NAVIGATOR"
2060 PRINT
2070 PRINT TAB(9)"BASIC LISTING UTILITY"
2080 PRINT TAB(8)"AND FLOW CHART FORMATTER"
"
2090 PRINT
2100 INVERSE
2110 PRINT BLANK$
2120 VTAB 7
2130 PRINT " "HEADERS$
2140 NORMAL
2150 VTAB 10
2160 RETURN
2170 REM -----
2180 REM MENU FOOTER
2190 REM -----
2200 VTAB 21
2210 INVERSE
2220 PRINT BLANK$
2230 VTAB 21
2240 PRINT " "FOOTER$ "
2250 NORMAL
2260 VTAB 23
2270 PRINT TAB(4)"COPYRIGHT 1985 MICROSPAR
C. INC."
2280 RETURN

```

```

2290 REM -----
2300 REM ONE-DRIVE
2310 REM -----
2320 HOME
2330 VTAB 12: POKE - 16368,0
2340 PRINT "BE SURE PRINTER IS ON LINE.": PRINT
: PRINT "AND PRESS ANY KEY";
2350 GET AS: PRINT
2360 GOSUB 1630: REM VERIFY EXISTENCE OF PR
OGRAM
2370 PRINT D$"OPEN EX1,D"D1
2380 PRINT D$"WRITE EX1"
2390 PRINT "LOAD"NAME$
2400 F$ = "RUN NAVIGATOR." + STR$(NR) + ",D
" + STR$(D1)
2410 PRINT "0HOME:D$=CHR$(4):?D$?Q$"OPENEXX,
D"(D1)Q$":?D$?Q$"WRITEEXX"Q$":LIST1-:?D$
"Q$"CLOSE"Q$":?D$?Q$F$Q$
2420 PRINT "RUN"
2430 PRINT D$"CLOSE"
2440 PRINT D$"EXEC EX1,D"D1
2450 END
2460 E = PEEK(222):EL = PEEK(218) + 256 *
PEEK(219): POKE 216,0
2470 IF EF = 1 AND E = 5 THEN PRINT D$"CLOS
E": PRINT D$"DELETE NAVIGATOR.TXT": GOTO
220
2480 IF EF = 3 AND E = 10 THEN ON (D1 = D2)
+ 1 GOTO 2320,1710
2490 IF EF = 3 THEN EF = 3 + 2 * (D1 = D2)
2500 IF E = 4 THEN HOME : VTAB 12: PRINT "D
ISK WRITE PROTECTED": GOTO 2530
2510 IF E = 6 THEN HOME : VTAB 12: PRINT "F
ILE NOT FOUND":EF = EF + (EF = 3): ON (D
1 = D2) + 1 GOTO 2530,2540
2520 HOME : VTAB 12: PRINT "ERROR #":E IN L
INE "EL".
2530 PRINT : PRINT "PRESS ANY KEY TO CONTINU
E": GET Z$: PRINT : ON EF GOTO 160,1400
,1850,300,2360
2540 PRINT : PRINT "YOUR PROGRAM AND NAVIGAT
OR MUST BE": PRINT " ON THE SAME DISK!":
END

```

END OF LISTING 1

KEY PERFECT 4.0
 RUN ON
 NAVIGATOR 1

CODE	LINE#	LINE#
563C	10	100
7F4F	110	200
513E	210	300
5D8A	310	400
4FD0	410	500
4BD9	510	600
4E77	610	700
7243	710	800
3167	810	900
4135	910	1000
4AB7	1010	1100
4D80	1110	1200
3D5B	1210	1300
275B	1310	1400
43F9	1410	1500
411C	1510	1600
42F8	1610	1700
5342	1710	1800
4BFA	1810	1900
63EC	1910	2000
3C94	2010	2100
2A30	2110	2200
2ED5	2210	2300
632D	2310	2400
9AB1	2410	2500
6816	2510	2540

PROGRAM CHECK IS : 1251

LISTING 2: NAVIGATOR.2

For ProDOS, delete lines 1370-1380.

```

10 REM *****
20 REM * NAVIGATOR.2 *
30 REM * LISTING FORMATTER *
40 REM * BY LEE SWOBODA *
50 REM * COPYRIGHT (C) 1985 *
60 REM * BY MICROSPARC, INC *
70 REM * CONCORD, MA 01742 *
80 REM *****
90 REM
100 REM
110 DS$ = CHR$(4)
120 HOME
130 EF = 1: ONERR GOTO 1210
140 PRINT DS"OPEN NAVIGATOR.TXT"
150 PRINT DS"READ NAVIGATOR.TXT"
160 INPUT NAMES, AUTH$, DT$, SELECT, COMMENTS, LM
    ARGIN, RMARGIN, LPAGE, BMARGIN, D1, D2, SLOT
170 PRINT DS"CLOSE"
180 POKE 216,0
190 REM -----
200 REM DEFAULT VALUES
210 REM -----
220 SLOT$ = "PR#" + STR$(SLOT)
230 PAGE = 0: REM PAGE NUMBER
240 LNLGTH = RMARGIN - LMARGIN: REM LINE L
    ENGTH
250 HEADERS(1) = NAMES$
260 FOR I = 1 TO LNLGTH - LEN (NAMES) - 9
270 HEADERS(1) = HEADERS(1) + " "
280 NEXT I
290 HEADERS(1) = HEADERS(1) + "PAGE "
300 HEADERS(2) = AUTH$
310 FOR I = 1 TO LNLGTH - LEN (AUTH$) - 9
320 HEADERS(2) = HEADERS(2) + " "
330 NEXT I
340 HEADERS(2) = HEADERS(2) + DT$
350 FOR I = 1 TO LNLGTH
360 HEADERS(3) = HEADERS(3) + "-"
370 NEXT I
380 LPAGE = LPAGE - BMARGIN
390 DIM AS(40)
400 PRINT D$SLOT$: PRINT CHR$(9)"80N"
410 GOSUB 1470
420 EF = 2: ONERR GOTO 1210
430 PRINT DS"OPEN EXX,D1D1"
440 PRINT DS"READ EXX"
450 REM -----
460 REM BUILD STRING
470 REM -----
480 BS$ = ""
490 GET AS
500 IF AS$ = CHR$(13) GOTO 530
510 BS$ = BS$ + AS$: IF LEN (BS$) = 248 THEN BS$ =
    "<<< LISTED LINE TOO LONG FOR NAVIGATOR
    >>>": GOSUB 1650: GOTO 610
520 GOTO 490
530 B = VAL (BS)
540 IF B = 0 GOTO 490
550 AS$ = STR$(B)
560 B = LEN (AS$)
570 AS$ = LEFT$(AS$ + " ",6)
580 BS$ = RIGHT$(BS$, LEN (BS$) - B - 1)
590 IF LEFT$(BS$,1) = " " THEN BS$ = RIGHT$
    (BS$, LEN (BS$) - 1): GOTO 590
600 BS$ = AS$ + BS$
610 REM -----
620 REM FORMAT STRING
630 REM -----
640 J = 1: REM STRING SUBSCRIPT
650 L = 0: REM SUBSTRING LENGTH
660 K = LEN (BS$)
670 IF MID$(BS$,7,3) = "REM" OR MID$(BS$,7
    ,4) = "DATA" THEN RFLAG = 1
680 AS$(1) = ""
690 BS$ = BS$ + " "
700 REM FORMAT LINES
710 QFLAG = 0
720 FOR I = 1 TO K
730 I = I + 1

```

