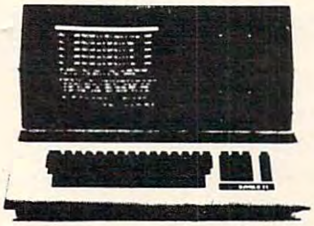
FRANKLIN ACE1000

64K Personal Computer

Hardware, Software and peripheral compatable with the Apple II and even has some features not found on the Apple.

EAGLE COMPUTER



64K RAM
780 KB Disk Storage
Word Processing, Ultracalc CP/M, CBasic Software
Smith Corona TP1 Letter Quality Printer

Our Price **$2995.00**

Retail Value $4895.00

commodore

8032	$999.00
CBM 64	CALL
4032	$749.00
8096 Upgrade Kit	$369.00
Super Pet	$1599.00
2031	$369.00
8250 Double Sided Disk Drive	$1699.00
D9060 5 Megabyte Hard Disk	$2399.00
D9090 7.5 Megabyte Hard Disk	$2699.00
8050	$1299.00
4040	$969.00
8300 (Letter Quality)	$1549.00
8023	$599.00
4022	$399.00
New Z-Ram, Adds CP/M and 64K Ram	$549.00
The Manager	$209.00
Magis	CALL
Word Pro 5 plus	$319.00
Word Pro 4 plus	$299.00
Word Pro 3 plus	$199.00
The Administator	$379.00
InfoPro Plus	$219.00
Power	$79.00
VIC 20 Dust Cover	$6.99
CBM 8032 Dust Cover	$14.99
CBM 8050/4040 Dust Cover	$10.99

HEWLETT PACKARD

HP•85 $1969

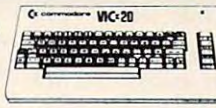
HP-125	$1969.00
HP-85 16K Memory Module	$169.00
5 1/4" Dual Master Disk Drive	$1799.00
Hard Disk w/ Floppy	$4349.00
Hard Disk	$3549.00
"Sweet Lips" Plotter	$1199.00
80 Column Printer	$649.00

IBM® PC

Call...
for price and availability on IBM-PC hardware, soft ware and peripherals.

NEC 3550 Printer (for IBM)	$2099.00

VIC 20 $179



VIC 1530 Commodore Datassette	$69.00
BIC 1540 Disk Drive	$339.00
VIC 1541 (64 Disk Drive)	CALL
VIC 1525 Graphic Printer	$339.00
VIC 1210 3K Memory Expander	$32.00
VIC 1110 8K Memory Expander	$53.00
16K VIC Expansion	$94.00
VIC 1011 RS232C Terminal Interface	$43.00
VIC 1112 VIC IEEE-488 Interface	$86.00
VIC 1211 VIC 20 Super Expander	$53.00
VIC Mother Board	$99.00

HEWLETT PACKARD HP 41CV CALCULATOR $209

HP 41C	$149.00
HP 10C	$69.00
HP 11C	$79.00
HP 12C	$114.00
NEW 115C	$109.00
NEW 16C	$114.00

HPIL PERIPHERALS IN STOCK!

Televideo Terminals

910	$579.00
912C	$699.00
920C	$749.00
925C	$749.00
950	$950.00
800A	$1319.00
802	$2649.00
802H	$4695.00
806	$5795.00
816	$9495.00

PRINTERS

Smith-Corona TP-1 $599

C.ITOH (TEC)

Starwriter (F10-40CPS)	$1399.00
Printmaster (F10-55CPS)	$1749.00
Prowriter 80 Col (P)	$499.00
Prowriter 80 Col (S)	$629.00
Prowriter 2 (132 Col)	$799.00

Okidata

82A	$429.00
83A	$659.00
84P	$1079.00
84S	$1199.00

IDS

132 (fully configured)	$1599.00
80 (fully configured)	$1399.00

Call for other configurations.

Daisywriter

Letter Quality	$1049.00

SHARP PC-1500 POCKET COMPUTER

ALSO AVAILABLE:
Printer w/cassette interface
cassette tape recorder
and 4K and 8K RAM EXTENSIONS

NEC COMPUTERS

8001-A	$749.00
8031	$749.00
8012	$549.00

Printers

8023	$549.00
7710/7730	$2399.00
3510/3530	$1599.00

Monitors

JB-1201	$159.00
JC-1201	$329.00
JC-1202	$899.00

computer mail order east

east **800-233-8950** east

IN PA. CALL (717) 327-9575

477 E. THIRD ST., WILLIAMSPORT, PA. 17701

In-stock items shipped same day you call. No risk, no deposit on C.O.D. orders. Pre-paid orders receive free shipping within the continental United States with no waiting period for certified checks or money orders. Add 3% (minimum $3.00) shipping and handling on all C.O.D. and Credit Card orders. NV and PA residents add sales tax. All items subject to availability and price change. **NOTE:** We stock manufacturer's and third party software for most all computers on the market! Call today for our new catalogue.

or greater than signs in the sort comparisons.

If you are alphabetizing, the variable terms will be string variables, such as A$(I).

You may have several items which need to be associated as they are sorted. For example, suppose you have names and scores to be arranged by score. The names and scores are first arranged as N$(1), S(1); N$(2), S(2); etc. In the interchange you would need to sort the S values, and then switch both terms, such as:

```
SS=S(I)
NN$=N$(I)
S(I)=S(I+1)
N$(I)=N$(I+1)
S(I+1)=SS
N$(I+1)=NN$
```

Keep in mind that for sorts for the TRS-80 Color Computer and the VIC-20, you should use lower line numbers and leave out spaces to conserve memory. You may also save memory by naming your variables with only one letter. Too, you could combine a few more lines than I did in these examples. You should, of course, use the VIC-20 abbreviations wherever possible (such as D-shift-I for DIM).

TI-99/4 BASIC Sorts

```
100 REM  TI BASIC BUBBLE SORT
110 DIM A(50)
120 FOR I=1 TO 50
130 RANDOMIZE
140 A(I)=INT(RND*100+1)
150 PRINT A(I);
160 NEXT I
170 PRINT ::
200 LIM=49
210 SW=0
220 FOR I=1 TO LIM
230 IF A(I)<=A(I+1)THEN290
240 AA=A(I)
250 A(I)=A(I+1)
260 A(I+1)=AA
270 SW=1
280 LIM=I
290 NEXT I
300 IF SW=1 THEN 210
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

```
100 REM  TI BASIC SHELL SORT
110 DIM A(50)
120 FOR I=1 TO 50
130 RANDOMIZE
140 A(I)=INT(RND*100+1)
150 PRINT A(I);
160 NEXT I
170 PRINT ::
200 B=1
210 B=2*B
220 IF B<=50 THEN 210
230 B=INT(B/2)
240 IF B=0 THEN 500
250 FOR I=1 TO 50-B
260 C=I
270 D=C+B
280 IF A(C)<=A(D)THEN 340
290 AA=A(C)
300 A(C)=A(D)
310 A(D)=AA
320 C=C-B
330 IF C>0 THEN 270
340 NEXT I
350 GOTO 230
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

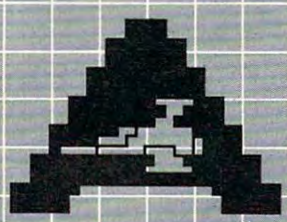
```
100 REM  TI BASIC SORT C
110 DIM A(50)
120 N=50
130 FOR I=1 TO N
140 RANDOMIZE
150 A(I)=INT(RND*100+1)
160 PRINT A(I);
170 NEXT I
180 PRINT ::
200 M=A(1)
210 IM=1
220 FOR I=2 TO N
230 IF A(I)<M THEN 260
240 M=A(I)
250 IM=I
260 NEXT I
270 AA=A(N)
280 A(N)=A(IM)
290 A(IM)=AA
300 N=N-1
310 IF N>1 THEN 200
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

```
100 REM  TI BASIC SORT D
110 DIM A(50)
120 N=50
130 FOR I=1 TO 50
```

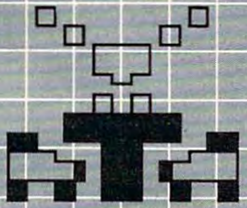
TODAY'S MENU

For Your VIC™ 20 and ATARI® 400/800

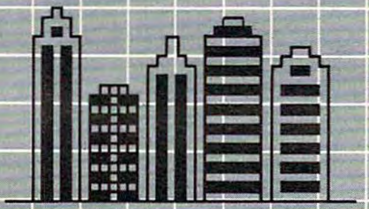
Games



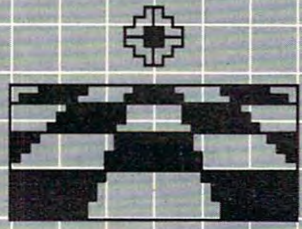
ASTROBLITZ
Protect your planet by destroying enemy saucers.



TRASHMAN
Drive the garbage truck and empty the city's trash cans. But watch out for the flies.



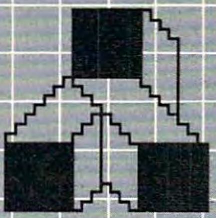
CITY BOMBER
Level a city to make it easy to land. Take off and do it again.



ACTION GAMES
SEAWOLF, BOUNCEOUT, or VIC TRAP. You'll need sharp eyes and quick hands for these.

Education

EDUCATIONAL/RECREATIONAL (I & II)
Put fun into learning math, spelling, and spatial relationships.



LOGIC GAMES
The computer is thinking. You should be, too.

Choose an Entree:

Maybe it's a colorful and challenging game like ASTROBLITZ, TRASHMAN, or CITY BOMBER; perhaps an educational game like HANGMAN or MATH HURDLER; maybe a basic diet of household concerns like HOME INVENTORY, HOUSEHOLD FINANCE, or DECISION MAKER.

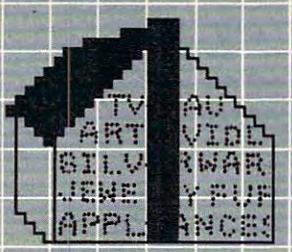
Take your pick. All you have to supply is your VIC™ 20 or ATARI® 400/800 and your own ideas about how to put our software to use. We supply the rest: competitive games, educational games, down-to-earth personal programs, and simple instructions that make everything easy to digest.

What you see here is part of the menu for today. We'll be adding more, so check our menu from time to time. It's growing, and it's all take home.

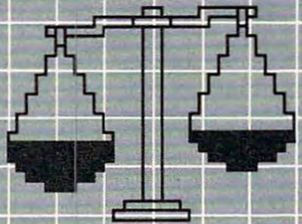
Contact your local outlet.

Personal

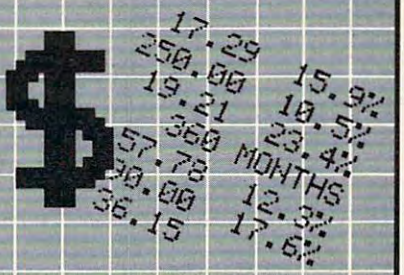
HOUSEHOLD FINANCE
Schedule the family budget, account for expenditures, and face the tax man with a smile.



HOME INVENTORY
Make a record of what you own and revise it as you go. Find out what you've got in seconds.



DECISION MAKER
What should you do about virtually anything? Here's help.



LOAN ANALYZER
Home, car, boat . . . whatever: How much can you afford? Find out.

CREATIVE SOFTWARE
A Division of ASCI, Inc.

201 San Antonio Circle, Mountain View, CA 94040 U.S.A. (415) 948-9595

```
140 RANDOMIZE
150 A(I)=INT(RND*100+1)
160 PRINT A(I);
170 NEXT I
180 PRINT ::
200 S=1
210 MN=A(S)
220 IMIN=S
230 MX=MN
240 IMAX=S
250 FOR I=S TO N
260 IF A(I)<=MX THEN 290
270 MX=A(I)
280 IMAX=I
290 IF A(I)>=MN THEN 320
300 MN=A(I)
310 IMIN=I
320 NEXT I
330 IF IMIN<>N THEN 350
340 IMIN=IMAX
350 AA=A(N)
360 A(N)=A(IMAX)
370 A(IMAX)=AA
380 N=N-1
390 AA=A(S)
400 A(S)=A(IMIN)
410 A(IMIN)=AA
420 S=S+1
430 IF N>S THEN 210
500 FOR I=1 TO 50
510 PRINT A(I);
520 NEXT I
530 END
```

TI-99/4 Extended BASIC Sorts

```
100 REM TI EXTENDED BASIC BUBBLE SO
    RT
110 DIM A(50)
120 FOR I=1 TO 50:: RANDOMIZE:: A(I
    )=INT(RND*100+1):: PRINT A
    (I);:: NEXT I :: PRINT : :
200 LIM=49
210 SW=0 :: FOR I=1 TO LIM :: IF A(
    I)<=A(I+1)THEN 230
220 AA=A(I):: A(I)=A(I+1):: A(I+1)=
    AA :: SW=1 :: LIM=I
230 NEXT I
240 IF SW=1 THEN 210
500 FOR I=1 TO 50 :: PRINT A(I);:: ~
    NEXT I
510 END
```

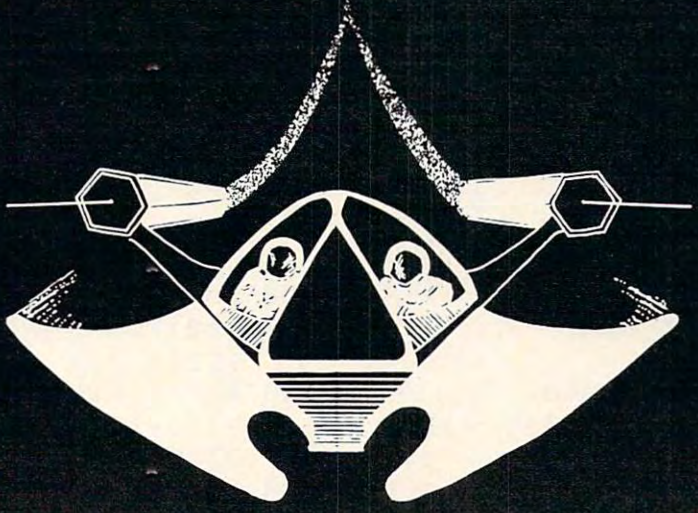
```
100 REM TI EXTENDED BASIC SHELL SOR
    T
110 DIM A(50)
120 FOR I=1 TO 50::RANDOMIZE::A(I)=
    INT(RND*100+1)::PRINTA(I);
    ::NEXT I::PRINT : :
200 B=1
210 B=2*B :: IF B<=50 THEN 210
220 B=INT(B/2):: IF B=0 THEN 500
230 FOR I=1 TO 50-B :: C=I
240 D=C+B :: IF A(C)<=A(D)THEN260
250 AA=A(C):: A(C)=A(D):: A(D)=AA :
    : C=C-B :: IF C>0 THEN 240
260 NEXT I :: GOTO 220
500 FOR I=1 TO 50 :: PRINT A(I);:: ~
    NEXT I
510 END
```

```
100 REM TI EXTENDED BASIC SORT C
110 DIM A(50):: N=50
120 FOR I=1 TO N::RANDOMIZE::A(I)=I
    NT(RND*100+1)::PRINT A(I);
    ::NEXT I::PRINT : :
200 M=A(1):: IM=1
210 FOR I=2 TO N
220 IF A(I)>=M THEN M=A(I):: IM=I
230 NEXT I
240 AA=A(N):: A(N)=A(IM):: A(IM)=AA
     :: N=N-1 :: IF N>1 THEN 2
    00
500 FOR I=1 TO 50 :: PRINT A(I);:: ~
    NEXT I
510 END
```

```
100 REM TI EXTENDED BASIC SORT D
110 DIM A(50):: N=50 :: S=1
120 FOR I=1 TO 50::RANDOMIZE::A(I)=
    INT(RND*100+1):: PRINT A(I
    );:: NEXT I::PRINT : :
200 MN=A(S):: IMIN=S :: MX=MN :: IM
    AX=S
210 FOR I=S TO N
220 IF A(I)>MX THEN MX=A(I):: IMAX=
    I
230 IF A(I)<MN THEN MN=A(I):: IMIN=
    I
240 NEXT I
250 IF IMIN=N THEN IMIN=IMAX
260 AA=A(N):: A(N)=A(IMAX):: A(IMAX
    )=AA :: N=N-1
270 AA=A(S):: A(S)=A(IMIN):: A(IMIN
    )=AA :: S=S+1
280 IF N>S THEN 200
500 FOR I=1 TO 50 :: PRINT A(I);:: ~
    NEXT I
510 END
```

continued on p. 104

GAME PROGRAM DEVELOPMENT KIT



for the

COMMODORE VIC - 20

VIC - 20 is a registered trademark of Commodore Business Machines, Inc.

SIX TOOLS TO HELP YOU WRITE YOUR OWN FAST ACTION ARCADE-STYLE GAMES

DECODER — Decodes programs written in machine language (like game cartridges, utility cartridges, and even the computer's own internal operating programs). Produces a program in an English-like language (Assembler) which can be studied to figure out how they did it. The programs created with the decoder can be customized with the **EDITOR** AND INCORPORATED INTO YOUR OWN NEW GAME PROGRAM. The **ASSEMBLER** turns your programs created with the Decoder and the Editor back into machine language and puts them out to tape or disk so the **LOADER** can load them into the computer's memory to be tested and RUN. The **MONITOR** assists you in debugging your new game program by allowing you to run it a step at a time and making modifications if you need to. The **INSTRUCTION GUIDE** is written so that even a beginner can learn the skills needed to become a pro!!!

$49.95 plus $2.00 p&h buys the kit that could make you rich. Why wait?

Send check, M.O., VISA/MC ($2.00 s.c., please include expiration date), or specify COD (add $3.00) to:

French Silk smooth ware

P.O. Box 207, Cannon Falls, MN 55009

507-263-4821

TRS-80 Color Computer Sorts

```
100 REM TRS80C BUBBLE SORT
110 DIM A(50)
120 FOR I=1 TO 50:A(I)=RND(100):PRI
    NTA(I);:NEXT:PRINT:PRINT
200 LIM=49
210 SW=0:FOR I=1 TO LIM:IF A(I)<=A(
    I+1)THEN 230
220 AA=A(I):A(I)=A(I+1):A(I+1)=AA:S
    W=1:LIM=I
230 NEXT
240 IF SW=1 THEN 210
500 FOR I=1 TO 50:PRINTA(I);:NEXT
510 END
```

```
100 REM TRS80C SHELL SORT
110 DIM A(50)
120 FOR I=1 TO 50:A(I)=RND(100):PRI
    NTA(I);:NEXT:PRINT:PRINT
200 B=1
210 B=2*B:IF B<=50 THEN 210
220 B=INT(B/2):IF B=0 THEN 500
230 FOR I=1 TO 50-B:C=I
240 D=C+B:IF A(C)<=A(D)THEN 260
250 AA=A(C):A(C)=A(D):A(D)=AA:C=C-B
    :IF C>0 THEN 240
260 NEXT:GOTO 220
500 FOR I=1 TO 50:PRINTA(I);:NEXT
510 END
```

```
100 REM TRS80C SORT C
110 DIM A(50):N=50
120 FOR I=1 TO N:A(I)=RND(100):PRIN
    TA(I);:NEXT:PRINT:PRINT
200 M=A(1):IM=1
210 FOR I=2 TO N
220 IF A(I)>=M THEN M=A(I):IM=I
230 NEXT
240 AA=A(N):A(N)=A(IM):A(IM)=AA:N=N
    -1:IF N>1 THEN 200
500 FOR I=1 TO 50:PRINTA(I);:NEXT
510 END
```

```
100 REM TRS80C SORT D
110 DIM A(50):N=50:S=1
120 FOR I=1 TO N:A(I)=RND(100):PRIN
    TA(I);:NEXT:PRINT:PRINT
200 MN=A(S):IM=S:MX=MN:IX=S
210 FOR I=S TO N
220 IF A(I)>MX THEN MX=A(I):IX=I
230 IF A(I)<MN THEN MN=A(I):IM=I
240 NEXT
250 IF IM=N THEN IM=IX
260 AA=A(N):A(N)=A(IX):A(IX)=AA:N=N
    -1
270 AA=A(S):A(S)=A(IM):A(IM)=AA:S=S
    +1
280 IF N>S THEN 200
500 FOR I=1 TO 50:PRINTA(I);:NEXT
510 END
```

VIC-20 Sorts

```
100 REM VIC 20 BUBBLE SORT
110 DIM A(50)
120 FORI=1 TO 50:A(I)=INT(RND(X)*10
    0+1):PRINTA(I);:NEXT:PRINT
    :PRINT
200 L=49
210 S=0:FOR I=1 TO L:IF A(I)<=A(I+1
    )THEN 230
220 AA=A(I):A(I)=A(I+1):A(I+1)=AA:S
    =1:L=I
230 NEXT:IF S=1 THEN 210
500 FOR I=1 TO 50:PRINTA(I);:NEXT
510 END
```

```
100 REM VIC 20 SHELL SORT
110 DIMA(50)
120 FORI=1 TO 50:A(I)=INT(RND(X)*10
    0+1):PRINTA(I);:NEXT:PRINT
    :PRINT
200 B=1
210 B=2*B:IF B<=50 THEN 210
220 B=INT(B/2):IF B=0 THEN 500
230 FOR I=1 TO 50-B:C=I
240 D=C+B:IF A(C)<=A(D) THEN 260
250 AA=A(C):A(C)=A(D):A(D)=AA:C=C-B
    :IFC>0 THEN 240
260 NEXT:GOTO220
500 FOR I=1 TO 50:PRINTA(I);:NEXT
510 END
```

```
100 REM VIC 20 SORT C
110 DIM A(50):N=50
120 FOR I=1 TO N:A(I)=INT(RND(X)*10
    0+1):PRINTA(I);:NEXT:PRINT
    :PRINT
200 M=A(1):IM=1
210 FORI=2 TO N
220 IF A(I)>=M THEN M=A(I):IM=I
230 NEXT
240 AA=A(N):A(N)=A(IM):A(IM)=AA:N=N
    -1:IF N>1 THEN 200
500 FORI=1 TO 50:PRINTA(I);:NEXT
510 END
```

continued on p. 106

VIC-20 CASSETTE SOFTWARE

Syntax SOFTWARE INC.

THE GAMES PEOPLE PLAY **NOW!**

CRABS

: A new challenge every time.

Agility is the key to successfully guiding HERBIE (the halibut) through the maze, avoiding the deadly gaze of SONIC CRABS while feeding on delectable night crawlers.

The more you eat, the higher your score. Each time you clear the maze of tasty morcels, you will receive more time, additional lives, and a new group of night crawlers, as the game of SURVIVAL continues.

But beware! With the passing of time your presence becomes increasingly aggravating to the KILLER crabs who lurk within, improving the accuracy of their menacing sonic waves.

Set at beginner or advanced levels, each game is played in a totally new maze, and may consist of any number of rounds that start identically for each player.

CRABS can be played using your VIC-20 keyboard or joystick, and will work on all standard VIC-20 memory configurations.

TANK WAR

Your opponent watches closely as the BATTLEFIELD unfolds, and you both carefully plan strategies for the pending CONFLICT. Suddenly, both LASER TANKS fire to initiate movement. You begin to thread the way through your home territory, avoiding obstructions and buildings, as you proceed toward enemy ground.

Outscore the rival tank by destroying enemy buildings, as well as placing direct hits on your opponent during one to one combat. Higher skill levels will add additional targets, mountain ranges and landmines to the battle zone for increasing EXCITEMENT.

One of three skill levels, with a new battlefield created for each game, provides a new challenge for both players every time.

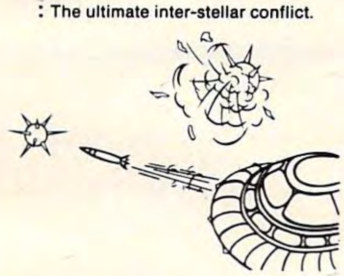
TANK WAR may be played using your VIC-20 keyboard or paddles, and will work on all standard VIC-20 memory configurations.

: Exciting action for two players.

CYCLONS

Full Hi-Res Graphics, Arcade-Like Action

: The ultimate inter-stellar conflict.



Continuing with their plan to conquer the universe, the CYTRON EMPIRE has chosen your sector as the first target in our galaxy. As COMMANDER of the protective forces, you must manoeuvre your craft, avoiding collision and enemy missiles, to attack and destroy enemy war ships.

The CYCLON fighters relentlessly enter the battle zone, attempting to lure you into making errors that will lead to your destruction. The menacing PULSAR DEATH SHIP also begins to attack, its only purpose to zero in on your location, chase you down, and put an end to your defense of civilization as we know it.

Our future lies with your skill.

CYCLON requires memory expansion to function. When loaded on a system with a 3K expander (or Super Expander) you will play an advanced level game. Loading the cassette onto a system with 8K or more expansion, you will be allowed to choose between a variety of difficulty/game-feature options. The game is controlled with the VIC-20 joystick.

Check for availability with your local dealer, or use the order form provided.
Dealer enquiries are welcome.

FORWARD TO:
SYNTAX SOFTWARE INC.
33 ELMHURST AVE., SUITE 502
WILLOWDALE, ONTARIO, CANADA M2N 6G8
PHONE (416) 221-8008

☐ CERTIFIED CHEQUE ☐ MONEY ORDER

Please Forward Charges To:

☐ VISA

☐ MASTER-CARD

☐ AMERICAN EXPRESS

CARD NO. ____________________

EXPIRY DATE ____________________

SIGNATURE ____________________

PLEASE SEND ME:

_____ **CRABS** @ $15.95 (U.S.)/$18.95 (CDN.) = _____

_____ **TANK WAR** @ $15.95 (U.S.)/$18.95 (CDN.) = _____

_____ **CYCLONS** @ $19.95 (U.S.)/$23.95 (CDN.) = _____

SHIPPING & HANDLING @ $1.00 PER CASSETTE = _____

ONTARIO RESIDENTS PLEASE ADD 7% SALES TAX _____

Please Print: **TOTAL** _____

NAME ____________________

ADDRESS ____________________

____

____________________ POSTAL CODE _____

VIC-20 is a registered trademark of Commodore Business Machines Inc.

```
100 REM VIC 20 SORT D
110 DIM A(50):N=50:S=1
120 FOR I=1 TO 50:A(I)=INT(RND(X)*1
    00+1):PRINTA(I);:NEXT:PRIN
    T:PRINT
200 MN=A(S):IM=S:MX=MN:IX=S
210 FOR I=S TO N
220 IF A(I)>MX THEN MX=A(I):IX=I
230 IF A(I)<MN THEN MN=A(I):IM=I
240 NEXT
250 IF IM=N THEN IM=IX
260 AA=A(N):A(N)=A(IX):A(IX)=AA:N=N
    -1
270 AA=A(S):A(S)=A(IM):A(IM)=AA:S=S
    +1
280 IF N>S THEN 200
500 FORI=1 TO 50:PRINTA(I);:NEXT
510 END
```

Apple Sorts

```
100 REM APPLE BUBBLE SORT
110 DIM A(50)
120 FOR I = 1 TO 50:A(I) = INT ( RND (1) * 100
     + 1): PRINT A(I);" ";: NEXT :PRINT:P
    RINT
200 L = 49
210 S = 0: FOR I = 1 TO L: IF A(I) < = A(I + 1
    ) THEN 230
220 AA = A(I):A(I) = A(I + 1):A(I + 1) = AA:S ~
    = 1:L = I
230 NEXT : IF S = 1 THEN 210
500 FOR I = 1 TO 50: PRINT A(I); " ";: NEXT
510 END
```

```
100 REM APPLE SHELL SORT
110 DIM A(50)
120 FOR I=1 TO 50:A(I) = INT ( RND (1) * 100 +
     1): PRINT A(I);" ";: NEXT :PRINT:PRI
    NT
200 B = 1
210 B = 2 * B: IF B < = 50 THEN 210
220 B = INT (B / 2): IF B = 0 THEN 500
230 FOR I = 1 TO 50 - B:C = I
240 D = C + B: IF A(C) < = A(D) THEN 260
250 AA = A(C):A(C) = A(D):A(D) = AA:C = C - B:
     IF C > 0 THEN 240
260 NEXT : GOTO 220
500 FOR I = 1 TO 50: PRINT A(I); " ";:  NEXT
510 END
```

```
100 REM APPLE SORT C
110 DIM A(50):N = 50
120 FOR I = 1 TO N:A(I) = INT ( RND (1) * 100 ~
    + 1): PRINT A(I);" ";:NEXT:PRINT:PRIN
    T
200 M = A(1):IM = 1
210 FOR I = 2 TO N
220 IF A(I) > = M THEN M = A(I) :IM = I
230 NEXT
240 AA = A(N):A(N) = A(IM):A(IM) = AA:N = N - ~
    1: IF N > 1 THEN 200
500 FOR I = 1 TO 50: PRINT A(I); " ";: NEXT
510 END
```

