On-Line Shopping: Today's Computer Catalogs

COMPUTE!

$2.95
November
1984
Issue 54
Vol. 6, No. 11
$3.75 Canada
02193
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

Reflection

One- Or Two-Player Strategy Game For Atari, Commodore 64, VIC-20, Apple, PC/PCjr, TRS-80 Color Computer, & TI-99/4A

The Bulletin Boarding Of America

Plus: Understanding Modems & Bulletin Board Basics

FOR-NEXT Loop Etiquette

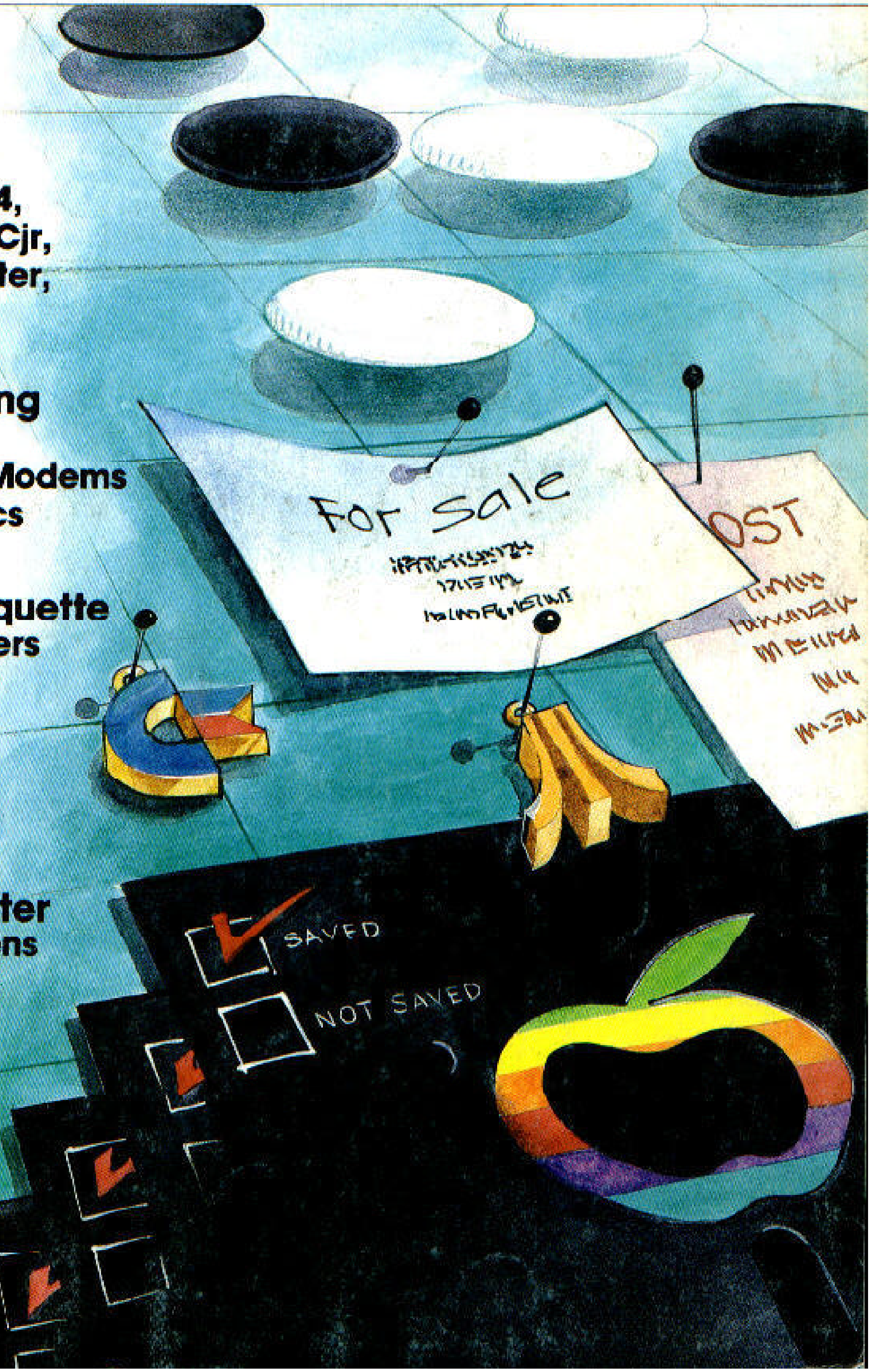
For BASIC Programmers

Apple Disk Verify

Is Your Program Really Saved?

IBM Screen Formatter

Auto-Adjusting Screens For 40 & 80 Columns



November 1984 Vol. 6, No. 11

GUIDE TO ARTICLES AND PROGRAMS

FEATURES

30 On-Line Shopping: Today's Computer Catalogs . . . . Selby Bateman •
32 Understanding Modems . . . . Sharon Darling •
46 The Bulletin Boarding Of America . . . . Kathy Yakal •
50 Bulletin Board Basics . . . . Gregg Peele •

EDUCATION AND RECREATION

58 Reflection . . . . Sean Puckett AT/64/V/TI/AP/PC/PCjr/CC
85 Spiders . . . . Joe Rocke V/64/AP/PC/PCjr

REVIEWS

120 KoalaPad For PCjr . . . . Lee Noel PCjr
127 *OmniWriter* & *OmniSpell* . . . . Joseph R. Sutton 64
131 *WizType* . . . . James V. Trunzo AT/64/AP

COLUMNS AND DEPARTMENTS

6 The Editor's Notes . . . . Robert Lock •
10 Readers' Feedback . . . . The Editors and Readers of COMPUTE! •
108 The Beginner's Page . . . . Tom R. Halfhill •
112 On The Road With Fred D'Ignazio: More Ways Computers Made Me Smarter After Only Thirteen Years Of Daily Use . . . . Fred D'Ignazio •
132 Computers And Society . . . . David D. Thornburg •
151 INSIGHT: Atari . . . . Bill Wilkinson AT
155 Machine Language: Stack Tricks . . . . Jim Butterfield •
157 Programming The TI: Algebra Tutorial, Part 2 . . . . C. Regena TI
160 IBM Personal Computing . . . . Donald B. Trivette PC/PCjr

THE JOURNAL

100 PC Monochrome Graphics . . . . Michael A. Covington PC/PCjr
102 Update On COMAL: A SuperBASIC . . . . Jim Butterfield 64/V/P
134 All About The Status Register, Part 2 . . . . Louis F. Sander 64/P/AP/AT
163 Enhanced Commodore 64 DOS Support . . . . Stephen S. Melsheimer 64
171 IBM Screen Formatter . . . . David Leithauser PC/PCjr
172 Apple Disk Verify . . . . Ilan Reuben AP
174 Commodore Potpourri . . . . COMPUTE! Readers and Todd Heimarck 64/V/P
177 Atari Easy Scroll . . . . Eugene D. McMillin AT
179 FOR-NEXT Loop Etiquette . . . . Jim Butterfield •

182 CAPUTE! Modifications Or Corrections To Previous Articles
184 COMPUTE!'s Guide To Typing In Programs
187 MLX Machine Language Entry Program For Commodore 64 And Unexpanded VIC-20
195 Product Mart
196 Advertisers Index

NOTE: See page 184 before typing in programs.

AP Apple **AT** Atari, **P** PET/CBM, **V** VIC-20, **C** Radio Shack Color Computer, **64** Commodore 64, **TS** Timex/Sinclair, **TI** Texas Instruments, **PCjr** IBM PCjr, **PC** IBM PC, **AD** Coleco Adam, *All or several of the above.

COMPUTE! Publications, Inc. abc
One of the ABC Publishing Companies

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

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 914, Farmingdale, NY 11737. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright © 1984 by COMPUTE!* Publications, Inc. All rights reserved, ISSN 0194-357X.

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

TI Reverse Flash

I own a TI-99/4A with Extended BASIC, but have programmed on a number of computers. Several of these computers, such as the Apple and Atari, have reverse video characters. Since the TI lacks reverse characters, I wrote the following short routine to simulate them:

```
100 REM INVERSE CHAR
110 CALL SCREEN(2)
120 FOR I=65 TO 90 :: CALL CHARPA
    T(I,A$):: CALL CHAR(I+32,A$):
    : NEXT I
130 CALL CLEAR
140 FOR I=9 TO 12 :: CALL COLOR(I
    ,2,16):: NEXT I :: FOR I=5 TO
     8 :: CALL COLOR(I,16,2):: NE
    XT I
150 A$="INVERSE"
160 B$="inverse"
170 DISPLAY AT(11,11):A$ :: FOR I
    =1 TO 50 :: NEXT I :: DISPLAY
     AT(11,11):B$ :: FOR I=1 TO 5
    0 :: NEXT I :: GOTO 170
180 END
```

This routine replaces the lowercase letters (produced with the ALPHA LOCK key up) with inverse capitals. First, in line 120, the CHARTPAT and CHAR subprograms replace the lowercase letters (characters 97–122) with capitals. Next, in line 140, color codes are assigned to the redefined characters to create inverse characters.

For added effect, a flashing routine similar to that produced with the Apple's FLASH command has been added in line 170.

J. P. Lester

Thank you for contributing this handy routine.

Commodore 1541 Head Alignment

I own a Commodore 64 and a 1541 disk drive. I am having problems loading programs that were saved about two months ago. Programs that were recently saved don't present a problem. When I attempt to load the older programs, the red read/write light flashes the entire time the program is loading. Some programs won't load, period. I've tried to clean my drive, but the problem persists. Can you please tell me what is causing this? I remember reading an article that said when programs are saved in different temperatures, problems may arise. If this is true, can this be the nature of my problem?