```

740 AS$ = MID$(BS$,I,1)
750 IF AS$ < " " THEN AS$ = "^" + CHR$(ASC
    (AS$) + 224): REM CONTROL CHARACTERS
760 IF AS$ = CHR$(34) THEN QFLAG = NOT QFL
    AG: REM STRINGS
770 IF QFLAG AND AS$ = " " THEN AS$ = CHR$(1
    26): REM CHANGE SPACE TO TILDE
780 IF OFLAG OR RFLAG GOTO 910
790 IF MID$(AS$(J),7,2) = "IF" AND MID$(B
    $,I,4) = "GOTO" THEN J = J + 1:AS$(J) = "
    ":L = 9: GOTO 860
800 IF MID$(AS$(J),7,2) = "IF" AND MID$(B
    $,I,4) = "THEN" THEN J = J + 1:AS$(J) = "
    ":L = 9: GOTO 860
810 IF MID$(BS$,7,2) = "IF" AND MID$(AS$(J
    ),5,1) < > ":" AND MID$(BS$,I,4) = "GO
    TO" THEN J = J + 1:AS$(J) = " ":L =
    9: GOTO 830
820 IF MID$(BS$,7,2) = "IF" AND MID$(AS$(J
    ),5,1) < > ":" AND MID$(BS$,I,4) = "TH
    EN" THEN J = J + 1:AS$(J) = " ":L =
    9: GOTO 830
830 IF AS$ < > ":" GOTO 860
840 IF MID$(BS$,I - 3,4) = "MEM:" GOTO 860
850 J = J + 1:AS$(J) = " ":AS$ = "":L = 6:
    REM NEWLINE
860 IF I < 7 GOTO 910
870 IF AS$ = " " AND RIGHTS (AS$(J),1) = " " GOTO
    920
880 IF AS$ = " (" AND RIGHTS (AS$(J),1) = "(" GOTO
    920
890 IF AS$ = " " AND MID$(BS$,I + 1,1) = "("
    GOTO 920
900 IF AS$ = "," THEN AS$ = " ,"
910 AS$(J) = AS$(J) + AS$
920 NEXT I
930 REM -----
940 REM PRINT LINES
950 REM -----
960 GOSUB 1540: IF LCOUNT + J > (LPAGE) THEN
    GOSUB 1410: REM NEW PAGE?
970 FOR I = 1 TO J
980 IF MID$(AS$(I),9,4) = "THEN" THEN TFLAG
    = 1
990 IF COMMENTS = 0 AND MID$(AS$(I),7,3) =
    "REM" GOTO 1170: REM SKIP REMARKS?
1000 IF MID$(BS$,7,3) = "REM" AND R1 = 0 THEN
    PRINT :R1 = 1:LCOUNT = LCOUNT + 1: REM
    SKIP LINE BEFORE REMARK
1010 IF MID$(BS$,7,3) < > "REM" AND R1 = 1
    THEN PRINT :R1 = 0:LCOUNT = LCOUNT + 1
    : REM SKIP LINE AFTER REMARK(S)
1020 LCOUNT = LCOUNT + 1
1030 IF MID$(AS$(I),7,3) = "FOR" THEN INDEN
    T = INDENT + 5: LNLGTH = LNLGTH - 5: REM
    INDENT FOR...NEXT LOOPS
1040 REM PRINT THE LINE
1050 PRINT CHR$(1) SPC( LMARGIN):
1060 IF MID$(AS$(I),5,1) < > ":" GOTO 1090
1070 IF TFLAG THEN PRINT " "
1080 PRINT SPC( INDENT)AS$(I): GOTO 1120
1090 PRINT LEFT$(AS$(I),6)
1100 IF MID$(AS$(I),7,3) < > "REM" AND MID$
    (AS$(I),7,4) < > "DATA" THEN PRINT SPC(
    INDENT);
1110 PRINT MID$(AS$(I),7, LEN (AS$(I)))
1120 IF MID$(AS$(I),7,4) < > "NEXT" GOTO 1
    170
1130 IF INDENT > 4 THEN LNLGTH = LNLGTH + 5:
    INDENT = INDENT - 5: REM UNINDENT F
    OR...NEXT LOOP
1140 FOR Z = 7 TO 80
1150 IF MID$(AS$(I),Z,1) = " " THEN IF IND
    ENT > 4 THEN LNLGTH = LNLGTH + 5: INDENT =
    INDENT - 5
1160 NEXT Z
1170 NEXT I
1180 RFLAG = 0: REM CANCEL "REM" FLAG
1190 TFLAG = 0: REM CANCEL "THEN" FLAG
1200 GOTO 480
1210 REM ==
1220 REM END

```