```
100 REM APPLE SORT D
110 DIM A(50):N = 50:S = 1
120 FOR I = 1 TO N:A(I) = INT ( RND (1) * 100 ~
    + 1): PRINT A(I);" ";:NEXT:PRINT:PRIN
    T
200 MN = A(S):IM = S:MX = MN:IX = S
210 FOR I = S TO N
220 IF A(I) > MX THEN MX = A(I): IX = I
230 IF A(I) < MN THEN MN = A(I): IM = I
240 NEXT
250 IF IM = N THEN IM = IX
260 AA = A(N):A(N) = A(IX):A(IX) = AA:N = N - ~
    1
270 AA = A(S):A(S) = A(IM):A(IM) = AA:S = S + ~
    1
280 IF N > S THEN 200
500 FOR I = 1 TO 50: PRINT A(I);" ";: NEXT
510 END
```

©

Why use other computer media when you could be using MEMOREX high quality error free media?

Free Memorex Mini-Disc Offer – Get free discs!
You'll save money when you buy Memorex, because every carton of 10 Memorex 5¼ inch mini-discs sold by Communications Electronics has a coupon good for a free Memorex mini-disc. For every case of 100 Memorex mini-discs you buy from CE, you'll get 10 free Memorex mini-discs, directly from Memorex. The more you order, the more you save. Offer expires December 31, 1982. All Memorex flexible discs sold by CE are of the highest quality, certified 100% error free and backed by a full one year factory warranty.

Flexible Disc Quantity Discounts Available
Memorex Flexible Discs are packed 10 discs to a carton and 10 cartons to a case. Please order only in increments of 100 units for quantity 100 pricing. We are also willing to accommodate your smaller orders. Quantities less than 100 units are available in increments of 10 units at a 10% surcharge. **Quantity discounts** are also available. Order 500 or more discs at the same time and deduct 1%; 1,000 or more saves you 2%; 2,000 or more saves you 3%; 5,000 or more saves you 4%; 10,000 or more saves you 5%; 25,000 or more saves you 6%; 50,000 or more saves you 7% and 100,000 or more discs earns you an 8% discount off our super low quantity 100 price. Almost all Memorex Flexible Discs are immediately available from CE. Our warehouse facilities are equipped to help us get you the quality product you need, when you need it. If you need further assistance to find the flexible disc that's right for you, call the Memorex flexible disc compatibility hotline. Dial toll-free 800-538-8080 and ask for the *flexible disc hotline* extension 0997. In California dial 800-672-3525 extension 0997. Outside the U.S.A. dial 408-987-0997 between 9 AM to 4 PM Pacific Time.

SAVE ON MEMOREX FLEXIBLE DISCS Product Description	Part #	CE quant. 100 price per disc ($)
8" SSSD IBM Compatible (128 B/S, 26 Sectors)	3062	2.09
8" SSSD Shugart Compatible, 32 Hard Sector	3015	2.09
8" SSSD CPT 8000 Compatible, Soft Sector	3045	2.99
8" SSDD IBM Compatible (128 B/S, 26 Sectors)	3090	2.74
8" DSDD Soft Sector (Unformatted)	3102	3.34
8" DSDD Soft Sector (128 B/S, 26 Sectors)	3115	3.34
8" DSDD Soft Sector (256 B/S, 26 Sectors)	3103	3.34
8" DSDD Soft Sector (512 B/S, 15 Sectors)	3114	3.34
8" DSDD Soft Sector (1024 B/S, 8 Sectors)	3104	3.34
5¼" SSDD Soft Sector w/Hub Ring	3481	2.34
5¼" SSDD 10 Hard Sector w/Hub Ring	3483	2.34
5¼" SSDD 16 Hard Sector w/Hub Ring	3485	2.34
5¼" DSDD Soft Sector w/Hub Ring	3491	3.09
5¼" DSDD 10 Hard Sector w/Hub Ring	3493	3.09
5¼" DSDD 16 Hard Sector w/Hub Ring	3495	3.09
5¼" SSQD Soft Sector w/Hub Ring (96 TPI)	3504	2.99
5¼" DSQD Soft Sector w/Hub Ring (96 TPI)	3501	3.99

SSSD = Single Sided Single Density; SSDD = Single Sided Double Density; DSDD = Double Sided Double Density; SSQD = Single Sided Quad Density; DSQD = Double Sided Quad Density; TPI = Tracks per inch.

Special offer on Memorex computer tape.
If you mail your order to us and enclose prepayment, deduct $1.00 per reel from our quantity 100 prices. This means Memorex 25JW can be as low as $12.99 in 100 quantities. Memorex Computer Tapes are packed 10 tapes to a carton. Please order only in increments of 100 units for quantity 100 pricing. Quantities less than 100 units are available in increments of 10 units at a 10% surcharge. **Quantity discounts** are also available. Order 500 or more tapes at the same time and deduct 1%; 1,000 or more saves you 2%; 2,000 or more saves you 3%; 3,000 or more saves you 4%; 4,000 or more saves you 5% and 5,000 or more tapes earns you a 6% discount off our super low quantity 100 price. If you need further assistance or information to find the tape that's right for you, call the Memorex Computer Tape Technical Support Group at (408) 987-2937.

SAVE ON MEMOREX COMPUTER TAPE Product Description	Part #	CE quant. 100 price per reel ($)
Memorex IV 2400 feet Wrightline Seal	25JW	13.99
Memorex IV 2400 feet Easy Load II Cartridge	25JR	14.99
Memorex IV 1200 feet Wrightline Seal	25FW	10.65
Memorex Quantum 2400 feet Wrightline Seal	27JW	16.20
Memorex Quantum 2400 feet Easy Load II Cart.	27JR	16.99
Memorex Quantum 1200 feet Wrightline Seal	27FW	12.50
Memorex Cubic HD 2400 feet Wrightline Seal	39JW	18.99
Memorex Cubic HD 2400 feet Easy Load II	39JR	19.99
Memorex Cubic HD 1200 feet Wrightline Seal	39FW	13.99

New Memorex Lifetime Rigid Disc Pack Product Warranty
All Memorex disc packs sold by CE have a lifetime product warranty. This is your assurance that Memorex disc packs will give you a lifetime of performance and service. Only Memorex can offer you the superior reliability of their exclusive M Formula. In addition, Memorex will assist the original user in isolating and correcting any technical issues that relate to the Memorex product as well as, when appropriate, replace up to one set of read/write heads. If you need further information to find the rigid disc that's right for you, call the Memorex rigid disc compatibility hotline. Dial toll-free 800-538-8080 and ask for the *rigid disc hotline* extension 1642. In California dial 800-672-3525 extension 1642. Outside the U.S.A. dial 408-987-1642.

SAVE ON MEMOREX RIGID DISC PACKS Product Description	Part #	CE quant. one price per pack ($)
Mark III 5 MB. Cartridge Front Load (8 to 32 Sect.)	95-522XX-03	65.00
Top Load (1-to 24 sectors)	94-522XX-03	70.00
CMD-16 "Phoenix Type" CDC Cartridge	98-26600-31	160.00
NCR Cartridge	98-26600-32	160.00
Mark VIII 80 MB. Error Free	72-16600-03	330.00
Flag Free	72-26600-03	320.00
Mark XI 200 MB. Error Free	03-35041	720.00
Flag Free	03-35031-02	560.00
DEC Flag Free	03-35031-03	560.00
Mark XII 200 MB. NCR/CDC Flag Free	03-39001-01	515.00
Honeywell Flag Free	03-39000-01	515.00
Mark XIII 300 MB. Error Free	03-47021	795.00
Flag Free	03-47009	670.00
Mark XIV 80 MB. Unformated Error Free	74-16600-03	365.00
Flag Free	74-26600-03	300.00
Honeywell Format Flag Free	74-26600-08	315.00
CDC Format Flag Free	74-26600-09	315.00
Mark XV 300 MB. Error Free	03-49011	825.00
Flag Free	03-49001-01	725.00

Smith-Corona TP-1 Letter Quality Printer Special Offer
Buy any Memorex product on this page, and get a Smith-Corona TP-1 letter quality printer for only $585.00 plus $20.00 shipping. Specify serial or parallel version.

Buy with Confidence
To get the fastest delivery from CE of your Memorex computer products, send or phone your order directly to our Computer Products Division. Be sure to calculate your price using the CE prices in this ad. Michigan residents please add 4% sales tax. Written purchase orders are accepted from approved government agencies and most well rated firms at a 30% surcharge for net 30 billing. All sales are subject to availability, acceptance and verification. All sales are final. Prices, terms and specifications are subject to change without notice. Out of stock items will be placed on backorder automatically unless CE is instructed differently. Minimum *prepaid* order $50.00. Minimum *purchase order* $200.00. International orders are invited with a $20.00 surcharge for special handling in addition to shipping charges. All shipments are F.O.B. Ann Arbor, Michigan. No COD's please. Non-certified and foreign checks require bank clearance.

For **shipping charges** add $8.00 per case or partial-case of 100 8-inch flexible discs or $6.00 per case or partial case of 100 5¼-inch mini-discs. For tape shipping, add $1.00 per reel. For Disc packs add $10.00 per cartridge (Mark III or CMD-16) or $15.00 per disc pack for U.P.S. *ground* shipping and handling in the continental U.S.A.

Mail orders to: Communications Electronics, Box 1002, Ann Arbor, Michigan 48106 U.S.A. If you have a Master Card or Visa card, you may call and place a credit card order. Order toll-free. Dial 800-521-4414. If you are outside the U.S. or in Michigan, dial 313-994-4444. Order your Memorex computer products from CE today.

 Ad #090282

MasterCard VISA MEMBER DIRECT MAIL/MARKETING ASSOCIATION CODE OF ETHICS

Order Toll-Free!
(800) 521-4414
In Michigan (313) 994-4444

MEMOREX
COMPUTER TAPE
DISC CARTRIDGES · RIGID DISCS
FLEXIBLE DISCS

™ COMMUNICATIONS ELECTRONICS™

Computer Products Division

854 Phoenix ☐ Box 1002 ☐ Ann Arbor, Michigan 48106 U.S.A.
Call TOLL-FREE (800) 521-4414 or outside U.S.A. (313) 994-4444

It hit the national news wires and was quickly picked up by local media: a letter in October's New England Journal of Medicine *suggested that home computers and video game machines used with old color TV sets could expose people to potentially hazardous doses of radiation.*

Is Your TV A Radiation Hazard?

Tom R. Halfhill
Features Editor

It might be considered a flattering measure of the exploding popularity of home computing that a small item in a medical journal could attract so much attention. Could an old color TV hooked up to your computer or video game really create a radiation hazard? Or was all the fuss just a rerun of the color TV "radiation scare" of the late 1960s? What does it really mean to home computerists and video game addicts?

First, in case you missed the story – or more likely, in case your local media carried a frustratingly abbreviated version – here are the details.

The *New England Journal of Medicine,* a respected medical publication closely watched by the general news media, published a letter from two doctors at the Veterans Administration Medical Center in Washington, D.C. The letter warned that pre-1970 color TVs emit more X-radiation than sets built later. This could pose a danger, especially to young people, when these TVs are hooked up to home computers and video game machines. The doctors reasoned that many families plug their computers and game machines into "spare" color TVs to avoid tying up the household's main set. Also, they noted that people playing video games or involved in programming tend to sit much closer to the screen than they do when watching TV shows. They also tend to become engrossed for hours.

Doctors Suggest Caution

Close exposure over prolonged periods to older-model color TVs raises the possibility of radiation doses larger than recommended limits, suggested the doctors. Specifically, a young person using a computer or video game for two hours a day over one year would receive about eight times the government's recommended limit – which is 100 millirems per year for a person under 18. The two-hour-a-day game addict would absorb 780 millirems in the eyes and 890 millirems in the thyroid gland. (The radiation limits are different for adults, and some adults get higher doses because of their occupations; a typical flight attendant, for example, might get 500 millirems per year due to exposure in the upper atmosphere.)

Now, before you panic and start worrying about acquiring a permanent glow from playing *Space Invaders,* there are several things to keep in mind. First, the doctors' caution covers only *color* TVs made *before 1970* which are used at *closer than average* viewing distances. (The doctors defined the average viewing distances as roughly five feet for children and eight feet for adults.)

Second, the doctors did not actually measure radiation levels or perform any primary research. Instead, they took data published in the late 1960s on TV radiation emissions and used standard formulas to estimate the radiation absorption at closer distances. It was not a formal study.

"It was a lark," says Dr. Louis Korman, one of the letter's authors. "I am not a radiation expert. We were just sitting around one day talking about buying microcomputers, and the subject came up that most people who buy home micros tend to hook them up to older color TVs to avoid tying up the newer set. They'll get this TV from the attic, or buy it used at a shop.... We were aware of the radiation scare in the late sixties and just wanted to caution that these sets should be used with prudence.

"You'll probably see a lot of letters next month from people who'll say we don't know what we're talking about."

One of those letters may well be written by someone from the Electronic Instrument Association. A trade group representing TV manufacturers, the EIA did not take kindly to all the fuss. "We want to make two main points," says Alan Schlosser, EIA public relations director. "There are a statistically insignificant number of pre-1970 color TV sets out there. And also, we believe the people who use home microcomputers tend to use these state-of-the-art devices on up-to-date TV equipment. We don't want to pooh-pooh all this,

KMMM Pascal for PET/CBM $85

A subset of standard Pascal with extensions.
- Machine language Pascal Source Editor with cursor oriented window mode
- Machine Language P-Code Compiler
- P-Code to machine language t' anslator for optimized object code
- Run-time package
- Floating point capability
- User manual and sample programs

Requires 32K Please specify configuration.

EARL for PET (disk file based) $65

Editor, Assembler, Relocater, Linker

Generates relocatable object code using MOS Technology mnemonics. Disk file input (can edit files larger than memory). Links ,multiple object programs as one memory load. Listing output to screen or printer. Enhanced editor operates in both command mode and cursor oriented "window" mode.

RAM/ROM for PET/CBM

4K or 8K bytes of soft ROM with optional battery backup.

RAM/ROM is compatible with any large keyboard machine. Plugs into one of the ROM sockets above screen memory to give you switch selected write protectable RAM.

Use RAM/ROM as a software development tool to store data or machine code beyond the normal BASIC range. Use RAM/ROM TO LOAD A ROM image where you have possible conflicts with more than one ROM requiring the same socket. Possible applications include machine language sort (such as SUPERSORT), universal wedge, Extramon, etc.

RAM/ROM – 4K	$75
RAM/ROM – 8K	90
Battery Backup Option	20

SUBSORT by James Strasma $35

Subsort is an excellent general purpose machine language sort routine for PET/CBM computers. Sorts both one and two dimensioned arrays at lightning speed in either ascending or descending order. Other fields can be subsorted when a match is found, and fields need not be in any special order. Sort arrays may be specified by name, and fields are random length. Allows sorting by bit to provide 8 categories per byte. The routine works with all PET BASICs, adjusts to any memory size, and can co-exist with other programs in high memory.

SuperGraphics 2.0
NEW Version with TURTLE GRAPHICS

SuperGraphics, by John Fluharty, provides a 4k machine language extension which adds 35 full featured commands to Commodore BASIC to allow fast and easy plotting and manipulation of graphics on the PET/CBM video display, as well as SOUND Commands. Animations which previously were too slow or impossible without machine language subroutines now can be programmed directly in BASIC. Move blocks (or rocketships, etc.), or entire areas of the screen with a single, easy to use BASIC command. Scroll any portion of the screen up, down, left, or right. Turn on or off any of the 4000 (8000 on 8032) screen pixels with a single BASIC command. In high resolution mode, draw vertical, horizontal, and diagonal lines. Draw a box, fill a box, and move it around on the screen with easy to use BASIC commands. Plot curves using either rectangular or polar co-ordinates (great for Algebra, Geometry and Trig classes.)

The SOUND commands allow you to initiate a note or series of notes (or even several songs) from BASIC, and then play them in the background mode without interfering with your BASIC program. This allows your program to run at full speed with simultaneous graphics and music.

Seven new TURTLE commands open up a whole new dimension in graphics. Place the TURTLE anywhere on the screen, set his DIRECTION, turn him LEFT or RIGHT, move him FORWARD, raise or lower his plotting pen, even flip the pen over to erase. Turtle commands use angles measured in degrees, not radians, so even elementary school children can create fantastic graphic displays.

Specify machine model (and size), ROM type (BASIC 3 or 4)

SuperGraphics in ROM $45

Volume discounts available on ROM version for schools.

FLEX-FILE for PET/CBM Computers — NEW VERSION II

FLEX-FILE is a set of flexible, friendly programs to allow you to set up and maintain a data base. Includes versatile Report Writer and Mail Label routines, and documentation for programmers to use Data Base routines as part of other programs.

RANDOM ACCESS DATA BASE

Record size limit is 256 characters. The number of records per disk is limited only by record size and free space on the disk. File maintenance lets you step forward or backward through a file, add, delete, or change a record, go to a numbered record, or find a record by specified field (or partial field). Field lengths may vary to allow maximum information packing. Both subtotals and sorting may be nested up to 5 fields deep. Any field may be specified as a key. Sequential file input and output, as well as file output in WordPro and PaperMate format is supported. Record size, fields per record, and order of fields may be changed easily.

MAILING LABELS

Typical mail records may be packed 3000 per disk on 8050 (1400 on 4040). Labels may be printed any number wide, and may begin in any column position. There is no limit on the number or order of fields on a label, and complete record selection via type code or field condition is supported.

REPORT WRITER

Flexible printing format, including field placement, decimal justification and rounding. Define any column as a series of math or trig functions performed on other columns, and pass results such as running total from row to row. Totals, nested subtotals, and averages supported. Complete record selection, including field within range, pattern match, and logical functions can be specified.

FLEX-FILE II by Michael Riley $110

Please specify equipment configuration when ordering.

DISK I.C.U. $40

Intensive Care Unit by L.C. Cargile

COMPLETE DISK RECOVERY SYSTEM FOR CBM DRIVES
- edit disk blocks with ease
- duplicate disks, skipping over bad blocks
- complete diagnostic facilities
- un-scratch scratched files
- check and correct scrambled files
- recover improperly closed files
- extensive treatment of relative files
- optional output to IEEE488 printer
- comprehensive user manual (an excellent tutorial on disk operation and theory).

Furnished on copy-protected disk with manual.
Backup disk available, $10 additional.

PROGRAM YOUR OWN EPROMS $75

Branding Iron EPROM Programmer for PET/CBM software for all ROM versions. Includes all hardware and software to program or copy 2716 and 2532 EPROMs.

PORTMAKER DUAL RS232 SERIAL PORT $63

Two ports with full bipolar RS232 buffering. Baud rates from 300 to 4800. For PET/CBM, AIM, SYM.

CBM Software

TCL Pascal Version 1.6	135
Petspeed BASIC Compiler	225
Integer BASIC Compiler	110
CMAR Record Handler	110
UCSD Pascal (without board)	135
Wordcraft 80	300
BPI Accounting Modules	300
Professional Tax Prep Sys.	600
Intelligent Terminal Emulator	25
ASERT Data Base	375
Personal Tax Calc	55
Dow Jones Portfolio Mgmt.	110
Assembler Development	80
Legal Time Accounting	445

FORTH for PET

BY L. C. Cargile and Michael Riley $50

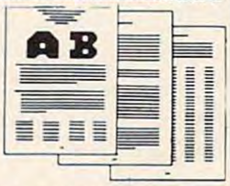
Features include:
- full FIG FORTH model.
- all FORTH 79 STANDARD extensions.
- structured 6502 Assembler with nested decision making macros.
- full screen editing (same as when programming in BASIC).
- auto repeat key.
- sample programs.
- standard size screens (16 lines by 64 characters).
- 150 screens per diskette on 4040, 480 screens on 8050.
- ability to read and write BASIC sequential files.
- introductory manual.
- reference manual.

Runs on any 16K or 32K PET/CBM (including 8032) with ROM 3 or 4, and CBM disk drive. Please specify configuration when ordering.

Metacompiler for FORTH $30

simple metacompiler for creating compacted object code which can be executed independently (without the FORTH system).

PaperMate 60 COMMAND WORD PROCESSOR
by Michael Riley



Paper-Mate is a full-featured word processor for CBM/PET by Michael Riley. Paper-Mate incorporates 60 commands to give you full screen editing with graphics for all 16K or 32K machines (including 8032), all printers, and disk or tape drives. Many additional features are available (including most capabilities of Professional Software's WordPro 3).

For writing text, Paper-Mate has a definable keyboard so you can use with either Business or Graphics machines. Shift lock on letters only, or use keyboard shift lock. All keys repeat.

Paper-Mate text editing includes floating cursor, scroll up or down, page forward or back, and repeating insert and delete keys. Text block handling includes transfer, delete, append, save, load, and insert.

All formatting commands are imbedded in text for complete control. Commands include margin control and release, column adjust, 9 tab settings, variable line spacing, justify text, center text, and auto print form letter (variable block). Files can be linked so that one command prints an entire manuscript. Auto page, page headers, page numbers, pause at end of page, and hyphenation pauses are included.

Unlike most word processors, CBM graphics as well as text can be used. Paper-Mate can send any ASCII code over any secondary address to any printer.

Paper-Mate functions with all CBM/PET machines with at least 16K, with any type of printer, and with either cassette or disk.

To order Paper-Mate, please specify machine and ROM type.

Paper-Mate (disk or tape) for PET, CBM, VIC, C64 **$40**

SM-KIT for PET/CBM $40

Enhanced ROM based utilities for BASIC 4. Includes both programming aids and disk handling commands.

BASIC INTERPRETER for CBM 8096 $200

A full interpreter implementation to automatically take advantage of the extra memory available with 8096.

PEDISK II Systems from cgrs Microtech available.
FILEX IBM 3741/2 Data Exchange Software available.

JINSAM Data Base Management System for CBM.

Comprehensive version available for most configurations.

COPY-WRITER Word Processor for PET/CBM. $159

Works like expensive word processors, plus has added features like double column printing and shorthand generator.

CASH MANAGEMENT SYSTEM $45

Easy to use disk system. Keeps track of cash disbursements, cash receipts, cash transfers, expenses for up to 50 categories.

252 Bethlehem Pike
Colmar, PA 18915 215-822-7727 **A B Computers**

WRITE FOR CATALOG

Add $1.25 per order for shipping. We pay balance of UPS surface charges on all prepaid orders. Prices listed are on cash discount basis. Regular prices slightly higher. Prices subject to change.

but we don't think the body of evidence supports it."

Solid State Is Safer

Congress passed radiation standards for color TVs in the late 1960s, but the standards applied only to new models. Models then in use were not required to be modified because it was never actually proven that they emitted dangerous radiation, says the EIA. The whole scare was triggered when one manufacturer recalled one model which leaked radiation through a small vent-hole in the bottom of the set.

Before Congress passed the regulations, about 25 million color TVs were made between 1960 and 1970, nearly all in the late 1960s. It has been estimated that 1.3 to 16 percent of these sets exceeded the radiation limits set by the Food and Drug Administration's Bureau of Radiologic Health in 1971. Since the average life of a tube-type TV is 11 years, most of these sets are no longer in use.

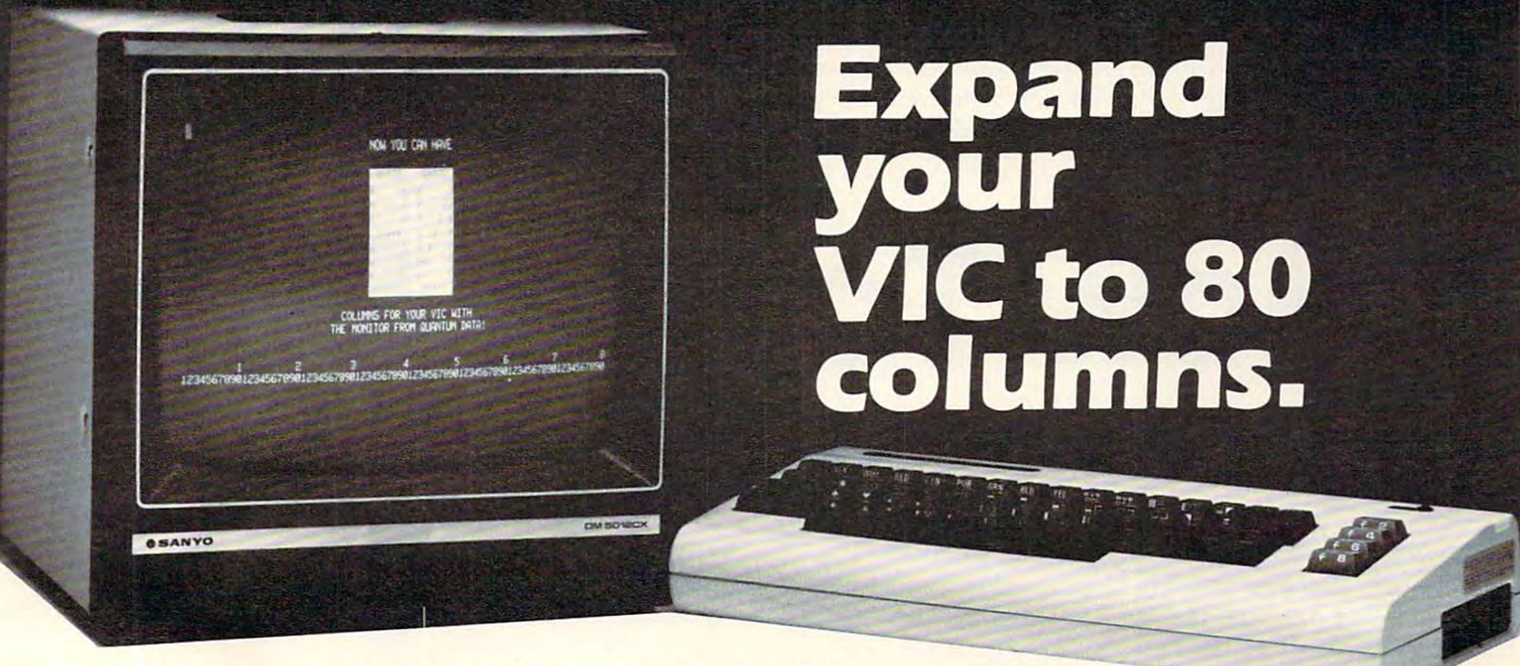
Nearly all the radiation is emitted from the vacuum tubes, not the picture tube. After the scare, manufacturers beefed up the shielding and turned toward safer solid-state circuitry. By 1972, virtually all TVs were solid-state. The greatest hazard is from older TVs which were improperly serviced, says Gene Koschella, who heads the EIA's technical training program. If a serviceman did not replace the tube shielding, or jacked up the voltage to prolong the life of a fading set, more radiation than normal may be leaking from the TV. Due to the nature of the radiation, the dosage is more acute at close range.

"The radiation decreases rapidly as you back away from the set," explains Koschella. "We've taken measurements and found that at four or five feet there's practically no radiation at all. At any rate, the radiation we're talking about is very soft. It's not anything like the radiation you'd get from an atomic bomb or something. In fact, it will be absorbed by clothing or glasses."

(That's why the VA doctors calculated radiation absorption in the eyes and thyroid, areas normally unprotected by clothing – unless the computerist is wearing glasses and a neck scarf.)

If you are using a pre-1970 set for prolonged periods at close range, and are still worried about radiation exposure, Koschella suggests having the TV checked out to insure that no shielding was removed and that the picture tube voltage was not cranked up. But he emphatically warns against checking the voltage yourself – the voltage is very high and probably a lot more dangerous than the radiation. ©

22-40-80 HIKE!



Expand your VIC to 80 columns.

NOW YOU CAN GET A $30.00 FACTORY REBATE WHEN YOU PURCHASE A QUANTUM DATA VIDEO CARTRIDGE OR VIDEO COMBO CARTRIDGE!!

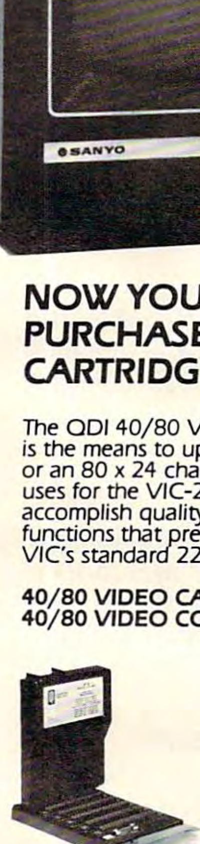
The QDI 40/80 Video Cartridge and Video Combo Cartridge is the means to upgrade the VIC-20 computer to a 40 x 24 or an 80 x 24 character display, providing a wealth of new uses for the VIC-20. With appropriate software, you can now accomplish quality word processing and various business functions that previously were very difficult with only the VIC's standard 22 character video display.

REBATE

40/80 VIDEO CARTRIDGE 0-K memory: $219.95
40/80 VIDEO COMBO CARTRIDGE w/16-K RAM: $319.95

- Features a high quality 8 x 8 dot matrix.
- Character-by-character reverse video attributes allowing adjacent characters to have different attributes.
- All features are accessible through BASIC using POKE commands.
- Black & White composite video.
- Plugs directly into the VIC-20 memory expansion slot or the QDI Mini-Mother and Maxi-Mother boards.
- Contains 2K of CMOS internal video RAM; no system RAM is used by the Video Cartridge.
- 40 Columns can be viewed using your home T.V. while 80 columns require using a video monitor.

Offer expires 12:00 midnight US, pacific standard time, December 31, 1982.

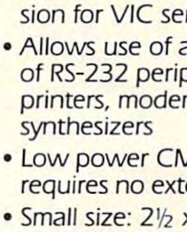


QDI Maxi-Mother: $99.95

SPECIAL INTRODUCTORY OFFER $15.00 FACTORY REBATE!!!

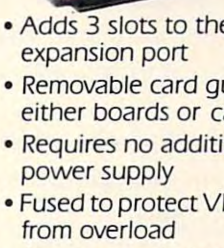
- Adds 6 slots to the memory expansion port.
- Three slots fixed and three slots power switch selectable.
- On board master reset button allows you to reset cold start your VIC-20 without powering down.
- .1" Cannon rear connector accepts common external power supply when fuse is removed.
- Simple plug-in installation.
- Small size 5 x 8 inches encased.

REBATE



QDI RS-232C interface: $49.95

- Provides RS-232 voltage conversion for VIC serial port
- Allows use of a wide variety of RS-232 peripherals including printers, modems and voice synthesizers
- Low power CMOS circuitry requires no external power supply
- Small size: 2½ x 3 inches



QDI Mini-Mother: $59.95

- Adds 3 slots to the memory expansion port
- Removable card guides allow either boards or cartridges
- Requires no additional power supply
- Fused to protect VIC power supply from overload
- Simple plug-in installation

QUANTUM DATA, INC. (714) 966-6553
DEALER HOT LINE (714) 754-1945
3001 Redhill Bldg. 4, Suite 105, Costa Mesa, CA 92626

Shipping cost not included — VISA & MASTERCARD accepted. Dealer inquiries are welcomed. For further information please contact your local Commodore dealer or QUANTUM DATA for a listing of dealers near you. All the above prices retail in US dollars

Part II:

This is the conslusion of a tutorial begun last month. Part II demonstrates how to handle complex multiplication in machine language. Though specific to Commodore machines, the techniques can apply to any microcomputer. In addition to providing an introduction to the use of SYS which allows you to take advantage of the machine language routines in your BASIC's ROM chips – this article also demonstrates a way to pass information between BASIC and machine language.

How To Use SYS

John C. Johnson
McKinney, TX

With knowledge of the subroutines discussed last month, it is now possible to write some extremely powerful machine language extensions to BASIC with reduced effort. Our example problem is a complex arithmetic subroutine; the complex multiplication portion will be discussed in detail. This problem was selected both because it is useful and because it illustrates the concepts of multiple inputs and outputs. (A discussion of the rules for complex arithmetic is given in Ruel V. Churchill's *Complex Variables and Applications*, McGraw-Hill, 1960.) The format for the statement is that given last month in line 200. A and B are the outputs, and C, D, E, and F are inputs; the asterisk (*) signals complex multiplication. The sequence of steps required to produce the result is given below.

1. Fetch the operation character (* or /) and save it.

2. Save the line scanner address for later use.

3. Scan past the output variables.

4. Evaluate each input expression and save it.

5. Save the line scanner position onto the stack, and reset the line scanner to locate the output variables.

6. Test for operation character.

7. Perform the multiplication operation for the real part.

8. Save the result in the output variable #1.

9. Perform the multiplication operation for the imaginary part.

10. Save the result in output variable #2.

11. Fix up the stack and CHRGET address.

A description of the program operation tied to the above description follows. The initialization portion is contained in lines 52 to 64. The purpose of this section is to change the USR vector to point to the start of the subroutine to allow a call with SYS 0. This is important because the conversion time for ASCII 0 is quite efficient, but the time to convert 30747 is substantial.

For example, you could avoid this by assigning 30747 to some variable and call by SYS A1. The efficiency of this approach is slightly better than SYS 0, but lacks the programming convenience. The initialization also sets the top of memory to protect the machine code from BASIC strings. Type SYS 30720 to initialize; the screen will clear and show READY.

The first two steps are accomplished by lines 68-73 and 77-80, respectively. The line scanner is operated to retrieve the operation character, * or /, to determine which of two subroutines will be active. Some error checking is accomplished, and the address of the line scanner is saved.

Accommodating Commodore BASIC

Step 3 is accomplished by lines 84-89. This section merely scans the line for all items between the commas so the line scanner will be positioned for accessing the inputs. One may ask, "Why omit picking up the output addresses at this point?" The reason is strategic and involves the way in which Commodore BASIC handles variables that are subscripted.

The subroutine as written allows subscripted variables as inputs and outputs. If an array element's address is determined before computing the inputs, then the output variable's location may change. This will occur only when a variable is used as an input before it has been defined. The BASIC interpreter will put the variable into the variable list and move array elements as necessary. If the destination variables are skipped at this point and all inputs are evaluated first, this problem will not exist.

DYNACOMP

*Quality software for**:

APPLE II Plus
OSBORNE-1
NORTH STAR***
ATARI
PET/CBM
ALTAIR****
NEC PC-8000
TRS-80 (Level II)**
SUPERBRAIN*****
CP/M Disks/Diskettes
(See Availability box MBASIC/CBASIC)

CARD GAMES

NEW **BRIDGE MASTER** (Available for all computers) Price: $21.95 Diskette
If you liked DYNACOMP'S BRIDGE 2.0, you will absolutely love BRIDGE MASTER. BRIDGE MASTER is a comprehensive bridge program designed to provide hours of challenging competition. Bidding features include the Blackwood convention, Stayman convention, pre-emptive openings, and recognition of demand bids and jump-shift responses. After playing a specific hand, you may replay the same hand, with the option of switching cards with your computer opponents. This feature allows you to compare your bidding and playing skills to BRIDGE MASTER. Bonuses for game contracts and slams are awarded as in duplicate bridge. Doubled contracts are scored based upon a computer assigned vulnerability. A score card is displayed at the conclusion of each hand. The score card displays a summary of total hands played, total points scored, number of contracts made and set, and % bids made. BRIDGE MASTER is clearly the best computer bridge program available.
DYNACOMP'S previous BRIDGE 2.0 customers may upgrade to BRIDGEMASTER for a nominal charge of $5.00 plus postage and handling (see ordering information box). Original cassette/diskette must be returned.

BACCARAT (Atari only) Price: $18.95 Cassette/$22.95 Diskette
This is the European card game which is the favorite of the Monte Carlo jet set. Imagine yourself at the gaming table with 007 to your left and Goldfinger to your right. Learn and play BACCARAT at your leisure on the Atari. Contains full high resolution color graphics and matching sound. Runs in 16K. Requires one joystick.

GIN RUMMY (Apple diskette only) Price: $22.95 Diskette
This is the best micro computer implementation of GIN RUMMY existing. The computer plays exceptionally well, and the HIRES graphics are superb. What else can be said?

POKER PARTY (Available for all computers) Price: $19.95 Cassette/$23.95 Diskette
POKER PARTY is a draw poker simulation based on the book, POKER, by Oswald Jacoby. This is the most comprehensive version available for microcomputers. The party consists of yourself and six other (computer) players. Each of these players (you will get to know them) has a different personality in the form of a varying propensity to bluff or fold under pressure. Practice with POKER PARTY before going to that expensive game tonight! Apple cassette and diskette versions require a 32K (or larger) Apple II.

GO FISH (Available for all computers) Price: $14.95 Cassette/$18.95 Diskette
GO FISH is a classic children's card game. The opponent is a friendly computer with user inputs that are simple enough for small children to easily master. The Apple and Atari versions employ high resolution graphics for the display of hands. A must for children! Runs in 16K.

NEW **BLACKJACK COACH** (32K TRS-80 only) Price: $29.95 Cassette/$33.95 Diskette
BLACKJACK COACH teaches and evaluates professional playing methods. This program will coach you using the Basic and the Complete Card Counting Methods. The BLACKJACK COACH can be used in automatic, unattended play to test the playing and betting strategies you select. Extensive summary reports pinpoint the strengths and weaknesses of various methods of play. All the standard player choices are included: Insurance, splitting pairs, double downs and surrender (optional). A line printer may be used to collect data. If you risk money at the tables, increase your skills with the BLACKJACK COACH!

THOUGHT PROVOKERS

MANAGEMENT SIMULATOR (Available for all computers) Price: $25.95 Cassette/$29.95 Diskette
This program is both an excellent teaching tool as well as a stimulating intellectual game. Based upon similar games played at graduate business schools, each player or team controls a company which manufacturers three products. Each player attempts to outperform his competitors by setting selling prices, production volumes, marketing and design expenditures etc. The most successful firm is the one with the highest stock price when the simulation ends.

FLIGHT SIMULATOR (Available for all computers) Price: $19.95/Cassette/$23.95 Diskette
A realistic and extensive mathematical simulation of take-off, flight and landing. The program utilizes aerodynamic equations and the characteristics of a real airfoil. You can practice instrument approaches and navigation using radials and compass headings. The more advanced flyer can also perform loops, half-rolls and similar aerobatic maneuvers. Although this program does not employ graphics, it is exciting and very addictive. See the software review in COMPUTRONICS. Runs in 16K Atari.

VALDEZ (Available for all computers) Price: $17.95 Cassette/$21.95 Diskette
VALDEZ is a computer simulation of supertanker navigation in the Prince William Sound/Valdez Narrows region of Alaska. Included in this simulation is a realistic and extensive 256 x 256 element map, portions of which may be viewed using the ship's alphanumeric radar display. The motion of the ship itself is accurately modelled mathematically. The simulation also contains a model for the tidal patterns in the region, as well as other traffic (outgoing tankers and drifting icebergs). Chart your course from the Gulf of Alaska to Valdez Harbor! See the software reviews in 80 Software Critique, Personal Computing and Creative Computing.

BACKGAMMON 2.0 (Available for all computers) Price: $19.95 Cassette/$23.95 Diskette
This program tests your backgammon skills and will also improve your game. A human can compete against a computer or against another human. The computer can even play against itself. Either the human or the computer can double or generate dice rolls. Board positions can be created or saved for replay. BACKGAMMON 2.0 plays in accordance with the official rules of backgammon and is sure to provide many fascinating sessions of backgammon play.

NEW **FROG MASTER** (Atari only) Price: $17.95 Cassette/$21.95 Diskette
The Atari APEX first prize winner, FROG MASTER contains exciting arcade features in addition to being a highly educational program. It is a fast-moving high-concentration game for 1-4 players. You score by making touchdowns on the opponents' goal line – if his goalie doesn't get there first. But your players (tadpoles and frogs) must be trained. This is accomplished by giving them a reward at just the right moment when they do something right. This takes precise timing and judgement. Your critters must penetrate barriers and avoid evil line backs if they are to score. Many will fall by the wayside, but some will get through. As they learn you can look inside their heads to see how they think. As you reward them, they reward you (the "thought processes" simulated demonstrate the basic type of animal learning - operant conditioning - widely studied in high school and college courses). As you teach them they teach you how learning takes place! Great graphics! Runs in 16K. Requires two joysticks.

FOREST FIRE! (Atari only) Price: $14.95 Cassette/$18.95 Diskette
Using excellent graphics and sound effects, this simulation puts you in the middle of a forest fire. Your job is to direct operations to put out the fire while compensating for changes in wind, weather and terrain. Not protecting valuable structures can result in startling penalities. Life-like variables are provided to make FOREST FIRE! very suspenseful and challenging. No two games have the same setting and there are 3 levels of difficulty.

CRANSTON MANOR ADVENTURE (North Star, SuperBrain and CP/M only) Price: $19.95 Diskette
At last! A comprehensive Adventure game for North Star and CP/M systems. CRANSTON MANOR ADVENTURE takes you into mysterious CRANSTON MANOR where you attempt to gather fabulous treasures. Lurking in the manor are wild animals and robots who will not give up the treasures without a fight. The number of rooms is greater and the associated descriptions are much more elaborate than the current popular series of Adventure programs, making this game the top in its class. Play can be stopped at any time and the status stored on diskette.

SPACE EVACUATION! (Available for all computers) Price: $15.95 Cassette/$19.95 Diskette
Can you colonize the galaxy and evacuate the Earth before the sun explodes? Your computer becomes the ship's computer as you explore the universe to relocate millions of people. This simulation is particularly interesting as it combines many of the exciting elements of classic space games with the mystery challenge of ADVENTURE.

MONARCH (Atari only) Price: $14.95 Cassette/$18.95 Diskette
MONARCH is a fascinating economic simulation requiring you to survive an 8-year term as your nation's leader. You determine the amount of acreage devoted to industrial and agricultural use, how much food to distribute to the populace and how much should be spent on pollution control. You will find that all decisions involve a compromise and that it is not easy to make everyone happy. Runs in 16K Atari.

NEW **RUBIK'S CUBE SOLVER** (Available for all computers) Price: $14.95 Cassette/$18.95 Diskette/$21.45 Disk
Solving the Rubik's cube puzzle is an exercise in algorithimic logic, and is a "natural" for computer calculation. The RUBIK'S CUBE SOLVER permits you to input the starting state of the 24 facing elements of the cube. It then solves the problem one step at a time, with each step shown as a unfolded view of the cube. Can you solve the cube in fewer steps. In any case, it sure beats disassembling the cube or peeling off and replacing the colors! Requires 16K.

AVAILABILITY

DYNACOMP software is supplied with complete documentation containing clear explanations and examples. Unless otherwise specified, all programs will run within 16K program memory space (ATARI requires 24K). Except where noted, programs are available on ATARI, PET, TRS-80 (Level II), NEC and Apple (Applesoft) cassette and diskette as well as North Star single density (double density compatible) diskette. Additionally, most programs can be obtained on standard (IBM 3740 single density/double density compatible format) 8" CP/M floppy disks for systems running under MBASIC or CBASIC (for example, Altos, Xerox 820, Heath/Zenith and many others). 5¼" CP/M diskettes are available for the North Star, SuperBrain and Osborne computer systems.

**ATARI, PET CBM, NORTH STAR, CP/M, IBM, OSBORNE, SUPERBRAIN, NEC PC-8000 and XEROX are registered tradenames and/or trademarks.*
***Except where noted, all TRS-80 Model I software is available on cassette (only) for the TRS-80 Model III. Exceptions: VALDEZ, CRIBBAGE, GRAFIX, CHESSMASTER. TRS-80 diskettes are not supplied with either DOS or BASIC.*
****For most North Star disk-based systems.*
*****For Altair systems having Microsoft BASIC.*
******For SUBERBRAIN systems running under MBASIC or CBASIC (state which).*

DYNACOMP OFFERS THE FOLLOWING

- Widest variety
- Guaranteed quality
- Fastest delivery
- Friendly service
- Free catalog*
- Toll free order phone

*$2.00 U.S. funds for foreign mailing.

AND MORE...

STARBASE 3.2 (Available for all computers) Price: $13.95 Cassette/$17.95 Diskette
This is the classic space simulation, but with several new features. For example, the Croylins now shoot at the Invincible without warning while also attacking starbases in other quadrants. The Croylins also attack with both light and heavy cruisers and move when shot at! The situation is hectic when the Invincible is besieged by three heavy cruisers and a starbase S.O.S. is received! The Croylins get even! See the software reviews in A.N.A.L.O.G., 80 Software Critique and Game Merchandising.

LIL' MEN FROM MARS (Atari only) Price: $19.95 Cassette/$23.95 Diskette
Defend yourself! The little men from Mars are out to get you if you don't get them first. This is a hilarious high resolution animated graphics (arcade) game which exercises much of the Atari's power. Requires one joystick.

NEW **ALVIN** (Atari only) Price: $17.95 Cassette/$21.95 Diskette
ALVIN is a great arcade game. You are commanding a highly maneuverable ship seeking to destroy several enemy cities. You are attempting to bomb these cities while at the same time trying to avoid their defensive fire (MISSILE COMMAND" in reverse?). Also, your radar has been damaged so that you can only see downwards. This would normally not be much of a problem except that you also have to contend with high-flying enemy aircraft. As long as you are above these aircraft you have the advantage and are safe. However, high level bombing takes considerable skill. Therefore to achieve your goal the best strategy is to swoop down for a bombing run while the enemy craft is out of range, and quickly retreat to the skies. A fun game. Requires 16K.

ESCAPE FROM VOLANTIUM (Atari only) Price: $15.95 Cassette/$19.95 Diskette
Bring the action and excitement of an arcade into your home with ESCAPE FROM VOLANTIUM! To escape you must maneuver your space ship around obstacles and laser blast the guardian (without being eaten). If he is killed with a direct shot (not just a leg lopped off), a door opens to the outside. However, the door does not stay open indefinitely. If you fail to escape in time, the door closes and a new guardian appears. Sometimes you can smash through the door by repeatedly chipping away at it. Other times it is impervious. At the higher levels of play more obstacles and guardians appear, adding to the excitement. Uses high resolution graphics and sound. Runs in 16K.

ALPHA FIGHTER (Atari only) Price: $13.95 Cassette/$17.95 Diskette
Two excellent graphics and action programs in one! ALPHA FIGHTER requires you to destroy the alien starships passing through your sector of the galaxy. ALPHA BASE is in the path of an alien UFO invasion; let five UFO's get by and the game ends. Both games require the joystick and get progressively more difficult the higher you score! ALPHA FIGHTER will run on 16K systems.

THE RINGS OF THE EMPIRE (Atari only) Price: $14.95 Cassette/$18.95 Diskette
The empire has developed a new battle station protected by rotating rings of energy. Each time you blast through the rings and destroy the station, the empire develops a new station with more protective rings. This exciting game runs on 16K systems, employs extensive graphics and sound and can be played by one or two players.

INTRUDER ALERT (Atari only) Price: $15.95 Cassette/$19.95 Diskette
This is a fast paced graphics game which places you in the middle of the "Dreadstar" having just stolen its plans. The droids have been alerted and are directed to destroy you at all costs. You must find and enter your ship to escape with the plans. Five levels of difficulty are provided. INTRUDER ALERT requires a joystick and will run on 16K systems.

MIDWAY (Atari 32K only) Price: $14.95 Cassette/$18.95 Diskette
MIDWAY is an exciting extension of the game of Battleship. It mixes the challenges of strategy and chance. Your opponent can be another human or the computer. Color graphics and sound are both included. Runs in 16K.

NEW **GOLF PRO** (Atari only) Price: $17.95 Cassette/$21.95 Diskette
Both realism and beautiful graphics are joined together in GOLF PRO to produce the best golf simulation available. To really appreciate this game, you should have a color TV so that you can see the green of the fairway, the blue of the water hazards, and the white sand of the traps. You tee off with a wood, use your wedge in the sand trap, and putt on the green just as would be done on the course. Show off the Atari to your friends with GOLF PRO. Requires 16K and one joystick.

GAMES PACK I (Available for all computers) Price: $14.95 Cassette/$18.95 Diskette
GAMES PACK I contains the classic computer games of BLACKJACK, LUNAR LANDER, CRAPS, HORSERACE, SWITCH and more. These games have been combined into one large program for ease in loading. They are individually accessed by a convenient menu. This collection is worth the price just for the DYNACOMP version of BLACKJACK.

GAMES PACK II (Available for all computers) Price: $14.95 Cassette/$18.95 Diskette
GAMES PACK II includes the games CRAZY EIGHTS, JOTTO, ACEY-DUCEY, LIFE, WUMPUS and others. As with GAMES PACK I, all the games are loaded as one program and are called from a menu. You will particularly enjoy DYNACOMP's version of CRAZY EIGHTS.
Why pay $9.95 or more per program when you can buy a DYNACOMP collection for just $14.95?

MOON PROBE (Available for all computers) Price: $12.95 Cassette/$16.95 Diskette
This is an extremely challenging "lunar lander" program. The user must drop from orbit to land at a predetermined target on the moon's surface. You control the thrust and orientation of your craft plus direct the rate of descent and approach angle. Runs in 16K Atari.

SPACE TRAP (Atari only, 16K) Price: $14.95 Cassette/$18.95 Diskette
This galactic "shoot 'em up" arcade game places you near a black hole. You control your spacecraft using the joystick and attempt to blast as many of the alien ships as possible before the black hole closes about you.

SUPER SUB CHASE (Atari only) Price: $19.95 Cassette/$23.95 Diskette
SUPER SUB CHASE simulates a search and destroy mission. Set your course and keep an eye on the sonar readings as you hunt for the hidden submarine. Set the depth charge explosion depth and watch them sink towards the sub. This is an addictive game which takes advantage of the Atari's graphics and sound capabilities. One or two players. Joystick(s) required.

TWO PLAYER GAMES

NEW **TWO PLAYER GAMES** (Available for all computers: 32K disk/diskette only)
DYNACOMP has acquired the distribution rights to the best eight of Xitron's war games. These two-player games were originally written for the North Star computer, but have since been converted to play on all of the computers currently supported by DYNACOMP. Because our licensing and development costs were so low, DYNACOMP offers these programs two to a disk(ette) for only $19.95/diskette, $23.95/disk. If you like war games, then this is a bargain you can not pass up.

Set #1: PANZER and BLITZKRIEG
PANZER
Date: 23 Nov. 1943 Place: Several miles west of Kiev, Russia. The Russians have just liberated Kiev and are moving quickly to reach the German forces which are preparing for a last desperate attempt to halt the Russian advance.
BLITZKRIEG
Date: Spring 1940 Place: Northern France. The German blitzkrieg in the east was complete. Germany had turned its attention to the west: France. The German forces has penetrated the Ardennes and Meuse. The heroism of Dunkirk, the defense of the Aisne-Somme position, and the final collapse of the French armies in the south has all passed. And, now, the drive on Paris . . .

Set #2: STARSHIP TROOPERS and INVASION OF THE MUD PEOPLE
STARSHIP TROOPERS
Date: Fortieth Century Place: Arachnid planet of Sheol. The first all-out battle on the planet Sheol which will match equal forces of Terran and alien units. The outcome will set the course of the conflict, for the planet of Sheol is a key position in the solar war.
INVASION OF THE MUD PEOPLE
A Peruvian army battalion has been dispatched to a remote village area to investigate the destruction of many local dwellings and the disappearance of most of the villagers. Eye-witnesses have reported strange creatures appearing from scores of slimy mud holes which have oddly begun forming across the terrain.

Set #3: FALL OF THE THIRD REICH and ARMORCAR
FALL OF THE THIRD REICH
Date: March, 1945 Place: Remagen, Germany. The allies under General Eisenhower had reached the Rhine. The Germans had failed in destroying the Ludendorff railroad bridge, allowing several allied divisions to cross before it finally collapsed on March 17 ... and so, the allies began their drive on Berlin.
ARMORCAR
Date: 2 Feb. 1944 Place: Minsk, Russia. A German front-line unit is hard pressed for radio equipment and medical supplies. A relief convoy of armored cars must reach them through partisan-infested territory.

Set #4: MOUNT SURIBACHI and MIDDLE EARTH
MOUNT SURIBACHI
Date: 16 Feb. 1945 Place: Iwo Jima. The Japanese opened fire from Mount Suribachi as the marines landed on the porkchop-shaped island. Gunfire from the hill could cover the entire island, thus it was a critical objective if the Americans were to capture and utilize the all-important air field. Mount Suribachi proved to be one of the most strongly defended positions in the Japanese theatre of war.
MIDDLE EARTH
Date: 1997 Place: MIDDLE EARTH. Through a maze of tunnels, crevices, and rocky passageways discovered leading from an inactive volcano in South America, a team of United Nations' researchers have undertaken a mission to an uncharted frontier: the center of the Earth. After a perilous journey spanning a period of several months, the mission has arrived at the Earth's core, a land of flames, steam, oceans, and unforeseen vegetation. And then the creatures of MIDDLE EARTH appeared . . . unmatched by the most frightening horror stories created by man . . .

MISCELLANEOUS

CRYSTALS (Atari only) Price: $14.95 Cassette/$18.95 Diskette
A unique algorithm randomly produces fascinating graphics displays accompanied with tones which vary as the patterns are built. No two patterns are the same, and the combined effect of the sound and graphics are mesmerizing. CRYSTALS has been used in local stores to demonstrate the sound and color features of the Atari. Runs in 16K.

NORTH STAR SOFTWARE EXCHANGE (NSSE) LIBRARY
DYNACOMP now distributes the 23 volume NSSE library. These diskettes each contain many programs and offer an outstanding value for the purchase price. They should be part of every North Star user's collection. Call or write DYNACOMP for details regarding the contents of the NSSE collection.
Price: $9.95 each/$8.95 each (4 or more)
The complete collection may be purchased for $159.95.

NEW **5¼" DISKETTES** (soft sectored/ten sectored) Price: $39.95/20 Diskettes
As you might imagine, DYNACOMP purchases diskettes in large quantities and at wholesale prices. We want to pass the savings along to you!

Step 4 activates the expression evaluator and computes input expressions. This is done by lines 93-107. The program sets the number of inputs to four (line 93), and a loop evaluates the input expressions and stores them in memory locations labeled V1 to V4. Some error checking is also employed. When this loop is finished, the last input will be in V4 and will also remain in the floating accumulator.

Step 5 is accomplished by lines 111-118. This step saves the line scanner address onto the 6502 stack. It will be necessary to restore it before returning to BASIC. The line scanner position is reset so that the subroutine is left in a position to scan for the output variables.

Step 6, lines 122-127, is required to direct the subroutine to the proper segment of code. This method is adequate for small table sizes like this example, but for larger table sizes this technique would certainly not be optimum. An alternate technique can be found in the MONITOR listing in the PET manual.

Steps 7 and 9 (lines 160-173 and 181-196, respectively) begin the actual computation for a complex multiply. The real part is computed first, and the result in line 173 is incorrect by a sign which subroutine NEGATE corrects. Step 9 does much the same thing for the imaginary part.

Steps 8 and 10 are identical in code allowing the use of a subroutine. DEST, lines 217-260, activates the variable lookup for each output variable and stores the contents of the FACC there. This subroutine could be used for any number of numerical outputs. Lines 250-256 handle the special case when the output variable is INTEGER.

Lines 217-227 handle the divide option and cause the FACC to be divided by the magnitude of the complex divisor which was calculated in lines 131-154 if a divide was specified. In this way the complex multiply section is common to both and saves memory. Lines 228-233 saves the FACC temporarily onto the stack, and lines 235-241 restore the FACC to allow the use of subscripted variables as outputs.

Step 11, lines 204-208, concludes the subroutine by retrieving the line scanner address from the stack and placing it into TXTPTR. When BASIC resumes control, the line scanner will be positioned at the end of the calling statement either on a colon or null character to allow BASIC to continue normally.

Speed Increases

The above technique for creating machine language subroutine linkages with BASIC offers considerable flexibility in passing information between the BASIC program and the subroutine. It avoids the problem of having to POKE and PEEK the transferred information.

The program, as written, incorporates a few optimizing decisions both from the standpoint of conserving memory and speeding execution. No claims are made that the program is optimum in either respect. Optimizing in either case is frequently accomplished at the expense of the other. The program was written, however, in a manner that would make the linking concepts described easy to understand.

Ultimately, the results will be put to the test with timing comparisons and with as many different results as there are people trying them. My results, which may not be optimum, show about an eight percent faster execution for a complex multiply and about 30 percent for a complex divide. These results were obtained by carefully allocating the variables for BASIC so that the variable lookup times would be minimized; however, in actual programs the machine language version could show even greter improvement.

The algorithm for the BASIC and machine language versions are the same; they even use the same arithmetic subroutines in ROM. The only saving comes from the variable lookup, which must be done twice for a BASIC program and only once for the machine language version. More complicated subroutines could save considerably more time than this.

Mr. Johnson has offered to supply tape or disk copies of the program for Commodore computers. Send tape (or

Expression Evaluator Summary

1. Uses the line scanner, CHRGET, to obtain input.

2. Starts with the current position of the line scanner.

3. Alternate entry point $CC9F causes the line scanner to back up one address location before evaluating an expression.

4. Uses any valid format for a BASIC statement that can be used on the right-hand side of an equals sign.

5. Leaves evaluated numeric results in the floating accumulator, FACC.

6. Leaves the line scanner on the separator character (comma, colon, or null).

7. Leaves pointers to the location of the string's length and address in the table at $14 and $15.

8. Leaves $7 set to the type of result.

Peripherals Unlimited...

FANTASTIC PRICES!

OUR FAST SERVICE, PRODUCT SELECTION AND OUR CUSTOMERS' SATISFACTION MAKE US #1.

Microsoft Z-80 Softcard	$214
Microsoft RAMCARD	$129
Z-90-64K DO	$2588
Z-19 Terminal	$777
Z-121 Monitor 12"	$149

ATARI COMPUTERS

Atari 800 16K	$649
Atari 400	$318
Atari Interface Module	$174
Atari 810 Single Disk	$444
Atari 830 Modem	$166
Programmer	$59
Entertainer	$84
Star Raiders	$34
16K Mem. Exp. for Atari	$74
32K Mem. Exp. for Atari	$114

NEC PRINTERS

7710/30 Spinwriter R/O	$2295
7720 Spinwriter KSR	$2649
3510/30 Spinwriter R/O	$1689

NEC DOT MATRIX

PC-8023	$474

Call for prices on ribbons, etc.

OKIDATA & MORE

Okidata Microline 82A	$459
Okidata Microline 83A	$699
Okidata Microline 84 (P)	$1069
Okidata Microline 84 (S)	$1099
Tractor (OKI 80 & 82 only)	$60
Diablo 620	$1239
Diablo 630	$1999
IDS Microprism	$539

AMDEK & NEC MONITORS

Amdek 12" 300 GRN Phosphor	$149
Amdek 13" Color I	$319
Amdek 13" Color II	$739
Amdek 13" Color III	$429
NEC JB1201 GRN Phosphor	$149
NEC JC1201 Color	$339

EPSON PRINTERS

MX-80 w/Graphtrax Plus	$469
MX-80FT (Friction & Tractor)	$529
MX-100 (15" Carriage)	$699
Grappler Interface	$129

Call for prices on Ribbons, Cables & Interfaces

NEC-PC 8000 Series Microcomputer

PC-8001A Computer w/32K	$888
PC-8012A w/32K + Exp. Slots	$588
PC-8031A Dual Mini Disk	$888
PC-8032A Add-on Mini Disk	$777

Call for Software Prices

FOR THE LOWEST PRICE CALL TOLL FREE

1-800-343-4114

ORDERING INFORMATION

Our order lines are open 9AM to 6PM EST Monday thru Friday. Phone orders are welcome; same day shipment on orders placed before 10AM. Free use of Mastercard and VISA. Personal checks require 2 weeks clearance. Manufacturer's warranty included on all equipment. Prices subject to revision. C.O.D.'s accepted.

For service, quality and delivery call:

PERIPHERALS UNLIMITED
(617) 655-7400
62 N. Main St. • Natick, MA 01760

disk), a SASE mailer, and $3.

John C. Johnson
RFD 2 Box 260
McKinney, TX 75069

References:

1. Donahue, Carroll S., and Janice K. Enger. *PET/CBM Personal Computer Guide.* Berkeley, California: Osborne/McGraw-Hill, 1980.

2. Butterfield, Jim. "Machine Language Scanning the Stack." **COMPUTE!**, January 1981, #8, pp. 102-106.

3. Churchill, Ruel V. *Complex Variables and Applications.* New York: McGraw-Hill, 1960.

