Telecommunications: How To Get Started

COMPUTE!

$2.50
November
1983
Issue 42
Vol. 5, No. 11

$1.65 UK 3.25 Canada
63379
ISSN 0194-347X

The Leading Magazine Of Home, Educational, And Recreational Computing

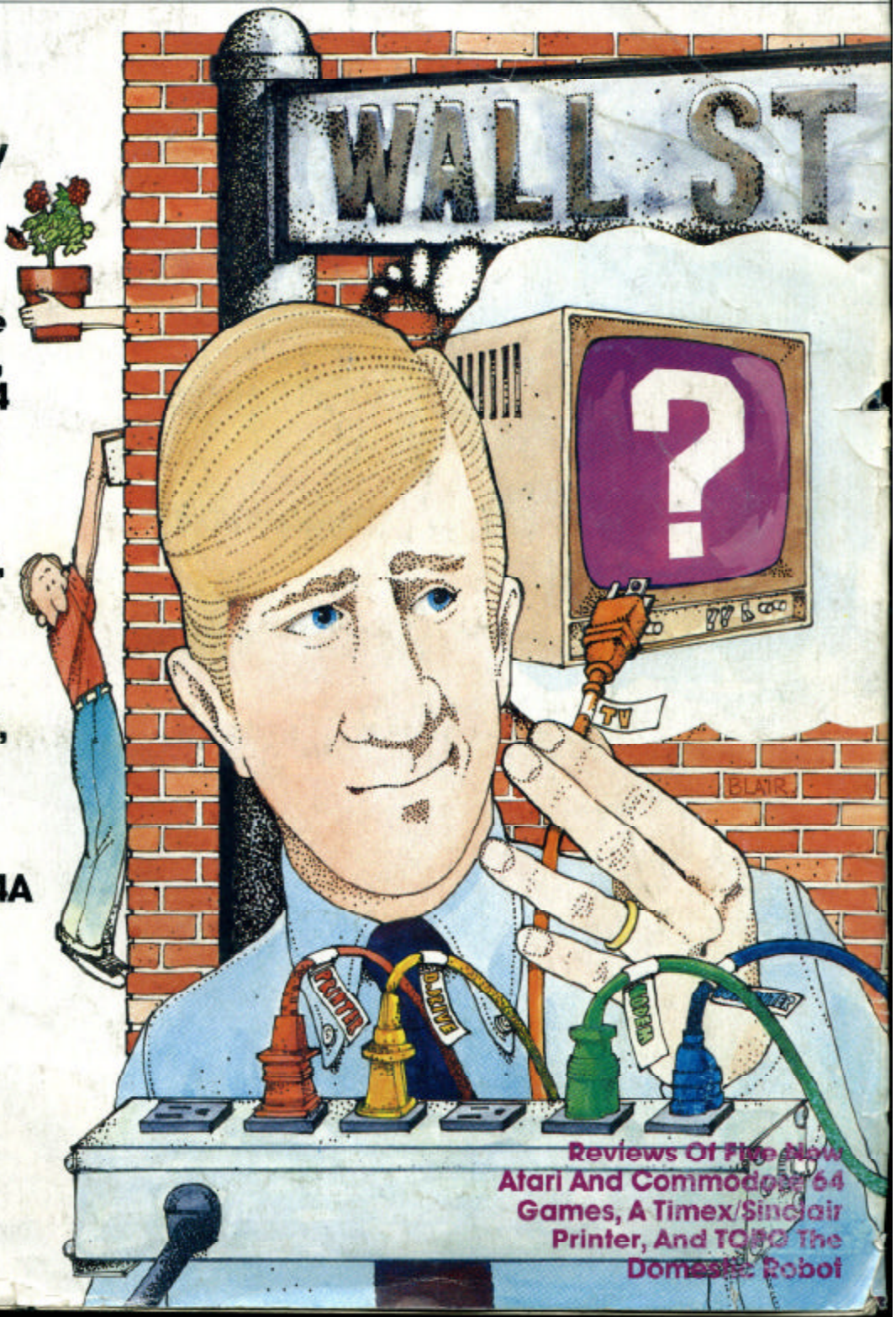
How To Add Peripherals And Choose The Right TV For Your Computer

***Climber*: A Fast-paced Game For VIC-20, TI-99/4A, And Commodore 64**

***Goodbye Charlie*: An Exciting Game For Commodore 64, Atari, And VIC-20**

Stock Market Analyzer For VIC-20, Commodore 64, And Atari

All About The TI-99/4A Character Set



Reviews Of Five New Atari And Commodore 64 Games, A Timex/Sinclair Printer, And TOPO The Domestic Robot

11
74470 63379

November 1983 Vol. 5, No. 11

FEATURES

EDUCATION AND RECREATION

REVIEWS

COLUMNS AND DEPARTMENTS

THE JOURNAL

NOTE: See page 329 before typing in programs.

GUIDE TO ARTICLES AND 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, *All or several of the above.

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

COMPUTE! Publications, Inc. abc
A Subsidiary Of American Broadcasting Companies, Inc.

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 12 times each year by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 505 Edwardia Drive, Greensboro, NC 27409. Domestic Subscriptions: 12 issues, $20.00. Send subscription orders or change of address (P.O. form 3579) to Circulation Dept., **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X

Publisher	Gary R. Ingersoll
Editor In Chief	Robert C. Lock
Publisher's Assistant	Alice S. Wolfe
Senior Editor	Richard Mansfield
Managing Editor	Kathleen E. Martinek
Assistant Managing Editor	Tony Roberts
Production Editor	Gail Walker
Features Editor	Tom R. Halfhill
Technical Editor	Ottis R. Cowper
Program Editor	Charles Brannon
Assistant Editors	Dan Carmichael Lance Elko
Assistant Copy Editor	Juanita Lewis
Editorial Assistant	Kathy Yakal
Programming Supervisor	Patrick Parrish
Assistant Programming Supervisor	Gregg Peele
Technical Assistant	Dale McBane
Administrative Assistants	Vicki Jennings Laura MacFadden Julia Fleming
Copy Assistants	Becky Hall Linda Shaw Martha Banks
Associate Editors	Jim Butterfield, Toronto, Canada Harvey Herman, Greensboro, NC Fred D'Ignazio, 2117 Carter Road, S.W., Roanoke, VA 24015 David Thornburg, P.O. Box 1317, Los Altos, CA 94022
Contributing Editor	Bill Wilkinson
COMPUTE!'s Book Division Editor	Orson Scott Card
Assistant Editors	Stephen Levy Gregg Keizer
Artists	Janice Fary Debbie Bray
Art Director/Production Manager	Georgia Bikas Davis
Assistant	Irma Swain
Mechanical Art Supervisor	De Potter
Artists	Leslie Jessup Cindy Mitchell
Typesetting	Terry Cash Debi Thomas
Illustrator	Harry Blair
Promotion Assistant	Todd Heimarck
Associate Publisher/National Advertising Sales Manager	Andy Meehan
Advertising Coordinator	Patti Williams
Advertising Accounts	Bonnie Valentino
Sales Assistant	Rosemarie Davis
Operations/Customer Service Manager	Carol Lock
Customer Representatives	Patty Jones Chris Patty Christine Gordon
Dealer Coordinator	Fran Lyons
Assistants	Gail Jones Sharon Minor
Individual Order Supervision Assistants	Dorothy Bogan Rhonda Savage Lisa Flaharty Anita Roop Sharon Sebastian Debi Goforth
Operators	Cassandra Robinson Mary Sprague
Shipping & Receiving	Jim Coward Larry O'Connor Dal Rees Chris Cain John B. McConnell
Data Processing Manager	Leon Stokes
Assistant	Joan Compton
Accounting Manager	W. Jerry Day
Comptroller	James M. Hurst
Accounting Assistants	Linda Miller Doris Hall
Assistants	Ruth Granger Anna Harris Emilie Covil Anne Ferguson

Robert C. Lock, Chief Executive Officer
Gary R. Ingersoll, President
W. Jerry Day, Vice President of Finance
James M. Hurst, Comptroller
Debi Nash, Executive Assistant
Carol Dickerson, Assistant

Member ABC Audit Bureau of Circulations

COMPUTE! Publications, Inc. publishes:

COMPUTE!
The Microcomputer Programmer Magazine
COMPUTE! Books

Telephone: 919-275-9809

Corporate office:
505 Edwardia Drive,
Greensboro, NC 27409 USA

Mailing address: COMPUTE!
Post Office Box 5406
Greensboro, NC 27403 USA

Subscription Information

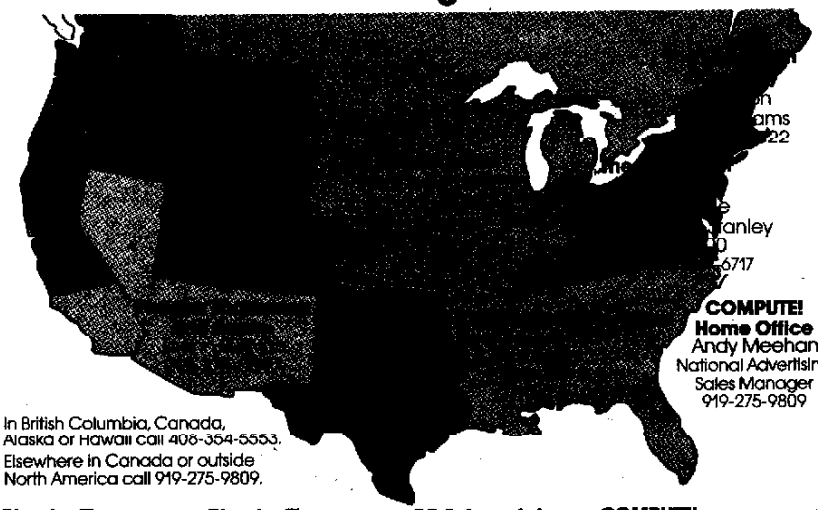
COMPUTE! Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403

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

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.) $20	Air	
	(two yrs.) $36	Europe, Australia	$38
	(three yrs.) $54	Middle East, Central America and North Africa	$48
Canada and Foreign Surface Mail	$25	South America, South Africa, Far East	$68

Advertising Sales



COMPUTE! Home Office
Andy Meehan
National Advertising Sales Manager
919-275-9809

In British Columbia, Canada, Alaska or Hawaii call 408-354-5553.
Elsewhere in Canada or outside North America call 919-275-9809.

Phoebe Thompson and Associates
101 Church Street
Suite 13
Los Gatos, CA 95030
PHOEBE THOMPSON

Phoebe Thompson and Associates
2556 Via Tejon
Palos Verdes Estates, CA 90274
JOANN SULLIVAN

GB & Associates
P.O. Box 335
Libertyville, IL 60048
GORDON BENSON

COMPUTE! Home Office
505 Edwardia Drive
Greensboro, NC 27409
HARRY BLAIR
Southeastern Sales Representative

The Gittelman Company
Statler Office Building
Suite 518
20 Providence Street
Boston, MA 02116
AL DALTON
RUTH WILLIAMS

The Gittelman Company
Summit Office Center
7266 Summit Avenue
Fort Washington, PA 19034
SHARON BRODIE
JOE PORTER
MIKE STANLEY

Address all advertising materials to:
Patti Williams
Advertising Production Coordinator
COMPUTE! Magazine
505 Edwardia Drive
Greensboro, NC 27409

READERS' FEEDBACK

The Editors and Readers of COMPUTE!

Tone Generator For The VIC

I own a VIC-20 and I am very pleased with it. However, I would like to know if it's possible to add a tone generator so that when I hit a key on the keyboard a tone is heard to indicate that an entry was made. I've seen this on other computers and find it most useful.

Kevin M. Regenhard

The positive stroke keyboard tone generator you mention is not built into the VIC-20. However, it is possible to program this useful function into your VIC.

Type in, SAVE, and then RUN the following short BASIC loader which will POKE in a machine language program. The program is written to run in the cassette buffer, so it shouldn't interfere with your BASIC programming memory. Once the program is POKEd into memory, SYS828 to start, and press RUN/STOP – RESTORE to stop.

```
60000 FOR A=828 TO 861: READ B: POKE A,B:
      NEXT: END
60010 DATA169,15,141,14,144,120,169,78,14
      1,20,3,169,3,141,21,3,88,96
60015 DATA165,197,201,128,240
60020 DATA7,101,197,105,128,141,12,144,76
      ,191,234
```

The Atari Mystery Connection

I took apart the Atari 400 and noticed on the back of the main board chip that there is a connection not in use. What is this connection for and why is it not in use?

Ki Jeong Yun

This connection, also found on the Atari 800, is used by Atari to test the machines on the assembly line. Due to its inaccessibility, it has never had any other use, although it could possibly be used for expansion.

Double-spaced Listings On Commodore Printers

I have a Commodore 64 and a Commodore 1525E printer. There are many occasions while I'm debugging a new program when a double-spaced paper listing would provide a lot more room to make corrections and additions.

Is there a way to force the LIST command to double-space on the printer without modifying the program being listed? Perhaps Jim Butterfield could suggest a short machine language routine.

Stephen D. Eitelman

Yes, it is possible to command the printer to double-space during the LIST command. In fact, all Commodore printers and many other printers have this ability. You will also, of course, see double-spaced listings on your screen. With any Commodore printer, you can enter and run one of these short BASIC programs, which will POKE a machine language program into the cassette buffer to create double-spacing.

```
5 AD = PEEK(55)+PEEK(56)*256-20
10 I=AD
15 POKE 55,AD AND 255 : POKE 56,AD/256
20 READ A:IF A=256 THEN END
25 IF A=-1THENA=(AD+11)AND255
26 IF A=-2THENA=(AD+11)/256
30 POKE I,A:I=I+1:GOTO 20
35 PRINT"{CLR}RUN WITH : {RED}SYS"AD
40 CLR
828 DATA 169,-1,141,38,3
834 DATA 169,-2,141,39,3
840 DATA 96,201,13,208,5
846 DATA 32,122,242,169,13
852 DATA 76,122,242,256
```

```
5 PRINT"{CLR}RUN WITH SYS 679"
10 I=679
20 READ A:IF A=256 THEN END
30 POKE I,A:I=I+1:GOTO 20
679 DATA 169,178,141,38,3,169,2
687 DATA 141,39,3,96,201,13,208
695 DATA 5,32,202,241,169,13,76
703 DATA 202,241,256
```

TI Free Memory Techniques

I would like to clarify and expand upon the ideas expressed by Howard Patlik in "More on TI Memory" (Readers' Feedback, August 1983). This two-line program for determining free memory on the TI-99/4A was offered:

```
1 A=A+8
2 GOSUB 1
```

If you RUN this program alone and PRINT the variable A, you will find the amount of available memory less the 37 bytes required by the program itself. (A equals 14536, so A plus 37 equals 14573 bytes of memory for programming.) When the program RUNs, the GOSUB in line 2 will ex-

ecute 1817 times before a MEMORY FULL error is issued in line 1. Each time this GOSUB executes, 8 bytes of memory are consumed and thus 1817 times 8 is 14536. As many as 7 bytes may still be unused.

Of course, if this two-line routine is entered with a program already in memory (providing the variable A is not used within this program), you will find the amount of free memory less 37 bytes when you RUN and PRINT A.

There are some more considerations. String variables and graphic characters defined above CHR$(127) will require additional memory that this two-line routine will not pick up. The TI-99/4 has 256 more free bytes than the TI-99/4A, but if graphic characters above 127 are used, then both will use the same amount of memory.

Screen and BASIC overhead is 1792 bytes. And if all or only the graphics character 159 is used, then overhead is 2048 bytes (1792 + 8*(159-127)). The 16K TI-99/4A is, of course, actually 16,384 bytes. So, 16,384 minus 1792 is 14,592 bytes free. (There is a slight discrepancy from the two-line method of up to 19 bytes.)

Here is another way to find memory size using CALL PEEK. For TI BASIC, you need the Mini Memory or the Editor/Assembler cartridges. Type in CALL PEEK(-31974,A,B). Again the variables A and B should not be in the program. Then PRINT A*256+B-1776. With no program present, this will give a size of 14577 bytes free.

To check this with TI Extended BASIC (Version 110), type PRINT A*256+B-2455. The TI's response is 13886. Now type SIZE. The TI responds with 13886 BYTES FREE.

Another user of memory in TI BASIC is Terminal Emulator II. It takes about 512 bytes. It is also well-known that the disk controller uses console memory. Even after ENTERing CALL FILES(1) and the NEW command, it still uses 1052 bytes. Here's how to free up this memory. Type CALL LOAD(-31888,63,255) and NEW.

Again, in TI BASIC the Mini Memory or Editor/Assembler cartridge is needed, but in TI Extended BASIC, the 32K Expansion is needed. To restore disk drive(s) and disk controller, use the command BYE or FCTN QUIT.

Paul E. Schippnick

Atari Upgrade Update

We have recently heard from several factory authorized Atari service center representatives who tell us that there is an inexpensive (under $30) upgrade for the 10K ROM operating system. The original operating system of the 400/800 had a number of minor errors (bugs), and the Revision B operating system corrected these problems. This is the first time to our knowledge that Atari has offered this upgrade. To check which operating system you have, enter: PRINT PEEK(58383). If you get a 56, you may want to visit your nearest service center and get the upgrade (a value of 0 returned means you already have Revision B).

