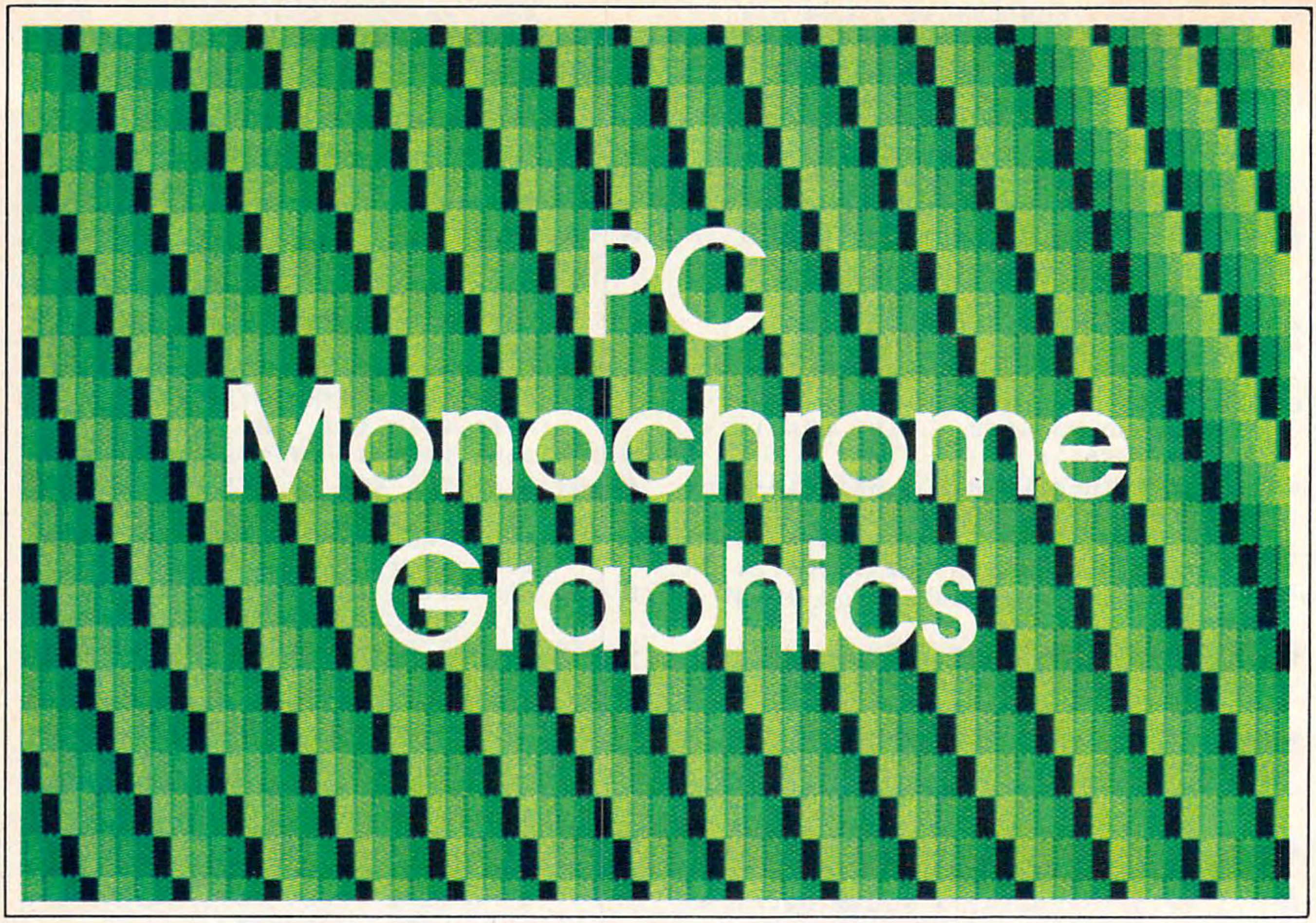
```
NO 250 LR=LR-1:FOR C=1 TO 3:FOR W=1 TO
        30-C*5:SOUND 37+W,.1:CIRCLE (B
       X*8-5,171),W,4-C:NEXT:NEXT
DP 260 FOR W=30 TO 60 STEP 2:SOUND W+1
       0,.1:CIRCLE(BX*8-5,171),W:NEXT:
       IF LR THEN 140 ELSE 510
EB 270 BX=STICK(0)/3:IF BX<1 THEN BX=1
        ELSE IF BX>40 THEN BX=40
HO 280 LOCATE 22,BX:PRINT CHR$(129):IF
        STRIG(1)=0 THEN LATCH=0:GOTO 3
       40
NO 290 IF LATCH THEN 340
FP 300 LATCH=1:FOR Y=21 TO 1 STEP-1:IF
        SCREEN(Y,BX)=0 THEN NEXT
EJ 310 RX=BX*8-5:LINE (RX,167)-(RX,Y*8
       ),2:SOUND 5000,1:LINE (RX,167)-
       (RX,Y*8),0
EM 320 IF Y THEN LOCATE Y,BX:PRINT CHR
       $(32):HITS=HITS+1:SOUND 37,2:SC
       R!=SCR!+Y*10:Z$=STRING$(8,48):A
       $=MID$(STR$(SCR!),2):MID$(Z$,9-
       LEN(A$))=A$:LOCATE 25,8:PRINT Z
       $;
KN 330 IF Y>0 AND Y<10 THEN R=INSTR(AX
       $(Y),CHR$(BX)):AX$(Y)=LEFT$(AX$
       (Y),R-1)+MID$(AX$(Y),R+1)
GJ 340 IF WAVEOVER THEN 400
LI 350 Z=0:FOR I=1 TO 9:Z=Z+LEN(AX$(I)
       ):NEXT:IF Z=0 THEN WAVEOVER=1:G
       OTO 400
GI 360 Y=INT(9*RND+1):A$=AX$(Y):R=LEN(
       A$):IF R=0 THEN 360
NO 370 R=INT(R*RND+1):X=ASC(MID$(A$,R)
       ):IF SCREEN(Y+1,X)=128 THEN 400
JJ 380 AX$(Y)=LEFT$(A$,R-1)+MID$(A$,R+
       1) ELSE 400
NJ 390 POKE &H4E,POINT(X*8-7,Y*8-7):LO
       CATE Y,X:PRINT CHR$(32):LOCATE
       Y+1,X:PRINT CHR$(128):POKE &H4E
       ,1:IF Y<9 THEN AX$(Y+1)=AX$(Y+1
       )+CHR$(X)
NE 400 Z!=FNML!(0):CALL Z!:PSET (319*R
       ND,72+8*RND),3*RND:GOTO 220
BI 410 LINE (64,72)-(240,104),0,BF:LIN
       E (64,72)-(240,104),1,B
DK 420 WAVENUM=WAVENUM+1:LOCATE 11,11:
       POKE &H4E,2:PRINT"Wave";WAVENUM
       ;"Completed.":POKE &H4E,1
IG 430 FOR I!=0 TO SCR! STEP 10^(LEN(S
       TR$(SCR!))-3):GOSUB 490::SOUND
       37,.1:NEXT:I!=SCR!:GOSUB 490
JP 440 LOCATE 11,10:POKE &H4E,1:PRINT
       SPACE$(20):LOCATE 11,17:PRINT"B
       ONUS":POKE &H4E,3
LN 450 Z=0:FOR I!=SCR! TO SCR!+10000 S
       TEP 250:GOSUB 490:SOUND 100+Z,.
       1:Z=Z+1:NEXT:SCR!=SCR!+10000:I!
       =SCR!:GOSUB 490:IF AL-HITS=0 TH
       EN 480
DE 460 LOCATE 11,13:PRINT"Penalty for"
       ;AL-HITS;:LOCATE 12,13:PRINT"sp
       iders escaped":POKE &H4E,2:Z=AL
       *10