```
LINE# LOC   CODE      LINE
0009  0000            ;SYSTEM EQUATES

0011  0000            USRVEC=0              ;USR VECTOR JUMP
INSTRUCTION
0012  0000            INTFLG=8              ;INTEGER FLAG
0013  0000            STRFLG=7              ;STRING FLAG
0014  0000            INDEX1=$1E            ;INDIRECT INDEX #1
0015  0000            MEMSIZ=34             ;TOP OF RAM POINTER
0016  0000            FACC=$5E              ;FLOATING ACCUMULATOR
#1
0017  0000            FSIGN=FACC+5          ;SIGN OF FACC
0018  0000            VARADR=$44            ;LOCATION OF VARIABLE
IN TABLE
0019  0000            CHRGET=$70            ;GET NEXT CHARACTER
0020  0000            CHRGOT=$76            ;GET LAST CHARACTER
0021  0000            TXTPTR=$77            ;CHRGET ADDRESS
0022  0000            NEGATE=$DEA1          ;CHANGE SIGN OF FACC
0023  0000            FACALT=$DB18          ;TRANSFER FACC TO AFAC
0024  0000            LDFACC=$DAAE          ;LOAD FACC FROM MEMORY
0025  0000            STFACC=$DAE0          ;STORE FACC INTO MEMORY
0026  0000            FLPINT=$D09A          ;FLOAT TI INT
CONVERSION
0027  0000            LOOKUP=$CF6D          ;LOCATE VARIABLE IN
TABLE
0028  0000            EXEVAL=$CCA7          ;EVALUATE EXPRESSIONS
0029  0000            STXERR=$CE03          ;PRINT 'SYNTAX ERROR'
0030  0000            PRTERR=$C357          ;PRINT ERROR MESSAGE
0031  0000            FADD=$D773            ;FLOATING ADDITION
0032  0000            FSUB=$D733            ;FLOATING SUBTRACTION
0033  0000            FMUL=$D934            ;FLOATING
MULTIPLICATION
0034  0000            FMUL1=FMUL+3          ;FACC=AFAC*FACC
0035  0000            FDIV=$DA1B            ;FLOATING DIVISION
MEM/AFAC
0036  0000            FDIV1=$DA11           ;AFAC/MEM WITHOUT SIGN
0037  0000            FDIV2=$DA1E           ;DIVIDE AFAC BY FACC
WITH /0 CK
0038  0000            WRT=$FFD2             ;OUTPUT CHARACTER


0040  0000                  *=30720         ;$7800 STARTING ADDRESS

0042  7800            ;COMPLEX ARITHMETIC

0044  7800            ;SYNTAX FORMAT:  SYS 0,*,A,B,C,D,E,F
0045  7800            ;
0046  7800            ;* IS OP CHARACTER * OR /, A & B ARE OUTPUTS
0047  7800            ;
0048  7800            ;C, D, E, AND F ARE INPUTS
0049  7800            ;
0050  7800            ;INPUTS CAN BE ANY VALID BASIC NUMERIC
EXPRESSION

0052  7800  A9 78     INIT   LDA #>INIT      ;SET TOP OF MEMORY
0053  7802  30 06            BMI B1          ;IF < $8000
0054  7804  A2 00            LDX #<INIT      ;TO PROTECT THIS
PROGRAM
0055  7806  85 23            STA MEMSIZ+1    ;FROM BASIC
0056  7808  86 22            STX MEMSIZ
0057  780A  A9 4C     B1     LDA #$4C        ;JUMP INSTRUCTION
0058  780C  A0 78            LDY #>CARITH    ;PATCH IN THE USR
VECTOR
0059  780E  A2 1B            LDX #<CARITH    ;TO ALLOW CALL WITH
SYS0,...
0060  7810  85 00            STA USRVEC      ;JMP
0061  7812  86 01            STX USRVEC+1    ;LO
0062  7814  84 02            STY USRVEC+2    ;HI
0063  7816  A9 93            LDA #147        ;CLEAR SCREEN
0064  7818  4C D2 FF         JMP WRT

0066  781B            ;1. FETCH THE OPERATION CHARACTER AND SAVE IT

0068  781B  20 70 00  CARITH JSR CHRGET      ;GET THE OP CHAR
0069  781E  F0 5E            BEQ ER1         ;EXIT IF : OR NULL
0070  7820  8D 70 79         STA OPCHAR      ;AND SAVE IT
0071  7823  20 70 00         JSR CHRGET      ;GET COMMA SEPARATOR
0072  7826  C9 2C            CMP #',         ;CHECK IT
0073  7828  D0 54            BNE ER1

0075  782A            ;2. SAVE LINE SCAN ADDRESS

0077  782A  A5 77            LDA TXTPTR      ;GET AND SAVE
0078  782C  8D 72 79         STA ASAVE       ;THE ADDRESS OF
0079  782F  A5 78            LDA TXTPTR+1    ;THE LINE SCAN
0080  7831  8D 73 79         STA ASAVE+1     ;FOR FUTURE REFERENCE

0082  7834            ;3. SCAN PAST 2 OUTPUT VARIABLES

0084  7834  A2 02            LDX #2          ;SET UP FOR 2 OUTPUTS
0085  7836  20 70 00  SC1    JSR CHRGET      ;GET A CHARACTER
0086  7839  C9 2C            CMP #',         ;CHECK FOR COMMA
0087  783B  D0 F9            BNE SC1         ;LOOP UNTIL COMMA
0088  783D  CA               DEX
0089  783E  D0 F6            BNE SC1         ;LOOP FOR 2 OUTPUT
VARIABLES

0091  7840            ;4. EVALUATE 4 INPUT EXPRESSIONS

0093  7840  A9 04            LDA #4          ;SET FOR 4 PARAMETERS
0094  7842  8D 71 79         STA PARMS       ;SAVE COUNT
0095  7845  20 A7 CC  EV1    JSR EXEVAL      ;EVALUATE EXPRESSION
0096  7848  A5 07            LDA STRFLG      ;CHECK FOR STRING
0097  784A  F0 03            BEQ EV2         ;NOT STRING
0098  784C  4C 14 79         JMP TYPMIS      ;PRINT 'TYPE MISMATCH'
0099  784F  AD 71 79  EV2    LDA PARMS       ;SET UP DESTINATION
ADDRESS
0100  7852  0A               ASL A           ;FOR A FACC STORE OUT
OF
0101  7853  AA               TAX             ;THE VARIABLE TABLE
0102  7854  BC 93 79         LDY VTAB+1,X    ;GET ADDRESS
0103  7857  BD 92 79         LDA VTAB,X      ;MSB IN Y
0104  785A  AA               TAX             ;LSB IN X
0105  785B  20 E0 DA         JSR STFACC      ;STORE #
0106  785E  CE 71 79         DEC PARMS       ;DECREMENT COUNT
0107  7861  D0 E2            BNE EV1         ;CONTINUE UNTIL 0

0109  7863            ;5. ADJUST LINE SCAN

0111  7863  A5 77            LDA TXTPTR      ;GET LINE SCAN ADDRESS
0112  7865  48               PHA             ;AND SAVE IT ON STACK
0113  7866  A5 78            LDA TXTPTR+1
0114  7868  48               PHA
0115  7869  AD 72 79         LDA ASAVE       ;GET PREVIOUS LINE SCAN
0116  786C  85 77            STA TXTPTR      ;AND RESTORE IT
0117  786E  AD 73 79         LDA ASAVE+1
0118  7871  85 78            STA TXTPTR+1

0120  7873            ;6. TEST FOR OPERATION CHARACTER

0122  7873  AD 70 79         LDA OPCHAR      ;CHECK ARITHMETIC
                                             ;'TOKENS'
0123  7876  C9 AC            CMP #172        ;MULTIPLY?
0124  7878  F0 42            BEQ CMUL        ;YES
0125  787A  C9 AD            CMP #173        ;DIVIDE?
0126  787C  F0 03            BEQ CDIV        ;YES
0127  787E  4C 11 79  ER1    JMP ERR         ;WRONG SYMBOL

0129  7881            ;COMPLEX DIVIDE

0131  7881  AD 84 79  CDIV   LDA V4+1        ;CONJUGATE THE DIVISOR
0132  7884  49 80            EOR #$80
0133  7886  8D 84 79         STA V4+1
0134  7889  20 18 DB         JSR FACALT      ;PUT INTO AFAC
0135  788C  A5 5E            LDA FACC        ;SET Z-FLAG
0136  788E  20 37 D9         JSR FMUL1       ;SQUARE IT
0137  7891  A0 79            LDY #>V6        ;SET UP V6 ADDRESS
0138  7893  A2 8D            LDX #<V6
0139  7895  20 E0 DA         JSR STFACC      ;SAVE N V6
0140  7898  A0 79            LDY #>V3        ;SET UP V3 ADDRESS
0141  789A  A9 7E            LDA #<V3
0142  789C  20 AE DA         JSR LDFACC      ;LOAD V3 INTO FACC
0143  789F  20 18 DB         JSR FACALT      ;PUT INTO AFAC
0144  78A2  A5 5E            LDA FACC        ;SET Z-FLAG
0145  78A4  20 37 D9         JSR FMUL1       ;SQUARE IT
0146  78A7  A0 79            LDY #>V6        ;SET UP V6 ADDRESS
0147  78A9  A9 8D            LDA #<V6
0148  78AB  20 73 D7         JSR FADD        ;ADD PREVIOUS RESULT
0149  78AE  A0 79            LDY #>V6        ;SET UP V6 ADDRESS
0150  78B0  A2 8D            LDX #<V6
0151  78B2  20 E0 DA         JSR STFACC      ;SAVE IN V6
0152  78B5  A0 79            LDY #>V4        ;SET UP V4 ADDRESS
0153  78B7  A9 83            LDA #<V4
0154  78B9  20 AE DA         JSR LDFACC      ;RETRIEVE LAST
                                             ; PARAMETER

0156  78BC            ;COMPLEX MULTIPLY

0158  78BC            ;7. FIND REAL PART

0160  78BC  A0 79     CMUL   LDY #>V2        ;SET UP MULTIPLY
0161  78BE  A9 79            LDA #<V2        ;BY MEMORY V2
0162  78C0  20 34 D9         JSR FMUL        ;MULTIPLY
0163  78C3  20 69 79         JSR SAVTMP      ;SAVE FACC IN
                                             ; TEMPORARY V5
0164  78C6  A0 79            LDY #>V1        ;GET ARGUMENT 1
0165  78C8  A9 74            LDA #<V1        ;THE REAL PART
0166  78CA  20 AE DA         JSR LDFACC      ;INTO FACC
0167  78CD  A0 79            LDY #>V3        ;GET ARGUMENT 3
0168  78CF  A9 7E            LDA #<V3        ;THE REAL PART
0169  78D1  20 34 D9         JSR FMUL        ;AND MULTIPLY
0170  78D4  A0 79            LDY #>V5        ;GET TEMPORARY
0171  78D6  A9 88            LDA #<V5        ;RESULT ADDRESS
0172  78D8  20 33 D7  CM1    JSR FSUB        ;SUBTRACT IF
                                             ; MULTIPLICATION
0173  78DB  20 A1 DE         JSR NEGATE      ;FACC=-FACC

0175  78DE            ;8. SAVE RESULT IN FIRST OUTPUT VARIABLE

0177  78DE  20 19 79  CM2    JSR DEST        ;LOOK UP & SAVE RESULT

0179  78E1            ;9. FIND IMAGINARY PART

0181  78E1  A0 79            LDY #>V2        ;GET ADDRESS
0182  78E3  A9 79            LDA #<V2        ;OF V2 AND
0183  78E5  20 AE DA         JSR LDFACC      ;LOAD IT
0184  78E8  A0 79            LDY #>V3        ;DO SAME
0185  78EA  A9 7E            LDA #<V3        ;FOR V3
1086  78EC  20 34 D9         JSR FMUL        ;AND MULTIPLY
1087  78EF  20 69 79         JSR SAVTMP      ;SAVE FACC IN TEMP V5
0188  78F2  A0 79            LDY #>V1        ;LOAD V1
0189  78F4  A9 74            LDA #<V1        ;INTO FACC
0190  78F6  20 AE DA         JSR LDFACC
0191  78F9  A0 79            LDY #>V4        ;AND MULTIPLY
0192  78FB  A9 83            LDA #<V4        ;BY V4
0193  78FD  20 34 D9         JSR FMUL
0194  7900  A0 79            LDY #>V5        ;GET TEMPORARY
```

```
7902  A9 88             LDA #<V5          ;RESULT AND
7904  20 73 D7   CM4    JSR FADD          ;ADD IT

7907             ;10. SAVE RESULT IN SECOND OUTPUT VARIABLE

7907  20 19 79   TERM   JSR DEST          ;LOOK UP,SAVE RESULT

790A             ;11. FIX UP THE STACK AND CHRGET ADDRESS

790A  68                PLA
790B  85 78             STA TXTPTR+1
790D  68                PLA
790E  85 77             STA TXTPTR
7910  60                RTS

7911  4C 03 CE   ERR    JMP STXERR        ;PRINT 'SYNTAX' & EXIT
7914  A2 A3      TYPMIS LDX #$A3          ;PRINT 'TYPE MISMATCH'
7916  4C 57 C3          JMP PRTERR        ;AND EXIT

7919             ;LOOK UP DESTINATION AND STORE FACC THERE

7919  AD 70 79   DEST   LDA OPCHAR        ;SEE IF MULTIPLY
791C  C9 AC             CMP #172          ;TOKEN FOr '*'
791E  F0 10             BEQ D1            ;SKIP NORMLZ.IF MULTIPLY
7920  20 18 DB          JSR FACALT        ;PUT FACC INTO AFAC
7923  A5 63             LDA FSIGN         ;SAVE FACC SIGN
7925  48                PHA
7926  A0 79             LDY #>V6          ;IF DIVIDE THEN
7928  A9 8D             LDA #<V6          ;NORMLZ BY MAG SQUARED
792A  20 11 DA          JSR FDIV1         ;OF DIVISOR
792D  68                PLA               ;RESTORE FACC SIGN
792D  68                PLA               ;RESTORE FACC SIGN
792E  85 63             STA FSIGN
7930  20 70 00   D1     JSR CHRGET        ;MOVE PAST COMMA
7933  A0 05             LDY #5            ;SAVE FACC IN CASE OF
7935  B9 5E 00   D1A    LDA FACC,Y        ;SUBSCRIPTED VARIABLES
7938  48                PHA
7939  88                DEY
793A  10 F9             BPL D1A
793C  20 6D CF          JSR LOOKUP        ;GET DESTINATION ADDR
793F  A0 00             LDY #0            ;RESTORE THE FACC
7941  A2 05             LDX #5
7943  68         D1B    PLA
7944  99 5E 00          STA FACC,Y
7947  C8                INY
7948  CA                DEX
7949  10 F8             BPL D1B
794B  A5 07             LDA STRFLG        ;CHECK FOR STRING TYPE
794D  D0 C5             BNE TYPMIS        ;AND BRANCH IF IT IS
794F  A5 08             LDA INTFLG        ;CHECK FOR INTEGER
7951  F0 0E             BEQ D2

7953             ;CONVERT TO INTEGER FORMAT IF THE
7953             ;DESTINATION VARIABLES IS INTEGER

7953  20 9A D0          JSR FLPINT        ;CONVERT RESULT
7956  A0 01             LDY #1
7958  B9 61 00   D3     LDA FACC+3,Y      ;TRANSFER 2 BYTES
795B  91 44             STA (VARADR),Y    ;FROM FACC TO MEMORY
795D  88                DEY
795E  10 F8             BPL D3
7960  60                RTS
7961  A4 45      D2     LDY VARADR+1      ;FETCH
7963  A6 44             LDX VARADR        ;ADDRESS
7965  20 E0 DA          JSR STFACC        ;AND SAVE RESULT
7968  60                RTS
7969             ;SAVE FACC INTO TEMPORARY V5

7969  A0 79      SAVTMP LDY #>V5          ;SET UP V5 ADDRESS
796B  A2 88             LDX #<V5          ;FOR TRANSFER
796D  4C E0 DA          JMP STFACC

7970             ;STORAGE FOR VARIABLES AND CONSTANTS

7970             OPCHAR *=*+1             ;OPERATION CHAR. + - * /
7971             PARMS  *=*+1             ;# OF PARAMETERS TO GO
7972             ASAVE  *=*+2             ;LINE SCAN ADDRESS
7974             V1     *=*+5             ;FIRST ARGUMENT
7979             V2     *=*+5             ;SECOND ARGUMENT
797E             V3     *=*+5             ;THIRD ARGUMENT
7983             V4     *=*+5             ;FOURTH ARGUMENT
7988             V5     *=*+5             ;TEMPORARY REGISTER #1
798D             V6     *=*+5             ;TEMPORARY REGISTER #2

7992             ;VARIABLE TABLE

7992  88 79      VTAB   .WORD V5,V4,V3,V2,V1,V6
7994  83 79
7996  7E 79
7998  79 79
799A  74 79
799C  8D 79
799E                    .END
```

©

A Monthly Column

The Beginner's Page

An all-purpose learning game for children illustrates how easy it is to make major changes to programs.

Easy Changes

Richard Mansfield
Senior Editor

A program is really two things working together: data and instructions. The instructions are in a numbered list and they are the jobs for the computer to do. The data is the information that gets worked on. That's why computing is sometimes called *data processing*. Your list of instructions to the computer (your program) will process information the way a food processor transforms food. You put in a potato and it comes out french fries.

"Processors" have several advantages over conventional tools. For instance, they are quite versatile. By slipping different cutting disks into a food processor, you instantly change the process. The potato can come out as hash browns, slices, or even soup. A similarly simple adjustment will change a program which calculates home mortgages into one which analyzes inflation or general investment strategy.

Data, the other part of a computing process, is even simpler to change. Change one number and a mortgage-calculating program will print out the payment schedule for a different interest rate. Change another number and you can see the effects of a 20-year instead of a 30-year mortgage. To see how instructions and data interact, and how easily one program can serve many purposes, let's make a general-purpose educational game.

Easy Transformations

One of the most valuable uses for a computer in the home is computer assisted instruction, often called CAI. Using the little program below, you can bring your child's textbooks to life. And if you add color, sound, or animation to this program, you'll have made learning into an exciting game. Good CAI can bring a child the best possible kind of education: joy in learning. Don't be surprised if your child heads for the computer instead of Saturday morning cartoons.

If you type in Program 1, your child can play a short, personalized vocabulary game. You'll want to change the name in lines 100, 190, and 210. Line 130 contains the answers and lines 230 and 240 contain the questions, each followed by the number of the correct answer. The BASIC instruction "READ" will go down these DATA lists, picking each one in order and keeping track of where it left off. To make a much larger game, just add more questions and answers in the same fashion. And be sure to change the number in line 10 to equal the total number of questions in the quiz. To print more answers on the screen, just add more PRINT statements anywhere between lines 130-150.

To easily transform this game into a test of world capitals, just replace the DATA and change the messages in lines 100 and 150. Program 2 demonstrates how little effort it takes to change this into CAI on another topic. Take any textbook and make a list of the facts being taught in it and enter them into the DATA of this program. You could even use numbers like "1 + 5" in place of word answers.

If you make the screen change colors, or add music, or design some graphics characters which dance around ecstatically after a correct answer – you'll add to the attractiveness of this learning game. Perhaps have a little figure put a picture puzzle together, adding new pieces each time the child makes the right guess in the quiz. Or you could construct a game around your child's favorite cartoon character. Have the "hero" of the game climb stairs. A perfect score puts the character at the top where he can open the treasure chest.

Whatever special touches you decide to add, your child is sure to respond to this personalized, interactive, and very patient teacher. And no matter how elaborate the game becomes, it can always be quickly transformed with new questions and answers in the DATA lines. *(continued on p. 124)*

QUICKSOFT™

THE SOFTWARE HOUSE WITH THE DIFFERENCE:

SPEED!

CONGRATULATIONS! You've just taken the First Step in ordering from the Fastest, Lowest Priced Software-House in the Northern Hemisphere!

QUICKSOFT is the Software-House with a difference, **SPEED!!!** When you place an order with **QUICKSOFT**, we Guarantee that it will be shipped from stock within **Twenty-four hours.***

QUICKSOFT HAS THEM ALL, and right now we're having specials on these items from our **"MOST WANTED"** list:

- CYBORG
- BANDITS
- CHOPLIFTER
- PROTECTOR
- ULYSSES
- DUELING DIGITS
- FROGGER
- ALI BABA
- GOLD RUSH
- RASTER BLASTER

YOUR CHOICE ONLY $24.95 each. OR CHOOSE ANY TWO! ONLY $47.95 for both.

Offer Valid Thru This Issue.

FREE DELIVERY: QUICKSOFT offers Free Delivery on **ALL** orders of $50.00 or more. We not only ship within 24 hours but we pay the freight! (For orders of less than $50 add $1.50 for shipping and handling.) Plus **NO SURCHARGE** on **VISA** and **MASTERCARD** orders!

THE NEXT STEP. Pick up the nearest telephone and call us **TOLL FREE** anytime of the day or night. For placing an order or for requesting our **FREE CATALOG** simply dial:

1-800-547-8009

IN OREGON CALL 1-342-1298

Please specify for Apple or Atari computer on your order

***GUARANTEE: QUICKSOFT** guarantees shipment within 24 hours on each phone order. If the item is not readily available and cannot be shipped on time, you will receive an additional **5% OFF** the regular **QUICKSOFT** price.

VISA-MASTERCARD AND C.O.D. WELCOME!

QUICKSOFT™

Department 3, P.O. Box 10854 Eugene, OR 97440

Copyright 1982 Microcomputer Support Group, INC.

*Foreign and Airmail will be extra. C.O.D. orders will be accepted with an additional charge of $5.00 per order. Please allow sufficient time (one week) for personal checks to clear. Prices subject to change without notice.

Program 1: Vocabulary Game

```
10 NUMBER = 5
100 PRINT "HELLO, SUSAN, LET'S PLAY THE VOCABU
    LARY GAME."
110 FOR T = 1 TO 1000: NEXT T: PRINT
120 FOR I = 1 TO NUMBER
130 PRINT "1.SILENT  2.HOPE  3.PERFECT  4.DENT
    IST  5.PRETTY
140 PRINT
150 READ QU$: PRINT QU$ "  -- MEANS THE SAME A
    S WHAT NUMBER ABOVE?"
160 INPUT GUESS: IF GUESS < 1 OR GUESS > NUMBE
    R THEN GOTO 160
170 READ KEY
180 IF KEY <> GUESS THEN PRINT " SORRY, THE RI
    GHT ANSWER IS " KEY: GOTO 200
190 PRINT " GOOD!  YOU GOT IT RIGHT, SUSAN!": ~
    S = S + 1
200 PRINT:NEXT I
210 PRINT: PRINT "SUSAN'S FINAL SCORE IS " S
220 PRINT " TO PLAY AGAIN, JUST TYPE  RUN AND ~
    PRESS THE RETURN KEY"
230 DATA COULDN'T BE BETTER, 3, EXPECT, 2, LOO
    KS NICE, 5
240 DATA FIXES TEETH, 4, MAKES NO NOISE, 1
```

Program 2: Capitals Game

```
100 PRINT "HELLO, SUSAN, LET'S PLAY THE CAPITA
    LS GAME."
130 PRINT "1.ENGLAND 2.FRANCE 3.CHINA 4.EGYPT ~
    5.RUSSIA
150 READ QU$: PRINT QU$ "  -- IS THE CAPITAL O
    F WHICH COUNTRY ABOVE?"
230 DATA PEKING, 3, PARIS, 2, CAIRO, 4
240 DATA MOSCOW, 5, LONDON, 1
```

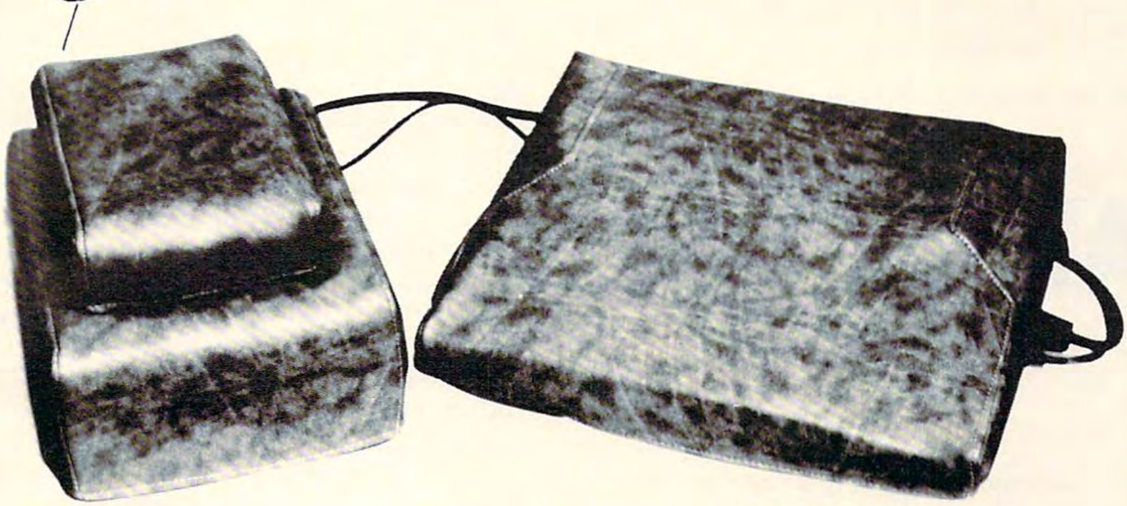
©

Statement Of Ownership, Management And Circulation
As Required by 39 U.S.C. 3685

1. COMPUTE! The Journal for Progressive Computing
1a. 537250
2. 10-1-82
3. Monthly 3A. 12 3B. $20
4. 625 Fulton Street, P.O. Box 5406, Greensboro, NC 27403
5. Same
6. Robert C. Lock, Publisher and Editor-in-Chief, 625 Fulton Street, Greensboro, NC 27403
Kathleen Martinek, Managing Editor, 625 Fulton Street, Greensboro, NC 27403
7. Small System Services, Inc.; Robert C. Lock, William E. Knox, Joretta E. Klepfer, Marlene R. Pratto, P.O. Box 5406, Greensboro, NC 27403
8. None
9. NA

10. Extent and nature of circulation	Average no. copies each issue during preceding 12 months	Actual no. copies of single issue published nearest to filing date
A. Total No. Copies (Net Press Run)	70,640	103,483
B. Paid Circulation		
1. Sales through dealers and carriers, street vendors and counter sales	33,232	66,303
2. Mail Subscription	25,409	30,931
C. Total Paid Circulation	58,641	97,234
D. Free distribution by mail, carrier or other means: samples, complimentary and other free copies	915	2,121
E. Total Distribution	59,556	99,355
F. Copies Not Distributed		
1. Office use, left over, unaccounted, spoiled after printing	5,705	4,128
2. Returns from news agents	5,380	None to date
G. Total	70,640	103,483

I certify that the statements made by me above are correct and complete.
W. Jerry Day, Vice President/Comptroller



For PET, VIC, and Atari, "Name Play" is a user-friendly program aimed at preschoolers. Children who are just learning to recognize letters will be able to take pride in their ability to write the names of their favorite people. Up to nine different names may be included. A printer is required for program output.

To run the PET/CBM version on the VIC-20 the following changes must be made:

line 120 – change PRINTTAB(12) to PRINT
line 305 – change 44 to 22

On the VIC-20, names must be no more than 15 characters long.

NAME PLAY

Bob Sullivan
Oak Park, IL

Youngsters will enjoy producing a printout of the names typed into the computer. These printouts are great for copying with crayons.

After the REM statements are removed, the program uses less than 1K and takes only a few minutes to type into the computer. First, personalize the data list in lines 1000-1080 with the names of family members, pets, friends, and close relatives. Next, assist your neophyte computerist with the following commands:

1) Press 1-9 for the desired name.
2) Press the correct sequence of letters.
3) Press @ for a printout of copied names.
4) Press the home key to turn the screen off or on.

To break into the program, make sure that the screen is off, and then press the STOP key.

This program works well with the QUADRA-PET techniques that were outlined in the July 1981 issue of **COMPUTE!**:

1) Load and run QUADRA-PET.
2) SYS926 and NEW each PET.
3) Append NAME SUCCESS into PET 4.
4) SYS926 to PET 1.

If you avoid machine language and greater than (>) DOS commands, PET 1 will operate, load, and save as an ordinary 8K PET. Additionally, you will be able to switch from PET 1 to PET 2 in less than six seconds, thus allowing yourself a short break while the young ones are in the mood for *their* program.

Beginner's Note

The key to this program is in line 400:

CL$=MID$(D$(A),I,1)

A MID$ function is used to look at each letter in the name. The instruction is set up to take the letters one at a time from left to right. The first item in the parentheses, D$(A), indicates the word chosen from the menu. The next item, I, refers to the current number in the for-next loop and insures that we progress from letter number one to the last letter in the word.

Conveniently, the MID$ function uses this center area to designate the number of spaces in from the left side of the string to start identifying characters. The 1 at the right in the parentheses shows that the function is to use only one letter at a time. Finally, we let this function equal CL$. After this line in the program instructions, CL$ is used to represent the next letter that should be pressed by the user.

Program 1: PET/CBM Version

```
0 CLR:PRINT"{CLEAR}":POKE59468,12
1 POKE144,49:REM ### DISABLE STOP KEY (UPGRA
    DE ROM) ########
110 N=9:DIMD$(N),P$(20)
120 FORI=1TON:READD$:D$(I)=D$:PRINT:PRINTTAB(1
    2)I")  "D$(I):NEXT
199 :
200 REM *** MENU COMMANDS ********************
    **************
210 GOSUB63998:IFA$="@"THENGOSUB600:GOTO0
220 IFA$="{HOME}"THENPRINT"{CLEAR}":GOSUB63997
    :GOTO0
230 A=VAL(A$):L=LEN(D$(A)):IFA>NORA<1THEN210
240 V=V+1:P$(V)=D$(A):REM *** LOAD PRINTOUT LI
    ST ***
299 :
300 REM *** DISPLAY NAME AND GET READY FOR COP
    Y ***********
305 L2=(40-L)/2
310 PRINT"{CLEAR}{07 DOWN}":GOSUB380:PRINTD$(A
    ):PRINT"{04 DOWN}":GOSUB380
320 GOTO400
380 FORI=1TOL2:PRINT"{RIGHT}";:NEXT:RETURN:REM
     *** MOVE CURSOR TO CENTERING POSITION
399 :
400 REM *** ACCEPT ONLY CORRECT RESPONSES ****
    **************
410 FORI=1TOL:CL$=MID$(D$(A),I,1)
420 GOSUB63998:IFA$=CL$THENPRINTCL$;:GOTO440
430 GOTO420
440 NEXT
499 :
500 REM *** RETURN TO MENU ******************
    **************
510 GOSUB63998:PRINT"{CLEAR}":RESTORE:GOTO120
599 :
600 REM *** PRINTOUT THE NAMES ***************
    **************
610 OPEN4,4:PRINT#4:FORI=1TOV:PRINT#4:PRINT#4,
    P$(I):NEXT:PRINT#4:CLOSE4:RETURN
699 :
```

```
1000 REM *** DATA LIST OF NAMES ***************
     *************
1010 DATAMOM
1020 DATAPRETZEL
1030 DATADAD
1040 DATAMELISSA
1050 DATABETH
1060 DATAGRANDMA
1070 DATAAUNT DENISE
1080 DATAGRANDPA
1090 DATAGRANDMA SULLIVAN
63995 :
63996 REM *** WAIT & GET SUBROUTINE ************
     ************
63997 POKE144,46:REM ### ENABLE STOP KEY (UPGRAD
     E ROM) #####
63998 GETA$:IFA$=""THEN63998
63999 RETURN
```

Program 2: Atari Version

```
110 DIM N$(20),P(9)
115 OPEN #1,4,0,"K:"
120 GRAPHICS 2+16:RESTORE
130 FOR I=1 TO 9:SOUND 0,I*20,10,8
140 READ N$:? #6;CHR$(I+176);CHR$(169
    );" ";N$
150 NEXT I:SOUND 0,0,0,0
170 GET #1,A
180 IF A=64 THEN 500
190 IF A=27 THEN GRAPHICS 2+16:GET #1
    ,A:GOTO 120
200 A=A-48:IF A<1 OR A>9 THEN 170
210 FOR I=1 TO V:IF P(I)<>A THEN NEXT
    I:V=V+1:P(V)=A
220 RESTORE
230 FOR I=1 TO A:READ N$:NEXT I
240 GRAPHICS 2+16
250 POSITION 9-LEN(N$)/2,5:? #6;N$
260 FOR I=1 TO LEN(N$)
270 GET #1,A:IF A<>ASC(N$(I)) THEN 27
    0
280 POSITION 9-LEN(N$)/2-1+I,6
290 PUT #6,A+128
295 FOR W=15 TO 0 STEP -1:SOUND 0,A,1
    0,W:NEXT W
300 NEXT I
310 FOR W=1 TO 50:POKE 710,PEEK(53770
    ):SOUND 0,PEEK(53770),10,8:NEXT W
    :SOUND 0,0,0,0
320 GOTO 120
500 REM PRINT OUT
505 TRAP 580
510 GRAPHICS 2+16:? #6;"printing name
    s{3 N}"
520 FOR I=1 TO V
530 RESTORE
540 FOR J=1 TO P(I):READ N$:NEXT J
545 FOR W=15 TO 0 STEP -1:SOUND 0,W,0
    ,W:NEXT W
550 ? #6;N$:LPRINT N$:LPRINT
560 NEXT I
570 RUN
580 GRAPHICS 2+16:? #6;"PRINTER NOT O
    N!"
590 FOR W=1 TO 500:NEXT W:GOTO 120
1000 REM LIST OF NAMES (9)
1010 DATA MOM
1020 DATA PRETZEL
1030 DATA DAD
1040 DATA MELISSA
1050 DATA BETH
1060 DATA GRANDMA
1070 DATA AUNT DENISE
1080 DATA GRANDPA
1090 DATA GRANDMA SULLIVAN
```

©

A Monthly Column

Being language literate is absolutely essential in our society. Being computer literate is a great advantage and is rapidly becoming a necessity. What is being done to meet the need for this new area of education? Getting computers into classrooms across the country is a start, but just a start. There is a great deal more involved.

Learning With Computers

Computer Literacy: Can We Get There From Here?

Mary Humphrey
Teaching tools: Microcomputer Services, Palo Alto, CA

literate / *adj. 1: educated, cultured 2: able to read and write*

Computer literacy is now a common term in education circles, and with it has come a growing demand to develop programs to teach it. For some educators, a new literacy is the chance to open additional avenues of thinking and communicating. For others, the mention of computer literacy causes reactions from deep sighs and "here-we-go-again" looks to near panic. Why the difference?

Language Literacy And Computer Literacy

An analogy between language literacy and computer literacy is often made, and there are many useful similarities. Just as one need not know how to physically make a book, but should be able to create and comprehend a written passage, one need not know how to build a computer, but should be able to successfully use one and to create at least a simple program. This analogy has been the basis for several recent articles proposing definitions or guidelines for computer literacy. The difference in reactions is not due to debate over what it is. The goals of computer literacy, like the goals of language literacy, are valuable skills that can be generally regarded as critical for members of our society.

Those who react to computer literacy with eager anticipation are thinking about the end product; those who dread it are thinking about delivering that product. Here the analogy between language literacy and computer literacy breaks down. Many education departments have been given a mandate to develop definitions of computer literacy, establish criteria for teacher certification, and begin pre-service and in-service teacher training programs.

Shortly thereafter, school districts and local boards are expected to create and implement student curricula. For these administrators and teachers, the concerns are not "Where are we going?" but rather "How are we going to get there?" For them, the differences between language literacy and computer literacy are glaring.

Becoming A Computer Teacher

Reading and writing competency criteria, instructional programs and standardized tests have been developed over many years with the support of much study. Computer literacy has been pondered for only a relatively short period of time. There has been little opportunity to test any of the guidelines offered, and many authors on the subject encourage educators to develop their own definitions.

Reading and language arts teachers have themselves received many years of training in these skills and in how to teach them. The criteria for teacher certification are quite explicit. Teachers charged with computer education have typically had little computer training and even less instruction in how to teach computer skills. Becoming a computer instructor is often more a matter of personal interest and initiative than of formal qualifications.

Support materials for teaching reading and language arts are big business. Teachers are accustomed to readily available, high quality textbooks, films and slides, classroom display materials, worksheets, and student activity kits for reading and writing. Currently there are few computer literacy materials. Publishers and software developers have had time to produce only a first generation of computer literacy materials, and as yet have had little feedback from educators.

The role of home-based education is also quite different for these two types of literacy. There have always been some parents who have actively encouraged their children to learn to read and write, but reading or writing together as a family activity is usually limited to bedtime stories and thank you letters to Grandma.

Those parents who have personal computers at home seldom have to coax their children, no matter what age, to use the computer. There is a great deal of commercial promotion of various uses of computers as family activities. Teachers are realizing that this considerable amount of home learning is a welcome change, but also a challenge to the schools.

Added to these differences are two common misconceptions about computer literacy. First, it's a new and often unfamiliar area to many educators. Unfamiliarity can be confused with difficulty. This has been especially true of computing. The stereotype of high technology as a scientist's domain still lingers, despite the current efforts to promote personal and home computers as "user friendly." Because many educators have not been given adequate training in computer skills, they suspect that this new curriculum area may be beyond the capabilities of the schools, particularly the elementary schools.

There are also many educators who are confident computer-users, but who fall prey to a different intimidation. They are aware of the potential of computers in education and the amazing pace at which new developments are occurring. For them the implementation of a computer literacy program is a scramble to get it all done within the current school year. The pressure to catch up to the needs of business and industry for computer literate graduates can seem overwhelming if viewed from this perspective.

Getting There From Here

At this point it all sounds very discouraging, and you may be wondering whether schools can overcome these obstacles and go on to develop a new curriculum. There is lack of teacher training, lack of support materials, and pressure from outside the school. Do schools even want to try? The best answers to these questions come from the schools' own reports.

During the 1981-82 school year, many districts and local school boards began computer literacy programs. This year they were joined by more schools, and still others are laying the groundwork for programs in the 1983-84 school year. Several school districts, computer-education groups, and even individual teachers have written reports on their own computer literacy programs. Their enthusiasm is clear. The strongest encouragement can be found in evaluations of existing programs.

These "how-to" accounts are sincere attempts to help others through the first steps of implementing a computer literacy curriculum. Many are available for the asking or for a minimal charge to cover costs. I strongly recommend that those involved with a computer literacy project get these materials.

Several reports are of interest for those who are beginning a computer literacy program. They are particularly helpful in dealing with the difficulties of establishing a program of teacher training. "Instructional Uses of Microcomputers: A Report on British Columbia's Pilot Project" (research conducted by JEM Research) describes the planning and implementing stages, the training and other services provided, and a complete evaluation of the impact of these services. This report is also useful as a guide to planning for future evaluation. Requests for copies of the report should be sent to: Project Planning Centre, Ministry of Education, Legislative Building, Victoria, British Columbia, Canada V8V-1X4.

Computers in the Classroom is another especially thorough guide. This "booklet explaining the process of implementing computers into the elementary classroom" is written by Susan Burleson, an assistant principal in the San Ramon Valley Unified School District. It is a step-by-step account of what this district did and did not do and their recommendations to others.

Chapters cover setting goals in a district, identifying resources, computer awareness and readiness for in-service training, obtaining funds and budgeting, in-service training, school-wide use and home use of computers, anticipating problems, and evaluating progress. Copies cost $11 and are available from Susan Burleson, 599 Bridgewater Rd., Danville, CA 94526.

An energetic group of teachers and specialists in Utah is developing a kindergarten through high school computer curriculum. Their project provides plans for a three-year development cycle to train teachers, begin limited field testing, and then conduct a formal field test of several pilot projects. Curriculum objectives, teaching activities, information and materials resources, and evaluation criteria are detailed for each grade level across several "strands" of computer skills.

Other materials include such specific help as a principal's checklist for interviewing computer hardware dealers (a useful document for dealers too). Inquiries about cost and availability of part or all of their materials should be sent to: Curriculum Development Office, Jordan School District, 9361 South 400 East, Sandy, UT 84070.

A lighthearted but quite useful guide to over-coming commonly encountered problems is "The Mother Quail Syndrome: Managing Micros on Site. 10 Sanity Savers for Educators." Write to Suzanne Powers-Bailey, Computer Coordinator, Solano County Office of Education, 655 Washington St., Fairfield, CA 94533, for information about cost and availability.

Developing A Curriculum

In addition to issues of teacher training, I discussed the need to develop a student curriculum and the lack of supporting teaching materials as difficulties in teaching computer literacy. Again, the response from those with experience is encouraging and enthusiastic. Many groups have committed a great deal of time and effort to developing curriculum guides complete with resource lists and bibliographies, tables of computing topics and their objectives, and descriptions of classroom activities and necessary materials.

An excellent example is the CLAS (Computer Literacy and Awareness for Students) package developed by the TRI-County Computer Consortium of Southeastern Michigan. Macomb County Intermediate School District, Oakland Schools, and Wayne City Intermediate School District combined efforts to produce a comprehensive and detailed computer curriculum. The cost is $10. Write to Tom Hartsig, Macomb County School District, 44001 Garfield Rd., Mt. Clemens, MI 48044.

The "home-made" materials developed by teachers and school groups may not have the glossy, typeset appearance of professionally produced materials, but they are carefully constructed and genuinely useful teaching aids. There aren't enough of them. Educators still have to search them out, and the schools are not prepared for mass distribution, but they are invaluable models. Publishers and software developers will also find them useful guides.

A quick look at the resources I've mentioned here is enough to demonstrate how much interest and effort is being generated. Schools are putting more into computer literacy than just computers. Even those who sigh or panic at the mention of computer literacy can see evidence of the payoff. It may be sooner than we think that we will be able to spend less time accomplishing computer literacy and more time enjoying the benefits of its new avenues of thinking and communicating. ©

A Monthly Column

Friends Of The Turtle

David D. Thornburg
Associate Editor

Recursion – Part 2

Last time, we explored recursion as a powerful programming tool. The basic elements of a recursive procedure include:

> **1.** A conditional statement to tell when to stop the recursive process;
> **2.** A series of commands to be executed at each recursive level; and,
> **3.** The use of the procedure itself with, perhaps, new values for the procedure's variables.

The sequence and intermixing of these elements determine the type of recursive process being followed. Recursion can range from simple looping to the more complex forms we used for drawing fractals.

Because of the obvious visual relationship between certain fractals and the recursive procedures that generate them, we will examine some more of these this month.

Before doing that, however, let's make a small digression to examine the difference between the conditional branching commands commonly used with Logo programs for the Apple computer and the conditional branching command used by TI Logo.

The structure of the command we have been using is:

```
IF predicate instructionlist
```

This means that the structure of the command is the word IF followed by an operation whose result is either true or false (the predicate), followed by a list of instructions to be executed if the predicate is true. An alternate form of this command is:

```
IF predicate THEN instructionlist
```

This form of the command is common to most BASICs as well, and might be familiar to many of you.

TI Logo uses a different type of conditional command, one which is more reminiscent of PILOT. In TI Logo the IF ... THEN ... construction is replaced by:

```
TEST predicate
IFT instructionlist1
```

and also

```
IFF instructionlist2
```

This construction allows you to test a predicate in a line all by itself, and to then execute certain instructions selectively, based on the result of the test, anywhere after the TEST command. The command IFT will execute instructionlist if the result of the test was true, and the command IFF will execute the list if the result was false.

In Apple Logo our conditional command in the fractal procedure is:

```
IF :SIZE < :LIMIT [FORWARD :SIZE STOP]
```

In TI LOGO this would be replaced by:

```
TEST :SIZE < :LIMIT
IFT FORWARD :SIZE STOP
```

One other note for TI Logo users: you may find that your turtle's pen "runs out of ink" on the more complex curves. You might want to try drawing smaller versions of them to minimize this problem. Of course, you should be sure to clear the screen before drawing anything, just to be sure you have recovered as much "ink" as possible.

And now, on with the show!

One type of fractal that generates pretty pictures is the Koch curve we drew last time. In its most general form, we can define the motif for this type of curve as starting with a horizontal line, making some construction using line segments of the same length, and ending with a horizontal line on the same level as the first one. The following three fractals are particularly pleasing to me and to the people who have seen them exhibited at shows, so I am pleased to also share them with you. As in the past, all procedures will be shown in Apple Logo, and you can easily translate these to any other version of the language you might be using.

Before creating the curves, we will define a general setup procedure that puts the turtle in the correct starting position and orientation for each curve:

Apple* ★ Atari* ★ Commodore* Franklin* ★ IBM* ★ TRS-80*

Krell's College Board SAT* Preparation Series

New for '83

A COMPREHENSIVE PREPARATION PACKAGE / MORE THAN 40 PROGRAMS

1. Diagnostic analysis
2. Prescription of individual study plans
3. Coverage of all SAT* skills
4. Unlimited drill and practice
5. SAT* Exam Question simulator
6. All questions in SAT* format and at SAT* difficulty level
7. Instantaneous answers, explanations and scoring for problems
8. Worksheet generation and performance monitoring - (optional)
9. A complete record management system- (optional)
10. Systematic instruction in pertinent math, verbal & test taking skills - (optional)

Krell's unique logical design provides personalized instruction for each student according to individual needs.

Krell's College Board SAT* 81/82 Prep. Series has demonstrated a mean combined math and verbal score increase of more than 100 points by using sophisticated drill & practice techniques alone. This new series is much expanded and updated. In addition extra cost options are now available to provide worksheet generation, record management, and systematic instruction in all math and verbal areas tested by the SAT*.

All versions of Krell College Board SAT* Preparation Series provide answers, explanations and instantaneous raw scoring. Standard package $299.95

APPLE, ATARI, COMMODORE, CP/M, IBM, AND RADIO SHACK*

SWORD OF ZEDEK

Fight to overthrow Ra, The Master of Evil. Treachery. deceit, and witchcraft must be faced in your struggles as you encounter wolves, dwarves, elves, dragons, etc. Each of the twelve treasures will enhance your power by making you invisible, invulnerable, etc. Each game is unique in this spectacular and complex world of fantasy. $24.95

LOGO

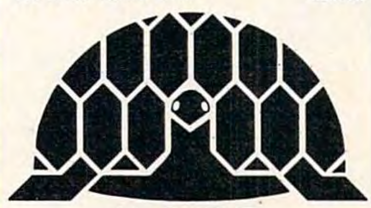
©M.I.T. 1981

$99 NO FRILLS for Apple II®

NO FRILLS PAK

1. Two copies of Krells LOGO for Apple II*
2. Utility Disk with M.I.T.'s valuable demo programs including Dynatrack
3. The official M.I.T technical manual LOGO FOR APPLE II by H. Abelson & L. Klotz

No Frills Turtle Price $99.00



LOGO FRILLS ONLY

FRILLS FOR LOGO / Support Pak for: M.I.T. LOGO, KRELL & TERRAPIN INC.

1. Krell utility disk
2. Alice in LOGOLAND
3. LOGO for Apple II - by H. Abelson
4. Alice in Logoland Primer
5. Comprehensive wall chart
6. LOGO & Educational Computing Journal

$89.95

NO FRILLS LOGO AND ALL THE FRILLS COMBO FOR APPLE II* $149.95

SPRITES NOW AVAILABLE FOR LOGO Call for information

Prices slightly higher outside U.S.

TIME TRAVELER

Using the Time Machine, players must face a challenging series of environments that include: The Athens of Pericles, Imperial Rome, Nebuchadnezzar's Babylon, Ikhnaton's Egypt, Jerusalem at the time of the crucifixion, The Crusades, Machiavelli's Italy, the French Revolution, the American Revolution, and the English Civil War. Deal with Hitler's Third Reich, Vikings, etc. Players may set level of difficulty. Each game is unique. $24.95

Shelby Lyman Chess*

Shelby's Socrates Chess Tutorial Series uses the latest AI techniques to customize each lesson for you. Instruction modules cover every aspect of the game. For all micro's. Call or write for details.

Amazing Ben The Royal Road to Artificial Intelligence. $79.95

Connections

A complete game system. Learn the principles of scientific reasoning in your choice of game formats. Expand minds at all ages. Subject areas include: Geography, Biology, Everyday Objects, etc.

Game System $99.95

Data Base, $24.95 per subject, 3 for $50

Isaac Newton + F.G. Newton

Perhaps the most fascinating and valuable educational game ever devised - ISAAC NEWTON challenges the players to assemble evidence and discern the underlying "Laws of Nature". FULL GRAPHICS NEWTON - presents all data in graphic form. This game is suitable for children. Players may select difficulty levels challenging to the most skilled adults. Both Games $49.95

Pythagoras and the Dragon

Mathematics in a fantasy game context. Based on THE SWORD OF ZEDEK introduces Pythagoras as a mentor to the player when called on for aid, Pythagoras poses math questions and depending on the speed and accuracy of response, confers secret information. With Pythagoras as an ally, the quest to overthrow Ra, The Master of Evil, assumes a new dimension of complexity. Depending on the level chosen problems range from arithmetic through plane geometry. APPLE, TRS-80, PET, ATARI* 32K $39.95

KRELL® SOFTWARE CORP.

The state of the art in educational computing

1320 Stony Brook Road / Stony Brook NY 11790
Telephone 516-751-5139

Krell Software Corp. has no official ties with the College Entrance Examination Board or the Educational Testing Service. Krell is, however, a supplier of products to the E.T.S.

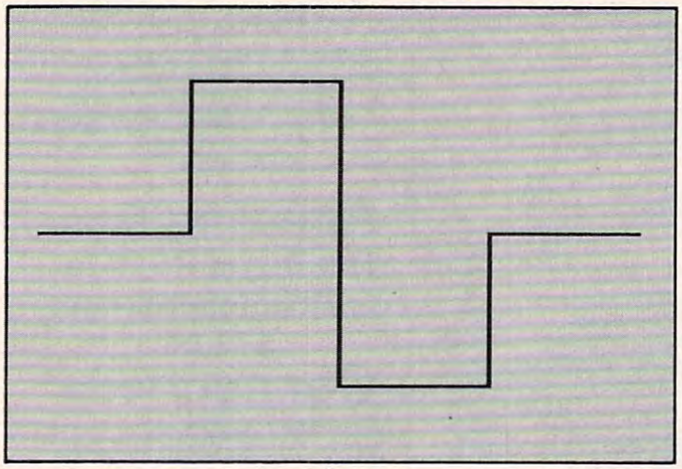
*Trademarks of Apple Comp. Corp., Tandy Corp., Commodore Corp., Digital Research Corp., I.B.M., Atari Corp., Terrapin Inc., Socrates Chess Corp.

VISA

N.Y.S. residents add sales tax