```

1230 REM ===
1240 E = PEEK (222):EL = PEEK (218) + 256 +
      PEEK (219): POKE 216,0
1250 IF EF = 2 AND E = 5 GOTO 1300
1260 HOME : VTAB 12: PRINT "ERROR #"E" IN LI
      NE "EL":
1270 PRINT : PRINT "PRESS <ESC> TO QUIT,": PRINT
      "OR ANY OTHER KEY TO TRY AGAIN."
1280 GET Z$: PRINT : IF Z$ = CHR$(27) THEN
      PRINT D$"CLOSE": END
1290 ON EF GOTO 130,420,1390
1300 FOR I = 1 TO LPAGE + BMARGIN - LCOUNT
1310 PRINT
1320 NEXT I
1330 IF SELECT = 3 THEN PRINT D$"RUN NAVIGA
      TOR 3.D"D1
1340 PRINT D$"PR#0"
1350 PRINT D$"CLOSE"
1360 PRINT D$"DELETE EXX"
1370 PRINT D$"OPEN EXE": PRINT D$"CLOSE EXE"
      : PRINT D$"DELETE EXE"
1380 PRINT D$"OPEN EX1": PRINT D$"CLOSE EX1"
      : PRINT D$"DELETE EX1"
1390 POKE 216,0
1400 END
1410 REM =====
1420 REM PRINT HEADER
1430 REM =====
1440 FOR P = 1 TO LPAGE + BMARGIN - LCOUNT
1450 PRINT
1460 NEXT P
1470 PAGE = PAGE + 1
1480 PRINT SPC( LMARGIN)HEADERS$(1)PAGE
1490 PRINT SPC( LMARGIN)HEADERS$(2)
1500 PRINT SPC( LMARGIN)HEADERS$(3)
1510 PRINT
1520 LCOUNT = 4
1530 RETURN
1540 SI = LNLGTH - 10:I = 1
1550 T$ = A$(I)
1560 IF LEN (T$) < = SI GOTO 1630
1570 IF LEFT$( T$,6) = " " GOTO 1590
1580 SI = SI - 5 + ( MID$( T$,7,3) = "FOR" ) +
      5 * ( MID$( T$,7,4) = "NEXT" ): IF SI > L
      NLGTH - 10 THEN SI = LNLGTH - 10
1590 K = J:L = SI - 1
1600 ALS = MID$( A$(I),L,1): IF ALS < > " "
      AND ALS < > "$" AND ALS < > ",," AND A
      L$ < > CHR$(126) THEN L = L - 1: GOTO
      1600
1610 A$(K + 1) = A$(K):K = K - 1: IF K > = I
      GOTO 1610
1620 A$(I) = LEFT$( T$,L):A$(I + 1) = "
      " + MID$( T$,L + 1):J = J + 1
1630 I = I + 1: IF I < = J GOTO 1550
1640 RETURN
1650 GET A$: IF A$ < > CHR$(13) GOTO 16
      50
1660 RETURN

```

END OF LISTING 2

KEY PERFECT 4.0
RUN ON
NAVIGATOR.2

CODE	LINE# - LINE#
51DE	10 - 100
6A78	110 - 200
6E38	210 - 300
570E	310 - 400
3615	410 - 500
616F	510 - 600
59C2	610 - 700
98E0	710 - 800
A77E	810 - 900
8799	910 - 1000
9CB9	1010 - 1100
8368	1110 - 1200
74F7	1210 - 1300
5910	1310 - 1400
501A	1410 - 1500
726F	1510 - 1600
3B5F	1610 - 1660

PROGRAM CHECK IS : 0F9C

LISTING 3: NAVIGATOR.3

For ProDOS, delete lines 4140-4150.

```

10 REM *****
20 REM * NAVIGATOR.3 *
30 REM * FLOW CHART PRINTER *
40 REM * BY LEE SWOBODA *
50 REM * COPYRIGHT (C) 1985 *
60 REM * BY MICROSPARC, INC *
70 REM * CONCORD, MA. 01742 *
80 REM *****
90 REM
100 REM
110 D$ = CHR$(4)
120 HOME
130 EF = 1: ONERR GOTO 4430
140 PRINT D$"OPEN NAVIGATOR.TXT"
150 PRINT D$"READ NAVIGATOR.TXT"
160 INPUT NAME$,AUTH$,DT$,SELECT,COMMENTS,LM
      ARGIN,RMARGIN,LPAGE,BMARGIN,D1,D2,SLOT
170 PRINT D$"CLOSE"
180 REM -----
190 REM DEFAULT VALUES
200 REM -----
210 DIM A$(40),B$(40),C$(40)
220 LCOUNT = 9: REM LINE COUNT
230 PAGE = 0: REM PAGE NUMBER
240 LNLGTH = RMARGIN - LMARGIN: REM LINE LENG
      TH
250 LPAGE = LPAGE - BMARGIN
260 LOOP = 0: REM LOOP FLAG
270 PRNT = 0: REM OUTPUT FLAG -> VIDEO SCREEN
280 INPT = 0: REM INPUT FLAG <- KEYBOARD
290 SLOT$ = "PR#" + STR$(SLOT)
300 US = CHR$(95): REM UNDERSCORE
310 S$ = CHR$(92): REM BACKSLASH
320 VS = CHR$(124): REM VERTICAL LINE
330 FOR I = 1 TO 30
340 BLANK$ = BLANK$ + CHR$(32)
350 NEXT I
360 FOR I = 1 TO 42
370 UNDERLNS = UNDERLNS + US
380 NEXT I
390 FLOW$ = V$
400 K = 1: REM SUB-LINE SUBSCRIPT
410 B$(1) = BLANK$ + FLOW$
420 REM -----
430 REM START
440 REM -----
450 IF SELECT = 2 THEN PRINT D$SLOT$: PRINT
      CHR$(9)"80N"
460 GOSUB 4340: REM PRINT HEADER
470 PRINT LEFT$( BLANK$,22) LEFT$( UNDERLNS
      ,17)
480 PRINT LEFT$( BLANK$,21)VS LEFT$( BLANK$
      ,17)VS
490 PRINT LEFT$( BLANK$,21)VS" START
      "VS
500 PRINT LEFT$( BLANK$,21)VS LEFT$( UNDERL
      N$,17)VS
510 PRINT BLANK$FLOW$
520 EF = 2: ONERR GOTO 3980
530 PRINT D$"OPEN EXX.D"D1
540 PRINT D$"READ EXX"
550 REM -----
560 REM BUILD STRING
570 REM -----
580 B$ = ""
590 GET A$
600 IF A$ = CHR$(13) THEN QFLAG = 0: GOTO
      630
610 B$ = B$ + A$: IF LEN (B$) > 249 THEN B$(
      1) = "<<< LISTED LINE TOO LONG FOR NAVIG
      ATOR >>>": GOSUB 4470:K = 1:EF = 2: GOTO
      3570
620 GOTO 590
630 PRINT CHR$(1):
640 B = VAL (B$)
650 IF B = 0 GOTO 590
660 A$ = STR$(B)
670 B = LEN (A$)

```

LISTING 3: NAVIGATOR.3 (continued)