Gerry Robinson

Although temperature extremes can damage stored disks, it is probably not the source of your problem. As long as disks are used and stored within the recommended range of 50 to 125 degrees Fahrenheit, you shouldn't have any trouble.

The alignment of the read/write head in your disk drive may be skewed. The stepper motor sometimes slips out of alignment on some models of the 1541. This motor is responsible for precisely positioning the read/write head when the disk is reading or writing data. You should consider taking your drive to a Commodore Service Center to have it checked out.

If the red busy light on the front of the drive blinks while you're loading programs, this can indicate the drive is having trouble reading the data on the disk. This is not to be confused with the steadily blinking light encountered with a DOS (Disk Operating System) error. Ideally, the busy light should constantly glow red while reading data on the disk.

Computers And Laser Discs

I was wondering if Atari was planning to produce a laser disc machine for use with its computers. I had read they had planned to do so, but then decided to drop the idea. Is this true?

John Engman

Originally designed to store high-quality video images, the laser disc's power is only now being tapped. Unlike a videocassette recorder, which works like a computer tape drive, a laser disc player has fast random access to any frame, analogous to a computer disk drive. Theoretically, any computer can be interfaced with the relatively simple controls required to drive a laser disc. Digital Research, Inc., *the other machine, but don't want to change BASIC, you must copy the contents of the BASIC ROM to the underlying RAM with this statement:*

```
FOR I=40960 TO 49151:POKE I,PEEK(I):NEXT
```

After the Kernal and BASIC have been copied or loaded into RAM, use the aforementioned POKE, or simply POKE 1,53.

A BASIC Sort

My daughter has written an inventory program to list our music cassettes. It uses DATA statements to list type of music, name of cassette, and performer. We have for several months attempted to write a routine whereby we can list all the performers in alphabetical order, but without success. Is there any way we can do this and not have the program running forever?

Don Cordry

There are a number of good, fast sorts, but the bubble sort is one of the shortest and easiest to understand and modify. It works by comparing every item to the one beneath it. If the two items are out of order, they are switched. The sort continues until no more exchanges are necessary.

The name comes from the way lower-ranked data tends to "bubble" upwards. The small subroutine below can be used to sort string arrays. It's easy to modify for whatever purpose you need. The variable N should be set to the number of performers, and all the performers should be read into the array prior to the sort. This program will work as is with most versions of BASIC, but would need to be modified to run on an Atari.

```
5ØØØ EX=Ø
5Ø1Ø FORI=1TON-1
5Ø2Ø IFA$(I)>A$(I+1)THENT$=A$(I):A$(I)=A$
     (I+1):A$(I+1)=T$:EX=1
5Ø3Ø NEXT I
5Ø4Ø IFEX<>ØTHEN5ØØØ
5Ø5Ø RETURN
```

Commodore Compatibility

I have a Commodore 4032 computer with a Commodore 2031 disk drive. I am thinking about buying a Commodore 64, but only if the 2031 drive can be used with it. Is there any way this can be done?

Robert D. Byers

The 4032 computer and 2031 disk drive communicate over the IEEE-488 parallel bus. Bytes are sent eight bits at a time. The Commodore 64 and its 1541 disk drive use a serial bus that is similar to the IEEE-488, but it sends bytes one bit at a time. You cannot directly attach your IEEE-488 disk drive to the 64, but several manufacturers sell IEEE interfaces for the Commodore 64, some as low as $100. With an IEEE interface plugged into the cartridge port, your 2031 will transfer data faster than a 1541. There are also IEEE interfaces that attach through the serial port.

In addition, your drive is read and write compatible with the 1541, so you should be able to load most commercial software. Unfortunately, few of these interfaces are perfect. Some software just won't work with them, due to changes in the memory map caused by the addition of the interface.

VIC Paddle PEEKs

I own a Commodore VIC-20 and a set of paddle controllers, but cannot find the commands used to incorporate the paddles into my programming.

Brad Mills

Although there are no built-in commands in VIC BASIC for reading the paddles, there are two memory locations you can read. Location 36872 returns a value from 0 to 255 (corresponding to a counterclockwise rotation) for paddle 1. Paddle 2 is read by location 36873 in the same manner. In BASIC, use PEEK(36872) or PEEK(36873) to read the paddle position. The paddle buttons are read by checking the locations normally used to read the joystick. Paddle 1's fire button corresponds to a joystick position of west (left). Paddle 2's fire button is synonymous with a right deflection of the joystick. Also, be aware that Atari paddle controllers used on the VIC do not return the full 0–255 range provided by Commodore paddle controllers. Additional information can be found in the VIC-20 Programmer's Reference Guide, *or* COMPUTE!'s Mapping the VIC.

Commodore Colons

I have seen Commodore 64 programs that have a line number followed by a colon. What purpose does the colon serve?

Mike Wells

Most Microsoft BASICs allow you to put a colon as the first character in a line, and this has no effect on the running of the program (except to slow execution a bit). The superfluous colon is often used to merely insert a visual gap in the program listing, since you can't store a blank program line. Since many BASICs delete any leading spaces after a line number, the colon is also used to indent lines for increased readability, since spaces after a colon are preserved.

Atari Versus Commodore Disk Drives

I read in a lot of articles that the Atari disk drive is an intelligent drive like the Commodore 1541. But isn't it true that you have to load the disk operating system (DOS) into the Atari before it *operations are provided which could do the same job.*

The MLI is called by a JSR $BF00 instruction, followed by three bytes of data. The first byte is the number of the MLI function being requested, and the second and third bytes contain the address of the parameter list for the request. These three bytes must be placed in your program immediately after the JSR $BF00 instruction. The MLI function dispatcher increases the return address on the stack by three to skip over these bytes.

Although the MLI performs many of the same functions as the DOS File Manager, there is no compatibility between the two. ProDOS has a completely different set of function codes, error codes, and parameter list formats. Information about these codes, the structure of ProDOS, and lots more, is available in the Apple ProDOS Technical Reference Manual. *This publication is available from most Apple dealers and is intended for advanced programmers who want to use ProDOS from machine language.*

Commodore 64 Audio Input

I own a Commodore 64 and have had no problems with it at all. Documentation of all its features is another story. I know that the 64 has an audio input located on the audio/video port on the back of the unit. However, I have not been able to find any literature on how to access this feature. Could you please tell me how to use it? What memory locations are affected?

Kevin Caylor

The audio input pin is used to mix in an external sound source. You can test this by feeding the sound output of another 64 into the audio input. When mixing in another audio source, be sure it's at the same low level as SID chip output. (Feeding in an amplified signal could destroy your SID chip.) Intended for chaining SID chips together, the audio input becomes a kind of fourth voice, and is affected by the SID chip's volume and filter settings. Bit 3 of location 54295 enables the filtering of external audio. You cannot process sound per se, but you can use the SID chip's filter as a simple, programmable equalizer which will emphasize or reduce various frequencies.

IBM PC/PCjr BASIC Compatibility

I would like to know if a program written for the PCjr in Cartridge BASIC would work on the PC with a color/graphics adapter and BASICA.

Richard Bookal

PCjr Cartridge BASIC is a superset *of BASICA, which means that it contains all the commands found in BASICA plus some new ones. Likewise, the PCjr has all the graphics and sound features found in an IBM PC equipped with the color/graphics adapter, plus some enhancements. Therefore, programs written for a PCjr with Cartridge BASIC will run on a PC with a color/graphics adapter and BASICA only if the extra commands and features are not used.*

An example of a new Cartridge BASIC command is PCOPY. Briefly, this command copies an image from one screen page to another. But only the PCjr with Cartridge BASIC has this capability. If you attempt to run the program on a PC, BASICA won't know how to interpret PCOPY and an error will result.

An example of an enhanced feature on the PCjr is SCREEN 5, a graphics mode with 320 × 200-pixel resolution and 16 simultaneous colors. A program written for the PCjr using SCREEN 5 won't run on a PC equipped with the color/graphics adapter, because the PC's 320 × 200 graphics mode (SCREEN 1) is capable of displaying only four simultaneous colors.

If you want to write programs on a PCjr with Cartridge BASIC that will be compatible with a PC and BASICA, you'll have to avoid using all of these new commands and features. For your guidance, IBM's Cartridge BASIC manual generally states when a command is available only in Cartridge BASIC. It would also help to acquire a BASICA manual and familiarize yourself with a PC outfitted with the IBM color/graphics adapter.

Instant TI RUNs

Quite awhile ago I read about a command for the TI-99/4A which causes a program to RUN instantly after you hit ENTER. I looked through many books and articles and did not find this information. Can you help?

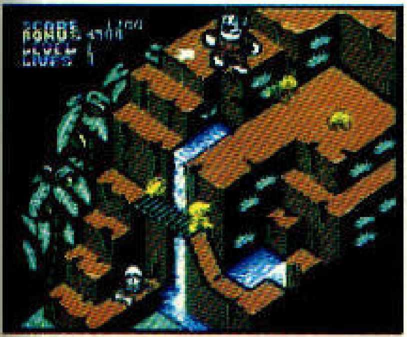
Dorr Wilson

It sounds like you are describing the pre-scan commands available with Extended BASIC. *These commands (!@P−and !@P+) are documented on pages 7 through 10 in the Addendum of the* TI Extended BASIC Manual.

When you enter RUN on the TI, there is a brief pause before the program executes. During this pause (most evident with long programs), the computer "pre-scans" the program and sets aside memory for variables, arrays, and data.