```
TO SETUP :LIST
PENUP
SETPOS :LIST
SETHEADING 90
PENDOWN
END
```

The first curve we will explore is a square meander.



The procedure for creating fractals based on this figure is the following:

```
TO MEANDER :SIZE :LIMIT
IF :SIZE < :LIMIT [FORWARD :SIZE STOP]
MEANDER :SIZE / 4 :LIMIT
LEFT 90
MEANDER :SIZE / 4 :LIMIT
RIGHT 90
MEANDER :SIZE / 4 :LIMIT
RIGHT 90
REPEAT 2 [MEANDER :SIZE / 4 :LIMIT]
LEFT 90
MEANDER :SIZE / 4 :LIMIT
LEFT 90
MEANDER :SIZE / 4 :LIMIT
RIGHT 90
MEANDER :SIZE / 4 :LIMIT
END
```

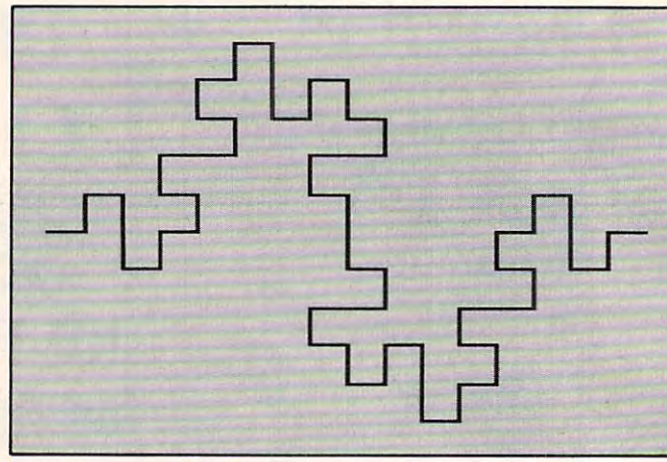
Before using this procedure, let's examine it. The first thing to notice is that the value of SIZE is reduced by a factor of four for each successive use of the procedure. The reason for this is that the total horizontal extent of the original motif is four times the length of the line segment. The second thing to notice is that the double length of line in the motif is created by a double repetition of the procedure. To see the motif, enter:

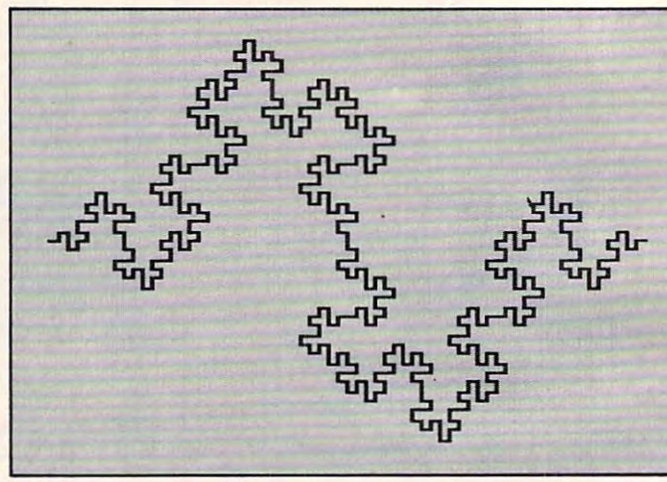
```
CLEARSCREEN
SETUP [-128 0]
MEANDER 256 256
```

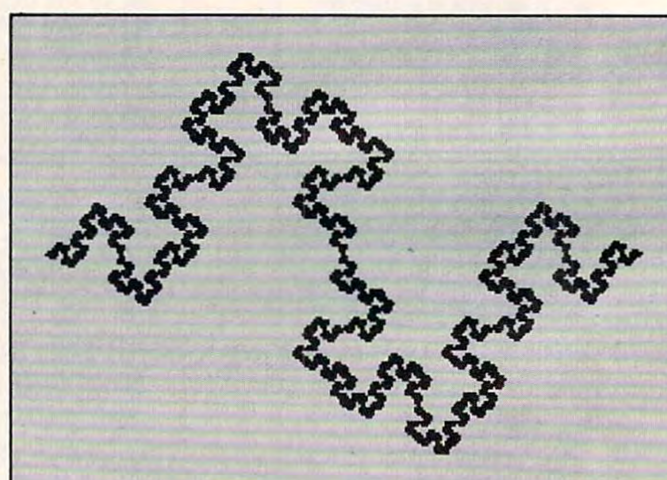
Successive generations can be seen by entering:

```
MEANDER 256 64
MEANDER 256 16
MEANDER 256 4
```

(Remember to clear the screen and use the SETUP procedure before drawing each curve.)

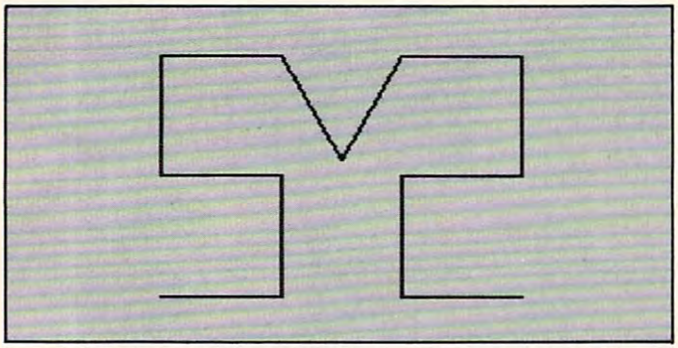






As you look at each successive generation of this figure, it is interesting to note the development of secondary meanders resulting in a final highly convoluted (but strangely symmetrical) form.

The second curve I want to share is called the T-shirt fractal, since it was designed for use on a T-shirt (write me at Friends of the Turtle for details). In making this design, I thought that a fractal T-shirt should use a T-shirt fractal, thus carrying the recursive process one step backwards to the overall shirt itself. The motif I designed looks like this:



The fractal procedure based on this motif is given by:

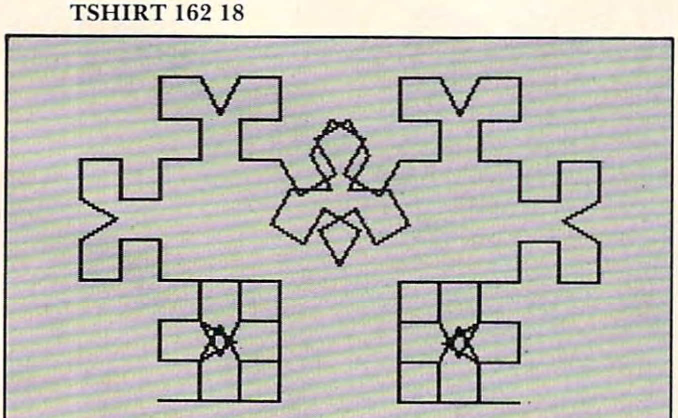
```
TO TSHIRT :SIZE :LIMIT
IF :SIZE < :LIMIT [FORWARD :SIZE STOP]
TSHIRT :SIZE / 3 :LIMIT
LEFT 90
TSHIRT :SIZE / 3 :LIMIT
LEFT 90
TSHIRT :SIZE / 3 :LIMIT
RIGHT 90
TSHIRT :SIZE / 3 :LIMIT
RIGHT 90
TSHIRT :SIZE / 3 :LIMIT
RIGHT 60
TSHIRT :SIZE / 3 :LIMIT
LEFT 120
TSHIRT :SIZE / 3 :LIMIT
RIGHT 60
TSHIRT :SIZE / 3 :LIMIT
RIGHT 90
TSHIRT :SIZE / 3 :LIMIT
RIGHT 90
TSHIRT :SIZE / 3 :LIMIT
LEFT 90
TSHIRT :SIZE / 3 :LIMIT
LEFT 90
TSHIRT :SIZE / 3 :LIMIT
END
```

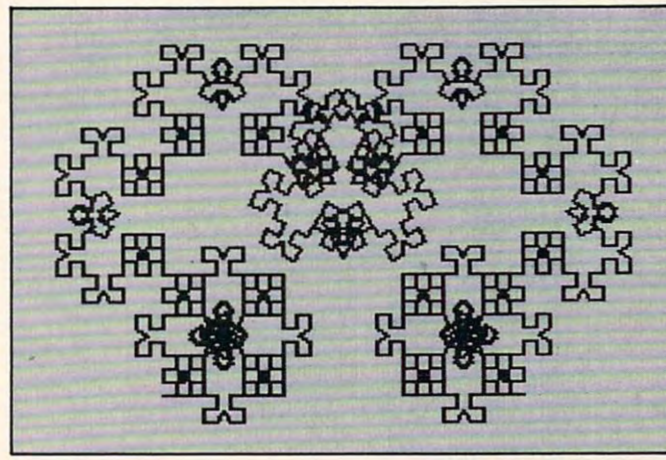
To generate the motif on the display, enter:

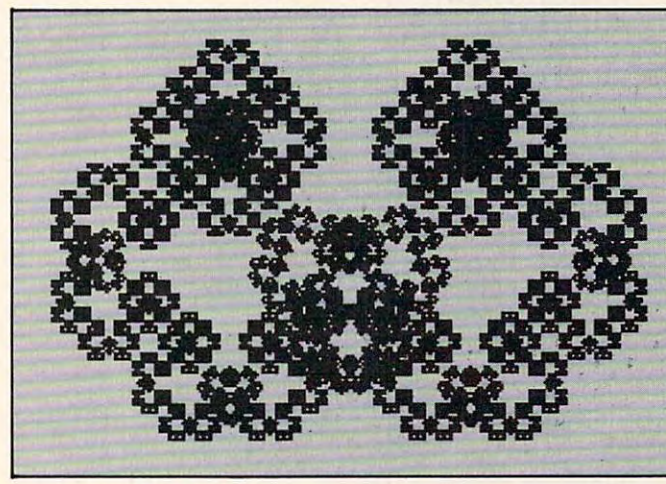
```
CLEARSCREEN
SETUP [-81 -60]
TSHIRT 162 162
```

Successive generations can be formed with the following commands:

```
TSHIRT 162 54
TSHIRT 162 18
```



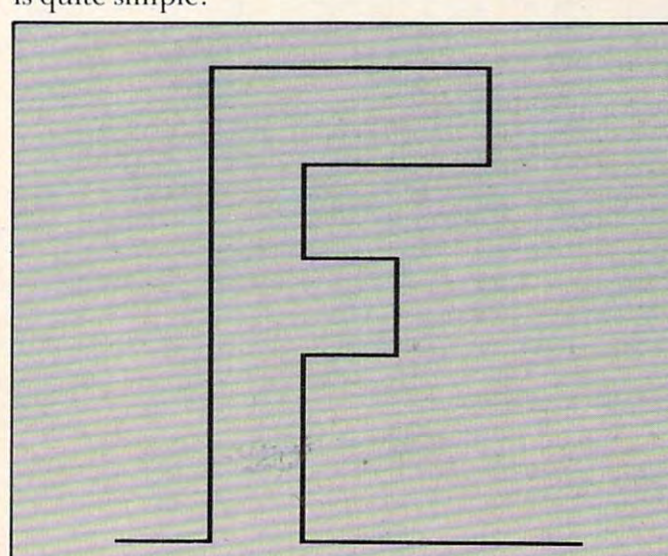




TSHIRT 162 6

Notice that, for this pattern, there is a lot of overlapping in successive generations that makes it harder to identify the original motif. But, if you look closely, you will be able to see the motif hidden (in full size) in each generation.

The last pattern I wanted to show is from a piece of artwork entitled *F is for Fractal.* The motif is quite simple:



The procedure for this curve is a bit on the lengthy side:

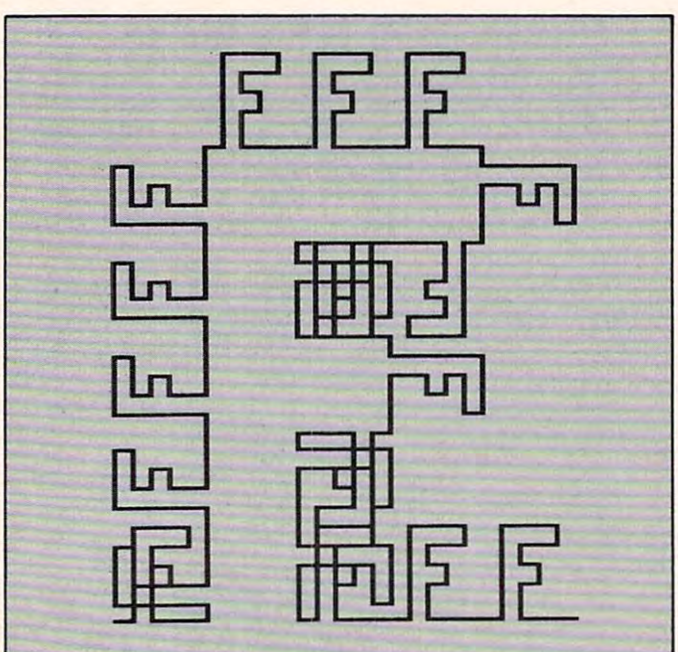
```
TO F :SIZE :LIMIT
IF :SIZE < :LIMIT [FORWARD :SIZE STOP]
F :SIZE / 5 :LIMIT
LEFT 90
REPEAT 5 [F :SIZE / 5 :LIMIT]
RIGHT 90
REPEAT 3 [F :SIZE / 5 :LIMIT]
RIGHT 90
F :SIZE / 5 :LIMIT
RIGHT 90
REPEAT 2 [F :SIZE / 5 :LIMIT]
LEFT 90
F :SIZE / 5 :LIMIT
LEFT 90
F :SIZE / 5 :LIMIT
RIGHT 90
F :SIZE / 5 :LIMIT
RIGHT 90
F :SIZE / 5 :LIMIT
LEFT 90
REPEAT 2 [IF :SIZE / 5 :LIMIT]
LEFT 90
REPEAT 3 [F :SIZE / 5 :LIMIT]
END
```

The motif can be generated by entering:

```
CLEARSCREEN
SETUP [-85 -110]
F 175 175
```

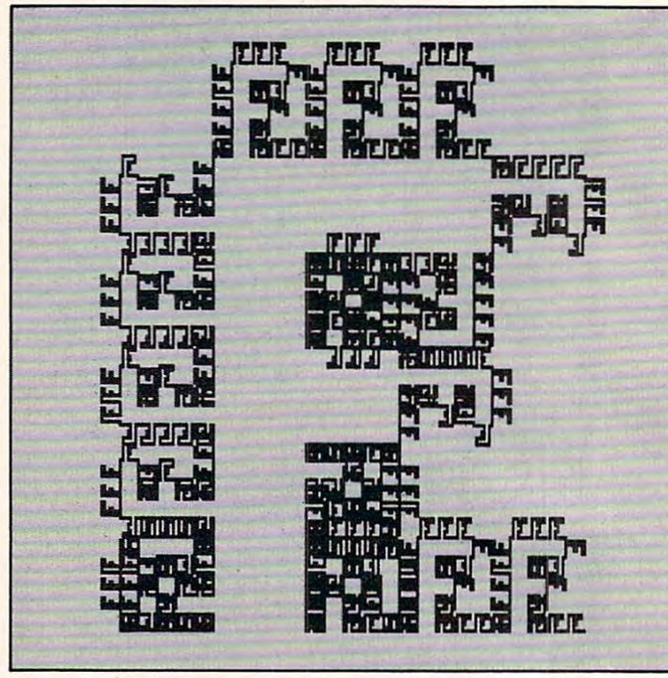
Further generations are created with the commands:

```
F 175 35
F 175 7
```



What I find particularly interesting is the manner in which the figure of the F in the motif becomes the background in the third generation.

By now, you probably have recursive programming firmly under control. You should continue to experiment on your own. The results may surprise you with their beauty!



Calling All Atari PILOTs

COMPUTE! reader Elliot Maggin sent me a delightful extension of a fractal program we described some months back. His program generates King Tut's Headdress. I think you will like the result.

```
2 R:*****************
3 R:*               *
4 R:*   90-DEGREE   *
5 R:*               *
6 R:*    FRACTAL    *
7 R:*               *
8 R:*****************
10 GR:PEN RED
20 GR:CLEAR
30 C:#A=54
40 GR:GOTO -79,-31
50 GR:TURNTO 90
60 U:*FO
70 GR:PEN BLUE
80 GR:GOTO -24,-32;TURN -90;FILL #A
90 GR:PEN RED
100 C:#A=#A/3
110 GR:GOTO -79,-31
120 GR:TURNTO 90
130 U:*F1
140 C:#A=#A/3
150 GR:GOTO -79,-31
160 GR:TURNTO 90
170 U:*F2
180 C:#A=#A/3
190 GR:GOTO -79,-33
200 GR:TURNTO 90
```

```
210 GR:PEN YELLOW
220 U:*F3
230 T:         KING TUT'S HEADDRESS
240 E:
250 *FO
260 GR:DRAW #A
270 GR:TURN -90
280 GR:DRAW #A
290 GR:TURN 90
300 GR:DRAW #A
310 GR:TURN 90
320 GR:DRAW #A
330 GR:TURN -90
340 GR:DRAW #A
350 E:
360 *F1
370 U:*FO
380 GR:TURN -90
390 U:*FO
400 GR:TURN 90
410 U:*FO
420 GR:TURN 90
430 U:*FO
440 GR:TURN -90
450 U:*FO
460 E:
470 *F2
480 U:*F1
490 GR:TURN -90
500 U:*F1
510 GR:TURN 90
520 U:*F1
530 GR:TURN 90
540 U:*F1
550 GR:TURN -90
560 U:*F1
570 E:
580 *F3
590 U:*F2
600 GR:TURN -90
610 U:*F2
620 GR:TURN 90
630 U:*F2
640 GR:TURN 90
650 U:*F2
660 GR:TURN -90
670 U:*F2
680 E:
```

A Year-end Note To All

Before leaving this year behind, I thought you should know some of the things we have in store for you in 1983. First, I have received the Turtle Graphics package for the VIC designed and manufactured by HES, and will report on it in January. Also, I am now using the Radio Shack Color Logo package and will be reporting on it in the same issue. Those of you who are interested in fractals may be interested in *The Fractal Geometry of Nature*, a new book by the father of this study, Benoit Mandelbrot. I will be reviewing this book and commenting on the controversy in this field in a forthcoming "Computers and Society" column.

In the meantime, let me know what *you* want to read, and I'll see what I can do to meet your needs.

Friends of the Turtle
P.O. Box 1317
Los Altos, CA 94022

©

This Commodore version of the language concludes the series on PILOT which began four issues ago and included Apple and Atari versions. This program needs at least 8K memory and works on tape or disk-based systems.

VIC And PET PILOT Interpreter

Michael Tinglof
Merrimack, NH

PILOT is an acronym for Programmed Instruction, Learning, or Teaching. Because it is a simple language, teachers can easily develop lesson programs, and beginning students can quickly learn how to program.

This version of PILOT contains all of the core commands used for displaying information and accepting responses. It also has some mathematical capabilities.

The interpreter is written in BASIC so that it is transportable between machines. There is, however, one machine language routine called by line 3 and loaded by the following statement in line 20:

```
20 ....:FORX=826 TO 831:READ Z:POKEX,Z:
   NEXT:......
```

The routine can be loaded anywhere to suit your system needs by simply changing the 826 and 831 values. For the VIC, I would suggest changing the values to 820 and 825. Don't forget to change the SYS call in line 3 if you change the above values.

For computers other than Commodore, the routine must be replaced by an input routine which will accept colons and commas.

The next section describes the editor, the commands, and the implemented PILOT statements.

The Editor

The editor behaves just like the BASIC editor. To enter a line, type the line number, the PILOT statement, and hit RETURN. Any statement entered without a line number is assumed to be a command (see Commands) and is executed as such.

The screen editor is fully active during program entry. To correct an error in a statement or command, just move the cursor to it and enter the correction. Remember, the RETURN key must be pressed for it to be changed in memory.

When the editor is storing a PILOT program line in memory, it first removes the PILOT command and tokenizes it. Thus, if an illegal command is used, an error message will be generated before the program is run.

Commands

The following describes the editor's commands.

LIST xx-yy – Lists the specified lines from memory. xx, yy, or both can be removed.
RUN – Executes the PILOT program currently in memory.
SAVE 0:name – Saves the program in memory to disk on drive 0. No quotes are necessary.
LOAD name – Loads the program from disk. No quotes are necessary.
NEW – Clears the current program from memory.
BASIC – Exits the interpreter and returns to BASIC.
PLIST xx-yy – Same as the list command, except the output is sent to device 4.

PILOT Variables And Statements

PILOT variables consist of either a "$" for a string variable or a "#" for a numeric variable, followed by a single letter. For example, #N and $S are correct, whereas $NAME is not.

The PILOT statements implemented are:

T: Type
Outputs text and variables to the screen. For example:

```
1 T: VALUE #X
```

will type "VALUE xx".

If the statement is ended by a ";" no carriage return will be printed.

J: and U: Jump and Use
Transfer program execution to the specified routine. In the case of Use, the current line number is stored so it can be returned to (see End). For example:

```
2 J:*PRINT
```

jumps to the routine labeled PRINT. Labels are designated by beginning a line with an "*" sign. No statement should follow this label on the same line.

E: End
Transfers control back to the statement following the last Use statement executed.

M: Match
Match is the most complicated and powerful of the PILOT commands. It checks to see if certain keywords are present in a string variable or in the input buffer (see Accept). For example:

THE LEARNING CENTER

"Perhaps the finest educational software that I have ever had the pleasure of reviewing. It's easy for kids to use and effective in teaching basic concepts and skills.... My kids are learning with it right now!"

Fred D'Ignazio, Associate Editor—*Compute!*, Associate Editor—*Softside*,
Author of bestseller—*Katie and the Computer*

We hope you've been using your **ATARI** for more than just games ... it is, and can be, a valuable educational tool for you and your children.

Bruce and Dianne Mitchell realized this potential and designed a series of programs for use in their Small World Preschool & Kindergarten located in Durham, N.C. Presented on TV's *PM Magazine* these unique educational programs will introduce your 3-9 year olds to the era of learning with computers. Using the graphics and sound capabilities of the Atari, each program develops a particular skill and reinforces correct responses with happy faces and music.

Beginning with basic concepts such as colors, shapes and the alphabet, your child will progress to an understanding of counting, arithmetic, and language skills.

Widely acclaimed, classroom designed and tested, these unique educational tools are now available to you ... *for your children.*

SPECIAL SKILLS	Preschool	K	1	2
* Color For The Non-Reader	●	●		
* Name That Color		●	●	
* Like Shape Identification	●	●	●	
* Different Shape Identification	●	●	●	
Cave Game	●	●	●	●
MATH AND NUMBER SKILLS				
Count With Me	●	●	●	
Number Recognition	●	●	●	●
Addition		●	●	
Subtraction		●	●	
Add.—Vertical/Horizontal		●	●	
Sub.—Vertical/Horizontal		●	●	
Advanced Addition/Subtraction		●	●	●
Ones and Tens			●	●
LANGUAGE SKILLS				
* Alphabet Recognition			●	●
* Letter Sequence	●	●	●	
* Like Symbol Discrimination	●	●		
* Different Symbol Discrimination	●	●		

Pricing Information	Cass.	Disk		Cass.	Disk
Single Program	6.95	9.95	Pre-School	36.95	41.95
Special Skills	24.95	29.95	Kindergarten	54.95	59.95
Language Skills	19.95	24.95	First Grade	49.95	54.95
Math and Number Skills	34.95	39.95	Second Grade	21.95	26.95
Complete Set	59.95	64.95			
Edumate™ Light Pen	19.95		Add $2.00 for postage & handling		

* Compatible with our Edumate™ Light Pen
NOTE: All software requires 8K cassette/16K disk

Start competing with your 4-year-old for computer time. Enroll in **THE LEARNING CENTER**, it could be the best investment you ever make ... *in your children's future.* See your local dealer or order direct.

Available now for the
ATARI 400/800
COMING SOON for the
TRS-80 Model I, III, Color
Apple, VIC, TI-99
Free Catalog Upon Request

THE PROGRAMMER'S INSTITUTE
—a division of **FUTUREHOUSE**™
P.O. Box 3191 Dept C
Chapel Hill, NC 27514

1-919-967-0861 10 am - 9 pm, Monday - Saturday

10 M:YES,OK,ALRIGHT

checks to see if YES, OK, or ALRIGHT are present in the input buffer. To check a string for keywords:

15 M:$n YES,OK,...

If a match is found the Y flag is set; otherwise the N flag is set (see Modifiers).

I: If

If is a nonstandard command which allows for mathematical testing. It can check to see if a given variable is less than, greater than, or equal to a second given value or variable. For example:

20 I:#N<9

or

25 I:#C=#F

Only >,<, and = can be used.

C: Compute

Performs simple four-function calculations in a linear order (no parentheses) and assigns the value to a numeric variable. The calculations are performed in floating point mode so reasonable accuracy can be expected. For example:

30 C:#N=#G*10/#T+10

If a "#R" is encountered in the expression, a random number between 0 and 1 is substituted.

A: Accept

Inputs a response from the user. If no destination variable is given, the response is stored in a buffer which can be used by Match. For example:

40 A:#N inputs a value into N
41 A: inputs a response into the buffer

H: Home

Clears the screen and returns the cursor to home.

End

Stops the program execution and returns to the editor. This statement cannot be modified by a "Y" or "N". For example:

50 END

Modifiers

Any of the commands can be modified with either a "Y" or "N." If a command is modified, it will be executed only if the specified flag is set. For example:

1 TY:YES

will print YES only if the Y flag is set. The Y and N flags are set by either a Match or If statement.

Error Messages

The following are the error codes generated during program run:

1 – Illegal variable name
2 – Unknown label
3 – Stack overflow (too many Uses)
4 – Stack empty (an E: with no Use)
5 – Bad format
6 – Division by zero
7 – Numeric out of range (greater than 32767)

Notes On Program Operation

1. To stop a PILOT program run, hit the "@" key. To stop a list, hit any key.

2. If for some reason the program returns to BASIC level, just type GOTO 40 <RETURN> to re-enter without losing the current program.

3. If a NEW statement is not given before loading a new program, the current program and the new program will be merged in memory.

4. The maximum number of lines allowed is contained in the variable M and is set in line 10. This can be changed.

5. For cassette operation, make the following changes:

500 OPEN1,1,1,R$:PRINT"SAVING"R$
600 OPEN1,1,0,R$:PRINT"LOADING"R$

6. This interpreter is about 3K bytes long, and about 4K bytes are taken after system initialization. This still leaves 3K on an 8K PET!

This program gives the user access to a fairly complete set of PILOT commands, while at the same time leaving enough space for program development even on an 8K PET.

```
1 GOTO10:REM***PILOT***
2 I$=""
3 SYS826:IFPEEK(0)=13THENRETURN
4 I$=I$+CHR$(PEEK(0)):GOTO3
10 CLR:M=200:X=0:Y=0:A=0:P=0:Z=0:I$="":DIMS%(
    9),N%(26),S$(26),L$(M),C$(15):F%=0
20 PRINT"{CLEAR}**** PILOT V2.1 ****":FORX=82
    6TO831:READZ:POKEX,Z:NEXT:FORX=0TO15
25 READC$(X):NEXT:DATA32,207,255,133,0,96
30 DATALIST,RUN,SAVE,LOAD,NEW,BASIC,PLIST,T,J
    ,E,U,M,C,A,I,H
40 PRINT"{DOWN}PILOT."
50 GOSUB2:PRINT:IFASC(I$)=32ANDLEN(I$)=1THENG
    OTO50
60 IFLEFT$(I$,1)=" "THENI$=MID$(I$,2):GOTO60
70 L=VAL(I$):IFL<>0THENGOTO200
80 L=1:H=M:R$="":FORX=1TOLEN(I$):IFMID$(I$,X,
    1)<>" "THENNEXTX:GOTO140
90 R$=MID$(I$,X+1):I$=LEFT$(I$,X-1)
100 L=VAL(R$):H=L:FORX=1TOLEN(R$):IFMID$(R$,X,
    1)<>"-"THENNEXT:GOTO120
110 L=VAL(LEFT$(R$,X-1)):H=VAL(MID$(R$,X+1))
120 IFL=0THENL=1
130 IFH=0THENH=M
140 FORX=0TO6:IFI$<>LEFT$(C$(X),LEN(I$))THENNE
    XT:PRINT"UNKNOWN COMMAND.":GOTO40
150 ONX+1GOTO400,1000,500,600,700,800,390
200 IFL>MTHENPRINT"LINE NUMBER OUT OF RANGE.":
    GOTO40
210 X=LEN(STR$(L)):X$=MID$(I$,X):IFX$=""THENL$
    (L)="":GOTO50
220 IFLEFT$(X$,1)=" "THENX$=MID$(X$,2):GOTO220
230 X=3:IFMID$(X$,2,1)<>":"THENX=4:IFMID$(X$,3
    ,1)<>":"THENL$(L)=X$:GOTO50
```

```
240 FORZ=7TO15:IFLEFT$(X$,1)<>C$(Z)THENNEXT:PR
    INT"ILLEGAL COMMAND.":GOTO40
250 IFMID$(X$,2,1)="Y"THENZ=Z+10
260 IFMID$(X$,2,1)="N"THENZ=Z+20
270 L$(L)=CHR$(Z-6)+MID$(X$,X):GOTO50
390 OPEN1,4:GOTO410
400 OPEN1,3
410 FORX=LTOH:IFL$(X)=""THEN450
420 X$=":":Z=ASC(L$(X)):IFZ>30THENX$=LEFT$(L$(
    X),1):GOTO440
425 IFZ>20THENZ=Z-20:X$="N"+X$
430 IFZ>10THENZ=Z-10:X$="Y"+X$
435 X$=C$(Z+6)+X$
440 PRINT#1,X;X$;MID$(L$(X),2)
450 GETX$:IFX$<>""THENCLOSE1:GOTO40
460 NEXT:CLOSE1:GOTO40
500 OPEN1,8,2,R$+",S,W":PRINT"SAVING "R$
510 FORX=1TOM:IFL$(X)=""THEN530
520 PRINT#1,X;CHR$(13)CHR$(34)L$(X)CHR$(34)CHR
    $(13);
530 NEXTX:CLOSE1:GOTO40
600 OPEN1,8,2,R$+",S,R":PRINT"LOADING "R$
610 INPUT#1,X:IFSTGOTO630
620 INPUT#1,L$(X):IFST=0GOTO610
630 CLOSE1:GOTO40
700 GOTO10
800 PRINT"{DOWN}EXITING TO BASIC....":END
1000 L=0:FORX=1TO25:N%(X)=0:S$(X)="":NEXT:P=0:F
    %=0
1010 L=L+1:IFL=>MORL$(L)="END"THEN40
1011 GETX$:IFX$="@"THEN40
1015 IFL$(L)=""THEN1010
1020 X=ASC(L$(L)):IFX>40THEN1010
1030 IFX>20THENX=X-20:IFF%=1THEN1010
1040 IFX>10THENX=X-10:IFF%=0THEN1010
1050 C$=MID$(L$(L),2):ONXGOTO1100,1220,1300,120
    0,1500,1600,1700,1800,1900
1090 PRINT"ERROR #"E"IN LINE"L:GOTO40
1100 Z=0:IFRIGHT$(C$,1)=";"THENZ=1:C$=LEFT$(C$,
    LEN(C$)-1)
1105 FORX=1TOLEN(C$):X$=MID$(C$,X,1):IFX$="#"TH
    EN1150
1110 IFX$="$"THEN1160
1120 PRINTX$;:NEXT:IFZ=0THENPRINT
1130 GOTO1010
1150 GOSUB1190:X$=STR$(N%(Y)):GOTO1120
1160 GOSUB1190:X$=S$(Y):GOTO1120
1190 X=X+1:Y=ASC(MID$(C$,X,1))-64:IFY<1ORY>26TH
    ENE=1:GOTO1090
1195 RETURN
1200 IFP>8THENE=3:GOTO1090
1210 P=P+1:S%(P)=L
1220 IFVAL(C$)<>0THENL=VAL(C$)-1:GOTO1010
1230 FORX=1TOM:IFC$<>L$(X)THENNEXT:E=2:GOTO1090
1240 L=X:GOTO1010
1300 IFP=0THENE=4:GOTO1090
1310 L=S%(P):P=P-1:GOTO1010
1500 X=1:C$=C$+",":X$=AC$:IFLEFT$(C$,1)="$"THEN
    GOSUB1590
1510 FORZ=XTOLEN(C$):IFMID$(C$,Z,1)<>","THENNEX
    T
1520 Z$=MID$(C$,X,Z-X):FORY=1TOLEN(X$):IFMID$(X
    $,Y,LEN(Z$))=Z$THENF%=1:GOTO1010
1560 NEXT:IFZ<LEN(C$)THENX=Z+1:GOTO1510
1570 F%=0:GOTO1010
1590 Y=ASC(MID$(C$,2))-64:IFY<1ORY>26THENE=1:GO
    TO1090
1595 X$=S$(Y):X=4:RETURN
1600 A=3:Z=0:X$="":IFLEFT$(C$,1)<>"#"ORMID$(C$,
    3,1)<>"="THENE=5:GOTO1090
1610 Y=1:X$=MID$(C$,A,1):A=A+1:IFMID$(C$,A,1)="
    -"THENA=A+1:Y=-1
1620 IFMID$(C$,A,1)<>"#"THENY=Y*VAL(MID$(C$,A))
    :A=A+LEN(STR$(Y))-1:GOTO1650
1630 X=ASC(MID$(C$,A+1))-64:IFX<1ORX>26THENE=1:
    GOTO1090
```

```
1635 IFX=18THENY=Y*RND(1):GOTO1650
1640 Y=Y*N%(X):A=A+2
1650 IFX$="="THENZ=Y
1655 IFX$="-"THENZ=Z-Y
1660 IFX$="+"THENZ=Z+Y
1665 IFX$="/"ANDY=0THENE=6:GOTO1090
1670 IFX$="*"THENZ=Z*Y
1675 IFX$="/"THENZ=Z/Y
1680 IFA<=LEN(C$)GOTO1610
1685 Y=N%(X1):IFX-4>0THENIFMID$(C$,X-4)="-"ANDX
    -4<>ATHENZ=-Z
1690 X=ASC(MID$(C$,2))-64:IFX<1ORX>26THENE=1:GO
    TO1090
1692 IFZ>32767ORZ<-32767THENE=7:GOTO1090
1695 N%(X)=Z:GOTO1010
1700 IFC$=""THENGOSUB2:AC$=I$:PRINT:GOTO1010
1720 X=ASC(MID$(C$,2))-64:IFX<1ORX>26THENE=1:GO
    TO1090
1730 GOSUB2:Z=VAL(I$):PRINT:IFLEFT$(C$,1)="#"TH
    ENN%(X)=Z
1740 IFLEFT$(C$,1)="$"THENS$(X)=I$
1750 GOTO1010
1800 IFLEFT$(C$,1)<>"#"THENE=5:GOTO1090
1810 X=ASC(MID$(C$,2))-64:IFX<1ORX>26THENE=1:GO
    TO1090
1820 A=N%(X):X$=MID$(C$,3,1):IFMID$(C$,4,1)<>"#
    "THENX=VAL(MID$(C$,4)):GOTO1840
1830 X=ASC(MID$(C$,5))-64:IFX<1ORX>26THENE=1:GO
    TO1090
1835 X=N%(X)
1840 F%=0:IFX$="<"ANDA<XTHENF%=1
1850 IFX$=">"ANDA>XTHENF%=1
1860 IFX$="="ANDA=XTHENF%=1
1870 GOTO1010
1900 PRINT"{CLEAR}";:GOTO1010
```

20 Questions

```
1 H:
2 T:WELCOME TO THE GAME
  OF TWENTY QUESTIONS.
3 T:BY ASKING QUESTIONS
  WHICH HAVE YES OR
4 T:NO ANSWERS, TRY TO
  GUESS THE OBJECT
5 T:WHICH HAS BEEN SELECTED.
6 T:
7 T:BE SURE TO END EACH
  QUESTION WITH A '?'.
8 T:
9 T:
10 C:#C=0
11 *ROUND
12 C:#C=#C+1
13 *QUESTION
14 T:ENTER QUESTION #C
15 A:
16 M:?
17 TN:THAT ISN'T A QUESTION.
18 JN:*QUESTION
19 M:A?,E?,I?,O?,U?,Y?
20 TY:YES
21 TN:NO
22 T:
23 I:#C<20
24 JY:*ROUND
25 T:END OF TWENTY
   QUESTIONS. PRESS RETURN
26 T:TO START AGAIN.
27 A:
28 J:1
29 END
```

Guess

