Word Search: Puzzle Program Inside For Commodore, Atari, Apple, IBM, & TI

COMPUTE!

$2.95
September
1985
Issue 64
Vol. 7, No. 9
$3.75 Canada
02193
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing

THE AMIGA FROM COMMODORE: An In-Depth Review

Programs Inside:

Disk Commander
Add 18 Powerful Commands To Your Commodore 64

Atari Animation

All About IBM Batch Files

Jump Search For Commodore, Atari, Apple, IBM, And TI

Easy Apple Screen Editing Enhanced BASIC Line Editor For Apple II+, IIe, IIc

71486 02193

COMPUTE!

SEPTEMBER 1985
VOLUME 7
NUMBER 9
ISSUE 64

NOTE: See page 70 before typing in programs.

TOLL FREE Subscription Order Line
800-334-0868 (In NC 919-275-9809)

AP Apple, **Mac** Macintosh, **AT** Atari, **V** VIC-20, **64** Commodore 64, **+4** Commodore Plus/4, **16** Commodore 16, **128** Commodore 128, **P** PET/CBM, **TI** Texas Instruments, **PC** IBM PC, **PCjr** IBM PCjr, **CC** Radio Shack Color Computer.
*General interest.

COMPUTE! Publications, Inc.
One of the ABC Publishing Companies:
ABC Publishing, President, Robert G. Burton
1330 Avenue of the Americas, New York, New York 10019

Address all inquiries to:
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, $24. Send subscription orders or change of address (P.O. form 3579) to **COMPUTE!** Magazine, P.O. Box 10954, Des Moines, IA 50340. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X.

```
20 GOTO 10
30 END
```

Save the program by entering this line: BSAVE "UNPRO.CIM",&H400,&H7F. To unprotect a protected program, load the protected program into memory, then enter this line: BLOAD "UNPRO.CIM". I suspect that Nicholson's procedure may be required on some compatibles, since Pollock's does not simply query a standard location for standard information. A quick test on my friend's Sperry PC-compatible showed that it disables the BLOAD command while a protected program is in memory. However, Pollock's procedure does have the advantage of requiring much less typing.

Guy R. Winters

We tested this method on the PC and PCjr and found that you need to BSAVE only one byte of memory. Type in any one-line program such as 10 END. Then enter this command: BSAVE"UN.PRO",1124,1. The BSAVE command saves one byte of memory at location 1124 (&H464 hexadecimal). Now load a protected program (one that was saved with SAVE"filename",P), and load the one-byte file with BLOAD"UN-.PRO". On the PC/PCjr, the protection evaporates and you can list, edit, or save the program as usual. Also, PEEK and POKE are reenabled in direct mode.

The PC and PCjr use location 1124 as a flag: It contains 0 when an unprotected program is in memory and 254 after you load a protected program. The BSAVE shown above saves location 1124 at a time when we know the flag is set to 0. The BLOAD simply loads the 0 back into location 1124, resetting the flag to signify no protection. As you found by testing your friend's Sperry, "compatibility" is a relative concept. Evidently one of the Sperry designers knew or anticipated this trick, and prevented it by disabling BLOAD.

Although program protection disables POKE and PEEK in immediate mode, both commands are still legal in program mode (at least on the PC/PCjr). Thus, a protected program can unprotect itself while running (for instance, if you enter a password) and an unprotected program can protect itself as well. The PCs we tested put a 254 in location 1124 to indicate protection, but in fact any nonzero value seems to set the protection flag: Editing, listing, PEEKing, and POKEing are ruled out, and you can resave the program only in protected format.

Disabling Apple's Break Key

According to your answer to Alex Tarlecky's letter in December 1984, the RESET key can be disabled on the Apple IIc with the command POKE 1012,PEEK(1012) AND 10. But is there a way to also disable the CONTROL-C function to keep people from breaking out of my programs?

Mike Sanders

Yes, there is. After Applesoft BASIC executes a program statement, it checks for any errors that might have occurred. At the same time, it checks to see if CTRL-C was pressed. If so, Applesoft responds as it does when it encounters a syntax error or illegal quantity error. Normally, it stops the program and displays an appropriate error message (BREAK IN line#).

The secret to trapping CTRL-C is an instruction that changes the way Applesoft handles such errors—the ONERR statement. For instance, once the computer executes a statement such as ONERR GOTO 1000, it responds to any error—including the CTRL-C function—by transferring control to line 1000 (or any other line you specify with ONERR). Make sure, however, that the line specified in the ONERR statement actually exists in your program. Otherwise, Applesoft searches for an undefined line when an error happens, causing another error. The result is an endless loop and a locked-up computer.

You should put an error-handling routine starting at the line number referred to by ONERR. This routine should PEEK location 222, which contains an error code. If this location contains 255, then CTRL-C was pressed. The best way to deal with CTRL-C is to have your error routine GOTO the program's main menu or some other predictable location, so that CTRL-C still causes a break but doesn't stop the program.

If PEEK(222) isn't 255, then CTRL-C wasn't pressed—an actual error occurred. This could be a disk error (wrong disk in the drive, no disk, disk full, etc.) or an error in your program. It is usually easier to let Applesoft handle the errors that you aren't expecting. You can do this by POKEing memory location 216 with 0 to cancel the ONERR trap. Then use the Applesoft RESUME instruction, which re-executes the statement that caused the error in the first place. Since the instruction didn't finish the first time, you should get the same error, but this time the program halts with an appropriate error message.

TI Supplies

Just after I purchased a TI-99/4A computer, the company went out of business. Does this mean I won't be able to purchase anything for my computer? I would like to purchase Extended BASIC, a printer, and other peripherals.

Kathy Armstrong

Texas Instruments is still very much in business; it has simply stopped manufacturing home computers such as the TI-99/4A. Fortunately, TI-99/4A products are still available. The following firms carry software, hardware, and peripherals (this is the most complete and accurate list we were able to compile at time of publication):

Triton Products
P.O. Box 8123
San Francisco, CA 94128
1-800-227-6900

Unisource Electronics, Inc.
P.O. Box 64240
Lubbock, TX 79464
1-800-858-4580

MSW Computers & Electronics
22 East Tioga Street
Tunkahannock, PA 18657
1-800-233-3266

Tenex Computer Express
P.O. Box 6578
South Bend, IN 46660
219-259-7051

Reader Cynthia Becker informs us that hardware and software are also available through the TI-99/4A National Assistance Group. After paying a $10 membership fee, you are entitled to purchase TI products from this organization and receive its newsletter as well:

TI-99/4A National Assistance Group
P.O. Box 290812
Ft. Lauderdale, Florida 33329
(305) 583-0467

Commodore 16 Conversions

I have found that programs written for the VIC-20 Super Expander will run on the Commodore 16 as well if you add the BASIC 3.5 statement SCALE 1=1023*1023 to the beginning of the program. The 16 uses different tokens for graphics keywords like DRAW, POINT, and so on. But the programs will load without any problem from disk or tape. After you load the program, edit the lines that contain those keywords and save it again. It should run just fine.

John Elliot

Thanks for the information.

Trapping IBM's Break Key

I own an IBM PC and have been trying to trap the Ctrl-Brk keys. I have looked in a tremendous number of books, but still couldn't find anything about it. I haven't been able to scan the keyboard for the information I need. How can I trap those keys?

Patrick McGarry

Since many readers have asked this question, we'll show you two techniques that work with BASICA or Cartridge BASIC on either the PC or PCjr. The following program traps both Ctrl-Break (break) and Ctrl-Alt-Del (reboot).

Word Search

Original Program By Michael B. Williams

This computerized puzzle-maker can provide hours of challenging fun. We've included versions for Commodore, IBM PC/PCjr, Apple II-series, TI-99/4A, and Atari computers. A printer is required.

You're probably familiar with word search puzzles: Certain words are hidden in a rectangle of nonsense letters, and it's your job to hunt them down. "Word Search" lets you create such puzzles on your computer's printer with words of your own choice. Since you design the puzzle, you can make it as easy or as difficult as you want, using up to 100 different words on some computers. Topical puzzles make the game even more interesting. For example, you might include only computer words, the names of foreign cities, or stumpers like "uxorious" and "bougainvillaea." Parents and teachers can make puzzles for children using weekly vocabulary lists.