```

680 A$ = LEFT$(A$ + " ", 6)
690 B$ = RIGHT$(B$, LEN(B$) - B - 1)
700 IF LEFT$(B$,1) = " " THEN B$ = RIGHT$(
    (B$, LEN(B$) - 1)): GOTO 700
710 B$ = A$ + B$
720 IF MID$(B$,7,3) = "REM" OR MID$(B$,7
    ,4) = "DATA" GOTO 1600
730 REM -----
740 REM FORMAT LINES
750 REM -----
760 JJ = 1: REM STRING SUBSCRIPT
770 IF DOS = 1 THEN DOS = 0
780 KK = LEN(B$)
790 C$(1) = ""
800 FOR MM = 1 TO KK
810 A$ = MID$(B$,MM,1)
820 IF A$ = CHR$(4) THEN DOS = 1
830 IF MID$(A$(JJ),7,8) = "PRINT D$" OR MID$(
    (A$(JJ),7,14) = "PRINT CHR$(4)" THEN DO
    S = 1
840 IF A$ < " " THEN A$ = "^" + CHR$(ASC
    (A$) + 224): REM CONTROL CHARACTERS
850 IF A$ = CHR$(34) THEN QFLAG = NOT QFL
    AG: REM STRINGS
860 IF QFLAG AND A$ = " " THEN A$ = CHR$(1
    26): REM CHANGE SPACE TO TILDE
870 IF QFLAG GOTO 980: REM DO NOT FORMAT STR
    INGS
880 IF A$ < > ":" GOTO 910
890 IF MID$(B$,MM - 3,4) = "MEM:" GOTO 910
900 IF MID$(B$,7,2) < > "IF" AND MID$(C
    $(JJ),7,2) < > "IF" THEN JJ = JJ + 1: C$(
    (JJ)) = " " + A$: REM NEW LINE
910 IF MM < 7 GOTO 980
920 IF A$ = " " AND RIGHTS$(C$(JJ),1) = " "
    GOTO 990
930 IF A$ = " " AND RIGHTS$(C$(JJ),1) = "("
    GOTO 990
940 IF A$ = " " AND MID$(B$,II + 1,1) = "("
    GOTO 990
950 IF A$ = "." THEN A$ = ". "
960 IF MID$(B$,MM,8) = "PRINT D$" OR MID$(
    (B$,MM,15) = "PRINT CHR$(4)" THEN DOS =
    1
970 IF A$ = ";" THEN A$ = "; "
980 C$(JJ) = C$(JJ) + A$
990 NEXT MM
1000 REM ---
1010 REM DOS
1020 REM ---
1030 FOR II = 1 TO JJ
1040 B$ = C$(II)
1050 REM
1060 REM SCAN FOR CHANGES IN CONTROL
1070 REM
1080 IF DOS = 0 GOTO 1380
1090 L1 = LEN(B$)
1100 FOR I = 15 TO L1
1110 IF MID$(B$,I,4) = "PR#0" THEN PRNT =
    0: SHAPE = 2: FLAG = 1: GOTO 1860
1120 IF MID$(B$,I,4) = "SLOT$" THEN PRNT = 1
    : SHAPE = 2: FLAG = 3: GOTO 1860
1130 IF MID$(B$,I,4) = "OPEN" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1140 IF MID$(B$,I,6) = "VERIFY" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1150 IF MID$(B$,I,6) = "APPEND" THEN SHAPE =
    2: FLAG = 4: DOS = 2: GOTO 1860
1160 IF MID$(B$,I,4) = "EXEC" THEN SHAPE =
    2: FLAG = 5: GOTO 1860
1170 IF MID$(B$,I,6) = "DELETE" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1180 IF MID$(B$,I,5) = "CLOSE" AND INPT =
    6 THEN INPT = 0: SHAPE = 2: FLAG = 4: DOS =
    1: GOTO 1860
1190 IF MID$(B$,I,5) = "CLOSE" AND PRNT =
    6 THEN PRNT = 0: SHAPE = 2: FLAG = 4: DOS =
    1: GOTO 1860
1200 IF MID$(B$,I,5) = "WRITE" THEN PRNT =
    6: SHAPE = 2: FLAG = 4: DOS = 2: GOTO 1860
1210 IF MID$(B$,I,4) = "READ" THEN INPT =
    6: SHAPE = 2: FLAG = 5: DOS = 2: GOTO 1860
1220 IF MID$(B$,I,5) = "BLOOD" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1230 IF MID$(B$,I,4) = "BRUN" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1240 IF MID$(B$,I,3) = "CAT" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1250 IF MID$(B$,I,7) = "CATALOG" THEN SHAP
    E = 2: FLAG = 4: GOTO 1860
1260 IF MID$(B$,I,5) = "CHAIN" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1270 IF MID$(B$,I,6) = "CREATE" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1280 IF MID$(B$,I,5) = "FLUSH" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1290 IF MID$(B$,I,3) = "FRE" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1300 IF MID$(B$,I,4) = "LOCK" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1310 IF MID$(B$,I,6) = "PREFIX" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1320 IF MID$(B$,I,6) = "RENAME" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1330 IF MID$(B$,I,3) = "RUN" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1340 IF MID$(B$,I,4) = "SAVE" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1350 IF MID$(B$,I,5) = "STORE" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1360 IF MID$(B$,I,6) = "UNLOCK" THEN SHAPE =
    2: FLAG = 4: GOTO 1860
1370 NEXT I
1380 REM -----
1390 REM DETERMINE SHAPE
1400 REM -----
1410 SHAPE = 1
1420 IF MID$(B$,7,5) = "PRINT" THEN SHAPE =
    2
1430 IF MID$(B$,7,5) = "INPUT" THEN SHAPE =
    2
1440 IF MID$(B$,7,3) = "GET" THEN SHAPE =
    2
1450 IF MID$(B$,7,2) = "IF" THEN SHAPE = 3
1460 REM -----
1470 REM DETERMINE FLAG
1480 REM -----
1490 IF SHAPE < > 2 GOTO 1560
1500 FLAG = 1
1510 IF MID$(B$,7,5) = "PRINT" AND PRNT =
    0 THEN FLAG = 1
1520 IF (MID$(B$,7,5) = "INPUT" OR MID$(
    (B$,7,3) = "GET") AND INPT = 0 THEN FLAG =
    2
1530 IF MID$(B$,7,5) = "PRINT" AND PRNT =
    1 THEN FLAG = 3
1540 IF MID$(B$,7,5) = "PRINT" AND PRNT =
    6 THEN FLAG = 4
1550 IF (MID$(B$,7,5) = "INPUT" OR MID$(
    (B$,7,3) = "GET") AND INPT = 6 THEN FLAG =
    5
1560 REM -----
1570 REM HANDLE REMARKS AND DATA STATEMENTS
1580 REM -----
1590 IF MID$(B$,7,3) < > "REM" AND MID$(
    (B$,7,4) < > "DATA" GOTO 1710
1600 IF COMMENTS = 0 AND MID$(B$,7,3) = "R
    EM" GOTO 580
1610 B$ = B$ + " "
1620 L2 = 1
1630 L1 = LNLGTH - 25
1640 IF LCOUNT + 3 > (LPAGE) THEN GOSUB 418
    0
1650 IF MID$(B$,L1,1) < > " " AND MID$(
    (B$,L1,1) < > " ", THEN L1 = L1 - 1: IF L
    1 > LNLGTH - 35 GOTO 1650
1660 PRINT LEFT$(B$(K),8) LEFT$(B$,6) MID$(
    (B$(K),15,31)) " MID$(B$,L2 + 5,L1 - 5)
    : LCOUNT = LCOUNT + 1
1670 B$ = " " + MID$(B$,L1 + 1, LEN(B
    $))

```

LISTING 3: NAVIGATOR.3 (continued)