Only certain instructions in a TI BASIC program require pre-scanning. These include the first DATA statement, the first use of each variable and/or array, the first reference to each CALL statement of any subprogram, all DEF statements (for user-defined functions), and all SUB and SUBEND statements (and any variables introduced in the user-defined subprogram). So, rather than pre-scanning an entire program, you can pre-scan only part of it by appropriately positioning the pre-scan

WERE SUCH BIG HITS,
THEM HOME.

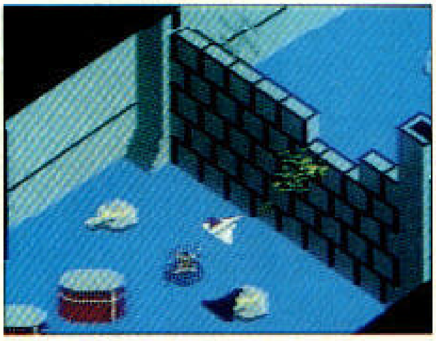


Sega's Congo Bongo rocked the home game world when it shot up to Number 3 on the Billboard chart this summer.

And now it's available for even more home systems. So check the chart and get ready for jungle action. You'll pursue the mighty ape Congo up Monkey Mountain and across the Mighty River. Do battle with dangerous jungle creatures. Ride hippos, dodge charging rhinos and try to avoid becoming a snack for a man-eating fish.

Congo Bongo. It's fast and it's fun. But be careful. It's a jungle in there.

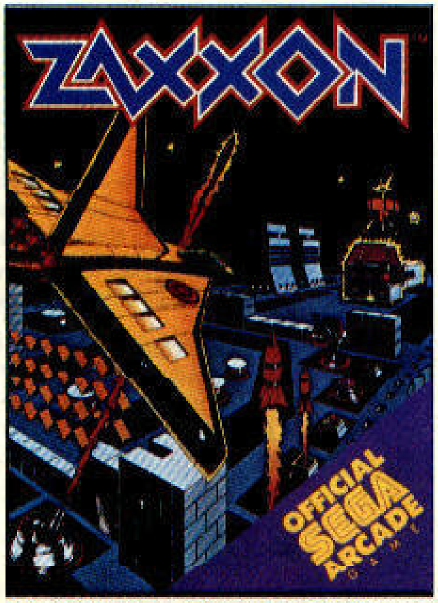
Arcade and Home Smash. Hit #3 on *Billboard* Magazine's Top Video Games survey.



Sega's Zaxxon. If you haven't played Zaxxon, you must have been living on another planet for the past few years.

And now the ultimate space combat game is available for even more home systems. You'll pilot a space fighter through force fields and enemy fire on your way to do battle with the mighty Zaxxon robot. Countless others have gone before you in this Hall of Fame game. But this time your life is in your own hands.

Zaxxon killed them in the arcades. But compared to what it will do to you at home, that was child's play.



One of only ten games ever to make *Electronic Games'* Hall of Fame.

	SPY HUNTER	TAPPER	UP 'N DOWN	CONGO BONGO	ZAXXON
Atari 2600 cartridge	✓ NEW	✓ NEW	✓ NEW	✓	✓
Atari 5200 cartridge				✓	✓ NEW
Atari Computers* cartridge	✓ NEW	✓ NEW	✓ NEW	✓	✓ NEW
Atari Computers† diskette	✓ NEW	✓ NEW	✓ NEW		✓
ColecoVision & ADAM cartridge	✓ NEW	✓ NEW	✓ NEW	✓ NEW	✓
Commodore 64 cartridge	✓ NEW	✓ NEW	✓ NEW	✓	✓ NEW
Commodore 64 diskette	✓ NEW	✓ NEW	✓ NEW	✓ NEW	✓
Apple II, IIe, IIc diskette	✓ NEW	✓ NEW	✓ NEW	✓ NEW	✓
IBM PC diskette	✓ NEW	✓ ** NEW	✓ NEW	✓ ** NEW	✓ ** NEW

✓ Published by Sega Enterprises, Inc.
✓ Published by Datasoft, Inc. under license from Sega Enterprises, Inc.
✓ Published by Coleco Industries, Inc. under license from Sega Enterprises, Inc.
✓ Published by Synapse Software Corporation under license from Sega Enterprises, Inc.
*Atari 400, 800, 600XL, 800XL and 1200XL.
(Congo Bongo cartridge: 400, 800 and 800XL.)
†Atari 800, 600XL, 800XL and 1200XL.
**Also available for IBM PCjr.

All new games are scheduled to be in your stores for Christmas. Check your local dealer.

‡Number of game levels varies on cartridges for Atari and Commodore systems. Atari, 2600, 5200, 400, 800, 600XL, 800XL, and 1200XL are trademarks of Atari Corporation. Commodore 64 is a trademark of Commodore Electronics, Inc. ColecoVision and ADAM are trademarks of Coleco Industries, Inc. Apple, II, IIe, and IIc are trademarks of Apple Computer, Inc. IBM, PC and PCjr are trademarks of International Business Machines Corp. UP 'N DOWN is a trademark of Sega Enterprises, Ltd., manufactured under license from Sega Enterprises, Ltd., Japan. Videogame copyright ©1983 Sega Enterprises, Ltd. BALLY MIDWAY is a trademark of Bally Midway Mfg. Co. Package and program copyright ©1984 Sega Enterprises, Inc. TAPPER and SPY HUNTER are trademarks of Bally Midway Mfg. Co. Videogame copyright ©1983 Bally Midway Mfg. Co. All rights reserved. ZAXXON is a trademark of Sega Enterprises, Inc. Copyright ©1984, Sega Enterprises, Inc. CONGO BONGO is a trademark of Sega Enterprises, Inc. Copyright ©1983, Sega Enterprises, Inc.

Reflection

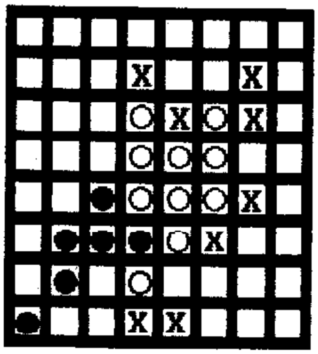
Sean Puckett

"Reflection" is a fast-paced computer version of reversi. You can play it as a strategy game with two people or challenge the brain of the computer. It was originally written for the Atari (24K), and we've added versions for the Commodore 64, unexpanded VIC-20, TI-99/4A (16K and regular BASIC), Apple, IBM PC (with 64K, BASICA, and the color/graphics adapter), PCjr (with Cartridge BASIC), and TRS-80 Color Computer (with Extended Color BASIC). A joystick is required for the Atari, 64, VIC, and Color Computer.

Through the ages, people have devised many pastimes to exercise their minds. The most well-known match of wits is chess, with backgammon and checkers running close behind. Another board game, reversi, is not as popular, but combines the logic of these games with the action and excitement of a good football game.

The trouble is, some players can become so excited that they tend to get carried away and attempt a forward pass with the board, or they fumble and scatter the chips everywhere (a method most often employed by sore losers). A computer version of reversi is ideal. The computer can act as a referee, permitting only legal moves, or it can be a ruthless opponent.

"Reflection" gives you the option of playing either way—against another person or against the computer. The rules are quite simple. Players take turns placing chips on the board, one piece per turn. To capture your opponent's pieces, you sandwich a row of them between one of your existing pieces and the one you're laying down. You can capture one or several pieces this way. The row can be vertical, horizontal, or diagonal. Once a piece is captured, it turns into your color and effectively becomes one of your pieces.



In this example, the black player can capture pieces by placing one of his chips on any spot marked here with an X:

The best move is either the one that captures the most pieces, or the one which leaves your own pieces less vulnerable—depending on the stage of the game. Sometimes you can place a single piece to capture more than one row of chips. Each player must capture at least one enemy piece per turn, or the turn is forfeited. When all of one player's pieces have been captured, or when neither player can make a legal move, the chips are tallied and the victory is awarded.

Because capturing an enemy piece converts it to your color, the game can reverse directions very quickly. A winning player can suddenly find himself far behind, with most of his chips flipped to the opponent's color.

Playing Reflection

The Atari version of Reflection uses one or two joysticks. You can play against another player or against the computer, and you can select whether black or white moves first. Move the large cursor with the joystick, then press the button to place your piece. You can put down only one piece per move, and only on empty squares. If you place your chip so it doesn't capture any enemy pieces, the program removes the piece and you forfeit your turn. You must purposely forfeit in this way if you can't make a legal move. If neither player can make a move, press E on the keyboard to end the game.

All other versions except the VIC version play much like the Atari version, but have extra options. When playing against the computer, there are two levels of computer intelligence. Level two plays better, but naturally it takes longer for the computer to make up its mind.

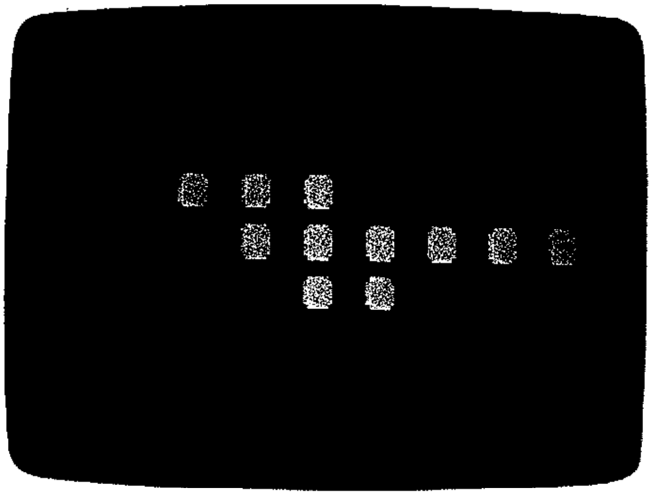
These versions also let you set up the board prior to play. On all computers except the Color Computer, press W to set down a white chip, B for a black chip, and space to skip a square. You continue left to right, top to bottom, until you reach the lower-right corner. On the Color Computer, use a joystick plugged into port 2 to move to any square, where you type W for a white chip, B for a blue chip, or space bar to leave an empty square.