```
1 H:
2 T:THIS IS THE GAME OF
  GUESS.
3 T:TRY TO GUESS A NUMBER
  BETWEEN 1
4 T:AND 100.
5 C:#G=0
6 C:#N=#R*100
7 *GUESS
8 C:#G=#G+1
9 T:
10 T:GUESS NUMBER #G ?;
12 A:#Q
15 I:#Q<#N
16 TY:TO LOW.
17 I:#Q=#N
18 TY:RIGHT! IN #G GUESSES.
20 JY:*END
21 I:#Q>#N
22 TY:TO HIGH.
23 J:*GUESS
25 *END
26 T:
27 T:PLAY AGAIN ? ;
28 A:
29 M:YES,OK,ALRIGHT,GOOD,Y
30 JY:1
31 T:O.K., GOODBYE!
32 END
```

©

Part II

Last month we featured the skeleton of the world's most intelligent Christmas card – an Atari program which would use several of the machine's special features to delight youngsters and involve them right away in using their computer Christmas present. The article concludes this issue with the spectacular music and animation version of the program. It requires 16K RAM.

An Atari For Christmas

Adding Music And Movement

Brenda Balch
Redondo Beach, CA

We completed the basic framework last month for the Christmas computer program. It should introduce my sister's family to computers in a most friendly way. Now I can think of something unique about each person who will be there Christmas day and turn that into a picture and melody. After a number of attempts my list looks like this:

Name	Picture	Song
Brenda	Renaissance instruments	*Battle Pavane* (this sounds good only in four parts)
Carolyn	Children	*It's a Small World*
Kathy	Dancer	*Dance of the Sugar Plum Fairy*
Ruth	Cake	*This is the Way We ...*

Making Melodies

First the music. All I want is short melodies. All these tunes are in my head, but how do I get them into my computer? I don't play by ear, but fortunately I have a friend who does. I watch his fingers and write down the notes as he plays.

Now I need to determine how long each note is. Out comes the Music Composer, and a lot of trial and error begins. Any mistakes left in these tunes are probably in my head, as well as in my Atari.

Since I think of music in terms of quarter notes, eighth notes, etc., I would like to enter each note as a pitch number (using the table in the *BASIC Reference Manual* is easy), followed by a 4, 8, etc. I also want to change the tempo easily until I like the speed. Thus the PNOTE (play note) subroutine is born. (Later I notice a dotted quarter becomes an awkward 2.66. Next time I'll try something different.)

The first time I wrote this subroutine, I tried to use the variable NOTE, which BASIC would stubbornly turn into NOT E. I finally decided to heed the advice to stay away from variables which start with keywords. (The use of INPUT$ can also give problems in certain contexts.)

The only four-part music I attempt is the Renaissance *Battle Pavane*. One interesting characteristic of this musical phrase is how the parts move at different times. This makes data entry difficult, and requires a different philosophy about when to turn off a note. I use zero as a flag to indicate that a note is not to be turned off (i.e., that it is to be held). I turn each note off just before the next note in that voice starts. If zero were needed to provide rests, one could be used as the flag to hold a note.

As I try various options, the code in the routine takes long enough that it affects the tempo. After several attempts I get a slow but regular beat (see lines 850-895). This involves using the subroutine to give a sixteenth note duration and using the main routine and hold flags to fill out quarter notes, etc. (I'll leave finding a better solution to a rainy day.)

Animation

I look through my list of pictures for required motion. Dancers certainly must dance, and I want to try simple player/missile graphics. Horizontal movement will be enough for me. I bring out my graph paper and discover my major problem is lack of artistic talent. How *do* dancers dance? Several tries (including walking around on my toes to watch what I do) produce the routine at lines 17000-17240.

A dancer should move in time to the music, so calls to PNOTE are alternated with changing the dancer's feet.

Nothing else in my list seems to require motion. But I find I can give the impression of something

Welcome: THORN EMI & Spinnaker

New Software Companies for the Atari 400/800

SUBMARINE COMMANDER

From THORN EMI

Your mission: destroy all enemy merchant shipping in the Mediterranean. Locate enemy ships using your Sonar and periscope, attacking when they're within firing range. But watch your instrument panel carefully to monitor your fuel, oxygen, battery charge and Sonar levels, so you'll be ready for instant action—diving to avoid enemy depth charges, or firing your own.

ROM Cartridge, ~~$49.95~~ Save 15%

Now Thru Dec. 31, You Pay Only $42.46

From THORN EMI

A Batch of British Favorites, From Children's Games to Sophisticated Simulations.

DARTS

From THORN EMI

The traditional pubroom game of darts, depicted in strikingly beautiful graphics and sound. Use your joystick to position the thrower's hand—10 skill levels allow for all ages of players. Once thrown, the dart either finds its mark or bounces off the wire separators (with a realistic "clink" sound).

1 to 4 players including the computer, if you like.

16K Tape, ~~$29.95~~ Save 15%

Now Thru Dec. 31 You Pay Only $25.46

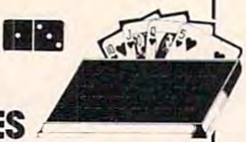
JUMBO JET PILOT

From THORN EMI

You're the pilot in this sophisticated flight simulation, with full instrument panel plus views through the cockpit windows. You'll take off, navigate to your required destination, then land safely. Once you've mastered the basics, try your hand at the "extras": can you fly upside down? Loop the loop? At the end of each "flight", the computer rates your performance. 10 game variations. Requires joystick.

ROM Cartridge, ~~$49.95~~ Save 15%

Now Thru Dec. 31, You Pay Only $42.46



CRIBBAGE & DOMINOES

From THORN EMI

Challenge your computer to a game of either of these favorites. At first, the computer may leave you far behind, but you'll even the score as you work out and try new strategies. Great way to practice and surprise your friends.

16K Tape, $29.95

POOL

From THORN EMI

Great action simulation of the game, with 3 variations: Tournament, 8-Ball and Practice Pool. Play against a friend or the computer, using your joystick to maneuver your "cue stick". With instruction booklet for beginners. Requires joystick.

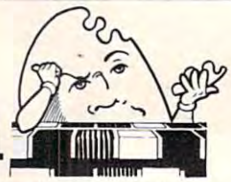
16K Tape, $29.95

SNOOKER & BILLIARDS

From THORN EMI

Great way to learn or practice your technique with the computer. Then, when you're an "accomplished" player, challenge your friends to a few matches. Full joystick control for angle and strength of each shot. Fun for beginners and advanced players; 4 game variations. Requires joystick.

16K Tape, $29.95



HUMPTY DUMPTY & JACK & JILL

From THORN EMI

A new twist on an old favorite: "jigsaw" puzzles for the whole family! The picture of each nursery rhyme theme is shown on the computer screen-divided into 9, 16, 25 or 36 squares, with choice of difficulty levels. The computer jumbles up the picture by moving the squares around, and you have to put them back together again. 48 game variations to delight everyone from pre-school children to adults. Only 1 player at a time, please.

16K Tape, $29.95

Also available:

HICKORY DICKORY DOCK
BAA BAA BLACK SHEEP
16K Tape, $29.95

SNOOPER TROOPS

By Tom Snyder from Spinnaker

As a Snooper Trooper, you're a detective assigned to solve mysteries. You drive around town in your Snoop-Mobile, with wrist radio, SnoopNet computer, camera for SnoopShots, and notebook. Full color graphics and sound.

Case #1: The Granite Point Ghost
Someone is trying to scare the Kim family. Who—or What? And Why? Ages 10 to adult.

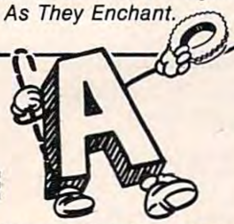
48K Disk, $44.95

Case #2: The Disappearing Dolphin
Someone Stole Lily the Dolphin from the Tabasco Aquarium. Can you find her? Ages 10 to adult.

48K Disk, $44.95

From Spinnaker

A Delightful Collection of Children's Programs That Teach As They Enchant.



STORY MACHINE

From Spinnaker

Imaginative teaching tool that encourages young children to write sentences and stories of their own as the computer ***acts them out!*** They must become familiar with the computer keyboard and use the 38-word vocabulary list correctly. (The computer will not accept incorrect grammar). As they type, the computer shows their words ***in action.*** Delightful and motivational. Ages 5 to 9.

48K Disk, $34.95

FACEMAKER

From Spinnaker

Clever and fun introduction to computer programming and keyboard familiarity. In game 1, the child completes a blank face, choosing from sets of eyes, ears, noses. Game 2 starts with completed face. Child enters instructions to make face smile, wink, wiggle its ears. Game 3 presents sequences of faces which the child must reproduce in correct order. Full color graphics and sound. Ages 4 to 8.

48K Disk, $34.95

CHRISTMAS SAMPLER

From Spinnaker

Great holiday package that brings the classics "to life" with full color graphics and sound, interspersed with animation. Stories are told, plus a selection of Christmas carols, accompanied by screen texts that are easy to follow. Christmas music is the 3rd ingredient—perfect for family gatherings.

48K Disk, ~~$29.95~~ Save 20%

Now Thru Dec. 31, You Pay Only $23.96

Over 1500 Programs for TRS-80, ATARI 400/800, APPLE & IBM.

For Information Call
202-363-9797

Visit our other stores:
829 Bethel Rd., Columbus OH
Seven Corners Center, Falls Church, VA
W. Bell Plaza, 6600 Security Blvd., Baltimore MD

the PROGRAM STORE™

To Order Call Toll-Free
VISA 800-424-2738 MasterCard

MAIL ORDERS: Send check or M.O. for total purchase price, plus $2.00 postage & handling. D.C., MD. & VA.: add sales tax. Charge cards: include all embossed information on the card.

4200 Wisconsin Avenue, NW, Dept. U212 Box 9582 Washington, D.C. 20016

Atari Holiday Reading

the PROGRAM STORE™

YOUR ATARI COMPUTER

By Poole, McNiff & Cook from Osborne/McGraw-Hill
Comprehensive, all-in-one guide for Atari 400/800 computers that is helpful to the beginning and advanced programmer. Complete operating instructions; detailed BASIC programming, including a handy alphabetical glossary of BASIC statements and functions; advanced BASIC; computer graphics; and tips on hardware, peripherals and compatible software.

Softcover, $16.95

COMPUTE!'S SECOND BOOK OF ATARI

From Compute! Books
An **all new** collection of **never-before** published articles on the Atari for beginner to advanced users. Topics include: Utilities, Programming, Graphics, Applications and Beyond BASIC.

Spiral-bound, $12.95

GAMES & RECREATIONS

From Reston
This new book is packed with games and ideas on how to create your own. You'll discover the special Atari capabilities of the keyboard, graphics, sound and color. The book provides an entertaining way to learn more about general programming, too.
Part I: Learning Through Games. Part II: Games for the Atari. Part III: The Atari Special. Plus seven appendices.

Softcover, $14.95 Hardcover, $19.95

MASTER MEMORY MAP

From Santa Cruz Software
If you are serious about programming the Atari, this booklet could become one of your most important tools. A highly detailed map of the Atari's memory, it details thousands of locations and routines. Using this booklet makes easier the use of display list, player/missile, and interrupt graphics. The "Miscellaneous Notes" section contains a wealth of knowledge picked up by Santa Cruz in their explorations of the Atari. Also included are notes on the new GTIA graphics chip.

Softcover, $6.95

VISICALC HOME AND OFFICE COMPANION

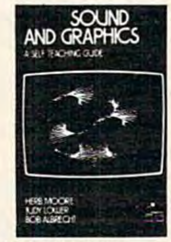
By Castlewitz and Chisausky from Osborne/McGraw Hill
Fifty VisiCalc models for home and office, including: investments, inventory, sales forecasts, payroll, personal net worth, home budget planning, family insurance needs. Each comes with model listings, sample printed reports and narrative.

Softcover, $15.95

6502 ASSEMBLY LANGUAGE SUBROUTINES

By Leventhal and Saville from Osborne/McGraw-Hill
If you're interested in using assembly language quickly, this book is ideal. It describes general 6502 programming methods, provides code for more than 40 subroutines to help you improve your programming skills, debug or revise an existing program; add instructions and addressing modes.

Softcover, $15.95



ATARI SOUND AND GRAPHICS

By Moore, Lower and Albrecht from John Wiley
This self-paced, self-teaching guide will have you seeing and hearing things on your Atari in no time—even if you're a complete beginner. You'll learn to compose and play melodies, draw cartoons, create sound effects and games. Each section teaches something new in BASIC, the most commonly used computer language.

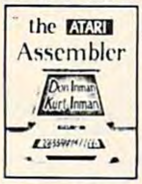
Softcover, $9.95

PICTURE THIS!

An Introduction to Computer Graphics for Kids of All Ages

By David D. Thornburg from Addison-Wesley
This book promises to become the "modern replacement for coloring books and crayons". It's a learn-by-doing manual that uses PILOT, a simpler language than BASIC, and Turtle Geometry to teach kids to create pictures in full color from simple lines to complex angles and curves. Recommended for use in conjunction with PILOT Cartridge.

Spiral-bound, $14.95



ATARI ASSEMBLER

By Don & Kurt Inman from Reston
While the Atari Assembler Cartridge comes with an operating manual, it assumes that you already know assembly language. If you're new to the Atari or its 6502 processor, this book is a must. The Inmans guide you through the rudiments of this fascinating type of programming in clear, easy steps. Includes full listing and description of 6502 mnemonics and addressing modes. Recommended for use in conjunction with Assembler Cartridge.

Softcover, $12.95

INSIDE ATARI DOS

By Bill Wilkinson from Compute
The comprehensive manual on the disk File Manager System (FMS), commonly known as Atari DOS 2.0S. Contains the only complete and official listing for the system, plus a full description of: the external view, charts & tables, various interfaces and functions of individual subroutines.

Spiral-bound, $19.95

DE RE ATARI

From APX
Translated from Latin, the title of this book is "All About Atari" and it means what it says! Used in combination with Atari's Technical Reference Manual, advanced programmers will be able to learn to exploit the many hardware and operating system features that make the Atari 400/800 so tremendously versatile. Includes a useful discussion of the new GTIA chip. Once you know Atari BASIC and assembler, this book is a must.

Loose leaf (binder not supplied), $19.95

COMPUTERS FOR KIDS

Atari Edition By Sally Larsen from Creative
The BASIC programming manual written for kids, from flowcharts to color graphics, including the sure-to-please program, "Scare Mom with an Elephant." Detailed instructions and sketches plus glossary of statements and commands. With lesson plans and tips for parents and teachers.

Softcover, $4.95

STIMULATING SIMULATIONS,

Atari Version, 2nd Edition

By C.W. Engel from Hayden
A handbook of 12 simulation games including Art Auction, Starship Alpha, Monster Chase and Devil's Dungeon—each complete with listing, sample run, instructions and program documentation, including flowchart and ideas for variations.

Softcover, $5.95

THE PROGRAM STORE • Dept. U212 • Box 9582 • 4200 Wisconsin Avenue, NW • Washington, D.C. 20016

Item	Tape/Disk/Book	Price	Postage $2.00	Name
			Total ______	Address
			☐ CHECK ☐ VISA	City ______ State ______ Zip ______
			☐ MASTERCARD	Card # ______ Exp ______

Orders Must Be Received By December 10th for Christmas Delivery

Repeat of Last Month's Surprise:

Limited Quantities, So Order Early.

FAR PROTECTOR
From Epyx
YOU are the final defense against the enemy's nuclear attack! Six cameras are your eyes on the sky; you're armed with a Repulser Beam and laser system. Your charge: Destroy the enemy's satellites and missiles. Requires joystick.
ROM Cartridge, $59.95 16K Required

MINER 2049ER
By Bill Hogue from Big Five
This is the author's first game for the Atari—he's already well known for his bestsellers for the TRS-80—and we think you'll really enjoy it. There are more than ten screens of colorful mining-related machinery that you'll move around the screens, ducking, dodging and bobbing your way to a high score. Requires joystick.
ROM Cartridge, $49.95 16K Required

SPEEDWAY BLAST
By Dave Morock from IDSI
Racing game with a twist: you must guide your racer on an overland trip, dodging (or blasting) the asphalt-eating monsters that try to keep you from capturing their precious diamonds. Steer carefully! Requires joystick.
ROM Cartridge, $39.95 16K Required

DEADLY DUCK
From Sirius
The Crabs have taken to the air, armed with bricks and bombs, to drive our friend, DEADLY DUCK, out of his pond. But Deadly's gonna fight back, with a gun tucked into his bill. Lots of fun and action; 6 levels of play. Requires joystick.
ROM Cartridge
So new, price not available at press time. Please call. **Also available for VCS.**

DELUXE INVADERS
From Roklan
All the fun, excitement and video quality of an arcade game in your home! If you **hate** Alien Invaders, you'll **love** this one! Battle the Invaders, and as your speed and defense skills improve, the aliens get meaner. With each successful defense you'll be challenged to an even tougher competition. Not for the timid! Requires joystick.
ROM Cartridge, $39.95 16K Required

FAST EDDIE
From Sirius
On your mark; get set; ***go Eddie!!*** FAST EDDIE's off and running, dashing up and down every ladder he can find, hunting for prizes. Quick, there's a heart floating on the 2nd floor! Ooops, look out—there's a Sneaker, sneaking up on you —***jump, Eddie!*** Great animation; 8 skill levels; lots of action. Requires joystick.
ROM Cartridge
So new, price not available at press time. Please call.

GORF
From Roklan
A unique sight and sound adventure in the interstellar war against the Gorfian Empire. You must repel attacks by Droids, Anti-Gravity Bonks, Anti-Particle Lasers, Gorfian fighters and torpedos, etc. Four levels, from an Astrobattle to a full-fledged Space War. Requires joystick.
ROM Cartridge, $44.95 16K Required
Disk, $39.95 24K Required

FISHES
From Epyx
You lead your school of fish through the pleasures and perils of oceanic life. As you find food, your school will grow. But hungry sharks and other predators also inhabit this tranquil world, so ***swim carefully***. Requires joystick.
ROM Cartridge, $39.95 16K Required

K-RAZY SHOOT-OUT
From K-Byte/CBS
The object of the game is to advance your Space Commander into the Alien Control Sector, eliminate the Alien Forces and escape to the next (more difficult) sector. **No two games are alike**, since the Alien Control Sectors are created at random, giving you **millions** of combinations of barriers and escape routes. Requires joystick.
ROM Cartridge, $49.95 8K Required

WORM WAR I
From Sirius
Hoards of gigantic worms are slithering around the city of Teriyaki. . . WORM WAR I has begun! Much of the city has been crushed under the weight of these mutant creatures, and the only thing that can stop them is the specially armed anti-worm tank. Are you brave enough to drive it, and make Teriyaki safe again? 9 play options; great color and sound; for 1 or 2 players. Requires joystick.
ROM Cartridge
So new, price not available at press time. Please call. **Also available for VCS.**

EMBARGO
By Bill Hooper from Gebelli
There is a strict trade embargo on Zorel 6. The Council has ruled that all foodstuffs, materials, trade goods and fuel must pass the close scrutiny of the Orelian Guards. Only goods essential for galactic security are guarded; all others are disbursed to the Council's overflowing warehouses. Select from 9 levels of play. Requires joystick.
ROM Cartridge, $44.95 8K Required

Over 1500 Programs for TRS-80, ATARI 400/800, APPLE & IBM.

For Information Call
202-363-9797

Visit our other stores:
829 Bethel Rd., Columbus OH
Seven Corners Center, Falls Church, VA
W. Bell Plaza, 6600 Security Blvd., Baltimore MD

the PROGRAM STORE™

To Order Call Toll-Free
VISA **800-424-2738** MasterCard

MAIL ORDERS: Send check or M.O. for total purchase price, plus $2.00 postage & handling. D.C., MD. & VA.: add sales tax. Charge cards: include all embossed information on the card.

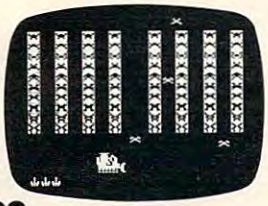
4200 Wisconsin Avenue, NW, Dept. U212 Box 9582 Washington, D.C. 20016

First-Ever Atari ROM Cartridges!

THE PROGRAM STORE™

Limited Quantities, So Order Early.

PLATTERMANIA
From Epyx
If you've always wanted to juggle, here's your chance. Start with 3 spinning plates, and juggle your way up to expert status: keeping 18 plates going at once! You'll have to concentrate: spin too fast and they fly off the screen; spin too slowly and they fall. Requires joystick.
ROM Cartridge, $39.95 16K Required

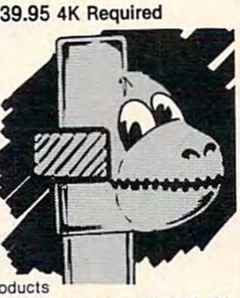


K-RAZY KRITTERS
From K-Byte/CBS
Your Command Ship is faced with ***eight columns*** of Alien Kritters. You have Regular or "Supermissiles" to eliminate the Kritters, descending on your Star Base at various speeds and frequencies. Ten levels of play—great fun for the nimble-fingered! Requires joystick.
ROM Cartridge, $49.95 8K Required

CROSSFIRE
From On-Line
The most original, exciting arcade game—designed exclusively for the home computer—that we've seen! The aliens have taken over your city, and everyone left peaceably—***except you.*** You wander the streets, shooting at aliens firing at you ***from all 4 directions.*** You must move quickly to avoid their CROSSFIRE, gather bonus tokens, and recharge your laser. It's one of our favorites! Requires Joystick.
ROM Cartridge, $44.95 16K Required
16K Tape or 32K Disk, $29.95

SOLDIERS OF SORCERY
From Epyx
A multi-player fantasy role-playing game in which you, as a warrior or wizard, search the land for wealth and experience. The game is ***different each time*** you enter the world of wolves, bats, dragons, and more! Requires joystick.
ROM Cartridge, $59.95 16K Required

FIREBIRD
From Gebelli
You are Piggo the Firefighter in this fast-action game. The firebird drops fire on buildings in your district. You must put out the fires; try to save the people who jump, and get them on rescue helicopters. Accumulate points for each successful action. Requires joystick.
ROM Cartridge, $39.95 4K Required



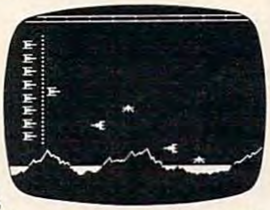
MONKEY WRENCH
From Eastern Data Products
A BASIC and machine language programmers' aid for the Atari 800. Works with BASIC, adding 9 new direct mode commands including: auto line numbering, delete lines, change margins, memory test, hex/dex conversion, renumber BASIC, cursor exchange and machine language monitor. Monitor contains 15 commands used to interact with the 6502.
ROM Cartridge, $49.95.

BEANY BOPPER
From Sirius
Watch out—those Beanies are back, buzzing the city, and that means ***trouble!*** Shooting them makes them meaner; poison doesn't work—what can we do? It's BEANY BOPPER to the rescue, with his pivotal laser and rapid-fire stunt gun. Fast action, exciting sound and color; 6 play options. Requires joystick. **Also available for VCS.**
ROM Cartridge
So new, price not available at press time. Please call.

ALIEN GARDEN
From Epyx
This fantasy world is inhabited by a collection of "Incredible Edibles": some delicious, some poisonous, some explosive. You must discover the best way to eliminate them from the garden—without eliminating yourself! The faster you go, the more points you earn. Requires joystick.
ROM Cartridge, $39.95 16K Required

POOL 400
From IDSI
Looks and plays just like the real thing! With straight pool, nine ball, eight ball and rotation. Features include: instant replay, slow motion, 5 friction levels, and choice of colored or numbered balls. Play against a friend or the computer. Requires joystick.
ROM Cartridge, $39.95 16K Required

K-RAZY ANTIKS
From K-Byte/CBS
The White Ant needs all your help! You must guide it safely through the maze of tunnels in the Anthill; help it deposit and protect its White Eggs—while looking out for the Anteater and Enemy Ants who are trying to hatch their Enemy Eggs. Choice of 6 mazes and 99 levels of difficulty. Requires joystick.
ROM Cartridge, $49.95 8K Required



K-STAR PATROL
From K-Byte/CBS
Your lead Star Ship must destroy the Alien Attack Ships, and eliminate the Intergalactic Leeches that are invading your territory. You must also replenish your Force Field Energy periodically by diving between jagged mountains into the lakes below. 10 levels of difficulty. Requires joystick.
ROM Cartridge, $49.95 8K Required

WIZARD OF WOR
From Roklan
Can you defeat the WIZARD OF WOR? First you must descend into the ever-changing maze of Dungeons with your Worriors, and do battle with the monsters you encounter, like the Burwors, Garwors, Worluk and enemy Worriors. Only then can you turn your attention to the Wizard, who can teleport magically around the screen, hurtling lightning bolts as he moves. Simultaneous 1 or 2 player action.
ROM Cartridge, $44.95 16K Required
Disk, $39.95 32K Required

THE PROGRAM STORE • Dept. U212 • Box 9582 • 4200 Wisconsin Avenue, NW • Washington, D.C. 20016

Item ____ Tape/Disk/ROM ____ Price ____
Postage $2.00
Total ________
☐ CHECK ☐ VISA
☐ MASTERCARD

Name________________
Address________________
City________ State________ Zip________
Card #________________ Exp________

Orders Must Be Received By December 10th for Christmas Delivery

happening by building cakes in layers, and changing background colors for the children.

The Pictures

I draw all of my pictures on graph paper and then turn them into X,Y coordinates. I try to standardize colors, but end up with a sizable list anyway. The only color which gives me much trouble is yellow. I need two sets of parameters for yellow. The color I get seems to depend upon the context of the colors around it. (If the coconut cake looks green on your screen, try the other yellow.) The colors I used are:

SETCOLOR x,a,b

a	b	color
0	0	black
0	4	dark gray
0	8	gray
0	14	white
1	4	gold
1	14	yellow #1
2	14	light orange
3	4	red
4	12	pink
5	4	purple
9	6	light blue
12	2	green
13	12	yellow #2
14	2	brown

Common Subroutines

I need common subroutines in this program for four things: delays, plotting, sound, and checking input strings.

Delay Subroutines:

Name	Location	Function
SDELAY	900	Short delay
MDELAY	930	Medium delay
LDELAY	960	Long delay (to allow a first grader to read two lines)

The delay subroutines simply loop a fixed number of times.

Plotting Subroutines:

Name	Location	Function
HPLOT	100-110	Plot horizontal lines, reading start x,y values and length
VPLOT	150-160	Plot vertical lines and reflected vertical lines (around an x-axis of REFL) reading start x,y values and length
PPLOT	200	Plot points, reading the number of points, and then the x,y values
HPLOTT	250-260	Same as HPLOT, except lines are translated by (OFFX, OFFY)
HPLOTTRF	300-310	Same as HPLOTT except lines are reflected around an x-axis of REFL
SQPLOT	350	Plot 3x3 squares, reading the number of squares and the x,y values of the upper left corner of each square

The plotting subroutines are written as they are needed. For example, the only times I need vertical line segments to make my picture, the picture is symmetrical around an x-axis. Therefore, the only vertical plot routine plots the original and the reflected values.

Sound Subroutines:

Name	Location	Function
PNOTE	800-810	Reads a pitch and duration and plays a note; if it reaches the end it starts over
PCHORD	850-895	Plays the chord in ANOTE for one sixteenth duration (see earlier discussion)

Input Checking Subroutine:

Name	Location	Function
CHECKI	700-720	Described in Part I

Main Subroutines

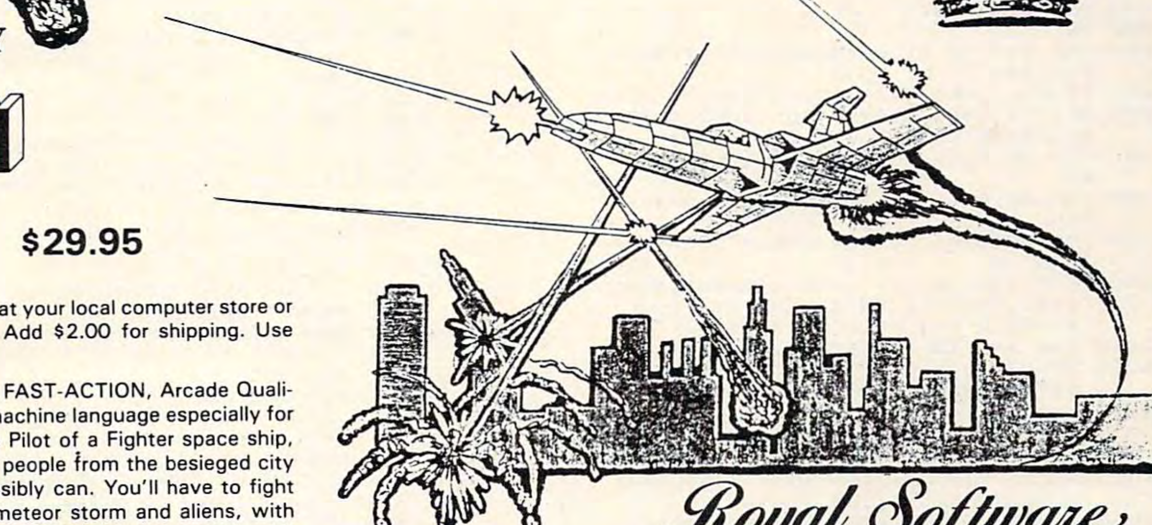
The main subroutines are entered by using the GOSUB expression in line 3050. Therefore, each routine starts on a line number which is a multiple of 1000. Note that printing to the screen after graphics mode x + 16 returns to graphics mode 0. Therefore, the only explicit Graphics 0 commands are required after the Christmas tree which uses graphics mode 3. There is one main subroutine for each person on Christmas day:

Location	Picture
11000-11330	Renaissance instruments
15000-15440	Children
17000-17240	Dancer
19000-19210	Cake

One miscellaneous note: the Renaissance instruments pictured are krumhorns.

```
10 GOTO 1000
100 READ X,Y,NUM:IF NUM=0 THEN RETURN
110 FOR CT=0 TO NUM-1:PLOT X+CT,Y:NEX
    T CT:GOTO 100
150 READ X,Y,NUM:IF NUM=0 THEN RETURN
160 FOR CT=0 TO NUM-1:PLOT X,Y+CT:PLO
    T REFL-X,Y+CT:NEXT CT:GOTO 150
200 READ NUM:FOR L=1 TO NUM:READ X,Y:
    PLOT X,Y:NEXT L:RETURN
250 READ X,Y,NUM:IF NUM=0 THEN RETURN
260 FOR CT=0 TO NUM-1:PLOT X+OFFX+CT,
    Y+OFFY:NEXT CT:GOTO 250
300 READ X,Y,NUM:IF NUM=0 THEN RETURN
310 FOR CT=0 TO NUM-1:PLOT REFL-X+OFF
    X-CT,Y+OFFY:NEXT CT:GOTO 300
350 READ NUM:FOR CT=1 TO NUM:READ X,Y
    :FOR SQCT=0 TO 2:PLOT X,Y+SQCT:DR
    AWTO X+2,Y+SQCT:NEXT SQCT:NEXT CT
    :RETURN
700 READ COMPARE$,INDEX:IF INDEX=0 TH
    EN RETURN
```

METEOR STORM



MADE EXCLUSIVELY FOR THE

***ATARI**

24K Disk
16K Tape **$29.95**

Ask for Meteor Storm at your local computer store or order direct from us. Add $2.00 for shipping. Use your Visa, M/C.

METEOR STORM Is a FAST-ACTION, Arcade Quality, game. Written in machine language especially for the Atari. You, as the Pilot of a Fighter space ship, must rescue as many people from the besieged city of Dracon as you possibly can. You'll have to fight your way through a meteor storm and aliens, with your lasers blasting a path. Then you must dock with the mother ship and safely deliver your precious cargo of human lives. This is your chance to be the HERO of Dracon.

Dealer Inquiries Invited
*Atari is a Trademark of Atari Inc.

Royal Software
2160 West 11th Ave., Eugene, Oregon 97402
(503) 683-5361

Beat the clock! Outsmart your friends!

WORDRACE

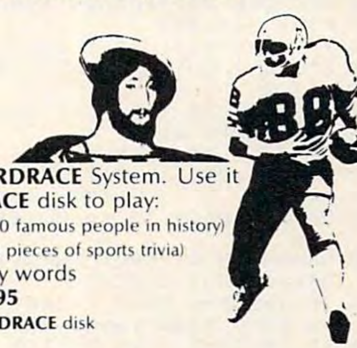
The high-speed dictionary game

At last, an educational game that's really fun to play! You start each turn with a word, 6 definitions, and a counter set to 600 points. As time passes the points tick away. The sooner you pick the **correct** definition, the more points you get, but the sooner you guess wrong, the more points you **lose**.

3 levels of play on one disk:
• Beginner (ages 9-14) • Regular • Challenge
2000 words and definitions
$24.95
ATARI: 32K — disk — BASIC **APPLE:** 48K — disk — APPLESOFT

Now available on cassette for the ATARI 400/800 $19.95
Features: Beginner's and Intermediate WORDRACE

YOU CAN PLAY 3 NEW GAMES WITH YOUR WORDRACE DISKETTE:



The next disk in the **WORDRACE** System. Use it along with your **WORDRACE** disk to play:
- **CLAIM TO FAME** (600 famous people in history)
- **SPORTS DERBY** (600 pieces of sports trivia)
- Plus more vocabulary words

$19.95
Requires **WORDRACE** disk

WORDRACE ACCESSORY #1

Try **ABUSE**
The insult program.
• Funny • Unpredictable • Interactive
• Guaranteed to call you something you've never been called before!
$19.95
ATARI: 40K — disk — BASIC
APPLE: 48K — disk — APPLESOFT

DON'T ASK
COMPUTER SOFTWARE INC.

2265 Westwood Blvd., Ste. B-150
Los Angeles, California 90064.
(213) 397-8811

Dealer inquiries welcome

Available at your computer store or direct from DON'T ASK
Include $2.00 shipping for each program (Calif. residents add 6% tax)

ATARI is a trademark of ATARI INC. APPLE and APPLESOFT are trademarks of APPLE COMPUTER

```
710 IF COMPARE$=INPUT$ THEN RETURN
720 GOTO 700
800 READ MNOTE,LENGTH:IF LENGTH=0 THE
    N RESTORE MUSIC:GOTO 800
810 SOUND 0,MNOTE,10,8:FOR DELAY=1 TO
     (16/LENGTH)*TEMPO:NEXT DELAY:SOU
    ND 0,0,0,0:RETURN
850 FOR I=0 TO 3:IF ANOTE(I)=0 THEN G
    OTO 875
870 SOUND I,0,0,0
875 NEXT I
880 FOR I=0 TO 3:IF ANOTE(I)=0 THEN G
    OTO 895
890 SOUND I,ANOTE(I),10,VOL(I)
895 NEXT I:FOR DELAY=1 TO TEMPO:NEXT
    DELAY:RETURN
900 FOR DELAY=1 TO 20:NEXT DELAY:RETU
    RN
930 FOR DELAY=1 TO 200:NEXT DELAY:RET
    URN
960 FOR DELAY=1 TO 2000:NEXT DELAY:RE
    TURN
1000 DIM INPUT$(25),COMPARE$(25),NAME
     (20),HUE(2),LUM(2),ANOTE(3),VOL(
     4):PEOPLE=4:POKE 53277,0
1010 FOR I=1 TO 20:NAME(I)=0:NEXT I
1020 HPLOT=100:VPLOT=150:PPLOT=200:HP
     LOTT=250:HPLOTTRF=300:SQPLOT=350
1030 CHECKI=700:PNOTE=800:PCHORD=850:
     SDELAY=900:MDELAY=930:LDELAY=960
     :S11050=11050
1040 HUE(0)=3:LUM(0)=4:HUE(1)=0:LUM(1
     )=14:HUE(2)=8:LUM(2)=4
2000 GRAPHICS 3:SETCOLOR 0,3,4:SETCOL
     OR 1,13,12:SETCOLOR 2,12,2:SETCO
     LOR 4,0,0
2010 ? "MERRY CHRISTMAS!":? "I AM YOU
     R FRIENDLY COMPUTER"
2020 ? "WILL YOU TALK TO ME?";
2030 COLOR 3:PLOT 26,15:DRAWTO 18,0:P
     OSITION 10,15
2040 POKE 765,3:XIO 18,#6,0,0,"S:"
2050 PLOT 19,18:DRAWTO 19,16:DRAWTO 1
     7,16:POSITION 17,18
2060 XIO 18,#6,0,0,"S:"
2070 RESTORE 2200:COLOR 2:GOSUB PPLOT
2080 RESTORE 2300:COLOR 1:GOSUB PPLOT
2090 POKE 764,255
2100 IF PEEK(764)<>255 THEN 2900
2110 FOR L=1 TO 100:NEXT L:SETCOLOR 0
     ,12,2
2120 FOR L=1 TO 100:NEXT L:SETCOLOR 0
     ,3,4:GOTO 2100
2200 DATA 11,18,0,19,3,17,5,20,7,16,8
     ,19,10,23,11,17,12,12,13,22,14,1
     5,15
2300 DATA 5,20,5,15,10,20,12,19,14,12
     ,15
2900 GRAPHICS 0:INPUT INPUT$:RESTORE
     2960:GOSUB CHECKI
2910 IF INDEX=0 THEN ? "ANY ANSWER IS
      A GOOD SIGN":GOTO 2950
2920 IF INDEX=1 THEN ? "I'M GLAD":GOT
     O 2950
2930 ? "YOU MUST HAVE GOTTEN OUT OF T
     HE WRONG SIDE OF BED THIS MORNIN
     G"
2940 ? "ANYWAY, ";:GOTO 2950
2950 ? "MY NAME IS ATARI":GOTO 3000
2960 DATA YES,1,Y,1,YEAH,1,NO,2,N,2,E
     ND,0
3000 ? "WHAT IS YOUR NAME";:INPUT INP
     UT$
3010 RESTORE 3110:GOSUB CHECKI
3020 IF INDEX=0 THEN ? "HMM ... I DON
     'T KNOW YOU.":? "ARE YOU SURE YO
     U SPELLED YOUR NAME{4 SPACES}RIG
     HT?":GOTO 3000
3030 IF INDEX=20 THEN ? "THERE ARE TO
     O MANY ";:? INPUT$;:? "'S HERE":
     ? "TRY AGAIN":GOTO 3000
3040 NAME(INDEX)=1:? :? INPUT$;:? ",
     I KNOW SOMETHING ABOUT YOU.":?
3050 GOSUB 10000+INDEX*1000:TOTAL=0
3060 FOR I=1 TO 10:TOTAL=TOTAL+NAME(I
     ):NEXT I
3070 IF TOTAL=PEOPLE THEN ? "IT HAS B
     EEN NICE TALKING TO EVERYONE.":G
     OSUB LDELAY:GOTO 3090
3080 ? "I HAVEN'T TALKED TO EVERYONE
     YET":? "I HOPE SOMEONE ELSE WANT
     S TO TALK TO  ME.":GOTO 3000
3090 GRAPHICS 18:SETCOLOR 4,12,2:SETC
     OLOR 0,3,4:POSITION 2,5:? #6;"ME
     RRY CHRISTMAS"
3095 RESTORE 3400:MUSIC=3400:TEMPO=30
     :FOR I=1 TO 11:GOSUB PNOTE:NEXT
     I:END
3110 DATA BRENDA,1,BB,1
3150 DATA CAROLYN,5
3170 DATA KATHY,7,KATHERINE,7
3190 DATA RUTH,9,GRANDMA,9
3300 DATA MOM,20,MOTHER,20,MOMMY,20,D
     AD,20,DADDY,20,FATHER,20,END,0
3400 DATA 144,4,144,4,144,2,144,4,144
     ,4,144,2,144,3,121,4,182,2.66,16
     2,8,144,1,0,0
11000 ? "YOU PLAY RENAISSANCE INSTRUM
      ENTS.":GOSUB LDELAY
11010 GRAPHICS 19:SETCOLOR 4,1,14:SET
      COLOR 0,14,2:COLOR 1:REFL=40:X=
      12:Y=23
11020 FOR J=1 TO 15:FOR I=0 TO 2:PLOT
       X,Y-I:PLOT REFL-X,Y-I:NEXT I:X
      =X+1:Y=Y-1:NEXT J
11040 RESTORE 11200:GOSUB VPLOT:RESTO
      RE 11300:TEMPO=2:VOL(0)=8:VOL(1
      )=4:VOL(2)=4:VOL(3)=6
11050 FOR I=0 TO 3:READ X:ANOTE(I)=X:
      NEXT I:READ LENGTH:IF LENGTH=0
      THEN 11110
11055 COUNT=16/LENGTH-1
11060 GOSUB PCHORD:FOR I=0 TO 3:ANOTE
      (I)=0:NEXT I:FOR CT=1 TO COUNT:
      GOSUB PCHORD:NEXT CT:GOTO S1105
      0
11110 SOUND 0,0,0,0:SOUND 1,0,0,0:SOU
      ND 2,0,0,0:SOUND 3,0,0,0:RETURN
11200 DATA 11,22,2,10,22,2,9,19,5,8,1
      6,7,7,15,5,6,14,3,27,4,5,28,3,5
      ,29,2,6,30,1,7,31,0,7,32,1,5,33
      ,2,3,34,3,1
11210 DATA 0,0,0
11300 DATA 53,64,81,162,4,47,60,0,121
      ,8,60,60,96,121,8,53,64,81,162,
      8,53,64,0,162,8,47,60,0,121,8,6
      0,60,96,121,8
11310 DATA 53,64,81,162,8,53,64,0,162
      ,8,47,60,0,121,8,60,60,96,121,8
      ,53,64,81,162,8,0,81,0,162,8,47
      ,60,0,121,8
11320 DATA 0,0,91,0,8,0,81,96,121,8,5
      3,0,0,0,8,60,72,91,182,4,0,81,1
      08,162,4,64,0,0,0,4,60,96,121,2
      43,1
11330 DATA 0,0,0,0,1,0,0,0,0,0
15000 ? "YOU TEACH CHILDREN.":GOSUB L
      DELAY
15010 GRAPHICS 21:SETCOLOR 4,1,4:SETC
      OLOR 0,2,14:SETCOLOR 1,12,2:SET
      COLOR 2,3,4:REFL=13
15020 OFFX=10:OFFY=0:RESTORE 15200:CO
      LOR 3:GOSUB HPLOTT:RESTORE 1521
      0:COLOR 1:GOSUB HPLOTT
15030 OFFX=54:OFFY=26:RESTORE 15200:C
      OLOR 2:GOSUB HPLOTTRF:RESTORE 1
      5210:COLOR 1:GOSUB HPLOTTRF
15040 OFFX=32:OFFY=26:RESTORE 15250:C
      OLOR 3:GOSUB HPLOTT:RESTORE 152
      60:COLOR 1:GOSUB HPLOTT
15050 OFFX=54:OFFY=0:RESTORE 15200:CO
      LOR 2:GOSUB HPLOTTRF:RESTORE 15
      210:COLOR 1:GOSUB HPLOTTRF
15060 OFFX=32:OFFY=0:RESTORE 15250:CO
      LOR 3:GOSUB HPLOTT:RESTORE 1526
      0:COLOR 1:GOSUB HPLOTT
15070 OFFX=10:OFFY=26:RESTORE 15250:C
      OLOR 2:GOSUB HPLOTT:RESTORE 152
      60:COLOR 1:GOSUB HPLOTT
15080 RESTORE 15400:MUSIC=15400:TEMPO
```

```
       =50:FOR CT=1 TO 21:GOSUB PNOTE:
       NEXT CT
15090 FOR CT=6 TO 14 STEP 2:SETCOLOR
       4,1,CT:GOSUB PNOTE:NEXT CT:RETU
       RN
15200 DATA 4,7,5,4,8,6,4,9,10,4,10,10
       ,4,11,6,3,12,8,2,13,10,1,14,12,
       0,15,14,0,0,0
15210 DATA 5,0,4,4,1,6,4,2,3,8,2,2,4,
       3,6,4,4,6,4,5,6,5,6,4,4,16,2,4,
       17,2,4,18,2,4,19,2,4,20,2,4,21,
       2
15220 DATA 8,16,2,8,17,2,8,18,2,9,19,
       2,10,20,2,11,21,2,0,0,0
15250 DATA 4,8,6,0,9,14,0,10,14,4,11,
       6,4,12,6,4,13,6,4,14,6,4,15,6,4
       ,16,2,8,16,2,4,17,2,8,17,2,0,0,
       0
15260 DATA 5,0,4,4,1,6,4,2,1,6,2,2,9,
       2,1,4,3,6,4,4,6,4,5,6,5,6,4,5,7
       ,4,4,18,2,4,19,2,3,20,2,3,21,2,
       8,18,2,8,19,2
15270 DATA 9,20,2,9,21,2,0,0,0
15400 DATA 121,2.66,121,8,96,4,121,4
15410 DATA 108,2.66,108,8,108,2
15420 DATA 108,2.66,108,8,91,4,108,4,
       96,2.66,96,8,96,2
15430 DATA 96,2.66,96,8,81,4,96,4,91,
       2.66,92,8,91,4
15440 DATA 96,8,108,8,162,2,128,2,121
       ,2,0,0
17000 ? "YOU LIKE TO DANCE.":GOSUB LD
       ELAY
17010 GRAPHICS 19:SETCOLOR 4,2,14
17020 POKE 559,46:A=PEEK(106)-8:POKE
       54279,A:PMBASE=A*256:Y=52
17030 RESTORE 17200:FOR I=PMBASE+512
       TO PMBASE+639:POKE I,0:NEXT I
17040 FOR I=PMBASE+512+Y TO PMBASE+51
       9+Y:READ V:POKE I,V:NEXT I
17050 POKE 704,132:POKE 53277,3:MUSIC
       =17210:TEMPO=30
17060 FOR X=192 TO 49 STEP -2:POKE 53
       248,X:GOSUB PNOTE
17070 POKE PMBASE+519+Y,48:POKE 53248
       ,X-1:GOSUB PNOTE
17080 POKE PMBASE+519+Y,72:NEXT X
17090 POKE 53277,0:RETURN
17200 DATA 152,88,56,24,60,126,40,72
17210 DATA 121,8,128,8,121,8,128,8,12
       1,16,0,16,0,8,128,4,108,4,121,4
       ,96,2
17220 DATA 91,8,96,8,91,8,96,8,108,8,
       121,8,144,8,162,8,162,4,0,8,121
       ,8,128,2
17230 DATA 144,8,144,8,144,8,144,8,14
       4,8,0,8,162,4,144,8,144,8,144,8
       ,144,8,144,8,0,8,162,4
17240 DATA 121,8,121,8,121,8,121,8,12
       1,8,128,8,108,8,121,8,121,8,128
       ,8,144,8,162,8,91,2,0,0
19000 ? "YOU LIKE TO BAKE -":? :? "CO
       CONUT CAKE.":GOSUB LDELAY
19010 GRAPHICS 19:RESTORE 19200:SETCO
       LOR 0,13,12:SETCOLOR 1,0,14:MUS
       IC=19200:TEMPO=30
19020 COLOR 1:PLOT 10,17:DRAWTO 29,17
       :PLOT 10,16:DRAWTO 29,16:FOR CT
       =1 TO 4:GOSUB PNOTE:NEXT CT
19030 COLOR 2:PLOT 10,15:DRAWTO 29,15
       :FOR CT=1 TO 4:GOSUB PNOTE:NEXT
        CT
19040 COLOR 1:PLOT 10,14:DRAWTO 29,14
       :PLOT 10,13:DRAWTO 29,13:FOR CT
       =1 TO 3:GOSUB PNOTE:NEXT CT
19050 COLOR 2:PLOT 10,12:DRAWTO 29,12
       :FOR CT=1 TO 3:GOSUB PNOTE:NEXT
        CT
19060 COLOR 1:PLOT 10,11:DRAWTO 29,11
       :PLOT 10,10:DRAWTO 29,10:FOR CT
       =1 TO 4:GOSUB PNOTE:NEXT CT
19070 COLOR 2:PLOT 10,9:DRAWTO 29,9:F
       OR CT=1 TO 4:GOSUB PNOTE:NEXT C
       T
19080 COLOR 1:PLOT 10,8:DRAWTO 29,8:P
       LOT 10,7:DRAWTO 29,7:FOR CT=1 T
       O 5:GOSUB PNOTE:NEXT CT
19090 COLOR 2:PLOT 10,6:DRAWTO 29,6:S
       ETCOLOR 0,0,14:FOR CT=1 TO 2:GO
       SUB PNOTE:NEXT CT:RETURN
19200 DATA 162,8,162,8,162,8,162,4,12
       8,8,108,4,128,8,162,2.66,144,4,
       144,8,144,2.66,173,4,193,8,217,
       2.66
19210 DATA 162,8,162,8,162,8,162,4,12
       8,8,108,4,128,8,162,4,162,8,144
       ,4,144,8,217,4,173,8,162,2.66,1
       62,2.66,0,0,0
```

©

You are trapped inside a maze, and you can see only a short distance along its dark corridors as you try to find your way out. A challenging game for the OSI, PET/CBM, Atari, VIC-20, and Apple.

Hidden Maze

Gary Boden
Narragansett, RI

Mazes present a challenge different from arcade-type "shootout" games, but the appeal of a maze can quickly fade once it has been solved. A special program, "Maze Generator" (**COMPUTE!**, December 1981, #19), remedies that problem by drawing a different maze on each run. I have enhanced its challenge by hiding the complete maze from the player and showing only a realistically limited view from any position inside it. Although the view is from above rather than ground level, the player still gets a claustrophobic feeling similar to that of actually being inside the maze and groping along the corridors.

The objective is simply to find a way out of the maze in the least amount of time. Realism is added by showing at most only seven cells in any of the four possible directions of movement. This simulates holding up a lantern and peering down various avenues of escape – at a certain point the light either illuminates a wall or disappears into the gloom.

Moves are made by pressing a key for a particular direction. If no wall obstructs, the player's token advances one cell and a new limited view is displayed. Time ticks on relentlessly whether the player is moving or thinking. Hitting a "panic button" reveals a quick glimpse of the whole maze, but at a high price – 500 time units.

After instructions are given, a seed number is typed in to start the game and feed a random number generator used for drawing the maze and placing the exit. Because the original maze generator results in a maze with only one possible path to the exit, I use the RND function to knock out some interior walls randomly to produce more pathways and more choices for the player. A greater value for the seed removes more walls.

Next the maze is generated, but in memory rather than on the screen. Starting and finishing locations are established, the player's token is moved to the start, and play begins with display of the first limited view. The start, determined in line 660, is at the center of the maze, and the exit is placed at a randomly selected point on either the left or right wall (lines 360-370). The updated score is given with each new limited view, and play continues until the exit is reached.

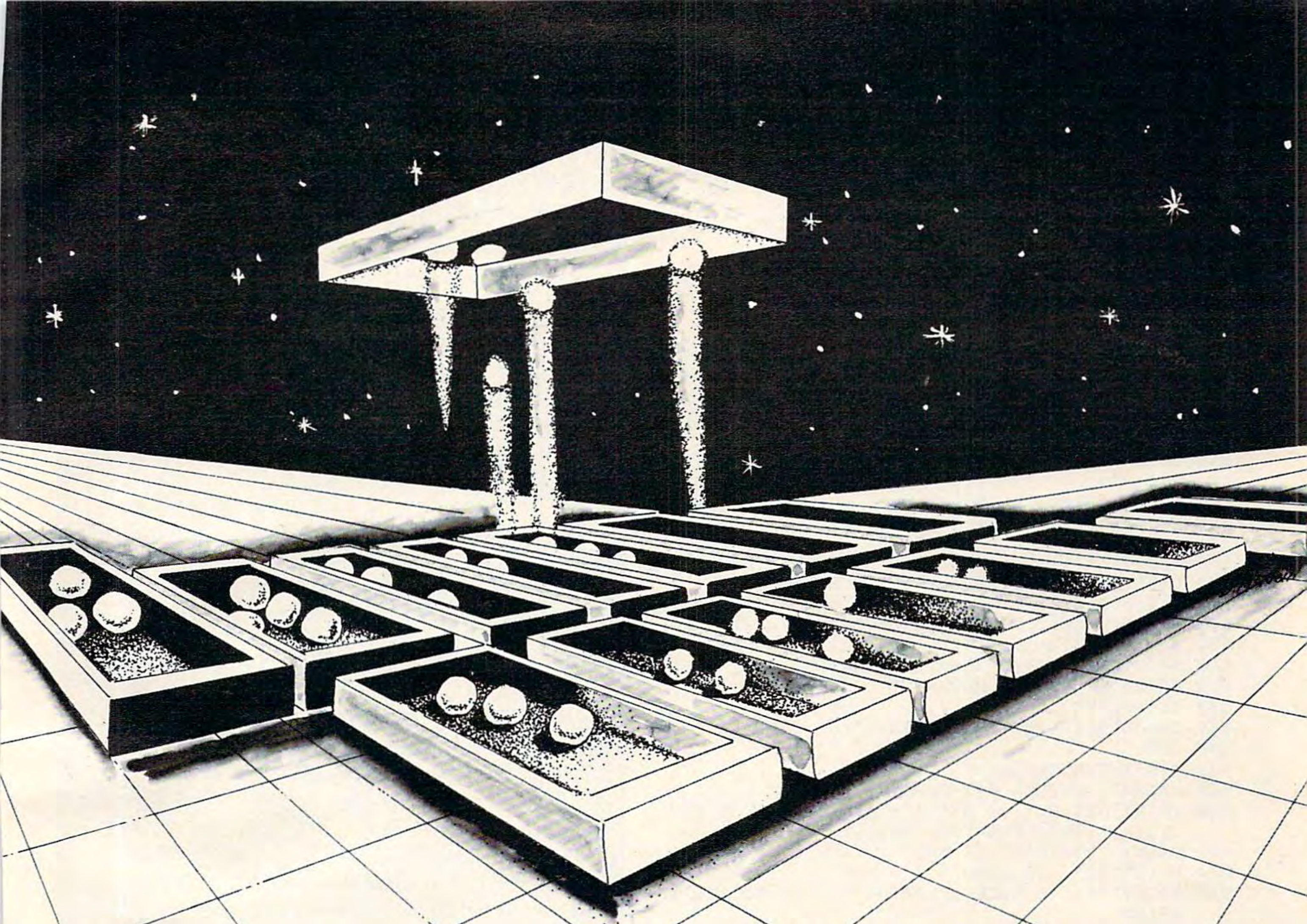
Three final notes: 1) The program requires about 30 seconds to set up the maze. To indicate all is working well, a POKE S2,J in lines 210 and 320 produces a rapidly changing character in the center of the screen. 2) The maze size given in this listing is 23 x 23 cells, but smaller sizes can be created by changing the values of H in line 605 according to this table:

7 x 7 cells	**H = 7**
11 x 11 cells	**H = 11**
15 x 15 cells	**H = 15**
19 x 19 cells	**H = 19**

Other sizes do not work. Centering of the whole view is done by line 650. 3) Several OSI-specific items were changed to convert the program for other machines. The video display on the C1P is 32 characters/line; in line 605 change VL to an appropriate value. Also, variables WL, HL, S2, and symbol numbers for the token (240 in line 160) and the exit (69 in lines 370, 440) were changed to something meaningful in each computer's graphic character set.

Program 1: PET Version

```
1Ø GOTO4ØØ
1ØØ REM-LIMITED VIEW
11Ø GOSUB73Ø:PRINTCT:FORJ=ØTO3:D=A:C=S2
12Ø FORI=1TO7:POKEC,M(D)
13Ø POKEC-E(J),M(D-D(J)):POKEC+E(J),M(D+D(J))
14Ø IFM(D)=WLTHEN16Ø
15Ø D=D+A(J)/2:C=C+E(J+1):NEXTI
16Ø NEXTJ:POKES2,24Ø:M(A)=24Ø:RETURN
2ØØ REM-LAYOUT FIELD
21Ø FORI=1TOH:FORJ=2TOH+1:POKES2,J
22Ø M((I*(H+2))+J)=WL:NEXTJ:NEXTI
3ØØ REM-GENERATE MAZE
31Ø M(A)=4
32Ø J=INT(RND(R)*4):Z=J:POKES2,J
33Ø B=A+A(J):IFM(B)=WLTHENM(B)=J:M(A+A(J)/2)=H
    L:A=B:GOTO32Ø
34Ø J=(J+1)*-(J<3):IFJ<>ZTHEN33Ø
35Ø J=M(A):M(A)=HL:IFJ<4THENA=A-A(J):GOTO32Ø
36Ø T1=(3*H)+5:T2=INT(RND(R)*2):IFINT(T2/2)*2=
    T2THENT1=(2*H)+6
362 Q1=-1:Q2=-(H+1):Q3=H+3
364 IFT1=(2*H)+6THENQ1=-Q1:Q2=-Q2:Q3=-Q3
366 Z=INT(RND(R)*(H-3))*(H+2)+T1:IFM(Z+Q1)<>HL
    THEN366
37Ø M(Z)=69:M(Z+Q2)=WL:M(Z-Q1)=WL:M(Z+Q3)=WL
38Ø FORI=1TOH
382 M(3*(H+2)+4+INT(RND(R)*(H-5))*(H+2)+INT(RN
    D(R)*(H-5)))=HL
384 NEXTI:RETURN
```



The Stone Age Meets The Space Age!

MANKALA, the ancient African stone game, is now available in a new fast-paced version for your personal computer. Challenge a friend or pit your skill against a formidable computer opponent in this fascinating game of tactics and strategy. Transport your stones to the safety of your home bin or move out for a daring capture. But beware! Just when you think you're safe, you may fall victim to a fiendishly clever raid by your opponent.

• Lively sound and colorful graphics • One or two players • Six different games • Four levels of difficulty plus matchups of the machine against itself • Available for Apple, Atari, and Commodore personal computers.

ATARI PRIZE WINNER

See your dealer or order now.

MANKALA disk or cassette $22.95 plus $2.00 for shipping. Virginia residents add 4% sales tax.

aldebaran INCORPORATED

Box 3469 • McLean, VA 22103
(703) 556-9747

Atari and Apple are registered trademarks of Apple Computer, Inc. and Atari, Inc., respectively.