Another Perfect Commodore INPUT

In the last several issues I have seen a number of "Perfect INPUTS" to avoid Commodore's return to READY from input. All of these methods work. In my opinion, however, since you can continue from READY by entering CONT on a clear line, their disadvantages outweigh their advantages.

I have another way of avoiding this problem. It uses the standard INPUT statement and CBM's active screen. When an input statement is executed, CBM BASIC prints a ? at the current cursor position, then moves the cursor to the right one additional space. All of the positions to the right of the cursor are automatically allocated for inputting data (up to 80 characters). By printing some default value into this area before executing the input statement, not only do you avoid the return to READY, but you also allow for inputting default values by just hitting RETURN.

For example:

```
10 VA = 10 : REM SET DEFAULT VALUE TO 0
20 PRINT "WHAT IS THE NEW VALUE": PRINT
   " "; VA;"{UP}"
30 INPUT VA
40 PRINT "THE CURRENT VALUE IS"; VA
50 INPUT "IS THE VALUE CORRECT (Y/N)
   {3 SPACES}Y{3 LEFT}"; A$
60 PRINT "WHAT IS THE VALUE"; TAB(20);"D
   EFAULT VALUE"
70 PRINT TAB(18);"{UP}";
80 INPUT DV$
```

This appears on the screen as:

```
WHAT IS THE NEW VALUE
? 10
THE CURRENT VALUE IS 10
IS THE VALUE CORRECT [Y/N] ? Y
WHAT IS THE VALUE ? DEFAULT VALUE
```

The flashing cursor is positioned over the 1, Y, and D respectively.

By hitting only the RETURN key (3 times), you INPUT 10 to VA, Y to A$, and DEFAULT VALUE to DV$. If you want some other value, you need only type it in before hitting RETURN.

By taking a little time in choosing default values when writing a program, you can save a lot of time when running it and entering data.

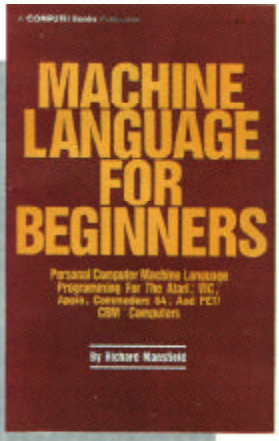
Dennis D. Duke

Atari Listings

How can hard copy be printed from RAM memory which has been loaded from a user-made BASIC program on a cassette? I realize that this is not too difficult a problem, but I have not found a solution

COMPUTE! Books

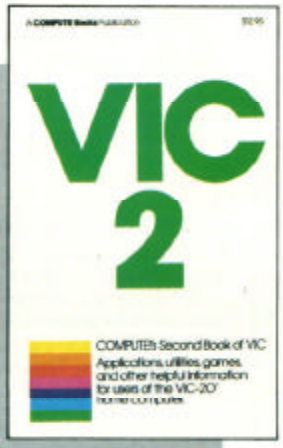
AUTUMN 1983



$14.95
ISBN 0-942386-11-6

Machine Language For Beginners. Much commercial software is written in machine language because it's faster and more versatile than BASIC. *Machine Language For Beginners* is a step-by-step introduction to the subtleties of machine code. Includes an assembler, a disassembler, and utilities, to help beginners write programs more quickly and easily. Covers many popular home computers.

350 pages, paperback.
Spiral bound for easy access to programs.



$12.95
ISBN 0-942386-16-7

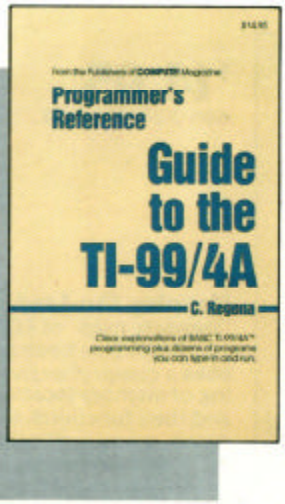
COMPUTE!'s Second Book Of VIC. This is just the book to follow the bestselling *First Book of VIC:* clear explanations of programming techniques, an extensive memory map, a mini-word processor, a system for creating sound effects, a custom character maker, a machine language assembler, and "Snake," an extraordinary all-machine language game.

274 pages, paperback.
Spiral bound for easy access to programs.

$12.95
ISBN 0-942386-20-5

COMPUTE!'s First Book Of Commodore 64. An excellent resource for users of the 64, with something for everyone: BASIC programming techniques, a memory map, a machine language monitor, and information about writing games and using peripherals. Many ready-to-type-in programs and games.

264 pages, paperback.
Spiral bound for easy access to programs.

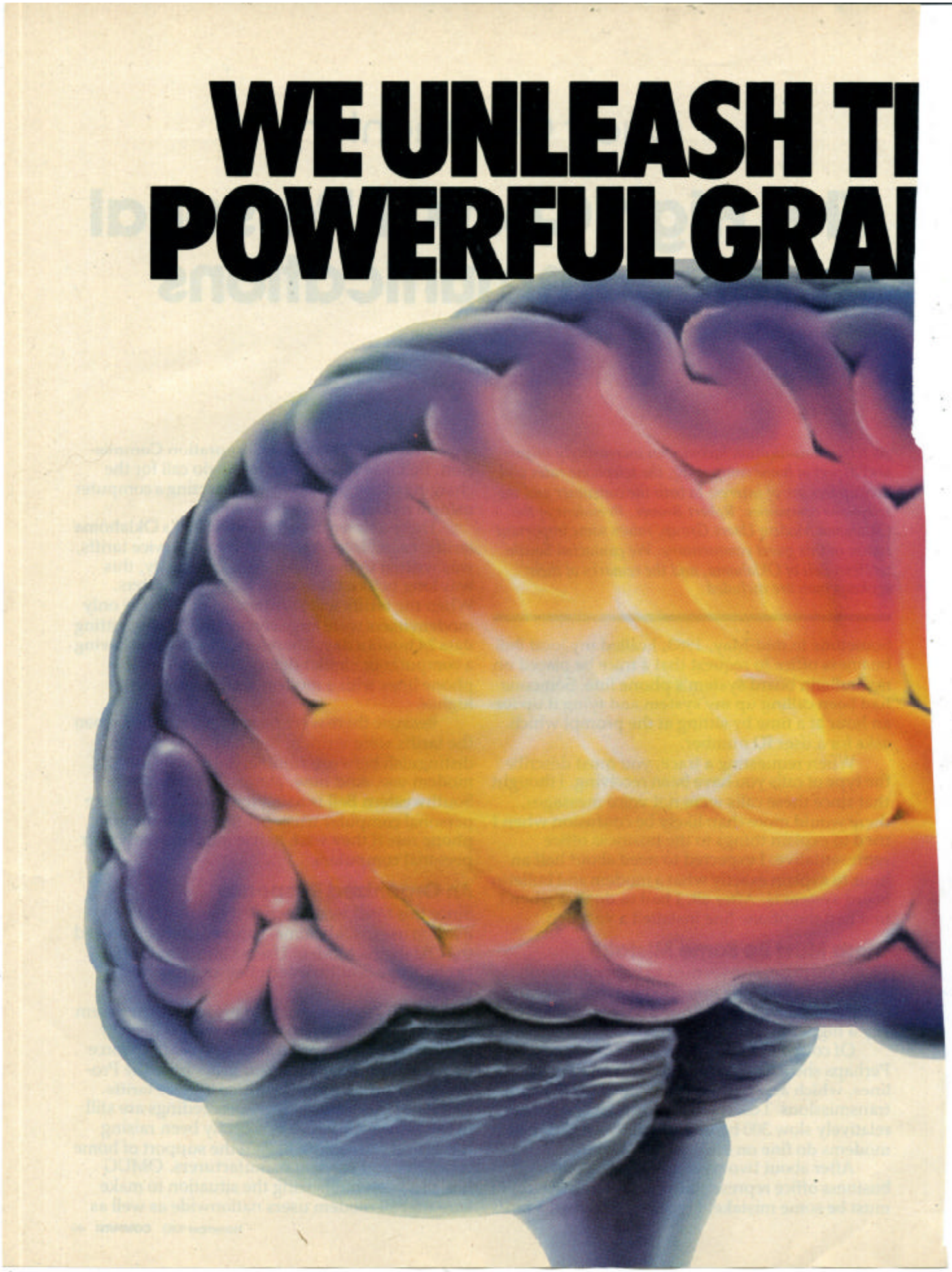


$14.95
ISBN 0-942386-12-4

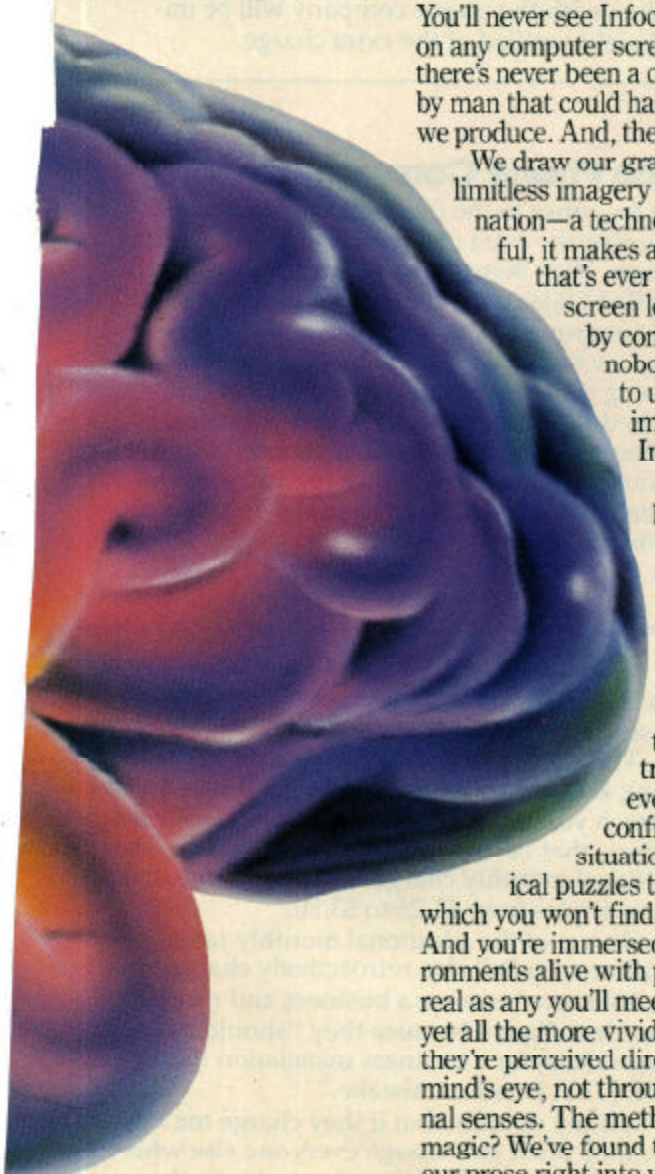
Programmer's Reference Guide To The TI-99/4A. Useful for everyone from beginners to experienced programmers. Contains instructions for programming in BASIC, plus more than 40 ready-to-type-in programs for this popular computer. Explanations and examples of variables, files, arrays, high-resolution graphics, music, speech, editing and debugging, conserving memory, and more.

312 pages, paperback.
Spiral bound for easy access to programs.

WE UNLEASH TI
POWERFUL GRAI



WORLD'S MOST
ICS TECHNOLOGY.



You'll never see Infocom's graphics on any computer screen. Because there's never been a computer built by man that could handle the images we produce. And, there never will be.

We draw our graphics from the limitless imagery of your imagination—a technology so powerful, it makes any picture that's ever come out of a screen look like graffiti by comparison. And nobody knows how to unleash your imagination like Infocom.

Through our prose, your imagination makes you part of our stories, in control of what you do and where you go—yet unable to predict or control the course of events. You're confronted with situations and logical puzzles the like of which you won't find elsewhere. And you're immersed in rich environments alive with personalities as real as any you'll meet in the flesh—yet all the more vivid because they're perceived directly by your mind's eye, not through your external senses. The method to this magic? We've found the way to plug our prose right into your psyche, and catapult you into a whole new dimension.

Take some tough critics' words about our words. SOFTALK, for example, called ZORK® III's prose "far more graphic than any depiction yet achieved by an adventure with graphics." And the NEW YORK TIMES saw fit to print that our DEADLINE™ is "an amazing feat of programming." Even a journal as video-oriented as ELECTRONIC GAMES found Infocom prose to be such an eye-opener, they named one of our games their Best Adventure of 1983.

Better still, bring an Infocom game home with you. Discover firsthand why thousands upon thousands of discriminating game players keep turning everything we write into instantaneous bestsellers.

Step up to Infocom. All words. No graffiti. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

INFOCOM™

The next dimension.

Infocom, Inc., 55 Wheeler St., Cambridge, MA 02138

For your: Apple II, Atari, Commodore 64, CP/M 8", DEC Rainbow, DEC RT-11, IBM, NEC APC, NEC PC-8000, Osborne, TI Professional, TRS-80 Model I, TRS-80 Model III.

Crazy Climber

Ted Reynolds

Harry Blair

As you scale the side of a building, maneuvering around windows, watch out for the falling flowerpots and attacking birds. Originally written on the VIC with joystick, versions are included for the 64 (with joystick) and the TI-99/4A with Extended BASIC.

"Crazy Climber" is a game requiring manual dexterity and judgment. With a joystick, you control the Crazy Climber as he scales the side of a brick building, avoiding windows and falling objects. The higher he climbs, the higher your score.

The VIC Version

First, type in and save Program 1. Then type in Program 2 and save it immediately following Program 1 on the same tape. When Program 1 is RUN, it will cause Program 2 to be loaded from tape and RUN automatically.

When the game starts, you will see a demonstration of the Crazy Climber in action. After this brief display, he will come to a stop, and you can start play by pressing the fire button.

If the Crazy Climber falls, the game ends and your score is displayed. Simply press the fire button to start another game. The climber will fall if *any* contact is made with a window. It's easy going until your score reaches 100. At this point, you'll have to contend with falling flowerpots. At 300, the flowerpots will stop falling, but you'll have to watch out for birds flying from the left side of the screen. If you reach 600, you've made it to the top of the building, and you start at the bottom of the next building.

The 64 Version

In this version, there are some major differences in play. First, plug the joystick into Port 1. Hit the space bar to start the game. Unlike the VIC version, you can climb onto a window as long as you have some contact with the wall. A variety of objects are tossed down at you – TVs, pianos, barbells, safes (it's one of those wild and crazy apartment buildings) – and they come twice as fast after your score reaches 150.

This version also includes a high score feature. Pressing the space bar will start a new game.