The 64 version of Reflection requires a joystick plugged into port 2. The VIC-20 uses a single joystick for both players. Both the Apple and IBM versions use a diamond-shaped arrangement of keys to move the cursor: I for Up, M for Down, J for Left, and K for Right. The TI-99/4A version uses the arrow keys E, S, D, and X. When you've moved the cursor to the desired position, press the space bar to place your piece. As with the Atari version, you forfeit your turn and lose the piece if you place it so that no enemy pieces are captured. Press Q to end the game on the TI-99/4A, and E for all other versions.

Before loading the Apple version, first enter this direct statement:

```
POKE 104,64: POKE 16384,0: NEW
```

Similarly, enter PCLEAR 1 before loading the Color Computer version.



"Reflection," Atari version.

Program 1: Reflection For Atari

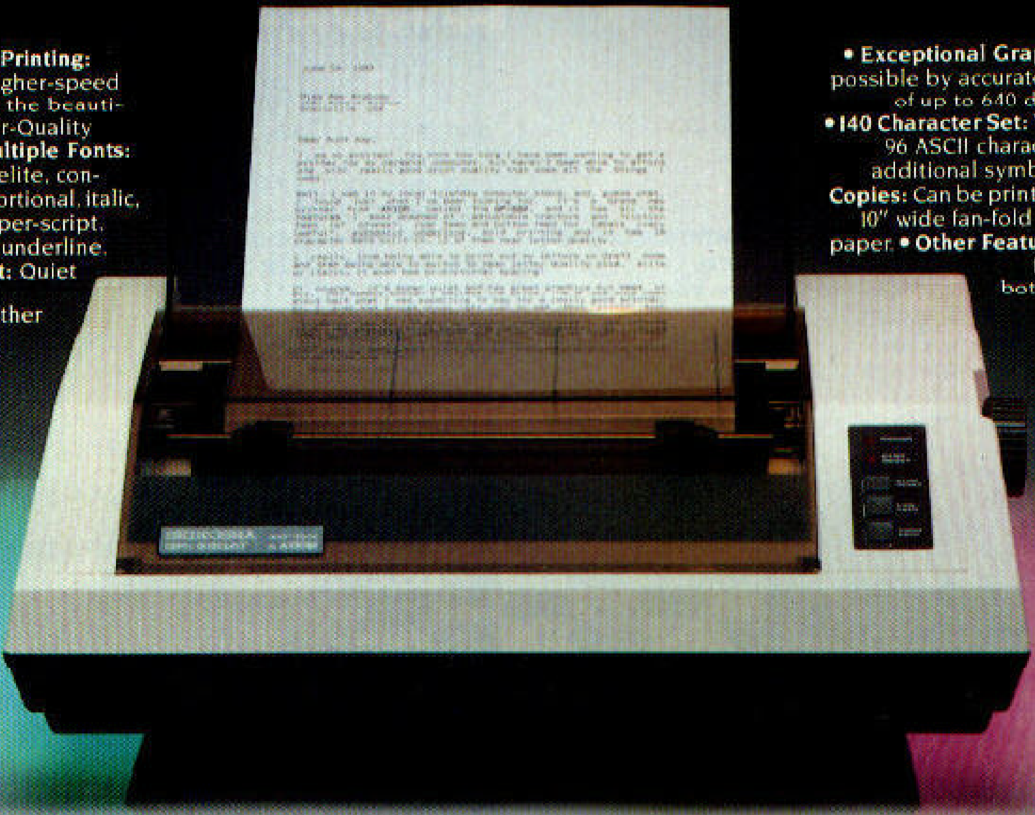
Refer to "COMPUTE!'s Guide To Typing In Programs" before entering this listing.

```
ML 1000 N1=1:N2=2:N0=0:N3=3:N4=4:N
        5=5:N6=6:N7=7:N8=8:N9=9:O2
        =N2
BB 1009 GRAPHICS 23:POKE 708,20:PO
        KE 709,0:POKE 710,16:POKE
        712,198:POKE 711,30:GOSUB
        1950:GOSUB 1720:UI=N1
LE 1010 DL=PEEK(560)+256*PEEK(561)
        :POKE DL+N3,70:POKE DL+N6,
        N6:DIM M$(40):DL=DL+UI:H=I
        NT(DL/256):L=DL-H*256
GI 1011 M$="{5 SPACES}reflection
        {5 SPACES}":GOSUB 1940
EF 1020 M$="{4 SPACES}press  start
        ":POKE 560,L:POKE 561,H
CI 1030 COLOR UI:FOR A=N1 TO 88:PL
        OT 16,A:DRAWTO 142,A:NEXT
        A
EL 1040 DIM X(N8),Y(N8):Z=UI:COLOR
         0:FOR A=N1 TO 88 STEP 11:
        Y(Z)=A+N2:Z=Z+UI:PLOT N0,A
        :DRAWTO 142,A:DRAWTO 146,A
        +4:NEXT A
KC 1050 Z=UI:FOR A=16 TO 142 STEP
        16:X(Z)=A+N4:Z=Z+UI:PLOT A
        ,UI:DRAWTO A,88:DRAWTO A+4
        ,92:NEXT A
JI 1051 COLOR N0:PLOT 143,N1:DRAWT
        O 143,89:DRAWTO 0,89
LN 1060 DIM BO(N9,N9)
BI 1070 FOR A=N0 TO N9:FOR B=N0 TO
         N9:BO(A,B)=N0:NEXT B:NEXT
         A
KC 1080 RESTORE 1080:FOR A=N1 TO N
        4:READ B,C,D:BO(B,C)=D:NEX
        T A:GOSUB 1810:DATA 4,4,2,
        5,5,2,4,5,3,5,4,3
OD 1090 GOSUB 1940:E=PEEK(711)
BG 1110 IF PEEK(53279)=N6 THEN FOR
         A=53248 TO 53251:POKE A,N
        0:NEXT A:GOTO 1130
MA 1120 GOTO 1110
AJ 1130 M$="{4 SPACES}<O>NE PLAYER
        {8 SPACES}<t>wo player":GO
        SUB 1940
JE 1132 OPEN #UI,12,0,"K:":GET #UI
        ,K:IF K=79 THEN PL1=1:GOSU
        B 2100
AC 1140 M$="{3 SPACES}<W>HITE FIRS
        T{7 SPACES}<b>lack first":
        GOSUB 1940
CP 1150 GET #UI,K:IF K=87 THEN TUR
        N=UI:GOTO 1180
LN 1160 IF K=66 THEN TURN=N2:GOTO
        1180
KF 1170 SOUND N0,255,10,15:POKE 53
        768,UI:FOR D=UI TO 500:NEX
        T D:SOUND N0,N0,N0,N0:GOTO
         1150
CO 1180 MOVE=N4:M$="  use joystick
         to{7 SPACES}move cursor"
        :GOSUB 1940:FOR D=N1 TO 10
        00:NEXT D:NW=N2:NB=N2
AG 1190 M$="  press trigger to
        {6 SPACES}enter move":GOSU
        B 1940:FOR D=UI TO 500:NEX
        T D:DIM F$(10):F$="WHITEbl
        ack"
NE 1200 M$(1)=" ":M$(40)=" ":M$(2)
        =M$:XP=4:YP=4:M$="
        {4 SPACES}":M$(5)=F$(TURN*
        5-4,TURN*5):M$(10)="'S MOV
        E":M$(22)="BLACK:"
HB 1210 M$(28)=STR$(NB):M$(32)="WH
        ITE:":M$(38)=STR$(NW):GOSU
        B 1940:DATA 243,1,121,4
OM 1215 IF TURN=2 AND PL1 THEN GOS
        UB 2200:GOSUB 1355:GOTO 13
        47
PO 1220 RESTORE 1210:IF TURN=N2 TH
        EN RESTORE 1220:DATA 121,1
        ,243,4
DG 1230 TG=N2:GOSUB 1700:CO=N8:GOS
        UB 1690:F=N1:K=N1
GN 1240 POKE 77,N0:Q=STICK(N0):IF
        (Q=10 OR Q=14 OR Q=N6) AND
         (YP>N1) THEN YP=YP-N1
AL 1245 IF PEEK(764)=42 THEN 1600
MP 1250 IF (Q=10 OR Q=11 OR Q=N9)
        AND (XP>N1) THEN XP=XP-N1
BF 1260 IF (Q=N6 OR Q=N7 OR Q=N5)
        AND (XP<N8) THEN XP=XP+N1
PL 1270 IF (Q=N9 OR Q=13 OR Q=N5)
        AND (YP<N8) THEN YP=YP+N1
```

Dual Personality.

Meet the Axiom GP-550.
The First Real Personal Printer for Home and Business — Just $299!

• **Dual-Mode Printing:** Choose the higher-speed draft mode or the beautiful Near-Letter-Quality printout. • **Multiple Fonts:** Include pica, elite, condensed, proportional, italic, sub-script, super-script, boldface and underline. • **Super-Quiet:** Quiet operation is essential whether for home or business and the GP-550 is quieter than most.