MJ 470 S!=SCR!:SCR!=SCR!-(AL-HITS)*100
       0:SCR!=-SCR!*(SCR!>0):FOR I!=S!
        TO SCR! STEP -100:GOSUB 490:SO
       UND 100+Z,.1:Z=Z-1:NEXT:I!=SCR!
       :GOSUB 490
DH 474 IF SCR!=0 THEN LOCATE 13,10:PRI
       NT"  One life lost  ":LR=LR-1
       :IF LR=0 THEN 510
KK 480 FOR W=1 TO 3000:NEXT:FOR I=1 TO
        15:Z!=FNML!(0):CALL Z!:NEXT:GO
       TO 140
MA 490 Y=13
EF 500 Z$=STRING$(8,48):A$=MID$(STR$(I
       !),2):MID$(Z$,9-LEN(A$))=A$:LOC
       ATE Y,13:PRINT"Score:";Z$:RETUR
       N
NI 510 CLS:FOR I=1 TO 150:PSET (320*RN
       D,199*RND),3*RND+1:NEXT:LINE (6
       4,0)-(240,40),1,B
PB 520 LINE (0,176)-(319,199),3,BF
JI 530 S!=1:FOR X=0 TO 104 STEP 8:LINE
        (160-X,176)-(160-X-S!,199),2:L
       INE (160+X,176)-(160+X+S!,199),
       2:S!=S!*2:NEXT
EM 540 FOR Y=176 TO 199 STEP 4:LINE (0
       ,Y)-(319,Y),2:NEXT
HO 550 LOCATE 2,15:POKE &H4E,1:PRINT"G
       ame Over":POKE &H4E,3:LOCATE 3,
       11:PRINT"Play Again? (Y/N)":I!=
       SCR!:POKE &H4E,2:Y=4:GOSUB 500
KH 560 A$=INKEY$:IF A$<>"" THEN IF A$=
       "y" OR A$="Y" THEN RUN ELSE SCR
       EEN 0,0,0,0:END
JF 570 LOCATE 10,38*RND+1:POKE &H4E,3*
       RND+1:PRINT CHR$(128):Z!=FNML!(
       0):CALL Z!:PSET (319*RND,72+8*R
       ND),3*RND:GOTO 560
JA 580 GOTO 580
ON 590 'Redefine the character set
II 600 DEF SEG=0
HP 610 POKE 124,0:POKE 125,0:POKE 126,
       0:POKE 127,&H17
IH 620 DEF SEG=&H1700:FOR I=0 TO 15:RE
       AD A:POKE I,A:NEXT
MG 630 RETURN
MH 640 DATA 129,90,60,219,126,102,129,
       66
HM 650 DATA 24,24,60,126,126,255,219,1
       95
LD 660 DEF SEG:FOR I=1 TO 21:READ Z:ML
       $=ML$+CHR$(Z):NEXT:V!=VARPTR(ML
       $):DEF FNML!(Z)=PEEK(V!+1)+256*
       PEEK(V!+2):RETURN
GE 670 DATA &H55,&H1E,&HB8,1,7,&HB9,0,
       9,&HBA,39,21,&HBB,0,0,&HCD,16,&
       H1F,&H5D,&HCA,0,0
OE 680 DATA "3  1  2      2  2      2
       1  3
DJ 690 DATA "  2  3 313 3 3 3 313  3
        2
MM 700 DATA "   11  33 1  3333  1  33
       11
KG 710 DATA "     1 11 11 11  11 11 1
AB 720 DATA "   3 2  3 1 3 1 3 1 3  2
        3
IN 730 DATA "    1123232323232323232211
PO 740 DATA "    2 3 2 3 2 3 2 3 2 3
KP 750 DATA "       2 3 1 2 3 1 2 3
LD 760 DATA "     2  3  11    11  3  2
```