If you'd rather not type in the program, I'll make a copy (VIC version only) if you send a blank tape or disk, SASE, and $3 to:

Ted Reynolds
145 North Broadway #18
Tooele, UT 84074

```
1780 DATA 216,133,252,169,0,133,251
1790 DATA 169,10,160,0,162,4,145
1800 DATA 251,200,208,251,230,252,202
1810 DATA 208,246,169,0,141,33,208
1820 DATA 173,1,220,41,15,201,7
1830 DATA 208,36,173,16,208,41,1
1840 DATA 240,7,173,0,208,201,60
1850 DATA 176,22,24,173,0,208,105
1860 DATA 16,141,0,208,173,16,208
1870 DATA 105,0,141,16,208,169,1
1880 DATA 141,188,2,173,1,220,41
1890 DATA 15,201,11,208,36,173,16
1900 DATA 208,41,1,208,7,173,0
1910 DATA 208,201,32,144,22,56,173
1920 DATA 0,208,233,16,141,0,208
1930 DATA 173,16,208,233,0,141,16
1940 DATA 208,169,1,141,188,2,96
1950 DATA 76,24,192,24,169,40,109
1960 DATA 254,207,141,254,207,169,0
1970 DATA 109,255,207,141,255,207,56
1980 DATA 173,254,207,233,0,133,2
1990 DATA 173,255,207,233,52,5,2
2000 DATA 144,10,169,48,141,255,207
2010 DATA 169,0,141,254,207,76,24
2020 DATA 192,120,169,122,141,20,3
2030 DATA 169,193,141,21,3,88,96
2040 DATA 165,161,166,161,142,0,207
2050 DATA 165,161,205,0,207,240,5
2060 DATA 169,0,141,1,207,238,1
2070 DATA 207,173,1,207,141,3,208
2080 DATA 173,0,198,208,9,238,1
2090 DATA 207,173,1,207,141,3,208
2100 DATA 76,49,234,256
```

Program 4: Crazy Climber For The TI-99/4A

Extended BASIC Version by Pat Parrish, Programming Supervisor

```
100 DIM D(7),E(7),G(4),C$(4)
110 GOTO 140
120 REM RANDOMLY PICK WINDOW & PRIN
    T SINISTER MAN
130 V=INT(RND*8):: CALL DELSPRITE(#
    1):: CALL SPRITE(#4,96,2,D(V)*8
    +1,E(V)*8+1):: R=-1 :: Q=-1 ::
    RETURN
140 HS=0 :: RANDOMIZE
150 GOSUB 480
160 GOSUB 790
170 GOSUB 670
180 T=0 :: U=.1 :: Q=0 :: R=0 :: SC
    =0 :: B=112 :: Z=-3 :: ROW=13 :
    : COL=15
190 CALL SPRITE(#2,B,2,ROW*8+1,COL*
    8+1)
200 IF R THEN CALL SPRITE(#1,G(INT(
    RND*5)),INT(RND*14)+3,(D(V)+4)*
    8+1,E(V)*8+1,15,0):: SC=SC+10 :
    : R=0 :: CALL DELSPRITE(#4)
210 IF (RND<U)*(R=0)*(Q=0)THEN GOSU
    B 130
220 CALL KEY(0,K,ST)
230 IF K=69 THEN ROW=ROW-1 :: IF (R
    OW=-1)THEN ROW=23 :: SC=SC+100
    :: U=U+SGN(1-U)/20 :: GOTO 280
    ELSE 280
240 IF K=83 THEN COL=COL-SGN(COL-2)
    *2 :: GOTO 280
250 IF K=68 THEN COL=COL+SGN(26-COL
    )*2 :: GOTO 280
260 IF K=88 THEN ROW=ROW+1 :: IF (R
    OW=24)THEN ROW=0 :: SC=SC-100 :
    : GOTO 280 ELSE 280
270 GOTO 290
280 B=228-B :: Z=197-Z :: CALL SPRI
    TE(#2,B,2,ROW*8+1,COL*8+1):: CA
    LL SOUND(10,Z,2)

290 CALL COINC(ALL,C1):: IF C1 THEN
     320
300 CALL POSITION(#1,XROW,XCOL):: I
    F XROW<180 THEN 200
310 CALL DELSPRITE(#1):: Q=0 :: GOT
    O 200
320 T=1 :: V=2 :: CALL DELSPRITE(#1
    ):: CALL SAY("UHOH"):: REM REMO
    VE "CALL SAY" IF W/OUT SPEECH S
    YNTHESIZER
330 IF TP THEN T=40
340 CALL SPRITE(#2,108,2,ROW*8+1,CO
    L*8+1,25,0):: FOR I=1 TO 95 ::
    NEXT I :: CALL MOTION(#2,10,0)
```

TI-99/4A Version Notes

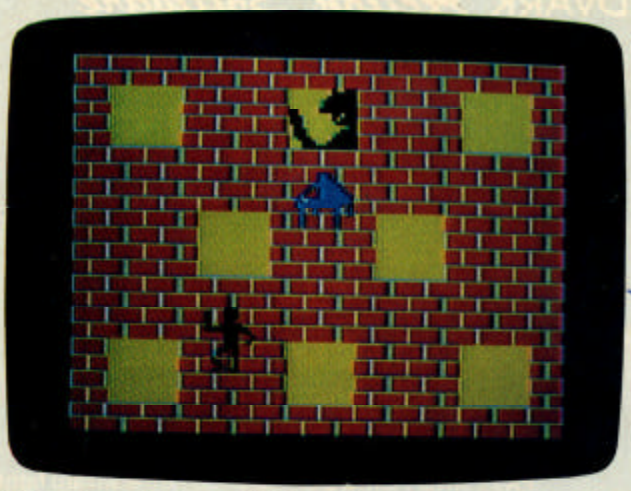
Pat Parrish, Programming Supervisor

In the TI-99/4A version of this game (written in Extended BASIC), you are the Crazy Climber, scrambling up the face of a building while avoiding numerous objects (piano, iron, broom, safe, barbells) tossed from the windows above. These objects are actually *hurled* down upon you by a relentless, sinister fellow who appears just briefly before throwing each object. If you are quick, you can dodge these oncoming objects. No one knows why he throws things; it's a quirk. Be ready to meet the challenge. For as the game progresses, the villain strikes with greater frequency.

In this game, you move the Crazy Climber over a stationary building with the E, S, D, and X keys. The screen will wrap around when you reach the top or bottom. A hundred points are awarded for crossing the top of the screen, while an equal number are deducted for crossing the bottom. In addition, ten points are given for each falling object that you avoid.

The game ends when you are hit by a falling object or are pushed from a window by the villain. Thus, you are allowed to climb over windows in this version, but you're taking a chance. If the villain emerges while you are in a window, it's curtains for you. (You'll be relieved to discover that the Crazy Climber carries a parachute.)

One line in this program requires that you have a TI Speech Synthesizer connected to your TI-99/4A. If you don't have this peripheral, remove the CALL SAY("UHOH") statement in line 320.



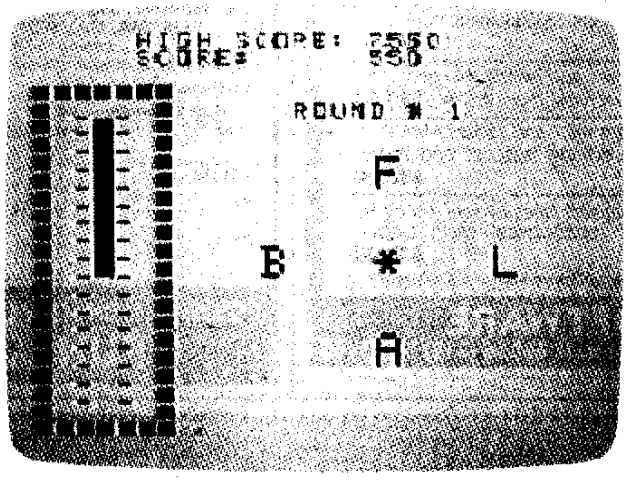
TI version of "Crazy Climber."

```
350 CALL SOUND(-50,-7,2):: CALL SPR
    ITE(#3,104,16,(ROW+2)*8+1,COL*8
    +1,10,0)
360 CALL SOUND(80,5000-T*40,V):: T=
    T+1 :: IF T<65 THEN 360
370 CALL POSITION(#1,DR,DC,#3,CR,CC
    ):: IF DR>192 THEN CALL DELSPRI
    TE(#1)
380 IF CR>165 THEN CALL DELSPRITE(#
    2,#3):: GOTO 400
390 GOTO 360
400 IF TP THEN GOTO 910
410 CALL DELSPRITE(ALL):: FOR L=1 T
    O 100 :: NEXT L :: CALL CLEAR :
    : CALL SCREEN(14)
420 DISPLAY AT(10,5):"YOUR SCORE :
    ";SC
430 IF SC>HS THEN HS=SC
440 DISPLAY AT(13,5):"HIGH SCORE :
    ";HS
450 DISPLAY AT(16,5):"PLAY AGAIN ?
    " :: ACCEPT AT(16,19)BEEP VALID
    ATE("YN")SIZE(1):ANS$
460 IF ANS$="Y" THEN 170
470 STOP
480 REM DEFINE CUSTOM CHAR
490 FOR I=120 TO 121
500 READ A$
510 CALL CHAR(I,A$)
520 NEXT I
530 DATA FF80808080808080,FF0101010
    1010101
540 CALL MAGNIFY(4):: CALL CHAR(100
    ,"")
550 CALL CHAR(112,"03232321273F0303
    0303060810080409808080000C0E0908
    8908080808080808080")
560 CALL CHAR(116,"0101010003070911
    0901010101010101C0C4C484E4FCC0C
    0C0C0601008102090")
570 FOR I=0 TO 4 :: READ C$(I):: G(
    I)=124+4*I :: CALL CHAR(G(I),C$
    (I)):: NEXT I
580 DATA 2070FF7020000000000000000
    0000000040EFF0E0400000000000000
    0000000
590 DATA 00000000F3F3F3F3F3F3F3F3F
    3F303000000000F8F8F9DAFCF8F8F8F
    8F81818
600 DATA 00000101030509112273F3F1212
    1010100000C0E0F0F8FCFEFEFAF2121
    2101010
610 DATA 0300000F0F0F1F0000000000000
    000000E02020E0F0F0F8000000000000
    0000000
620 DATA 0000000000000000001031E3C7C
    FCF8700103060C183060C0800000000
    0000000
630 CALL CHAR(104,"01070F1F1F151008
    0804040202010100 80E0F0F8F858081
    0102020404080800 0")
640 CALL CHAR(108,"0204050701010101
    0202020600000000804040C080808080
    0404040600000000")
650 CALL CHAR(96,"000000000001C1C0E
    0E0607070381C1F273F7FFFFFFFFF37
    2F1F030F1F3F7FFF")
660 RETURN
670 CALL COLOR(12,1,1):: CALL COLOR
    (9,1,1):: REM SET UP WALL
680 CALL CLEAR :: CALL SCREEN(2)::
    ROW$="xyxyxyxyxyxyxyxyxyxyxyxyxy
    xyxy"
690 ROW2$="yxyxyxyxyxyxyxyxyxyxyxyxyx
    yxyx"
700 FOR ROW=1 TO 23 STEP 2 :: DISPL
    AY AT(ROW,1):ROW$ :: NEXT ROW
710 FOR ROW=2 TO 24 STEP 2 :: DISPL
    AY AT(ROW,1):ROW2$ :: NEXT ROW
720 FOR ROW=3 TO 6 :: FOR COL=5 TO
    25 STEP 10 :: CALL HCHAR(ROW,CO
    L,100,4):: NEXT COL :: NEXT ROW
730 FOR ROW=19 TO 22 :: FOR COL=5 T
    O 25 STEP 10 :: CALL HCHAR(ROW,
    COL,100,4):: NEXT COL :: NEXT R
    OW
740 FOR ROW=11 TO 14 :: FOR COL=10
    TO 20 STEP 10 :: CALL HCHAR(ROW
    ,COL,100,4):: NEXT COL :: NEXT
    ROW
750 CALL COLOR(12,15,7):: CALL COLO
    R(9,1,11)
760 E(0)=4 :: E(1)=14 :: E(2)=24 ::
     E(3)=9 :: E(4)=19 :: E(5)=4 ::
     E(6)=14 :: E(7)=24
770 D(0)=2 :: D(1)=2 :: D(2)=2 :: D
    (3)=10 :: D(4)=10 :: D(5)=18 ::
     D(6)=18 :: D(7)=18
780 RETURN
790 REM TITLE PAGE
800 CALL CLEAR :: CALL SCREEN(15)
810 ROW$="xyxyxyxyxy" :: ROW2$="yxy
    xyxyxyx"
820 CALL COLOR(12,15,15)
830 FOR ROW=7 TO 23 STEP 2 :: DISPL
    AY AT(ROW,2):ROW$ :: NEXT ROW
840 FOR ROW=8 TO 24 STEP 2 :: DISPL
    AY AT(ROW,2):ROW2$ :: NEXT ROW
850 CALL COLOR(12,15,7)
860 DISPLAY AT(6,19):"T H E" :: DIS
    PLAY AT(8,17):"C R A Z Y" :: DI
    SPLAY AT(10,15):"C L I M B E R"
870 FOR I=1 TO 100 :: NEXT I :: B=1
    16 :: Z=200 :: ROW=19 :: COL=6
880 CALL SPRITE(#2,B,2,ROW*8+1,COL*
    8+1):: CALL SOUND(10,Z,2)
```

```
890 B=228-B :: Z=197-Z :: ROW=ROW-1
    :: FOR I=1 TO 50 :: NEXT I ::
    IF ROW>1 THEN 880
900 TP=-1 :: GOTO 320
910 TP=0 :: DISPLAY AT(16,13):"USE
    E,S,X,D KEYS" :: DISPLAY AT(17,
    13):"TO AVOID FALLING"
920 DISPLAY AT(18,13):"OBJECTS." ::
     DISPLAY AT(20,16):"GOOD LUCK!"
930 FOR I=1 TO 1000 :: NEXT I :: CA
    LL DELSPRITE(#2):: RETURN
940 END
```

ALPHA BLAST

Dave Miller



Color Computer version of "Alpha Blast."

A test of judgment, speed, and accuracy, this game is a good educational tool for children and fun for adults. Originally written for the Atari, versions are also included for the VIC, TI-99/4A (with Extended BASIC), and Color Computer. Joystick required (optional for TI version).

This game is fun to play and will also sharpen your alphabetizing skills. How many times do you find yourself saying, "Q, R, S, T, U – yes, T comes before U"? I know I've often said it.

The object of "Alpha Blast" is to shoot the four letters displayed on the screen in alphabetical order. Sounds simple enough, but it isn't. And to make it more difficult, you're being timed. For each correct answer you are awarded points based on the internal character set value. Since I am using lowercase letter values, an A would give you 97 points, G would give you 103 points, and so on. If you fail to shoot a letter in the correct order, the value of that letter will be subtracted from your score. If you get all four letters right, you will advance to the next round with new letters to shoot and less time in which to do it. If time runs out before you complete a round, the game ends, giving you a final score and the high score.

This program uses a redefined character set for the gun you use to shoot the characters. Memory location 20 is the timer. You may safely remove lines which contain only REMarks (no GOTOs will reference them).

A challenge: See if you can get past round 25.

Program 1: Alpha Blast – Atari

```
100 GOSUB 630:REM TITLE ROUTINE
110 GOSUB 690:REM REDEFINE CHSET
120 DIM N(4):HIGH=0
130 ROUND=0:SCORE=0:TIME=50
140 GRAPHICS 18:POKE 756,CHSET/256:S
    ETCOLOR 2,6,5:SETCOLOR 0,0,10:SE
    TCOLOR 1,9,6:LASTNUM=0
141 POSITION 15,6:? #6;SCORE
145 REM **** USE INVERSE CONTROL 'F'
     IN 1ST PRINT
150 FOR T=1 TO 10:POSITION 2,T:? #6;
    "{F}":POSITION 1,T:? #6;"_":NEXT
     T
165 REM **** USE INVERSE '*' FOR PRI
    NT
170 POSITION 10,6:? #6;"*":SCRN=PEEK
    (88)+256*PEEK(89):ROUND=ROUND+1:
    POSITION 7,0:? #6;"ROUND ";ROUND
180 TIME=TIME-2:REM SET INCREMENT FO
    R TIMER
185 REM **** RANDOM LETTER GENERATOR
190 N(1)=INT(RND(0)*26)+97:POKE SCRN
    +46,N(1)
200 N(2)=INT(RND(0)*26)+97:POKE SCRN
    +54,N(2):IF N(2)=N(1) THEN 200
210 N(3)=INT(RND(0)*26)+97:POKE SCRN
```

```
385 NM=N(T):IF LM< NM THEN387
386 SC=SC-(NM+96):GOTO390
387 SC=SC+NM+96:LM=NM
390 ON T GOSUB400,410,420,430
395 LC=1048:GOSUB2000
396 IFLM> NM THENGOSUB205:GOSUB215:
    GOSUB225:GOSUB235:FORI=1TO4:TT(
    I)=1:NEXT
397 FORI=1TO4:QQ=QQ+TT(I):NEXT:IFQQ
    =4THEN100
398 QQ=0:GOTO350
400 FORI=41TO33STEP-1:SET(I,I-31,3)
    :NEXT:FORI=41TO33STEP-1:RESET(I
    ,I-31):NEXT:GOSUB205:GOTO440
410 FORI=50TO56:SET(I,60-I,3):NEXT:
    FORI=50TO56:RESET(I,60-I):NEXT:
    GOSUB215:GOTO440
420 FORI=41TO33STEP-1:SET(I,60-I,3)
    :NEXT:FORI=41TO33STEP-1:RESET(I
    ,60-I):NEXT:GOSUB225:GOTO440
430 FORI=50TO58:SET(I,I-31,3):NEXT:
    FORI=50TO58:RESET(I,I-31):NEXT:
    GOSUB235
440 TT(T)=1:T=0:F=0:RETURN
1000 CLS0:PRINT@46,"GAME";:PRINT@51
    ,"OVER";:PRINT@105,"YOUR";:PRI
    NT@110,"SCORE";:PRINT@116,"WAS
    ";
1010 LC=1142:GOSUB2000
1011 GOTO1011
2000 SC$=STR$(SC):FORI=2TOLEN(SC$):
    DD$=MID$(SC$,I,1):POKELC+I,VAL
    (DD$)+48:NEXT:POKELC+I,32:POKE
    LC+I+1,32:RETURN
10000 DATA 6,9,11,7,8,4,14,9,14,13,
    12,8,14,12,10,0,12,12,14,9,10
    ,5,12,8,14,12,14,8,12,12
10010 DATA 14,12,14,8,8,0,14,8,10,1
    3,12,12,10,5,14,13,8,4,13,14,
    5,10,12,12,4,14,2,10,4,8,10,6
    ,14,2,8,4
10020 DATA 10,0,10,0,12,12,9,6,10,5
    ,8,4,15,5,10,15,8,4,14,13,10,
    5,12,12,14,13,14,12,8,0
10030 DATA 14,13,10,6,12,4,14,13,14
    ,9,8,4,14,12,12,13,12,12,13,1
    4,5,10,4,8,10,5,10,5,12,12
10040 DATA 10,5,9,6,4,0,10,5,9,6,8,
    4,10,5,6,9,8,4,9,6,5,10,4,8,1
    2,14,6,0,12,12
```

Program 4: Alpha Blast — TI-99/4A

Extended BASIC Version by Pat Parrish, Programming Supervisor

```
100 GOSUB 510
110 RANDOMIZE
120 DIM N(3)
130 CALL CLEAR :: CALL SCREEN(16)
140 CALL HCHAR(8,5,120,24):: DISPLA
    Y AT(10,4):"A L P H A -- B L A
    S T" :: CALL HCHAR(12,5,120,24)
150 CALL MAGNIFY(2):: FOR L=1 TO 28
160 CALL SPRITE(#L,INT(RND*26)+65,I
    NT(RND*13)+3,INT(RND*24)*8+1,IN
    T(RND*32)*8+1,INT(RND*60)-30,IN
    T(RND*60)-30)
170 IF L<25 THEN DISPLAY AT(21,10):
    "GET READY!"
```

TI-99/4A Version Notes

Pat Parrish, Programming Supervisor

The object of the TI-99/4A version of Alpha Blast (written in Extended BASIC) is to indicate within a certain time the alphabetical order of four scrambled letters appearing on the screen. In its present form, the game requires a joystick. Moving the joystick in the direction of any letter will cause that letter to disappear. At the same time, the order of this response is recorded.