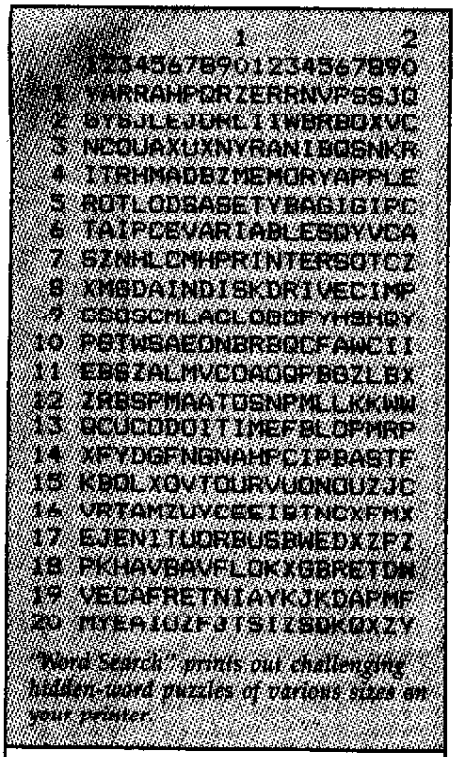
If you're using an Atari, type in and save Program 8, then skip to the program instructions below. For other computers, we've saved space by listing Word Search in the form of one main program with separate line changes and additions for each specific machine. If you're using a Commodore, Apple, IBM PC/PCjr, or TI-99/4A, the first step is to find the specific listing for your computer. Before typing anything, cross out every line in the main program (Program 1) that has the *same* line number as a line in the listing for your computer. Then type in all the lines listed for your computer, as well as all the lines in Program 1 that haven't been crossed out.

No matter which computer you're using, save a copy of Word Search and refer to the notes below before running the program. The following instructions apply to every version:

Word Search begins by asking you for the number of words to be hidden. When you've answered that question, the computer asks you to choose the number of rows and columns for the puzzle grid. Since the grid must be big enough to hide all the words, the computer tells you when you've made the grid too small and lets you try again.

Next, Word Search lets you enter the words one by one. There's no particular limit on word length, but keep in mind that the words must fit inside the grid. (For example, you can't fit a 12-letter word in a 6 × 6 grid.) Since longer words are harder to fit into the grid, the computer sorts the words by length (from longest to shortest) so it can place the longest words first. When many words are involved, this can take a few minutes, so be patient.

Once the words are sorted, you're allowed to name the puzzle. You also have the option of printing the solution to the puzzle (parents and teachers might want to separate the solution from the puzzle until the puzzle has been tried). After printing one puzzle, you can create another, using the same word list (the words will be rearranged) or entirely new words. Word Search is designed to permit a maximum of 100 words in a 99 × 99 grid (exceptions for certain computers are noted below). However, puzzles of that size can take a long time to create—over an hour in some cases. In addition, many



"Word Search" prints out challenging hidden-word puzzles of various sizes on your printer.

printers can't print more than 80 columns unless you first send the printer a special escape code for condensed type (see your printer manual).

Commodore Versions

The line changes listed as Program 2 are for the Commodore 64, 128, Plus/4, 16, PET, and VIC-20 (with at least 8K expansion). If you're using a VIC with only 8K expansion, type in the line changes shown in Program 2 and also substitute lines 95 and 100 in Program 4. If you're using a Commodore 16, type the line changes from Program 2 and also substitute lines 95 and 100 in Program 3. The VIC with only 8K expansion can hide a maximum of 50 words in a 50 × 50 grid; the 16 is limited to a maximum of 60 words in a 60 × 60 grid. If you're using a PET, you'll have to make similar adjustments, depending on the amount of memory available.

Apple And IBM

The Apple version of Word Search runs on any Apple II-series computer with either DOS 3.3 or ProDOS. Follow the general instructions above, typing in the line changes listed as Program 5. IBM users should enter the line changes in Program 6; this version runs on a PC or PCjr with any memory configuration.

TI Word Search

Program 7 lists the line changes required for TI. The unexpanded TI-99/4A is limited to 50 words in a 50 × 50 grid. However, with memory expansion this number can be increased by changing the value of MC in line 95 from 50 to the desired value. You will also need to increase every occurrence of 50 in line 100 to the same value. Adjust line 2000 for whatever configuration your particular printer requires.

Atari Version

The Atari version of Word Search is complete in itself. Simply type in Program 8, save a copy, and run it. Ataris with 32K or 48K memory can create puzzles with up to 100 words in a 99 × 99 grid. If your Atari has 16K, you're limited to 25 words in a 25 × 25 grid. To run Word Search on a 16K Atari you must make two additional changes in line 100 of Program 8: Change the 99 and the 100 to 25.

Program 1: Word Search (Main Program)

Version By Patrick Parrish, Programming Supervisor

Please refer to the article instructions before entering this listing.

```
95 MC=99
100 DIM FF$(100),S$(99),W$(100
    ),CC(100),RR(100),L(100),E
    $(2,2)
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 READ E$(I+1,J+1)
140 NEXT J
150 NEXT I
160 DATA "NW"," N","NE"," W","
    {2 SPACES}"," E","SW"," S"
    ,"SE"
170 FOR I=1 TO MC
180 G$=G$+" "
190 NEXT I
200 FOR I=1 TO 8
210 READ D(1,I),D(2,I)
220 NEXT I
230 DATA -1,-1,-1,0,-1,1,0,-1
240 DATA 0,1,1,-1,1,0,1,1
250 GOTO 1220
260 REM SHELL SORT
270 PRINT "SORTING..."
280 X=1
290 X=2*X
300 IF X<=W0 THEN 290
310 X=INT(X/2)
320 IF X<>0 THEN 340
330 RETURN
340 FOR Y=1 TO W0-X
350 Z=Y
360 A=Z+X
370 IF L(Z)>=L(A) THEN 460
380 X$=W$(Z)
390 W$(Z)=W$(A)
400 W$(A)=X$
410 B=L(Z)
420 L(Z)=L(A)
430 L(A)=B
440 Z=Z-X
450 IF Z>0 THEN 360
460 NEXT Y
470 GOTO 310
480 REM HIDE WORDS
490 FOR X=1 TO W0
500 FOR Y=1 TO 50
510 R1=INT(RND(1)*R0)
520 C1=INT(RND(1)*C0)
530 D1=INT(RND(1)*8)+1
540 O1=D1
550 DX=D(1,D1)
560 DY=D(2,D1)
570 IF R1+DX*L(X)<1 OR R1+DX*L
    (X)>R0 OR C1+DY*L(X)<1 THE
    N 590
580 IF C1+DY*L(X)<=C0 THEN 630
590 D1=D1*(D1<8)*(1=1)+1
600 IF D1<>O1 THEN 550
610 NEXT Y
620 GOTO 800
630 FOR Z=1 TO L(X)
640 IF MID$(W$(X),Z,1)<"A" OR
    {SPACE}MID$(W$(X),Z,1)>"Z"
     THEN 680
650 R1=R1+DX
660 C1=C1+DY
670 IF MID$(S$(R1),C1,1)<>" "
    {SPACE}AND MID$(S$(R1),C1,
    1)<>MID$(W$(X),Z,1) THEN 5
    90
680 NEXT Z
690 FOR Z=L(X) TO 1 STEP -1
700 IF MID$(W$(X),Z,1)<"A" OR
    {SPACE}MID$(W$(X),Z,1)>"Z"
     THEN 770
710 S$(R1)=MID$(S$(R1),1,C1-1)
    +MID$(W$(X),Z,1)+MID$(S$(R
    1),C1+1)
720 RR(X)=R1
730 CC(X)=C1
740 FF$(X)=E$(DX+1,DY+1)
750 R1=R1-DX
760 C1=C1-DY
770 NEXT Z
780 NEXT X
790 GOTO 890
800 GOSUB 1720
810 PRINT "SORRY, BUT I CAN'T
    {SPACE}FIT WORD NUMBER ";S
    TR$(X);" , ";W$(X);" , ";
820 PRINT "INTO THE GRID. SHOU
    LD I SKIP IT, START OVER,
    {SPACE}OR TRY AGAIN"
830 INPUT X$
840 IF MID$(X$,1,2)="ST" THEN
    {SPACE}1660
850 IF MID$(X$,1,2)="TR" THEN
    {SPACE}500
860 IF MID$(X$,1,2)<>"SK" THEN
     830
870 W$(X)="/"
880 GOTO 780
890 FOR X=1 TO R0
900 FOR Y=1 TO C0
910 IF MID$(S$(X),Y,1)<>" " TH
    EN 930
920 S$(X)=MID$(S$(X),1,Y-1)+CH
    R$(INT(26*RND(1)+65))+MID$
    (S$(X),Y+1)
930 NEXT Y
940 NEXT X
950 REM DONE
960 PRINT
970 PRINT "I AM FINISHED. WHAT
     DO YOU WANT TO CALL THE W
    ORD SEARCH"
980 INPUT T$
990 SL=0
1000 PRINT
1010 PRINT "DO YOU WANT TO PRI
     NT THE SOLUTION (Y/N)"
1020 GOSUB 1180
1030 IF A$="N" THEN 1050
1040 SL=1
```

```
1050 GOSUB 2000
1060 GOSUB 1720
1070 F=0
1080 PRINT "DO YOU WANT ANOTHE
     R GRID (Y/N)"
1090 GOSUB 1180
1100 IF A$="Y" THEN 1120
1110 END
1120 PRINT
1130 PRINT "DO YOU WANT TO USE
      THE SAME WORDS (Y/N)"
1140 GOSUB 1180
1150 IF A$="N" THEN 1280
1160 F=1
1170 GOTO 1340
1180 INPUT A$
1190 IF A$<>"Y" AND A$<>"N" TH
     EN 1180
1200 RETURN
1210 REM INITIALIZATION
1220 GOSUB 1720
1230 LL=6
1240 GOSUB 1740
1250 PRINT "{8 SPACES}WORD SEA
     RCH"
1260 LL=4
1270 GOSUB 1740
1280 FOR I=1 TO W0
1290 W$(I)=""
1300 L(I)=0
1310 NEXT I
1320 PRINT "HOW MANY WORDS WOU
     LD YOU LIKE IN YOUR WORD
     {SPACE}SEARCH"
1330 INPUT W0
1340 PRINT
1350 PRINT "HOW MANY ROWS AND
     {SPACE}COLUMNS IN THE GRI
     D"
1360 INPUT R0,C0
1370 PRINT
1380 PRINT
1390 IF R0*C0>=10*W0 THEN 1440
1400 PRINT "I DON'T THINK I CO
     ULD DO THIS."
1410 FOR I=1 TO 1000
1420 NEXT I
1430 GOTO 1340
1440 PRINT "I THINK I CAN DO T
     HIS."
1450 IF C0<=MC THEN 1470
1460 PRINT "(BUT IT WON'T FIT
     {SPACE}ON THE PAPER.)"
1470 IF F=1 THEN 1660
1480 LL=3
1490 GOSUB 1740
1500 PRINT "ENTER THE ";STR$(W
     0);" WORDS. TO CORRECT A
     {SPACE}MISTAKE, ENTER X"
1510 PRINT
1520 FOR I=1 TO W0
1530 PRINT "WORD NUMBER ";I;":
     "
1540 INPUT X$
1550 IF LEN(X$)<=R0 AND LEN(X$
     )<=C0 AND X$<>"X" THEN 16
     10
1560 IF X$<>"X" THEN 1590
1570 I=I-(I>1)*(1=1)
1580 GOTO 1530
1590 PRINT "OOPS...THE WORD IS
      TOO LONG."
1600 GOTO 1530
1610 W$(I)=X$
1620 L(I)=LEN(X$)
1630 NEXT I
1640 GOSUB 1720
1650 GOSUB 270
1660 PRINT
1670 PRINT "OKAY, I WILL GO TO
      WORK (WISH ME LUCK...)."
1680 FOR I=1 TO R0
1690 S$(I)=LEFT$(G$,C0)
1700 NEXT I
1710 GOTO 490
1730 RETURN
1740 FOR I=1 TO LL
1750 PRINT
1760 NEXT I
1770 RETURN
1999 REM PRINTER ROUTINE
```