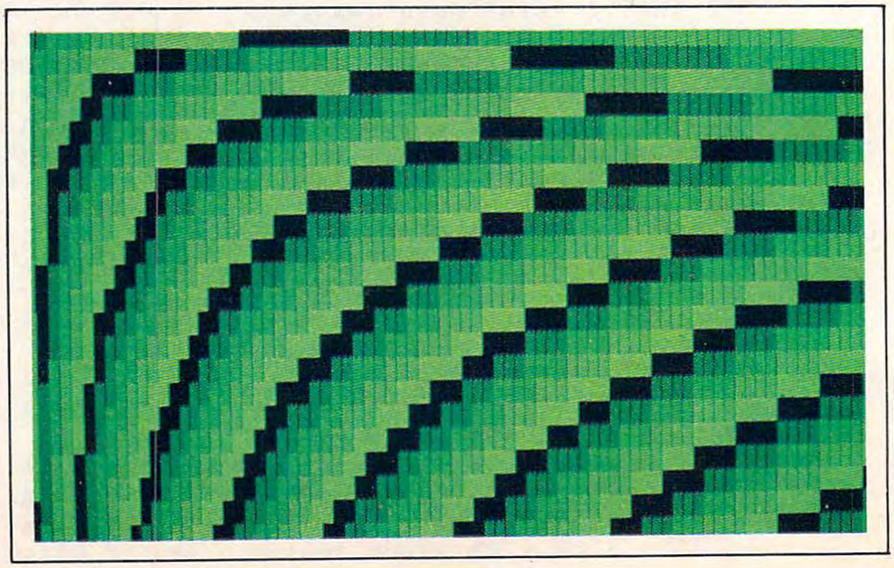
PC Monochrome Graphics

Michael A. Covington

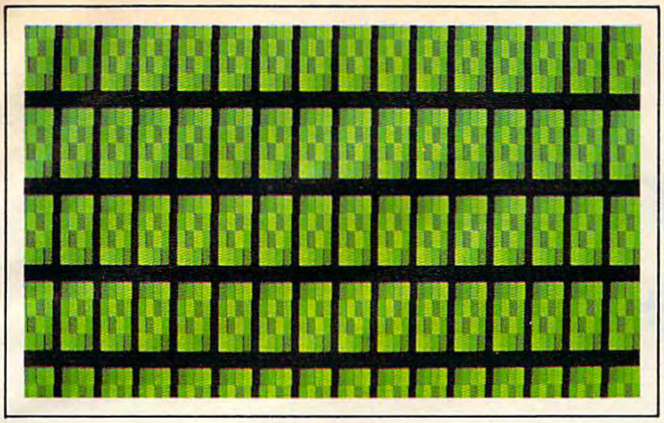
Here are some graphics screens which don't require a color/graphics adapter—they'll work on any IBM PC (or PCjr), even those equipped only with the monochrome display board.

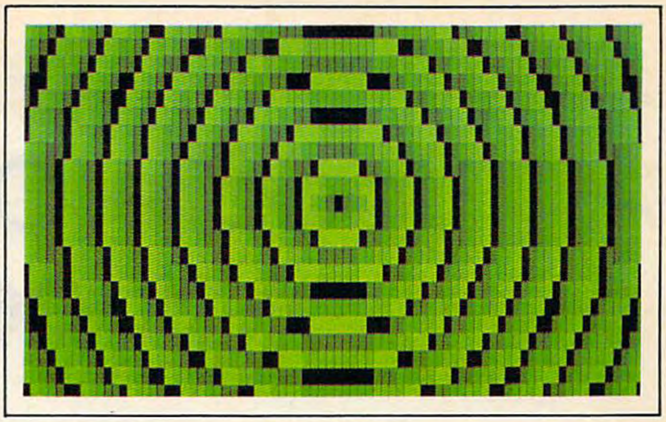
Owners of the IBM monochrome display often feel a bit left out because their systems do not support point-by-point graphics. The monochrome display adapter can produce a variety of special characters that could be used to construct drawings, but most programs never employ more than a few of them.

The brief program accompanying this article takes advantage of these special characters to create dramatic-looking patterns—actually contour maps of three-dimensional mathematical functions. Some of the displays look rather like Scottish tartans. The program runs on a PC or PCjr with any display, but the IBM monochrome display yields the best results.