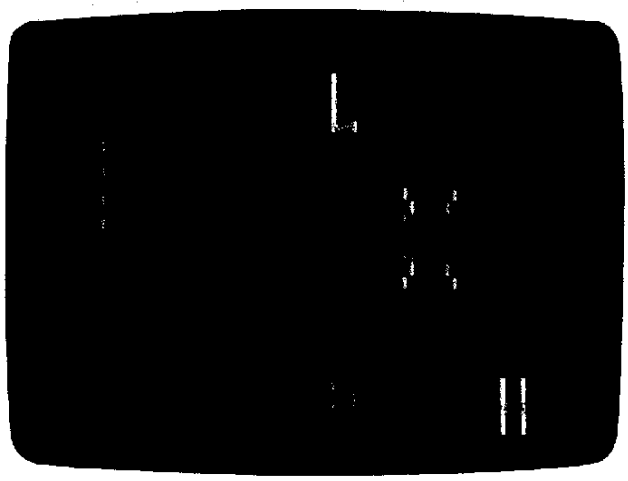
After you have guessed the sequence of the first four letters, a new round begins with the appearance of four new letters. As the game continues, you are given less and less time to respond. The game ends when you can no longer provide four responses within the allotted time.

Scoring for the game is calculated on lines 470 and 480. It is based on three factors: the round number, the time that it took you to respond, and the order of your answers. The faster you can provide the correct answers, the higher will be your score. For an incorrect response, points are deducted at a rate one and a half times the number awarded for correct answers.

To convert this game to keyboard control, substitute the following lines:

```
350 IF (X=A91$(A)THEN CALL PATTERN(
    #2,32,#6,43):: V(1)=A :: A=A+1
    :GOTO 390
360 IF (X=A91$(B)THEN CALL PATTERN(
    #3,32,#6,43):: V(2)=B :: B=B+1
    :GOTO 390
370 IF (X=A91$(C)THEN CALL PATTERN(
    #4,32,#6,43):: V(3)=C :: C=C+1
    :GOTO 390
380 IF (X=A91$(D)THEN CALL PATTERN(
    #5,32,#6,43):: V(4)=D :: D=D+1
    :GOTO 390
```

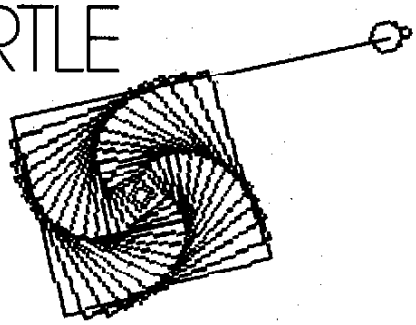
```
180 NEXT L :: CALL DELSPRITE(ALL)::
    CALL CLEAR :: HS=0
190 CALL COLOR(12,6,1)
200 DISPLAY AT(1,6):"HIGH SCORE:";H
    S :: U=0 :: R=0 :: SC=0
210 U=U+.03*SGN(1-U):: R=R+1 :: DIS
    PLAY AT(5,14):"ROUND #";R :: DI
    SPLAY AT(2,6):"SCORE:
    (5 SPACES)";SC
220 FOR I=6 TO 21 :: CALL HCHAR(I,6
    ,128):: NEXT I
230 FOR I=5 TO 7 STEP 2 :: CALL VCH
    AR(5,I,95,17):: NEXT I
240 FOR I=3 TO 9 STEP 6 :: CALL VCH
    AR(4,I,120,20):: NEXT I :: CALL
```



TI version of "Alpha Blast."

```
     HCHAR(4,4,120,5):: CALL HCHAR(
    23,4,120,5)
250 FOR I=0 TO 3
260 N(I)=INT(RND*26)+65
270 FOR J=0 TO I-1 :: IF N(J)=N(I)T
    HEN 260
280 NEXT J :: NEXT I
290 CALL SPRITE(#6,42,3,97,153)
300 CALL SPRITE(#2,N(0),14,57,153):
    : CALL SPRITE(#3,N(1),14,97,201
    ):: CALL SPRITE(#4,N(2),14,137,
    153):: CALL SPRITE(#5,N(3),14,9
    7,105)
310 ROW=21 :: A=-1 :: B=-1 :: C=-1
    :: D=-1
320 T=0
330 CALL JOYST(1,X,Y):: IF ABS(X)+A
    BS(Y)<>4 THEN CALL HCHAR(ROW,6,
    32):: ROW=ROW-U :: IF ROW<5 THE
    N 400 ELSE 330
340 IF (X=0)*(Y=4)*(A)THEN CALL PAT
    TERN(#2,32,#6,43):: V(T)=0 :: A
    =0 :: GOTO 390
350 IF (X=4)*(Y=0)*(B)THEN CALL PAT
    TERN(#3,32,#6,43):: V(T)=1 :: B
    =0 :: GOTO 390
360 IF (X=0)*(Y=-4)*(C)THEN CALL PA
    TTERN(#4,32,#6,43):: V(T)=2 ::
    C=0 :: GOTO 390
370 IF (X=-4)*(Y=0)*(D)THEN CALL PA
    TTERN(#5,32,#6,43):: V(T)=3 ::
    D=0 :: GOTO 390
380 CALL HCHAR(ROW,6,32):: ROW=ROW-
    U :: IF ROW<5 THEN 400 ELSE 330
390 CALL SOUND(-10,200,2):: CALL PA
    TTERN(#6,42):: T=T+1 :: IF T=4
    THEN 450 ELSE 330
400 DISPLAY AT(22,11):"YOUR TIME IS
     UP!"
410 CALL SOUND(800,110,5,120,5):: F
    OR I=1 TO 200 :: NEXT I
420 DISPLAY AT(24,10):"PLAY AGAIN(Y
    /N)?" :: IF SC>HS THEN HS=SC
430 CALL KEY(0,KEY,ST):: IF ST=0 TH
    EN 430
440 IF (KEY=89)+(KEY=121)THEN CALL
    CLEAR :: CALL DELSPRITE(ALL)::
    GOTO 200 ELSE 560
450 REM EVALUATE ANSWERS
460 FOR T=0 TO 2 :: IF N(V(T))<N(V(
    T+1))THEN 480
470 SC=SC-INT(1.5*R*ROW):: GOTO 490
480 SC=SC+INT(R*ROW)
490 NEXT T
500 CALL DELSPRITE(ALL):: GOTO 210
510 REM CHAR
520 CALL COLOR(14,9,1)
530 CALL CHAR(120,"007E7E7E7E7E7E00
    "):: CALL CHAR(128,"")
540 CALL COLOR(12,6,10):: CALL COLO
    R(13,1,9)
550 RETURN
560 END
```

FRIENDS OF THE TURTLE



David D Thornburg, Associate Editor

Bucky And The Turtle:
Exploring The Geometry Of Thinking

The philosopher, mathematician, inventor, and citizen of the whole earth – R. Buckminster Fuller – died this past July at the age of 86. Bucky influenced many people through his mathematical discoveries, and delighted millions more through his designs resulting from these discoveries. The geodesic dome stands as the most easily recognized of his creations.

While we can appreciate the results of his thought, it is especially valuable for readers of this column to acquaint themselves with his "geometry of thinking" – a philosophy called "Synergetics."

Synergetics is a mathematical formalism that, according to Fuller, embodies the design principles of the physical universe. His exposition of these principles formed the subject of two books, *Synergetics* and *Synergetics 2* (Macmillan, 1975 and 1979, respectively). While many people marvel at the beautiful simplicity of the geodesic dome or of the tensegrity structures Fuller discovered, few have taken the time to understand the underlying mathematical principles that led to the creation of these structures.

It so happens that the principles of Fuller's geometry are easily grasped once one realizes that Synergetics is identical to the mathematical formalism of turtle geometry.

Process Descriptions

In turtle geometry one deals with process descriptions rather than with static descriptions of geometric figures. The two operators (FORWARD and RIGHT) change the state of the turtle and can be used to move it anywhere on a surface. As a result, any static figure can be equivalently described by the process that created it. Process-based descriptions are central to Synergetics as well.

While it is impossible to do justice to the formalism of Synergetics in the short space of this article, several key concepts (and their equivalent expressions in turtle geometry) will be described. Each concept will be presented first from the perspective of Synergetics and then from the perspective of turtle geometry.

1. *There is no continuum.* There are no solid surfaces in the universe, no flat or smooth areas. Wherever scientists have looked, they have only uncovered localized energy fields which we perceive as discrete countable atoms. These atoms establish spatial relationships with other atoms through mutual optimization of their energy fields. The idea that the universe is composed of countable parts, that it is somehow granular, has an interesting expression in turtle geometry. Since the turtle responds to one command at a time, either it can move or it can turn. The fact that the turtle cannot turn while moving means that, in common with Synergetics, turtle geometry does not allow continuous curved surfaces.

2. *Measurements in geometry need only two parameters – frequency and angle.* These two parameters are sufficient to describe the location and placement of the nodes associated with the discrete quantized atoms which comprise the matter of our physical universe. The process by which one can move between any two nodes in the universe is capable of being expressed in terms of a combination of linear movements along nodes

and angular reorientations.

In turtle geometry, this central concept is expressed by the fact that combinations of the commands FORWARD and RIGHT are capable of repositioning the turtle to any desired location. Fuller's use of frequency instead of distance is a result of his desire to remove absolute scale from his geometry.

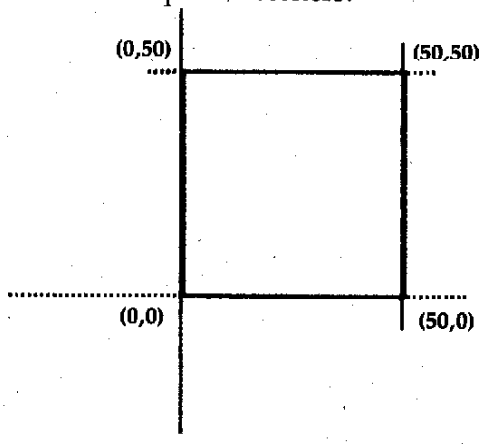
3. *There is no simultaneity.* The physical universe is an unfolding scenario of nonsimultaneous (but partially overlapping) energy events. The finite speed of light governs our perception of the physical universe. A pair of events that appear to be simultaneous to one observer will appear to be nonsimultaneous to a second observer at another location. Since nothing happens "all at once," then all events and structures are the result of a *process* which created them. Traces of completed events resulting from separate and distinct processes may appear similar to each other.

Simple Is Powerful

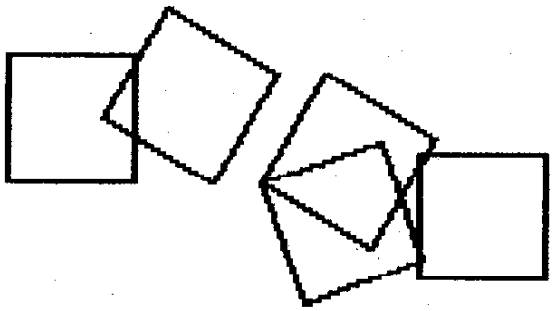
A problem with static descriptions of systems is that they do not preserve the details of the processes which created them. Since the process contains more information than the static trace of its result, a process description is inherently more fundamental. Furthermore, process descriptions are often more compact than static descriptions. This surprising result lends force to the idea that simpler descriptions are more powerful.

Turtle geometry *defines* objects through the description of the processes that create them. Computer-based implementations of turtle geometry allow the explicit creation of *procedures* that describe the steps needed to create various geometrical shapes. These procedures can often be treated as extensions of the computer language itself. Logo is a prime example of a language that does this.

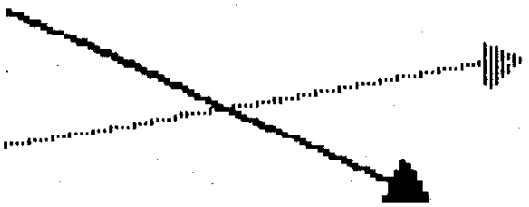
There are many advantages of process-based descriptions. In conventional coordinate geometry, for example, the static description of a square located on a grid consists of specifying the coordinates of the square's corners.



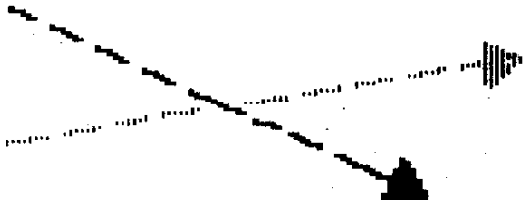
To create a new square at another location, one must create a new set of coordinates for each corner. In turtle geometry, once one has defined a procedure which creates a square, additional squares can be created by moving the turtle to a new location and using the "square" procedure at that point.



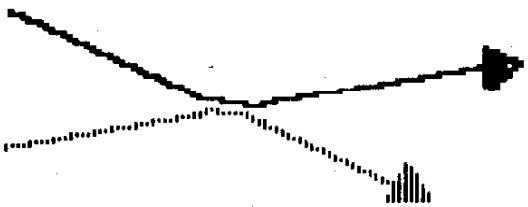
4. *No two events can occupy the same space at the same time.* Two energy events that are in close temporal and physical proximity will interact with each other in one of several ways, including:



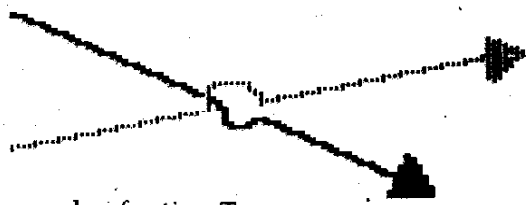
a. *tangential avoidance.* One event can cross over or under another event.



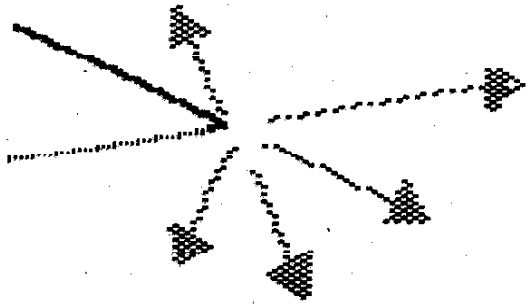
b. *modulated noninterference.* If the energy events consist of a train of pulses and spaces, their paths can cross in a fashion similar to that displayed by two rows of cars which are changing lanes on a freeway.



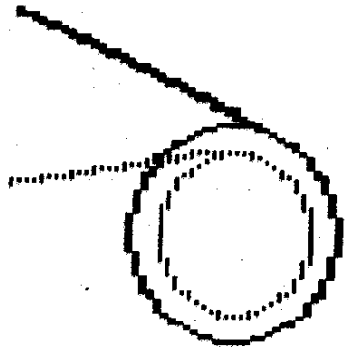
c. *reflection.* Two events can reflect from each other and acquire new paths.



d. *refraction*. Two events can, on achieving close proximity, perturb each other's path to avoid a collision.



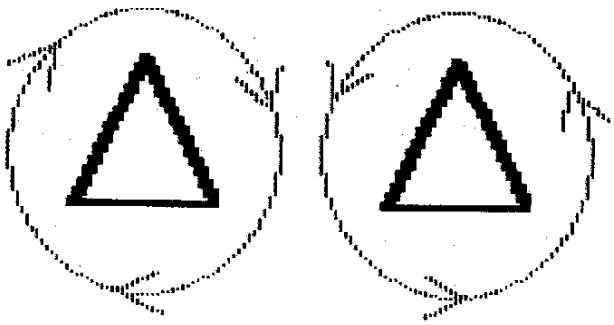
e. *collision*. When two energy events come into sufficiently close proximity, they may collide and smash into several other energy events which go off in a multitude of new paths.



f. *attainment of critical proximity*. When two energy events become sufficiently close, they may go into orbit around each other. As a result of this coupling, they form a new system.

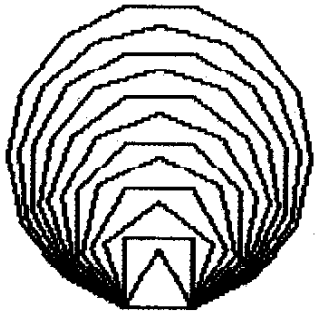
There are six ways in which two energy events can interact. There is *no* way two energy events can occupy the same place at the same time. The concept of a dimensionless point resulting from the intersection of two lines is thus meaningless in the physical universe.