Program 2: Line Changes For Commodore 64, 128, Plus/4, 16, PET, and VIC-20

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
1720 PRINT CHR$(147)     :rem 69
2000 OPEN3,4:PRINT#3,T$:PRINT#
     3                   :rem 101
2010 PRINT#3,"{4 SPACES}";:FOR
     I=1TOC0:IFI/10<>INT(I/10)
     THENPRINT#3," ";:GOTO2030
                         :rem 101
2020 PRINT#3,MID$(STR$(I),2,1)
     ;                   :rem 207
2030 NEXTI:PRINT#3       :rem 106
2040 PRINT#3,"{4 SPACES}";:FOR
     I=1TOC0:PRINT#3,RIGHT$(ST
     R$(I),1);:NEXTI:PRINT#3
                         :rem 172
2050 FORX=1TOR0:IFX<10THENPRIN
     T#3," ";            :rem 20
2060 PRINT#3,STR$(X)" ";
                         :rem 28
2070 FORY=1TOC0:PRINT#3,MID$(S
     $(X),Y,1);          :rem 98
2080 NEXTY:PRINT#3:NEXTX:PRINT
     #3:PRINT#3:PRINT#3,"WORD
     {SPACE}LIST:"       :rem 201
2090 FORX=1TOW0:IFW$(X)="/"THE
     N2110               :rem 50
2100 PRINT#3,W$(X)       :rem 246
2110 NEXTX:FORI=1TO5:PRINT#3:N
     EXTI:IFSL=0THEN2160
                         :rem 185
2120 PRINT#3,"SOLUTION LIST:":
     PRINT#3,"WORD{21 SPACES}R
     OW{3 SPACES}COLUMN";
                         :rem 213
2130 PRINT#3,"{3 SPACES}DIR"
                         :rem 248
2140 FORX=1TOW0:IFW$(X)="/"THE
     N2170               :rem 52
2150 PRINT#3,W$(X);LEFT$(G$,25
     -LEN(W$(X)));RR(X);LEFT$(
     G$,8-LEN(STR$(RR(X))));
                         :rem 218
2160 PRINT#3,CC(X);LEFT$(G$,6-
     LEN(STR$(CC(X))));FF$(X)
                         :rem 61
2170 NEXTX               :rem 97
2180 CLOSE3:RETURN       :rem 142
```

Program 3: Additional Line Changes For Commodore 16

```
95 MC=60
100 DIM FF$(60),S$(60),W$(60),
    CC(60),RR(60),L(60),E$(2,2
    )
```

Program 4: Additional Line Changes For 8K VIC-20

```
95 MC=50                 :rem 160
100 DIM FF$(50),S$(50),W$(50),
    CC(50),RR(50),L(50),E$(2,2
    )                    :rem 25
```

Program 5: Line Changes For Apple

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
38 90 D$ = CHR$ (4):I$ = CHR$ (9
   )
4E 1720 HOME
E1 2000 PRINT D$;"PR#1": PRINT I
   $;"80N"
9E 2010 PRINT T$: PRINT
3F 2020 PRINT "   ";: FOR I = 1
   TO C0: IF I / 10 < > INT
    (I / 10) THEN PRINT " "
   ;: GOTO 2040
1A 2030 PRINT MID$ ( STR$ (I),1,
   1);
77 2040 NEXT I: PRINT
98 2050 PRINT "   ";: FOR I = 1
   TO C0: PRINT RIGHT$ ( ST
   R$ (I),1);: NEXT I: PRIN
   T
C0 2060 FOR X = 1 TO R0: IF X <
   10 THEN PRINT " ";
61 2070 PRINT STR$ (X)" ";
3A 2080 FOR Y = 1 TO C0: PRINT M
   ID$ (S$(X),Y,1);
21 2090 NEXT Y: PRINT : NEXT X:
   PRINT : PRINT : PRINT "W
   ORD LIST:"
3B 2100 FOR X = 1 TO W0: IF W$(X
   ) = "/" THEN 2120
C2 2110 PRINT W$(X)
27 2120 NEXT X: FOR I = 1 TO 5:
   PRINT : NEXT I: IF SL =
   0 THEN 2160
85 2130 PRINT "SOLUTION LIST:":
   PRINT "WORD
            ROW   COLUMN   D
   IR": FOR X = 1 TO W0: IF
    W$(X) = "/" THEN 2150
4B 2140 PRINT W$(X) LEFT$ (G$,26
    - LEN (W$(X)))RR(X) LEF
   T$ (G$,7 - LEN ( STR$ (R
   R(X))))CC(X) LEFT$ (G$,6
    - LEN ( STR$ (CC(X))))F
   F$(X)
91 2150 NEXT X
BF 2160 PRINT : PRINT D$;"PR#0":
    RETURN
```

