

ROBERT
83 THOMAS

THE nibble PROGRAMMER

The Nibble Programmer

by Paul Hyman
811 Azalea Ave.
Placentia, CA 92670

INTRODUCTION

If you've done a lot of BASIC programming, you've undoubtedly realized that large programs tend to become an unwieldy mess, with GOTO's going every which way. There are other problems too: since subroutines have to be referenced by line number, every time the program is renumbered you need to learn where the subroutines are all over again. Also, since REM statements take up memory space and slow down execution speed, they tend to be left out, making programs harder to understand and debug.

Of course, these are some of the reasons that people use languages like Pascal to write complex programs; but what can you do if you don't have the Pascal system or need to write a program that can run on Apples without Pascal?

The program presented here is one answer. It is a preprocessor which translates a "structured" BASIC program into standard Apple-soft BASIC. Such a scheme has been used successfully with FORTRAN (the Ratfor preprocessor is a good example), a language which has some of the same problems as BASIC.

S-BASIC PROGRAM DESCRIPTION

The "structured" BASIC programs that you'll write using The Programmer will be referred to as **S-BASIC programs** through the remainder of the article. Your S-BASIC programs which are processed by The Programmer can use all of the normal Applesoft statements (except that GOTO's, GOSUB's, and ON statements are a little different and IF's are quite a bit different). S-BASIC programs can also use a couple of new statements, LOOP and WHILE.

In addition, The Programmer allows your new S-BASIC programs to avoid using line numbers. Instead, you can use labels which can be referenced by ON, GOTO, and GOSUB statements.

If you look at one of the listings of an S-BASIC program, you will notice that statements that occur within LOOP, WHILE, and IF statements are indented. This indentation of program lines is not required, but adds considerably to the readability of the program.

DISK FILES

S-BASIC programs are read from one or more disk files by the preprocessor. These disk files can be prepared by any text editor which generates standard DOS sequential text files (e.g. Apple Writer II and Screenwriter). It makes no difference to the preprocessor whether the program it is reading is contained in one or several files. Labels declared in one file can be referenced in another and S-BASIC statements which start in one file may end in another.

There are two main reasons for allowing programs to be split into more than one file. The first is that it is handy to keep commonly used subroutines in separate files so that they can be used with more than one program without having to copy them. The second

reason is that, depending on the text editor being used, the size of an S-BASIC program might exceed the capacity of the editor. With the ability to put the program in more than one file, the only limit on program size is the amount of disk space available. (The resulting translated Applesoft program must, of course, be able to fit into memory.)

HOW TO ENTER THE PROGRAMS

First, type in The Nibble Programmer (from Listing 1) and SAVE it to disk. Then type in the machine language program shown in Listing 2 and save it to disk with the command BSAVE NL.BIN, A\$93A8, L\$252. See the Letters section if you need more information on typing in machine language programs.

RUNNING THE NIBBLE PROGRAMMER

When you RUN the Nibble Programmer, you will first be prompted for the name of the text file which contains your S-BASIC program. You will then see information on the screen as the Nibble Programmer proceeds to convert your S-BASIC program into Apple-soft.

When the message PROGRAM IS NOW LOADED appears on the screen, your converted Applesoft program will be in memory, ready for you to LIST, edit and SAVE it.

WRITING S-BASIC PROGRAMS

S-BASIC programs can be constructed using any word processor that generates sequential DOS text files, or with the Text Editor program presented at the end of this article. You may use any Applesoft commands in addition to the new S-BASIC commands below. The only exception is that you should always use the S-BASIC form of the IF-THEN statement. See Listing 3 for a sample S-BASIC program.

DESCRIPTION OF LANGUAGE FEATURES

LABELS

Labels are used in the event that you want to reference a line with a GOTO, GOSUB, or ON statement. A line is given a label by including a line containing:

```
#<label name>
```

before the line to be referenced. In the following example, the GOTO statement will cause a jump to the assignment statement, A=B.

GOTO XYZ

```
...  
...
```

#XYZ

```
A=B  
...
```

Labels may be used in GOTO, GOSUB, and ON statements. In the case of an ON statement, several labels may be used, as in **ON A GOTO AA, BB, CC, DD.**

Labels must start with an alphabetic character and may be up to 255 characters long, but since the length of a statement is limited to 255 characters it would be a good idea to keep labels considerably shorter than the maximum.

WHILE STATEMENT

The WHILE statement is used to cause a series of statements to be executed as long as some condition is true. The WHILE statement has the form:

WHILE condition

```
...  
statements
```

ENDWHILE

As long as the condition is true, the statements will continue to be executed. If the condition is not true the first time, the statements will not be executed at all.

As an example, the following short program using a WHILE loop will plot random blocks on the Low-Res graphics screen until a key is pressed.

GR

```
X=PEEK (49234)
```

```
X=PEEK (49238)
```

```
WHILE PEEK (-16384) <127
```

```
I = RND(1) * 39
```

```
J = RND(1) * 47
```

```
COLOR = RND(1) * 16
```

```
PLOT I,J
```

```
ENDWHILE
```

```
TEXT
```

```
QUIT
```

LOOP STATEMENT

The LOOP statement is similar to the WHILE statement, except that the check to exit the loop can be done at any point in the loop. It is also allowable not to have any way to exit the loop at all.

The LOOP statement has the form:

LOOP

```
...  
statements
```

```
EXIT LOOP IF condition
```

```
...  
more statements
```

```
ENDLOOP
```

More than one EXIT LOOP is allowed, and EXIT LOOP statements can be at the beginning, end, or anywhere in the middle of the LOOP. It is also permissible to have no EXIT LOOP statement within a loop.

Following is a short sample program using a LOOP statement. This program reads decimal numbers and displays their hexadecimal equivalents. The LOOP statement is used to have the program repeatedly read and output numbers until a zero is entered.

LOOP

```
INPUT Z
```

```
EXIT LOOP IF Z = 0
```

```
FOR I = 1 TO 4
```

```
N%(I) = INT ((Z / 16 - INT (Z / 16)) * 16)
```

```
Q=N%(I)+48+7*(N%(I)>9)
```

```
HTAB 5 -1
```

```
PRINT CHR$(Q);
```

```
Z = INT (Z / 16)
```

```
NEXT
```

```
PRINT
```

```
ENDLOOP
```

```
QUIT
```

continued on next page

The Nibble Programmer (Cont.)

IF STATEMENT

The IF statement in S-BASIC is considerably more powerful than the normal Applesoft IF. It has the form:

```
IF condition
...
statements
...
ELSE IF condition
...
statements
...
ELSE
...
statements
...
ENDIF
```

The ELSE IF's and ELSE are optional. There is no limit on the number of ELSE IF's allowed.

The following program, which converts a hexadecimal number to its decimal equivalent, illustrates the use of the IF statement.

```
N=0
PRINT "?";
LOOP
N=N*16
GET Z$
PRINT Z$;
IF Z$>="0" AND Z$<="9"
N=N+VAL(Z$)
ELSE IF Z$>="A" AND Z$<="F"
N=N+ASC(Z$)-55
ELSE IF Z$=CHR$(13)
N=N/16
PRINT N
END
ELSE
PRINT
PRINT "ILLEGAL HEX CHARACTER"
END
ENDIF
ENDLOOP
```

COMMENT LINES

Comment lines may be included in the program. Any line beginning with an asterisk (*) is interpreted as a comment line and is not passed on to Applesoft. This means that you can include all the comments you want without affecting the size or speed of the final program. REM statements could still be used, but there is no reason to use them.

QUIT STATEMENT

The QUIT statement may be used as the last statement of an S-BASIC program. This will cause the preprocessor to assume that this is the end of the program. If no QUIT is present, you will be asked to enter the name of the next source file. In this manner you can chain S-BASIC files together.

SOME WARNINGS

There are limits to the number of allowable labels, nested statements (IF's within IF's, LOOP's within LOOP's and WHILE's within WHILE's), and S-BASIC statements in a program. The allowable quantities are all quite large, and should be sufficient for normal use. Any of them may be changed by re-dimensioning the arrays for the stacks and Label Tables in the program (see later section on how the program works).

Don't try to put statement numbers on lines. They aren't allowed.

Don't use the normal Applesoft IF statement. Use the S-BASIC variety.

Multiple statements on a line will not work in many cases, and should be avoided. The preprocessor does not check the syntax of the BASIC statements and may get confused if multiple statements are on a line. It is best to keep labels considerably shorter than the maximum.

TWO PASSES

The Nibble Programmer makes two passes through the S-BASIC Source Code. Pass 1 reads the input program, builds up Symbol Tables, and generates the corresponding Applesoft statements for S-BASIC IF, WHILE, and LOOP statements. Pass 2 reads the code generated by pass 1 and substitutes actual line numbers for the labels the user has declared, and for the "internal" labels generated by the first pass.

Pass 1 writes its output on a disk file called **STB.P1.TEMP**. Pass 2 reads this file and writes its output on a file called **STB.P2.TEMP**. At the end of pass 2, the translated program is loaded into memory by EXECing **STB.P2.TEMP**.

INTERNAL LABELS

Internal labels are used to generate code for forward GOTO's. A forward GOTO is one in which the line number to which control is to be transferred is somewhere ahead of the current line. Forward GOTO's occur in IF, WHILE, and EXIT LOOP statements, because it is necessary to generate a GOTO to the end of the statement; but the first pass has no idea where the end of the statement is. The first pass therefore picks an internal label name and generates a GOTO to it. When the end of the particular S-BASIC statement is found, the label and its corresponding line number are entered into the Internal Label Table. Later on, the second pass will look up the internal label and substitute the actual line number for it in the GOTO statement.

Internal labels actually consist of a period followed by a number. The period allows the second pass to distinguish between a user label and an internal label, since user labels must start with a letter. To substitute the line number for the label, the second pass takes the number and uses it as an index into the Internal Label Table, where it finds the actual line number.

WHAT THE PREPROCESSOR CHECKS

The preprocessor does very little checking of the syntax of BASIC statements. If it were to do so, it would be a very large program and very slow also, unless it were written entirely in machine language. Therefore, when there are errors in BASIC statements, you will get a SYNTAX ERROR message from Applesoft when you run the program. However, you can simply fix the line in the translated program and continue (making a mental note to go back and fix the line in the S-BASIC program later).