```

1680 IF LEN (B$) > LNLGTH - 25 GOTO 1620
1690 PRINT LEFT$ (B$(K),31) MID$ (B$,L2 + 5
,L1 - 5):LCOUNT = LCOUNT + 1
1700 GOTO 580
1710 REM -----
1720 REM HANDLE CONDITIONALS
1730 REM -----
1740 IF MID$ (B$,7,2) < > "IF" GOTO 1860
1750 L1 = LEN (B$)
1760 ARROWS = "---->"
1770 FLOW = 0
1780 FOR I = 10 TO L1
1790 IF MID$ (B$,I,4) = "THEN" THEN FLOW =
1:L2 = I: IF MID$ (B$,I + 5,5) = "GOSUB"
" THEN FLOW = 3:ARROWS = "---->"
1800 IF MID$ (B$,I,4) = "GOTO" THEN FLOW =
2:L2 = I
1810 IF FLOW > 0 THEN I = L1 + 1
1820 NEXT I
1830 ARROWS = ARROWS + MID$ (B$,L2 - 1,L1)
1840 B$ = LEFT$ (B$,L2 - 2)
1850 REM #11000
1860 REM -----
1870 REM FORM TEXT
1880 REM -----
1890 L2 = 29
1900 IF SHAPE = 2 THEN L2 = 23
1910 IF SHAPE = 3 THEN L2 = 27
1920 LNUM$ = LEFT$ (B$,5)
1930 L1 = LEN (B$)
1940 J = 1
1950 FOR I = 7 TO L1 STEP L2
1960 A$(J) = MID$ (B$,I,L2 + 1)
1970 A$(J) = LEFT$ (A$(J) + BLANK$,L2)
1980 J = J + 1
1990 NEXT I
2000 J = J - 1
2010 IF L1 < L2 + 5 THEN A$(1) = LEFT$ ( LEFT$
(BLANK$,L2 - L1 + 6) / 2) + A$(1) + BLA
NK$,L2)
2020 ON SHAPE GOTO 2030,2310,3030
2030 REM -----
2040 REM SHAPE 1
2050 REM -----
2060 B$(1) = LEFT$ (BIANK$,14) + " " + LEFT$
(UNDERLN$,15) + "V" + LEFT$ (UNDERLN$,1
5) + " "
2070 IF FLOW$ = " " THEN B$(1) = LEFT$ (B$(
1),30) + US$ + MID$ (B$(1),32, LEN (B$(1
))) : FLOW$ = VS
2080 B$(2) = LEFT$ (BLANK$,8) + LNUM$ + " " +
VS$ + LEFT$ (BLANK$ + BLANK$,31) + VS$
2090 FOR I = 1 TO J
2100 B$(I + 2) = LEFT$ (BLANK$,14) + VS$ + "
" + A$(I) + " " + VS$
2110 NEXT I
2120 B$(J + 3) = LEFT$ (BLANK$,14) + VS$ + LEFT$
(UNDERLN$,31) + VS$
2130 L = INT (J / 2) + 3
2140 IF MID$ (B$,7,4) = "GOTO" THEN B$(L) =
B$(L) + "---->" + MID$ (B$,11, LEN (B$))
: FLOW$ = " "
2150 IF MID$ (B$,7,5) = "GOSUB" THEN B$(L) =
B$(L) + "---->" + MID$ (B$,12, LEN (B$))
2160 IF MID$ (B$,7,2) < > "ON" GOTO 2270
2170 FOR AA = 10 TO LEN (B$)
2180 IF MID$ (B$,AA,4) = "GOTO" GOTO 2210
2190 NEXT AA
2200 IF AA = > LEN (B$) GOTO 2220
2210 B$(L) = B$(L) + "---->" + MID$ (B$,AA +
4, LEN (B$)) : FLOW$ = " " : GOTO 2260
2220 FOR AA = 10 TO LEN (B$)
2230 IF MID$ (B$,AA,5) = "GOSUB" GOTO 2250
2240 NEXT AA
2250 B$(L) = B$(L) + "---->" + MID$ (B$,AA +
5, LEN (B$)) : FLOW$ = "V"
2260 IF LEN (B$(L)) > LNLGTH + (LMARGIN) THEN
B$(L + 1) = B$(L + 1) + " " + MID$