Program 6: IBM PC/PCjr Line Changes

For instructions on entering this listing, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

```
IC 10 DEF SEG=0:POKE 1047,(PEEK(
   1047) OR 64)
JB 20 WIDTH 40:KEY OFF:DEF SEG=&
   H40:RANDOMIZE PEEK(&H6D)
ND 1720 CLS
NF 2000 ON ERROR GOTO 2170
EK 2010 OPEN "LPT1:" FOR OUTPUT
    AS #1:PRINT #1,T$:PRINT
   #1,
MH 2020 PRINT #1,"    ";:FOR I=1
    TO C0:IF I/10<>INT(I/10
   ) THEN PRINT #1," ";:GOT
   O 2040
NH 2030 PRINT #1,MID$(STR$(I),2,
   1);
KE 2040 NEXT I:PRINT #1,
AF 2050 PRINT #1,"    ";:FOR I=1
    TO C0:PRINT #1,RIGHT$(S
   TR$(I),1);:NEXT I:PRINT
   #1,
EH 2060 FOR X=1 TO R0:IF X<10 TH
   EN PRINT #1," ";
PH 2070 PRINT #1,STR$(X)" ";
```

```
2080 FOR Y=1 TO C@:PRINT #1,MID$(S$(X),Y,1);
2090 NEXT Y:PRINT #1,:NEXT X:PRINT #1
     ,:PRINT #1,"WORD LIST:"
2100 FOR X=1 TO W@:IF W$(X)="/" THEN 2120
2110 PRINT #1,W$(X)
2120 NEXT X:FOR I=1 TO 5:PRINT #1,:NEXT I:IF SL
     =@ THEN 2160
2130 PRINT #1,"SOLUTION LIST:":PRINT #1,"WORD
                          ROW   COLUMN   DIR":FOR
      X=1 TO W@:IF W$(X)="/" THEN 2150
2140 PRINT #1,W$(X);LEFT$(G$,25-LEN(W$(X)));RR(
     X);LEFT$(G$,9-LEN(STR$(RR(X))));CC(X);LEFT
     $(G$,6-LEN(STR$(CC(X))));FF$(X)
2150 NEXT X
2160 CLOSE #1:ON ERROR GOTO @:RETURN
2170 CLOSE #1:PRINT "PRINTER ERROR #";ERR;"OCCU
     RRED.":PRINT "TRY AGAIN."
2180 PRINT:PRINT "HIT A KEY TO CONTINUE"
2190 A$=INKEY$:IF A$="" THEN 2190
2200 RESUME 2010
```

Program 7: TI-99/4A Line Changes

```
80 RANDOMIZE
95 MC=50
100 DIM FF$(50),S$(50),W$(50),CC(50),RR(50),L(5
    0),E$(2,2)
180 G$=G$&" "
510 R1=INT(RND*R@)
520 C1=INT(RND*C@)
530 D1=INT(RND*8)+1
570 IF (R1+DX*L(X)<1)+(R1+DX*L(X)>R@)+(C1+DY*L(
    X)<1)THEN 590
640 IF (SEG$(W$(X),Z,1)<"A")+(SEG$(W$(X),Z,1)>"
    Z")THEN 680
670 IF (SEG$(S$(R1),C1,1)<>" ")*(SEG$(S$(R1),C1
    ,1)<>SEG$(W$(X),Z,1))THEN 590
700 IF (SEG$(W$(X),Z,1)<"A")+(SEG$(W$(X),Z,1)>"
    Z")THEN 770
710 S$(R1)=SEG$(S$(R1),1,C1-1)&SEG$(W$(X),Z,1)&
    SEG$(S$(R1),C1+1,LEN(S$(R1))-C1)
840 IF SEG$(X$,1,2)="ST" THEN 1670
850 IF SEG$(X$,1,2)="TR" THEN 500
860 IF SEG$(X$,1,2)<>"SK" THEN 830
910 IF SEG$(S$(X),Y,1)<>" " THEN 930
920 S$(X)=SEG$(S$(X),1,Y-1)&CHR$(INT(26*RND+65)
    )&SEG$(S$(X),Y+1,LEN(S$(X))-Y)
1190 IF (A$<>"Y")*(A$<>"N")THEN 1180
1550 IF (LEN(X$)<=R@)*(LEN(X$)<=C@)*(X$<>"Y")TH
     EN 1610
1690 S$(I)=SEG$(G$,1,C@)
1720 CALL CLEAR
2000 OPEN #1:"RS232"
2010 PRINT #1:T$
2020 PRINT #1
2030 PRINT #1:"(3 SPACES)";
2040 FOR I=1 TO C@
2050 IF I/10=INT(I/10)THEN 2080
2060 PRINT #1:" ";
2070 GOTO 2090
2080 PRINT #1:SEG$(STR$(I),1,1);
2090 NEXT I
2100 PRINT #1
2110 PRINT #1:"(3 SPACES)";
2120 FOR I=1 TO C@
2130 PRINT #1:SEG$(STR$(I),LEN(STR$(I)),1);
2140 NEXT I
2150 PRINT #1
2160 FOR X=1 TO R@
2170 IF X>=10 THEN 2190
2180 PRINT #1:" ";
2190 PRINT #1:STR$(X);" ";
2200 FOR Y=1 TO C@
2210 PRINT #1:SEG$(S$(X),Y,1);
2220 NEXT Y
2230 PRINT #1
2240 NEXT X
2250 PRINT #1
2260 PRINT #1
2270 PRINT #1:"WORD LIST:"
2280 FOR X=1 TO W@
2290 IF W$(X)="/" THEN 2310
2300 PRINT #1:W$(X)
2310 NEXT X
2320 FOR I=1 TO 5
2330 PRINT #1
2340 NEXT I
2350 IF SL=@ THEN 2450
2360 PRINT #1:"SOLUTION LIST:"
2370 PRINT #1:"WORD(21 SPACES)ROW(3 SPACES)COLUM
     N";
2380 PRINT #1:"(3 SPACES)DIR"
2390 FOR X=1 TO W@
2400 IF W$(X)="/" THEN 2440
2410 PRINT #1:W$(X);SEG$(G$,1,25-LEN(W$(X)));RR
     (X);
2420 PRINT #1:SEG$(G$,1,7-LEN(STR$(RR(X))));CC(
     X);SEG$(G$,1,4-LEN(STR$(CC(X))));
2430 PRINT #1:FF$(X)
2440 NEXT X
2450 CLOSE #1
2460 RETURN
```

enchanting blend of an action and adventure game. It has been designed for players aged ten to adult, but my seven-year old daughter was able to enjoy the game while playing with a grownup. It's even more enjoyable when several people join together to guide the quest. Indeed, one of the game's strong points is that it encourages cooperation rather than isolated play or deadly competition.

Colorful Graphics

One of the first things that impresses you about *Below the Root* is the quality of the screen graphics—the color and detail rival that of any arcade game. There are more than 100 different screens, each a delight to the eye.

Unlike text adventures, *Below the Root* doesn't require you to enter your commands by typing short sentences such as "Look North" or "Take Object." Instead, you select functions from various menus of choices (with the joystick, if you're using one). This makes the game more suitable for younger children. For example, the main menu lets you start a new game, save a current game on disk, continue a previously saved game, or view a sample game simply by indicating your choice. The last option, by the way, is particularly recommended for first-time players—it's wise to take a few minutes to orient yourself before plunging headlong into this unknown world.

After reading the well-written instructions and viewing the sample game, you're ready to start. First, the program asks which of five questers you wish to adopt. Each comes with varying degrees of stamina and "spirit skill." Questers also represent the two races which occupy Green Sky: the tree-loving Kindar and their cousins, the Erdling. Each race has its own attributes and limitations. All the questers, however, can grow in strength and spirit as they progress through the game.

What really sets this game apart is that questers can be either male or female. My daughter thought it was unfair that she was limited to choosing between three male characters and only two female characters, but still, at a time when computers are becoming increasingly important, it's gratifying to find a game that goes out of its way to encourage young girls as well as boys.

The level of each quester's spirit skill is an important factor in mastering the environment of Green Sky and successfully completing the quest. Spirit skills include the ability to read the emotions and thoughts of others (*pensing*), to heal yourself if injured, to influence tree growth (*grunspreke*), or to move yourself or other objects with your mind (*kiniport*). Each requires higher levels of spirit skill, and it's up to the player to determine how to raise this level. Those new to Green Sky should select questers with more spirit skill, while those who have played before may want to try questers with less spirit skill for a more challenging game.

Once you've selected your quester, the game begins in the quester's home. At this point, you have 50 days (in game time) to complete your quest and save Green Sky. Initial supplies are available in the quester's home, and players decide their course of action by making selections from the options menu. Many of these options are familiar to those who have played text adventures. You can examine, take, buy, eat, offer, drop, or sell various objects. You can also list an inventory of what you're carrying and call upon your spirit skills.