Line 160 in the program defines the variable W as a function of ROW and COL. The function can be anything you wish. Here are some functions that result in attractive patterns:





W = ROW+COL
W = ROW*COL
W = LOG(ROW)−LOG(COL)
W = LOG(ROW²+COL²)
W = 5*SIN((ROW+COL)/10)

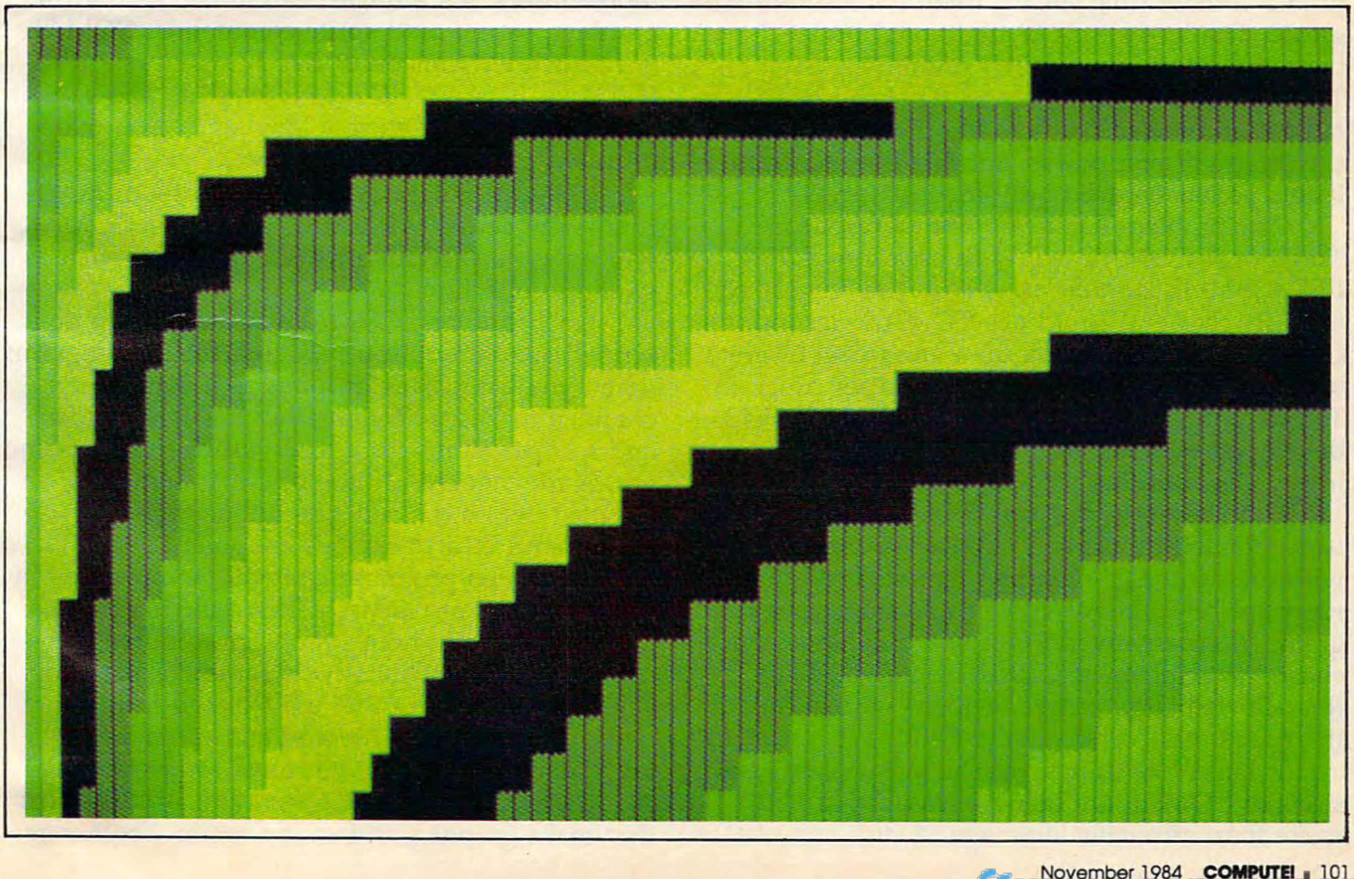
An almost infinite variety of other functions are possible. Just substitute your function for line 160, save, and then run the program. Exit with Ctrl-Break.

PC Monochrome Graphics

Refer to "COMPUTE!'s Guide For Typing In Programs" article before typing this program in.

```
BP 30 SCRNWIDTH = 80
CE 50 DIM A$(4)
KH 60 A$(0)=" "
MM 70 A$(1)=CHR$(176)
NF 80 A$(2)=CHR$(177)
PO 90 A$(3)=CHR$(178)
HC 100 A$(4)=CHR$(219)
HI 110 CLS:KEY OFF
CP 120 FOR ROW=1 TO 22
JM 130 FOR COL=1 TO SCRNWIDTH-1
NP 140 ' Change the following line
EK 150 ' to get different patterns
OH 160 W = SQR((2*(ROW-11))^2 + (COL-4
   0)^2)
AF 170 PRINT A$(ABS(W) MOD 5);
ON 180 NEXT
JC 190 PRINT
NO 200 NEXT
PP 210 BEEP
```

©



Update On COMAL: A SuperBASIC

Jim Butterfield, Associate Editor

COMAL is a computer language which was developed in Denmark as an offshoot of BASIC. Some programmers prefer it and think of it as a streamlined, extended, and systematized BASIC.

COMAL has been available for some time, mostly for Commodore 32K PET/CBM machines. Previous versions have had some success despite a few drawbacks and implementation problems. Now, new versions of COMAL are being released, and the new packages look promising.

The New Generation

COMAL has traditionally been available in the public domain (that means free). It's a sound language: The loose ends of traditional BASIC have been tightened into a much smoother system. The ease of use for beginners has been maintained, and added features significantly enhance the programming power. Yet this free, powerful language has enjoyed only modest success.