• **Exceptional Graphics:** Made possible by accurate placement of up to 640 dots per line. • **140 Character Set:** You can print 96 ASCII characters plus 44 additional symbols. • **Three Copies:** Can be printed on 4.5" to 10" wide fan-fold or cut-sheet paper. • **Other Features:** Pin and friction feed, bottom feed for labels, and cassette ink ribbon.

Imagine doing all your word processing, data processing, graphics and documentation on a printer that sells for only $299. You can stop imagining! The Axiom GP-550 — which offers both draft mode and Near-Letter-Quality printing — is here.

IBM-PC Compatible:

The GP-550 is available with control codes and character set that match with IBM-PC and all of its "Look Alikes." This means that your software, including Lotus 1-2-3, will work with our GP-550 PC model.

Direct-Connect to Atari, Apple, Commodore and T.I.:

More good news. Axiom has additional models of the GP-550 which include built-in interfaces for the most popular personal computers. These "Direct-Connect"™ versions of the GP-550 will work with your computer without additional interface or cable. Just plug our printer into your computer and start printing. Priced from $319, including interface.

GP-550CD for VIC-20/C64

GP-550AT for All Atari

GP-550AP for Apple II/IIe

GP-550TI for 99/4A

AXIOM®

AXIOM CORPORATION

1014 Griswold Avenue • San Fernando, CA 91340 • Telephone: (818) 365-9521 • TWX: 910-496-1746

GP-Series Printers Built for Lasting Quality by SEIKOSHA.

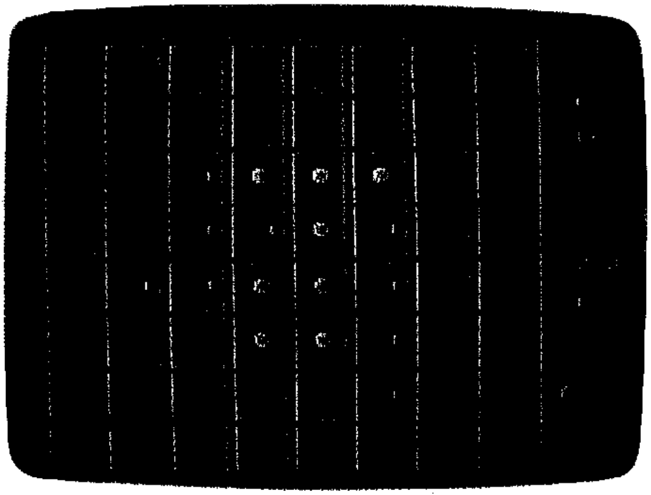
```
QP 9110 DATA 16,-6,6,2,2,6,-6,16,Ø,-6,
        -12,-2,-2,-2,-2,-12,-6,Ø
LJ 9120 DATA 6,-2,6,2,2,6,-2,6,Ø,2,-2,
        2,1,1,2,-2,2,Ø
BE 9130 DATA 2,-2,2,1,1,2,-2,2,Ø,6,-2,
        6,2,2,6,-2,6,Ø
PE 9140 DATA -6,-12,-2,-2,-2,-2,-12,-6
        ,Ø,16,-6,6,2,2,6,-6,16,Ø
DF 11000 DATA 1,1,1,1,1,Ø,Ø,1,1,Ø,Ø,1,
         1,1,1,1
```

Program 5: Reflection For TI-99/4A

Version by Pat Parrish, Programming Supervisor

Refer to "COMPUTE!'s Guide To Typing In Programs" before entering this listing.

```
10 DIM BO(80),TA(71),PT(71),A(71),P
   O(80)
20 GOTO 70
30 FOR I=1 TO LEN(A$)
40 CALL HCHAR(R,C+I,ASC(SEG$(A$,I,1
   )))
50 NEXT I
60 RETURN
70 TU=1
80 RANDOMIZE
90 GOSUB 3850
100 GOSUB 1310
110 IF DE=0 THEN 130
120 GOSUB 4090
130 GOSUB 1540
140 IF DE=0 THEN 170
150 GOSUB 1650
160 GOTO 330
170 RESTORE 4060
180 FOR Y=2 TO 5
190 FOR X=2 TO 5
200 READ PO(Y*9+X)
210 NEXT X
220 NEXT Y
230 BO(30)=2
240 BO(31)=1
250 BO(39)=1
260 BO(40)=2
270 BC=2
280 WC=2
290 CALL HCHAR(11,13,128)
300 CALL HCHAR(11,16,120)
310 CALL HCHAR(14,13,120)
```