Quester, Heal Thyself

Questers are free to move throughout Green Sky in various ways: They can walk, run, jump, glide, climb, crawl, or enter and exit buildings. Since much of the action occurs in the treetops of Green Sky, you must be careful not to fall—unless you have a *shuba* for gliding, your quester will suffer a bump on the head. But watching the comical way in which questers rub their heads after a fall may help soothe the pain.

When you first encounter other characters in the game, an important spirit skill to use is pensing. This allows you to determine if they're friendly before speaking to them. This is vital, because some inhabitants are hostile. From time to time, it's also important to check your status, get adequate rest, eat when you're hungry, and heal yourself of any injuries. If your situation becomes too desperate, you may have to *renew* yourself. This option returns you home, but costs you a day from your quest.

The renew option, incidentally, spotlights another attractive feature of *Below the Root*: Questers are never killed or destroyed during their quest. While the world may be lost, violence rarely befalls the quester. This may be an important consideration for young players who would become upset if a character they created was destroyed during a game, or for parents who are disturbed by violence in computer games.

Below the Root
Windham Classics/Spinnaker Software
One Kendall Square
Cambridge, MA 02139
$26.95

Companion

Roger B. Crampton

Requirements: TI-99/4A with 32K RAM expansion card or box, Extended BASIC, a disk drive, and a printer.

Until I saw *Companion*, I considered replacing my TI-99/4A with a much more expensive computer for my serious word processing needs. I had tried several other word processors and found them either too slow, too cumbersome, or lacking essential features. But *Companion*, an inexpensive program written entirely in machine language, solves all of those problems.

Companion's editing features are superb—you have instantaneous full-screen editing capability. And the editing comes naturally, because all normal features of the TI keyboard retain their functions. For example, pressing Function 2 (Insert) works the same way with *Companion* as it does when you're entering a program in console or Extended BASIC. There are no surprises or tricky key sequences with *Companion*. Everything is logical and works in much the same manner as screen editing in BASIC. A delightful exception is the up- and down-arrow keys—they really move the cursor up and down, the way you wish they did in BASIC.

Of course, *Companion* has all of the usual word processing features. You can center headings, set tabs, automatically indent new paragraphs, search for text strings, and move or copy blocks of text. And you don't have to memorize a complex series of keystrokes to do simple things. For instance, pressing CTRL-P automatically generates a linefeed, a carriage return, and indents five spaces for the next paragraph.

The manual is well-written, succinct, and most important, understandable. At 142 pages, it may seem intimidating at first, but there is a good reason for its length. *Companion* has so many features that it takes that many pages to describe them.

Companion works flexibly with different kinds of printers. It lets you send control characters so you can switch to compressed or expanded fonts, or any other fonts allowed by your printer. A little judicious study of your printer manual, along with the *Companion* manual, should enable you to produce a brief list of control characters to adjust nearly any printer parameter.

Companion
Intelpro
5825 Baillargeon Street
Brossard, Quebec
Canada J4Z 1T1
$79.95

the IF test is true, and skipped when the test is false. Consequently, in COPYUNQ.BAT, the ECHO command (which prints "*filename* WILL NOT BE COPIED") executes only when the file in question exists on both the source and target disks.

Once you understand that much of COPYUNQ.BAT, the rest is not hard to decipher. PAUSE makes the system stop and display the message "Strike any key when ready." This is the only batch command that allows user input. Unfortunately, your choices are severely limited: You can continue only by pressing a key (perhaps after changing disks, etc.) or end the program by pressing Ctrl-Break. In Part 2 of this article, we'll show how to expand this number of options.

NOT And ERRORLEVEL

The second FOR line in COPYUNQ .BAT has a FOR loop and an IF test very similar to the first. However, in this case NOT reverses the logic of the IF test. When the named file *does not exist* on the target disk, the IF test is true and the file is copied.

In addition to testing EXIST (with or without NOT), IF can test two conditions: the equality symbol (==) and ERRORLEVEL. The equality symbol tests whether two strings are identical. ERRORLEVEL is always a number, ordinarily used to pass information from one program to another (indicating whether the first worked successfully and thus set ERRORLEVEL to the expected value). ERRORLEVEL is discussed further in Part 2.

As shown in these brief examples, batch programs can be very powerful: IF lets you pick only the files you want, and FOR lets you repeat commands until the whole task is done. In one sense, the lack of opportunity for user input is an advantage: The entire procedure is automated, and you don't need to understand anything except how to type in the program name. On the other hand, batch programming can seem rigid, limiting, and visually quite dull. Part 2 improves on that situation, offering program examples and a routine that adds colorful graphic displays and multiple-option menu selection to batch programs. ©

News & Products

Commodore Memory Expansion, Interface

Cardco, Inc., has announced *S'more* (Super Memory Optimized RAM/ROM Expansion), a cartridge utility for the 64 which allows more than 60K RAM for programming and adds over 60 new and enhanced BASIC commands and functions. The memory increase is not restricted, and can be used for arrays, variables, and BASIC programs which would normally overload a Commodore 64. *S'more* provides such programming aids as CATALOG (view disk directory), AUTO (line numbering), FIND, CHANGE, TRACE, DUMP, KEY (define function keys), and others.

Function keys are preprogrammed, but can be redefined. For example, F2 runs the current program in memory, F3 reads and displays the disk drive error channel, and F7 displays the current disk directory. The suggested retail price is $69.95. Cardco also plans to introduce the *S'more BASIC Compiler* for $39.95.

Also recently introduced is G Whiz, an improved version of Cardco's +G printer interface, which allows Commodore computers to be hooked up to virtually any Centronics printer. Additional features include faster printing speed (up to 18 times faster with many dot matrix printers), and increased speed on high-resolution screen dumps. The interface also comes with two character sets and open access to DIP switches. The interface attaches directly to the parallel port, eliminating the ribbon connector. Suggested retail price is $69.95.

Cardco, Inc., 300 S. Topeka, Wichita, KS 67202

Circle Reader Service Number 232.

IBM, ST Expert Investment Help

Batteries Included has introduced the first product in its Integral Solutions line of productivity software. The *Isgur Portfolio System* was designed by Lee Isgur, a well-known Wall Street analyst and first vice president of Paine-Webber, Inc. The program allows both casual and professional investors to track up to ten portfolios, each with 50 stocks and 15 separate holdings. With a ten-megabyte hard disk, storage capacity jumps to 1,000 portfolios, with more than 2,000 stocks and 600 holdings of each.

Special tracking and advisory features help determine how and when to raise money, when to sell holdings, and how to prepare for changes in the status of holdings. Built-in telecommunications functions put the user online with major telecommunications services at the touch of a key or two.

The *Isgur Portfolio System* is available for the Atari 520 ST and IBM PC for $249.95.

Batteries Included, 30 Mural St., Richmond Hill, Ontario, Canada L4B 1B5

Circle Reader Service Number 233.

Home Control Package

The X-10 Powerhouse interface is a freestanding controller for lights, heating, cooling, security devices, and other appliances, which you preset with your computer by following simple software-driven onscreen icons representing controllers for each room of your home or business. Available initially for the Apple II series, the system is scheduled to be available for the Commodore 64/128 in September and the IBM PC/PCjr in October.

The Powerhouse lets you control up to 72 lights and appliances plugged into System X-10 modules, which in turn are plugged into your home's electrical outlets. To program the Powerhouse interface, you use a joystick to graphically "install" lights and appliances in each room in positions which correspond to the actual locations in your own home. Once programmed with your computer, the system operates independently. X-10 modules can be purchased at electronics stores. The Powerhouse interface sells for approximately $125, while the appropriate software and connecting cable retails for an additional $25.

X-10 (USA), Inc., 185A LeGrand Avenue, Northvale, NJ 07647

Circle Reader Service Number 234.

PlayWriter Series Expands

Woodbury Computer Associates, Inc.,

Programming the TI

C. Regena

The OPEN Statement

Recently I received a call from a young programmer who wanted to know more about the OPEN statement. I really couldn't give him an adequate answer over the phone ("look at your manuals"), so I'll give several examples here.

The OPEN statement means about the same thing in all versions of BASIC, but each computer has its own variations. As the statement implies, the function of OPEN is to open a *file*—or, as I like to think of it, to get the attention of another device to be used with the main console. Various forms of the OPEN statement are described in the manuals that come with the peripherals.