The problem has been limited resources. Traditional COMAL would fit only into a 32K PET/CBM; thus, only users with the biggest systems (of that era) could use COMAL. Second, the COMAL interpreter took up a good deal of memory, leaving room for only a small user program. To offset this difficulty, a "split" COMAL was developed which used a separate editor and interpreter; but this proved awkward to use.

Until recently, the best COMAL arrangement was obtained by using a CBM 8096 computer; with the extra 64K memory, there was plenty of room for both the interpreter and the user's program. Alternatively, a ROM board could be purchased to implement COMAL on a CBM 8032; this allowed large programs to be written, but the board was costly—about $400.

It looks like things are changing. New versions of COMAL are being released that will make it an attractive language. A disk version for the Commodore 64 is now in the public domain; a cartridge version is soon to be made available for sale; and even the ROM board for the PET/CBM 8032 is being redesigned to incorporate interesting new features.

What Is COMAL?

COMAL may be described in a number of ways. It's as easy as BASIC for beginners, but has the power of Pascal when advanced features are needed. It's a tightly defined superversion of BASIC, with much more precisely defined keywords and with statements that interrelate more closely. It's a BASIC extension, with extra graphics, sound, and sprite commands. COMAL is a structured version of BASIC, complete with IF-THEN-ELSE, WHILE-ENDWHILE, and similar features. It's an extensible language—you may even write your own keywords.

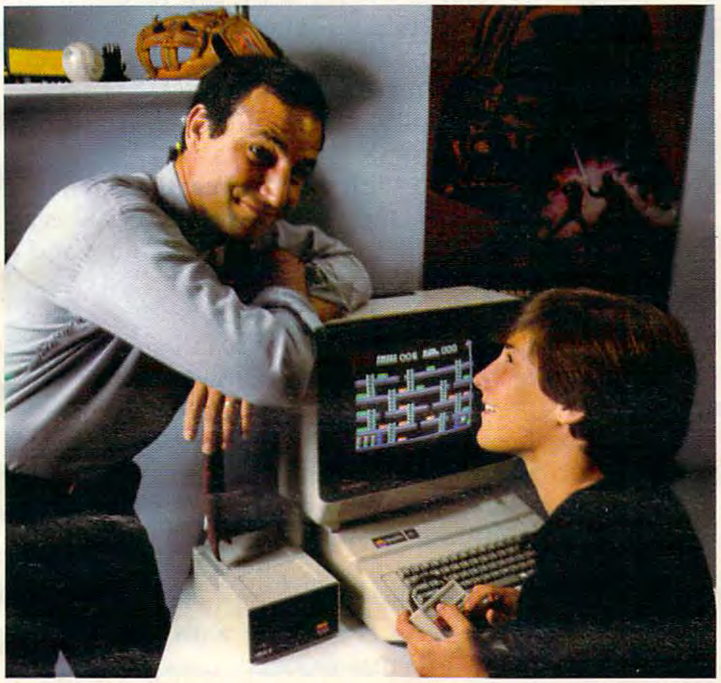
COMAL programs run as fast as—or faster than—the equivalent BASIC programs. COMAL will never have certain BASIC problems such as garbage collection.

For beginners, COMAL can be crudely described as follows: You type in a BASIC program; when you list it back, it looks as if it has been changed to Pascal. Indentation has been added and formats trimmed; it looks much neater. Some errors are checked at time of entry; some are checked before the run gets under way. Meaningful error messages are given.

Advanced programmers will be interested in other features of COMAL. The structured forms are all there, but there's more: procedures (subroutines) and functions with parameter passing; "closed" procedures that are isolated from the main coding; end-of-file and end-of-data flags; recursion; and in some versions, error trapping.

Users will find new commands and features that make graphics and sound easier. To change

Give your children more than the advantage of a computer. Give them the Scholastic Advantage.



You probably realize the computer is only the first step. Now you need Scholastic Software to turn it into both a powerful and entertaining learning tool for your child.

Scholastic Software is the only software that comes with the Scholastic Advantage. This means software which uses your child's own natural curiosity and love of adventure to make learning valuable skills fun. Software based on what your child is actually studying in school, which makes it relevant and practical. And software which is already being used successfully by millions of kids in schools around the country.

At Scholastic, we've put everything we've discovered about how youngsters learn into our new software line. It's the result of over 60 years' experience making learning both challenging and fun. In fact, you probably grew up reading Scholastic books and magazines in school. Nobody else knows kids like we do and it shows in our new family of products. That's the Scholastic Advantage.

Our software family includes *Agent U.S.A.,* which turns geography into an exciting adventure trip around the country; *Story Maker,* a program that helps kids create their own illustrated story book, and *Math Man,* an action game that makes learning math fun. *Microzine* is America's most popular classroom software, featuring four fun learning programs on one disk. *Story Tree* is both a creative writing tool and a word processor, and *Operation: Frog* is a fascinating simulation of a biology lab.

So give the Scholastic Advantage. You'll be surprised how fast the computer becomes one of your child's favorite—and most exciting—teachers.

Agent U.S.A. available in Apple, Atari, Commodore and IBM. Story Maker and Microzine available in Apple. Math Man and Story Tree available in Apple and IBM. Operation: Frog available in Apple and Commodore. Spelldiver available in Apple, Atari and Commodore.

Clip this and save $5.00 on any purchase of Scholastic Software!

Not a store coupon. Redeem by attaching dated cash register receipt and warranty card. Send to: Scholastic Software, Dept. C.M., 730 Broadway, New York, NY 10003.