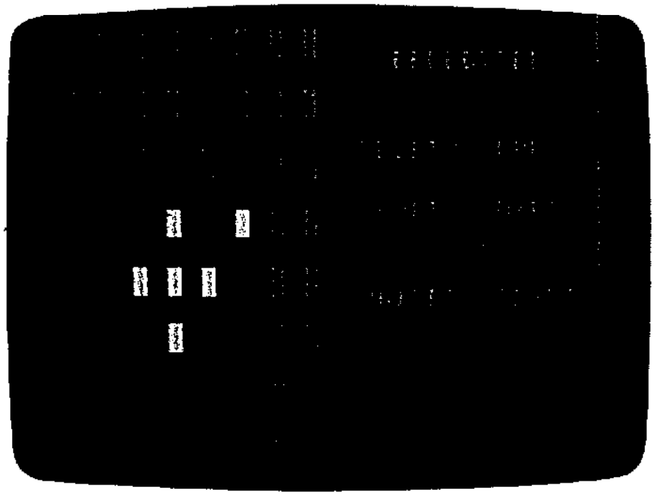
"Reflection," TI-99/4A version.

```
320 CALL HCHAR(14,16,128)
330 FL=1
340 X=4
350 Y=4
360 KH=128
370 IF TU<>1 THEN 390
380 KH=120
390 CALL HCHAR(4,28,KH)
400 A$=STR$(BC)&"  "
410 R=17
420 C=27
430 GOSUB 30
440 R=22
450 A$=STR$(WC)&"  "
460 GOSUB 30
470 IF (CM<>1)+(TU<>1)THEN 500
480 GOSUB 2730
490 GOTO 900
500 KH=1
510 CALL GCHAR(3*Y+2,3*X+4,GG)
520 KH=1-KH
530 CALL HCHAR(3*Y+2,3*X+4,120+8*KH
    )
540 CALL KEY(0,K,S)
550 IF S=0 THEN 520
560 IF (K<>ASC("E"))+(Y<1)THEN 600
570 CALL HCHAR(3*Y+2,3*X+4,GG)
580 Y=Y-1
590 GOTO 510
600 IF (K<>ASC("S"))+(X<1)THEN 640
610 CALL HCHAR(3*Y+2,3*X+4,GG)
620 X=X-1
630 GOTO 510
640 IF (K<>ASC("D"))+(X>6)THEN 680
650 CALL HCHAR(3*Y+2,3*X+4,GG)
660 X=X+1
670 GOTO 510
680 IF (K<>ASC("X"))+(Y>6)THEN 720
690 CALL HCHAR(3*Y+2,3*X+4,GG)
700 Y=Y+1
710 GOTO 510
720 IF K<>ASC("Q")THEN 870
730 A$="SURE YOU WANT TO END (Y/N)?
    "
740 R=24
750 C=2
760 GOSUB 30
770 CALL KEY(0,K,S)
780 IF S=0 THEN 770
790 IF K<>89 THEN 820
800 EE=1
810 GOTO 830
820 IF K<>78 THEN 770
830 A$=C$&"{3 SPACES}"
840 C=2
850 GOSUB 30
860 IF EE=1 THEN 2370
870 IF K<>ASC(" ")THEN 520
880 XY=Y*9+X
890 IF BO(XY)>0 THEN 520
900 IF FL=0 THEN 990
910 CALL HCHAR(Y*3+2,X*3+4,120+(TU-
    1)*8)
920 IF PO(XY)=0 THEN 990
930 CALL SOUND(100,440,2)
940 GOSUB 2060
950 IF CHIPS<1 THEN 990
960 GOSUB 2300
970 BO(XY)=TU
980 GOTO 1110
990 R=24
```

```
1000 CALL SOUND(100,110,2)
1010 C=2
1020 A$="ILLEGAL MOVE - LOSE TURN"
1030 GOSUB 30
1040 FOR I=1 TO 500
1050 NEXT I
1060 A$=C$
1070 GOSUB 30
1080 IF FL=0 THEN 1100
1090 CALL HCHAR(3*Y+2,3*X+4,32)
1100 GOTO 1210
1110 IF TU<>1 THEN 1150
1120 BC=BC+CHIPS+1
1130 WC=WC-CHIPS
1140 GOTO 1170
1150 WC=WC+CHIPS+1
1160 BC=BC-CHIPS
1170 FOR Q=1 TO 8
1180 IF XY+OF(Q)<0 THEN 1200
1190 PO(XY+OF(Q))=1
1200 NEXT Q
1210 TU=3-TU
1220 IF (WC=0)+(BC=0)+(WC+BC=64)THE
     N 2370
1230 GOSUB 1270
1240 IF (XY<>0)*(XY<>7)*(XY<>63)*(X
     Y<>70)THEN 1260
1250 GOSUB 3540
1260 GOTO 330
1270 FOR I=0 TO 71
1280 TA(I)=0
1290 NEXT I
1300 RETURN
1310 CALL CLEAR
1320 CALL SCREEN(11)
1330 PRINT TAB(10);"REFLECTION": :
     :
1340 PRINT TAB(11);"1ST MOVE"
1350 INPUT "{5 SPACES}(B)LACK/(W)HI
     TE: ":A$
1360 PRINT : :
1370 IF (A$<>"B")*(A$<>"W")THEN 134
     0
1380 IF A$="B" THEN 1400
1390 TU=2
1400 PRINT TAB(10);"GAME BOARD"
1410 INPUT "  (N)ORMAL/(D)ESIGN ONE
     : ":A$
1420 PRINT : :
1430 IF (A$<>"D")*(A$<>"N")THEN 140
     0
1440 IF A$="N" THEN 1460
1450 DE=1
1460 INPUT "{3 SPACES}# OF PLAYERS
     [1/2] ?: ":CM
1470 IF (CM<>1)*(CM<>2)THEN 1460
1480 PRINT : :
1490 CM=(CM=2)*2+CM
1500 IF CM=0 THEN 1530
1510 INPUT "{4 SPACES}SKILL LEVEL [
     1/2] ?: ":LE
1520 IF (LE<>1)*(LE<>2)THEN 1510
1530 RETURN
1540 A$="pqrqqrqqrqqrqqrqqrqqrqqr"
1550 B$="s t  t  t  t  t  t  t  t"
1560 C$="uvwvvwvvwvvwvvwvvwvvwvvw"
1570 CALL SCREEN(2)
1580 CALL COLOR(11,1,1)
1590 CALL COLOR(13,1,1)
1600 PRINT A$,B$,B$,A$&"  'S",B$,B$
     &" UP",A$,B$,B$,A$,B$,B$,A$&"
     SUM",B$,B$&" x:",A$,B$,B$,A$,B
     $&" "&CHR$(128)&":",B$,A$,B$,C
     $;
1610 CALL SCREEN(11)
1620 CALL COLOR(11,5,1)
1630 CALL COLOR(13,16,1)
1640 RETURN
1650 KH=0
1660 FOR Y=0 TO 7
1670 FOR X=0 TO 7
1680 KH=1-KH
1690 CALL HCHAR(3*Y+2,3*X+4,120+8*K
     H)
1700 CALL KEY(0,K,S)
1710 IF S=0 THEN 1680
1720 XY=X+Y*9
1730 IF K<>87 THEN 1770
1740 WC=WC+1
1750 BO(XY)=2
1760 GOTO 1850
1770 IF K<>66 THEN 1810
1780 BC=BC+1
1790 BO(XY)=1
1800 GOTO 1850
1810 IF K<>32 THEN 1680
1820 CALL HCHAR(3*Y+2,3*X+4,32)
1830 BO(XY)=0
1840 GOTO 1900
1850 CALL HCHAR(3*Y+2,3*X+4,120+8*(
     BO(XY)-1))
1860 FOR E=1 TO 8
1870 IF XY+OF(E)<=-1 THEN 1890
1880 PO(XY+OF(E))=1
1890 NEXT E
1900 NEXT X
1910 NEXT Y
1920 A$="OK?"
1930 R=22
1940 C=27
1950 GOSUB 30
1960 CALL KEY(0,K,S)
1970 IF S=0 THEN 1960
1980 IF (K<>78)*(K<>89)THEN 1960
1990 IF K<>89 THEN 2020
2000 CALL HCHAR(22,27,32,4)
2010 RETURN
2020 WC=0
2030 BC=0
2040 GOSUB 1540
2050 GOTO 1650
2060 CHIPS=0
2070 FOR I=1 TO 8
2080 L=1
2090 V=0
2100 XX=0
2110 V=V+OF(I)
2120 IF (XY+V>70)+(XY+V<0)THEN 2200
2130 IF BO(XY+V)=5 THEN 2200
2140 IF BO(XY+V)<>3-TU THEN 2180
2150 XX=1
2160 L=L+1
2170 GOTO 2110
2180 IF (XX<>1)+(BO(XY+V)<>TU)THEN
     2200
2190 GOSUB 2220
2200 NEXT I
2210 RETURN
2220 W=1
2230 V=0
2240 V=V+OF(I)
2250 TA(XY+V)=TU
```

```
2260 W=W+1
2270 IF W<L THEN 2240
2280 CHIPS=CHIPS+W-1
2290 RETURN
2300 FOR I=0 TO 71
2310 IF TA(I)=0 THEN 2350
2320 L=INT(I/9)
2330 CALL HCHAR(L*3+2,(I-9*L)*3+4,1
     20+(TU-1)*8)
2340 BO(I)=TU
2350 NEXT I
2360 RETURN
2370 REM WINNER
2380 IF BC<=WC THEN 2430
2390 A$="BLACK"
2400 HI=BC
2410 LO=WC
2420 GOTO 2490
2430 IF BC=WC THEN 2480
2440 A$="WHITE"
2450 HI=WC
2460 LO=BC
2470 GOTO 2490
2480 A$="TIE GAME."
2490 R=24
2500 C=3
2510 IF SEG$(A$,1,1)="T" THEN 2540
2520 CALL VCHAR(3,27,32,96)
2530 A$=A$&" WINS "&STR$(HI)&" TO "
     &STR$(LO)&" !"
2540 GOSUB 30
2550 BC-0
2560 WC=0
2570 DE=0
2580 TU=1
2590 FOR I=0 TO 71
2600 PO(I)=0
2610 BO(I)=0
2620 TA(I)=0
2630 NEXT I
2640 FOR I=1 TO 750
2650 NEXT I
2660 A$="  PLAY AGAIN (Y/N)?  "
2670 GOSUB 30
2680 CALL KEY(0,K,S)
2690 IF S=0 THEN 2680
2700 IF (K<>78)*(K<>89)THEN 2680
2710 IF K=89 THEN 100
2720 STOP
2730 HY=-32000
2740 HI=-32000
2750 XY=0
2760 IF (BO(XY)>0)+(PO(XY)=0)THEN 2
     960
2770 GOSUB 2060
2780 IF CHIPS=0 THEN 2960
2790 QW=(TT/8)*CHIPS+PT(XY)*(65-(TT
     /8))
2800 IF (LE<>2)+(CHIPS<>A1)THEN 282
     0
2810 QW=10000
2820 IF (LE<>2)+(RE<>0)THEN 2850
2830 GOSUB 3100
2840 GOTO 2960
2850 IF (QW<=HI)THEN 2890
2860 HI=QW
2870 H1=XY
2880 GOTO 2960
2890 IF HI=0 THEN 2960
2900 IF (QW/HI<.86)+(QW/HI>1.14)THE
     N 2960
2910 RANDOMIZE
2920 ZZ=INT(RND*2)+1
2930 IF ZZ<>1 THEN 2960
2940 HI=QW
2950 H1=XY
2960 XY=XY+1
2970 IF XY<71 THEN 2760
2980 IF (LE<>2)+(RE<>1)THEN 3000
2990 RETURN
3000 IF ((HI<>-32000)+(LE<>1))*((HY
     <>-32000)+(LE<>2))THEN 3030
3010 FL=0
3020 CHIPS=0
3030 XY=H1
3040 IF LE<>2 THEN 3060
3050 XY=H2
3060 GOSUB 1270
3070 Y=INT(XY/9)
3080 X=XY-Y*9
3090 RETURN
3100 A1=BC+1
3110 FOR E=0 TO 70
3120 A(E)=BO(E)
3130 IF TA(E)<1 THEN 3160
3140 BO(E)=TA(E)
3150 A1=A1+1
3160 NEXT E
3170 BO(XY)=1
3180 FOR Q=1 TO 8
3190 IF XY+OF(Q)<0 THEN 3210
3200 PO(XY+OF(Q))=PO(XY+OF(Q))+1
3210 NEXT Q
3220 NW=QW
3230 RE=1
3240 Y1=XY
3250 TU=2
3260 GOSUB 2740
3270 RE=0
3280 QY=NW-HI
3290 TU=1
3300 IF QY<=HY THEN 3340
3310 HY=QY
3320 H2=Y1
3330 GOTO 3410
3340 IF HY=0 THEN 3410
3350 IF (QY/HY<.86)+(QW/HY>1.14)THE
     N 3410
3360 RANDOMIZE
3370 ZZ=INT(RND*2)+1
3380 IF ZZ<>1 THEN 3410
3390 HY=QY
3400 H2=Y1
3410 XY=Y1
3420 FOR E=0 TO 70
3430 BO(E)=A(E)
3440 NEXT E
3450 GOSUB 1270
3460 FOR Q=1 TO 8
3470 IF Y1+OF(Q)<0 THEN 3520
3480 IF PO(Y1+OF(Q))<>2 THEN 3510
3490 PO(Y1+OF(Q))=1
3500 GOTO 3520
3510 PO(Y1+OF(Q))=0
3520 NEXT Q
3530 RETURN
3540 IF XY=7 THEN 3640
3550 IF XY=63 THEN 3710
3560 IF XY=70 THEN 3780
3570 FOR I=9 TO 13
3580 PT(I)=15-I
3590 NEXT I
```

```
3600 FOR I=1 TO 37 STEP 9
3610 PT(I)=6-INT(I/9)
3620 NEXT I
3630 RETURN
3640 FOR I=6 TO 42 STEP 9
3650 PT(I)=6-INT(I/9)
3660 NEXT I
3670 FOR I=16 TO 12 STEP -1
3680 PT(I)=I-10
3690 NEXT I
3700 RETURN
3710 FOR I=54 TO 58
3720 PT(I)=60-I
3730 NEXT I
3740 FOR I=64 TO 28 STEP -9
3750 PT(I)=INT(I/9)-1
3760 NEXT I
3770 RETURN
3780 FOR I=61 TO 57 STEP -1
3790 PT(I)=I-55
3800 NEXT I
3810 FOR I=69 TO 33 STEP -9
3820 PT(I)=INT(I/9)-1
3830 NEXT I
3840 RETURN
3850 FOR I=1 TO 8
3860 READ OF(I)
3870 NEXT I
3880 FOR X=0 TO 71
3890 READ PT(X)
3900 NEXT X
3910 FOR I=8 TO 71 STEP 9
3920 BO(I)=5
3930 NEXT I
3940 FOR I=0 TO 7
3950 READ A$
3960 CALL CHAR(I+112,A$)
3970 NEXT I
3980 CALL CHAR(120,"003C7E7E7E7E3C0
     0")
3990 CALL CHAR(128,"003C7E7E7E7E3C0
     0")
4000 RETURN
4010 DATA -10,-9,-8,-1,1,8,9,10
4020 DATA 16,-6,6,2,2,6,-6,16,0,-6.
     -12,-2,-2,-2,-2,-12,-6,0
4030 DATA 6.-2.6.2.2.6.-2.6.0.2.-2.
     2,1,1,2,-2,2,0
4040 DATA 2,-2,2,1,1,2,-2,2,0,6,-2,
     6,2,2,6,-2,6,0
4050 DATA -6,-12,-2,-2.-2.-2.-12.-6
     ,0,16,-6,6,2,2,6,-6,16,0
4060 DATA FFFFC0C0C0C0C0C0,FFFF0000
     00000000,FFFF030303030303,C0C0
     C0C0C0C0C0C0
4070 DATA 0303030303030303,C0C0C0C0
     C0C0FFFF,000000000000FFFF,0303
     03030303FFFF
4080 DATA 1,1,1,1,1,0,0,1,1,0,0,1,1
     ,1,1,1
4090 CALL CLEAR
4100 CALL SCREEN(13)
4110 PRINT TAB(3);"TYPE (B) FOR BLA
     CK CHIP": : : :
4120 PRINT TAB(3);"TYPE (W) FOR WHI
     TE CHIP": : : :
4130 PRINT TAB(3);"TYPE SPACE FOR N
     O CHIP": : : : : : :
4140 FOR T=1 TO 750
4150 NEXT T
4160 RETURN
```