OPEN statements are generally followed by the number of the device you want to address. In TI BASIC, you may use any constant or variable with a value of 1 to 255 for the device number. The number is preceded by the # sign, such as OPEN #1: to open file #1.

Whenever you use an OPEN statement, it is good programming practice to include a CLOSE statement when you're finished with the device. If your program stops with an error, the files are automatically closed.

Speech Synthesis

If you have the TI Speech Synthesizer and the *Terminal Emulator II* command module, use an OPEN statement to make the computer talk:

```
OPEN #1:"SPEECH","OUTPUT
```

This alerts the speech device to be ready for output. Then all you need is a PRINT #1 statement (pronounced "print file one"):

```
PRINT #1:"HELLO"
```

Within a program, you can print on the screen with a regular PRINT statement and produce speech with the PRINT # statement:

```
10 OPEN #5:"SPEECH","OUTPUT
20 PRINT "THIS IS A TEST."
30 PRINT #5:"THIS IS A TEST."
40 CLOSE #5
```

By the way, if you'd like to hear your program listing, use the command LIST "SPEECH."

Printing

To get the most out of a printer, you really need to study your printer and interface manuals. The Texas Instruments RS-232 interface manual shows all the different parameters for accessing your printer. Here are some examples of OPEN statements:

```
OPEN #1:"TP"
OPEN #1:"PIO"
OPEN #1:"RS232.BA=600"
OPEN #1:"RS232.TW.BA=110"
```

Once you've determined the necessary OPEN statement for your hardware configuration, you can use PRINT #1 (or whatever file number you opened) to send any command to the printer. If someone else wants to modify your program for another configuration, they can simply change the OPEN statement for their setup.

PRINT # lets you print constants, variables, and strings. You can align columns with the TAB function. In Extended BASIC, the PRINT #1, USING statement also is handy to format the output. Here's a short example of sending output to the printer:

```
10 OPEN #1:"RS232.BA=600"
20 PRINT #1:TAB(10);"THIS SHOULD
   PRINT."
30 CLOSE #1
```

File Processing

If you want to learn more about file processing with the OPEN statement, the manual that comes with the TI-99/4A contains a good description of various forms of OPEN. I also discussed file processing in my COMPUTE! columns of March, April, and May 1984. And a program which saves names and addresses on cassette is in my book, *Programmer's Reference Guide to the TI-99/4A.*

This month's example program shows how to use the OPEN statement to save a drawing on cassette. Type in and run the program, then press the arrow keys to draw a low-resolution picture on the screen. When you're done, press CTRL-S to save the picture on tape. You can load it by pressing CTRL-L.

The program uses different character numbers for the different-colored drawing squares. These are defined in lines 140–200. When the program loads a picture, it uses the character numbers to determine the locations of the colored squares.

Lines 540–870 contain the drawing procedure. The variable X is the row and Y is the column. C is the character number. If you press the space bar, C is incremented by 4 and the color of the square changes. The arrow keys move the square, and it stops at each screen edge.

Lines 890–990 keep track of the character numbers for each column in each row if you want to save the picture. Lines 1000–1050 save the strings of G$, which contain the character numbers on cassette. The procedure takes quite a while because each item saved has its own leader. You can hear the cassette recording during this process. The OPEN statement in line 1000 opens device #1 as "CS1," or cassette, for OUTPUT. INTERNAL and FIXED are two options available in the OPEN statement for cassette that specify how to save the data. FIXED 96 is used because each G$ will be 96 characters long.

Lines 1150–1210 load the picture from cassette. Notice how the OPEN statement in line 1160 matches the format of line 1000, except that it specifies INPUT instead of OUTPUT. The INPUT #2 statement reads G$ row by row.

Input variables must match the way they were previously saved, although you can use different variable names. Lines 1230–1320 recreate the picture on the screen from the information read off tape.

If you'd like to save typing effort, you can obtain a copy of this program by sending a blank cassette or disk, a stamped, self-addressed mailer, and $3 to:

C. Regena
P.O. Box 1502
Cedar City, UT 84720

Doodle With CS1

```
100 REM  DOODLE WITH CS1
110 DIM B$(24)
120 CALL CLEAR
130 PRINT TAB(11);"DOODLE":
    :::
140 FOR C=10 TO 16
150 D=C*8+24
160 CALL CHAR(D,"")
170 CALL CHAR(D+4,"FFFFFFFF
    FFFFFFFF")
180 CALL COLOR(C,C,C-7)
190 NEXT C
200 CALL COLOR(10,2,3)
210 PRINT "CHOOSE:"
220 PRINT :"1 DRAW"
230 PRINT :"2 LOAD PICTURE"
    :::
240 CALL KEY(0,K,S)
250 IF K=50 THEN 1160
260 IF K<>49 THEN 240
270 REM
280 CALL CLEAR
290 PRINT "PRESS SPACE BAR
    TO CHANGE"
300 PRINT "SCREEN COLOR."
310 PRINT :"PRESS <ENTER> F
    OR DESIRED(3 SPACES)COL
    OR."
320 SC=3
330 CALL SCREEN(SC)
340 CALL SOUND(100,1497,2)
350 CALL KEY(0,K,S)
360 IF K=13 THEN 420
370 IF K<>32 THEN 350
380 SC=SC+1
390 IF SC=16 THEN 300
400 IF SC=17 THEN 320 ELSE
    330
410 REM
420 CALL CLEAR
430 PRINT "MOVE ARROW KEYS
    TO DRAW."
440 PRINT :"PRESS SPACE BAR
     TO CHANGE(3 SPACES)COL
    ORS."
450 PRINT :"PRESS CTRL S TO
     SAVE."
460 PRINT :"PRESS CTRL L TO
     LOAD."
470 PRINT :"PRESS CTRL E TO
     END."
480 PRINT ::"NOW PRESS ANY
    KEY TO START."
490 X=12
500 Y=14
510 C=104
520 CALL KEY(0,K,S)
530 IF S<1 THEN 520
540 REM  DRAW
550 CALL CLEAR
560 CALL SCREEN(SC)
570 CALL KEY(0,K,S)
580 CALL HCHAR(X,Y,32)
590 CALL HCHAR(X,Y,C)
600 IF K=147 THEN 890
610 IF K=140 THEN 1160
620 IF K=133 THEN 1350
630 IF K<>32 THEN 680
640 C=C+4
650 IF C<>160 THEN 570
660 C=104
670 GOTO 570
680 IF K<>69 THEN 730
690 X=X-1
700 IF X>0 THEN 570
710 X=1
720 GOTO 570
730 IF K<>83 THEN 780
740 Y=Y-1
750 IF Y>0 THEN 570
760 Y=1
770 GOTO 570
780 IF K<>68 THEN 830
790 Y=Y+1
800 IF Y<33 THEN 570
810 Y=32
820 GOTO 570
830 IF K<>88 THEN 570
840 X=X+1
850 IF X<24 THEN 570
860 X=24
870 GOTO 570
880 REM  SAVE
890 CALL SOUND(150,1200,2)
900 FOR ROW=1 TO 24
910 B$(ROW)=""
920 FOR COL=1 TO 32
930 CALL GCHAR(ROW,COL,G)
940 IF G<>32 THEN 960
950 G=200
960 B$(ROW)=B$(ROW)&STR$(G)
970 NEXT COL
980 CALL SOUND(30,1200,2)
990 NEXT ROW
1000 OPEN #1:"CS1",OUTPUT,I
     NTERNAL,FIXED 96
1010 FOR ROW=1 TO 24
1020 PRINT #1:B$(ROW)
1030 NEXT ROW
1040 PRINT #1:X,Y,C,SC
1050 CLOSE #1
1060 PRINT ::"CHOOSE:"
1070 PRINT :"1 GO BACK TO S
     AME DRAWING"
1080 PRINT :"2 START NEW DR
     AWING"
1090 PRINT :"3 SAVE ANOTHER
      COPY"
1100 PRINT :"4 LOAD PICTURE
     "
1110 PRINT :"5 END"
1120 CALL KEY(0,K,S)
1130 IF (K<49)+(K>53)THEN 1
     120
1140 ON K-48 GOTO 1230,280,
     1000,1160,1350
1150 REM  LOAD
1160 OPEN #2:"CS1",INPUT ,I
     NTERNAL,FIXED 96
1170 FOR ROW=1 TO 24
1180 INPUT #2:B$(ROW)
1190 NEXT ROW
1200 INPUT #2:X,Y,C,SC
1210 CLOSE #2
1220 REM
1230 CALL CLEAR
1240 CALL SCREEN(SC)
1250 FOR ROW=1 TO 24
1260 FOR COL=1 TO 32
1270 G=VAL(SEG$(B$(ROW),COL
     *3 2,3))
1280 IF G<>200 THEN 1300
1290 G=32
1300 CALL HCHAR(ROW,COL,G)
1310 NEXT COL
1320 NEXT ROW
1330 GOTO 570
1340 REM
1350 CALL CLEAR
1360 END
```

Jump Search

Jerry Sturdivant

Learn how the binary search method can speed up data handling. The short demonstration program listed below runs on the Atari 400/800, XL, and XE series; Apple II-series; IBM PC/PCjr; all Commodore computers; TI-99/4A; the Radio Shack Color Computer; and other personal computers with BASIC.