In turtle geometry a secondary consequence of this concept is that different procedures can be used to create figures which appear to be identical. A triangle, for example, can be created by following a left- or right-handed path.



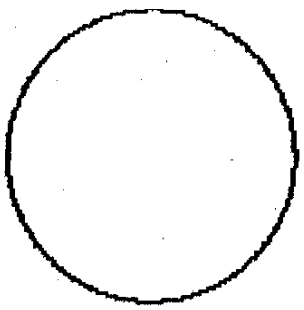
Even though the finished figures are identical (such paths are called *state change invariant*), the fact that they result from different procedures can have important consequences. For example, an assembly-line robot that moves parts between three work stations will only perform its job properly for one path description.

5. *Irrational numbers are unnecessary*. Synergetics involves a system of measurement based on discrete angles and countable frequency increments. Space-filling structures are formed from polyhedra, the minimum configuration of which is the tetrahedron. As the frequency of a structure is increased (by constructing polyhedra with greater numbers of nodes), one approaches the construction of objects that appear nearly round.

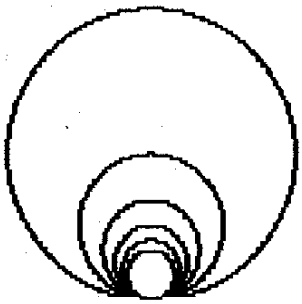


These objects are composed of a vast (but countable) number of discrete chords. Since such surfaces can be formed with any complexity desired, and since each surface is still bounded by chords, there is no need in Synergetics for irrational numbers such as pi.

This is easily demonstrated in turtle geometry. To send the turtle on a circular trip, one might instruct it to take 360 steps, turning by one degree after each step.



Circular paths with different sizes can be created by changing the size of the step or by changing the amount turned at the end of each step. Instead of dealing with the concepts of diameter and area, turtle geometry creates circles through the concepts of perimeter and curvature.



6. *Meaningful descriptions of processes are local.* Every celestial object is in motion with respect to every other object. These motions, viewed as a set, are nonsimultaneous.

Furthermore, the interactions of these various motions vary widely over the eons of time. As a result, any meaningful system of geometry must describe local processes without reference to an absolute origin. A description of a triangle must describe only the triangle itself and not be dependent on the reference frame in which the triangle is being envisioned.

The concept of local descriptions of geometrical figures is central to turtle geometry. In contrast to conventional coordinate geometry, turtle procedures provide *intrinsic* descriptions of objects. As mentioned before, a coordinate representation of a square applies to that one square only. The points on this one square are fixed in relationship to the origin of the coordinate system. In turtle geometry, on the other hand, a square is defined by the local steps that are needed to create it. A procedure such as:

```
TO SQUARE
   REPEAT 4 [FORWARD 25 RIGHT 90]
END
```

will always create a square path regardless of the turtle's location and orientation.

If, as Fuller believed, Synergetics provides the proper geometric framework with which to view the universe, then the incorporation of turtle geometry in various popular and user-friendly computer languages promises to help expand the awareness and creativity of all its users. The fact that many of the users of turtle geometry are children suggests that the child's view of the physical universe might have more power than we ever expected. ©

NOW, PLATO COURSEWARE MAKES ALGEBRA FRIENDLY.

Algebra can be a fascinating learning experience with the help of PLATO® educational courseware.

PLATO helps bring out the best in your child... and the best in your Apple II Plus or Apple IIe.
A new series of PLATO lessons can help your kids feel comfortable and confident about Algebra. Practice problems change at random to challenge them; examples of solutions guide them; performance reports encourage them. If a mistake is made, PLATO shows kids the correct answer or how to correct it. So before they know it, they're into Roots, Polynomials, Factoring, Equations and much more.

These Elementary Algebra lessons join the growing PLATO library of quality educational courseware for microcomputers. Other lessons include Math, Foreign Languages, Physics-Elementary Mechanics, Computer Concepts, Computer Literacy and Keyboarding.

See the PLATO line at selected retail outlets. All PLATO micro courseware is available for the Apple II Plus and Apple IIe. Some lessons are also available for the TI99/4A and Atari 800.
For a free PLATO catalog:
Call toll-free: 800-233-3784.
(In Calif., call 800-233-3785.)
Or write Control Data Publishing Co., P.O. Box 261127, San Diego, CA, 92126.

Warranty available free from Control Data Publishing Co., 4455 Eastgate Mall, San Diego, CA 92121.

PLATO
COMPUTER-BASED EDUCATION

CONTROL DATA
PUBLISHING

Phone Directory And Dialer For The TI

Ken McCann

This useful program will work as a phone number file as well as an automatic dialer. It will run on the TI with or without Extended BASIC.

Your computer, cassette recorder, and TV are all you need to run this program. Although it is written in standard TI BASIC, it will run faster if you have Extended BASIC.

DATA statements are included in the program, so only one load is required. Also, with the data files at the beginning of the program, data entry is simple and straightforward. The line numbers and the file numbers are the same, to make updating and deleting less complicated.

DATA Line Format

Enter the name and phone number information for your personal directory in the form shown below (also see line 1 in program for same example):

Line No.	File No.	Name	Phone Number
1 DATA	1,	MCCANN.K,	1,2,3,4,5,6,7
variables (in program)	A	B$	CDEFGHI

The last DATA entry must be followed by an END line. For example:

```
140 DATA  140,END
```

Auto Dialing

In most cases, the accuracy of the frequencies generated with the CALL SOUND statement is not close enough to use as "touch tones" to dial the phone. Therefore, I executed CALL SOUND statements with a frequency counter hooked up to the TV audio output, and added or subtracted until I got the proper frequency.

To use the Auto Dial feature, hold the phone up to the speaker of your TV and press C. Note that you must have a Touch-Tone type phone to use the Auto Dial feature. Two tones were used, and the frequencies for each digit of the phone dial are as follows:

1 697,1209	6 770,1447
2 697,1336	7 852,1209
3 697,1447	8 852,1336
4 770,1209	9 852,1447
5 770,1336	0 941,1336

Program Operation

Search Name and Dial: Type RUN, then press N to enter the Search mode. You are prompted to enter the Name *exactly* as it is in the files and then to press ENTER. The computer will display the DATA item called for. Press C, and the computer will PRINT the number and sound the dial tones for the number.

List: Type RUN, then press L to enter the List mode. You will see DATA as it is in the files. Press C to look at the entire list.

Letter Index List: Type RUN, then press I to enter the Index Mode. Then simply enter the first letter of the last name and press ENTER. The computer will display all the entries beginning with that letter.

Program Explanation

Lines 1-299 can be used for DATA statements. Remember to put the END statement last, after all files are listed.

Lines 300-480 set up the menu.

Lines 510-740 list all DATA items in the order they appear.

Lines 810-990 search DATA for a particular name.

Lines 1000-1820 dial the number and print the file number.

Lines 1830-2030 search for and print all names beginning with a given letter.

Phone Directory

```
1 DATA 1,MCCANN.K,2,1,2,4,4,4,4
2 DATA 2,CLAUSS.S,5,5,5,5,1,2,1,2
3 DATA 3,NIXON.R,3,3,3,4,5,4,5
4 DATA 4,END
300 CALL CLEAR
```

```
310 PRINT "PHONE DIRECTORY"
320 PRINT "(3 SPACES)& DIALER":::
330 PRINT "SELECT MODE DESIRED"
340 PRINT
350 PRINT
360 PRINT
370 PRINT "LIST ALL ENTRIES
    (6 SPACES)(L)"
380 PRINT
390 PRINT "SEARCH NAME & DIAL
    (4 SPACES)(N)"
400 PRINT
410 PRINT "LETTER INDEX LIST
    (5 SPACES)(I)"
420 CALL KEY(0,KEY,STATUS)
430 CALL SOUND(50,4000,0)
440 IF STATUS=0 THEN 420
450 IF KEY=76 THEN 500
460 IF KEY=78 THEN 780
470 IF KEY=73 THEN 1920
480 GOTO 1260
490 REM
500 REM
510 CALL CLEAR
520 CALL SCREEN(16)
530 PRINT "PHONE DIRECTORY LIST"
540 PRINT
550 FOR Z=1 TO 200
560 PRINT
570 READ A
580 READ B$
590 IF B$="END" THEN 1770
600 READ C,D,E,F,G,H,I
610 PRINT "FILE NUMBER  >";A
620 PRINT
630 PRINT "NAME  >";B$
640 PRINT
650 PRINT "NUMBER>";C;D;E;"-";F;G;H
    ;I
660 PRINT
670 PRINT "PRESS <C> TO PROCEED WIT
    H"

680 PRINT "LIST"
690 REM
700 CALL SOUND(100,1000,2)
710 CALL SOUND(75,675,2)
720 CALL KEY(0,KEY,STATUS)
730 IF STATUS=0 THEN 720
740 IF KEY=67 THEN 760
750 PRINT
760 NEXT Z
770 REM
780 REM(3 SPACES)
790 CALL CLEAR
800 CALL SCREEN(12)
810 REM
820 PRINT "NAME SEARCH"
830 PRINT
840 PRINT "ENTER NAME TO SEARCH FOR
    "
850 INPUT I$
860 PRINT
870 PRINT
880 PRINT
890 FOR S=1 TO 2000
900 REM
910 IF B$="END" THEN 1780
920 READ B$
930 IF B$<>I$ THEN 1530
940 PRINT "INDEX LETTER >";SEG$(B$,
    1,1)
950 PRINT
960 PRINT "NAME >";B$
970 PRINT
980 PRINT "READY TO DIAL"
990 PRINT
1000 PRINT "PRESS >C< TO DIAL NUMBE
     R"
1010 PRINT
1020 CALL SOUND(100,1000,2)
1030 CALL SOUND(75,675,2)
1040 CALL KEY(0,KEY,STATUS)
1050 CALL SOUND(50,2000,6)
1060 IF STATUS=0 THEN 1040
1070 IF KEY=67 THEN 1090
1080 REM
1090 REM
1100 READ C
1110 PRINT C;
1120 N=C

1130 GOSUB 1430
1140 READ D
1150 PRINT D;
1160 N=D
1170 GOSUB 1430
1180 READ E
1190 PRINT E;
1200 N=E
1210 GOSUB 1430
1220 READ F
1230 PRINT F;
1240 N=F
1250 GOSUB 1430
1260 READ G
1270 PRINT G;
1280 N=G
1290 GOSUB 1430
1300 READ H
1310 PRINT H;
1320 N=H
1330 GOSUB 1430
1340 READ I
1350 PRINT I;
1360 N=I
1370 GOSUB 1430
1380 READ A
1390 A=A-1
1400 PRINT "(5 SPACES)"::
1410 PRINT "FILE NUMBER >";A
1420 GOTO 1740
1430 IF N=1 THEN 1540
1440 IF N=2 THEN 1560
1450 IF N=3 THEN 1580
1460 IF N=4 THEN 1600
1470 IF N=5 THEN 1620
1480 IF N=6 THEN 1640
1490 IF N=7 THEN 1660
1500 IF N=8 THEN 1680
1510 IF N=9 THEN 1700
1520 IF N=0 THEN 1720
1530 NEXT S
1540 CALL SOUND(100,1209,0,697,0)
1550 RETURN
1560 CALL SOUND(100,1336,0,697,0)
1570 RETURN
1580 CALL SOUND(100,1447,0,697,0)
1590 RETURN
1600 CALL SOUND(100,1209,0,770,0)
```

PROGRAMMING THE TI

C. Regena

Answers To Common Questions

I have appreciated your comments and feedback. Your letters help me in several ways to write a better column. I thought this month I would try to answer some general questions that I frequently see. Most of the questions concern peripherals or debugging, so I'll discuss these two main topics this month.

Do You Really Need Peripherals?

Peripherals are anything extra that you add on to your computer. To use your TI-99/4A, all you really need is the computer itself, a television or monitor so you can see what you're doing, the cord to connect the television and the computer, and the power cord (these cords are included with the computer). If you are writing your own programs, purchasing programs on cassette, or typing programs from magazines, you will need a cassette recorder and a cassette cable. You can use just about any kind of cassette recorder, but the TI Program Recorder is more reliable. The TI 99/4A console seems quite sensitive to the setting of the volume control. Your recorder does need to have a volume control and a tone control. Your *User's Reference Guide* tells how to use the cassette recorder.

To save a program you've written or typed in, use the command SAVE CS1 then press ENTER and follow the cassette instructions. To load a purchased program or a previously saved program, use OLD CS1 and follow the instructions. After you have pressed STOP on the cassette and ENTER on the keyboard, wait for the cursor to return (it may take a few seconds on longer programs), then type RUN to start the program.

By the way, as you are typing in a program, it's a good idea to SAVE your program every 20 minutes or so. It's a disaster to have a program all typed in after hours of effort, then have a sudden power failure that wipes out your program. I always use two cassettes and alternate them during the SAVE procedure just in case the power fails while I'm saving the program.

Most of my writing is for unexpanded computers with no peripherals other than the cassette. TI computers are very powerful machines just as they are, and I like to show readers how much they can do without investing any more money. The TI has many nice features and a very powerful built-in BASIC. The average household or educational user will not need any peripherals to enjoy and use the TI.

What Do You Buy First?

Many computer users soon want to do even more with their computers and begin to add peripherals. Many readers ask what peripherals to buy, and I can't really answer that because it depends on what *you* want to do with the computer. I added a printer first because I needed (wanted) listings of the programs I was writing. Other people can't live without a disk drive, so that's their first purchase.

There are many, many brands of printers available. To use a printer with the TI you need the RS-232 Interface; just make sure your printer is RS-232 compatible. My first printer was an old teletype. If you need to make a cable to connect the printer to the RS-232, the RS-232 manual has the pin connections and all the configuration information. My next printer included the cable – so work with your dealer to make sure you have everything you need.

It is still possible to use just the RS-232 without the peripheral box (known as the "old-style" peripheral system). If the only peripheral you will need is the RS-232, it is unnecessary to buy the Peripheral Expansion Box plus the RS-232 Interface Card. In fact, if you have the Peripheral Box, you can use the old-style RS-232 or the RS-232

card. If you have the old-style RS-232, just plug the Peripheral Box flex cable into the side of the RS-232.

Using the Peripheral Expansion Box is the present method of adding on peripherals to the TI. Inside the box are slots for various cards. There is also room for one disk drive to be inside the box. You may add cards as they become available (or as you can afford them or need them).

A disk drive can be used in many ways. To add a disk drive you also need the Disk Controller box or the Disk Controller card and the Peripheral Expansion Box. The main advantage of a disk system over a cassette system is speed. You may SAVE programs on disk just as on cassette. A full-memory program may take about 3 minutes to load with cassette but only about 20 seconds with the disk system. The disk system is also much faster on any file processing, and thus practically a necessity for business programs. Many business programs require two disk drives. One disk controller can control up to three disk drives. Disk systems are possibly undesirable for some home use or for use in elementary schools because the cassette system is easier for children to use, less expensive, and not as fragile.

The 32K Memory Expansion is available either as a separate box or as a card to go in the Peripheral Expansion Box. One irate reader wrote that in my January column I did not mention that to add the Memory Expansion you also have to buy the Peripheral Expansion Box. The answer is that the Memory Expansion is still available separately in a box that attaches to the side of the computer. The Peripheral Expansion Box is the best way to go if you are adding several peripherals, but if you need only one unit the "old-style" still works. The 32K Memory Expansion does require a command module that can access it. You cannot use the Memory Expansion with the built-in BASIC. TI Extended BASIC and Logo are examples of two of the modules that can use the Memory Expansion.

Computer Enhancements

A modem allows telecommunication – you can connect your computer through telephone lines to another computer such as a large data base or a mainframe "host" computer. Your home computer thus acts as a terminal. To use a modem you'll need the RS-232 Interface and the Terminal Emulator command module. The RS-232 has two ports so you can interface with both a printer and a modem. There are two terminal emulator command modules, and either one will work. They contain the software necessary to set up the communications. Terminal Emulator II also contains speech capabilities, so it is a dual-purpose module. There are many brands of modems and telephone couplers; you just need to make sure the one you use is RS-232 compatible.

The Speech Synthesizer is the peripheral that makes the computer talk. With the "free speech" offer (buy six command modules and get the Speech Synthesizer free), every home with young children should get one. The speech feature adds an extra touch to educational programs. To enable the computer to talk, you need a command module that has speech capabilities.

To program your own speech, you'll need Terminal Emulator II. Words are pronounced phonetically, or you can use numbered allophones, so programming speech takes some experimentation. You can also use the TI Extended BASIC module, but this module has limited speech – only a certain vocabulary (and variations of those words) can be used.

Wired Remote Controllers are available for games or for educational programs. The TI version comes as a pair of joysticks for two-player games.

With a Hex-Bus adapter you can save a program with the Texas Instruments Compact Computer 40 (CC-40) onto wafertape, then load it onto the TI-99/4A.