"Reflection," TRS-80 Color Computer version.

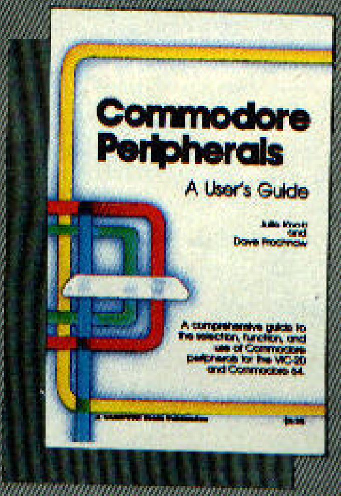
Program 6: Reflection For TRS-80 Color Computer

Version By Chris Poer, Editorial Programmer

Refer to "COMPUTE!'s Guide To Typing In Programs" before entering this listing.

```
10 CLEAR:DIM BO(80),TA(71),PT(71),A
   (71),PO(80)
20 BT$=CHR$(161)+CHR$(162):BB$=CHR$
   (164)+CHR$(168):WT$=CHR$(193)+CH
   R$(194)=WB$=CHR$(196)+CHR$(200)
30 CT$=CHR$(177)+CHR$(178):CB$=CHR$
   (180)+CHR$(184):ET$=CHR$(241)+CH
   R$(242):EB$=CHR$(244)+CHR$(248)
40 CLS:TU=1:PL=1
50 GOSUB 670
60 GOSUB 590
70 GOSUB 900
80 IF DE=1 THEN GOSUB 1040:GOTO 140
90 FOR Y=2TO5:FORX=2TO5
100 READA:PO(Y*9+X)=A:NEXTX:NEXTY
110 BO(30)=2:BO(31)=1:BO(39)=1:BO(4
    0)=2:BC=2:WC=2
120 PRINT@198,WT$;:PRINT@200,BT$;:P
    RINT@230,WB$;:PRINT@232,BB$;
130 PRINT@262,BT$;:PRINT@264,WT$;:P
    RINT@294,BB$;:PRINT@296,WB$;
140 FL=0:WC$=STR$(WC)+"   ":BC$=STR$
    (BC)
150 IF TU=1 THEN A$="BLUE'S TURN":G
    OTO180
160 A$="WHITE'S TURN"
170 PRINT@51,"REFLECTION"
180 PRINT@146,A$:PRINT@210,"BLUE'S
    CHIPS":PRINT@248,BC$
190 PRINT@306,"WHITE'S CHIPS":PRINT
    @344,WC$
200 IF PL=1 THEN AL=BC+1:GOTO220
210 AL=WC+1
220 IF TU=PL AND CM=1 THEN GOSUB 16
    20:GOTO340
230 A=JOYSTK(0):X=INT(JOYSTK(2)/8):
    Y=INT(JOYSTK(3)/8)
240 SP=Y*64+X*2:XY=X+Y*9
250 PRINT@SP,CT$;:PRINT@SP+32,CB$;
260 FORI=1TO50:NEXTI
270 IF (PEEK(65280)=253 OR PEEK(652
    80)=125) AND BO(X+Y*9)=0 THEN 3
    50
```

COMPUTE! BOOKS


Commodore Peripherals
A User's Guide
Julie Knott and Dave Prochnow
A comprehensive guide to the selection, function, and use of Commodore peripherals for the VIC-20 and Commodore 64.

Commodore Peripherals: A User's Guide

Julie Knott and David Prochnow

This easy-to-understand book guides the computer owner through the myriad of Commodore peripherals available for the 64 and VIC-20. Clear descriptions explain the use and function of disk drives, printers, expanders, cartridges, Datassette, modems, and CP/M. For all VIC and 64 users who plan to add to their basic computer system.

$9.95

ISBN 0-942386-56-6

33 Programs for the TI-99/4A

Brian Flynn

Contains something for everyone: chapters on games ("Rings and Poles"), money management ("IRA Planner"), business ("Internal Rate of Return"), simple statistics ("Mean, Variance, and Standard Deviation"), and more. A wide variety of applications software, plus games. Thirty-three ready-to-type-in programs at a low cost.

$12.95

ISBN 0-942386-42-6

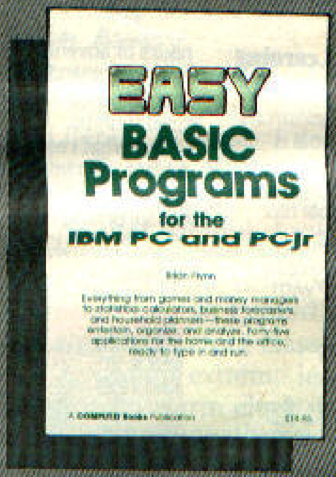
VIC-20 and Commodore 64 Tool Kit: BASIC

Dan Heeb

This comprehensive guide to the BASIC ROM routines in the VIC-20 and Commodore 64, thoroughly documented and clearly written, shows programmers how to call these routines from their own BASIC or machine language programs. Not only a "how-to," the *Tool Kit* is also an extensive reference guide to this important part of Commodore BASIC. For intermediate to advanced programmers.

$16.95

ISBN 0-942386-32-9



Easy BASIC Programs for the IBM PC and PCjr

Brian Flynn

From games and money managers to statistical calculators, business forecasters, and household planners, this wide-ranging collection of 45 programs for your PC or PCjr has something for almost everyone. "Pie Plot" draws charts, "Computer Cash Register" turns your computer into an electronic register, and "Time-Series Forecasting" predicts future trends. "Household Budget," "Paycheck Analysis," and "Municipal Bond Buyer" help you organize your personal and household finances. Or play strategic games, compute calculus problems, and even analyze statistics.

$14.95

ISBN 0-942386-58-2

To order your copy, call toll-free **1-800-334-0868** or write:
COMPUTE! Books, P.O. Box 5406, Greensboro, NC 27403

Please include a $2.00 shipping and handling charge per book on all book orders.

C. Regena

Algebra Tutorial

Part 2

Last month's column introduced "Algebra Tutorial," an educational program for students learning higher math. Part 2 presents the rest of the program listing and line-by-line explanation.

You'll recall that "Algebra Tutorial" is intended for students who already have some knowledge of algebra. It assumes the student is familiar with terms usually introduced before binomial multiplication. (A binomial is a numeric expression of two terms.) Basically, the tutorial covers the multiplication of one binomial by another—such as (x+5) times (x+4).

Last month's column included the program listing for lines 110 through 1300. Briefly, these lines redefine a few characters into special algebraic symbols (160–170); print a screen showing a comparison of binomial multiplication and numeric multiplication problems (190–300); display the general form of the multiplication problem and its answer (310–460); present a problem to the student (470–950); print a screen showing numeric coefficients for the first term (960–1110); present a problem to the student involving numeric coefficients (1120–1180 and the subroutine starting at 1960); and display a screen of information about using positive and negative numbers (1190–1300).

Picking up where we left off, lines 1310–1320 (and the subroutine at 1880) give the student a problem which may contain positive and negative numbers and coefficients in the first term. Lines 1330–1520 are two more screens of final information.

Helping Where It Is Needed

One advantage a computer tutorial has over a textbook is that a student can work at his or her own pace, yet get immediate feedback. Random numbers make the problems different each time so the student doesn't just memorize a sequence. If the student has trouble with one section, the computer can repeat the section many times. On the other hand, if the student knows the subject, the computer can keep track of the progress and advance accordingly.