The preprocessor does check that lines do not start with a number, and IF statements do not contain a GOTO, or have a THEN followed by anything (this would make them normal Applesoft IF's). In addition, checks are made that all IF, LOOP, and WHILE statements have corresponding ENDIF, ENDLOOP, and ENDWHILE statements, and vice versa; and that ELSE's occur within an IF, and EXIT LOOP's occur within LOOP's.

STATEMENT ENCOUNTERERS

Each time the first pass detects an S-BASIC statement, it goes to a special processing routine. These routines are different depending on the statement type, but all do similar things. On encountering a LOOP, IF, or WHILE statement, the first pass picks an internal label number (one number greater than the last), and pushes it on the LOOP, IF, or WHILE stack, as appropriate. These stacks have nothing to do with the 6502 hardware stack; they are simply arrays with a pointer to the last entry. Each time something is pushed on the stack, the pointer is incremented; and each time something is pulled off the stack, the pointer is decremented.

When an ENDOLOOP, ENDIF, or ENDWHILE statement is encountered, the label is pulled off the appropriate stack, and the line number of the current statement is entered into the corresponding entry of the Internal Label Table. This table is then used by the second pass to fix up the GOTO statements with internal label names.

IF statements are somewhat more complicated than the others because they can have several parts. There are two stacks for IF statements — one is used to hold the label for the end of the entire IF, while the other (the IF-segment stack) is used to hold the label for the next ELSE IF or ELSE statement.

PROGRAM SPEED

The first and second pass are both contained in a single BASIC program with a machine language input routine. The machine language routine called NL.BIN reads and deblanks the input line and searches for keywords. If it recognizes any keyword, it passes a number back to the BASIC program indicating which keyword was found. This routine was written in machine language to speed up the program.

Also added to speed up the program were three PEEK (49385) statements. These statements (one in the first pass, one in the second pass, and one on the file which is EXECed to load the final program) keep the disk motor running constantly while the preprocessor is running.

Prior to adding the PEEK statements, it seemed that the preprocessor was just slightly too slow to keep the motor running all the time, and the constant starting and stopping of the disk slowed down processing. These PEEKs will only work for disks using slot 6. If your disk is in another slot, you should take the PEEK's out, or change them to the appropriate values.

THE TEXT EDITOR

What's that you say? The S-BASIC preprocessor sounds like a good idea but you have no text editor to use? Well, here's your answer.

This text editor, originally written in S-BASIC, is no match for a word processor, but it has all of the features you'll need to write and edit S-BASIC programs.

HOW TO TYPE IN THE EDITOR PROGRAMS

First, type in the Text Editor (from Listing 4) and SAVE it to disk. Then type in the machine language program shown in Listing 5 and save it to disk with the command BSAVE STRINGIN, A\$300, L\$FA. See the Letters section if you need more information on typing in machine language programs.

When you RUN the Text Editor program, you will first be prompted for a text file name

— just press the RETURN key unless you are editing a previously constructed file. Then you will be asked to give one of the following commands.

USING THE TEXT EDITOR

The text editor has several commands. Each command consists of a single letter, and most of them allow a number to follow the letter. The editor is line oriented, and always has one line designated as the current line. The default mode of operation is for all commands to operate on the current line unless a number follows the command, in which case that number of lines, beginning with the current line, is used.

The commands recognized by the editor are:

- + or null:** Advance current line.
- :** Move back in file.
- I:** Insert lines after current line. All subsequent input is placed in the file after the current line until a null line is entered.
- B:** Same as 'I' except lines are entered before current line.
- J:** Jump to beginning or end of file.

- S:** Save file to disk.
- Q:** Quit editor.
- A:** Adjust lines right or left.
- D:** Delete lines.
- P:** Pick lines to be copied.
- M:** Merge in picked or deleted lines.
- R:** Replace current line. Allows entering a single line to replace the current line.
- F:** Find line containing specified string.
- X:** Exchange one string for another. The first occurrence of the first string is replaced by the second.
- L:** List current line and the 10 lines before and after it. The current line is shown in inverse so that it can be distinguished from the others. This display is done normally after most commands execute.
- V:** Executes a FRE(0) to display available memory.

Some of the commands, such as JUMP, EXCHANGE, and FIND, ask for additional input after the command has been entered.

Unfortunately, the FIND command is exceedingly slow. This sort of searching for strings should really be done in machine language.

HOW THE EDITOR WORKS

The editor is written entirely in S-BASIC with the exception of a small machine language routine which allows inputting of strings with commas. This routine is from *Nibble*, Vol. 2, No. 2.

All of the files to be edited are read into an array called L\$, and all editing takes place in memory. To avoid moving lines around when lines are added or deleted, the lines are maintained as a linked list, with each line having a link to the previous line and the following line. This allows insertions and deletions to be done by simply changing the links. In addition, lines that have been deleted are linked into an available list so that they can be reused for new lines as new lines are added.

The program is, on the whole, very straightforward. The only area which might be confusing to some people is the use of the linked list. Although linked lists are not complicated, it is somewhat beyond the scope of this article to attempt an explanation here. If you are not familiar with them, good explanations can be found in many books on programming.

LISTING 1: NIBBLE PROGRAMMER

```

10 REM *****
11 REM * NIBBLE PROGRAMMER *
12 REM * BY PAUL M. HYMAN *
13 REM * COPYRIGHT (C) 1983 *
14 REM * BY MICROSPARC, INC *
15 REM * LINCOLN, MA. 01773 *
16 REM *****
100 HOME
105 HIMEM: 37799
110 ER = 0
115 POKE 222,0: POKE 216,0
120 ONERR GOTO 810
130 REM CONSTANT 1
140 CI = 1
200 HTAB 12: PRINT "NIBBLE PROGRAMMER": PRINT : PRINT
   "** COPYRIGHT 1983 BY MICROSPARC, INC. **"
500 D$ = CHR$(4)
510 PRINT D$;"BLOAD NL.BIN"
600 VTAB 20
610 HOME
700 INPUT "INPUT FILE-";FF$
800 PRINT D$;"OPEN ";FF$
805 GOTO 2700
810 REM ***** ERROR ROUTINE
820 X = PEEK(222)
830 IF X = 5 GOTO 900
840 PRINT D$;"CLOSE"
850 PRINT "APPLESOFT/DOS ERROR ";X;" AT LINE "; PEEK
   (218) + PEEK(219) * 256
860 END
900 PRINT D$;"CLOSE ";FF$
910 INPUT "MORE FILES? (Y/N) ";X$
920 IF X$ = "N" THEN PRINT "YOU SHOULD PUT A QUIT AT
   THE END OF THE LAST FILE": GOTO 7710
940 INPUT "FILE NAME - ";FF$
950 PRINT D$;"OPEN";FF$
960 GOTO 6800
1100 REM *** SUBR TO WRITE TO FILE ****
1200 REM
1300 PRINT D$;"WRITE STB.P1.TMP"
1400 PRINT LN;GG$
1500 PRINT D$
1700 LN = LN + 10
1800 RETURN
2700 PRINT D$;"OPEN STB.P1.TMP"
3000 REM THE 4 DIM'S ARE THE SYMBOL TABLE
3100 REM NUMBER OF LABELS
3200 NL = 100
3300 REM NAME FIELD
3400 DIM L$(NL)
3500 REM LINE#
3600 DIM LI$(NL)
4100 REM LABEL TABLE INDEX
4200 LX = 0
4300 LN = 10
4400 REM WHILE STACK
4500 DIM WS$(10)
4600 REM WHILE LABEL STACK

```

```

4700 DIM WL$(10)
4800 REM INIT WHILE STACK PTR
4900 WP = - 1
5000 REM INTERNAL LABEL TABLE
5010 DIM IL$(200)
5100 REM LOOP STACK
5200 DIM LS$(10)
5300 REM LOOP LABEL STACK
5400 DIM LL$(10)
5500 REM INIT LOOP STACK PTR
5600 LP = - 1
5700 REM INITIAL LABEL NUMBER
5800 LI = 0
5900 REM *** THE IF STACK
6000 DIM I1$(10)
6100 REM THE IF-SEG STACK
6200 DIM I2$(10)
6300 REM IP IS PTR TO IF STACK
6400 IP = - 1
6500 REM IQ IS PTR TO IF-SEG STACK
6600 IQ = - 1
6800 A$ = ""
6900 PRINT D$;"READ ";FF$
6910 X = PEEK(49385)
7000 POKE 37800,1
7010 CALL 37802,A$
7100 AL$ = LEFT$(A$,1)
7200 IF AL$ = "*" GOTO 6800
7300 IF AL$ > = "0" AND AL$ < = "9" THEN FLASH : PRINT
   "LINE NUMBERS NOT ALLOWED": NORMAL : ER = ER + 1: GOTO
   6800
7500 X = PEEK(37801)
7510 IF X = 0 GOTO 24100
7520 ON X GOTO 15700,16900,17700,19100,18400,19610,23
   210,21000,22405,7710,23930,8700,11410,11410
7700 STOP
7710 GG$ = "END"
7720 GOSUB 1300
7730 PRINT D$;"WRITE STB.P1.TMP"
7740 PRINT "QUIT"
7800 GOSUB 26300
8000 PRINT D$;"CLOSE STB.P1.TMP"
8100 PRINT D$;"CLOSE ";FF$
8120 IF ER = 0 GOTO 27600
8130 PRINT ER;" ERRORS DETECTED"
8140 PRINT D$;"DELETE STB.P1.TMP"
8150 END
8300 REM
8400 REM ***** LABEL *****
8500 REM
8700 LA = LEN(A$)
8900 A$ = RIGHT$(A$,LA - 1)
8910 SF$ = A$
8920 GOSUB 13900
9000 GOSUB 9300
9100 GOTO 6800
9200 REM
9300 REM **** LABEL DEF. SUBR ****