Alternatives To BASIC

Several languages are available for the TI-99/4A. TI Extended BASIC is probably the first one I would get for someone who likes to program. Extended BASIC comes as a command module, and no extra peripherals are required. Extended BASIC allows multistatement lines, actual subprograms, and complex IF-THEN-ELSE logic. If you like to convert programs from other versions of BASIC, Extended BASIC makes it a little easier. Another feature of Extended BASIC is the DISPLAY AT command to print at a specific location on the screen – and the PRINT USING command allows formatting, which makes it easier to print reports or line up numbers in a column of numbers.

Another main feature of Extended BASIC is sprite capability. You may custom design your own objects just like in TI BASIC, but then you can place the sprite on the screen, designate a color, and put the sprite in motion (all in one statement). For people who like to design games, Extended BASIC is a must. The sprites are a lot of fun to work with.

Logo and Logo II are command modules which are popular in introducing children to programming. The TI version of Logo allows all the common turtle commands, and you can define your own characters and choose colors. The Logo II version has music capabilities. To use Logo or Logo II, you need the memory expansion. If you are a teacher using Logo, be sure to get the *Logo Curriculum Guide*. It is a manual of excellent ideas

for using Logo in the classroom. It also includes sample programs.

For machine language, you can get the Editor/Assembler cartridge. The 32K Memory Expansion, Disk Drive, and Disk Controller are required. A less expensive way to try machine language is to use the Mini-Memory module (no peripherals required).

Another language available to TI users is Pascal. Peripherals required are the P-Code Card, 32K Memory Expansion, Disk Drive, Disk Controller, and Peripheral Expansion Box.

I'm sorry I cannot answer your questions about machine language or Pascal. My programming so far has been in BASIC (for several computers) and TI Extended BASIC.

Why Won't The Program Run?

Now to the second main topic – debugging. *Debugging* is a computer term which means finding what's wrong with a program that doesn't work correctly. This month I'd like to give you some tips on how you might pinpoint errors in a program you've typed in but won't run correctly.

Syntax errors are the easiest to find and correct. If you RUN the program, it will stop at any syntax error and tell you exactly what's wrong and in which line. *Syntax* usually refers to a typing error such as a word spelled incorrectly, a comma in the wrong place, unmatched parentheses or quotes, or the wrong number of parameters in a CALL command. The TI catches a lot of typing errors as you are typing in the lines. Others are detected as the program is RUN. Remember that you can type the line number then the down arrow (with FCTN on the TI-99/4A and SHIFT on the TI-99/4) to edit a particular line, then use the arrow keys to move the cursor to the error.

Check line numbers in program transfer statements – GOTO, GOSUB, ON-GOTO, ON-GOSUB, and IF-THEN-ELSE statements. One digit can make a difference in the proper program control. For example, my coordinate geometry program in the February 1983 issue had a typesetting error. Line 760 should have been GOSUB 1860 instead of GOSUB 1850. That one digit caused an error. Several people wrote in very complex solutions to a problem I didn't know existed until I compared line numbers and noticed that one digit. This was one case where there really was a printing error. Now COMPUTE! has the listings printed directly from the computer to avoid such errors.

Check Your DATA Statements

Check to make sure DATA statements are typed correctly. If your program has DATA statements and doesn't run properly, the most likely place for a typing error is in a DATA statement. You may want to review the description of DATA statements in your *User's Reference Manual* or my August 1983 column on DATA and READ statements so you can follow the logic of the READ statements and corresponding DATA statements.

If you get a DATA ERROR, you may not have enough data items to fulfill the READ requirements. The line number given in the error message is the READ statement, so you'll have to find the corresponding DATA statement. Check the DATA statements for the proper placement of commas. It is possible there are commas together with nothing between them – this indicates a null string or "", and every comma is necessary. Also, make sure you do *not* have a comma at the end of a DATA statement.

Another type of DATA error is that the computer is trying to read a numeric value but gets a string (letters). Again, check the commas. Also make sure you haven't mistyped the number zero and the letter O.

If you have a lot of DATA statements, your eyes may get tired trying to compare printed statements with your typed statements. To try to pinpoint the trouble spot, LIST the lines around the READ statement referred to in the DATA error message. Remember you can list specific lines, such as LIST 640-660. Now PRINT the variables you are reading to find the last good values that were accepted. If you are reading within a FOR-NEXT loop, you can PRINT the index counter to see how far along the loop you are.

Anytime the program stops (BREAK), in this case with an error message, and as long as you don't do any editing, you may PRINT the value of any variable. For example, you may type PRINT B and press ENTER, and the present value of B will be printed. You can then look in the DATA statements to see where that particular value is. The value printed will be the last acceptable value for B, so the next couple of items may contain the error.

An error in a DATA statement may actually cause a problem in a statement other than a READ statement. For example, suppose you have this section of a listing:

```
650 FOR I=1 TO N
660 READ X,Y,G
670 CALL HCHAR(X,Y,G)
680 NEXT I
```

You could get the error message BAD VALUE IN 670. This means X, Y, or G is not acceptable. X must be a number from 1 to 24 for the row number. Y must be from 1 to 32 for the column number, and G must be an ASCII code number. You can PRINT X;Y;G to see what the values for X, Y, and G are. The next step is to see how you got the bad value. In this case, line 660 READs the values from DATA, so you can search through the DATA

statements to find a sequence of the three numbers the computer printed. The error will probably be a typing error just before those numbers.

Other Common Errors

There are also errors unrelated to DATA statements. FOR-NEXT errors are usually not difficult to find. Every FOR statement must have a corresponding NEXT statement. Once in a while, however, you can search and search and everything seems matched up correctly. The most likely cause for the error is that a line just before a FOR statement or just before a NEXT statement has 28 characters (or a multiple of 28), so the cursor goes to the next line. You need to press ENTER, but the cursor makes you think you have already pressed ENTER, and you may go ahead and type the next line. The result is a run-together line. If you list that line among several others, they all look right because the numbers line up properly.

To see if this is the problem, LIST only the line containing the FOR or the NEXT to see if it's really there. Warning: The FOR-NEXT error message may list a line number that is really OK; the run-together FOR or NEXT statement may occur before the one listed in the error message. If you use the automatic numbering feature as you type in programs, this problem is less likely to occur.

The run-together line problem may occur anytime you are typing lines that have 28 characters and could cause other problems.

A "glitch" type problem may occur in ON-GOTO and ON-GOSUB statements. The line should be typed in the following example form:

```
200 ON A GOTO 340,550,760,800
```

where there are no spaces between the line numbers. If you happen to type a space in between line numbers then later LIST the line, the space will not be there but it could still be causing an error. If you suspect you are having trouble with an ON-GOTO or ON-GOSUB statement, retype the whole statement. By the way, don't try to second-guess the author. The line numbers do not have to be in numerical order, and you can use the same line number in several of the positions.

These are answers to the most common questions I've been asked. If you still have problems getting a program to run, you may write to me. Be sure to tell me which program you are typing, which computer you are using, the exact error message with the line number, and what happens plus whatever other conditions may contribute to the problem. I want you to be able to use and enjoy these programs; however, it is difficult for me to help you debug if I don't know the exact conditions and line numbers. Please do not ask me how to solve problems with programs written by other authors.

Also, please do not ask me to debug one of your programs or to write a program (or convert a program) for you. It isn't your project if I do it for you, and the joy of programming is accomplishing your own goals. I also am not set up to review programs for you. You may submit them to COMPUTE! directly. I *do* welcome comments or suggestions for future columns that will interest the general TI user.

Since I haven't written about a specific programming technique this month, here's a short graphics display program to try this time. I'll try to have a Christmas present for you in my December column.

```
100 DEF R=INT(16*RND+1)
110 RANDOMIZE
120 FOR I=1 TO 16
130 CALL SOUND(-50,R*110,4)
140 CALL COLOR(I,R,R)
150 CALL SCREEN(R)
160 CALL HCHAR(R+4,R*2,R*R/2,R*R)
170 CALL VCHAR(R+4,R*2,R*R/2,R*R)
180 NEXT I
190 GOTO 110
200 END
```

Line 100 defines a function R to be a random integer from 1 to 16. Every time R is used in later lines, R will be a random integer from 1 to 16 – a lot less typing by using the DEFinition function.

All About The TI Character Set

Michael A. Covington

This brief outline of the TI character set explains how the computer recognizes each character. The author discusses some uses of the characters' numeric codes and indicates which characters' graphic representations can be assigned or changed.

Chances are you've never given your computer's character set much thought. You press keys on the keyboard and the characters appear on the screen; that's all there is to it, or so it seems. But there's a lot more going on than meets the eye.

Inside the computer, each character is represented by a *numeric code* – a number between 0 and 255 inclusive. For instance, the code for capital E is 69; the code for an exclamation mark is 33; the code for a blank (a blank is a character just like all the others) is 32. To associate these codes with the characters you see on the screen, the computer has to know two more things about each of them: a *graphic representation* that describes how the character is supposed to look on the screen, and a *key assignment* that indicates what key or combination of keys you can hit on the keyboard to type the character. For instance, the character string "HELLO THERE!" (not counting the quotation marks) involves the following:

Graphic representation:	H	E	L	L	O	!	T	H	E	R	E	!
Numeric code:	72	69	76	76	79	33	84	72	69	82	69	33
Key assignment:	H key	E key	L key	L key	O key	shift & 1 keys	T key	H key	E key	R key	E key	shift & 1 keys

Statements Using Numeric Codes

Normally (when you type characters in response to a string INPUT statement or when you type them as part of a program) you enter characters by hitting the keys that correspond to them. That is, you access them by means of their key assignments, and within the program you treat them as character-string data. But there are ways of referring to characters by their numeric codes and treating them as numbers. For instance, the CALL HCHAR and CALL VCHAR statements, which you meet at an early stage as you work through the manuals that come with the computer, refer to characters by their numbers. The statement

```
CALL HCHAR(3,3,69,20)
```

will place a row of 20 capital E's (character number 69) on the screen beginning at row 3, column 3.

Also, you can input characters as numeric codes. The CALL KEY statement senses whether a particular key on the keyboard is up or down, when a key is pressed, CALL KEY gives you the numeric code corresponding to it. For instance, here is a program which will tell you the numeric code of any key on the keyboard:

```
10 PRINT "PRESS ANY KEY..."
20 CALL KEY(5,CODE,STATUS)
30 IF STATUS <> 1 THEN 20
40 PRINT CODE
50 GO TO 10
```

The heart of the program is lines 20 and 30. Line 20 tells the CALL KEY subroutine to look at the keyboard and report what's going on. The variable STATUS will equal 1 only if the condition of the keyboard has changed since the last time the routine looked at it. If STATUS does not equal 1, we simply go back to line 20, since we don't want to do anything more if the user hasn't pressed a key or hasn't yet let go of the one already looked at. The variable CODE contains the numeric code associated with the key being pressed, if any. (The first parameter of CALL KEY, the number 5, simply indicates that we want the

usual BASIC set of codes; specifying other numbers there instructs the computer to use other sets of key assignments for various special purposes.)

The ASC and CHR$ functions allow you to convert back and forth between numeric codes and character strings. If A$ is a character string, ASC(A$) is the numeric code of its first character; thus ASC("E") is 69. Conversely, if N is a number, CHR$(N) is a one-character string of which N is the numeric code; thus CHR$(69) is E. If we want the program above to print the characters themselves rather than their codes, we can convert the codes into characters by changing line 40 to:

```
40 PRINT CHR$(CODE)
```

The CALL CHAR subroutine allows you to alter graphic representations using a hexadecimal code that the manual describes in detail. For instance, if you want to change the dollar sign ($) into a British pound sign (£), just execute the statement:

```
CALL CHAR(36,"001C22207C20207E")
```

That will do it, at least as long as the program is running: the key assignment and numeric code will be the same, but the dollar sign will look like a pound sign. (It will revert to its original appearance when your program stops executing.)

What's Not In The Manual

Those are the preliminaries; now we get to the really interesting part (the part that isn't in the manual, at least not entirely). Internally, the computer can use any number from 0 to 255 as a character code; any such code can be an element in a character string and can be referred to by CALL VCHAR, CALL HCHAR, and CHR$. (In fact, CALL VCHAR, CALL HCHAR, and CHR$ will actually take numbers up to 32767; multiples of 256 are subtracted as necessary to get a number in the 0-to-255 range.) But not all the codes have key assignments or graphic representations. The breakdown (by numeric codes) is as follows:

0 – Undefined (no key assignment, no graphic representation).

1 to 15 – Function keys (Table 1). Most of these characters can be input by means of the CALL KEY statement, but they cannot be typed in normal contexts (for example, in response to an INPUT) because there they are interpreted as requests to perform cursor movements or the like. They have no graphic representations (if you print them, you get blanks or garbled patches).

16 to 29 – Undefined (like 0, these codes have no key assignments and no graphic representations, and there is no straightforward way of giving them either).

30 – The graphic representation of this character is the black square that marks the cursor; thus, CHR$(30) is handy if you want a black square. No key is assigned to it.

31 – This is the screen border character – a blank that is the color of the border rather than the typing area. No key is assigned to it.

32 to 126 – Standard ASCII characters (Table 2). These are the characters you use every day, including the alphabet, the numbers, and all the punctuation marks and mathematical symbols. Their graphic representations can be changed with CALL CHAR but will revert to their original form when the program ends.

127 to 159 – User-defined characters (Table 3). These start out with no graphic representations, but you can define them with CALL CHAR, and, contrary to what the TI manual says, such definitions remain in effect after the program stops running (though most are disrupted when another program is loaded).

What most people don't realize is that these characters can be typed – they have key assignments and are acceptable in the same context as any other character (that is, in response to an INPUT or CALL KEY, or within quotes in a program). All but one of them require you to hold down the CTRL key (at the lower-left corner of the keyboard) when typing them; character number 127 uses the FCTN key instead.

160 to 175 – Undefined.

176 to 198 – These characters have key assignments (Table 4), but no graphic representations and no direct way of giving them any. They can be used as special function keys of some sort (in response to either CALL KEY or INPUT), but not as displayable characters.

199 to 255 – Undefined.

Even the "undefined" character codes (those that cannot be typed on the keyboard or displayed on the screen) are not completely useless. You can refer to them by means of CHR$ and ASC and use them as special markers of various kinds when manipulating character strings. They also may come into play when you are transmitting data to other devices (for example, printers or other computers) that have definitions for characters that are undefined on the TI-99.

Finally, consider this possibility. Each character in a character string has a code between 0 and 255 inclusive, accessible through CHR$ and ASC. Also, the SEG$ function allows you to address individual characters in a string, and the & (concatenation) operator allows you to construct strings out of individual characters. This means that a character string gives you a compact way of storing a set of integers between 0 and 255 – each element occupies only one byte in memory, as compared to the eight bytes normally needed to store a number. So if you have a program that

needs to keep track of thousands of small integers – more than will fit in available memory in numeric form – then character strings may be the answer.

Table 1:
Function Key Codes

(None of these characters have graphic representations, nor can they be given them. They can be typed only through the CALL KEY statement, not in response to a string INPUT statement, or within a program.)

Code	Key
1	FCTN 7 ("AID")
2	None usable. The key definition associated with this code is FCTN 4, but in BASIC, hitting that key interrupts the program.
3	FCTN 1 ("DELETE")
4	FCTN 2 ("INSERT")
5	None usable. The key definition associated with this code is FCTN =, but hitting that key forces a machine reset and the program in memory is lost.
6	FCTN 8 ("REDO")
7	FCTN 3 ("ERASE")
8	FCTN S (left arrow)
9	FCTN D (right arrow)
10	FCTN X (down arrow)
11	FCTN E (up arrow)
12	FCTN 6 ("PROC'D")
13	ENTER
14	FCTN 5 ("BEGIN")
15	FCTN 9 ("BACK")

Table 3:
User-definable Graphics Characters

These characters can be typed using the key combinations listed and are acceptable in any context (that is, they can be input using the CALL KEY or INPUT statements and can appear between quotes within a BASIC program).

Graphic representations can be given to these characters with the CALL CHAR statement. Contrary to TI documentation, such representations, once assigned, will persist after the program stops running.

Code	Key	Code	Key
127	FCTN V	144	CTRL P
128	CTRL , (comma)	145	CTRL Q
129	CTRL A	146	CTRL R
130	CTRL B	147	CTRL S
131	CTRL C	148	CTRL T
132	CTRL D	149	CTRL U
133	CTRL E	150	CTRL V
134	CTRL F	151	CTRL W
135	CTRL G	152	CTRL X
136	CTRL H	153	CTRL Y
137	CTRL I	154	CTRL Z
138	CTRL J	155	CTRL . (period)
139	CTRL K	156	CTRL ;
140	CTRL L	157	CTRL =
141	CTRL M	158	CTRL 8
142	CTRL N	159	CTRL 9
143	CTRL O		

Table 2:
ASCII Graphic Characters On The TI-99/4A

(This table gives the numeric codes and graphic representations; the key assignments are marked on the keyboard. The graphic representations can be changed by the CALL CHAR statements but revert to their original form when the program stops running.)