(B$(L),LNLGTH + LMARGIN, LEN (B$(L))) : B$
(L) = LEFT$ (B$(L),LNLGTH + LMARGIN)
2270 IF MID$ (B$,7,6) = "RETURN" OR MID$ (
B$,7,3) = "END" THEN FLOW$ = " "
2280 B$(J + 4) = BLANK$ + FLOW$
2290 K = J + 4
2300 GOTO 3490
2310 REM -----
2320 REM SHAPE 2
2330 REM -----
2340 FOR I = 1 TO J
2350 D$(I) = A$(I)
2360 NEXT I
2370 IF J > 2 GOTO 2410
2380 J = J + 1
2390 D$(J) = LEFT$ (BLANK$,L2)
2400 GOTO 2370
2410 L1 = 19
2420 B$(J + 3) = LEFT$ (BLANK$,L1 - 1) + S$ +
LEFT$ (UNDERLN$,23) + "/"
2430 B1$ = " "
2440 B1 = 2
2450 FOR K = J TO 1 STEP - 1
2460 B$(K + 2) = LEFT$ (BLANK$,L1 - B1) + S$
+ B1$ + D$(K) + B1$ + "/"
2470 B1$ = B1$ + " "
2480 B1 = B1 + 1
2490 NEXT K
2500 B$(2) = LEFT$ (BLANK$,L1 - 6 - B1) + LN
UM$ + " " + S$ + LEFT$ (BLANK$ + BLANK$
,L2 + 2 * B1 - 2) + "/"
2510 B$(1) = LEFT$ (BLANK$,L1 - B1) + LEFT$
(UNDERLN$,L2 / 2 + B1) + "V" + LEFT$ (U
NDERLN$,L2 / 2 + B1) + " "
2520 IF FLOW$ = " " THEN B$(1) = LEFT$ (B$(
1),30) + US$ + MID$ (B$(1),32, LEN (B$(1
))) : FLOW$ = VS
2530 ON FLAG GOSUB 2570,2680,2790,2900,2900
2540 B$(J + 4) = BLANK$ + FLOW$
2550 K = J + 4
2560 GOTO 3490
2570 REM -----
2580 REM SCREEN
2590 REM -----
2600 K = INT ((J + 3) / 2) + 1
2610 B$(K - 3) = B$(K - 3) + " " + LEFT$
(UNDERLN$,10)
2620 B$(K - 2) = B$(K - 2) + " " /
" + S$: REM 4/10 SPACES
2630 B$(K - 1) = B$(K - 1) + " " /
" + VS$: REM 4/12 SPACES
2640 B$(K) = B$(K) + "---->(" VIDEO " + V
$
2650 B$(K + 1) = B$(K + 1) + " " + S$ +
LEFT$ (BLANK$,12) + VS$
2660 B$(K + 2) = B$(K + 2) + LEFT$ (BLANK$,8
) + S$ + LEFT$ (UNDERLN$,10) + "/"
2670 RETURN
2680 REM -----
2690 REM KEYBOARD
2700 REM -----
2710 K = INT ((J + 3) / 2) + 1
2720 B$(K - 3) = B$(K - 3) + " " + LEFT$
(UNDERLN$,13)
2730 B$(K - 2) = B$(K - 2) + " " /
"/": REM 4/12/ SPACES
2740 B$(K - 1) = B$(K - 1) + " " /
"/": REM 4/12/ SPACES
2750 B$(K) = B$(K) + "----/ KEYBOARD /"
2760 B$(K + 1) = B$(K + 1) + " " /
"/": REM 4/12/ SPACES
2770 B$(K + 2) = B$(K + 2) + " " /" + LEFT$
(UNDERLN$,12) + "/"
2780 RETURN
2790 REM -----
2800 REM PRINTER
2810 REM -----
2820 K = INT ((J + 3) / 2) + 1
2830 B$(K - 3) = B$(K - 3) + " " + LEFT$ (U
NDERLN$,13)
2840 B$(K - 2) = B$(K - 2) + " " + VS$ + LEFT$
(BLANK$,13) + VS$
2850 B$(K - 1) = B$(K - 1) + " " + VS$ + "
" + VS$
2860 B$(K) = B$(K) + "---->" + VS$ + LEFT$ (BL
ANK$,13) + VS$

```