```

continued on next page

The Nibble Programmer (Cont.)

```

9310 IF LEFT$(A$,1) = "." GOTO 11110
9400 IF LX = 0 GOTO 10700
9500 FOR I = 0 TO LX - 1
9600 IF L$(I) < > A$ GOTO 10400
9800 PRINT L$(I); "---" DOUBLE DEFINITION"
9810 ER = ER + 1
9900 I = 1000
10400 NEXT
10500 IF I = 1001 THEN RETURN : REM DBL DEF ERR
10700 L$(LX) = A$
10800 LIX(LX) = LN
10900 LX = LX + 1
11100 RETURN
11110 A$ = RIGHT$(A$, LEN(A$) - 1)
11120 ILX(VAL(A$)) = LN
11130 RETURN
11200 REM ***** GOTO *****
11300 REM
11410 GOSUB 11800
11420 GG$ = A$
11500 GOSUB 1200
11600 GOTO 6800
11700 REM ** SUBR TO PROC GOTOS **
11800 REM
11810 LM = 0
11830 SF$ = "GOTO"
11900 REM SEARCH LINE FOR GOTO
12000 GOSUB 24600
12100 REM LM=0 MEANS NO GOTO
12200 IF LM < > 0 GOTO 12400
12210 SF$ = "GOSUB"
12220 GOSUB 24600
12230 IF LM = 0 THEN RETURN
12400 LM = LM + 3
12500 REM EXTRACT LABEL AFTER GOTO
12600 GL$ = RIGHT$(A$,LA - LM)
12700 SF$ = GL$
12800 GOSUB 13800
13700 RETURN
13800 REM ***** SUBR TO CHECK LABEL VALIDITY ***
13900 IF LEFT$(SF$,1) > = "A" AND LEFT$(SF$,1) <
= "Z" THEN RETURN
14000 ER = ER + 1
14100 FLASH : PRINT SF$;"-ILLEGAL LABEL": NORMAL
14200 RETURN
15400 REM
15500 REM ***** WHILE *****
15600 REM
15700 R$ = RIGHT$(A$, LEN(A$) - 5)
15800 GG$ = "IF" + R$ + "GOTO" + STR$(LN + 20)
15900 GOSUB 1200
16000 A$ = "GOTO." + STR$(L1)
16100 GG$ = A$: GOSUB 1200
16200 WP = WP + 1
16300 WS$(WP) = LN - 20
16400 WL$(WP) = "." + STR$(L1)
16500 L1 = L1 + 1
16600 GOTO 6800
16800 REM ***** ENDDWHILE *****
16900 IF WP > = 0 GOTO 16990
16910 FLASH
16920 PRINT "NO MATCHING WHILE STATEMENT"
16930 NORMAL
16940 ER = ER + 1
16950 GOTO 6800
16970 GG$ = "GOTO" + STR$(WS$(WP))
17000 GOSUB 1200
17100 A$ = WL$(WP)
17200 WP = WP - 1
17300 REM GOTO LABEL DEF ROUTINE
17400 GOSUB 9300
17410 GOTO 6800
17600 REM ***** LOOP *****
17700 LP = LP + 1
17800 LS$(LP) = LN
17900 LL$(LP) = "." + STR$(L1)
17910 L1 = L1 + 1
18000 LM = LM + 1
18100 GOTO 6800
18300 REM ***** ENDDLOOP *****
18400 IF LP > = 0 GOTO 18490
18410 FLASH : PRINT "NO MATCHING LOOP"
18420 NORMAL
18430 ER = ER + 1
18440 GOTO 6800
18490 GG$ = "GOTO" + STR$(LS$(LP))
18500 GOSUB 1200
18600 A$ = LL$(LP)
18700 LP = LP - 1
18800 GOSUB 9300
18810 GOTO 6800
19000 REM ***** EXIT LOOP *****
19100 R$ = RIGHT$(A$, LEN(A$) - 8)
19110 IF LP > = 0 GOTO 19200
19120 FLASH
19130 PRINT "NO MATCHING LOOP"
19140 NORMAL
19150 ER = ER + 1
19160 GOTO 6800

```

```

19200 A$ = R$ + "GOTO" + LL$(LP)
19300 GG$ = A$: GOSUB 1200
19400 GOTO 6800
19600 REM ***** IF *****
19610 SF$ = "THEN"
19620 GOSUB 24600
19625 REM IF "THEN" EXISTS, NOTHING CAN FOLLOW IT
19630 IF LM = LEN(A$) - 3 GOTO 19700
19635 IF LM > 0 GOTO 19680
19637 REM NO GOTO ALLOWED HERE
19640 SF$ = "GOTO"
19650 GOSUB 24600
19660 IF LM = 0 GOTO 19700
19680 FLASH : PRINT "ILLEGAL IF STATEMENT":ER = ER +
1: NORMAL
19690 PRINT ".....";SF$;" AT CHAR ";LM
19700 GG$ = A$ + "GOTO" + STR$(LN + 20)
19800 GOSUB 1200
19900 IQ = IQ + 1
20000 IP = IP + 1
20100 I1$(IP) = "." + STR$(L1)
20200 L1 = L1 + 1
20300 I2$(IQ) = "." + STR$(L1)
20400 L1 = L1 + 1
20500 A$ = "GOTO" + I2$(IQ)
20600 GG$ = A$: GOSUB 1200
20700 GOTO 6800
20900 REM ***** ELSE IF *****
21000 X$ = A$
21005 IF IP < 0 OR IQ < 0 GOTO 23520
21010 A$ = "GOTO" + I1$(IP)
21020 GG$ = A$: GOSUB 1200
21100 A$ = I2$(IQ)
21200 IQ = IQ - 1
21300 REM DEFINE LABEL
21400 GOSUB 9300
21500 GG$ = RIGHT$(X$, LEN(X$) - 4) + "GOTO" + STR$(
(LN + 20))
21600 GOSUB 1200
21700 IQ = IQ + 1
21800 I2$(IQ) = "." + STR$(L1)
21900 L1 = L1 + 1
22000 A$ = "GOTO" + I2$(IQ)
22100 GG$ = A$: GOSUB 1200
22200 GOTO 6800
22400 REM ***** ELSE *****
22405 IF IP < 0 OR IQ < 0 GOTO 23520
22410 A$ = "GOTO" + I1$(IP)
22420 GG$ = A$: GOSUB 1200
22500 A$ = I2$(IQ)
22600 REM PUT DUMMY LABL ON STACK FOR ENDIF
22700 I2$(IQ) = "." + STR$(L1)
22800 L1 = L1 + 1
22900 GOSUB 9300
23000 GOTO 6800
23200 REM ***** ENDIF *****
23210 IF IQ < 0 GOTO 23520
23300 A$ = I2$(IQ)
23400 IQ = IQ - 1
23500 GOSUB 9300
23510 IF IP > = 0 GOTO 23600
23520 FLASH : PRINT "NO MATCHING IF STATEMENT": NORMAL

23530 ER = ER + 1
23540 GOTO 6800
23600 A$ = I1$(IP)
23700 IP = IP - 1
23800 GOSUB 9300
23900 GOTO 6800
23920 REM ***** ON *****
23930 IF MID$(A$,3,2) = "ERR" GOTO 24040
23940 SF$ = "GOTO"
23950 GOSUB 24600
23960 IF LM > 0 THEN J = LM + 4: GOTO 23990
23970 SF$ = "GOSUB": GOSUB 24600
23980 IF LM > 0 THEN J = LM + 5: GOTO 23990
23985 FLASH : PRINT "ILLEGAL ON STMT":ER = ER + 1: NORMAL
: GOTO 6800
23990 GOSUB 26210
24000 GOSUB 13800
24010 IF LL = 0 GOTO 23990
24015 GG$ = A$
24020 GOSUB 1200
24030 GOTO 6800
24035 REM ***** ONERR *****
24040 GOSUB 11700
24050 GOSUB 1200
24060 GOTO 6800
24070 REM ***** STANDARD BASIC LINE *****
24100 GG$ = A$
24200 GOSUB 1200
24300 GOTO 6800
24400 REM *****
24500 REM SUB TO SEARCH FOR SF$ IN A$
24600 LS = LEN(SF$)
24700 LM = 1
24800 LA = LEN(A$)
24805 REM CHECK FOR QUOTE
24810 IF MID$(A$,LM,1) < > CHR$(34) GOTO 24900
24820 REM SCAN FOR MATCHING QUOTE

```



```

24830 LM = LM + C1
24860 IF MID$(A$,LM,C1) < > CHR$(34) GOTO 24830
24870 LM = LM + 1
24900 IF MID$(A$,LM,LS) = SF$ GOTO 25300
25000 LM = LM + 1
25100 IF LM < LA - LS GOTO 24810
25200 LM = 0
25300 RETURN
25400 REM *****
25500 REM **SUBR TO SEARCH LABEL TABLE
25600 IF LEFT$(SF$,1) < > "." GOTO 25700
25610 SF$ = RIGHT$(SF$, LEN(SF$) - 1)
25620 IZ = IL$(VAL(SF$))
25630 RETURN
25700 IF LX = 0 THEN IZ = - 1: RETURN
25800 I = - 1
25900 FOR D = 0 TO 1
26000 I = I + 1
26010 D = L$(I) = SF$ OR I = LX
26100 NEXT
26110 IZ = I
26120 IF I = LX THEN IZ = - 1
26130 IF IZ < > - 1 THEN IZ = LIX(IZ)
26200 RETURN
26210 REM ***** SUBR TO EXTRACT ON...GOTO LABELS
26220 REM ***** J POINTS TO LABEL
26230 LB$ = ""
26240 LL = 0
26250 IF J > LEN(A$) THEN LL = 1: RETURN
26260 X$ = MID$(A$,J,1)
26270 IF X$ = "." THEN J = J + 1: RETURN
26280 LB$ = LB$ + X$
26290 J = J + 1: GOTO 26250
26300 PRINT D$
26400 PRINT "NUMBER OF LABELS=";LX
27000 IF WP = - 1 GOTO 27410
27100 FOR I = 0 TO WP - 1
27190 FLASH
27200 PRINT "UNTERMINATED WHILE AT STMT ";
27300 PRINT WS$(I)
27310 ER = ER + 1
27320 NORMAL
27400 NEXT
27410 IF LP < 0 GOTO 27420
27415 FLASH : PRINT LP + 1;"UNTERMINATED LOOP STATEME
NT(S)";ER = ER + 1
27420 IF IP < 0 AND IQ < 0 GOTO 27430
27425 FLASH : PRINT "UNTERMINATED IF STATEMENT";ER =
ER + 1
27430 NORMAL
27440 PRINT "----- END PASS 1 -----"
27500 RETURN
27600 PRINT D$;"OPEN STB.P2.TMP"
27700 PRINT D$;"OPEN STB.P1.TMP"
27800 PRINT D$;"WRITE STB.P2.TMP"
27900 PRINT "NEW"
28000 PRINT D$;"READ STB.P1.TMP"
28010 X = PEEK(49385)
28100 POKE 37800,X
28110 CALL 37802,A$
28200 IF PEEK(37801) < > 10 GOTO 29000
28210 IF ER = 0 GOTO 28300
28220 PRINT ER;" ERRORS DETECTED"
28230 PRINT D$;"CLOSE"
28240 PRINT D$;"DELETE STB.P1.TMP"
28250 PRINT D$;"DELETE STB.P2.TMP"