```
400 REM
410 GOSUB600:GOSUB900:GOSUB730:GOSUB200:GOSUB1
    00
415 GETA$:IFA$=""THENCT=CT+1:GOTO415
420 J=-( (A$="8")+2*(A$="4")+3*(A$="2") )
425 IFA$="?"THEN500
435 A2=A+A(J)/2
440 IFM(A2)=69THEN800
445 IFM(A2)<>HLTHEN415
450 M(A)=HL:A=A2:GOSUB100:GOTO415
500 REM-DISPLAY WHOLE MAZE
510 GOSUB730:F=0:FORI=1TOH:FORJ=2TOH+1:L=(I*(H
    +2))+J
520 POKES+J+F,M(L):NEXTJ
530 F=F+FF:NEXTI
540 FORI=1TO200:NEXTI:CT=CT+500:GOSUB100:GOTO4
    15
600 REM-INITIALIZE VARIABLES
605 VL=40:H=23:FF=VL:REM  FOR 80 COLS., CHANGE
     VL TO 80.
610 A(0)=2:A(1)=-(H+2)*2:A(2)=-2:A(3)=(H+2)*2
620 D(0)=H+2:D(1)=1:D(2)=-(H+2):D(3)=-1
630 E(0)=VL:E(1)=1:E(2)=-VL:E(3)=-1:E(4)=VL
640 WL=160:HL=32:S2=32768+VL*12+INT(VL/2):CT=0
650 S=S2-VL*((H+1)/2)+FF-(H+3)/2
660 A=(H+2)*(H+1)/2+(H+3)/2:DIMM(630)
730 PRINT"{CLEAR}";:RETURN
800 REM-SCORE
810 GOSUB730:PRINT"YOUR SCORE=";CT:END
900 REM
910 REM-INSTRUCTIONS
920 PRINT"  HIDDEN MAZE":PRINT
930 PRINT"GET OUT AS QUICKLY AS":PRINT"YOU CAN
     USING":PRINT"CONTROL KEYS."
940 PRINT:PRINT"'8' IS UP,"
950 PRINT"'2'  IS DOWN,"
960 PRINT"'4' IS RIGHT,":PRINT"'6' IS RIGHT,":
    PRINT
970 PRINT"'?' SHOWS THE FULL MAZE":PRINT"BUT C
    OSTS POINTS.":PRINT
980 R=0:PRINT"PRESS {REV}RETURN"
985 GETA$:IFA$=""THEN985
990 RETURN
```

Program 2: OSI Version
Make these changes to Program 1.

```
415 K=PEEK(57100):CT=CT+1
420 J=-(((K=252)*0)OR((K=222)*1)OR((K=250)*2)O
    R((K=255)*3))
425 IFK=126THEN500
430 IFK=254THEN415
520 POKES+L+F,M(L):NEXTJ
605 VL=32:H=23:FF=VL-(H+2)
640 WL=187:HL=32:S2=53776:CT=0
710 POKE11,34:POKE12,2:POKE574,96
720 FORX=0TO27:Y=PEEK(65036+X):POKE546+X,Y:NEX
    TX
730 X=USR(X):RETURN
940 PRINT:PRINT"'ESC' IS UP,"
950 PRINT"'CTRL' IS DOWN,"
960 PRINT"LEFT & RIGHT USE SHIFTS,":PRINT
970 PRINT"'REPEAT' SHOWS THE FULL MAZE BUT COS
    TS POINTS.":PRINT
980 INPUT"ENTER SEED NUMBER       (1 TO 9)";R:
    IFR<1ORR>9THEN980
```

Program 3: Atari Version

```
100 REM  HIDDEN MAZE: ATARI VERSION
110 GRAPHICS 17:GOSUB 360:GOSUB 480
120 PPOS=SC+230
130 POKE PPOS,5
```

Atari Notes

Charles Brannon
Editorial Assistant

For the OSI and PET versions, the maze is constructed inside an array, rather than directly in screen memory, as with the original maze-generator. This is necessary to allow an "invisible maze" which only gradually opens up as the player travels.

With the Atari, we have another option. We can construct the maze directly on the screen (GRAPHICS 1 is used here, with custom characters for the walls and player). We make it invisible by setting its color equal to the background color (done here with SETCOLOR 2,0,0).

Then, to open up the maze, we just have to PEEK (into screen memory) the eight characters surrounding the player character, and if the PEEKed character is an "invisible wall," replace it with a visible wall.

Scoring is provided with RTCLOCK, Atari's realtime clock, which is found at locations 18,19, and 20. These are used in the opposite of the normal LSB/MSB order. Chaining all three locations together will give the current "jiffy time" since the machine was turned on, measured in sixtieths of a second:

JIFFY = PEEK(20) + PEEK(19)*256 + PEEK(18) *65536

Since location 18 only ticks every once in a long while, you can leave it out for most measurements. Dividing the jiffy time by 60 gives you the time in seconds:

SEC = (PEEK(20) + 256*PEEK(19))/60

Playing Hidden Maze

Use the joystick to move your ebullient little character around the maze, your goal being the upper-left-hand corner of the screen. The challenge is in how long it takes you to get there. You can take a "cheat peek" of the entire maze by pressing the fire button. This will display the maze for about three seconds, then turn to black and delay your movement for another three seconds as a penalty. If you want a really good score, don't use it!

```
140 DIM DIR(3)
150 DIR(0)=20:DIR(1)=21:DIR(2)=19:DIR
    (3)=1
160 POKE 20,0:POKE 19,0
170 FOR I=0 TO 3
180 ZP=PPOS+DIR(I):PK=PEEK(ZP):POKE Z
```

FIVE NEW GAMES FOR CASSETTE OR DISK FROM MERLIN ENTERPRISES

MERLIN

ATARI® apple® VIC-20®

LASER ARENA (16K)............. $14.95
Be a Laser Gladiator; Race a Laser Chariot; Battle your opponent or the computer in a fight to the death.

MIND LEXICA (16K).................. $9.95
A compendium of familiar and futuristic challenges for everyone to enjoy. Watch out . . . . may be addictive.

CODENAME: COBALT (48K)..... $24.95

This is the one you've been waiting for! Not a text game! Experience the ultimate adventure

in this 3-D, full-color, high-resolution future-world. Can you outwit the guardians of this behemoth labyrinthion: a hurtling, living COBALT BOMB! Find and arm the self-destruct before the Earth is vaporized . . . but can you escape with your life?!

ARTHUR DAVIS DETECTIVE ENCOUNTER SERIES — TWO FOR ONE SPECIAL $19.95
Scotland Yard is baffled! Can you as Arthur Davis

— famous Welsh detective — unravel these tangled webs of intrigue?

#3 Murder in the Manor (48K)
Does everyone have a motive?!

#4 Death for Dessert (48K)
Why would anyone want to murder Mr. Pym?

California residents add 6% Sales Tax.
UPS, C.O.D. Prepaid orders shipped FREE!
Send check or money order to:

MERLIN ENTERPRISES
P.O. BOX 2876, TORRANCE, CA 90509 213/316-0945

WUNDERWARE

Munchmaid 5K!, It's a Living!, Killer Caterpillar!, Bank Robbers!, Snake!, Blowup!, Space Rescue!, Galaxy Invasion!, Shutterbug!, Mad Painter!, Munchmaid!, Sector Protector!, Executioner!, Skiing!, Add'em Up!, Snailbait!, Barbarian's Quest!*, Motocross!, Copy-Cat!, Arithmevic 1, Arithmevic 2, Superracer! All for your **VIC-20**. Send for game catalog.

Killer Caterpillar! Here he comes...the dreaded Killer Caterpillar! He's weaving his way through the mushrooms trying to get to you. You can't let him through! If that isn't enough, you occasionally get visits from crazed spiders leaving a trail of mushrooms behind. Shoot them for extra points. Great graphics. For 5K VIC 20, requires joystick. Cassette $9.95, Disk $12.95

Mad Painter! This game is a little unique and a lot of fun. You control a paint brush, moving it around a colorful maze. Your job is to paint the entire maze. This is not as easy as it sounds, because in the maze with you are two voracious Bristle Biters (they love paint brushes). Occasionally you will receive a visit from an Invisible Stomper who leaves footprints in your fresh paint. Requires joystick. Cassette $9.95, Disk $12.95

Shutterbug! This game was designed for kids but adults will find it hard to wait for their turn at the shutter! You are a passenger on a tourist bus. You have a camera and a roll of film. The object is to take pictures of houses, trees, and horses. Don't waste film on telephone poles! Shutterbug! is easy to learn (only one key to push) and has very nice graphics to keep kids entertained for hours (teaches eye/hand coordination). Plays from keyboard or joystick. For 5K VIC 20. Cassette $9.95, Disk $12.95

Snake! A fast and fun action game for one player. You're a big snake roaming around the screen. Mice, rabbits, eggs, and feet appear at random. Your mission in life is to bite these targets. You have to be quick—the targets don't stay for long. The main problem is; you always seem to be running into the wall or into yourself (the longer you play, the longer, and harder to avoid your tail)! Snake! Keeps high score and requires a joystick. Cassette $9.95, Disk $12.95

Munchmaid 5K Due to popular demand, Munchmaid is here in a new 5K version for the unexpanded VIC 20. Fun to play with great graphics, of course! Munch dots, power dots, chase monsters and be chased. Munchmaid 5K! keeps high score and requires joystick. Cassette $10.95, Disk $13.95

Snailbait! Don't let the name fool you. This game is arcade action all the way! Your job is to protect a flower bed from an onslaught of killer snails. You are armed with a spraygun full of insecticide (environmentally approved, of course). As they zip across the field on their way to chomp down your flowers, the snails lay eggs which you'd better destroy, too! For one player, keeps high score and requires joystick, 5K VIC 20. Cassette $10.95, Disk $13.95

* Will be available soon.

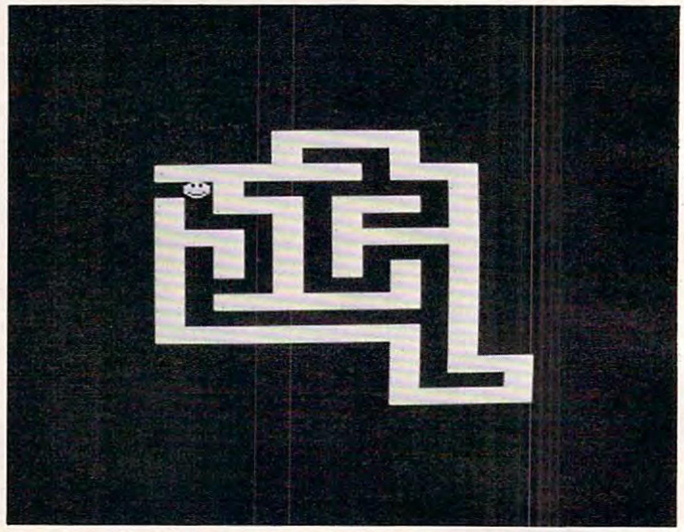
Send for information about the Commodore 64 programs.

■ Price includes postage & handling. ■ Catalog is included with order. ■ Foreign orders & COD's, please add $3.00. ■ Prices are subject to change without notice. ■ At your dealer or send check or money order to:
WUNDERWARE, P.O. Box 1287, Jacksonville, OR 97530 ☎503-899-7549. VIC-20 is a registered trademark of Commodore Business Machines.

```
     P,PK-64*(PK=129)
190  ZP=PPOS-DIR(I):PK=PEEK(ZP):POKE Z
     P,PK-64*(PK=129)
200  NEXT I
210  ST=STICK(0):TPOS=PPOS+20*(ST=13)-
     20*(ST=14)+(ST=7)-(ST=11)
220  CHR=3*(ST=11)+4*(ST=7)+5*(ST=14)+
     6*(ST=13)
230  IF STRIG(0)=0 THEN SETCOLOR 2,0,1
     4:FOR W=1 TO 500:NEXT W:SETCOLOR
     2,0,0:FOR W=1 TO 500:NEXT W
240  IF STRIG(0)=0 THEN 240
250  IF PEEK(TPOS) THEN 270
260  POKE PPOS,0:POKE TPOS,CHR:PPOS=TP
     OS
270  IF PPOS<>SC+21 THEN 170
280  FOR I=1 TO 50:FOR J=0 TO 3:POKE 7
     08+J,PEEK(53770):NEXT J:NEXT I
290  GRAPHICS 18:? #6;"you did it{P}"
300  SEC=INT((PEEK(20)+256*PEEK(19))/6
     0)
310  ? #6;"IN ";SEC;" SECONDS."
320  ? #6:? #6;"press FIRE to"
330  ? #6;"play again{N}"
340  IF STRIG(0) THEN POKE 711,PEEK(53
     770):GOTO 340
350  RUN
360  CHSET=(PEEK(106)-8)*256:FOR I=0 T
     O 7:POKE CHSET+I,0:NEXT I
370  RESTORE 410
380  READ A:IF A=-1 THEN RETURN
390  FOR J=0 TO 7:READ B:POKE CHSET+A*
     8+J,B:NEXT J
400  GOTO 380
410  DATA 3,56,124,174,174,254,186,68,
     56
420  DATA 4,56,124,234,234,254,186,68,
     56
430  DATA 5,56,84,214,254,254,186,68,5
     6
440  DATA 6,56,124,254,214,214,186,68,
     56
450  DATA 1,255,255,255,255,255,255,25
     5,255
470  DATA -1
480  GRAPHICS 17:POKE 756,CHSET/256
490  SC=PEEK(88)+256*PEEK(89):SETCOLOR
      2,0,0
500  DIM A(3):A(0)=2:A(1)=-40:A(2)=-2:
     A(3)=40:WL=129:HL=0:TRAP 32767
510  A=SC+21
520  FOR I=1 TO 21:? #6;"!!!!!!!!!!!!!!
     !!!!!!":NEXT I:POKE A,5
530  J=INT(RND(1)*4):X=J
540  B=A+A(J)
550  IF PEEK(B)=WL THEN POKE B,J+1:POK
     E A+A(J)/2,HL:A=B:GOTO 530
560  J=(J+1)*(J<3):IF J<>X THEN 540
570  J=PEEK(A):POKE A,HL:IF J<5 THEN A
     =A-A(J-1):GOTO 530
580  RETURN
```

Program 4: VIC-20 Version

```
100 REM HIDDEN MAZE: VIC-20 VERSION
110 PRINT"{CLEAR}";:GOSUB 360:GOSUB 480
120 PP=253
130 POKE SCR+PP,5:POKE CMEM+PP,2
140 DIM DIR(3)
150 DIR(0)=22:DIR(1)=23:DIR(2)=21:DIR(3)=1
160 T=TI
170 FOR I=0 TO 3
180 POKE CMEM+PP+DIR(I),5
190 POKE CMEM+PP-DIR(I),5
200 NEXT I
210 POKE37154,127:X=(NOTPEEK(37151))AND60-((PE
    EK(37152)AND128)=0):POKE37154,255
211 IFX=0THEN210
215 TP=PP-22*((XAND8)>0)+22*((XAND4)>0)-((XAND
    1)>0)+((XAND16)>0)
220 CHR=-(3*((XAND16)>0)+4*((XAND1)>0)+5*((XAN
    D4)>0)+6*((XAND8)>0))
230 IFCHR<3ORCHR>6THENCHR=5
240 IF(XAND32)THENPOKECC,8:FORW=1TO2000:NEXT:P
    OKECC,27:FORW=1TO2000:NEXT
250 IF PEEK(SC+TP)<>32 THEN 270
```



Hidden Maze – Atari and VIC-20 Version

VIC-20 Notes

The VIC-20 version of Hidden Maze will run on a standard 5K VIC. Use your joystick controller to move the smiling face around the maze, which gradually appears as you move about. Try to reach the upper left-hand corner of the maze as quickly as you can. You can press the fire button to see the entire maze for a few seconds, but you will be "paralyzed" for another few seconds as a penalty.

This game is a direct translation of the Atari version and, as such, is an illustration of some aspects of converting Atari programs to the VIC. The Atari GRAPHICS 1 screen is similar to the VIC-20 screen (20x24 vs. 22x23). Both machines store custom characters in the same format (but at different memory locations). POKEs can be used on both machines to manipulate redefined characters as "shapes," such as the face used in the VIC version.

```
26Ø POKE SCR+PP,32:POKE SCR+TP,CHR:POKE CMEM+T
    P,2:PP=TP
27Ø IF PP<>23 THEN 17Ø
28Ø FORI=1TO1ØØ:POKE CCTRL,255*RND(Ø):NEXT:POK
    ECCTRL,27
29Ø PRINT"{CLEAR}{REV}{PUR}YOU DID IT!":POKE36
    869,24Ø
3ØØ SEC=INT((TI-T)/6Ø)
31Ø PRINT"{GRN}IN";SEC;"SECONDS"
32Ø PRINT:PRINT"{CYN}PRESS {RED}{REV}SPACE{OFF
    OFF}{CYN} TO":PRINT"PLAY AGAIN.{BLU}"
34Ø GETA$:IFA$=""THEN34Ø
35Ø RUN
36Ø REM LOAD CHARACTER SET
365 CHSET=7168:POKE51,24Ø:POKE52,CH/256-1:POKE
    55,24Ø:POKE56,CH/256-1
37Ø FORI=ØTO7:POKECH+256+I,Ø:NEXT
38Ø READA:IFA=-1THENRETURN
39Ø FORJ=ØTO7:READB:POKECHSET+A*8+J,B:NEXTJ
4ØØ GOTO38Ø
41Ø DATA3,56,124,174,174,254,186,68,56
42Ø DATA4,56,124,234,234,254,186,68,56
43Ø DATA5,56,84,214,254,254,186,68,56
44Ø DATA 6,56,124,254,214,214,186,68,56
441 DATA 7,255,255,255,255,255,255,255,255
47Ø DATA -1
48Ø POKE36869,255
485 PRINT"{CLEAR}{22 DOWN}{REV}GENERATING MAZE
    {HOME}{OFF}";
49Ø SC=768Ø:CMEM=384ØØ:CCTRL=36879
5ØØ DIMA(3):A(Ø)=2:A(1)=-44:A(2)=-2:A(3)=44
51Ø A=SC+23:WL=7:HL=32
52Ø FORI=1TO21:PRINT"{WHT}GGGGGGGGGGGGGGGGGGGG"
    :NEXT:POKEA,5
53Ø J=INT(RND(1)*4):X=J:POKESC+5Ø5,J+128:POKEC
    M+5Ø5,8*RND(Ø)
54Ø B=A+A(J)
55Ø IFPEEK(B)=WLTHENPOKEB,J+1:POKEA+A(J)/2,HL:
    A=B:GOTO53Ø
56Ø J=-(J+1)*(J<3):IF J<>X THEN 54Ø
57Ø J=PEEK(A):POKEA,HL:IFJ<5THENA=A-A(J-1):GOT
    O53Ø
575 PRINT"{HOME}{22 DOWN}                    {HOME}
    ";:POKESC+5Ø5,32
58Ø RETURN
```

Program 5: Apple II Version

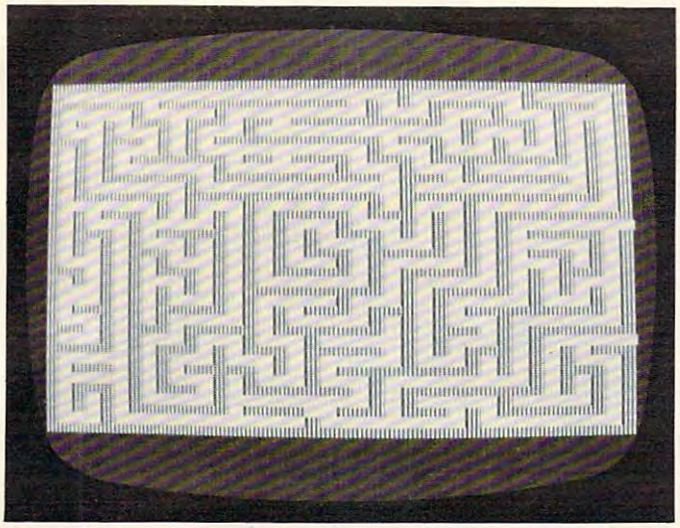
```
100  REM    HIDDEN MAZE:APPLE II
                        VERSION
110  HOME
120  GR : REM  GO INTO LO-RES MODE
130  HTAB 13: FLASH : PRINT "GENE
     RATING MAZE": NORMAL
140  REM  INITIALIZE VARIABLES
150 A(0) = 2:A(1) =  - 80:A(2) =
     - 2:A(3) = 80
160 WL = 8:HL = 7:CT = 0
170  DIM M(1680)
180 A = 859:L = 40
190  REM  GENERATE MAZE
200  FOR I = 2 TO 38: FOR J = 0 TO
     38
210 M(I * L + J) = WL: COLOR= J: PLOT
     L *  RND (1),L *  RND (1): NEXT
     : NEXT
220 M(A) = 4: GR
230 J =  INT ( RND (1) * 4):Z = J
240  COLOR= 16 *  RND (1)
250  PLOT L *  RND (1),L *  RND (1)
```

Apple II Notes

The Apple II version of Hidden Maze uses low-resolution graphics (40x40). The maze is generated inside a 40x40 array, and each part of the maze is displayed with the standard PLOT and COLOR commands. A good enhancement to the game would use page-flipping in the high-resolution mode (to quickly flash the completed maze), and a series of shapes for the player character.

Your player character is represented by a white square. Use the I, J, K, and M keys (I = up, M = down, J = left, and K = right) to move the square within the maze. Try to "escape" the maze by reaching the upper left-hand corner as quickly as possible. If you get stuck, press SPACE for a brief view of the entire maze (you will be charged 500 extra "time units" for this, however, and the screen will clear, erasing all the paths you've uncovered).

```
260 TT =  PEEK ( - 16336)
270 B = A + A(J): IF M(B) = WL THEN
     M(B) = J:M(A + A(J) / 2) = H
     L:A = B: GOTO 230
280 J = (J + 1) * (J < 3): IF J <
      > Z THEN 270
290 J = M(A):M(A) = HL: IF J < 4 THEN
     A = A - A(J): GOTO 230
300  GR :PX = 19:PY = 21: HOME
310 XD(0) = 1:YD(0) = 0:XD(1) = 0
     :YD(1) = 1:XD(2) =  - 1:YD(2
     ) = 1:XD(3) = 1:YD(3) = 1
320 D(0) = 1:D(1) = 40:D(2) = 39:
     D(3) = 41
330  REM  RANDOMLY KNOCK HOLES IN
      MAZE
340  FOR I = 1 TO 20:M((36 *  RND
     (1) + 2) * 40 + 38 *  RND (1
     )) = HL: NEXT
350 LOC = PX + PY * 40
360  FOR I = 0 TO 3
370  COLOR= M(LOC + D(I)): PLOT P
     X + XD(I),PY + YD(I)
380  COLOR= M(LOC - D(I)): PLOT P
     X - XD(I),PY - YD(I)
390  NEXT I
400  COLOR= 15: PLOT PX,PY
410 DIR =  PEEK ( - 16384): IF DI
     R < 128 THEN CT = CT + 1: GOTO
     410
420  POKE  - 16368,0:T =  PEEK ( -
     16336)
430 DIR = DIR - 128
440 TX = PX + (DIR = 75) - (DIR =
     74)
450 TY = PY + (DIR = 77) - (DIR = 73)
```



Hidden Maze – Apple Version

```
460  IF DIR = 32 THEN  GOSUB 620:
      GOTO 350
470  IF M(TX + TY * 40) <  > HL THEN
     410
480 M(TX + TY * 40) = 15:M(PX + P
     Y * 40) = HL
490  COLOR= HL: PLOT PX,PY: COLOR=
     15: PLOT TX,TY:PX = TX:PY = TY
500  IF (PX + PY * L) <  > 121 THEN
     350
510  FOR K = 1 TO 10
520  FOR I = 1 TO 10:A =  PEEK ( -
     16336): FOR W = 1 TO 10 - I:
      NEXT : NEXT
530  FOR W = 1 TO 50: NEXT : NEXT
540  TEXT : HOME : FLASH
550  FOR I = 1 TO 24: PRINT "
           ": REM  39 SPACES
560  NEXT : VTAB 11: HTAB 15: PRINT
     "YOU DID IT!"
570  FOR W = 1 TO 5000: NEXT
580  HOME : INVERSE
590  PRINT "YOUR SCORE:";CT
600  NORMAL
610  END
620  REM  DISPLAY WHOLE  MAZE
630  FOR I = 2 TO 39
640  FOR J = 0 TO 38
650  COLOR= M(I * L + J): PLOT J, I
660  NEXT J
670  NEXT I
680  FOR W = 1 TO 500: NEXT
690 CT = CT + 500: GR : RETURN
```

©

A Monthly Column

The World Inside The Computer

Fred D'Ignazio is a computer enthusiast and author of several books on computers for young people. He is presently working on two major projects: he is writing a series of books on how to create graphics-and-sound adventure games. He is also working on a computer mystery-and-adventure series for young people.
As the father of two young children, Fred has become concerned with introducing the computer to children as a wonderful tool rather than as a forbidding electronic device. His column appears monthly in **COMPUTE!**

Letters From Readers:

Software, Sexism, And Other Topics

Fred D'Ignazio
Associate Editor

I have received lots of mail from people who read this column. Most people write to compliment me and tell me I'm on the right track. But I also get letters that are critical. I welcome both types of letters. Please keep them coming.

Recently, I received a letter from Jan Murphy who wrote that she had been enjoying my column each month. Then she read the column on the computer friend (**COMPUTE!**, August 1982). On page 82, she read the following words:

Is this child a boy or a girl? The computer friend should know.

This line profoundly angered and upset Jan. Why? I'll let her tell you in her own words:

Why am I mad? I said to myself, 'How refreshing it would be to have a friend who didn't care *if you were a boy or a girl. And this computer friend idea would be a great chance to do that, but if everyone makes the computer friend* know *then that chance would be wasted.'*

Why *"should" the computer friend know the gender of a child? So the friend can treat the child in an "appropriate" way? How about letting the computer friend treat the child like a child instead?*

I don't go around saying all men are horrible, or get active in political things, or do many other things that people imagine "feminists" do. I reject the term feminist; I want human *liberation, freedom for both men and women to be who they are. So if my brother can't fix a car, who cares? That makes him bad at fixing cars, not "less masculine". And if I like computers, why shouldn't I? This is the kind of trap I see us all falling into when I read that the computer friend "should know" the child is a boy or a girl.*

It can't be coincidence that girls often do better in math in elementary school, yet by the time they are in the 12th grade, they can't cope with numbers. There were 3 out of 30 in my physics class, 3 girls and 27 boys. Why do you think I never learned which way to turn a screwdriver until one of my (male) friends taught me, in high school? Because "girls" don't get building toys for Christmas, that's why. Or telescopes, or chemistry sets, or tools. Why? There's no reason – it's just the way things have always been done, that's all, and we don't bother to cast off all the old baggage from the past when a better way of doing things turns up. (Perhaps I should say it this way: nobody likes to give up old software.) That's natural. But (as *usual) Ursula K. LeGuin has said it before, and said it better than I ever could:*

> *To oppose something is to maintain it. They say here that "all roads lead to Mishnory." To be sure, if you turn your back on Mishnory and walk away from it, you are still on the Mishnory road. To oppose vulgarity is inevitably to be vulgar. You must go somewhere else; you must have another goal; then you walk another road.*
>
> –The Left Hand Of Darkness
> *(Estraven, in chapter 11)*

I want our children to be people *first, and doers of great deeds, and makers of great works, creators and conservers and heroes; all these things first, and then, later, when they know who they are and that they can do great things, and that the world is full of wonderful things for them to learn, when they are firmly settled with a sense of their worth, then, they can also learn to be* boys *and* girls *(if it is still necessary) or* men *and* women. *But I want us to break the circle, as LeGuin would say, and go free. That's my goal. You want kids to be able to learn and grow, too, otherwise you wouldn't write such a neat column. Right?*

So when you make your computer friend with your child (and this goes for both your children) please *be careful and think about what you might be doing (uncosciously, I know) when you tell the computer friend things about your child. You've been pretty good so far, using "she" and "her" some of the time (I've noticed, yes, and I was impressed), so keep up the good work.*

Well, you wanted input on your idea for a computer friend/pet. I don't know if this is quite what you had in mind. But a computer named after the Archmage of Roke deserves the best, and it would break my heart to see him acting like the computer in a bad Heinlein novel.

Thanks for your columns (I loved the story about Eric and your floppy disks; I'm glad you didn't lose anything) – I'm looking forward to Catie and Eric's further adventures.

A Response To Jan's Letter

Jan's letter was thoughtful and thought-provoking. It angered me and upset me. It also convinced me that I had made a serious blunder.

I spent several days thinking about what Jan said and talking it over with my wife. I've come up with a tentative response.

First, I'm glad Jan wrote. The issue she raises is vital. The more "friendly" that software becomes – the more it acts like a person – the more it will carry hidden values. The question is: what are those values? Are they fair to all people? Or are they prejudiced and unfair?

Also, this issue assumes even greater importance now that young children are beginning to use personal computers. These children's values are largely unformed or, at least, extremely malleable. Values hidden inside computer-friend and computer-tutor software might be easily transmitted to young children. Again, it's important for us, as parents and teachers, to uncover these values and make sure they are similar to our own values.

In this specific case, however, I'm afraid that I disagree with Jan over whether the computer friend should ask the child's sex. I still think it should. It should for two reasons.

First, the child's sex is a biological fact. Second, the child's sex is an important, perhaps decisive, factor in determining how other people will treat the child. For good or bad, it is too big a factor for the child, or the friend, to ignore. Only by dealing with it can the child (and the friend) overcome it.

Back to reason number one: biology. For biological reasons alone, the child's sex is a central fact of the child's life. The child identifies herself or himself, in part, based on that fact.

I'm not saying what that fact means. I'm not saying that girls and boys shouldn't be free to express their personalities. I'm only saying it is an important fact and should not be covered up, rationalized, or denied. If I were a girl or a boy and I had a computer friend, I would want *my* friend to know my sex.

Second, unlike the computer friend, the child does not live in a vacuum. It lives in the real world. And the real world is filled with people who discriminate against women and men (in different ways, of course). Prejudice is built into the laws, into people's values and opinions, into institutions, and into almost every activity of our lives.

If the friend is to become a real friend, it must learn which side of the sexual fence the child is on. Only then will it be able to relate to the challenges the child will face in trying to overcome the injurious sexual stereotypes that pressure the child into a certain kind of behavior, career, style of life, or whatever.

This issue seems a little over-dramatized, given the extremely simple computer friend we have discussed so far. But a major trend in computer software is to anthropomorphize computers and make them more lifelike, human, and friendly. Computer friends in the near future won't be toy programs to amuse preschoolers. They will be built into silicon chips and be an intimate part of our daily lives – in the office, in the school, and in the home. Therefore, the type of values our "friends" should have is a good issue to be thinking about right now.

What Do You Think?

Now it is your turn, readers. What do you think about all this? Please write and tell me how you feel. Send your letters to:

Fred D'Ignazio
c/o **COMPUTE!**
P.O. Box 5406
Greensboro, NC 27403

A couple of months from now, I'll revive this issue and print some of the most thought-provoking letters I receive.

Computer Friends For Adults

A couple of weeks after I'd received Jan's letter, I got a letter from Irwin J. Davis of Bridgewater, New Jersey. In his letter, he proposed a computer friend for adults. Here is an excerpt from his interesting letter:

I read with interest your article about building a computer friend for a child. It did occur to me that the same concept could apply to adults. Why not build into the computer an adult personality like The Sage or Chief Mentor. The programmer could put in all his favorite sayings or aphorisms from secular or religious sources as Thoreau, Montaigne, The Bible, etc. The computer could suggest meditation exercises, relaxation techniques depending on how the person felt. In the past people would keep a journal and write sayings or thoughts of importance in them. Why not put them into a computer under certain categories and recall them for certain moods. Suggested types;

The Sage or Philosopher
The Psychologist
The Swami
The Man of Action
The Rabbi, Minister, or Priest

The programmer would have to know quite a bit about his character, which would be a good exercise for him.

What do you think about this adult computer friend? Write me if this letter has given you any ideas.

Computing In The Third World

I am tucked away in a nice little city in the U.S. This city has every kind of computer support system I could possibly want. But what would happen if I were a total novice who wanted to acquire a personal computer *and I lived somewhere in the Third World* (a developing country in Africa, Asia, or Latin America)?

This was the issue posed to me when I received a letter from J. J. Bichier, in Caracas, Venezuela. Bichier is a bush pilot and author. He wants to get a personal computer.

Here is his story:

I am a bush pilot-operator, out there in South America. Though the idea has been floating in my mind for a long time, a couple of months ago on a flight to Miami, I caught up with computers.

Flight plans, maintenance, operations, costs, losses and profits, all could nicely be automated and streamlined down to the meanest decimals, if I put together the proper hard- and software (within a reasonable budget) and learned how to use it.

To the good!

Besides the natural fascination for the technology itself, my main interest in computers lies in the fact that I am also an author.

When I think of the tedious time- and energy-consuming process of writing large books with paper, pen, typewriter, and dictionaries, my mind overflows with the reams of crumpled and unfinished versions I have to go through to get to the final copy. I am sixty and, besides the hard labor, there may not be that much time available. That's where an adequate word processor, proofreading attached, comes in.

When I think of the possibility of pouring schematics, material, partial or polished chapters into the box, with the ability to retrieve the text instantly, look at it, work on it and store it again to maturation of page, chapter, book, I drool.

I naturally surrounded myself with all the magazines and a couple of books I could find, haphazardly, to fill in the blanks. I went through them hungrily. I am learning PET/CBM BASIC and it doesn't seem that far off.

But all the ads do not tell the whole story. Venezuela, my country, lies thousands of miles and weeks away from the mainstream of marketing, support and maintenance infrastructures of any technology, computation to the fore. That has to be considered as well. Another fact is my total lack of experience with the equipment, technologies, and skills concerned.

For weeks I have sent letters to manufacturers, wholesalers, dealers and others, to make up my mind as to hardware, softwares, methods and prices. I thought their literature might fill some of the gaps. To my dismay, there was no feedback. Nothing flat.

Since no one in my surroundings is interested enough or possesses the necessary experience to help me, I am calling blind: I beg of you to do so, if you would and could find the time.

After reading all the ads and related articles, promising the "ultimate tool" for so many dollars less than their competitors and a lot of mulling over, I come up with the following system layout:

– Commodore 8032 (main unit)
– Z RAM board (summing 96K main working memory and Z 80 64K CPM compatible memory)
– Commodore 8250 2 mega floppy storage
– C Itoh F 10 daisy wheel silent printer
– Word Pro 5 +
– Compatible proof reader (unidentified)
– Small business management software I could easily adapt to airplanes (unidentified)

– Odds and cables
– Spare daisy wheels and ribbons
– Spare floppy disks
– All user and maintenance manuals for each piece of equipment
– Fast access to parts, boards, chips, bits and pieces
– Summing some $5000
– To be delivered at Fort Lauderdale Executive Airport, Florida, which is of easy reach for me.

Questions abound. Are the components wholly compatible with each other? How reliable and gremlin proof? Is the whole system compatible with my goal? Did I shoot short of the necessary memory to manipulate the makings of books? Etc.

My audacity may surpass my ignorance and you may have a good laugh.

There is no 100% proof reason for me to espouse the Commodore system rather than any other in its price range. It just seems to fit and for no valid reason at all I like it. My philosophy on the matter is that whichever system I end up with, my task will be hundreds of times easier, once I master its particularities, learn it inside out, and stick with it.

Another factor in favor of Commodore is that it is represented in Caracas, though it carries a 100% markup over stateside retail prices. There may be some support there. On the other hand, a son of mine is trouble shooting for Ohio Scientific in Venezuela. With the proper manuals in hand, there should be no fuss to keep going.

Still, the decision is intimidating.

Another interesting challenge I can come up with is this: I am trilingual, but do word processors and proofreaders exist for Spanish or French tongues? The answer to that might make of me another non-native English writer, though most of what I have to say is a lot tastier in its native Spanish.

This is my story. I hope your secretary will be kind enough to let it reach you – so you may decide to help.

Whichever happens, I shall be counting the days to thank you.

If you have any knowledge that would help Señor Bichier, please write him directly:

Cap J. J. Bichier
Apartado de Correo Este 60409
Caracas 1060
Venezuela
South America

Also, I would very much like to hear from readers who know what it is like to use personal computers outside the United States, particularly in the Third World. In a couple of months, I will touch on this subject again and print excerpts from some of the letters I receive.

Upcoming: Teacher And Pet

In two months, I'll return to my discussion of the computer friend. We'll make the friend capable of remembering things it learns from the child. Then we'll experiment with this feature by creating a "friendly" computer teacher and a "friendly" computer pet.

As you read the column and try the friend programs, please write me with your comments and send me copies of program enhancements you develop – on any of the popular machines. At the end of my discussion about computer friends, I will print the most helpful letters and listings. ©

High res graphics can be used for games, math equation plotting, light pen applications, drawing designs – any application where you want to turn on one dot on the TV screen. This article explores high resolution graphics on the 5K and extended 8K VIC-20.

Understanding VIC High Resolution Graphics

Roger N. Trendowski
Randolph, NJ

The VIC performs high res graphics through bit mapping the screen. Bit mapping is a method where each dot of resolution on the screen (called a *pixel*) is assigned its own bit in memory. If the bit is one, then the pixel is on; if zero, the pixel is off.

Your screen displays 506 alpha/numeric/graphic characters, 22 horizontal and 23 vertical. Since each character is made of 8x8 pixels, your screen consists of 32384 pixels. With high res graphics, you can selectively turn off or on each of these 32384 pixels – if you have enough memory (more about memory requirements later). Without enough memory, the X or horizontal coordinate may vary from 0 to 176, and Y from 0 to 184.

VIC Technique

Bit mapping is done on the VIC using the "programmable character" technique – when you POKE a screen location with a number from within that location. Try this on an unexpanded VIC: press the [RUN STOP] [RESTORE] keys, then type in:

```
POKE36879,62
POKE7690,0
```

This places a character display code of zero in the top middle of your screen (location 7690). An "@" character should appear. The first POKE turns the screen blue so that you can see the character. To display this character, VIC takes the display code and looks up the corresponding eight lines in ROM (Read Only Memory) starting with location 37768.

In the case of display code "0", the first eight bytes (memory locations) of ROM are used – 37768 through 37775. Each eight-bit byte in ROM defines a row of pixels which make up part of the "@" character. Now, if the display code "1" was POKEd instead of "0", an "A" would be displayed – it is stored in eight bytes of ROM starting at 32776.

The next step in understanding the bit mapping technique is to see how programmable characters are changed. Since the ROM area where the alpha/numeric/graphic characters are stored cannot be changed by a POKE command, we must change the VIC pointer from ROM to unused locations in RAM (Random Access Memory). To change this pointer, type in:

```
POKE36869,253
```

This memory location, which contains both the character memory pointer and a screen memory pointer, now points to RAM location 5120. The graphic garbage on your screen represents random data stored in the new eight-byte character RAM locations. Hit the [RUN STOP] [RESTORE] keys to clear the screen.

Try this short program which will show some of the fundamentals of high res graphics and bit mapping.

```
10 POKE36879,62
20 FORI=5120TO6143:POKEI,0:NEXT
30 POKE7680,0
40 POKE36869,253
50 POKE5120,1
60 GOTO50
```

Look at what has happened at the top left of the screen. A pixel has been turned on in the first row. Line 20 of the program cleared random data out of the RAM memory locations 5120-6143. Line 30 put a display character code of zero in 7680 (normally an @ character equals display code zero). Line 40 changed the character pointer from ROM to RAM location 5120. Line 50 created a new character in the first of eight bytes that define display character zero. The remaining seven bytes of display character zero (locations 5121 through 5127) remain cleared, meaning their bits are equal to zeros. Line 50 causes bit position 0 (right-most bit in the byte) to equal one. Line 60 causes VIC to remain in a loop so that the screen does not display "READY" and interrupt our demonstration. A conclusion from this exercise is that setting a bit to one in programmable character memory (e.g., 5120, bit #0) turns on a corresponding pixel.

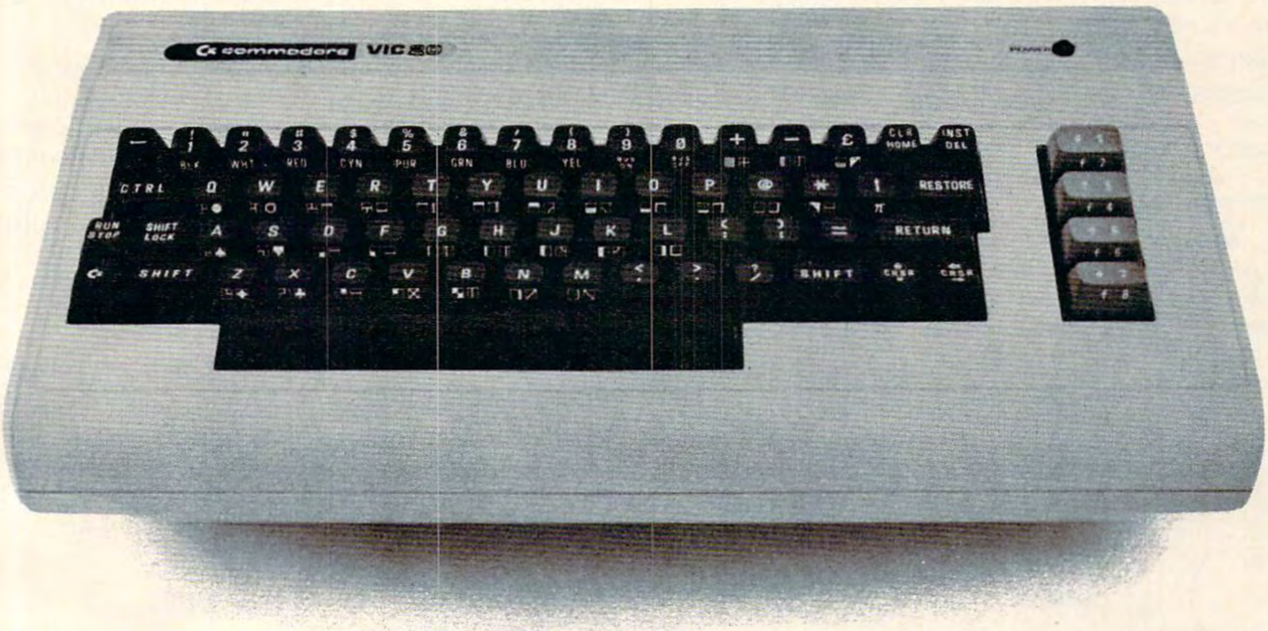
Try using binary word encoding with different values (0-255) in line 50 of the above program.

```
     Bit # 76543210
 Byte 5120 ^^^^^^^^
           00000001 = 1
           00000010 = 2
           10000000 = 128
```

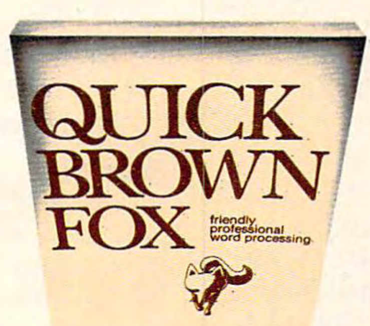
To expand your understanding, type the following change to the above program and run it:

TODD LERNER

WORD PROCESSOR?



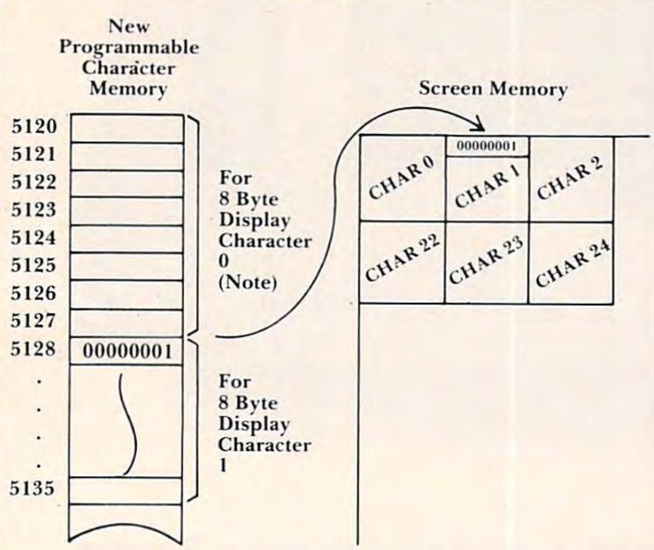
You bet! Quick Brown Fox word processing software has more features than Word Star and runs on even a standard VIC-20. And it can grow and grow. Add memory, 80 column display, disks, even a letter quality printer. We'll show you how a first-class word processor can be yours for less than $2000! It all starts with the Quick Brown Fox at $65.



Call or write for our free brochure
Quick Brown Fox 548 Broadway New York NY 10012 (800) 547-5995 Ext 194
Dealer Inquiries Invited (212) 925-8290

```
30 U=0:FORJ=7680TO7701:POKEJ,U:U=U+1:
   NEXT
50 POKE5128,1
```

The screen should show a pixel set in the 16th position from the left. Line 20 POKEd display codes of 0,1,2...21 into VIC's screen memory 7680 through 7701. Corresponding eight-byte blocks of RAM, starting with 5120, are cleared except for the bit 0 in byte 5128 – the top row of character number 1. Therefore, VIC turns on the corresponding screen pixel.



Note: When 8x16 character size is used, 16 bytes are used to define a display character on the screen.

Display Characters

If there are 506 character positions on the screen and only 256 possible display characters, then the question is: how do you fill up the rest of the screen? Use an obscure memory location – 36867, bit 0.

Type "NEW" and then type the following lines without line numbers:

```
POKE36879,62
POKE36867, (PEEK(36867)OR1
POKE7690,0
```

Among graphic garbage, two characters should have appeared at the top center of the screen: an "@" over an "A". The second line changed the VIC to a character matrix size of 8x16 (when bit 0 of this location equals 1). The VIC now uses the first 16 bytes to define display character 0. The third line POKEs display code zero into location 7690. In this way, by POKEing from 0 through 253 display codes on the screen, we can display all 506 character positions.

Memory Requirements

As mentioned earlier, bit mapping the entire screen would require 32384 pixels or 4048 bytes of RAM (32384 divided by eight bits per byte). With the original VIC-20, you have only 3583 bytes of BASIC RAM to work with for both the program and bit mapping. Therefore, you will have to limit the area of the screen you map. With a +3K or +8K memory expander cartridge, you can map a larger portion of the screen. It takes both the 3K and 8K expansions to bit map the entire screen.

When using an 8K expander, you must also perform some extra operations. A critical step will be to locate your high res program above screen memory and programmable character memory. I suggest location 8192, which is the first location in the 8K expander. The following 8K high resolution demonstration program will explain this technique.

X and Y Coordinate Calculations

Given that we now know how to turn a pixel off or on by changing a bit in programmable character memory (5120+), we still must have the program take an X or Y coordinate and translate it to the corresponding byte number and bit location. The following calculations must be made by the program:

```
CHAR = INT(X/8)*11+INT (Y/16)
```

This gives the display code of the character you want to change. Next, calculate the proper row in the character by using:

```
ROW = (Y/16-INT(Y/16))*16
```

From the CHAR# and ROW#, you can calculate the byte where X and Y lies.