Searching for a specific item in a collection of data is a fundamental computing task. Word processors, databases, and address book programs all need to locate data quickly and accurately. This article shows how to use the simple binary search method in BASIC programs for efficient data handling.

For a demonstration, type in, save, and run "Jump Search" below. Program 1 is a general version for Commodore, IBM, Apple, and the TRS-80 Color Computer. For the Atari, make the line changes listed in Program 2. For the TI-99/4A, one small change is needed to use Program 1. TI BASIC does not allow variables as arguments in DIM statements, so line 110 should be replaced with the following:

```
110 DIM S$(10), PP(10)
```

If you have another computer not mentioned above, use Program 1; it should run with little or no modification.

The demo program creates a list of ten city names in alphabetical order, with population figures for each city (of course, an actual program would contain much more data). Lines 100–140 store the city names in a string array and the population figures in a matching numeric array. (On the Atari, the string array is simulated by manipulating substrings within a single string variable, since there are no true string arrays in Atari BASIC.) Once this is done, you can find the population of any city in the list by searching for its name. For example, if your search finds that AKRON is stored in array element S$(2), then the population for Akron can be found in the numeric array element PP(2).

The city names are stored in the array in alphabetical order because *this search technique works only on data that has been arranged in alphabetical or numeric order.* If you consider the situation for a moment, you'll realize that no organized searching method can speed up the hunt for a particular item in a randomly arranged set of data. If you can't tell whether a word you've found should come before or after the word you're looking for, then you'll have to examine every word in the list until you find an exact match. Arranging the data into alphabetical or numeric order, called *sorting*, is a separate problem and has been considered in previous articles. Just remember that only ordered data can be searched efficiently.

The simplest way to find a word in an alphabetical list is to start at the A's and hunt forward through the alphabet until you find a match. A sequential search of this type is very easy to program (all you need is a FOR-NEXT loop), but it's also slow and inefficient. When the target word is toward the end of the alphabet, sequential searching wastes a lot of time looking through all the preceding words.

Jump To The Center

The binary search method (called *binary* because it repeatedly divides the data list in half) is much faster. Rather than starting at the beginning of the alphabet, it jumps in at the center. Let's look at the example program to see how this works.

The variable B stands for the beginning of the word list, E stands for the end, and C represents the center. Say that your target word is ATLANTA. When the search begins, line 200 finds the center of the ten-word list and jumps to that position (in this case finding the sixth word, ANAHEIM). Since ANAHEIM doesn't match ATLANTA, the program skips to line 250 for a critical test.

At this point the database is divided into two blocks, lower and higher. The program first decides which block holds the target word, then jumps to the center of that block to continue the search. Since ATLANTA comes after ANAHEIM in the alphabet, it must be stored in the higher block of words. Note that in just one step, you've eliminated the need to look at anything in the first half of the database. A sequential search (which compares ATLANTA to ABILENE, then to AKRON, then to ALBANY, etc.) takes six steps to accomplish the same result.

Now it's time for the second jump. Lines 260–270 set a new beginning point just above the center (B = C + 1) and go back to line 200. The program finds the center of the new list (which consists of four words, ANCHORAGE to AUSTIN) and jumps to that position. This time the target word matches the found word. While the binary method found the target word with only two comparisons, a sequential search would require nine (eight comparisons to eliminate ABILENE through ATHENS, and a ninth to confirm ATLANTA).

The more data you have, the more time the binary method saves. For instance, if the list contains 1,000 words, most words are found in about eight comparisons (the sequential method usually requires hundreds). If you expand the list to 10,000 words, only about twelve

comparisons are required (compared to thousands for the sequential method). The secret lies in the halving technique. By repeatedly chopping the list in half, this method quickly eliminates large chunks of data from consideration and zeros in on the target. Of course, you're not limited to string data. With slight modifications this routine can search numeric data as well

For instructions on entering these listings, please refer to "COMPUTE!'s Guide to Typing In Programs" published bimonthly in COMPUTE!.

Program 1: Jump Search (General Version)

```
100 N=10
110 DIM S$(N),PP(N)
120 FOR I=1 TO N
130 READ S$(I),PP(I)
140 NEXT I
150 E=N
160 B=1
170 P=0
180 PRINT "ENTER CITY"
190 INPUT C$
200 C=INT((E+1-B)/2)+B
210 IF E-B<3 THEN 300
220 IF C$<>S$(C) THEN 250
230 P=C
240 GOTO 340
250 IF C$<S$(C) THEN 280
260 B=C+1
270 GOTO 200
280 E=C-1
290 GOTO 200
300 FOR I=B TO E
310 IF C$<>S$(I) THEN 330
320 P=I
330 NEXT I
340 IF P<>0 THEN 370
350 PRINT "DATA NOT FOUND."
360 GOTO 150
370 PRINT S$(P),PP(P)
380 GOTO 150
999 REM CITY & POPULATION DATA
1000 DATA ABILENE,89000
1010 DATA AKRON,237000
1020 DATA ALBANY,250000
1030 DATA ALBUQUERQUE,332000
1040 DATA ALVERINA,29000
1050 DATA ANAHEIM,219000
1060 DATA ANCHORAGE,174500
1070 DATA ATHENS,150000
1080 DATA ATLANTA,425000
1090 DATA AUSTIN,346000
```

Program 2: Atari Line Changes

```
110 DIM C$(15),S$(N*15),P
    P(N):S$=" ":S$(N*15)=
    S$:S$(2)=S$
130 READ C$,A:S$((I-1)*15
    +1,I*15)=C$:PP(I)=A
190 INPUT C$:L=LEN(C$)
220 IF C$<>S$((C-1)*15+1,
    (C-1)*15+L) THEN 250
250 IF C$<S$((C-1)*15+1,(
    C-1)*15+L) THEN 280
310 IF C$<>S$((I-1)*15+1,
    (I-1)*15+L) THEN 330
370 PRINT S$((P-1)*15+1,P
    *15),PP(P)
```

128 Sound And Music

Part 2

Philip I. Nolson
Assistant Editor

The second installment of this two-part article explores the Commodore 128's FILTER, SOUND, and PLAY commands and includes three short demonstration programs.

In Part 1 (COMPUTE!, August 1985), we discussed the Commodore 128's VOL, TEMPO, and ENVELOPE commands as well as the basics of sound envelopes and waveforms. This month we'll examine the three remaining sound commands: FILTER, SOUND, and PLAY. Since your *128 User's Guide* explains the fundamentals, we'll focus on less obvious features and note how these complex commands interact with one another.

FILTER Needs PLAY

Like the ENVELOPE command (see Part 1), FILTER does nothing noticeable until you turn the filter on with a PLAY statement. Insert X1 inside the PLAY string wherever you want to turn the filter on, and X0 where you want to turn it off. If you leave out the X parameter, PLAY ignores preceding FILTER commands (the filter remains off). In the simplest case (a FILTER command followed by PLAY"X1"), the filter affects all three voices. However, you can also filter each voice individually:

```
FILTER 1000,1,0,0,15
PLAY "V1 X1 V2 X0 V3 X0
```

These statements turn the low-pass filter on for voice 1 and turn it off for voices 2 and 3. The 128 remembers which voice to filter when it executes subsequent PLAY statements (more about multivoice music is explained below). However, you can use only one filter *setting* at a time. For instance, you can't use a low-pass filter for voice 1 and a band-pass filter for voice 2. Whenever X1 appears in a PLAY string, the 128 uses the most recent FILTER setting. If no FILTER command has been executed, this may result in silence.

A FILTER Editor

As with other sound effects, the best way to learn is to listen and experiment; Program 1 below, "128 FILTER Editor," lets you do just that. It's self-prompting, so you need only type it in, save a copy, and run it. The menu screen displays all the current filter parameters and lets you change whatever you like. To select any option, press a number key from 0 to 9 and follow the prompts. The program begins with no filtering (all filters off) for comparison.