```

```

28260 END
28300 PRINT D$;"WRITE STB.P2.TMP"
28310 PRINT "TEXT"
28315 PRINT "HOME"
28320 PRINT "PRINT"; CHR$(34);"PROGRAM IS NOW LOADED
"; CHR$(34)
28400 PRINT "DELETE STB.P2.TMP"
28500 PRINT D$;"CLOSE STB.P1.TMP"
28600 PRINT D$;"CLOSE STB.P2.TMP"
28700 PRINT D$;"DELETE STB.P1.TMP"
28710 HOME : VTAB 10: HTAB 10: PRINT "LOADING"
28720 VTAB 23
28730 POKE 34,23
28800 PRINT D$;"EXEC STB.P2.TMP"
28900 END
29000 REM
29010 J = 1
29020 IF MID$(A$,J,1) < = "?" THEN J = J + 1: GOTO
29020
29030 SF$ = MID$(A$,J,2)
29050 IF SF$ < > "IF" AND SF$ < > "GO" AND SF$ < >
"ON" GOTO 30300
29100 SF$ = "GOTO"
29200 GOSUB 24600
29300 IF LM > 0 THEN GL = LM + 4: GOTO 29710
29400 SF$ = "GOSUB"
29500 GOSUB 24600
29600 IF LM = 0 GOTO 30300
29700 GL = LM + 5
29710 IF MID$(A$,J,2) = "ON" AND MID$(A$,J,5) < >
"ONERR" GOTO 30700
29800 GL$ = RIGHT$(A$, LEN(A$) - GL + 1)
29805 IF LEFT$(GL$,1) > = "0" AND LEFT$(GL$,1) <
= "9" GOTO 30300
29807 SF$ = GL$
29810 GOSUB 25500
29900 IF IZ = - 1 GOTO 30217
30100 A$ = LEFT$(A$,GL - 1) + STR$(IZ)
30210 GOTO 30300
30217 PRINT D$; FLASH
30220 PRINT "-----";GL$;" UNDEFINED"
30225 ER = ER + 1
30230 NORMAL
30300 PRINT D$;"WRITE STB.P2.TMP"
30400 PRINT A$
30410 V = VAL(A$)
30420 IF INT(V / 100) * 100 = V THEN PRINT "X=PEEK
(49385)"
30500 PRINT D$; PRINT A$
30600 GOTO 28000
30700 REM ** PROCESS ON STATEMENT
30800 GL$ = LEFT$(A$,GL - 1)
31000 J = GL
31100 GOSUB 26210
31200 SF$ = LB$
31300 GOSUB 25700
31400 IF IZ > = 0 GOTO 31500
31410 ER = ER + 1
31420 FLASH : PRINT SF$;" UNDEFINED"
31430 NORMAL
31440 IF LL = 0 GOTO 31100
31450 GOTO 30300
31500 GL$ = GL$ + STR$(IZ)
31600 IF LL = 0 THEN GL$ = GL$ + ","; GOTO 31100
31700 A$ = GL$
31800 GOTO 30300

```

KEY PERFECT 4.0
RUN ON
NIBBLE PROGRAMMER

CODE	LINE#	LINE#
569A	10	110
5184	115	700
53FC	800	910
5616	920	1700
47FD	1800	4100
3F0A	4200	5010
431D	5100	6000
462A	6100	7000
6F54	7010	7720
493C	7730	8300
3463	8400	9300
4E1A	9310	10700
3102	10800	11420
3CB9	11500	12200
3E43	12210	13000
49AA	13900	15900
4B19	16000	16910
3F6F	16920	17400
4591	17410	18400
36BB	18410	18810
4B71	19000	19300
4E6F	19400	19650
5924	19660	20300
4BA9	20400	21100
49A0	21200	22100

48C6	22200	22900
3C3A	23000	23540
4B55	23600	23970
411C	23980	24040
4093	24050	24700
5530	24800	25100
4522	25200	25800
416E	25900	26220
421D	26230	27000
4A68	27100	27420
5F9A	27425	28010
57A3	28100	28300
5E00	28310	28730
5A1E	28800	29300
4FF7	29400	29900
4738	30100	30420
3237	30500	31410
2D41	31420	31800

TOTAL PROGRAM CHECK IS : 1BB4

APPLE CHECKER

ON: NIBBLE PROGRAMMER
TYPE: A

LENGTH: 178B
CHECKSUM: 32

LISTING 2: NL.BIN

*93AB.95F9

```

93A8- 00 00 20 BE DE 20 E3 DF
93B0- 24 11 D0 05 A2 A3 4C 12
93B8- 04 85 85 84 86 A9 00 8D
93C0- A0 94 8D F7 95 8D F5 95
93C8- A5 B8 A4 B9 85 87 84 88
93D0- 20 6F FD 8E A1 94 A0 00
93D8- A2 00 AD A8 93 F0 06 B9
93E0- 00 02 20 ED FD B9 00 02
93E8- C9 A2 D0 0D AD A0 94 49
93F0- 01 8D A0 94 A9 A2 4C FD
93F8- 93 C9 A0 F0 2D 29 7F 9D
9400- A2 94 E8 C8 CC A1 94 90
9408- D1 AD A8 93 F0 05 A9 8D
9410- 20 ED FD AD A2 94 C0 F6
9418- 95 F0 B5 8E A1 94 8A AE
9420- A1 94 A9 00 9D A2 94 4C
9428- 36 94 A9 01 2C A0 94 F0
9430- D2 A9 A0 4C FD 93 A9 00
9438- 85 0D 85 0E AD F8 95 AC
9440- F9 95 20 ED E3 20 3D E7

```

continued on next page

The Nibble Programmer (Cont.)

```

9448- 20 7B DA A5 87 A4 88 85
9450- B8 84 B9 AC F7 95 BE A2
9458- 95 8E F4 95 A2 00 AC F5
9460- 95 BD A2 94 09 80 D9 B0
9468- 95 D0 0A E8 EC F4 95 F0
9470- 25 C8 4C 61 94 AD F4 95
9478- 18 6D F5 95 8D F5 95 AD
9480- F7 95 18 69 01 CD F3 95
9488- F0 06 8D F7 95 4C 53 94
9490- A9 00 8D A9 93 60 AD F7
9498- 95 18 69 01 8D A9 93 60
94A0- 00 00 00 00 00 00 00 00
94A8- 00 00 00 00 00 00 00 00
94B0- 00 00 00 00 00 00 00 00
94B8- 00 00 00 00 00 00 00 00
94C0- 00 00 00 00 00 00 00 00
94C8- 00 00 00 00 00 00 00 00
94D0- 00 00 00 00 00 00 00 00
94D8- 00 00 00 00 00 00 00 00
94E0- 00 00 00 00 00 00 00 00
94E8- 00 00 00 00 00 00 00 00
94F0- 00 00 00 00 00 00 00 00
94F8- 00 00 00 00 00 00 00 00
9500- 00 00 00 00 00 00 00 00

```

```

9508- 00 00 00 00 00 00 00 00
9510- 00 00 00 00 00 00 00 00
9518- 00 00 00 00 00 00 00 00
9520- 00 00 00 00 00 00 00 00
9528- 00 00 00 00 00 00 00 00
9530- 00 00 00 00 00 00 00 00
9538- 00 00 00 00 00 00 00 00
9540- 00 00 00 00 00 00 00 00
9548- 00 00 00 00 00 00 00 00
9550- 00 00 00 00 00 00 00 00
9558- 00 00 00 00 00 00 00 00
9560- 00 00 00 00 00 00 00 00
9568- 00 00 00 00 00 00 00 00
9570- 00 00 00 00 00 00 00 00
9578- 00 00 00 00 00 00 00 00
9580- 00 00 00 00 00 00 00 00
9588- 00 00 00 00 00 00 00 00
9590- 00 00 00 00 00 00 00 00
9598- 00 00 00 00 00 00 00 00
95A0- 00 00 05 08 04 0A 07 02
95A8- 05 06 04 04 02 01 04 05
95B0- D7 C8 C9 CC C5 C5 CE C4
95B8- D7 C8 C9 CC C5 CC CF CF
95C0- D0 C5 D8 C9 D4 CC CF CF
95C8- D0 C9 C6 C5 CE C4 CC CF
95D0- CF D0 C9 C6 C5 CE C4 C9

```

```

95D8- C6 C5 CC D3 C5 C9 C6 C5
95E0- CC D3 C5 D1 D5 C9 D4 CF
95E8- CE A3 C7 CF D4 CF C7 CF
95F0- D3 D5 C2 0E 00 00 AA 00
95F8- A2 94

```

KEY PERFECT 4.0		
RUN ON NL.BIN		
CODE	ADDR#	ADDR#
28F5	6000	604F
280C	6050	609F
293C	60A0	60EF
27C5	60F0	613F
00	6140	618F
00	6190	61DF
187E	61E0	622F
15FE	6230	6251
TOTAL PROGRAM CHECK IS : 0252		
APPLE CHECKER		
ON: NL.BIN		
TYPE: B		
LENGTH: 0252		
CHECKSUM: 2C		