```
Byte = 5120+16*CH+R0
```

The last calculation to be made identifies which bit must be changed.

```
Bit = 7-(X-(INT(X/8)*8))
```

To turn on any bit with the coordinates X,Y, use this formula:

```
POKE BY,PEEK (BY) OR (2↑BI)
```

Example

Program 1, for the unexpanded 5K VIC, bit maps approximately two-thirds of the screen and allows you to control pixel plotting with a joystick. The portion of the screen used for high res graphics is limited by your BASIC RAM area. Only 1022 bytes are left available for a BASIC program (locations 4096 to 5019). By changing the programmable character pointer from location 5120 to 6144 or 7168 (see Table 1), you make more bytes available for your BASIC program; therefore, there is less bit map area of the screen.

In Program 1, line 50 sets up parameters for joystick control and starting X and Y coordinates.

Line 60 colors the screen so that pixels will show. Line 70 clears all programmable character locations. Line 80 changes the VIC screen to an 8x16 character matrix size. Line 90 POKEs display codes zero through 153 in screen memory locations 7680 through 7832. If you insert an "END" statement between lines 90 and 100, you can see the display characters as taken from ROM. Line 100 changes the character pointer from ROM to RAM (location 5120). The screen clears to black because there are no programmable characters defined in 5120 to 7679.

The main program loop starts at line 110. This line points to the subroutine for reading the X and Y coordinates from the joystick. (If you want an explanation of this subroutine, look up David Malmberg's article in the fall 1981 issue of *Home and Educational Computing!*.) Lines 120 through 160 perform the necessary character (CH), row (RO), byte (BY), and bit (BI) calculations and operations to turn on a pixel. Warning: when you are playing with the demo program, don't go out of bounds or else you will invade other important memory locations. Strange things will appear!

Example Program For 8K Expanded VIC-20

This demonstration program will bit map approximately 75% of the screen, leaving 8192 bytes free for your application program. By the way, these 8192 bytes are all located in the 8K expander. The 75% limitation results from the VIC requirement that all screen memory and programmable character memory be resident in the VIC and not in the 8K RAM expander.

Before typing in or loading this program, type in the following:

POKE44,32
POKE642,32
POKE8192,0

These three POKEs are critical! The first and second commands place the new page number of where your BASIC program will be loaded into RAM. The page number is derived by dividing the intended starting address by 256 since there are 256 bytes per page in the VIC (8192/256=32). The third command zeros the first word of your BASIC program area – a must if you expect this thing to run. Now type in the program.

Except for a few lines, the explanation of this 8K program is the same as for the 5K demo program, except for three lines. Line 90 now contains the starting screen address of 4096 and character display codes up to 190. Line 100 POKEs a 205 into the character pointer to point to location 5120. This difference (253 VS 205) is due to the dual function that 36869 performs. Only the lower four bits of this location contain the character memory pointer. Line 295 is also changed. The Y represents the maximum Y coordinate you can turn on with the joystick.

Program 1.

```
10 REM ORIGINAL 5K VIC   EXAMPLE OF HIGH RES ~
     GRAPHICS
40 REM
50 DD=37154:P1=37151:P2=37152:X=10:Y=10
60 POKE36879,8:PRINT"{CLEAR}"
70 FORI=5120TO8185:POKEI,0:NEXT
80 POKE36867,PEEK(36867)OR1
90 FORI=0TO153:POKE7680+I,I:NEXTI
100 POKE36869,253
110 GOSUB200
120 CH=INT(X/8)+INT(Y/16)*22
130 RO=(Y/16-INT(Y/16))*16
140 BY=5120+16*CH+RO
150 BI=7-(X-(INT(X/8)*8))
160 POKEBY,PEEK(BY)OR(2^BI)
170 GOTO110
180 REM
200 POKEDD,127:P=PEEK(P2)AND128
210 J0=-(P=0)
220 POKEDD,255:P=PEEK(P1)
230 J1=-((PAND8)=0)
240 J2=-((PAND16)=0)
250 J3=-((PAND4)=0)
260 IFJ0=1THENX=X+1
270 IFJ2=1THENX=X-1
280 IFJ1=1THENY=Y+1
290 IFJ3=1THENY=Y-1
295 IFY>104THENY=104
300 RETURN
```

Program 2.

```
10 REM ORIGINAL 8K VIC   EXAMPLE OF HIGH RES ~
     GRAPHICS
40 REM
50 DD=37154:P1=37151:P2=37152:X=10:Y=10
60 POKE36879,8:PRINT"{CLEAR}"
70 FORI=5120TO8185:POKEI,0:NEXT
80 POKE36867,PEEK(36867)OR1
90 FORI=0TO190:POKE4096+I,I:NEXTI
100 POKE36869,205
110 GOSUB200
120 CH=INT(X/8)+INT(Y/16)*22
130 RO=(Y/16-INT(Y/16))*16
140 BY=5120+16*CH+RO
150 BI=7-(X-(INT(X/8)*8))
160 POKEBY,PEEK(BY)OR(2^BI)
170 GOTO110
180 REM
200 POKEDD,127:P=PEEK(P2)AND128
210 J0=-(P=0)
220 POKEDD,255:P=PEEK(P1)
230 J1=-((PAND8)=0)
240 J2=-((PAND16)=0)
250 J3=-((PAND4)=0)
260 IFJ0=1THENX=X+1
270 IFJ2=1THENX=X-1
280 IFJ1=1THENY=Y+1
290 IFJ3=1THENY=Y-1
295 IFY>143THENY=143
300 RETURN
```

Table 1. Important Memory Locations For High Res Graphics

5K (Unexpanded) VIC-20

7680	Start of screen memory
5120 or 6144 or 7168	Start of special RAM for programmable characters
63869	Pointer to character set RAM memory 253 for location 5120 254 for location 6144 255 for location 7168
36867	Sets 8x16 dot character size (Bit 0 = 1)

Table 2. VIC-20 With +8K Expander

43,44	Pointer to start of BASIC Program (Normally, 1,18; change to 1,32 for location 8193)
642,643	Pointer to start of BASIC Program (Normally, 0,18; change to 0,32 for location 8192)
5120 or 6144 or 7168	Start of special RAM for programmable characters
8192	First memory location of BASIC program area. Must be set to zero.
63869	Pointer to character set RAM memory, normally 192; must be set to: 205 for 5120 206 for 6144 207 for 7168
36867	Sets 8x16 dot character size (Bit 0 = 1)

©

For Apple Logo and Atari PILOT, this program provides a way to make the turtle draw the numerals from zero to nine. Using the techniques shown, you will be able to extend this method to include the alphabet as well. TI and Radio Shack Logo users can build a program from the examples given.

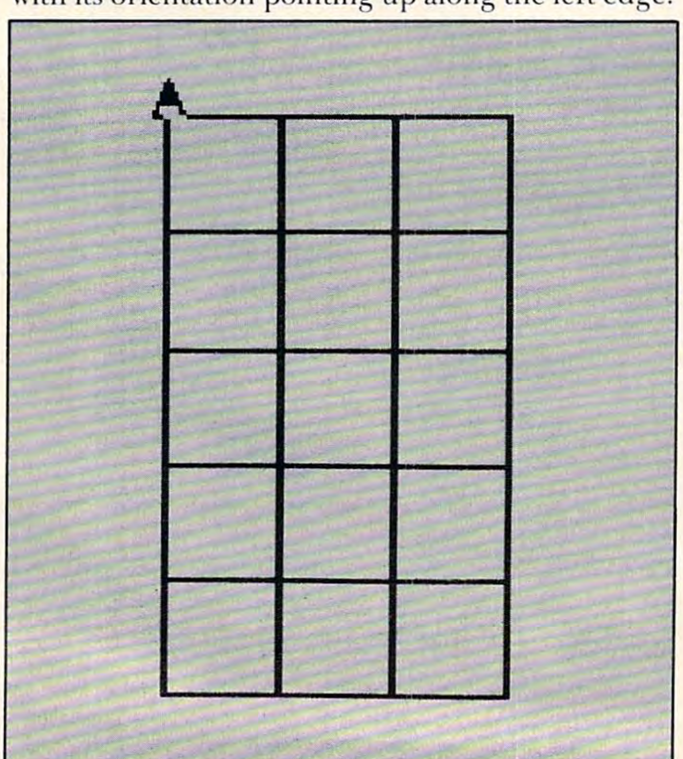
Making The Turtle Count

David D. Thornburg
Associate Editor

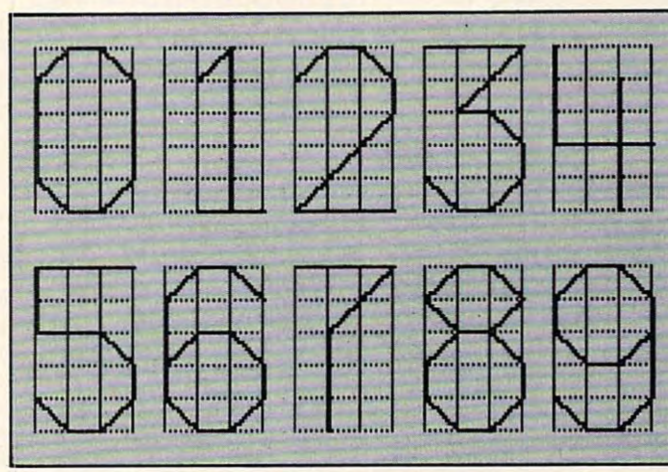
With the single exception of Apple SuperPILOT, none of the popular turtle graphics systems with which I am familiar allows the user to freely intermix text and graphics. One solution to this problem is to teach the turtle how to write!

If we are going to have the turtle draw numbers on the screen, we should pick a number drawing technique that lets us draw numbers of any size, orientation, location, and color we choose. The result will be a text display system that is more powerful than traditional dot matrix characters.

The character field I have chosen is three units wide and five units high. If the resultant characters are too high and skinny on your display, you will want to modify our method slightly to satisfy your own taste. The turtle starts and ends each character at the upper left corner of the grid, with its orientation pointing up along the left edge.



Using this grid we can design the numerals we want to draw, as shown below:



Each procedure for drawing consists of picking the turtle's pen up, moving the turtle to the starting position, putting the pen down, drawing the character in one continuous motion, picking the pen up, and moving the turtle back to its starting position and orientation. The shapes of the characters are defined so that each line segment is either along a grid length or along a grid diagonal. Since the length of the diagonal is larger than the grid length by the square root of two, our procedures need to incorporate this number.

This is fairly easy for the Apple Logos since they all use floating point arithmetic. Atari PILOT, TI Logo, and Radio Shack Color Logo, however, use only integer arithmetic. So, for these languages, we need to find a way to approximate the multiplication of a number by the square root of two. Obviously, we can't use the decimal number 1.414 because the language won't know what to do with it. Similarly, we can't just multiply by (1414/1000) because, if this division is performed first, the result will be one! But, if we first multiply the grid size by 1414 and then do the division by 1000, the result should be an effective approximation.

The following listings for the ten numeral procedures are shown in Apple Logo and Atari PILOT. Users of TI Logo, Radio Shack Color Logo, and other languages using integer arithmetic will have to mix and match from these two sets of procedures as needed.

```
Apple LOGO

TO ZERO :SIZE
MAKE "ROOT :SIZE * 1.41421
PENUP
BACK :SIZE
PENDOWN
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE * 3
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE * 3
PENUP
FORWARD :SIZE
PENDOWN
END

TO ONE :SIZE
MAKE "ROOT :SIZE * 1.41421
PENUP
BACK :SIZE RIGHT 90
FORWARD :SIZE LEFT 45
PENDOWN
FORWARD :ROOT
RIGHT 135 FORWARD :SIZE * 5
RIGHT 90 FORWARD :SIZE
BACK :SIZE * 2
PENUP
RIGHT 90 FORWARD :SIZE * 5
LEFT 90 FORWARD :SIZE * 3
RIGHT 90
PENDOWN
END

TO TWO :SIZE
MAKE "ROOT :SIZE * 1.41421
PENUP
BACK :SIZE
PENDOWN
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT * 3
LEFT 135 FORWARD :SIZE * 3
PENUP
LEFT 90 FORWARD :SIZE * 5
LEFT 90 FORWARD :SIZE * 3
RIGHT 90
PENDOWN
END

TO THREE :SIZE
MAKE "ROOT :SIZE * 1.41421
RIGHT 90 FORWARD :SIZE * 3
RIGHT 135 FORWARD :ROOT * 2
LEFT 135 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
PENUP
RIGHT 45 FORWARD :SIZE * 4
PENDOWN
END

TO FOUR :SIZE
MAKE "ROOT :SIZE * 1.41421
RIGHT 180 FORWARD :SIZE * 3
LEFT 90 FORWARD :SIZE * 3
BACK :SIZE
LEFT 90 FORWARD :SIZE * 2
```

```
Atari PILOT

*ZERO
C: #R=(#S*1414)/1000
GR: PENUP
GR: DRAW -#S
GR: PEN YELLOW
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S*3
GR: TURN 45: DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45: DRAW #R
GR: TURN 45; DRAW #S*3
GR: PENUP
GR: DRAW #S
GR: PEN YELLOW
E:

*ONE
C: #R=(#S*1414)/1000
GR: PENUP
GR: DRAW -#S; TURN 90
GR: DRAW #S; TURN -45
GR: PEN YELLOW
GR: DRAW #R
GR: TURN 135; DRAW #S*5
GR: TURN 90; DRAW #S
GR: DRAW -#S*2
GR: PENUP
GR: TURN 90; DRAW #S*5
GR: TURN -90; DRAW #S*3
GR: TURN 90
GR: PEN YELLOW
E:

*TWO
C: #R=(#S*1414)/1000
GR: PENUP
GR: DRAW -#S
GR: PEN YELLOW
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R*3
GR: TURN -135; DRAW #S*3
GR: PENUP
GR: TURN -90; DRAW #S*5
GR: TURN -90; DRAW #S*3
GR: RIGHT 90
GR: PEN YELLOW
E:

*THREE
C: #R=(#S*1414)/1000
GR: TURN 90; DRAW #S*3
GR: TURN 135; DRAW #R*2
GR: TURN -135; DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: PENUP
GR: TURN 45; DRAW #S*4
GR: PEN YELLOW
E:

*FOUR
C: #R=(#S*1414)/1000
GR: TURN 180; DRAW #S*3
GR: TURN -90; DRAW #S*3
GR: DRAW -#S
GR: TURN -90; DRAW #S*2
```

```
BACK :SIZE * 4
PENUP
FORWARD :SIZE * 5 LEFT 90
FORWARD :SIZE * 2 RIGHT 90
PENDOWN
END

TO FIVE :SIZE
MAKE "ROOT :SIZE * 1.41421
RIGHT 90 FORWARD :SIZE * 3
BACK :SIZE * 3
RIGHT 90 FORWARD :SIZE * 2
LEFT 90 FORWARD :SIZE * 2
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
PENUP
RIGHT 45 FORWARD :SIZE * 4
PENDOWN
END

TO SIX :SIZE
MAKE "ROOT :SIZE * 1.41421
PENUP
RIGHT 90 FORWARD :SIZE * 3
RIGHT 90 FORWARD :SIZE
RIGHT 135
PENDOWN
FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE * 3
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
PENUP
RIGHT 135 FORWARD :SIZE * 3
PENDOWN
END

TO SEVEN :SIZE
MAKE "ROOT :SIZE * 1.41421
RIGHT 90 FORWARD :SIZE * 3
RIGHT 135 FORWARD :ROOT * 2
LEFT 45 FORWARD :SIZE * 3
PENUP
RIGHT 180 FORWARD :SIZE * 5
LEFT 90 FORWARD :SIZE
RIGHT 90
PENDOWN
END

TO EIGHT :SIZE
MAKE "ROOT :SIZE * 1.41421
PENUP
RIGHT 90 FORWARD :SIZE
PENDOWN
FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 90 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
RIGHT 90 FORWARD :ROOT
PENUP
LEFT 135 FORWARD :SIZE
RIGHT 90
PENDOWN
END

TO NINE :SIZE
MAKE "ROOT :SIZE * 1.41421
PENUP
RIGHT 90 FORWARD :SIZE * 3
```

```
GR: DRAW -#S*4
GR: PENUP
GR: DRAW #S*5; TURN -90
GR: DRAW #S*2; TURN 90
GR: PEN YELLOW
E:

*FIVE
C: #R=(#S*1414)/1000
GR: TURN 90; DRAW #S*3
GR: DRAW -#S*3
GR: TURN 90; DRAW #S*2
GR: TURN -90; DRAW #S*2
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: PENUP
GR: TURN 45; DRAW #S*4
GR: PEN YELLOW
E:

*SIX
C: #R=(#S*1414)/1000
GR: PENUP
GR: TURN 90; DRAW #S*3
GR: TURN 90; DRAW #S
GR: TURN 135
GR: PEN YELLOW
GR: DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S*3
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: PENUP
GR: TURN 135; DRAW #S*3
GR: PEN YELLOW
E:

*SEVEN
C: #R=(#S*1414)/1000
GR: TURN 90; DRAW #S*3
GR: TURN 135; DRAW #R*2
GR: TURN -90; DRAW #S*3
GR: PENUP
GR: TURN 180; DRAW #S*5
GR: TURN -90; DRAW #S
GR: TURN 90
GR: PEN YELLOW
E:

*EIGHT
C: #R=(#S*1414)/1000
GR: PENUP
GR: TURN 90; DRAW #S
GR: PEN YELLOW
GR: DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 90; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN 45; DRAW #R
GR: TURN 90; DRAW #R
GR: PENUP
GR: TURN -135; DRAW #S
GR: RIGHT 90
GR: PEN YELLOW
E:

*NINE
C: #R=(#S*1414)/1000
GR: PENUP
GR: TURN 90; DRAW #S*3
```

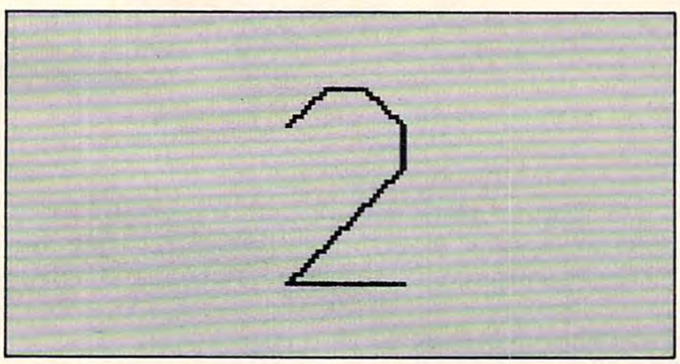
```
RIGHT 90 FORWARD :SIZE
RIGHT 135
PENDOWN
FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
LEFT 45 FORWARD :ROOT
LEFT 45 FORWARD :SIZE
RIGHT 180 FORWARD :SIZE * 3
RIGHT 45 FORWARD :ROOT
RIGHT 45 FORWARD :SIZE
RIGHT 45 FORWARD :ROOT
PENUP
RIGHT 45 FORWARD :SIZE * 4
PENDOWN
END
```

```
GR: TURN 90; DRAW #S
GR: TURN 135
GR: PEN YELLOW
GR: DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN -45; DRAW #R
GR: TURN -45; DRAW #S
GR: TURN 180; DRAW #S*3
GR: TURN 45; DRAW #R
GR: TURN 45; DRAW #S
GR: TURN 45; DRAW #R
GR: PENUP
GR: RIGHT 45; DRAW #S*4
GR: PEN YELLOW
E:
```

Now that these characters have been defined, it is easy to place a numeral anywhere you want on the graphics screen. For example, if (in LOGO) you enter:

CLEARSCREEN
HIDETURTLE
TWO 10

you will see the numeral 2 on the screen.



In Atari PILOT, the length of the grid unit is given by #S, so you must first enter:

C: #S=10
U: *TWO

to get this result.

But what about numbers longer than one digit? How does one print these? A LOGO procedure to print multiple digit numbers (using recursion) is shown below (you *have* been reading the "Friends of the Turtle" columns on recursion, haven't you?):

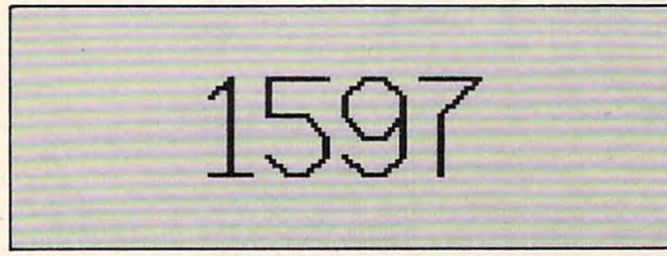
```
TO NUMB :LIST :SIZE
IF :LIST = [] [STOP]
RUN SENTENCE FIRST :LIST :SIZE
PENUP
RIGHT 90 FORWARD :SIZE * 4 LEFT 90
PENDOWN
NUMB BUTFIRST :LIST :SIZE
END
```

(Note: crafty Atari PILOT programmers will find at least two alternate ways to do this. At least one of these people will be kind enough to share the results with the rest of the readers.)

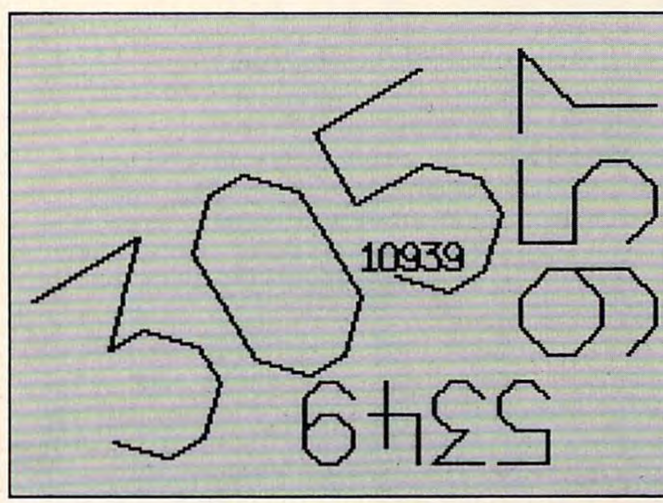
Now, with this procedure in hand, LOGO users should try entering something like:

CLEARSCREEN
NUMB [ONE FIVE NINE SEVEN] 5

to see what happens.



Experiment with different numbers, sizes, starting points, and orientations. You will find that you can print numbers at any angle. This is very handy for labeling graphs.



Expanding these ten numerals to the full alphabet is fairly straightforward. Any takers? ©

Review:

Supergraphics For PET

Elizabeth Deal
Malvern, PA

Supergraphics, written by John Fluharty, is a language extension for PET/CBM computers. Versions are available for Upgrade and 4.0 systems, 40 and 80 column. A ROM version (for $19000 or $A000 location) is currently available. The RAM versions are being discontinued.

First Impressions

Supergraphics seems to be a well thought out enlargement of the PET's resident BASIC interpreter. Its Turtle graphics, commands to plot lines in quarter-graphics mode, and general picture handling make it an ideal graphics package for children and adults who daily face the need to move spaceships around the screen.

The program does everything as described in the advertising and in the book. The book is clear and concise.

All commands work without a glitch. The mnemonics are well chosen, and there is no ambiguity. Kids can use the system and have, in fact, for over a year in various schools. Several similarities to the Radio Shack language permit children to switch between the PET and the other computer with little difficulty. Words such as CLS and HOME are understood by both languages. PRINT-at is a new concept for PET users, but is easy to grasp and quite efficient.

The housekeeping is fine. The PET is left in a relatively clean state during and after use, and even the memory locations used by such common utilities as the *Toolkit* and *Power* have not been clobbered (though some utilities might get disabled).

The demonstration programs are dazzling, though somewhat misleading. Some things are a bit more difficult to do than the demo would suggest. But then graphics are always tough. The package is well worth the money, and John Fluharty should be congratulated for enriching the PET's vocabulary.

Graphics Commands

There are commands to clear the screen, reverse it, place cursor home, and to list a program on a printer in program or direct mode. A dump of the screen to a printer is supported, but is not quite accurate (quotes are replaced by single quotes). You may switch text/graphic modes without POKE-ing. Screen images can be transferred to several adjacent alternate areas, permitting animation by quick transfers. The screen cannot be saved, but alternate areas can, so the effect is almost the same. This method is particularly useful to tape users. Saving is done through the monitor.

Quarter-graphics commands include setting and turning off points, drawing lines, drawing boxes, and filling them. Lines can be drawn in normal X-Y coordinates (0,0 in the upper left-hand corner) or in polar coordinates (0,0 in the center of the screen). The 80-column program supports 2:1 scaling of the X-axis. Lines and boxes drawn in quarter-graphics mode can be moved by the MOVE command. The motion can preserve whatever non-quarter-graphics characters are already on the screen. The unit of motion is quarter-graphic, that is, half a row or half a column at one time.

Normal size graphics commands include printing at specified coordinates, Radio Shack fashion (PRINT@col,row;"string"), defining a window for further operation, moving a window in four directions, filling one with a desired character, saving one in an alternate area, and bringing it back. Reversal of a window can't be done.

You may move anything you draw. You can put a spectacular spaceship on the screen using the PRINT@ command (or normal PRINT or POKEs), define its boundaries with CSET X,Y,X1,Y1 and zoom it around with words such as CMOVEU: CMOVEL. Diagonal motion is done by pairs, as in CMOVED:CMOVER. Motion is lightning-fast; you need PAUSE to keep it under control. More than one object can be moved "at the same time," but you'll need to keep track of the definitions, a process neither as easy nor as fast as it might seem.

The book provides little programming help in thinking graphics. Demonstration programs are hard coded with numbers, so you're on your own in the normal world of tedious graphics housekeeping. (Where are we, where are we going, what is there, what do we do if something is or isn't there, take it off, redraw, and back to start. Pheew!)

These block move commands get plenty of use. One-object motion is unquestionably splendid. Two or three objects – such as the background that wraps around or continuously scrolls left to right, and two competing spaceships controlled by users – get a bit sluggish. The reason is that you have to keep track of who is where at the moment, and you have to keep track of collision with another spaceship or walls of the screen. Even though it

takes only two or three extra lines of BASIC code to process the arrays of housekeeping definitions, it slows the process down, and the book confirms it.

I don't mean this note as a criticism of the program. None of the multipurpose graphics packages I have seen on the PET can handle motion of multiple objects or evaluate the situation at the edges of the screen. It requires tricky coding of tricky possibilities – a mind-reading program, *Wordpro* scope.

Turtle Graphics

Turtle graphics are included in the package: set and reset modes control the process. Work on the reversed screen is logically reversed. Additional commands position a turtle, set its direction, move pen up or down, and perform turns and units of forward motion. All Turtle commands work with quarter-graphics in polar coordinates.

Turtle graphics are a big hit in computer education. Children can learn programming by working with tangible things. We find this implementation nice and easy to use, but sometimes a bit abstract. Since the turtle is invisible, placing it on the screen and setting its direction provide no feedback until the turtle has moved forward. Should the turtle go over an existing line, it is again invisible until its direction is changed. A directional cursor might be helpful.

The turtle can accomplish some nice things, like drawing and rotating objects. The name of the game is learning geometry, and programming things such as rotation of objects should do the job.

I wish that Turtle graphics programs meant for small children limited out of bounds parameters. "Illegal quantity error" on a too large Y is a fact of life people must accept. But little users have enough trouble spelling words correctly; they could be helped by programs that avoided picture-destroying error messages. A no-action on the turtle's part would tell them they are wrong. I may be wrong: perhaps they *should* learn the hard facts of programming life right away.

Miscellaneous Commands

There are several other nice commands. For example, PAUSEX pauses execution for X jiffies. If a zero is given, a message prints "press space to continue" on the bottom line. This can be used instead of a GET loop. Pause is designed mainly to control the rate of animation.

An EXEC command in direct mode loads and runs a program. In program mode, it permits you to overlay a program longer than the calling program. Quite handy. The variables are cleared.

The OFF command turns *Supergraphics* off when you no longer need it, or when you write files from the machine language monitor or do several I/O commands to tape. *Supergraphics* turns its IRQ vector off for most I/O commands, hence it does not interfere – a nice and necessary touch.

The provision for repeating-key on all keys is useful in editing programs.

The SOUND commands are incredible. There are two versions. One is a simplified normal use of CB2 sound; the three POKEs have been squeezed into one command, "SOUND pitch, jiffies duration". The other is an elaborate system which can play music while the program is running or while you are editing the program. Once started, it will play on and on, until you turn it off with SOUND 0,0. A song maker provided in the book helps you include your own songs in a standardized manner.

User Extensions

We have seen that the IRQ routine has already taken a detour to repeat keys and play music. The IRQ routine can take another detour via a TASK command if you write a routine and tell the system where it is. This is valuable.

One more extension can be made in the IEEE vector: during IEEE processing the program checks to see if the user has his own wishes. Special routines can be added, such as a multi-user routine. One such routine has been implemented in a school system where the author teaches. I have not tried this command nor seen it in action.

The validity of the IRQ and the IEEE extension vectors is not checked. There is no extension of the CHRGET vector.

Documentation

The package includes a well-written, concise booklet. There are practically no muddy spots; all commands are explained clearly. The book does not say, for example, that the screen dump command forces paging on the printer.

Both the startup procedure and the various tips on using the system are unambiguous. A valuable set of hints is offered for speeding the processing.

Even though the demo programs show how to work the system, some graphics instructions might be more useful to kids who have never heard of X and Y coordinates. I am sure the schools will take care of it, though they will have to cope with the 0,0 in a funny place.

For programming types, one of the most valuable features is a listing of memory locations used by the program. This helps in understanding the system and permits you to use some values to advantage. You may wish to check the book; however, it seems to me that six more locations are zeroed than used, hence you should stay away from them.

Incidentally, the code is written in tiny, clear units. With Supermon's help, you can get at some little routines independently of *Supergraphics*.

Housekeeping

This section of my review deals with how the system is built, which has a bearing on how you use your PET. The discussion is not unique to this program; most programs of similar construction share these features.

Supergraphics adds some 35 commands to BASIC by intercepting the CHRGET routine. When you say SYS-supergraphics, you're asking PET to take a detour in its work to process the new commands. Unlike various editing utilities which are inactive during program execution, *Supergraphics* is *designed* to be enabled at all times. All commands are valid in program mode: they are interpreted, and, if needed, acted on, before BASIC gets a look at them. This slows BASIC down considerably – a do-nothing counting loop runs at a quarter of its normal speed.

PET needs time to process the new commands. Purely graphics programs, especially simpler ones of the type children write, don't suffer from the slow-down; in fact, it is unnoticeable. If you mix a lot of non-graphics commands, it is a good idea to use the OFF command frequently, for speed. The only time speed is a problem is in those calculations necessary to detect walls and collisions. At such times *Supergraphics* shouldn't be turned off, since turning it back on re-initializes all the working locations.

There are two things you should be aware of. First, a program written for *Supergraphics* obviously cannot run on a system that does not have *Supergraphics*. Users should be careful what they send to their friends, but this should cause no problem with its use in private or in schools. Second, while *Supergraphics* is enabled, any utility hooked up in any of the *first five bytes* of the CHRGET code is obviously disabled. You may have to cope with this in debugging. The current procedure is to do the OFF command before going SYS-utility. Use of OFF is mandatory: if you forget it, BASIC will not function.

Supergraphics
AB Computers
252 Bethlehem Pike
Colmar, PA 18915
$45 (school discount available)
specify which PET and desired chip location

©

Book Review:

TRS-80 Color Programs

Linton S. Chastain
Greensboro, NC

If you have a 4K or 16K Non-Extended or Extended BASIC Color Computer and a cassette recorder, you might be interested in a new book titled *TRS-80 Color Programs*, by Tom Rugg and Phil Feldman. The well-documented programs are useful to both the novice and the more advanced programmer. The book is not only a useful source of programs for the Color Computer, but also a teaching tool for beginning programmers.

The book has 37 programs, only nine of which will not run on a 4K machine. Divided into six sections, the book covers such topics as home and office applications, education, games, graphics, and math, and also has some short miscellaneous programs.

Section one contains eight programs – two new ones, and six modified for the Color Computer from the authors' previous book, *TRS-80 Programs*. Section two, with one new and six modified programs, deals with education. Section three, games, has one new and six modified programs, along with some color pictures of screen displays. Section four presents four modified graphics programs. Section five, math, has six modified programs. In section six are miscellaneous programs, one new and four modified.

The screen displays for the programs in the authors' new book are quite good. I have replaced some of my modified programs based on the earlier book with the new ones in the present book, primarily because of the enhanced screen displays.

TRS-80 Color Programs is a useful book for both the novice and the more advanced programmer. It goes beyond the example programs presented in the Color Computer manuals. In addition to helping you understand programming, the book also demonstrates some useful commands that help you shorten your programs and improve your screen displays.

TRS-80 Color Programs fortifies the authors' belief that most programs of similar language can be modified to run on other computers, and it helps to defuse the myth that the Color Computer is merely a toy or game computer.

TRS-80 Color Programs
dilithium Press
$19.95 paperback
332 pages

©

Review:

Apple Adventures

Dale Woolridge
Harrisburg, PA

Adventure games are older than Apple computers, and a high percentage of micro owners have played with them. These games give you a "world" containing dragons, demons, objects to be manipulated, etc. You use simple commands to move through the "world" and manipulate it.

Adventure – Colossal Cave

This is the original *Adventure* game, written first in FORTRAN for a PDP-10, by Willie Crowther and Don Woods. This program was implemented on the Apple by Master Jacobi. The program was compressed to fit entirely into 48K of RAM to avoid accesses to the disk.

Adventure has 15 treasures which add points to your score. It might not be obvious what a treasure is, so you might be tempted to pick up any object you find. There are 40 useful objects, but they have side effects. For example, the bird is afraid of the rod, and a certain magic word works only when you possess certain objects. The "world" is fairly large, containing 130 rooms. It is easy to find about a tenth of the rooms; the others are hard to find. In addition, there are 12 obstacles or opponents.

The game is complicated enough to keep you busy for a long time. If you are stumped, you can save the game to be resumed later. When you resume, you are asked if you want to load the saved game. If you say yes, you get back into the saved game, and the game is deleted from the disk. If you say no, you can start a new game while the saved game remains on the disk. You can save only one game.

Help, For A Price

A wizard, Arian, guides you through the world. A surprising, and amusing, feature of the game is that if you try many times to do a certain thing, but fail, the wizard will finally offer to help – for a price.

There is apparently a random element to the game. There is at least one situation in which you may or may not be killed, depending on chance.

The scoring scheme is somewhat unusual. You get points merely for discovering parts of the world and for finding objects. Getting killed costs you points. Your wizard might be able to bring you back to life, but you might lose the objects you were carrying.

The program is on a protected disk. The disk boots and the program loads in only nine seconds. At the beginning of the game a message appears briefly on the screen, and if you are a slow reader you might miss some of it. The message appears during the boot phase and disappears when the program executes. However, most of the program is well written and courteous to the user.

Adventureland

This Scott Adams' game has several features unusual in adventure games. The graphics were done using Penguin Software's Picture Editor, by Mark Pelczarski. The quality of the pictures is quite good. It takes 10-20 seconds, typically, to load a picture from the disk, and in case you don't have the time, the program lets you switch between graphics mode and all text mode. Often, a complete picture is "painted" on the screen, and then the disk drive comes on and certain objects are superimposed on the picture. This feature of the program gives you clues about the game, since the superimposed objects can generally move or be moved.

Use Peripherals

If you have a Votrax Type 'N Talk voice synthesizer, you can get the computer to speak the responses to your command. The responses will also be displayed on the screen.

If you have a lowercase adapter on your Apple, you can switch between all uppercase mode and upper/lowercase mode. And if you have a printer, you can get a hard copy of your adventure. The instruction booklet says that with some printer cards you might have to initialize the card in Applesoft before starting the adventure program. The Silentype printer does not require initialization before the game.

Another nice feature is that you can save up to four adventures to be resumed later. Considering that an adventure can occupy you for hours, this feature is desirable.

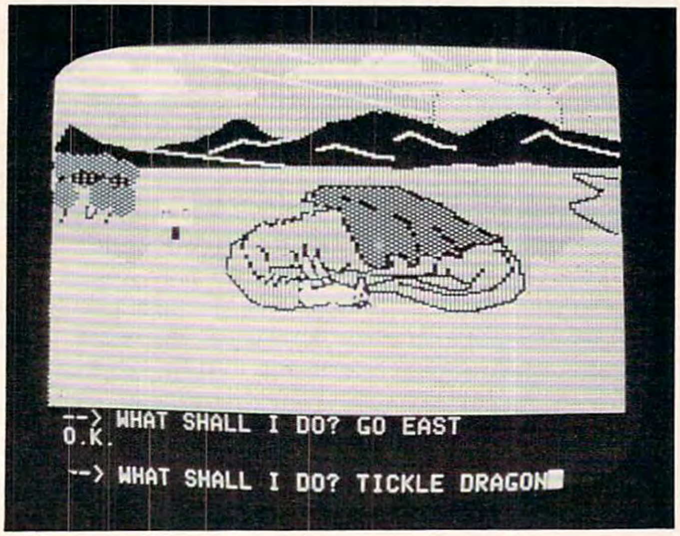
Before the game begins, you are invited to read an "open letter." The letter is a lecture on software piracy and includes several high resolution graphic pictures (of pirates, the American flag, etc.).

It is very important to have the proper mind-set when playing *Adventureland.* You must be able to tolerate some frustration, since you might get "stuck" in part of Adams' world. Also, you should realize that a game is not won in a few minutes of play; it might be complicated enough to keep you busy for weeks or months. Ideas may come to you while you are driving, and when you try them out that evening a whole new part of the world will be revealed to you.

The author's sense of humor is evident. He has apparently anticipated some of the commands you are likely to give and has prepared comebacks for you.

There is little randomness in *Adventureland.* As a rule, the same set of commands will have the same effects in different games. Success is obtained by using reason and common sense. However, there is an element of magic in the game; for example, you can come back to life if you give the right commands after being killed. There are also magic words.

It is very difficult to "crash" the program by giving bizarre input. It simply returns a message that it doesn't understand. Pressing RESET, however, will restart the game and clear out your adventure.



Teasing the dragon in Adventureland.

Taking inventory in Adventureland.

Adventure – Colossal Cave
Frontier Computing Inc.
P.O. Box 402
666 N. Main
Logan, UT 84321
$10 plus $1 shipping

Adventureland
Adventure International
507 East Street
Box 3435
Longwood, FL 32750
$29.95 disk

©

Review:

The VIC "Cardboard"

Harvey B. Herman
Associate Editor

Inserting a VIC cartridge is not a task for small and sometimes clumsy fingers. I have always insisted that my younger children call me when they want to change games. Thus, they are occasionally frustrated when I am not available for the task. The "Cardboard" promised to relieve this headache.

"What is it?" you ask. I believe the technical term is "motherboard." Its purpose is to extend, externally, the VIC expansion connector. All the pins on that connector are brought out by means of a ribbon cable to six exact duplicates of the VIC memory expansion port. You can plug in six cartridges, memory boards or games, and select any one of them easily with a dip switch. Yes, tiny fingers are ideal for this job, with no adult worries about mechanical damage to the VIC.

Next question, "Is it worth it?" The answer, "Yes and no."

Yes, because it enables little children to change applications easily. Also, it is solidly constructed and comes with an easy to understand, 18-page breezily written manual. It even has a reset switch which can extend the life of your VIC if you frequently turn it on and off to reset.

No, because it is relatively expensive (although cheaper than some) when compared with the VIC's original discounted price. Furthermore, the fact that it is not fused is bothersome. Can the VIC's power supply handle an indefinite number of plug-ins at the same time? I wonder.

On balance, I like this product and recommend it. I am using it with four or five popular games, and it has worked beautifully for the children. If you do buy it, keep a watchful eye out for power supply overheating or have someone knowledgeable fuse it for you. Then, enjoy the convenience.

Cardboard Expansion Interface
Cardco, Inc.
3135 Bayberry
Wichita, KS 67226
$119.95

©

Review:

Mikro Chip Assembler For PET

Arthur B. Hunkins
School of Music
University of N. Carolina at Greensboro
Greensboro, NC

Mikro, from Skyles Electric Works, is a 4K ROM assembler chip for PET/CBMs with Upgrade or 4.0 BASIC. It is not available for "Classic" PETs with original ROMs. Residing at hex address A000, Mikro is offered in a number of configurations that will otherwise accommodate virtually *all* PETs. For cassette-based systems, particularly those with limited memory, Mikro is a machine language programming boon, and well worth the $80.

Mikro is compatible with both Toolkit and Command-O. Indeed, a single SYS call initializes Mikro and the utility. All of Toolkit's commands are active while Mikro is running, and a number of them are applicable to machine language program development. One example: since Mikro uses BASIC line numbers, Toolkit's AUTO numbering command facilitates entering line numbers.

The user's manual for Mikro is both thorough and comprehensive. Although organized in a non-traditional manner that takes getting used to, its 49 pages contain a wealth of information. Included are sample programs, bibliography, installation and crash recovery procedures, a listing of the more than 15 error/warning messages with explanations, an overview of 6502 opcodes and addressing modes, and the few known bugs along with suggested remedies. The manual is *not* a treatise on 6502 machine language and its applications. The short, annotated bibliography will point you in the right direction, however. (Skyles recommends Leventhal's, DeJong's, and Zaks' books.)

Will Accept Four Number Bases

Since Micro operates with pseudo-BASIC statements (programs are SAVEd and LOADed as BASIC program files), PET's superior screen editing features are available to the user, in either LIST or Micro's FORMAT mode. Mikro's commands are: FORMAT, ASSEMBLE, and CONVERT (number base). The latter converts a number in decimal, hexadecimal, octal or binary to all the others. Incidentally, Mikro accepts numbers in any of these four bases!

Actual assembly of a short program is virtually instantaneous (hurrah for machine language assemblers!). Unless specified, assembly defaults to the second cassette buffer ($033A). Immediately following assembly, Mikro offers a partial or complete listing on a printer – the same listing as formatted input plus hex memory locations and their (hex) values. If you don't have a printer, you are out of luck here; Mikro will not print to the screen. I tested the print option with an Axiom EX-801 printer, and the operation went very smoothly. The only inconvenience was the fact that printer formatting (e.g., selecting 80 rather than the default 40 columns) must be done prior to assembly, by opening, formatting, then closing a file.

Once assembly has begun, Mikro is in control, and there is no way of interrupting it until after the printout. This can be more inconvenient than it might seem, because one of Mikro's "mites" is that during short printouts, it spews forth almost two extra pages of (often expensive) paper. The recommended fix is to turn off your printer. That effectively solves the immediate problem,.but also means that you must reformat your printer. Perhaps you will not experience this problem.

Includes Five Pseudo-ops And Append

As an assembler, Mikro is easy to use. On an 8K PET it reserves 1K at the top of memory for its own use; with 16 and 32K machines, it takes proportionately more. Syntax is standard, and the only crucial point to remember is that spaces are used as delimiters. A semicolon is required to

indicate a leading remark, and remarks are also possible at the end of lines. One problem the manual cautions the user against is spaces following commas in remarks. When I did it anyway, there were no bad consequences at all. Maybe I was just lucky.

Five pseudo-ops are implemented: 1) =, for label setting including *= for program origination, 2) TXT, for ASCII text within quotes, 3) BYT, 4) WOR, and 5) END (optional). A special application of END involves appending (or merging) a BASIC program onto the end of one in machine language. Following assembly of the ML program (up to END), the appended BASIC program can be run by commanding RUNxxxx or GOTOxxxx, where xxxx is the first line number of the BASIC program.

No comments are allowed following BYT or WOR, nor are spaces permitted at the commas in the list of values. Although all values are assembled, only the first three appear in any listing. A useful variety of arithmetic operators and labels is allowed in the argument field.

One of Mikro's handiest features is a GO option for JMP and branching statements. For example, JMP GO20 is a valid statement meaning jump to the instruction in BASIC line #20. These branches are also automatically handled by Toolkit's RENUMBER command.

Mikro represents an excellent, cost-effective investment for Upgrade and 4.0 PET (and CBM) owners wishing to do small to moderate amounts of machine language programming. I particularly recommend it for PETs that are cassette-based and have limited amounts of memory (such as 8K).

Mikro Chip Assembler
Skyles Electric Works
231E South Whisman Rd.
Mountain View, CA 94041
(415)965-1735
$80

©

Review:

Epson Graftrax-80

Charles Brannon
Editorial Assistant

Graftrax-80 is a ROM upgrade for the Epson MX-80 printer. Epson introduced their MX-80, a small, fast, relatively quiet 80 character printer at under $800. They packed it with more intelligence than some of the computers using it. Among its features are: two character widths (80 and 132); elongated, double-strike, and emphasized printing; horizontal and vertical tabs; and definable form length and line spacing. The standard MX-80 also provides block graphics (compatible with the TRS-80) that can be used for low resolution screen dumps, pictures, charts, and graphs.

Epson announced that a $100 upgrade could be made to the MX-80 to provide graphics capabilities. And it would provide graphics *twice* as dense as the MX-70 (MX-80's lower priced relative).

Installation

Upgrading your MX-80 is easy, if you know how to remove and install IC's. Otherwise, you should have it installed by an authorized technician. The upgrade consists of three ROM chips that replace a single ROM resident on the board. With three times the memory, this should give you a hint of the potential of Graftrax. You also have to cut a jumper and set 12 tiny DIP switches.

After you have performed this surgery, what do you get? Well, prepare for a surprise – this transplant does more than add graphics – you've got a whole new printer!

Graftrax-80 adds a plethora of new features, and improves on others. All the modes can be mixed on a single line, a trick formerly impossible. The duration of the bell has been reduced from three seconds to a bearable 1/3 second. A backspace function permits underlining (but it's slow).

A popular new feature is the alternate character set – italics. This looks quite fancy. You can easily mix the italics font with standard text. You can now go into the TRS-80 mode via software (formerly you had to set a DIP switch). You can set

Figure 1. Graftrax-80 Character Sets

Default character set:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}
```

Emphasized printing:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}
```

Double-strike printing:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}
```

Double-width characters:

```
ABCDEFGHIJKL
```

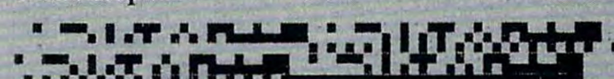
132 characters/line:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}
```

Italics character set:

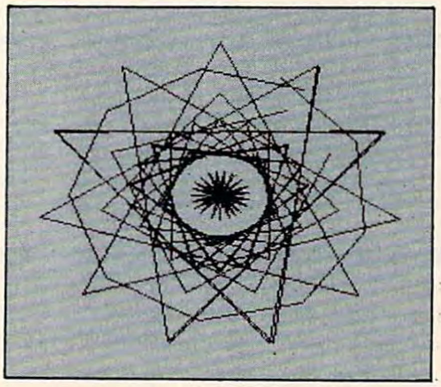
```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}
```

Block Graphics:



```
EMPHASIZED DOUBLE STRIKE
132 C/L, DOUBLE-WIDTH
MIX CONDENSED AND Italics
```

Figure 2. Graftrax Graphics



an "MSB mode" that will force bit seven high (for sending a character greater than ASCII 128). This is useful for computers and interfaces that can only send seven bits per byte (such as the Apple II).

One of the most significant new functions is the ability to redefine all the printer codes. You can change almost any of the special codes into any code you like. For example, double-strike is set with ESC-G (ASCII 27 followed by ASCII 71). You could change this to ESC-D (easier to remember), but you would be replacing the "Set horizontal tab" command which is normally keyed to ESC-D. One possibility of this feature is that you could change the MX-80's special codes to approximate the codes of, say, the Centronics 737. You could then run software written for the 737 without modification.

Extraordinary Graphics

The graphics capabilities are superb – up to 120 dots per inch. This permits a total horizontal width of up to 960 dots. This is more resolution than most computers can display, so it is more than adequate for screen dumps. The 480 mode (480 dots per line) is faster than the 960 mode, and it is usually used for screen dumps. The graphics are fairly easy to use: you send a code specifying which mode, and how many bytes of graphics you are sending. Then, a byte at a time is sent from the computer that specifies each bit of the eight dot (vertical) line. For example, to print a special character, ten bytes would be sent.