Name__________

Address__________

City__________ State_____ Zip_____

Store name__________

Product purchased_____ Computer owned_____

Offer good September 15, 1984–January 31, 1985. Limit one per family and one per product.

Scholastic

The Most Trusted Name in Learning

the background color to black, type BACKGROUND 0. On the high-power end, COMAL 64 comes complete with turtle graphics commands: FORWARD 40 and RIGHT 45 have the same meaning as they would have in Logo.

Versions Of COMAL

Disk-based COMAL for the Commodore 64 is named COMAL 0.14. It's free in North America, but it's not public domain. The package prints a copyright statement, but permits unrestricted distribution and copying. COMAL 0.14 is available from clubs, user groups, and the COMAL user group in Madison, Wisconsin. It's a good language implementation, complete with graphics and a complete set of error messages drawn from disk as needed.

COMAL 2.0 is a plug-in cartridge for the 64 which is expected to be available from Commodore in 1985 for less than $100. It's significantly more powerful than COMAL 0.14—it offers much more program memory and includes extra features such as error trapping and program chaining or overlaying.

The COMAL board has been redesigned for the 8032 computer. The price of the revised board is expected to be similar to the previous version—that is, around $400—but there are some new features. First, if you don't want to use COMAL immediately, you can use BASIC, and the computer becomes an 8096 with 64K extra memory supplied by the COMAL board. Second, a video board can be added to the assembly to perform high-resolution graphics, including turtle geometry. Third, the board contains a time-of-day and date clock which is battery-powered and keeps good time even when the unit is off. I installed a board and checked the clock; before I reset it, it gave the correct time for Copenhagen, where it was manufactured.

Documentation

The *COMAL Handbook*, by Len Lindsay (published by Reston Publishing), is a reference manual for the language. It's just that: a reference manual, and not a tutorial. You can look things up, but it's not for learning the language.

Fortunately, most COMAL versions come with a disk of sample programs that illustrate the features of the language well. And to be fair to Lindsay's book, it contains a considerable quantity of sample programming.

If you're interested in the new 64 versions of COMAL, look specifically for the second edition of the *COMAL Handbook*. The publication date for this new edition hasn't yet been set. The original handbook gives a great deal of information on COMAL, but doesn't show the new color and graphics commands or other new features such as error trapping.

Most users will benefit from the fact that COMAL is derived from BASIC. Their BASIC experience will generally carry over to the new language.

Beginning Characteristics

Users may start COMAL programming as they did BASIC. Direct commands are allowed so that statements such as PRINT 5+7 can be executed.

Spaces must be used after keywords. FORK=1TO5 must be written as FOR K=1 TO 5. There are two reasons for this: COMAL encourages legibility, and COMAL allows long variable names. FORK could be a variable; it won't be confused with FOR K because of the space. By the way, the whole variable name is used, so that FORK is distinct from FORM, and both are different from the variable called FOR'LOVE'OF'IVY.

Even though COMAL has a full set of structured statements available, it allows the use of a GOTO statement. However, you can't GOTO a line number; you must GOTO a labeled statement in your program. COMAL is quite insistent that line numbers should be used only for editing purposes. The use of GOTO is strongly discouraged; there are usually plenty of alternative ways of programming what you need.

Subroutines are called, not with GOSUB, but with EXEC for execute. COMAL uses the term *procedure* rather than subroutine; a procedure is given a name. So instead of GOSUB 500 we might code EXEC SUMMARY. Since procedures have names, we don't need the keyword EXEC. Instead of EXEC SUMMARY we may just write SUMMARY, and the interpreter will call the procedure when it reaches that point. In this way, the language is extensible. We might code a program starting as follows:

```
100 START
110 CONTINUE
120 FINISH
130 ....
```

What this sequence means is: Call procedure START, then call procedure CONTINUE, then call procedure FINISH. Each of these will need to be defined as a procedure (a type of subroutine) somewhere in the program, but here's the point: The above coding is quite readable; and we have essentially defined new program commands.

It's impossible in this short article to deal with the whole spectrum of commands. Perhaps the above will give a flavor of how COMAL extends the capabilities of BASIC.

A Mix Of Features

COMAL isn't new, but there's a new COMAL. If languages interest you, this one is well worth a

Now your teenagers can learn from teachers who teach teachers how to teach.

Introducing The Series by Master Teachers from Thoroughbred.

This unique series offers your teenagers the rare opportunity to learn—to reinforce and integrate concepts studied in school—through software developed by Master Teachers.

Teachers who have been recognized by other teachers as being truly outstanding for their knowledge. And for their ability to inspire people to learn—not by rote, but through *active participation*. So that learning is quick, thorough, and fun.

Master Teachers are so good at teaching, in fact, they teach *teachers* how to teach.