LISTING 3: SAMPLE S-BASIC PROGRAM

```

HOME
*****
* S-BASIC TEST PROGRAM THE NIBBLE PROGRAMMER *
* THIS PROGRAM GENERATES RANDOM LENGTH LINES IN LO-RES *
* GRAPHICS, WHOSE COLORS CHANGE DEPENDING ON *
* THE SETTING OF PADDLE 0 *
*****
PRINT "TURN PADDLE 0 TO GET DIFFERENT COLORS"
PRINT
#STARTOVER
LOOP
PRINT "TYPE V FOR VERTICAL LINES ONLY"
PRINT "TYPE H FOR HORIZONTAL LINES ONLY"
PRINT "TYPE B FOR BOTH"
INPUT C$
EXIT LOOP IF C$="V" OR C$="H" OR C$="B"
ENDLOOP
IF C$="V"
PRINT "GENERATING VERTICAL LINES ONLY"
ELSE IF C$="H"
PRINT "GENERATING HORIZONTAL LINES ONLY"
ELSE
PRINT "GENERATING BOTH VERTICAL AND HORIZONTAL LINES"
ENDIF
PRINT "PRESS ANY KEY TO QUIT"
FOR I=0 TO 3000
NEXT
OR
POKE 49234,0
*CLEAR THE ENTIRE LO-RES SCREEN
FOR I=1024 TO 2047
POKE I,0
NEXT
WHILE PEEK (-16384) < 127
I = RND (1) * 39
J = RND (1) * 47
P = INT ( PDL (0) / 19)
COLOR= RND (1) * 3 + P
IF C$="H" OR C$="B"
MH=RND(1)*39+1
LH=RND(1)*39
HLIN LH,MH AT J
ENDIF
IF C$="V" OR C$="B"
LV=RND(1)*47
MV=RND(1)*47+1
VLIN LV,MV AT I
ENDIF
ENDWHILE
TEXT
HOME
*CLEAR KEYBOARD STROBE
POKE 49168,0
INPUT "RUN AGAIN? ";C$
IF C$="Y" OR C$="YES"
GOTO STARTOVER
ENDIF
END
QUIT

```

LISTING 4: TEXT EDITOR

```

10 REM *****
11 REM * TEXT EDITOR *
12 REM * BY PAUL M. HYMAN *
13 REM * COPYRIGHT (C) 1983 *
14 REM * BY MICROSPARC, INC *
15 REM * LINCOLN, MA. 01773 *
16 REM *****
20 D$ = CHR$(4):ML = 1000
30 DIM L$(ML)
40 DIM FL$(ML)
50 DIM BL$(ML)
60 AVPTR = 0
70 IUPTR = 0
80 LL = 0
90 D$ = CHR$(4)
100 PRINT D$;"BLOAD STRINGIN"
110 INPUT "ENTER FILE NAME (RETURN IF NONE) ";FF$
120 IF FF$ < > "" GOTO 140
130 GOTO 310
140 ONERR GOTO 280
150 PRINT D$;"OPEN ";FF$
160 PRINT D$;"READ ";FF$
170 LL = LL + 1
180 CALL 760,A$
190 L$(LL) = A$
200 IF LL > 1 GOTO 220
210 GOTO 250
220 FL$(LL - 1) = LL
230 BL$(LL) = LL - 1
240 GOTO 260
250 IUPTR = 1
260 PRINT L$(LL)
270 GOTO 170
280 POKE 216,0
290 PRINT D$
300 CL = 1
310 HOME
320 GOSUB 3390
330 PRINT
340 INPUT "ENTER COMMAND ";C$
350 LC = LEN (C$)
360 IF LC > 1 THEN GOTO 380
370 GOTO 460
380 N$ = RIGHT$(C$,LC - 1)
390 N = VAL (N$)
400 C$ = LEFT$(C$,1)
410 IF N < = 0 GOTO 430
420 GOTO 450
430 PRINT "INVALID NUMBER"
440 GOTO 330
450 GOTO 470
460 N = 1
470 IF C$ = "Q" THEN GOTO 490
480 GOTO 560
490 IF SV = 0 GOTO 510
500 GOTO 540
510 PRINT "CHANGES NOT SAVED"
520 SV = 1
530 GOTO 550
540 END

```

continued on page 26

Do you want the #1 Seller or the #1 Financial System?

FEATURES	The Home Accountant	The ACCOUNTANT Finance Data Base System
Transactions Per Disk	1000	2000 4000
Number of Codes	1	63
Automatic Transactions	25	900
Number Frequency	Once a month	Unlimited
Double Entry	NO	YES
Accounting Background Required	NO	NO
Accommodates Any Type Transaction	NO	YES
Transaction Retrieval	One at a time	Screen at a time
Backdate Transactions	NO	YES
Ability to Interrupt While Printing	SOMETIMES	ALWAYS
132 COL. PRINTER REQUIRED	YES	NO
NUMERIC FORMATTING	292661	292,661.42
Optional VisiCalc Interface	NO	YES
PERFORMANCE		
Startup to Transaction Entry	113 sec	44 sec
Begin Printing Balance Sheet After Entering Transactions	162 sec	1 sec
Begin Printing Transactions After Entering Transactions	106 sec	2 sec
RATING		
Peelings II evaluation	B-D	A
PRICE		
APPLE II file Personal Version	\$75	\$129
IBM PC Personal Version	\$150	\$195
APPLE II file The Business ACCOUNTANT™	---	\$255
IBM PC The Business ACCOUNTANT™	---	\$295

*Based on APPLE benchmarks. The Accountant's performance superiority is even greater on the IBM PC.

MONEY MAGAZINE — Nov. 1982

"Among bookkeeping programs, earns high marks and is easy to use."

CREATIVE COMPUTING — Jan. 1983

"The documentation is thorough, easily read, and complete."

"The program is so easy to use that rarely will reference have to be made to the manual."

SOFTALK — Jan. 1982

"For the home user (and perhaps in some cases complex small business), the best package we evaluated was The ACCOUNTANT by Decision Support Software."

"The ACCOUNTANT does a great job of making financial management a simple and straightforward process."

INFOWORLD — Jan. 3/10, 1983

"Complete flexible financial data base package for the home user."

"... exceptionally fast ... highly recommend."

The ACCOUNTANT Finance Data Base System™

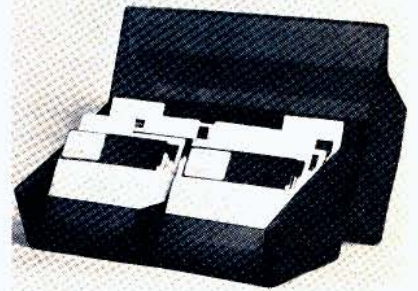
Decision Support Software Inc.

1438 Ironwood Drive, McLean, VA 22101 • (703) 241-8316 • Orders Only: (800) 368-2022

Apple™ IBM® VisiCalc™ The Home Accountant™ are trademarks of Apple IBM VisiCorp and Continental Software respectively.

CIRCLE NUMBER 15

GRAND SLAMS from HOME BASE SUPPLY



FLIP SORT PLUS™

5-1/4" Smoke acrylic flip sort holds 100+ diskettes!

\$27.95

DISK BANKS

Interlocking disk storage. Capacity 12 diskettes per unit.

\$4.50 ea.

VINYL PAGES

Clear vinyl disk storage pages to fit standard 3 ring binders.

2 Diskettes per page.

\$7.50/10 pk.

4 Diskettes per page.

\$9.50/10 pk.

COPY HOLDER

Beige colored steel, clamps to counter or desk.

\$19.95



WE PAY SHIPPING

HOME BASE SUPPLY

10096 Soquel Dr. #4
Aptos, California 95003

(408) 688-5597

CIRCLE NUMBER 16

NIBBLE/VOL. 4/NO. 6/1983 25

The Nibble Programmer (Cont.)

```

550 GOTO 3240
560 IF C$ = "F" GOTO 580
570 GOTO 740
580 INPUT "STRING TO SEARCH FOR:";SF$
590 I = CL
600 PRINT ".";
610 GOSUB 3760
620 IF LM < > 0 GOTO 660
630 I = FLX(I)
640 IF I = 0 GOTO 660
650 GOTO 680
660 PRINT
670 IF LM < > 0 GOTO 690
680 GOTO 720
690 CL = I
700 GOSUB 3390
710 GOTO 730
720 PRINT "STRING NOT FOUND"
730 GOTO 3240
740 IF C$ = "X" GOTO 760
750 GOTO 1080
760 PRINT "STRING EXCHANGE"
770 INPUT "ENTER STRING TO SEARCH FOR:";SF$
780 INPUT "ENTER REPLACEMENT STRING:";RS$
790 I = CL
800 NT = 0
810 NT = NT + 1
820 PRINT ".";
830 GOSUB 3760
840 IF LM < > 0 GOTO 880
850 I = FLX(I)
860 IF I = 0 OR NT = N GOTO 880
870 GOTO 810
880 PRINT
890 IF LM < > 0 GOTO 910
900 GOTO 1060
910 CL = I
920 IF LM > 1 GOTO 940
930 GOTO 960
940 LF$ = LEFT$(L$(CL),LM - 1)
950 GOTO 970
960 LF$ = ""
970 LR = LEN(L$(CL)) - LM - LEN(SF$) + 1
980 IF LR > 0 GOTO 1000
990 GOTO 1020
1000 LR$ = RIGHT$(L$(CL),LR)
1010 GOTO 1030
1020 LR$ = ""
1030 L$(CL) = LF$ + RS$ + LR$
1040 GOSUB 3390
1050 GOTO 1070
1060 PRINT "STRING NOT FOUND"
1070 GOTO 3240
1080 IF C$ = "P" GOTO 1100
1090 GOTO 1200
1100 I = 0
1110 L = CL
1120 IF I < N AND L < > 0 GOTO 1140
1130 GOTO 1180
1140 PB$(I) = L$(L)
1150 L = FLX(L)
1160 I = I + 1
1170 GOTO 1120
1180 NP = I
1190 GOTO 3240
1200 IF C$ = "M" GOTO 1220
1210 GOTO 1380
1220 SV = 0
1230 PRINT NP;" LINES IN PICK BUFFER"
1240 IF NP > 0 GOTO 1260
1250 GOTO 1350
1260 FOR I = 0 TO NP - 1
1270 GOSUB 3250
1280 L$(K) = PB$(I)
1290 BL$(K) = CL
1300 FLX(K) = FLX(CL)
1310 FL$(K) = K
1320 CL = K
1330 NEXT
1340 GOTO 1360
1350 PRINT "NOTHING IN PICK BUFFER"
1360 GOSUB 3390
1370 GOTO 3240
1380 IF C$ = "L" GOTO 1400
1390 GOTO 1420
1400 GOSUB 3390
1410 GOTO 3240
1420 IF C$ = "" OR C$ = "+" GOTO 1440
1430 GOTO 1550
1440 FOR I = 1 TO N
1450 IF FLX(CL) < > 0 GOTO 1470
1460 GOTO 1480
1470 CL = FLX(CL)
1480 NEXT
1490 IF N = 1 GOTO 1510
1500 GOTO 1530
1510 PRINT L$(CL)
1520 GOTO 1540
1530 GOSUB 3390