```
7    000000    128
6  0        0  64
5 0  0000    0 32
4 0 0        0 16
3 0 0        0 8
2 0  0000    0 4
1  0        0  2
0    000000    1
  1234567891
           0
```

The copyright symbol

The printhead is a strip with nine tiny needles set into it. Each needle is activated by a "1" bit, or left seated with a zero. Unfortunately, the ninth pin can't be fired because there are only eight bits in a byte. The first byte sent would look like: 00111100 (turned on its side). In this way, an 8x*n* "strip" of dots would be printed.

Speed

With 480 dots per line, using Graftrax from BASIC is unbearably slow, since 480 bytes have to be individually calculated and sent, one at a time. You would probably want a machine language program to do the printing.

Graftrax Plus

A new version of Graftrax, called Graftrax Plus, is now available for $65. It improves and expands upon the already enhanced features of Graftrax. In addition, owners of the MX-100 (which already has Graftrax) can upgrade to Graftrax Plus and enjoy compatibility with the MX-80 equipped with Graftrax Plus.

In addition to the italics character set (missing on MX-100 Graftrax), Graftrax Plus adds several features, including: superscript and subscript (the printer doesn't really adjust the paper; it just uses tiny half-height characters), improved graphics, and true underlining, with underlining on/off commands.

Graftrax-80
Epson America, Inc.
3415 Kawisha St.
Torrance, CA 90505
$99.00

©

A Monthly Column

Computers And Society

David D. Thornburg
Associate Editor

Inspector Fenwick – Please!

Hey, Rocky, I think I just saw the girl of my dreams!

Gee, Bullwinkle, where's that?

At the Moose America Pageant – where else?

A Saturday morning replay of *Rocky and His Friends*? No, this is a sample of the dialog that greets visitors to one of the newer purveyors of food and entertainment – Bullwinkle's. What does this have to do with the social impact of computers? Read on, dear readers, read on.

It all began in the 1950s when Walt Disney and his designers concocted Audio Animatronics, an analog-based control system that gave motion to the mannequins in such Disneyland favorites as the Enchanted Tiki Room. This technology was further advanced by the Disney group to make such shows as Pirates of the Caribbean, Country Bear Jamboree, and America Sings. The result was the creation of remarkably lifelike animated stage shows using automatons. In the hands of Disney designers, the result was magical.

Dining With Computers

A few years ago, Nolan Bushnell (founder of Atari and godfather to a host of innovative companies) developed Cyberamics to bring animated characters into a combined arcade/restaurant – Chuck E. Cheese's Pizza Time Theater. Visitors to this establishment are treated to various shows, including Dolly Dimples, a delightful animated hippopotamus night club singer, and, in another part of the restaurant, Chuck E. Cheese and his cohorts, who provide their own brand of cornball entertainment to go with the pizza. Central to Pizza Time is the arcade room, filled to the brim with a great diversity of video games, each operated with tokens marked "In Pizza We Trust."

The success of this technology-based restaurant has been phenomenal, and it was clear from the start that others would soon develop their own version of this concept.

Next enters David Brown, developer of two Marriott's Great America theme parks and the Roy Rogers' Family Restaurant chain. David thought that Pizza Time was a great concept, but that the food quality could be improved. Brown's idea was to create a place that was a restaurant first, but which incorporated entertainment in the dining area and a separate game room with a modest assortment of popular arcade games. This idea became Bullwinkle's.

As luck would have it, the world's first Bullwinkle's was constructed only a few miles from my humble abode. In the interest of keeping my readers abreast of the latest in technology, I had to visit Bullwinkle's many times, consuming vast quantities of chicken and pizza and ice cream, watching shows, and playing myriad games.

During one such visit it was my pleasure to meet their marketing maven, Larry Schuller. As he showed me around and answered my questions, it was clear that computer technology plays a critical role in this restaurant.

First, the animated characters themselves are controlled by Moosetronics, a set of distributed processors running off an S-100 bus. The song and dance routines are stored on both tape (audio and synchronization) and disk (for various body movements). Some of the characters are quite elaborate. Bullwinkle, for example, is about six feet tall. His eyes, mouth, head, arms, and legs all move in fairly realistic fashion (realistic for a moose based on a cartoon character), and this attention to detail characterizes several of the other eleven animals as well. The attention to detail includes placing the loudspeakers in each figure so the sound comes from each animal as it is singing or talking.

Fantasy Fountain

If this elaborate production weren't enough, visitors are also treated to a computer-controlled fantasy fountain show in which 250 jets propel 300 gallons of water in a dazzling array of arcs and spirals. All this takes place under colored lights in accompaniment to such melodies as *The Blue Danube* and *Raindrops Keep Fallin' on My Head*. This water show, more than anything else, appears to be the prime attraction to the over-30 crowd (your esteemed author included).

But the computers don't stop here. The system which notifies people when their order is ready is none other than a trusty Apple II located near the

kitchen. Monitors scattered throughout the restaurant show which orders are ready, and each new order is announced by a high resolution image of Bullwinkle holding up the new number.

The professionalism in their mechanical characters is reminiscent of Disney's Country Bear Jamboree, and for good reason. It was designed by a collection of Disney graduates who now ply their craft for others.

Aside from pure money, what motivated the people at Bullwinkle's to create this restaurant? According to Larry Schuller, microcomputer-based entertainment *belongs* in restaurants. The provision of electronic fun to go with the food is perhaps the next stage in the evolution of family dining.

Interestingly, the arcade seems to be almost an afterthought at Bullwinkle's. Off away from the eating area, 50 games provide entertainment for patrons who, in my opinion, show much greater care for the machines than I am used to seeing. While I was unable to get the exact figures, I found that Bullwinkle's derives a considerably smaller fraction of its revenues from the games than does Pizza Time Theater. That doesn't bother Bullwinkle's at all. As Schuller says, they are aware of the continuing controversy surrounding these games. When will the controversy go away? In Schuller's mind, the controversy surrounding these games will go away when the games become more educational.

The Next Step

There is no question that arcade games can be made more educational – Children's Television Workshop has shown that. But just as Bullwinkle's feels that it has improved the electronic entertainment/restaurant idea of Nolan Bushnell, they also feel that they can someday make improvements in the design of the arcade games themselves.

As nice as such improvements might be, they are not their first order of business. The next step is to carefully locate the next several restaurants. Not surprisingly, their first announced franchise was for 13 restaurants in Canada, with the first to open in Edmonton in March. Dudley Do Right of the Royal Canadian Mounted Police has been a popular character there for years, so the success of this expansion venture is virtually guaranteed.

This doesn' t mean that the United States has been ignored. Twenty-nine restaurants are scheduled for construction here in 1983, 20% of which will be company owned. In addition, negotiations are underway to share this technology with the United Kingdom. One has to be "moost" impressed with this expansion plan, especially since each restaurant costs well over a million dollars to set up.

As the water show comes to a close, and the curtain falls on Dudley Do Right, one must wonder what computer pioneers like John von Neuman would have thought. Computer technology has advanced extraordinarily in the past 30 years. Have its applications advanced as well?

Boris, if I hear one more moose joke I will blow up the stage!

Natasha, darling, that would be moost *devious of you!*

©

A Monthly Column

COMPUTE! *welcomes Keith Falkner, whose "Extrapolations" column begins this month. Keith, who has extensive experience at all levels of computing, has contributed several excellent Apple articles to* **COMPUTE!** *in the past. To start his monthly column, he demonstrates how to use a simple BRUN to bring in the power of the renumber program – without affecting the program in memory. There's also a way to make yourself a simple assembler if you don't have the Mini-assembler.*

Extrapolations

Beat The "Applesoft Renumber" Blues

Keith Falkner, Toronto

On your System Master diskette there is a very powerful utility program called *Renumber*. This program can merge two Applesoft programs and can move several lines from one place to another within an Applesoft program. Of course, Renumber will also renumber the lines of an Applesoft program, and the options it offers in this function are as complete as anyone could wish.

Furthermore, Renumber is cleverly packaged as an Applesoft program so that no complicated machine language instructions are needed to run it.

Protecting Memory

When you run Renumber, a hidden machine language component relocates itself to the top 2048 bytes of memory, prevents Applesoft from overwriting it, and enables the ampersand (&) command. Thereafter you can LOAD, RUN, SAVE, etc., as usual, and the ampersand command invokes one of the three functions of Renumber. This is very clever packaging, because this way only one version of Renumber is needed for 32K or 48K Apples, regardless of the current upper limit of memory.

Setting MAXFILES or running the utility known as Program Line Editor both alter the upper limit of memory, but Renumber does not care. This versatility is commendable, but it comes at a price. If you have not bothered to run Renumber, but are working on an Applesoft program and wish to renumber it, you must first SAVE it, then run Renumber, then reload your Applesoft program. Generally, you do not need this flexibility. For example, if you have a 48K Apple, the machine language component eventually resides in locations $8E00-$95FF (36352-38399).

I'll show you how to save this machine language routine, together with a prologue to do the minimum initialization. Then a simple BRUN command will activate the essence of the Renumber program, without affecting any Applesoft program in memory. At the same time we will deal with the more or less well-known bug. If the program being renumbered contains a multiplication by a constant, such as X1 = J * 100, and there is a line number 100 which becomes, say, line number 80 upon renumbering, the constant may become 80 as well.

This is a consequence of the clever relocation routine which makes the machine language code function in whatever memory locations it occupies. Specifically, the token for LIST is replaced by the token for multiplication because the sequence of tokens $AC $B0 $BC is taken for the instruction LDY $BCB0, and the relocation routine changes this to LDY $CAB0.

So $BC, the token for LIST, has been replaced by $CA, the token for *. Hence, line number references following LIST (a rare verb to find in a BASIC program) can never be renumbered, and constants which appear to be line number references in a multiplication statement are subject to bogus renumbering! Fortunately, this is easy to fix.

One more thing should be done to Renumber. Some of us have a program to load PET tapes into our Apples, and some of these programs have spaces between the words or numbers in the program. In PETs this practice improves legibility, but not so in Apples, so Applesoft removes any extra spaces you may type in. Thus, Renumber does not expect spaces in, for example, GOSUB________400. Those spaces prevent Renumber from changing that 400 if renumbering gives line 400 a new line number. The fix for this problem is included in Programs 1 and 2.

Now it's your turn to do some work: if you use DOS 3.2, type the lines in Program 1; if you use DOS 3.3, type the lines in Program 2. In either case, test your results as shown below.

Type in this trivial program:

```
1 INPUT X
2 IF X < 1 THEN 1
3 ON X GOSUB 39,87
27 END
39 LIST 87
45 RETURN
87 PRINT 99 * 39
99 GOTO 45
```

Now ready the renumbering routine:

BRUN BRENUMBER

Now renumber your program:

&
LIST

The result should look like this:

```
10 INPUT X
20 IF X < 1 THEN 10
30 ON X GOSUB 50,70
40 END
50 LIST 70
60 RETURN
70 PRINT 99 * 39
80 GOTO 60
```

With the stock Renumber program in a 48K Apple, line 50 would still say LIST 87 and line 70 would now say PRINT 99 * 50. Now type NEW ... the above is worthless. Don't proceed until you get it right, because an unreliable or inaccurate tool is much worse than none at all.

Here is what you have produced. *Brenumber* is a small (ten sector) binary program which loads into locations $8DE0-$95FF, sets the upper limit of memory to $8E00 (minus one), and sets up the ampersand (&) command to invoke the functions of Renumber. Brenumber may be used only in a 48K Apple, and then only when MAXFILES has its default value of three. There are no safeguards in Brenumber, so unpredictable results occur if these constraints aren't met.

Orderly Programs

Now, suppose you are working on an Applesoft program and you decide to renumber it. Without bothering to save it, just BRUN Brenumber and you have all the facilities of Renumber available. It's important to remember that the BRUN command *did not* renumber your program; it just enabled the ampersand (&) command which does the actual renumbering. So let's think of clever ways to use the Renumber program. The program, with 16 screens of instructions, can be formidable to try to understand, but it's worth learning.

Briefly, renumbering is done by typing the ampersand (&) and maybe some parameters. The parameters tell Renumber two things: what line numbers to assign and what portion of the program is to be renumbered. All the parameters are optional, the default being to renumber the whole program 10, 20, 30, etc.

FIRST = 1000 the first line number will be 1000.
INC = 20 successive line numbers increase by 20.
START = 5000 only lines 5000 and later will be renumbered.
END = 6990 only lines up to 6990 will be renumbered.

The FIRST and INC parameters are straightforward, so let's see how the START and END parameters can help us. One way I make my programs neat and readable, as well as accurate, is to have a main routine whose line numbers are less than 1000, and a menu which eventually says something like ON SEL GOSUB 1000,2000,3000, ... 11000, for example, if there are 11 selections from the main menu. Then I use line numbers 20000 and up for subordinate routines such as entitling the screen, formatting numbers, etc.

So how do I preserve this orderly scheme in renumbering the program? Well, consider the effect of these commands:

& F 100, S 0, E 999 (parameters can be abbreviated)
& F 1000, S 1000, E 1999
& F 2000, S 2000, E 2999

and so on, until finally

& F 20000, S 20000

The first command will renumber only the main routine; the second will renumber lines 1000-1999, etc., and the last will renumber only the elementary routines. All very fine, but who wants to type in 21 commands to renumber a program? Well, here is a simple six-line program to create an EXEC FILE named RENUM. Customize the program to suit yourself, then run it one time and keep its output on the same disk you have Brenumber on. Then, when you wish to renumber a program in the complex way outlined above, just type EXEC RENUM.

```
10 D$ = CHR$ (4):F$ = "RENUM":Q$ = CHR$ (34)
20 PRINT D$"OPEN"F$: PRINT D$"WRITE"F$: PRINT
     "MON I"
30 PRINT "IF PEEK(36352)<>164 THEN ?CHR$(4)"Q
     $"BRUN BRENUMBER"Q$
40 X = 100:Y = 999: FOR I = 0 TO 30: IF I THE
     N X = Y + 1:Y = Y + 1000
50 PRINT "& F"X",S"X",E"Y: NEXT
60 PRINT "? CHR$(7)": PRINT "NOMON I": PRINT
     D$"CLOSE"
```

RENUM can take several minutes to do its work on a large program, so you have an opportunity for a break. It is vital that you never press RESET while Renumber (or Brenumber) is operating – it's almost certain to destroy your program! The MON I statement at the start of the EXEC FILE causes each command to be listed as it is read from disk, so watch and wait patiently.

Hiding And Moving Lines

The HOLD and MERGE functions of the Renumber program are probably poorly understood; here is an example which barely hints at the power of these commands.

LOAD PHONE LIST	**from System Master disk**
BRUN BRENUMBER	**from the disk where you put it**
& 5400, F1000	**to make a gap for more lines of DATA**
SAVE PHONE INTERIM	**we need it on disk for a moment**
DEL 1,200	**discard the prologue and credits**
DEL 351,63999	**discard everything but DATA statements**
& F351, I1	**old DATA from 201-350 becomes 351-500**
& HOLD	**put 150 lines into "hold-file" in memory**
LOAD PHONE INTERIM	**you can DELETE it now or later**
& MERGE	**combine old and new, now 300 DATA statements**

In line 1720 and 2590, change the figure 150 to 300.
In line 1160, change the program name to PHONE LIST 300.

SAVE PHONE LIST 300	**wherever you want the finished product.**

We start with Phone List, a program on your DOS System Master disk, and double its capacity from 150 to 300 names.

This clever program actually stores names and telephone numbers in DATA statements with line numbers from 201 through 350. The two DEL statements eliminate all lines but these, which are then renumbered 351 through 500 by 1. The &HOLD command hides these lines and a LIST command at this point would show no lines. After the Phone Interim program is reloaded, the hidden lines are merged into the gap between lines 350 and 1000.

When you consider all that this involves, the process is very rapid. It's hard to see how such a significant change could have been wrought any other way, without a lot of tiresome typing. Using the techniques shown above, you can move a bunch of lines around within a program, combine two programs, and incorporate proven routines from one program to another without the error-prone step of retyping.

Some programs have lines with line numbers greater than 63999, the legal maximum. Renumber is clever enough to leave these alone, and this is probably for the best. A word of caution in this area: I once fabricated an illegal line number 65535 and spent several days looking for the mysterious cause of a number's silently changing from 2 to 2.000000007. The problem disappeared when I removed the bad line number.

As with most tools, practice improves skill. Do use the Brenumber program to its limit – it's very, very good. But, and it's a big *but,* be prudent. Save an important program before renumbering it, and don't overwrite that backup until the renumbered version is proven.

Homework Assignment: If you have an Apple II Plus with no Integer ROM Card nor Language Card, you may have no Apple Mini-Assembler. In that case, follow the instructions below. You will create a one-pass assembler which will be of use in future columns in this series. Please note that "CTRL-Y" means hold down the CTRL key and type "Y". In the lines where this is used, a space is shown for clarity only; do not type any spaces in those two lines!

How To Make A Mini-Assembler If You Have An Apple II Plus

Take a diskette to an Apple which has both Integer BASIC and Programmer's Aid.

If a 16K RAM card is installed, boot the System Master diskette.

```
]INT
>CALL -151
*D4D5G
*6000:4C 98 60
*6003<F500.F63C CTRL-Y *
*6003<F500.F63C CTRL-Y
*BSAVE MINI-ASSM,A$6000,L$140
(THANK THE NICE APPLE.)
```

Program 1.

```
RUN RENUMBER
(PRESS RETURN WHEN INVITED.)
CALL -151
8DE0: A9 8E 85 70 85 74 8D F7
8DE8: 03 A9 00 85 6F 85 73 8D
8DF0: F6 03 A9 4C 8D F5 03 20
8DF8: 6C D6 4C D0 03 4C D0 03
90DE: 20 F0 95
95ED: 30 8D 28
95F0: 20 B5 94 08 C9 20 F0 F7
95F8: 28 60
94D4: BC    <---BUG FIXER!!!
(INSERT DISK TO HOLD RESULT.)
BSAVE BRENUMBER,A$8DE0,L$820
```

Program 2.

```
RUN RENUMBER
(PRESS RETURN WHEN INVITED.)
CALL -151
8DE0: A9 8E 85 70 85 74 8D F7
8DE8: 03 A9 00 85 6F 85 73 8D
8DF0: F6 03 A9 4C 8D F5 03 20
8DF8: 6C D6 4C D0 03 4C D0 03
90DA: 20 F0 95
95ED: 30 8D 28
95F0: 20 B2 94 08 C9 20 F0 F7
95F8: 28 60
94D1: BC    <---BUG FIXER!!!
(INSERT DISK TO HOLD RESULT.)
BSAVE BRENUMBER,A$8DE0,L$820
```

©

For PET/CBM computers with a disk drive, this program will list any program in a way that can be easily understood: all the special characters for all the Commodore computers are taken into account.

A Universal Program Lister

Jim Butterfield
Associate Editor

You'll need a PET/CBM disk system to run Lister. It will neatly list any BASIC program you have on disk to the screen or printer.

There are lots of Lister type programs around. This one isn't much different, except that it is very complete. It runs very slowly; have a cup of coffee while it's running.

Why Another?

Several months ago, I passed out a program at the Toronto PET User Group meeting. It contained a number of the 4.0 disk commands. I confidently said at the time, "Those of you with earlier systems won't have any trouble converting DOPEN to OPEN and so on...."

What I didn't think of was this: users with an earlier system couldn't list the program properly. Their computers couldn't understand DOPEN tokens and printed nonsense instead.

The problem is more general. If you don't have an 80-column machine, you won't be able to make any sense out of the window-making characters that are used there. If you don't have a VIC, you'll be baffled by the characters that set color.

So I embarked upon a new Lister which would contain the special characters for all Commodore machines: PET, CBM, and VIC. It seemed like an easy project.

Code Inflation

But the program grew. As it was written, a number of possibilities kept cropping up – things that would be handy for the user if provided.

The listing job wasn't hard. Just pick it off disk, translate the tokens, and put it on the screen. But then – it would be nice if the output could go to the printer.

As long as output goes to the printer, it should be neat. Why not put spaces in strategic places? That way, ONCEGOTO5,6 might print as ON CE GOTO 5,6 and be much more readable.

If we're stretching out a line of code, it might not fit onto a single line of listing. If we need to break it in two, it would be nice to pick a logical break point, so that a word like PRINT doesn't get split in the middle.

It's often nice to see cursor movements spelled out – especially the ones that do not work on your machine. And repeated cursor movements should be numbered, so that you don't print DOWN,DOWN,DOWN. Instead, DOWN3 will deliver the message. Of course, there are other times when you would prefer to have the listing show in the same way that it does with a conventional screen LIST.

Sometimes, when your program is printing instructions, they are in upper- lowercase ("text mode") and you'd like the listing to reflect it. At other times, you need the graphics because that's what your program is printing.

Of course, if you want to do different parts of your program in different modes, you'll need a line number range in order to list the parts you want at any particular time.

The long lines combined with text mode create another problem. My printer (a 2023) is too dumb to realize that if I print over 80 characters, I want the continuation line to be in the same mode as before. Instead, it drops back to graphics mode. So I had to count characters carefully and arrange my own split lines.

Spaces are a special problem. Most of the time, they should be printed as spaces; but sometimes that's hard to read, especially when the spaces are part of a cursor-movement stream. I made a compromise on this one.

Program Details

The program is in BASIC, so you can modify it to your particular needs and printer. It won't quite fit the VIC; if you want to try a VIC modification don't forget to change the POKEs on line 630 and the PEEK at line 32768. PET/CBM machines will list VIC programs directly from disk, even where the BASIC programs can't be LOADed, LISTed or RUN on the PET.

A Few Comments On Program Variables

L9 is the length of a line, normally 40 or 80;

Q is quotes-mode; it also notes REM statements;

A$(J) is a table of cursor-control names, and **A(J)** is the corresponding character designations;

K$(J) is similarly a list of BASIC keywords;

C and **C1** are flags to tell whether adjacent characters are alphanumeric, so that we will split PRINTX into PRINT X but not PRINT"X";

B counts the number of repeated cursor movements; **B1$** is the current keyword;

JINSAM™
EXECUTIVE™
space
age
micro
software
USA
OFFICIALLY APPROVED BY COMMODORE
Used at NASA,
Kennedy Space Center
With Multiple Applications Related
to the Columbia Space Shuttle Project
including rescue operations, statistical
reports, inventory and vehicle tracking.
JINSAM EXECUTIVE™
has broken the 10,000 record limit. You may now have up to 65,000 records in one database.
We also have included a free form report generator for data entry, eliminating the need for WordPro™ and have included automatic mathematical relations eliminating the need for VisiCalc™. However, you still have these superb interfaces available.
Executive™ will be available for CBM and IBM personal computers.
JINI MICRO-SYSTEMS, Inc.
DATABASE MANAGEMENT SYSTEM DESIGN
BOX 274 KINGSBRIDGE STN., RIVERDALE, N.Y. 10463 (212) 796-6200
www.commodore.ca

F$ is the character preceding a spelled-out cursor movement; it is either a left-square-bracket or a comma;

M$ is the down-shift character for text mode printing, when needed;

P$ is the print string; everything is assembled here before printing.


```
90 REM LISTER           JIM BUTTERFIE
   LD
100 DATA 19,147,17,145,29,157,18,14
    6,20,148,141,32
110 REM 8Ø-COLUMN CURSOR STUFF
120 DATA 7,21,149,22,15Ø,14,142,25,
    153
13Ø DATA 15,143
14Ø REM VIC STUFF
15Ø DATA 144,5,28,159,156,3Ø,31,158
16Ø DATA 8,9,133,137,134,138,135,13
    9,136,14Ø
17Ø DATA HOME,CLEAR,DOWN,UP,RIGHT,L
    EFT,RVS,RVOFF,DEL,INST,RET
    URN,SPACE
18Ø DATA BELL,D.LINE,I.LINE,ER.BEGI
    N,ER.END,TEXT,GRAPHIC,SCRO
    LL.UP,SCROLL.DOWN
19Ø DATA TOP,BOTTOM
2ØØ DATA BLACK,WHITE,RED,CYAN,MAGEN
    TA,GREEN,BLUE,YELLOW
21Ø DATA LOCK,UNLOCK,F1,F2,F3,F4,F5
    ,F6,F7,F8
22Ø DIMA(4Ø),A$(4Ø),K$(9Ø)
23Ø FORJ=ØTO4Ø:READA(J):NEXTJ
24Ø FORJ=ØTO4Ø:READA$(J):NEXTJ
25Ø DATA END,FOR,NEXT,DATA,INPUT#,I
    NPUT,DIM,READ,LET,GOTO,RUN
    ,IF,RESTORE,GOSUB
26Ø DATA RETURN,REM,STOP,ON,WAIT,LO
    AD,SAVE,VERIFY,DEF,POKE,PR
    INT#,PRINT,CONT
27Ø DATA LIST,CLR,CMD,SYS,OPEN,CLOS
    E,GET,NEW,TAB(,TO,FN,SPC(,
    THEN,NOT,STEP
28Ø DATA +,-,*,/,^,AND,OR,>,=,<,SGN
    ,INT,ABS,USR,FRE,POS,SQR,R
    ND,LOG,EXP,COS
29Ø DATA SIN,TAN,ATN,PEEK,LEN,STR$,
    VAL,ASC,CHR$,LEFT$,RIGHT$,
    MID$,GO,CONCAT
3ØØ DATA DOPEN,DCLOSE,RECORD,HEADER
    ,COLLECT,BACKUP,COPY,APPEN
    D,DSAVE,CATALOG
31Ø DATA RENAME,SCRATCH,DIRECTORY
32Ø FORJ=ØTO89:READK$(J):NEXTJ
4ØØ CLOSE1:INPUT"NAME OF PROGRAM FI
    LE";G$
41Ø OPEN 1,8,3,G$+",P,R"
42Ø GET#1,A$,B$
43Ø IFA$<>CHR$(1)ANDA$<>""GOTO4ØØ
44Ø IFA$=""THENA$=CHR$(1):GET#1,X$
45Ø INPUT"LINE NUMBER RANGE  -{Ø3 L
    LEFT}";Z$
46Ø LØ=Ø:L1=Ø:L2=1E9
47Ø FORJ=1TOLEN(Z$):Y$=MID$(Z$,J,1)
48Ø  Y=ASC(Y$):IFY>=48ANDY<=57GOTO5
    1Ø
49Ø  IFY=32GOTO51Ø
5ØØ  LØ=J:IFY<>45GOTO6ØØ
51Ø NEXTJ
52Ø IFLØ<LEN(Z$)THENL2=VAL(MID$(Z$,
    LØ+1)):IFL2=ØTHENL2=1E9
53Ø IFLØ>1THENL1=VAL(Z$)
54Ø IFLØ=ØTHENL1=L2
6ØØ P3$="[":P4$="]":INPUT"LIST TO P
    RINTER  N{Ø3 LEFT}";Z$
61Ø P=3:IFASC(Z$)=89THENP=4:L$="{DO
    DOWN}":P3$=CHR$(219):P4$=C
    HR$(221)
62Ø P1$="[":P2$="]":INPUT"GRAPHICS
    OR TEXT  G{Ø3 LEFT}";Z$
63Ø POKE59468,12:IFASC(Z$)=84THENPO
    KE59468,14:M$=L$:P1$=P3$:P
    2$=P4$
64Ø INPUT"TRANSLATE CURSOR MOVES  N
    {Ø3 LEFT}";Z$
65Ø IFASC(Z$)=89THENT7=1
66Ø OPEN4,P:F$=P1$
67Ø J=8Ø:IFP<>3GOTO69Ø
68Ø PRINT"{CLEAR}":PRINT"++++++++++
    ++":FORJ=1TO81:IFPEEK(3276
    8+J)=32THENNEXTJ
69Ø L9=J:PRINT#4,"PROGRAM:  ";G$
7ØØ REM NEW LINE
71Ø GOSUB2Ø1Ø:Q=Ø:T1=1:C1=-1:GET#1,
    A$,B$:IFST<>ØGOTO3ØØØ
72Ø IFB$=""GOTO3ØØØ
73Ø GET#1,A$,B$
74Ø L=ASC(A$+CHR$(Ø))+ASC(B$+CHR$(Ø
    ))*256
75Ø IFL<L1GOTO1Ø8Ø
76Ø IFL>L2GOTO3ØØØ
77Ø F2=1:PRINT#4,M$;P$:P$=STR$(L)+"
    "
8ØØ REM START TEXT HERE
81Ø GET#1,A$:IFA$=""GOTO71Ø
82Ø T=Ø:A=ASC(A$):IFA=32ANDF$=","GO
    TO84Ø
83Ø IFQ=ØOR(AAND127)>31ORT7=ØGOTO9Ø
    Ø
84Ø FORJ=ØTO4Ø:IFA=A(J)THENB$=A$(J)
    :GOTO86Ø
85Ø NEXTJ:GOTO1ØØØ
86Ø IFB$=B1$THENB=B+1:GOTO81Ø
87Ø IFB>ØTHENA$=MID$(STR$(B+1),2)+F
    $+B$:GOTO89Ø
```

(continued on p. 196)

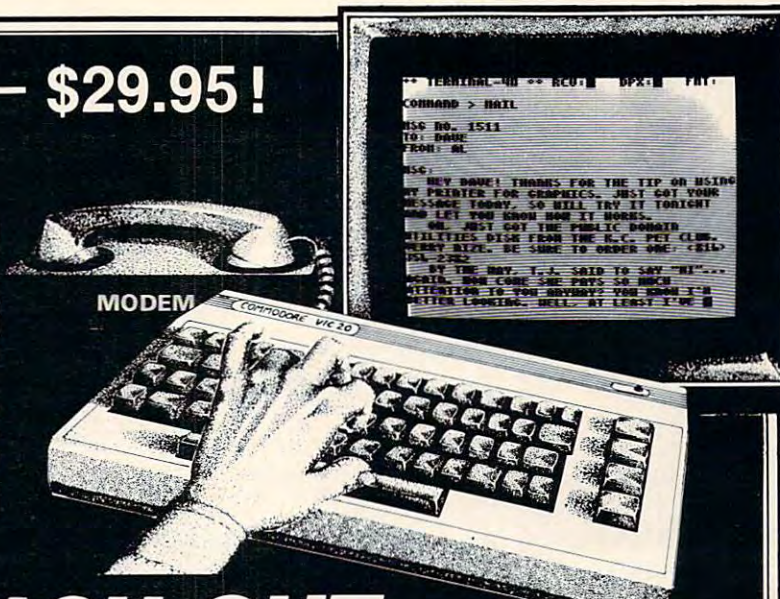
40-Column VIC — $29.95!

But . . . *there's a catch.*

We found a way to transform your cramped VIC screen into a *40-column, smooth scrolling* text display. So we created *Terminal-40*, a program for communicating by modem with networks such as CompuServe and Source. It's inexpensive, powerful, and a pleasure to use.

But we created a problem, too. Now everyone wants *all* their programs in 40 columns. That's the catch. Terminal-40 is only for telecommunications. Don't despair, though. You'll have the *finest* VIC terminal around.

So, plug in your modem and . . .



REACH OUT and BYTE SOMEONE!

TERMINAL-40 . . . $29.95
Requires 8K or larger memory expansion and modem. VIC printer optional. On cassette.

MIDWEST MICRO associates
PO BOX 6148, KANSAS CITY, MO 64110

ORDER DESK: open 9 am - 4 pm.
(816) 254-9600
VISA/Mastercard add 3% (card # and exp. date)
C.O.D. add $3.50

MAIL ORDER: Add $1.25 shipping and handling. Missouri residents include 4.6% sales tax. Foreign orders add $5 shipping and handling. **Write for free brochure. Dealer inquiries invited.**

WE'VE GOT AN OFFER YOU CAN'T REFUSE!

MAILING LIST and LABELS
TOTL. LABEL 2.0
(VIC and 64)
- Easy Editing
- User defines label size
- Automatically sorted
- Optional non-printing data lines
- Browse through list
- Select labels for printing

$20.00

WORD PROCESSING
TOTL. TEXT 2.0 (VIC only)
- Full capability word processing
- Margin and spacing control
- Centered title lines. Indentation and tabs. Upper and lower case and graphics. Full screen editing. Scrolling up and down
- No limit to document length

$25.00

TOTL. TEXT 2.5 (VIC)
TOTL. TEXT 2.6 (64)
All TOTL. TEXT 2.0 features plus:
- Heading lines (up to 4)
- Footing line and footnotes
- Keyboard input
- Right margin justification
- Print from memory or device (2.6 only)

$35.00

TIME MANAGEMENT— SCHEDULING, REMINDERS
TOTL. TIME MANAGER 2.0
(VIC and 64)
Enter and edit activities
- Uses dates and times
- Inquiry by date, person, project
- Bar chart reports from 2 days (24 hours) to 4 years
- Print reports by activity, person or project
- Reports may also be sorted by dates
- 56 possible report formats

$25.00

KEY WORD CROSS REFERENCE
RESEARCH ASSISTANT 2.0
(VIC and 64)
Great for students and authors
- Compile reference notes
- Save bibliographic data
- Quick cross reference by keyword
- Print data and cross reference lists
- Edit reference data (with disk only)

$25.00

SOFTWARE for **VIC 20*** *and* **COMMODORE 64***
QUALITY at a price that's music to your ears!

TOTL software

SHIPPING INCLUDED.
California Residents add 6% Sales Tax.
$3.00 charge for C.O.D.

Send check or money order to:
TOTL. Software
P.O. Box 4742
Walnut Creek, CA 94596
Call (415) 943-7877

SPECIAL VERSIONS AVAILABLE FOR QUANTUM DATA INC. 40/80 COLUMN VIDEO BOARD
Designed for tape and/or disk input/output.
VIC programs require 8K expansion.
Will run with VIC or RS-232 printers.
Specify machine when ordering.
All programs shipped on cassette tape.
DEALER INQUIRIES WELCOME.

*VIC 20 and COMMODORE 64 are trademarks of Commodore Business Machines.

```
880 A$=F$+B$
890 B=Ø:B1$=B$:F$=",":F1=1:GOTO1Ø1Ø
900 A=A-128:IFA<ØORQ<>ØGOTO1ØØØ
910 IFA=127THENA$="_":GOTO1ØØØ
920 T=1:A$=K$(A)
930 IFA=15THENQ=2
1000 GOSUB2Ø1Ø
1010 IFA$=CHR$(34)THENQ=1-Q
1020 REM C=-1 FOR ALPHANUMERIC
1030 C=ASC(LEFT$(A$,1)):C=(C<48ORC>5
     7)AND(C<65ORC>9Ø)
1040 IFT<>T1ORT=1THENT1=T:IFNOTCANDN
     OTC1THENP$=P$+" ":GOSUB25Ø
     Ø
1050 C=ASC(RIGHT$(A$,1)):C1=((C<48OR
     C>57)AND(C<65ORC>9Ø))ORA=3
     7
1060 P$=P$+A$:GOSUB25ØØ
1070 GOTO81Ø
1080 REM SKIP TO NEXT LINE
1090 GET#1,A$:IFA$=""GOTO71Ø
1100 GOTO1Ø9Ø
2000 REM CLOSE OFF CURSOR EXPRESSION
2010 IFF1=ØGOTO2Ø4Ø
2020 IFB>ØTHENP$=P$+MID$(STR$(B+1),2
     ):GOSUB25ØØ
2030 B=Ø:F1=Ø:B1$="":P$=P$+P2$:GOSUB
     25ØØ:F$=P1$
2040 RETURN
2500 IFLEN(P$)<L9GOTO26ØØ
2510 FORJ=L9TOL9*.6STEP-1
2520 IFMID$(P$,J,1)=":"GOTO258Ø
2530 NEXTJ:FORJ=L9-1TOL9*.6-1STEP-1
2540 P=ASC(MID$(P$,J))
2550 IFP=91GOTO258Ø
2560 IFP=59ORP=44ORP=93THENJ=J+1:GOT
     O258Ø
2570 NEXTJ:J=L9-1
2580 PRINT#4,M$;LEFT$(P$,J-1)
2590 P$="     "+MID$(P$,J)
2600 RETURN
3000 IFLEN(P$)>ØTHENF2=1:PRINT#4,M$;
     P$
3010 IFF2=ØTHENPRINT"** NO LINES FOU
     ND **"
3020 CLOSE1:GOSUB2ØØØ:CLOSE4
```

©

SJB DISTRIBUTORS. THE MOST COMPETITIVE PRICES ON COMMODORE PRODUCTS.

commodore

INTERFACES

ADA-1450 Serial	$149
ADA-1600 Parallel	149
RS232 cable for Vic or 64, 2m	30
Video/Audio cable for 64 & monitor	25

MONITORS — Great resolution for the CBM 64 or VIC

Panasonic, 13″ Color	$375
Amdek Color I	330
NEC JB 1201M, 12″ Color	330
NEC JB 1201, 12″ green phosphor	170
Amdek Video 300L, green phosphor	175

BUSINESS SOFTWARE

Spellmaster Dictionary (great for WordPro!)	$199
OZZ Data Base System (8050)	240
Silicon Office (database, wp)	995 (New)
Wordcraft 80	289
VisiCalc (new expanded)	199
Dow Jones Portfolio Management System (RS232)	120
WordPro 4 + or 5 +	299
The Manager	199
Legal Time Accounting	425
I.R.M.A.	295
BPI A/R, G/L, Job Cost, Inventory, Payroll	325/pkg

SJB will service any VIC or CBM64.

MasterCard, Visa, Money Order, Bank Check
COD (add $5) accepted.
Add 3% surcharge for credit cards.
In stock items shipped within 48 hours, F.O.B., Dallas, TX.
All products shipped with manufacturer's warranty.

TO ORDER CALL TOLL FREE
800-527-4893 800-442-1048 (Within Texas)

SJB will meet any competitive price under similar in-stock conditions.

SJB DISTRIBUTORS, INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238
(214) 343-1328

Business Hours:
M-F 8 to 6
Sat 10 to 2

Prices are subject to change without notice.

SOFTWARE FOR CBM 64

Word Processing	$90
Computer Tutoring Game (COCO)	50
General Ledger	199
Pet Emulator (emulates 4.0 basic)	30
CBM EasyCalc (for the 64)	99
CBM EasyFinance	50
CBM EasyPlot	80
CBM EasyScan (appointment manager)	80
Sprite-Magic (build sprites on screen with Joystick, save to disk or cassette)	30
Assembler Package for CBM 64 (cassette) Editor (creates and updates source code), Assembler, Loader, Disassembler	50
Mail Mate	50
IEEE Interface (64)	100
Parallel Interface	90
RS232 Interface (modems, printers)	45

VIC PRODUCTS

VIC 20 Computer, 5K	$199
Vic Datasette Recorder	60
Vic 1541 Disk Drive	395
VIC MODEM (for CBM 64)	100
VIC 1525 Graphic Printer (for CBM 64)	325
8K Memory Expansion Cartridge	49
16K RAM	99
24K RAM	155
IEEE Interface (VIC)	85
Gorf (great arcade game)	30
Omega Race	30
Midnight Drive	23
VIC 3 slot Expander	43
VIC 6 slot Expander	83
Seawolf	23
Cosmic Cruncher	23

Arcade Joysticks — Heavy duty with 2 firing buttons! Great for the VIC or 64	**$25**
SuperPET (5 languages, 2 processors)	$1409
CBM 8032 Computer, 80 column	1029
CBM Memory Expansion, 64K	359
PET 4032, 40 Column	950
CBM 8050, 1 Mg. Dual Drive	1259
CBM D9060, 5 Mg. Hard Disk	2240
CBM D9090, 7.5 Mg. Hard Disk	2600
CBM 4040, 340K Dual Drive	919
CBM 2031, 170K Single Drive	489

PRINTERS — LETTER QUALITY

CBM 8300,40cps	$1450
Diablo 620,25cps	1350
Nec Spinwriter 7700,55cps	2350
Nec Spinwriter 3500,35cps	1600

PRINTERS — DOT MATRIX

CBM 4022,80cps/graphics	$395
CBM 8023,150 cps/graphics	599
Okidata 82A,120cps/serial or par	449
Nec 8023A(parallel)	499

This program is only 48 lines long, loads in only 36 cassette turns, uses up only 1.6K, and costs nothing – but it will renumber your BASIC program in RAM, resolve line number references, and remain in memory so you can use it again and again. And all of it is in BASIC!

RENUMBER

(And a Brief Exploration Of BASIC)

Manny Juan
Dale City, CA

Type this program as is into your Atari – the first three lines must be entered exactly as shown – and save it on a cassette with the LIST"C" command. This command saves the program as ASCII text instead of as tokenized statements (as when a program is saved with CSAVE). A program saved this way may be reentered later to merge with another program already in memory, as described below.

Now type NEW to clear memory and CLOAD your favorite program. Make sure that the highest line number is less than 32100 and that it is an END statement. After the load is finished, place the tape containing RENUM (the renumbering program) into the cassette drive and type ENTER"C". This will make the Atari think that program statements (which are normally entered at the keyboard) are now being ENTERed from the cassette drive. After you have done this, RENUM becomes a part of your program, occupying the last 48 lines of it and ready to be invoked.

To renumber your program, simply type GOTO 32100. The program displays "FROM,BY?" and awaits your response. Type the line number you want your program to start with, followed by the increment value you desire. Please make sure that the potential line numbers will not extend beyond 32100. Sit back and wait for a couple of minutes. (The time varies according to the size of the program and the number of line number references RENUM has to resolve.)

This utility will renumber your program according to the starting number and increment value you supply. It also resolves all line number references in the following statement types: GOTO, GOSUB, IF...THEN, ON...GOTO, ON...GOSUB, TRAP, and RESTORE. It can recognize references to non-existent line numbers (e.g., TRAP 40000), and it attempts to recognize symbolic references (e.g., GOTO LABEL).

Whenever it encounters any of these conditions, RENUM will display, on the screen, the new line number of the current line being scanned, followed by "NF" if the referenced line was Not Found, or "SR" if a Symbolic Reference was encountered.

I suggest that you note these messages on paper so that you may investigate them later. Statements flagged with "NF" (other than some TRAP statements which may reference line numbers above 32768) usually imply that those statements are unexecutable. The presence of "SR" messages should tell you to look for those places in the program where the offending symbolic reference is assigned a value, so it can be adjusted according to the new numbering sequence.

When the renumbering process is completed, this utility displays the number of lines in your program, followed by this message:

LIST"C:",bbbbb,eeeee

where *bbbbb* is the beginning number and *eeeee* is the ending number of your program. You may position the cursor over this line and press the RETURN key if you are ready to save your program in ASCII format on cassette. (Note that a CSAVE command issued at this point would have saved your program and this utility on cassette in tokenized form.) Just remember to use the ENTER"C" command to reload your program next time, though. After that, you then CSAVE it again in a more compact form.

If you are doing program development, RENUM becomes a very handy tool to use to "open up" crowded line numbers to allow easy insertion of new lines. And if you are an author, RENUM adds a slight touch of professionalism to your articles with neatly renumbered program listings.

Program Logic

The logic of RENUM is very simple. Starting from the first line, it scans each statement and considers only those that may refer to a line number (GOTO,