Code	Graphic Representation	Code	Graphic Representation
32	(space)	53	5
33	!	54	6
34	"	55	7
35	#	56	8
36	$	57	9
37	%	58	:
38	&	59	;
39	'	60	<
40	(	61	=
41	)	62	>
42	*	63	?
43	+	64	@
44	,	65	A
45	- (minus)	66	B
46	.	67	C
47	/	68	D
48	0	69	E
49	1	70	F
50	2	71	G
51	3	72	H
52	4	73	I
74	J	97	a
75	K	98	b
76	L	99	c
77	M	100	d
78	N	101	e
79	O	102	f
80	P	103	g
81	Q	104	h
82	R	105	i
83	S	106	j
84	T	107	k
85	U	108	l
86	V	109	m
87	W	110	n
88	X	111	o
89	Y	112	p
90	Z	113	q
91	[	114	r
92	\ (back slash)	115	s
93	]	116	t
94	^	117	u
95	_ (underline)	118	v
96	`	119	w
120	x		
121	y		
122	z		
123	{		
124	|		
125	}		
126	~		
127	See Table 3. TI documentation mistakenly classifies this character with the wrong group.		

Table 4:
Characters With Key Assignments But No Graphic Representations

These characters are not mentioned in TI documentation. They can be typed in any context (that is, in response to an INPUT or CALL KEY statement or between quotes in a program), but they have no graphic representations and cannot be given any.

Code	Key	Code	Key
176	CTRL 0	188	FCTN 0 (zero)
177	CTRL 1	189	FCTN ;
178	CTRL 2	190	FCTN B
179	CTRL 3	191	FCTN H
180	CTRL 4	192	FCTN J
181	CTRL 5	193	FCTN K
182	CTRL 6	194	FCTN L
183	CTRL 7	195	FCTN M
184	FCTN , (comma)	196	FCTN N
185	FCTN . (period)	197	FCTN Q
186	FCTN /	198	FCTN Y
187	CTRL /		

```
260 DSN$="":FOR Z=K3 TO 13:IF Z=11 T
    HEN DSN$(LEN(DSN$)+K1)="."
265{3 SPACES}IF Y$(Z,Z)=" " THEN 275
270{3 SPACES}DSN$(LEN(DSN$)+K1)=Y$(Z
    ,Z)
275{3 SPACES}NEXT Z:IF DSN$="DOS.SYS
    " THEN 255
280 IF LEN(DSN$)<K12 THEN DSN$(LEN(D
    SN$)+K1,K12)="{11 SPACES}"
285 SEC=K0:TRAP 290:SEC=VAL(Y$(15,17
    ))
290 TRAP TOFF:PRINT IP+K1;:POKE COL,
    5:PRINT DSN$;:POKE COL,22:PRINT
    SEC;:POKE COL,28:PRINT "?";:GOSU
    B 340
295 IF Z=67 THEN PRINT CHR$(30);"C":
    GOTO 310
300 PRINT CHR$(ERASE);:IF Z=81 THEN
    320
305 GOTO 255
310 X$(LEN(X$)+K1)=DSN$:IP=IP+K1:IF
    IP=DSN THEN 320
315 GOTO 255
320 CLOSE #K3:RETURN
325 PRINT "Insert 'from' disk";
330 PRINT ", press any key!";
335 GOSUB 340:PRINT CHR$(ERASE);:RET
    URN
340 GET #K4,Z:IF Z=27 THEN POP :GOTO
     235
345 RETURN
350 GOSUB 340:PRINT :RETURN
355 PRINT :PRINT "Any more files?";:
    GOTO 335
360 Z=PEEK(764):POKE 764,HI-K1:IF Z=
    28 THEN 235
365 RETURN
370 PRINT "[";Z;"] I/O error on ";DS
    N$:PRINT "...skipping to next fi
    le!":PRINT :RETURN
375 REM Delete lines beginning here
    !
380 GRAPHICS K0:POSITION 13,12:PRINT
     " PLEASE WAIT! ";
385 K0=0:K1=1:K2=K1+K1:K3=K2+K1:K4=K
    3+K1:K12=K4*K3:HI=256:TOFF=40000
390 YES=89:LET ERASE=156:COL=YES-K4:
    OPEN #K4,K4,K0,"K:"
395 DSN=16:DIM DSN$(16),X$(DSN*K12),
    X(DSN-K1,K1)
400 CIO=960:FOR Y=K0 TO 42:READ Z:PO
    KE CIO+Y,Z:NEXT Y:POKE 709,PEEK(
    710)
405 DATA 104,104,104,10,10,10,10,170
410 DATA 104,104,157,66,3,104,157,69
415 DATA 3,104,157,68,3,104,157,73
420 DATA 3,104,157,72,3,32,86,228
425 DATA 189,72,3,133,212,189,73,3
430 DATA 133,213,96
435 MAX=842:APND=35
440 IP=375:OP=450:REM These are the
    'from' and 'to' delete line num
    bers!
445 GOTO 460:REM Remove this line t
    o allow deletes!
450 PRINT CHR$(125):PRINT :FOR Z=IP
    TO OP STEP 5:PRINT Z:NEXT Z:PRIN
    T "CONT":POSITION K0,K0:POKE MAX
    ,13
455 STOP
460 POKE MAX,K12:BUFF=FRE(K0)-APND:D
    IM Y$(BUFF):BUFF=BUFF-K3:GOTO 150
```

entry of word lists of up to 50 words. The program adjusts skill levels in relation to each student's progress. *Word Wizard* sells for $29.95.

Math Master is similar to *Word Wizard* in that it allows a parent or teacher to determine the problems to be solved and adjusts speed according to each student's success. The program is available for $29.95.

Advanced Computing Enterprises
5516 Rosehill Road
Shawnee Mission, KS 66216

TI-99/4A Peripherals And Software

Texas Instruments has changed its marketing strategy for TI-99/4A Home Computer peripherals. Prices for peripheral equipment and some software have been cut, and TI has begun selling its disk-storage related peripherals as a package.

The Disk Drive Memory System, which now has a suggested retail price of $550, consists of the peripheral expansion box, the disk controller card, one disk drive, and a 32K memory expansion card. Under TI's old pricing system, the suggested retail price for the above combination of equipment was $1,200.

Other reductions include: the P-Code Card, cut from $250 to $100; the RS-232 card, reduced from $175 to $100; the telephone coupler, from $200 to $100; the Impact Printer, from $750 to $500; and TI Logo II, from $129.95 to $99.95.

In other news from TI, the company has signed manufacturing and marketing agreements for software from Brøderbund Software, Spinnaker Software, and Sega Enterprises. Under the agreements, the software makers will supply programs to TI, and TI will translate them to ROM cartridge and market them.

As a result of the agreements, TI's software library soon will include Brøderbund's *Choplifter*, a daredevil rescue game, and *David's Midnight Magic*, a high-resolution pinball game simulation.

Spinnaker's first offerings to TI are *Facemaker*, an educational game for 4- to 12-year-olds, and *Story Machine*, a program that teaches children to write sentences, paragraphs, and simple stories, and then brings the stories to life through color graphics and sound.

Sega, a recognized leader in coin-operated amusement games, will provide TI with *Congo Bongo*, a cartoon-like adventure that takes players over and through jungle obstacles; *Star Trek*, in which the player controls the starship Enterprise against the Klingons; and *Buck Rogers: Planet of Zoom*, a space adventure filled with challenging tunnels and channels to negotiate and bizarre aliens to outsmart.

Most of the above software is expected to sell for $39.95 in cartridge format.

Texas Instruments, Inc.
P.O. Box 53
Lubbock, TX 79408
(800)858-4565

Baseball For Atari, Commodore 64

Star League Baseball, an action and strategy game from Gamestar, is available for Atari and Commodore 64.

The game features realistic animation including windup, delivery, and the arc and shadow of a fly ball. Sounds generated by the program include the crack of the bat, the cheers of the crowd, and the music of the seventh inning stretch.

Players choose their starting team and pitcher, can bring in relievers, and can take batting practice against the legendary "Heat" Muldoon.

Star League Baseball, which can be played against the computer or a human opponent, sells for $31.95.

Gamestar, Inc.
1302 State St.
Santa Barbara, CA 93101
(805) 963-3487

Investment Manager, Disk Manager, And Games

Bytes and Bits has released a handful of programs for the Commodore 64 and VIC-20 computers.

Investment Portfolio Manager is a program to track volatile assets such as stocks and stock options.

The program, for disk-based Commodore 64 systems, can handle entries of up to $99,999 and can report on nine investment categories. It is available for $14.95.

Disk Directory Manager, which can sort more than 1500 filenames into an organized list, is available for $19.95 for the VIC or 64. The program is written in machine language.

In *Dungeons* for the VIC-20 with 16K expansion, the player creates characters who explore a 12-level, 1200-room dungeon. The game is available on tape or disk for $19.95.

Pak Alien for the unexpanded VIC is a machine language game that includes 100 difficulty levels. Guide your alien through a maze of interplanetary space particles while dodging seven aliens. *Pak Alien*, which can be played with joystick or keyboard, sells for $14.95 on tape or disk.

Bytes and Bits
524 East Canterbury Lane
Phoenix, AZ 85022
(602) 942-1475

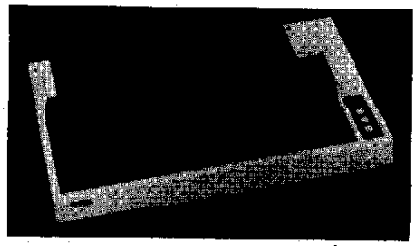
High Speed Printer

Epson America has introduced its new FX-100 dot matrix printer. The printer, which carries a suggested price of $895, provides a printing speed of 160 characters per second.

The printer offers a wide range of features including elite or pica spacing and a one-to-one graphics ratio so accurate graphics – including circles – can be drawn. Users also can create their own character formats on screen and download the font into the printer's memory.

The printer includes a rubber platen that can handle single sheet or roller-fed paper plus a removable tractor to handle pin-fed paper and forms.

Epson America, Inc.
3415 Kashiwa St.
Torrance, CA 90505
(213) 378-2220



The Epson FX-100 printer has a printing speed of 160 characters per second.

Holiday Adventures

Chartscan Data is producing *bitCards*, text and graphic adventures designed as holiday gifts. *bitCards* are customized with personal references to the recipient, and the player's reward for completing the adventure is a personal message chosen by the sender.

The first *bitCard*, *A Christmas Adventure*, is available on disk for 48K computers in the Apple II series, and on tape for Atari 400/800, Radio Shack Color Computer, Commodore 64, unexpanded VIC, and VIC with 8K expansion. All versions sell for $16.95.

bitCards
120 South University Drive, Suite F
Plantation, FL 33317
(305) 473-4741

Games Converted For Atari

Activision has converted two of its best-selling home videogames for Atari home computers.

The games, *River Raid* and *Kaboom!*, have been enhanced to make use of the increased power and memory availability of the Atari computers.

In *River Raid* the player pilots a B1 Strato-wing Assault Jet over a constantly changing river course. New twists added to the game include hovering hot-air reconnaissance balloons, helicopters and tanks that shoot back, vivid graphics and battle sounds, an expanded control panel, a pause feature, and the ability for more advanced players to select more difficult river sections.

Kaboom! features the Mad Bomber, a convict who roams the rooftops dropping bombs. The player maneuvers water buckets to catch the falling bombs. New additions to the game include a variation that allows one player to be the Mad Bomber and the other to catch the falling bombs, a high-score table, and a musical score which features the *1812 Overture*.

Both games retail for $34.95.

Activision, Inc.
2350 Bayshore Frontage Road
Mountain View, CA 94043
(415) 960-0410

Software Assortment For Timex/Sinclair

Dynacomp has introduced an assortment of software products for the 16K Timex/Sinclair computer.

The new programs are: *Blackjack*, $12.95; *Checkbook*, $14.95; *Data Filer*, $19.95; *Graphics Drawer*, $14.95; *Phone Book*, $12.95; *Pixel Drawer*, $14.95; *Tic-Tac-Toe*, $9.95; and *Word Scramble*, $9.95.

Each of the programs is described in the free product catalog available from Dynacomp.

Dynacomp, Inc.
1427 Monroe Ave.
Rochester, NY 14618

Game Cartridges For The TI

Funware, which recently was acquired by Creative Software, has added to its list of available cartridge software for the TI-99/4A.

Among the new games are:

St. Nick, in which the player helps Santa fly through a maze and pick up toys while avoiding flying witches.

Schnoz-ola, a game based on a tale about a Mayan prince with tremendous jumping powers and a tremendous nose. The prince must climb a four-tiered pyramid to collect sacred flowers, while dodging fireballs tossed at him by angry gods.

Ambulance, in which the player serves as an ambulance driver helping sick people get treatment. The player must decide who needs help first and whether the patient should be taken to the hospital or an emergency center.

Driving Demon, a game in which the player tries to drive as far and as fast as possible within a time limit. Problems include oil slicks, other cars, and engine troubles; navigational aids include a tachometer, speedometer, and a long-range scanner.

Funware's cartridge games sell for $44.95, and are guaranteed to work in all present and future versions of the TI-99/4A home computer.

Funware, Inc.
230 East Caribbean Drive
Sunnyvale, CA 94089
(408) 745-1655

Pinball Simulation For The 64

Brøderbund Software has converted *David's Midnight Magic*, a pinball simulation game, into Commodore 64 format.

The game, described as the next best thing to a real pinball machine, simulates dual flipper controls, bumper action, rollovers, sounds, and lights. The player can even put English on the ball, but too much will result in a tilt.

David's Midnight Magic, originally sold in Apple and Atari versions, is available for the Commodore 64 on disk for $34.95.

Brøderbund Software
500 Fifth Ave.
New York, NY 10110

Apple Graphics Program

T & G is an Apple graphics system from C & C Software. The program does not rely on shape tables, so execution is faster than for many other graphics programs.

T & G includes three character sets, the largest of which is 1½ times the size of standard Apple characters.

The characters can be placed on the screen using one of two commands. One is similar to BASIC's PRINT command; the other permits greater control over text placement. Lowercase is available in the two larger character sets, and the ability to print superscripts, subscripts, and text overlays is provided.

The program also is a powerful utility for producing static or animated graphics. The editor program supports full color and includes a command for clearing a precisely defined area of the screen.

T & G is available for $65.

C & C Software
5713 Kentford Circle
Wichita, KS 67220
(316) 683-6056

Hardware/Software Combo For Atari 800

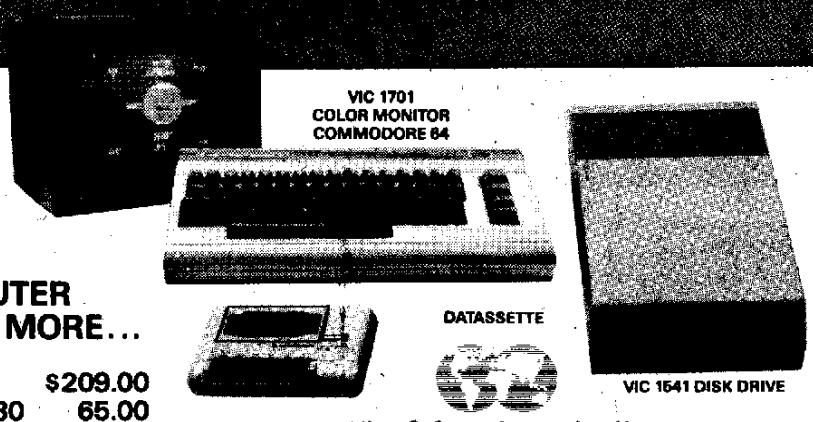
MEM/EX is a memory expansion device for the Atari 800 computer. It replaces the operating system board and includes 4K of low-power CMOS static RAM mapped to the unused $C000 address space.

Included with the board is a powerful command-extender program called *COM/EX*. This adds ten edit-mode commands to the standard BASIC or Assembler cartridges, including the most used disk commands.

MEM/EX is compatible with all standard Atari 800 configurations; COM/EX requires at least one disk drive and DOS 2.0. The package is available for $129.95.

Prairie Physics
P.O. Box 2369
Wichita, KS 67201

SAVE



PICK YOUR BEST COMMODORE COMPUTER PACKAGE AND SAVE EVEN MORE...

C 64	$209.00		C 64	$209.00
VIC 1541	239.00	or	VIC 1530	65.00
VIC 1701	249.00		VIC 1701	249.00
Total....	$680.00		Total....	$510.00
SAVE....	$ 37.00		SAVE....	$ 33.00

Everything you need to support your COMMODORE VIC20 and C64

Micro Software International Inc

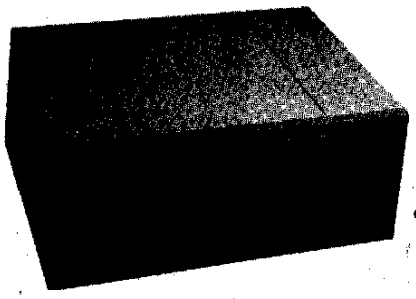
PractiCalc 20
For the Commodore VIC-20
16K RAM required

Tape (CV2T10)..... $28.00
Disk (CV2D10)..... 33.00

PractiCalc 64-For the Commodore 64

Tape (C60T11)..... $37.00
Disk (C60D11)..... 39.00

Cardco, Creative Software, EPYX, HES, UMI, WICO, Romox.