```

```

1540 GOTO 3240
1550 IF C$ = "--" GOTO 1570
1560 GOTO 1600
1570 FOR I = 1 TO N
1580 IF BLX(CL) < > 0 GOTO 1600
1590 GOTO 1610
1600 CL = BLX(CL)
1610 NEXT
1620 IF N = 1 GOTO 1640
1630 GOTO 1660
1640 PRINT L$(CL)
1650 GOTO 1670
1660 GOSUB 3390
1670 GOTO 3240
1680 IF C$ = "R" GOTO 1700
1690 GOTO 1750
1700 SV = 0
1710 CALL 760,A$
1720 L$(CL) = A$
1730 GOSUB 3390
1740 GOTO 3240
1750 IF C$ = "I" GOTO 1770
1760 GOTO 1920
1770 SV = 0
1780 CALL 760,A$
1790 IF A$ = "" GOTO 1900
1800 GOSUB 3250
1810 L$(K) = A$
1820 IF CL < > 0 GOTO 1840
1830 GOTO 1880
1840 BLX(K) = CL
1850 FLX(K) = FLX(CL)
1860 BL$(K) = BLX(CL)
1870 FL$(K) = K
1880 CL = K
1890 GOTO 1780
1900 GOSUB 3390
1910 GOTO 3240
1920 IF C$ = "B" GOTO 1940
1930 GOTO 2100
1940 SV = 0
1950 CALL 760,A$
1960 IF A$ = "" GOTO 2000
1970 GOSUB 3250
1980 L$(K) = A$
1990 FLX(K) = CL
2000 BLX(K) = BLX(CL)
2010 IF BLX(CL) = 0 GOTO 2030
2020 GOTO 2050
2030 IUPTR = K
2040 GOTO 2060
2050 FLX(BLX(CL)) = K
2060 BLX(CL) = K
2070 GOTO 1950
2080 GOSUB 3390
2090 GOTO 3240
2100 IF C$ = "D" GOTO 2120
2110 GOTO 2460
2120 SV = 0
2130 PRINT "DELETE ";N;" LINE(S)"
2140 I = 0
2150 NP = 0
2160 IF I < N GOTO 2180
2170 GOTO 2400
2180 PB$(I) = L$(CL)
2190 Z = CL
2200 PRINT PB$(I)
2210 IF FLX(CL) = 0 GOTO 2230
2220 GOTO 2300
2230 CL = BLX(CL)
2240 FLX(CL) = 0
2250 FLX(Z) = AVPTR
2260 AVPTR = Z
2270 PRINT "END OF BUFFER"
2280 GOTO 2400
2290 GOTO 2350
2300 CL = FLX(CL)
2310 BLX(CL) = BLX(Z)
2320 FLX(BLX(Z)) = CL
2330 FLX(Z) = AVPTR
2340 AVPTR = Z
2350 IF BLX(CL) = 0 GOTO 2370
2360 GOTO 2380
2370 IUPTR = CL
2380 I = I + 1
2390 GOTO 2160
2400 NP = N
2410 IF I < NP GOTO 2430
2420 GOTO 2440
2430 NP = I + 1
2440 GOSUB 3390
2450 GOTO 3240
2460 IF C$ = "A" GOTO 2480
2470 GOTO 2700
2480 SV = 0
2490 RL$ = ""
2500 IF RL$ < > "R" AND RL$ < > "L" GOTO 2520
2510 GOTO 2540
2520 INPUT "ADJUST LINES RIGHT OR LEFT (R/L)? ";RL$

```

KEY PERFECT 4.0
RUN ON
TEXT EDITOR

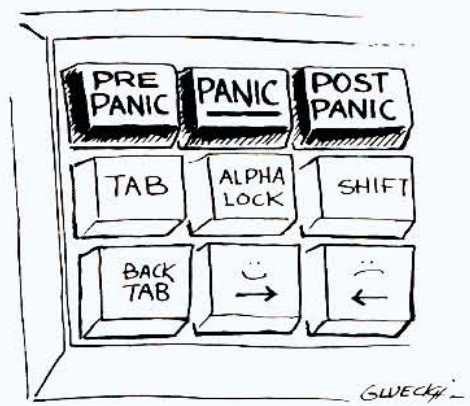
CODE	LINE#	LINE#
5850	10	40
3EE6	50	140
3700	150	240
2410	250	340
3957	350	440
27F7	450	540
338D	550	640
2C27	650	740
3D43	750	840
2F6C	850	940
3DCC	950	1040
2E00	1050	1140
345B	1150	1240
2C7A	1250	1340
384B	1350	1440
2949	1450	1540
301D	1550	1640
29B0	1650	1740
30FD	1750	1840
2F62	1850	1940
34ED	1950	2040
3262	2050	2140
2D1A	2150	2240
39F0	2250	2340
2AC4	2350	2440
4934	2450	2540
469F	2550	2640
3D05	2650	2740
2FEE	2750	2840
4022	2850	2940
37F4	2950	3040
658D	3050	3140
BF74	3150	3240
2937	3250	3340
2326	3350	3440
37BF	3450	3540
2ECC	3550	3640
2817	3650	3740
35F0	3750	3840
0BFD	3850	3890

TOTAL PROGRAM CHECK IS : 1434

APPLE CHECKER

ON: TEXT EDITOR
TYPE: A

LENGTH: 0F70
CHECKSUM: 4D

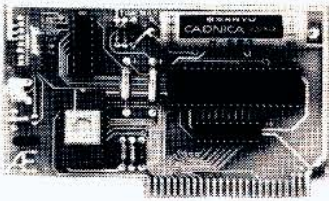


APPLIED ENGINEERING

THE BEST PERIPHERALS FOR THE BEST COMPUTER

The TIMEMASTER

Finally a clock that does it ALL!



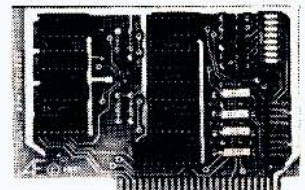
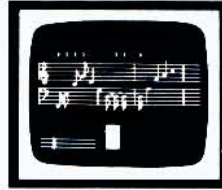
- Designed in 1983 using I.C. technologies that simply did not exist when most other Apple clocks were designed.
- Just plug it in and your programs can read the year, month, date, day, and time to 1 millisecond! The only clock with both year and ms.
- Powerful 2K ROM driver — No clock could be easier to use.
- Full emulation of most other clocks, including Mountain Hardware's Appletime (but you'll like the TIMEMASTER mode better).
- Basic, Machine Code, CP/M and Pascal software on 2 disks!
- Eight software controlled interrupts so you can execute two programs at the same time. (Many examples are included)
- On board timer lets you time any interval up to 48 days long down to the nearest millisecond.

The TIMEMASTER includes 2 disks with some really fantastic time oriented programs (over 25) plus a DOS dater so it will automatically add the date when disk files are created or modified. This disk is over a \$200.00 value alone — we give the software others sell. All software packages for business, data base management and communications are made to read the TIMEMASTER.

If you want the most powerful and the easiest to use clock for your Apple, you want a TIMEMASTER.

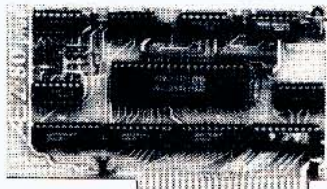
PRICE \$129.00

Super Music Synthesizer



- Complete 16 voice music synthesizer on one card. Just plug it into your Apple, connect the audio cable (supplied) to your stereo, boot the disk supplied and you are ready to input and play songs.
- It's easy to program music with our compose software. You will start right away at inputting your favorite songs. The Hi-Res screen shows what you have entered in standard sheet music format.
- Now with new improved software for the easiest and fastest music input system available anywhere.
- We give you lots of software. In addition to Compose and Play programs, 2 disks are filled with over 30 songs ready to play.
- Easy to program in Basic to generate complex sound effects. Now your games can have explosions, phaser zaps, train whistles, death cries. You name it, this card can do it.
- Four white noise generators which are great for sound effects.
- Plays music in true stereo as well as true discrete quadraphonic.
- Full control of attack, volume, decay, sustain and release.
- Will play songs written for ALF synthesizer (ALF software will not take advantage of all the features of this board. Their software sounds the same in our synthesizer.)
- Automatic shutoff on power-up or if reset is pushed.
- Many many more features.

PRICE \$159.00



Z-80 PLUS!

- **TOTALLY** compatible with ALL CP/M software.
- The only Z-80 card with a special 2K "CP/M detector" chip.
- Fully compatible with microsoft disks (no pre-boot required).
- All new 1983 design incorporates the latest in I.C. technologies.

- Red "CP/M WORKING" LED indicator, the Z-80 Plus does not interfere with non-CP/M programs.
- An on-card PROM eliminates many I.C.'s for a cooler, less power consuming board. (We use the Z-80A at a fast 4MHZ)
- Does EVERYTHING the other Z-80 boards do, plus Z-80 interrupts. Don't confuse the Z-80 Plus with crude copies of the microsoft card. The Z-80 Plus employs a much more sophisticated and reliable design. With the Z-80 Plus you can access the largest body of software in existence. Two computers in one and the advantages of both, all at an unbelievably low price.

PRICE \$139.00

COMING SOON: The Z-80 Plus for the Apple III

Viewmaster 80

There used to be about a dozen 80 column cards for the Apple, now there's only **ONE**.

- **TOTALLY** Videx Compatible
- 80 characters by 24 lines, with a sharp 7x9 dot matrix
- On-board 40/80 soft video switch with manual 40 column override
- Fully compatible with ALL Apple languages and software — there are NO exceptions
- Low power consumption through the use of CMOS devices
- All connections on the card are made with standard video connectors, no cables are soldered to the board
- All new 1983 design (using a new Microprocessor based C.R.T. controller)

JUST COMPARE!

PRICE	BUILT IN SOFTWARE	SHIFT KEY SUPPORT	PASCAL COMPATIBLE	7x9 DOT MATRIX	40/80 INPUTS	40 COLUMN OVERRIDE	INVERSE CHARACTERS
VIEWMASTER 169	YES	YES	YES	YES	YES	YES	YES
SUPRTERM 375	NO	YES	NO	NO	NO	YES	YES
WIZARD80 245	NO	NO	YES	YES	NO	YES	YES
VISION80 375	YES	YES	YES	YES	NO	NO	NO
OMNIVISION 295	NO	YES	NO	NO	NO	YES	YES
VIEWMAX80 219	YES	YES	YES	YES	NO	NO	YES
SMARTERM 360	YES	YES	YES	NO	NO	YES	NO
VIDEOTERM 345	NO	NO	NO	YES	YES	NO	YES

The VIEWMASTER 80 works with all 80 column applications including CP/M, Pascal, WordStar, Format II, Easywriter, Apple Writer II, Visicalc, and many others. The VIEWMASTER 80 is THE MOST compatible 80 column card you can buy at ANY price!