11 titles are available now

Biology
- ☐ Exploring that amazing food factory, the leaf
- ☐ Photosynthesis: unlocking the power of the sun
- ☐ The fascinating story of cell growth
- ☐ How plants grow: the inside story

Math
- ☐ Decimals made simple
- ☐ Mastering units of measurement
- ☐ Taking the mystery out of metrics

Social Science
- ☐ The U.S. Constitution: our guarantee of liberty
- ☐ The Constitutional amendments: what they mean to you

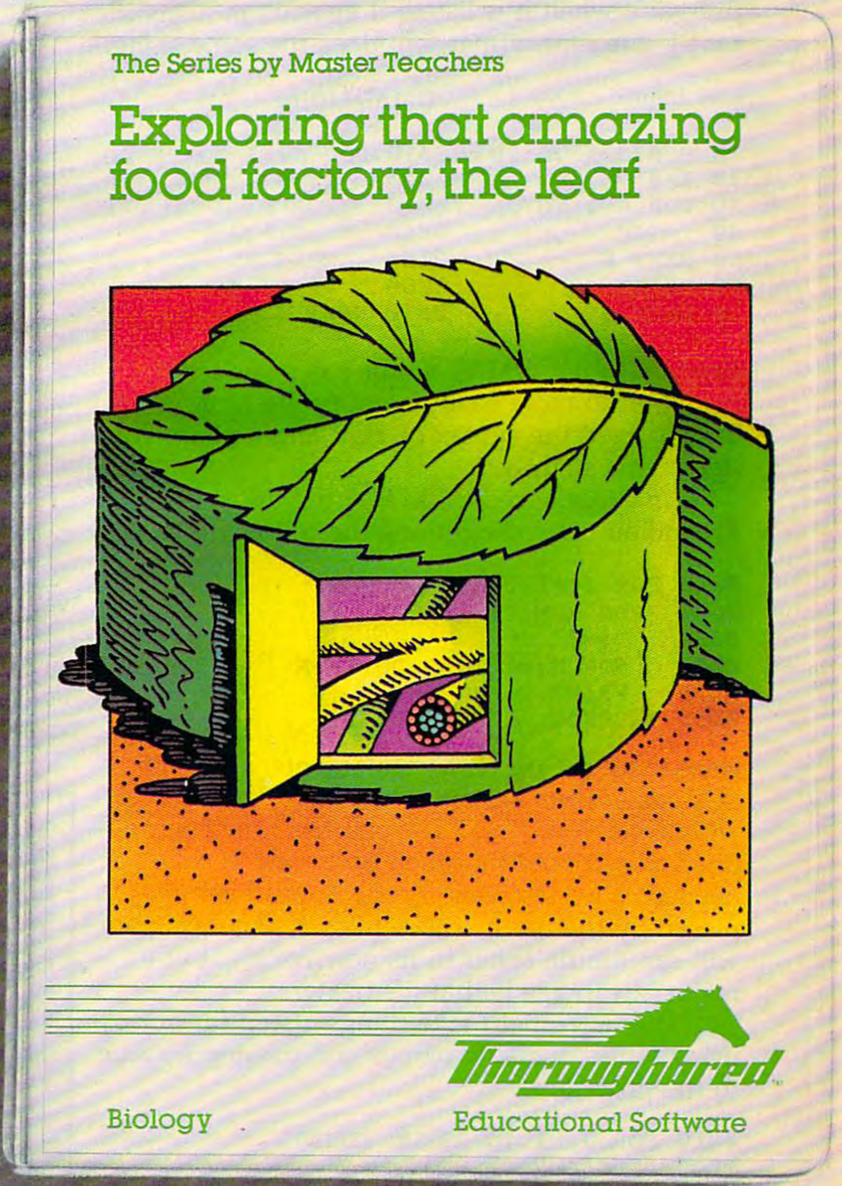
Chemistry
- ☐ Molecules and atoms: exploring the essence of matter
- ☐ The how's and why's of migrating molecules

More to come. Plus English, physics...
All, with advanced color graphics.

Level of knowledge covered: junior high to adult (except "Decimals," elementary to senior high).

For use with: IBM PC, PCjr.* Available soon for Apple IIe, IIc.**

Visit your dealer *today*. Or call us at 800-526-3968 (or 201-685-9000). Or write: SMC Software Systems, 1011 Route 22, Bridgewater, NJ 08807.



SMC Software Systems
A Division of Science Management Corporation

look. It's easy on the beginner, yet it's not limiting to the expert programmer. It's more than BASIC, not exactly a Pascal, and different from Logo—but it has some of the characteristics of all three languages.

It's inexpensive to try. You may find that it's just what you have been looking for.

Further information on COMAL may be obtained from:

COMAL User's Group
5501 Groveland Terrace
Madison, WI 53716

Reston Publishing
11480 Sunset Hills Road
Reston, VA 22090

COMAL Examples

The user may input what appears to be a BASIC program; when it is listed, it appears in a significantly different format, but does the same thing.

User Input	COMAL Listing
`10 FOR J=1 TO 20`	`0010 FOR J:=1 TO 20 DO`
`20 T=0`	`0020   T:=0`
`30 FOR K=1 TO J`	`0030   FOR K:=1 TO J DO`
`40 T=T+K`	`0040     T:=T+K`
`50 NEXT K`	`0050   NEXT K`
`60 PRINT T`	`0060   PRINT T`
`70 NEXT J`	`0070 NEXT J`

Note that the COMAL editor has provided indentation to more easily identify the loops, and has changed the syntax of FOR and assignment statements slightly.

An experienced COMAL programmer might streamline the coding along the following lines:

```
0010 FOR J:=1 TO 20 DO SUM
0020 PROC SUM
0030   T:=0
0040   FOR K:=1 TO J DO T:+K
0050   PRINT T
0060 ENDPROC
```

Here there are no NEXT statements, and both FOR statements are one-line loops. The coding within the FOR J-NEXT J loop has now been defined as a procedure called SUM and is called as needed. The statement T=T+K has been replaced by the more efficient T:+K. The program will run a little faster in its new coding, but its major advantage is that it's neater.