TEXAS INSTRUMENTS

SAVE

NEW **T.I. PERIPHERAL EXPANSION UNIT**

PHP 4000 includes:

• PHP 1200 Peripheral Expansion • PHP 1240 Disk Controller Card • PHP 1250 Expansion System Disk Drive • PHP 1260 Memory Expansion Card 32K.

AT THE NEW LOW PRICE OF ONLY $455.00
Original Retail $1,199.80

With the purchase of PHP 4000 package receive one of the following modules FREE!

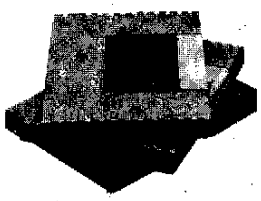
• **PHP 3111** T.I. Writer • **PHP 3113** Microsoft Multiplan • **PHM 3109** T.I. Logo II

EACH A $99.95 Retail Value!

GET A T.I. SPEECH SYNTHESIZER FREE

T.I. will give you a Speech Synthesizer a $99.95 retail value—FREE—when you buy six solid state software cartridges. Offer expires January 31, 1984. Ask for coupon.

PHP 1220 RS 232 Card	$ 83.95
PHP 1270 P-Code Card	83.95
PHP 1600 Modem	83.95
PHP 2500 T.I. Printer	349.95



We carry in stock all hardware and software for T.I. HOME COMPUTERS.

QUANTITIES LIMITED!

CAPUTE!

Modifications Or Corrections To Previous Articles

Atari Caves Of Ice

Due to a problem with variable initialization, the Atari version of this game from the September issue (p. 60) provides more exits the longer the game is played. To insure there is always only one exit from the Caves, reader Jim Vail suggests that the A$(750) = "O" in line 100 be changed to:

```
A$="O":A$(750)="O":A$(2)=A$:
```

Also, Ohio reader Neil Morris notes that the program can be modified to run in less than 16K of memory by changing the GRAPHICS 8 in line 305 to GRAPHICS 6, then dividing all the numbers after the PLOT and DRAWTO commands in lines 32-85 by two (dropping any fractions).

VIC Diamond Drop

Some Commodore 64 color controls slipped into the VIC version of this game from the September issue (p. 84). For proper operation, change the {6} in line 50 to {GRN} (CTRL-6 for green) and the {7} in lines 100 and 102 to {BLU} (CTRL-7 for blue).

Mystery Spell For TI And VIC

The TI version of this educational game from September (p. 117) stores word lists properly to disk. However, tape users need to make the following changes:

```
1810 DISPLAY AT(23,1):"PLACE TAPE O
     R DISK IN DEVICE" :: ACCEPT AT
     (11,3):F$
1815 IF SEG$(F$,1,1)="C" THEN 1842
     ELSE OPEN #1:F$,INTERNAL,UPDAT
     E,FIXED 50
1842 IF J=0 THEN 1846
1944 OPEN #1:F$,OUTPUT,INTERNAL,FIX
     ED 50 :: FOR I=1 TO 20 :: PRIN
     T #1:B$(I):: NEXT I :: CLOSE #
     1 :: GOTO 230
1846 OPEN #1:F$,INPUT ,INTERNAL,FIX
     ED 50 :: FOR I=1 TO 20 :: INPU
     T #1:B$(I):: NEXT I :: CLOSE #
     1 :: CALL CLEAR :: GOTO 1760
```

Also, there is a minor typo in the VIC version (p. 127). In line 2170, LEFRT$ should be LEFT$.

VIC PILOT

There is a minor flaw in the turtle graphics feature of the PILOT interpreter in BASIC from the September issue (p. 166). If the turtle goes off the graphics screen during a loop, subsequent graphics commands are not interpreted properly. If you are using Program 1, make this change:

```
128 IFOS%=1THENPRINT"*PLOT WENT OFF SCREE
    N":D=0
```

If you are using Program 2, the Super Expander version, make this change:

```
143 GOSUB127:IFOS%=1THEND=0:GOTO11
```

Cracking The Kernal

The instructions for using Kernal ROM routines to LOAD data from tape or disk into the 64 (September 1983, p. 270) are incorrect. The proper series of machine language instructions to OPEN logical file 2 from device 8 (disk) with secondary address 0 and LOAD a file into memory starting at address 8192 ($2000) is:

```
        LDA   #$02
        LDX   #$08
        LDY   #$00
        JSR   SETLFS
        LDA   #$04
        LDX   #L,NAME
        LDY   #H,NAME
        JSR   SETNAM
        LDA   #$00
        LDX   #$00
        LDY   #$20
        JSR   LOAD
        RTS
NAME    .BY   'FILE'
```

Note that the accumulator must be set to 0 for a LOAD and 1 for a VERIFY, not vice versa as stated in the article. Our thanks to Lewis Kleinsmith for pointing this out.

Atari Dragon

Program 2 (August 1983, p. 88) fails to leave the current score on the screen at the end of the game. To eliminate this minor flaw, Stuart Goldenberg suggests the following correction and addition:

```
410 COLOR 32:PLOT 0,0:DRAWTO 17,0:PL
    OT 0,1:DRAWTO 10,1:POSITION 0,0:
    ? #6;"GAME OVER Score ";W
435 POSITION 0,0:? #6;"{20 SPACES}"
```

TI Towers

The program for this article was renumbered before it was listed. As a result, the line numbers of the program do not agree with those mentioned in the article. However, the program will work as listed.

64 Ultrasort

When using Program 4 to test the 64 version of this machine language sorting routine (September 1983, p. 202), you should replace line 300 with:

```
300 SYS 49152,N,AA$(1)
```

The REM statement in line 291 is incorrect.

VIC Dots

In line 560 of this program from the September issue (p. 132), the spurious question mark should be removed. The first part of the line should read:

560 IFFNBX(ML+22)=...

Reduction of an actual sign

The Banner Machine™ —$49.95

- For the **Commodore 64,** tape or disk (5 fonts)
- For the **VIC-20** with 24K memory and disk
- For the **TRS-80** I & III with 32K tape or 48K disk
- Print on any Epson RX, FX, or MX with Graftrax; CBM 1525; Gemini 10X; or 8510 Prowriter
- Uses dot graphics instead of block graphics
- Menu-driven program operates like a word processor
- Makes signs up to 13" tall by any length (TRS-80: 10")
- Makes borders of variable width up to ¾ inch
- 8 sizes of letters from ¼" to 8½" high (TRS-80: 10 sizes)
- Proportional spacing
- Automatic centering; Right and left justifying; Tab
- Specify equipment and configuration when ordering

Cardinal Software
Virginia Micro Systems
13646 Jeff Davis Highway
Woodbridge, Virginia 22191
Phone (703) 491-6502

DISK DUPLICATOR FOR COMMODORE SINGLE DISK DRIVES

(1540, 1541 and 2031 used with a VIC-20, commodore 64, CBM 4000 or CBM 8000 computer)

DISK DUPLICATOR provides you a fast and easy way to make back-up copies of your precious, irreplaceable diskettes. Enjoy the convenience of a dual disk drive without the expense. DISK DUPLICATOR is 100% MACHINE LANGUAGE, 100% FAST, and most importantly, 100% AFFORDABLE!

Don't let an accident or mistake catch you without back-up copies of all your diskettes. ORDER "DISK DUPLICATOR" TODAY at the special introductory price of only $14.95 postage paid (check or money order only please).

H&H ENTERPRISES DEPT 123C
5056 NORTH 41st STREET
MILWAUKEE, WISCONSIN 53209

GAMBLERS T1-99/4A PROGRAMS

Practice "Texas Holdem". A poker game that's the rage of the gambling casinos and card rooms. Written in standard basic by a gambler for a gambler. You can bet, check, fold and analyze what hand is needed to win the pot. Why play against the house? "Holdem" is played at tables provided by the casinos. Practice at home then have the edge when you go to a casino or card room.

Also available "Keno". Practice the game with the big money payoff. Other casino games available soon.

Send $21.95 check or mo. + $2.00 shipping for each cassette and instructions. Washington residents add $1.60 sales tax.

GAMBLERS HELPER
HOUSE OF SOFTWARE
P.O. BOX 2797
TRI-CITIES, WA. 99302
"THE GAMBLERS FRIEND"

HOME BUDGET FOR COMMODORE 64

COMPLETE system for planning and tracking your home budget

RECORD tax deductable items, bank charges, deposits & expenses

RECALL data by category, month and amount

CALCULATION of all balances

POWERFUL tax tool

EASY to use documentation

MENU driven with **PRINTER** option

$29.95

SPECIFY TAPE OR DISK

RMC SYSTEMS CO., INC.
244 Scherr Hills
Berrien Springs, MI 49103
(616) 683-6829

Dealer inquiries invited.

Commodore 64 is a registered trademark of Commodore Business Machines.

DISKETTES

$1.35

100% CERTIFIED

TO ORDER CALL
1-800-322-DATA

DISCOUNT DATA SUPPLY
P.O. 02183
PORTLAND, OR 97202

LIMIT 100 DISKETTES PER ORDER

TI 99/4A

ENHANCE YOUR CHILD'S MUSIC EDUCATION

with fun skill-building games.
Sound, color graphics —
Progressing levels of play.

Levels 1-3
"Notes in a Nutshell"—5 finger patterns
"Lollipops and Lemons"—Note names
"Create a Creature"—Sightreading groups
"Ski with a Count"—Note values

Levels 2-4
"Artesian Escape"—8th or 16th note patterns
"On the Beating Path"—6/8 rhythm or 16th notes
"Extra-terrestrial Rescue"—Order of sharps or flats—(Joystick required)
"Pizza Go 'Round"—Major keys (Joystick required)
"Dine with Dragons"—Minor keys

Disk or Cassette —
EXTENDED BASIC REQUIRED
$19.95 plus $1.50 Shipping
Colorado residents add 4% tax

SOUNDWARE
P.O. Box 7026, Contract Stat. #6
Littleton, CO 80123

SAVE ON SOFTWARE

UP TO 30% OFF LIST FOR THE COMMODORE 64 ATARI 400/800/1200 VIC 20 · APPLE · TI99/4A

ALSO AVAILABLE AT LOW PRICES! GAME CARTRIDGES FOR COLECOVISION AND ATARI 2600 AND 5200 · MOVIES FOR VHS OR BETA UNITS.

Write for our FREE Catalog

ENCORE VIDEO software

354 Lancaster Ave., Suite 212
Haverford, PA. 19041

SOFTWARE COMMODORE 64

The Staff: Polyphonic Music Editor & generator. Enter up to 93 measures of 3 part harmony on easy to use graphic display. Disk: $22.95 for 64, $17.95 for VIC. Tape: $19.95 for 64, $14.95 for VIC. Add $1.50 for Postage and Handling.

We have a large selection of software for the 64—Word Processors, Data Base, Mailing List, Accounting Package, Spread Sheet, Educational Applications, Home and Personal Record Keeping, Programmers Aids, Games.

ASK FOR FREE CATALOG

Check or Visa, MasterCard accepted

PROFESSIONAL MICRO SERVICE
100 W. 22nd St., Baltimore, Md. 21218
301-366-0010
Dealer inquiries invited.

NEW **Centronics® 152-2**

Replacement Ribbon

Manufactured by Aspen Ribbons, Inc. Buy direct from manufacturer & save.

Standard ink color is black. Red, green, blue, brown, & purple colors are available for $2.00 extra per ribbon.

PRICES

Centronics® 152-2 $3.75 to $6.50 ea.
Price depends on quantity ordered.

CALL FOR FREE CATALOGUE.

*Aspen Ribbons, Inc. is not affiliated with any company mentioned in this ad.

Aspen Ribbons, Inc. 1700 N. 55th St.
Boulder, CO 80301-2796 (303)444-4054
Telex: 45-0055 End User: 800-525-0646
Wholesale: 800-525-9966

BEAT THE HIGH COST OF SOFTWARE

Before you buy any more software, you owe it to yourself to join SOFTRADERS™ the fastest growing worldwide trading network. Members swap software and info for all computers.

Annual membership includes:

- Quarterly trading directories
- Monthly and flash updates
- Personalized trader listings
- Trader support systems
- Plus much more

Imagine having access to hundreds, even thousands of programs, each costing no more than the price of a stamp or phone call. Join now to beat the price increase and find out about the new member $$ offer. $49.95 — U.S. $64.95 FOREIGN. SASE for info. Indicate computer model.

SOFTRADERS™ INTERNATIONAL
[illegible]

ATARI AND VIC-20

CONFRONTATION AT VEGA THREE: The spacewar strategy-adventure game that transforms your computer into a starship. Realistic tactical displays, intelligent enemy ships and full-blown command logic make this one a galactic commander's dream! (16K required) **$12.95 on cassette**

TRIPLE PLAY: Three games for less than the price of one: X-Patrol, Pickups, and Balls! (no expansion required) **$9.95 on cassette**

Send check or m/o to:

Cedar Island Software

P.O. Box 51545
Jacksonville Beach, FL 32250

Prices include postage/handling. Florida residents add 5% sales tax. Atari & VIC-20 are registered trademarks.

FLY Your Computer

FLIGHT SIMULATOR GAMES

Sky Pilot (8K VIC-20)	**$18.00**
Runway 20 (16K VIC-20)	**$25.00**
Runway 64	**$25.00**
Micro-Pilot (EPSON HX-20)	**$18.00**

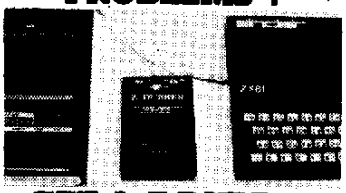
ADD $2.00 FOR DISK VERSION

TORPEDO!

Submarine Battle Games
8K VIC-20 or Commodore 64 **$25.00**

SUSIE SOFTWARE
709 Wilshire Dr. Mt. Prospect, IL 60056
(312) 394-5165

HAVING LOADING PROBLEMS ?



GET A Z-DUBBER

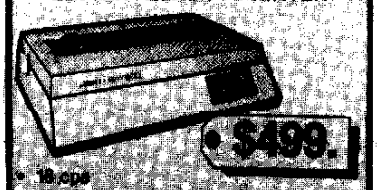
Because of the great variation in cassette recorders used with the ZX80/81/TS1000 (some in very poor shape), you may be having a hard time loading cassette programs. The (Z-Dubber) connects between the cassette recorder and the computer, and is circuitry produces a much better signal for the computer to read. The (Z-Dubber) also allows you to connect two cassette recorders together to make perfect backup copies. The Z-Dubber can be yours for $31.95 postpaid. Add $4.50 for shipping outside the US, Canada or Mexico.

CHARGE OR COD ORDERS CALL
1-800-227-3800
ASK FOR OPERATOR 225

BYTESIZE MICRO TECHNOLOGY PO BOX 12309 DEPT DM SEATTLE, WA 98111 (206) 236-BYTE

CALL OR WRITE FOR DEALER INFORMATION

JUKI 6100
LETTER QUALITY PRINTER



- 18 cps
- Word processing and graphic functions
- Proportional spacing control
- User-[illegible]
- [illegible]

Free shipping in U.S.

COMPUTERS and more

2420 University Ave., San Diego, Ca. 92104
To Order (619) 291-1442

SAVE 35%, 50% OR MORE

on your computer purchase.
Dealer close-out on micros, printers, programs, and accessories.

FOR EXAMPLE	RETAIL	YOUR PRICE
CBM 4022 Printer	695.00	395.00
Smith Corona TPI	995.00	595.00
ODM 64K Expansion	500.00	290.00
BPI A-R	495.00	195.00
Eagle III	4995.00	3995.00
Okidata 83A Printer	995.00	595.00

DESCRIPTION	Your Price	Save
NEC 9" green monitor	189.00	138.00
NEC general acct.	395.00	237.00
NEC DBASE II	700.00	450.00
EBS A/R CBM	1000.00	500.00
Info design A/R	750.00	400.00
Info design payroll	750.00	400.00
Cimmarron medical	495.00	250.00

Catalog lists hundreds of good buys from various dealers. You can make your own deal via phone. TO ORDER current catalog send $5.00 check or VISA/MC number with expiration date.

CompuAd Enterprises
1537A Fourth Street, Suite 125
San Rafael, California 95401

BUSINESS SOFTWARE PACKAGE

$59
Reg. $159

Features:

- Accounting Ledger
- Inventory Control
- Letter Writer
- Mail-Out

IBM (Peanut) VIC-20
TI-99/4A COM-64

Toll Free 1-800-221-7171
In Ga. 1-404-925-4686

BIZWARE, INC. 5014 Hwy. 29 Lilburn, Ga. 30247

ATTENTION
TEXAS INSTRUMENTS TI-99/4A OWNERS

We have hundreds of 3rd party independent software programs on cassette and disk ready to run on your TI-99/4A. Games, business, and educational programming at discount prices as low as $8.95 ea. Plus all TI hardware and software at incredibly low, low prices, including the new TI-99/2 and CC-40 computers. We also have dust covers, heavy duty joysticks with TI adapters, and many more accessories. Call or write now for a FREE listing. We ship your order U.P.S. the same or next business day to insure fast service. Visa and MasterCard accepted (NO service charges) or C.O.D. is okay.

THE MUSIC WORKSHOP
59 E. Tioga St.
Tunkhannock, PA 18657
CALL 1-717-836-4522