PRICE \$169.00

MemoryMaster IIe 128K RAM Card

- Expands your Apple IIe to 192K memory
- Provides an 80 column text display
- Compatible with all Apple IIe 80 column and extended 80 column card software (Same physical size as Apple's 64K card)
- Available in 64K and 128K configurations
- Bank select LED's for each 64K bank
- Permits your IIe to use the new double high resolution graphics
- Automatically expands Visicalc to 95K storage in 80 columns! The 64K configuration is all that's needed, 128K can take you even higher.

- Complete documentation included, we show you how to use all 128K. If you already have Apple's 64K card, just order the MEMORYMASTER with 64K and use the 64K from your old board to give you a full 128K. (The board is fully socketed so you simply plug in more chips.)

MemoryMaster with 128K \$249
Upgradeable MemoryMaster with 64K \$169
Non-Upgradeable MemoryMaster with 64K \$149

Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec. components used throughout. P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products work in APPLE IIe, II, II+ and Franklin (except MemoryMaster). Applied Engineering also manufactures a full line of data acquisition and control products for the Apple; A/D converters and digital I/O cards, etc. Please call for more information. All our products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no hassle **THREE YEAR WARRANTY.**

Send Check or Money Order to:
APPLIED ENGINEERING
 P.O. Box 470301
 Dallas, TX 75247

Call (214) 492-2027
 7 a.m. to 11 p.m. 7 days a week
 MasterCard, Visa & C.O.D. Welcome

All Orders Shipped Same Day. Texas Residents Add 5% Sales Tax. Add \$10.00 If Outside U.S.A. Dealer Inquiries Welcome.

Compute the Savings

Apple Hit List

Galactic Gladiator	\$ 27
The Cosmic Balance	\$ 27
Pursuit of the Graf Spree	\$ 39
Wordrace	\$ 17
David's Midnight Magic	\$ 23
Sargon II	\$ 23
Zork I, II, III	Ea. \$ 27
Cyborg	\$ 22
Crisis Mountain	\$ 23
Cytron Masters	\$ 27
S.E.U.I.S.	\$ 27
Knight of Diamonds	\$ 23
Way Out	\$ 27
Zaxxon	\$ 27
S.A.G.A. Adventure	Ea. \$ 27
Serpentine	\$ 23
Choplifter	\$ 23
Frogger	\$ 23
Sea Fox	\$ 20
Temple of Apshai	\$ 27
Ultima	\$ 27
Castle Wolfenstein	\$ 20
Wiz & Princess	\$ 22
Ulysses & The Golden Fleece	\$ 23
Wizardry	\$ 34
Tigers in the Snow	\$ 27
Sherwood Forest	\$ 23
Deadline	\$ 34
Bandits	\$ 23
Starcross	\$ 27
Aztec	\$ 27
Mask of the Sun	\$ 27
Pie Man	\$ 20
Miner 2049er	\$ 29
Wavy Navy	\$ 23
Ultima	\$ 39
Dark Crystal	\$ 27
Evolution	\$ 34
Millionaire	\$ 55
Hi Res Secret	\$ 79
Police Artist	\$ 23
Suspended	\$ 35
S.A.G.A.: Secret Mission	\$ 27

Educational

Master Type	\$ 27
Type Attack	\$ 27
Harcourt Brace S.A.T. Series	\$ 59
Anatomy	\$ 27
Compu Spell Gr. 4-8	Ea. \$ 15
Algebra I, II, III	Ea. \$ 27
Planetary Guide	\$ 23
Solar System	\$ 27
Sticky Bear Bop	\$ 30
Sticky Bear Numbers	\$ 30
Sticky Bear ABC	\$ 30
Alien Counter/Face Flash	\$ 24
Gulp & Arrow Graphics	\$ 24
Battling Bugs/Concentration	\$ 24
The Jar Game/Chaos	\$ 24
PDI Preschool IQ Builder	\$ 24
In Search of Amazing Things	\$ 27
Hey Diddle Diddle/Spinaker	\$ 20
Snooper Troops #1/Spinaker	\$ 30
Snooper Troops #2/Spinaker	\$ 30
Delta Drawing/Spinaker	\$ 45
Story Maker/Spinaker	\$ 26
Face Maker/Spinaker	\$ 26

Apple IIe

64K with 80 column card, one Apple drive, one Apple low glare monochrome (green phosphor) monitor, one monitor stand

Call for Price

Peripherals

Softcard	\$229	Buffered Dumping 32K	\$199
Ramcard	\$ 69	8088 Coprocessor Board	\$549
Microsoft Premium Sys	\$479	Versawriter Graphics Tablet	\$209
(Contains Softcard, Ramcard, Videx, Videoterm)		Videoterm 80 Column Board	\$229
System Saver Fan	\$ 85	Vista Vision 80 Column Board	\$189
Flip & File Diskette Box	\$ 21	Smarterm II 80 Column Board	\$149
Enhancer II	\$ 99	EZ Port	\$ 19
Lower Case Adapter	\$ 25	Game Paddles	\$ 29
Microtek Apple Parallell Interface	\$ 79	Joystick TG	\$ 39
Microbuffer II 32K Parallell	\$199	Select-A-Port	\$ 39
Apple Dumping GX	\$ 95	Kraft Joystick	\$ 45
Buffered Dumping 16K	\$155	Kraft Paddles	\$ 34
		The Joypoint	\$ 34
		TG Trak Ball	\$ 45

Graphics/Utilities

Beagle Brothers		Penguin Software	
Apple Mechanic	\$ 20	Complete Graphics System	\$ 49
Tip Disk #1	\$ 15	Special Effects	\$ 27
Flex Text	\$ 20	Graphics Magician	\$ 39
Frame Up	\$ 20		
Typefaces	\$ 15		

★★★ Specials of the Month ★★★

Franklin Ace 1000 w/Color	\$849
Elephant Disks s/s	\$ 18
Verbatim Disks s/d	\$ 24
Maxell Diskettes	\$ 30
Amdek Color I	\$299
Amdek Color II	\$659
Micromodem II	\$259
Micromodem II w/Terminal Pk.	\$299
Novation Apple Cat II Modem	\$329
Hayes Smartmodem 300	\$209
Hayes Smartmodem 1200	\$499
The Grappler +	\$119
Rana System 40 Track Drive	\$299
Rana System 80 Track Drive	\$659



1095 E. Twain (702) 796-0296

Las Vegas, Nevada 89109

Call Toll Free

1-800-634-6766

Order Line Only

Information & Order Inquiries (702) 369-5523

We accept VISA and MasterCard

Mon. — Fri. 8 A.M. to 6 P.M. — Sat. 9 A.M. to 6 P.M.

Dealers' Inquiries Invited

Top Sellers

Artsci	
Magic Window II	\$ 95
Ashton-Tate	
dBase II (Req. Z-80)	\$419
Broderbund	
Payroll	\$249
Accounts Receivable	\$249
The Bank Street Writer	\$ 45
Continental Software	
The Home Accountant	\$ 48
1st Class Mail	\$ 48
CPA Module #1 — G/L	\$159
CPA Module #2 — A/R	\$159
CPA Module #3 — A/P	\$159
CPA Module #4 — Payroll	\$159
CPA Module #5 — Property Mgt	\$159
The Form Letter	\$ 20
Datasoft	
Micropainter	\$ 23
Datasm 65 2.0	\$ 59
Lisp Interpreter	\$ 79
The Basic Compiler	\$ 65
Don't Ask Software	
S.A.M.	\$ 79
Microcomputer Products Hayes	
Hayes Terminal Program	\$ 65
Howard Software	
Creative Financing	\$129
Real Estate Analyzer II	\$119
Tax Preparer 1983	\$145
IUS	
Professional Easywriter	\$115
Original Easywriter	\$ 65
Original Easymailer	\$ 45
Pro. Easywriter/Mailer Combo	\$189
Orig. Easywriter/Mailer Combo	\$ 89
Link Systems	
Datafax	\$119
Datalink	\$ 62
Lotus	
Executive Briefing System	\$125
Alphabets Font	\$ 17
Alphabets Font	\$ 17
Micro Lab	
The Tax Manager	\$ 95
Data Factory 5.0	\$189
Payroll Manager	\$189
Microsoft	
Tasc Compiler	\$119
A.L.D.S. (Req. Z-80)	\$ 79
Multipan (Apple Dos or Z-80)	\$175
Systems Plus	
Accounting Plus — G/L	\$239
Accounting Plus (GL, AP, AR)	\$499
Accounting Plus (GL, AP, AR, INV)	\$629
On-Line	
Screenwriter II	\$ 82
The Dictionary	\$ 65
Screenwriter Professional	\$129
The General Manager II	\$149
PBL Corporation	
Personal Investor	\$ 95
Sensible Software	
Sensible Speller	\$ 79
Multi Disk Catalog III	\$ 17
Silicon Valley	
Word Handler II	\$129
List Handler	\$ 59
Dictionary	\$ 79
Software Publishing	
PFS: Report	\$ 85
PFS	\$ 85
PFS: Graph	\$ 85
Sorcim/ISA	
Supercalc (Req. Z-80)	\$179
Spellguard (Req. Z-80)	\$125
Southeastern Software	
Data Capture 4.0	\$ 47
Data Capture 4.0/80	\$ 59
Visicorp	
Visicalc 3.3	\$165
Visiplot	\$139
Visitrans/Plot	\$199
Visidex	\$165

ORDERING INFORMATION AND TERMS: For Fast Delivery send cashier checks, money orders or direct bank wire transfers. Personal and company checks allow 3 weeks to clear. C.O.D. orders (\$3.00 minimum) and 1% of all orders over \$300. School purchase orders welcomed. Prices reflect a cash discount only and are subject to change. Please enclose your phone number with any orders. **Shipping** — Software (\$2.50 minimum). Shipping — Hardware (please call). Foreign orders, APO & FPO orders — \$10 minimum and 15% of all orders over \$100. Nevada residents add 5 3/4% sales tax. All goods are new and include factory warranty. Due to our low prices, all sales are final. All returns must have a return authorization number. Call 702-369-5523 to obtain one before returning goods for replacement.