```
2870 BS(K + 1) = BS(K + 1) + " " + VS + "
" + LEFT$(UNDERLNS,6) + VS
2880 BS(K + 2) = BS(K + 2) + " " + VS +
LEFT$(UNDERLNS,6) + "/"
2890 RETURN
2900 REM ----
2910 REM DISK
2920 REM ----
2930 ARROWS = "----"
2940 IF FLAG = 5 THEN ARROWS$ = "<---"
2950 K = INT((J + 3) / 2) + 1
2960 BS(K - 3) = BS(K - 3) + " " + LEFT$(
UNDERLNS,14)
2970 BS(K - 2) = BS(K - 2) + " " / " + LEFT$(
(BLANKS,14) + S$
2980 BS(K - 1) = BS(K - 1) + " " + VS + "
DISK " + VS
2990 BS(K) = BS(K) + ARROWS$ + VS + S$ + LEFT$(
(UNDERLNS,14) + "/" + VS
3000 BS(K + 1) = BS(K + 1) + " " + VS + LEFT$(
(BLANKS,16) + VS
3010 BS(K + 2) = BS(K + 2) + " " + S$ +
LEFT$(UNDERLNS,14) + "/"
3020 RETURN
3030 REM -----
3040 REM SHAPE 3
3050 REM -----
3060 L = INT(J / 2)
3070 IF J / 2 = L THEN J = J + 1:AS(J) = LEFT$(
(BLANKS,L2)
3080 L1 = 17
3090 BS(1) = LEFT$(BLANKS,L1) + LEFT$(UND
ERLNS,13) + "V" + LEFT$(UNDERLNS,13)
3100 IF FLOWS$ = " " THEN BS(1) = LEFT$(BS(
1),30) + US$ + MIDS$(BS(1),32, LEN(BS(1
))) : FLOWS$ = VS
3110 BS(2) = LEFT$(BLANKS,L1 - 9) + LNUMS$ +
" " / " + LEFT$(BLANKS,27) + S$
3120 IF L = 0 THEN BS(3) = LEFT$(BLANKS,L1
- 2) + "( " + AS(1) + " )": GOTO 3280
3130 B1$ = " "
3140 B1 = 2
3150 FOR K = 1 TO L
3160 BS(K + 2) = LEFT$(BLANKS,L1 - B1) + "/"
+ B1$ + AS(K) + B1$ + S$
3170 B1$ = B1$ + " "
3180 B1 = B1 + 1
3190 NEXT K
3200 BS(L + 3) = LEFT$(BLANKS,L1 - B1) + "(
" + B1$ + AS(K) + B1$ + " )"
3210 B1$ = " "
3220 B1 = 1
3230 FOR K = L TO 1 STEP - 1
3240 BS(K + L + 3) = LEFT$(BLANKS,L1 - B1 -
1) + S$ + B1$ + AS(L + K + 1) + B1$ + "/"
"
3250 B1 = B1 + 1
3260 B1$ = B1$ + " "
3270 NEXT K
3280 BS(J + 3) = LEFT$(BLANKS,L1 - 1) + S$ +
LEFT$(UNDERLNS,27) + "/"
3290 BS(J + 4) = BLANKS$ + FLOWS$
3300 REM PAD LINES WITH BLANKS
3310 L1 = LEN(B$(L + 3)) + 5
3320 FOR I = L + 4 TO J + 4
3330 BS(I) = LEFT$(BS(I) + BLANKS,L1)
3340 NEXT I
3350 REM ADD TEXT
3360 K = L + 3
3370 BS(K) = BS(K) + ARROWS$
3380 L2 = RMARGIN
3390 IF LEN(B$(K)) < L2 GOTO 3470
3400 IF MIDS$(B$(K),L2,1) < > ":" AND MIDS$(
B$(K),L2,1) < > " " AND MIDS$(B$(K),L
2,1) < > " " AND MIDS$(B$(K),L2,1) < >
CHR$(126) AND MIDS$(B$(K),L2,1) < >
" " THEN L2 = L2 - 1: IF L2 > RMARGIN -
10 GOTO 3400
3410 ARROWS$ = MIDS$(B$(K),L2 + 1, LEN(B$(K)
))
3420 BS(K) = LEFT$(B$(K),L2)
3430 K = K + 1
3440 IF K > J + 4 THEN BS(K) = LEFT$(LEFT$(
B$(K - 1),31) + BLANKS,L1)
```

```
3450 BS(K) = BS(K) + ARROWS$
3460 GOTO 3380
3470 IF K < J + 4 THEN K = J + 4
3480 GOTO 3490
3490 REM =====
3500 REM PRINT LINES
3510 REM =====
3520 REM
3530 REM -----
3540 REM MARK LOOPS
3550 REM -----
3560 REM
3570 IF LCOUNT + K > (LPAGE) THEN GOSUB 418
0
3580 IF LOOP = 0 GOTO 3660
3590 FOR I = 1 TO K
3600 IF I = 2 GOTO 3620
3610 BS(I) = LEFT$(BS(I),12 - 2 * LOOP) + RIGHTS$(
(" " : : : : : "2 * LOOP) + MIDS$(BS(
I),13,79)
3620 NEXT I
3630 IF MIDS$(BS,7,4) = "NEXT" GOTO 3760
3640 IF MIDS$(BS,7,3) = "FOR" GOTO 3660
3650 GOTO 3880
3660 REM -----
3670 REM START LOOP
3680 REM -----
3690 IF MIDS$(BS,7,3) < > "FOR" GOTO 3880
3700 BS(L) = LEFT$(BS(L),10 - 2 * LOOP) + RIGHTS$(
(" " : : : : : "4 + 2 * LOOP) + MIDS$(
BS(L),15,79): REM 14 PERIODS
3710 LOOP = LOOP + 1
3720 FOR I = L + 1 TO K
3730 BS(I) = LEFT$(BS(I),12 - 2 * LOOP) + RIGHTS$(
(" " : : : : : "2 * LOOP) + MIDS$(BS(
I),13,79)
3740 NEXT I
3750 GOTO 3880
3760 REM -----
3770 REM END LOOP
3780 REM -----
3790 LOOP = LOOP - 1
3800 BS(L) = LEFT$(BS(L),10 - 2 * LOOP) + LEFT$(
(" " : : : : : "4 + 2 * LOOP) + MIDS$(
BS(L),15,79): REM 13 PERIODS AFTER COLO
N
3810 FOR I = 7 TO 80
3820 IF MIDS$(BS,I,1) = " " THEN LOOP = L00
P - 1
3830 NEXT I
3840 FOR I = L + 1 TO K
3850 BS(I) = LEFT$(BS(I),12 - 2 * (LOOP + 1
)) + " " + MIDS$(BS(I),14 - 2 * (LOOP +
1),79)
3860 IF LOOP = 0 THEN BS(I) = LEFT$(BLANKS
,12) + MIDS$(BS(I),13)
3870 NEXT I
3880 REM ----
3890 REM PRINT
3900 REM ----
3910 LCOUNT = LCOUNT + K
3920 FOR I = 1 TO K
3930 PRINT BS(I)
3940 NEXT I: IF EF = 2 THEN EF = 0: GOTO 580
3950 BS(K) = LEFT$(BS(K),31)
3960 NEXT I
3970 GOTO 580
3980 REM ---
3990 REM END
4000 REM ---
4010 IF PEEK(222) < > 5 GOTO 4430
4020 IF LCOUNT > LPAGE - 5 THEN GOSUB 4180
4030 IF FLOWS$ = VS THEN PRINT LEFT$(BLANK
$,22) LEFT$(UNDERLNS,8) "V" LEFT$(UNDER
LNS,8)
4040 IF FLOWS$ = " " THEN PRINT LEFT$(BLAN
K$,22) LEFT$(UNDERLNS,17)
4050 PRINT LEFT$(BLANKS,21)VS LEFT$(BLANK
$,17)VS
4060 PRINT LEFT$(BLANKS,21)VS" END
"VS
4070 PRINT LEFT$(BLANKS,21)VS LEFT$(UNDER
LNS,17)VS
```