Option 9 switches you to the display screen, plays an ascending musical scale with whatever filter-

Classified

COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

Rates: $25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add $15 per line for boldface words, or $50 for the entire ad set in boldface (any number of lines.)

Terms: Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted.

Form: Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

General Information: Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

Closing: 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

Notice: COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

SOFTWARE

LOTTO PICKER. Go for Million Dollar Jackpots! Picks all USA Lotto games ! ! PRO FOOTBALL ANALYST. Beat the points consistently and easily! They pay for themselves! Each $35. IBM-PC/TI99 ver. RIDGE, 170 Broadway, #201-C, NYC, NY 10038. 718-833-6335. Free Catalog avail.

SPANISH GERMAN & FRENCH. 1000 drills. Extensive vocab/exciting games. Allows 600 user exercises. Apple II+/e/c 64K. Specify language. $24.95 + $1 Shipping. AET, P.O. Box 6591, Bridgewater, NJ 08807.

FIXED ASSET DEPRECIATION on Commodore 64. Handles acrs and pre-acrs methods. $39.95. For free info write. MPM Software Prods., P.O. Box 3522, Glendale, CA 91201.

FOOTBALL PREDICTION PROGRAM for C64 & IBM. Impress friends with picks calculated from newspaper stats, incl. turnovers, injuries, etc. Pkg. includes 1984 year-end Pro stats. Send $25.95 to: JZL Systems, 201 Brookside Ln, Somerville, NJ 08876

FREE EDUCATIONAL SOFTWARE CATALOG. 112 pp, color pictures, 1000+ programs for Com, Apple, IBM, Atari. Send name + address to: Interstate Software, P.O. Box 8952, Boise, ID 83707, (208) 342-3347

WORD WARS!—A crossword game for 1 or 2. Beats Monty a mile! Send for FREE how-to bklet. C64 cass. $24.95. No-hassle guar. Rollingwood, POB 364, Richmond, CA 94806

TI-99/4A Software/Hardware bargains. Hard to find items. Huge selection. Fast Service. Free Catalog. D.E.C., Box 690, Hicksville, NY 11801

SUM COINS C64 COIN EDU. SOFTWARE. GRADES. K-3. Number & type of coin options. Speed Drills, & number of problems. $12.95 cass. $14.95 disc. HIS KIDS SOFTWARE, 2895S. Mudd Creek Rd., Oak Harbor, Ohio 43449

WAR GAMES DIALER-ATARI 835/1030 MODEMS. Send $11.95 to MK Software 39 Eaton St., Buffalo, NY 14209 Personal checks take 2–4 wks. to clear.

ATARI USERS. 4 games on cassette, $25. Saucers + monsters, poker, seafinder, pinball. Write or call Keith Anderson, 8435 12th Ave., Silver Spring, MD 20903 (301) 434-0285

AMCUP, AN ORIGINAL sailboat racing game. 1 or 2 players. C-64 w/1541 + joystick. $14.95 ch/m.o. SASE for info. Dream Dog Software, P.O. Box 1724, Eugene, OR 97440.

"CRAPSMAN!" The C-64 casino craps tutor! Learn all bets. Test systems. Up to four players. $31.95 pp, disk. EMCEE SOFT CORP., P.O. Box 402124, Miami Beach, FL 33140.

SHOPPING LIST PROGRAM for C-64 with 1541 disk and printer. It prepares a shopping list for meals you select, organized by your market sections. Send $42 to 4055 Woodman Ave., Sherman Oaks, CA 91423

SOFTWARE PROTECTOR 64. Copy and list protect your valuable Basic programs. Only $10 (disk only). Systems Software. 44 Rte 25A #209, Smithtown, NY 11787.

PINOCHLE for Commodore 64, PC and PCjr. Real, double-deck, partnership pinochle gives you a partner and two opponents! On disk, $25 postpaid. Jim Bernard, 8018 Sunset Path Ct., Springfield, VA 22153.

TI CHECKWRITER Letter code and dollar amount are all that's necessary to print a complete check. Provides yearly totals for taxes. Disk $24.95. R.I.P. Software, 248 Purisima Rd., Woodside CA 94062.

SCIENCE SOFTWARE with graphics for the Commodore 64. Astronomy programs, etc. Catalog and demo disk $3. David Eagle, 7952 W. Quarto Dr., Littleton, CO 80123.

HARDWARE

Trade in your used Commodore or Atari on a brand new C-128 or Atari ST. This offer may not be available through retail outlets. Brochure $1.50 and SASE. NEW WEST TECHNOLOGY, 4B Monroe Pkw., Box 200, Ste. 134, Lake Oswego, OR 97034

T.I. 99/4A: Full memory expansion for your T.I. Computer! New memory added entirely within the computer console! No plugs or external power needed. Write: Tim Northrup, 3305 Rutherglen, El Paso, Texas 79925

REPLACEMENT POWER SUPPLIES FOR C-64 or VIC-20. $29 ea. plus $3 shipping. 64 parts list $5-refundable w/1st order. AA Computer, 2726 Park St., JAX, FL 32205.

NUMERIC KEYPAD with joystick CURSOR control, gold contact $75. Centronics interface $40. Autodos cartridge automatically loads programs from easy-to-use menu $40. All C-64 fully compatible. SS 64 RF Mod. $40. Battery Back-up with best available protection for C-64 $50, Apple/IBM $150 (U.S.) APPLIED COMPUTING, Box 1566, PTBO, ONT. K9J 7H7. Send $2 for cat. Call (705) 745-8617 anytime.

MISCELLANEOUS

*YOU NEED A MAG-RAK! Ends awkwd fumbling when entering programs. Holds & lineguides an entire magazine! $34.95 V/Mstcd/Ck/MO—Stop suffering—Order yours today. (213) 662-6905, R&L, Bx 26A89, LA, CA 90026

Maxell MD1, $1.39 - MDZ, $1.99. Dysan 104/1D, $1.79 - 104/2D, $2.39. Shipping $3.75. Also Verbatim, IBM, 3M, BASF. TAPE WORLD, 220 Spring St., Butler, PA 16001, 1-800-245-6000, Visa, MC.

SELL YOUR COMPUTER National Listing Service: $19.95. Listed until sold. No commission fees. All types of computer equipment/peripherals. 1-800-TEC TRAN Visa/MC accepted. In MA (617) 491-4888.

FREE! SHOP BY MODEM in our revolutionary electronic shopping mall. We even accept credit cards! We have products YOU want! Call 1-818-840-8066 with your modem NOW!

FLYING COMPUTERS but stuck on the ground? Learn requirements to fly aircraft, with or without license. $3.00 to: Aviation Resource, P.O. Box 369-C, Lancaster, NY 14086.

Super 99 Monthly—quality info for TI-99/4A & compatibles since 09/84. US $12/yr. Other $16/yr. Bytemaster Computer Svces., 171 Mustang, Sulphur LA 70663.

BUY-SELL-TRADE computers, hardware, software. Announce clubs, BBS's, etc. Send for free sample of Micro-Swap, the computer classifieds. P.O. Box 24, Esmond, IL 60129.

COMPUTER OWNERS!!! Earn dollars and save money with your computer. Write C&D Associates, Box 851, Mt. Prospect, IL 60056. Specify computer type. No Investment!!

TI-99/4A Software/Hardware bargains. Hard to find items. Huge selection. Fast Service. Free Catalog. D.E.C., Box 690, Hicksville, NY 11801

HELP IS ON THE WAY! Just call 1-800-334-0868 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

SAFEGUARD YOUR COMPUTER KEYBOARD from contaminants with OVERBYTE molded keyboard covers. $24.95 xparent & durable. Call (213) 866-2583/send $2 further info. OVERBYTE, P.O. Box 10652, Burbank, CA 91510.

PHONEMARK DATASETTES: C64, Plus4, Vic-20. $25.00 plus shipping. We are direct importers. American Mercantile Co., 2450 1st Ave., South, Seattle, WA 98134 (206) 624-6141.

DISK SERVICE MANUAL. Service floppies without special software or equipment. FREE information. Consumertronics-DSM, P.O. Drawer 537, Alamogordo, NM 88310.

$$ MONEY MAKER $$ Increase your income by hundreds and even thousands of dollars. Money back guarantee! (Apple, Macintosh, IBM) Free brochure: 1-800-223-5838 (305-771-6498)