The Nibble Programmer (Cont.)

```

2530 GOTO 2500
2540 INPUT "HOW MANY SPACES? ";Q
2550 I = 0
2560 IF RL$ = "R" GOTO 2580
2570 GOTO 2610
2580 BB$ = " "
2590 L$(CL) = LEFT$(BB$,Q) + L$(CL)
2600 GOTO 2620
2610 L$(CL) = RIGHT$(L$(CL), LEN(L$(CL)) - Q)
2620 IF FL$(CL) < > 0 GOTO 2640
2630 GOTO 2650
2640 CL = FL$(CL)
2650 I = I + 1
2660 IF I = N OR FL$(CL) = 0 GOTO 2680
2670 GOTO 2560
2680 GOSUB 3390
2690 GOTO 3240
2700 IF C$ = "S" GOTO 2720
2710 GOTO 2840
2720 INPUT "FILE NAME-";F$
2730 PRINT D$;"OPEN";F$
2740 PRINT D$;"WRITE";F$
2750 I = IUPTR
2760 IF I < > 0 GOTO 2780
2770 GOTO 2810
2780 PRINT L$(I)
2790 I = FL$(I)
2800 GOTO 2760
2810 PRINT D$;"CLOSE";F$
2820 SV = I
2830 GOTO 3240
2840 IF C$ = "J" GOTO 2860
2850 GOTO 3020
2860 LC$ = ""
2870 IF LC$ < > "B" AND LC$ < > "E" GOTO 2890
2880 GOTO 2920
2890 INPUT "JUMP TO END OR BEGINNING?";C$
2900 LC$ = LEFT$(C$,1)
2910 GOTO 2870
2920 IF LC$ = "B" GOTO 2940
2930 GOTO 2960
2940 CL = IUPTR
2950 GOTO 3000
2960 IF FL$(CL) < > 0 GOTO 2980
2970 GOTO 3000
2980 CL = FL$(CL)
2990 GOTO 2960
3000 GOSUB 3390
3010 GOTO 3240
3020 IF C$ = "V" GOTO 3040
3030 GOTO 3060
3040 PRINT "AVAILABLE MEM="; FRE (0)
3050 GOTO 3240
3060 PRINT "??"
3070 PRINT "LEGAL COMMANDS ARE:"
3080 PRINT "+ OR NULL: MOVE FORWARD"
3090 PRINT "-: MOVE BACKWARD"
3100 PRINT "I: INSERT LINES AFTER CURRENT LINE"
3110 PRINT "B: INSERT LINES BEFORE CURRENT LINE"
3120 PRINT "J: JUMP TO BEGINNING OR END OF FILE"
3130 PRINT "S: SAVE TO DISK"
3140 PRINT "Q: QUIT"
3150 PRINT "A: ADJUST LINES RIGHT OR LEFT"
3160 PRINT "D: DELETE LINES"
3170 PRINT "P: PICK LINES (TO USE WITH M COMMAND)"
3180 PRINT "M: MERGE IN PICKED OR DELETED LINES"
3190 PRINT "R: REPLACE CURRENT LINE"
3200 PRINT "F: FIND LINE CONTAINING SPECIFIED STRING"

```

```

3210 PRINT "X: EXCHANGE STRING"
3220 PRINT "V: PRINT AVAILABLE MEMORY"
3230 PRINT "L: LIST CURRENT LINE AND 10 BEFORE AND AF
TER"
3240 GOTO 330
3250 IF IUPTR = 0 GOTO 3270
3260 GOTO 3310
3270 K = 1
3280 IUPTR = 1
3290 LL = 1
3300 GOTO 3380
3310 IF AUPTR = 0 GOTO 3330
3320 GOTO 3360
3330 LL = LL + 1
3340 K = LL
3350 GOTO 3380
3360 K = AUPTR
3370 AUPTR = FL$(K)
3380 RETURN
3390 IF IUPTR = 0 GOTO 3410
3400 GOTO 3420
3410 RETURN
3420 HOME
3430 N = 10
3440 I = CL
3450 NT = 0
3460 NT = NT + 1
3470 IF BL$(I) < > 0 GOTO 3490
3480 GOTO 3500
3490 I = BL$(I)
3500 IF BL$(I) = 0 GOTO 3530
3510 IF NT = 10 GOTO 3530
3520 GOTO 3460
3530 FD = I
3540 NT = 0
3550 I = CL
3560 NT = NT + 1
3570 IF FL$(I) < > 0 GOTO 3590
3580 GOTO 3600
3590 I = FL$(I)
3600 IF FL$(I) = 0 GOTO 3630
3610 IF NT = 10 GOTO 3630
3620 GOTO 3560
3630 LD = I
3640 I = FD
3650 IF I = CL GOTO 3670
3660 GOTO 3680
3670 INVERSE
3680 PRINT L$(I)
3690 IF I = CL GOTO 3710
3700 GOTO 3720
3710 NORMAL
3720 IF I = LD GOTO 3750
3730 I = FL$(I)
3740 GOTO 3650
3750 RETURN
3760 LS = LEN(SF$)
3770 LM = 1
3780 A$ = L$(I)
3790 LA = LEN(A$)
3800 IF LM < LA - LS + 1 GOTO 3820
3810 GOTO 3870
3820 IF MID$(A$,LM,LS) = SF$ GOTO 3840
3830 GOTO 3850
3840 GOTO 3880
3850 LM = LM + 1
3860 GOTO 3800
3870 LM = 0
3880 RETURN
3890 END

```

LISTING 5: STRING IN

*300.3F9

```

0300- 20 BE DE 20 E3 DF 24 11
0308- D0 05 A2 A3 4C 12 D4 85
0310- 85 84 86 A5 B8 A4 B9 85
0318- 87 84 88 20 2C D5 AD 00
0320- 02 C9 03 D0 03 4C 63 D8
0328- A9 00 85 0D 85 0E A9 00
0330- A0 02 20 ED E3 20 3D E7
0338- 20 7B DA A5 87 A4 88 85
0340- B8 84 B9 20 B7 00 60 00
0348- 00 00 00 00 00 00 00 00
0350- 00 00 00 00 00 00 00 00
0358- A8 B0 08 28 28 28 28 28
0360- 00 50 A8 10 10 10 28 48
0368- 60 00 E8 20 00 01 00 A8
0370- 02 03 A8 04 05 06 00 00
0378- 30 C8 00 48 07 08 00 00
0380- A0 09 0A 0B 0C 0D A0 88
0388- 0E 0F 10 11 12 13 EB 14

```

```

0390- 15 16 17 18 19 1A 78 78
0398- 78 18 20 20 20 20 20 20
03A0- 20 1B 20 1C 1D 1E 20 20
03A8- 20 1F 20 20 20 21 20 22
03B0- 23 24 25 26 27 28 20 20
03B8- 20 20 20 29 2A 2B 20 2C
03C0- 2D 2E 2F 30 31 32 20 20
03C8- 33 34 35 36 37 38 20 39
03D0- 4C BF 9D 4C 84 9D 4C FD
03D8- AA 4C B5 B7 AD 0F 9D AC
03E0- 0E 9D 60 AD C2 AA AC C1

```

```

03E8- AA 60 4C 51 A8 EA EA 4C
03F0- 59 FA BF 9D 38 4C 58 FF
03F8- 4C 65

```

KEY PERFECT 4.0 RUN ON STRINGIN

CODE	ADDR# - ADDR#
2973	0300 - 034F
25FB	0350 - 039F
28BA	03A0 - 03EF
051E	03F0 - 03F9

TOTAL PROGRAM CHECK IS : FA

APPLE CHECKER

ON: STRINGIN
TYPE: B

LENGTH: 00FA
CHECKSUM: 8B

Disk "A" Offer

ED. The Programmer, Magazine Tracking, and The Nibbler are available on diskette for an introductory price of \$19.95 plus \$1.50 shipping/handling (\$2.50 outside the U.S.) from Nibble, P.O. Box 325, Lincoln, MA 01773. Offer expires 11/30/83.

For
Franklin Ace,
Orange, Basis 108

For
Apple II
Apple II E

Disk Drive

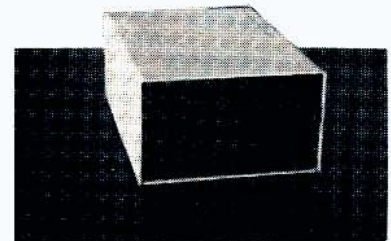
BY IMS

\$199

Disk Interface

\$65

- Direct Replacement for Apple Disk II
- Full Function Apple Compatible
- 35 Track • Half Tracking
- DOS 3.3 and 3.2 compatible
- 90 day warranty



SPECIALS

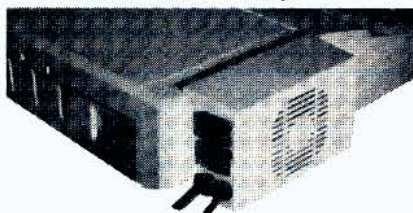
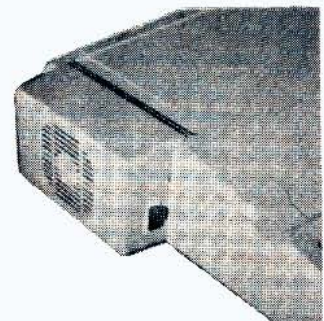
Amdek Color I - \$290	Wabash 5¼ Diskettes - \$19	IMS Centronics Interface & Cable - \$65 Gorilla-Pruf, self centering Joystick - \$40
16K Ram Card - \$45	View Max 80 Col. Board - \$175	

TORNADO FAN with surge suppression

\$49.95

By IMS

- New Dust Filter!!!
- High Reliability Cooling Fan
- Switched Duplex convenience outlets on rear
- Quiet & Efficient
- Security from glitches and voltage spikes
- Protect your computer and peripherals
- Clip on 1 second installation (on Apple and on Monitor stands)
- Lighted power switch
- 1 year warranty



IMS Gorilla - Pruf Joystick \$40

- Self-centering with trim tabs
- Unique strain relief
- Heavy duty metal cabinet
- Long life, oversized switches



VISA/MasterCharge

\$2 Phone Credit

Dealer Inquiries Invited



RJ ELECTRONICS LABORATORY

Box 112, Dept. A, Warrenville, Illinois 60555

Phone (312) 393-4755

CIRCLE NUMBER 137