Each time the student works on a problem, a flag F is set to zero. The student presses a key at each prompt. If the key pressed is incorrect, there is a low "uh-oh" sound, the flag F is set to 1, and the student must try again. For the numbers, the program won't continue until the correct numbers are pressed. On the + or − signs, however, the correct sign is printed and the program continues. If the problem is completed without any errors, the student has the option to try another problem or to continue the program. If an error has occurred, the flag F will be 1, and the student will be given another problem.

The program from line 1530 to the end contains subroutines which are used in several places. Lines 1530–1570 are the subroutine that checks if the ENTER key has been pressed. Lines 1580–1610 are the subroutine for an incorrect response—the computer plays an "uh-oh" sound and F is set to 1.

Lines 1620–1680 play a prompting beep, blink a question mark while waiting for the student to press a key, then print the key pressed.

The prompting is always done on the twenty-third row, or the row just printed. The column C is specified before the subroutine is called. Lines 1690–1730 play an arpeggio after the problem has been completed.

Lines 1740–1870 contain the subroutine to get an answer. P$ is the string value of the correct answer. The numbers may be one or two digits, so this subroutine also determines the number of digits in the answer and where to place the prompting positions. CC and C are variables used for determining the columns.

Presenting Problems

Lines 1880–2690 are the subroutine to present a problem to the student. The first type of problem (T=1) is for positive numbers only, and the subroutine is entered at line 1960. For the second type of problem (T=2), the subroutine is entered at line 1880. SD and SE are the signs for the second terms, D and E. SD$ and SE$ are the corresponding symbols. For the first type of problem, SD and SE are 1, but for the second type they may be 1 or −1.

Lines 1970–2020 choose the coefficients of the first terms, A and B, and the constants D and E for the second terms. Lines 2010–2020 make sure there will be a middle term in the multiplication. F is the flag for error checking. A$ and B$ are the string values of A and B, which are necessary for ease in printing. X$ is the variable in the binomials, which may be X, Y, or Z.

Lines 2070–2110 print the problem. Lines 2120–2600 print sections of the problem and get the student's answers. For each answer SS is the sign (plus or minus), C or CC is the column for the prompt and answer, and P$ is the correct answer. SGN is a function used to determine the sign. SGN returns 1 for a positive number, 0 for zero, and −1 for a negative number.

Lines 2700–2840 are the subroutine to get the student's answer for the + or − sign. When a sign needs to be chosen, a plus, minus, and question mark blink in position while waiting for the student's answer. The student must press the plus or the minus sign. If the answer is incorrect, a low "uh-oh" sounds, then the correct sign is printed. Both symbols are shifted. The student should use the LEFT SHIFT key to type these symbols to avoid an accidental FCTN + (QUIT). If you prefer to let the student press only the keys without SHIFT, change lines 2770 and 2810.

Customizing The Tutorial

Using the ideas in this program, you can design more subroutines to cover positive and negative first terms, alphabetic coefficients, alphabetic characters in the second terms, multiplying polynomials, factoring, etc.

If you prefer to save the typing time and effort, you may have a copy of this program by sending $3, a diskette (please pack in stiff cardboard) or blank cassette, and a stamped, self-addressed mailer to:

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

Please be sure to specify the name of the program and that you need the TI version.

Algebra Tutorial, Part 2

```
1310 T=2
1320 GOSUB 1880
1330 CALL CLEAR
1340 CALL SCREEN(4)
1350 PRINT "THERE MAY BE CASES WHEN
     "
1360 PRINT :"THE MIDDLE TERM BECOME
     S ZERO"
1370 PRINT :"SO YOU DO NOT NEED TO"
1380 PRINT :"SPECIFY A MIDDLE TERM.
     "
1390 PRINT ::"  X + 3";TAB(20);"4X
     + 2"
1400 PRINT "  X - 3";TAB(20);"4X -
     2"
1410 PRINT "  _____";TAB(19);"_____
     __"
1420 PRINT "{3 SPACES}^";TAB(22);"^
     "
1430 PRINT "  X - 9";TAB(19);"16X -
      4"::
1440 GOSUB 1530
1450 PRINT "OTHER MULTIPLICATION"
1460 PRINT :"PROBLEMS INCLUDE + AND
      -"
1470 PRINT :"NUMBERS IN THE FIRST T
     ERM"
1480 PRINT :"AND ALPHABETIC CHARACT
     ERS"
1490 PRINT :"FOR COEFFICIENTS."
1500 PRINT ::"THIS COMPLETES THIS U
     NIT"
1510 PRINT :"OF INSTRUCTION."::
1520 STOP
1530 PRINT ::"PRESS <ENTER>"
1540 CALL KEY(0,K,S)
1550 IF K<>13 THEN 1540
1560 CALL CLEAR
1570 RETURN
1580 CALL SOUND(100,165,2)
1590 CALL SOUND(100,131,2)
1600 F=1
1610 RETURN
1620 CALL SOUND(150,1497,2)
1630 CALL KEY(0,K,S)
1640 CALL HCHAR(23,C,63)
1650 CALL HCHAR(23,C,32)
1660 IF S<1 THEN 1630
1670 CALL HCHAR(23,C,K)
1680 RETURN
1690 CALL SOUND(100,262,2)
1700 CALL SOUND(100,330,2)
1710 CALL SOUND(100,392,2)
1720 CALL SOUND(200,523,2)
1730 RETURN
```

```
1740 L=LEN(P$)
1750 IF L=2 THEN 1770
1760 CC=CC+1
1770 C$=""
1780 FOR I=1 TO L
1790 C=CC+I
1800 GOSUB 1620
1810 C$=C$&CHR$(K)
1820 NEXT I
1830 IF C$=P$ THEN 1870
1840 GOSUB 1580
1850 CALL HCHAR(23,CC+1,32,L)
1860 GOTO 1770
1870 RETURN
1880 SD=(-1)^(INT(2*RND)+1)
1890 SD$="+"
1900 IF SD=1 THEN 1920
1910 SD$="-"
1920 SE=(-1)^(INT(2*RND)+1)
1930 SE$="+"
1940 IF SE=1 THEN 1960
1950 SE$="-"
1960 CALL CLEAR
1970 A=INT(7*RND)+1
1980 B=INT(7*RND)+1
1990 D=INT(7*RND)+1
2000 E=INT(7*RND)+1
2010 IF (A=B)+(D=E)=-2 THEN 1970
2020 IF (A*E*SE=(-1)*B*D*SD)THEN 19
     70
2030 F=0
2040 A$=STR$(A)
2050 B$=STR$(B)
2060 X$=CHR$(88+INT(3*RND))
2070 PRINT "MULTIPLY"
2080 PRINT TAB(19);A$;X$;" ";SD$;"
     ";D
2090 PRINT :TAB(19);B$;X$;" ";SE$;"
      ";E
2100 PRINT TAB(18);"________"
2110 PRINT :" ";STR$(E);"*TOP";TAB(
     20);X$;" +"
2120 IF T=1 THEN 2180
2130 IF SE=1 THEN 2150
2140 CALL HCHAR(23,3,45)
2150 SS=SE
2160 C=18
2170 GOSUB 2700
2180 P$=STR$(A*E)
2190 CC=19
2200 GOSUB 1740
2210 IF T=1 THEN 2250
2220 SS=SGN(SE*SD)
2230 C=24
2240 GOSUB 2700
2250 P$=STR$(D*E)
2260 CC=25
2270 GOSUB 1740
2280 PRINT TAB(14);"^"
2290 PRINT B$;X$;"*TOP";TAB(13);X$;
     "  +{3 SPACES}";X$
2300 P$=STR$(A*B)
2310 CC=12
2320 GOSUB 1740
2330 IF T=1 THEN 2370
2340 SS=SD
2350 C=18
2360 GOSUB 2700
2370 P$=STR$(B*D)
2380 CC=19
2390 GOSUB 1740
2400 PRINT TAB(11);"_______________
     "
2410 PRINT TAB(14);"^"
2420 PRINT "ADD";TAB(13);X$;"   +
     {3 SPACES}";X$;" +"
2430 P$=STR$(A*B)
2440 CC=12
2450 GOSUB 1740
2460 IF T=1 THEN 2500
2470 SS=SGN(A*E*SE+B*D*SD)
2480 C=18
2490 GOSUB 2700
2500 P$=STR$(ABS(A*E*SE+B*D*SD))
2510 CC=19
2520 GOSUB 1740
2530 IF T=1 THEN 2570
2540 SS=SGN(SD*SE)
2550 C=24
2560 GOSUB 2700
2570 P$=STR$(D*E)
2580 CC=25
2590 GOSUB 1740
2600 GOSUB 1690
2610 IF F=0 THEN 2640
2620 GOSUB 1530
2630 IF T=1 THEN 1960 ELSE 1880
2640 PRINT ::"CHOOSE:  1 ANOTHER PR
     OBLEM"
2650 PRINT TAB(10);"2 CONTINUE PROG
     RAM"
2660 CALL KEY(0,K,S)
2670 IF K=49 THEN 2630
2680 IF K<>50 THEN 2660
2690 RETURN
2700 CALL SOUND(150,1497,2)
2710 CALL KEY(0,K,S)
2720 CALL HCHAR(23,C,63)
2730 CALL HCHAR(23,C,43)
2740 CALL HCHAR(23,C,63)
2750 CALL HCHAR(23,C,45)
2760 IF S<1 THEN 2710
2770 IF (K=43)+(K=45)THEN 2780 ELSE
      2710
2780 S$="+"
2790 IF SS=1 THEN 2810
2800 S$="-"
2810 IF K=ASC(S$)THEN 2830
2820 GOSUB 1580
2830 CALL HCHAR(23,C,ASC(S$))
2840 RETURN
2850 END
```