```

4080 POKE 216,0
4090 FOR I = 5 TO LPAGE + BMARGIN - LCOUNT
4100 PRINT
4110 NEXT I
4120 PRINT D$"PR#0"
4130 PRINT D$"CLOSE"
4140 PRINT D$"OPEN EXE": PRINT D$"CLOSE EXE"
      : PRINT D$"DELETE EXE"
4150 PRINT D$"OPEN EX1": PRINT D$"CLOSE EX1"
      : PRINT D$"DELETE EX1"
4160 PRINT D$"OPEN EXX.D"D1: PRINT D$"CLOSE
      EXX": PRINT D$"DELETE EXX"
4170 NEW
4180 REM =====
4190 REM PRINT HEADER
4200 REM =====
4210 PRINT LEFT$(B$(K),30):
4220 IF FLOW$ = V$ THEN PRINT "V CONTINUE
      D ON PAGE "PAGE + 1:LCOUNT = LCOUNT + 1
4230 IF FLOW$ < > V$ THEN PRINT " CONTI
      NUED ON PAGE "PAGE + 1:LCOUNT = LCOUNT +
      1
4240 GOSUB 4280
4250 PRINT BLANK$FLOW$" CONTINUED FROM PAGE
      "PAGE - 1
4260 LCOUNT = 5
4270 RETURN
4280 REM -----
4290 REM PRINT TEXT
4300 REM -----
4310 FOR X = 1 TO LPAGE + BMARGIN - LCOUNT
4320 PRINT
4330 NEXT X
4340 PAGE = PAGE + 1
4350 PRINT SPC(LMARGIN)NAME$ SPC(LNLGTH -
      LEN(NAME$) - 9)"PAGE "PAGE
4360 PRINT SPC(LMARGIN)AUTH$ SPC(LNLGTH -
      LEN(AUTH$) - 9)DT$
4370 PRINT SPC(LMARGIN):
4380 FOR X = 1 TO LNLGTH
4390 PRINT "- ";

```

```

4400 NEXT X
4410 PRINT : PRINT
4420 RETURN
4430 E = PEEK(222):EL = PEEK(218) + 256 +
      PEEK(219): POKE 216,0
4440 HOME : VTAB 12: PRINT "ERROR #E" IN LI
      NE "EL.": PRINT : PRINT "PRESS <ESC> TO
      QUIT,": PRINT "OR ANY OTHER KEY TO TRY
      AGAIN."
4450 GET Z$: IF Z$ < > CHR$(27) THEN ON
      EF GOTO 130,4460
4460 PRINT D$"PR#0": PRINT D$"CLOSE": END
4470 GET A$: IF A$ < > CHR$(13) GOTO 4470
4480 RETURN

```

END OF LISTING 3

KEY PERFECT 4.0		80B8	2110 - 2200
RUN ON		A9A8	2210 - 2300
NAVIGATOR.3		307D	2310 - 2400
=====		6C1A	2410 - 2500
CODE	LINE# - LINE#	7AE8	2510 - 2600
----	-----	7F7D	2610 - 2700
5229	10 - 100	7C3C	2710 - 2800
6F7D	110 - 200	8DC5	2810 - 2900
78B2	210 - 300	7D8B	2910 - 3000
4839	310 - 400	7908	3010 - 3100
64FF	410 - 500	750E	3110 - 3200
49CF	510 - 600	5575	3210 - 3300
6F55	610 - 700	7D16	3310 - 3400
4BAD	710 - 800	5760	3410 - 3500
B52C	810 - 900	3B62	3510 - 3600
63E2	910 - 1000	8001	3610 - 3700
306D	1010 - 1100	749F	3710 - 3800
E7A3	1110 - 1200	532F	3810 - 3900
D3E3	1210 - 1300	3065	3910 - 4000
A302	1310 - 1400	80F2	4010 - 4100
6196	1410 - 1500	6979	4110 - 4200
C5BB	1510 - 1600	67C0	4210 - 4300
8AC7	1610 - 1700	526B	4310 - 4400
7638	1710 - 1800	649B	4410 - 4480
439C	1810 - 1900		
4244	1910 - 2000		
A676	2010 - 2100		

PROGRAM CHECK IS : 2BB2