The user might take things a step further by using longer variable names and passing a value to the procedure:

```
0010 FOR J:=1 TO 20 DO SUM(J)
0020 PROC SUM(VALUE)
0030   TOTAL:=0
0040   FOR INDEX:=1 TO VALUE DO TOTAL:+IN
       DEX
0050   PRINT TOTAL
0060 ENDPROC
```

Labels such as TOTAL and VALUE would be forbidden in BASIC (the keywords TO and VAL are there), but not in COMAL. The new program takes slightly more space than before and runs at the same speed. Procedure SUM could now be completely disconnected from the main routine (via PROC SUM(VALUE) CLOSED) since it is passed all the required data; it doesn't need to "raid" the main program variables.

We can go a step further by using a defined function in this case:

```
0010 FOR J:=1 TO 20 PRINT SUM(J)
0020 FUNC SUM(VALUE)
0030   TOTAL:=0
0040   FOR INDEX:=1 TO VALUE DO TOTAL:+IN
       DEX
0050   RETURN TOTAL
0060 ENDFUNC
```

The value of SUM is calculated each time it is used. Using a defined function can generate a very readable program. The statement RETURN is different from a BASIC subroutine RETURN. It means, "give back a value of TOTAL to the calling program."

Common Sense And COMAL

This program generates the sums of various series of numbers. It's meant to show COMAL options; it's certainly not the best way to do the job. As any math teacher will tell you, the sum of the numbers from 1 to J can be calculated by the formula (J+1)*J/2. We could therefore reduce the function to a single line and write the program as follows:

```
0010 FOR J:=1 TO 20 PRINT SUM(J)
0020 FUNC SUM(VALUE)
0030   RETURN (VALUE+1)*VALUE/2
0040 ENDFUNC
```

That's not an advantage caused by the language—that's just a little math and common sense. ©

IMAGINE IT. CREATE IT. EVEN PRINT IT.
IT'S AMAZING WHAT YOU CAN DO WITH THE KOALAPAD.

You have the vision. Now you've got the touch. The KoalaPad™ touch tablet plugs right into your computer, and makes beautiful high-resolution graphics easy. For charts and graphs, for cartoons and electronic artwork, or just for doodling around. You don't need programming skills. You don't need to remember commands.

The KoalaPad comes complete with graphics software that displays a full array of colors, shapes, patterns and functions right on the screen. Just point to what you want, then draw right on your KoalaPad. Instantly, your visions turn into spectacular full-color images right on your screen.

You can save your designs to disk. And now, with our Graphics Exhibitor™ (for Apple®) or KoalaPrinter™ (for Commodore 64™) software, you can even print out your designs – for the best reproduction of on-screen graphics available. Your imagination knows no limits, with the KoalaPad and Koala software.

For the Apple, Atari,® Commodore 64, IBM® PC and PCjr. computers.

Koala Technologies
800-KOA-BEAR

THE BEGINNER'S PAGE

Tom R. Halfhill, Staff Editor

Starting this month, "The Beginner's Page" will be written by Tom R. Halfhill, editor of COMPUTE!. *Halfhill's former column in* COMPUTE!, *"Questions Beginners Ask," will be incorporated into "The Beginner's Page."*

A Column For Everybody

Welcome to the new Beginner's Page. A popular column in COMPUTE! since March 1981, "The Beginner's Page" was conceived as a way to introduce the main concepts of personal computing to beginners. But it's also a way to unite the users of the many different computers covered in COMPUTE!.

That's why "The Beginner's Page" will continue to involve every computer brand covered by COMPUTE!. It doesn't matter if you have a $79 Commodore VIC-20 or a $4000 IBM PC. If you have an interest in learning more about personal computing, that's enough. As always, we look forward to your questions, comments, and suggestions. We'll devote most of each month's column to a specific topic, and then conclude with an answer to a general-interest question, much like "Questions Beginners Ask." Sometimes the question will come from one reader's letter, and other times it will be culled from a number of letters asking pretty much the same thing. So keep the mail coming.

Emulators: An Impossible Dream?

Certain types of questions consistently recur in the mail we receive from readers. One such question has to do with *emulators.*

An emulator is an add-on accessory or adapter that lets a computer run programs designed for another computer which is normally incompatible. The concept of an emulator is fascinating, almost mesmerizing, especially for beginners. Imagine having access to the hundreds and even thousands of programs available for other computers. It seems that your software problems would be solved overnight.

Unfortunately, it's not that simple. True emulators are very rare indeed. If ever there was reason to observe the warning "Let the buyer beware," you should heed it when encountering a sales pitch for an emulator.

For instance, when the Commodore 64 first hit the market, there were all kinds of rumors about forthcoming Apple emulators. Several companies, supposedly, were preparing plug-in modules that would let Commodore 64 owners simply load an Apple program off a disk and type RUN. Some companies even advertised their Apple emulators in magazines. But they never materialized.

Elusive Emulators

We received scores of letters from readers asking about these emulators, and we followed every lead. But each time we contacted the company involved, we got pretty much the same answer: "Available soon." Of course, "soon" stretched into "never."

At the time, the idea of an Apple emulator held great appeal for Commodore 64 owners because there wasn't much software for their brand-new computer. Now, two years later, there's a virtual glut of Commodore 64 software and the idea has lost some of its attraction. Even though the Apple has acquired a huge software library (estimated at 10,000 programs) after more than seven years on the market, the Commodore 64 software has all been written within the past two years and is generally more up to date. In fact, there are probably some Apple owners today who'd like to get their hands on a Commodore 64 emulator.

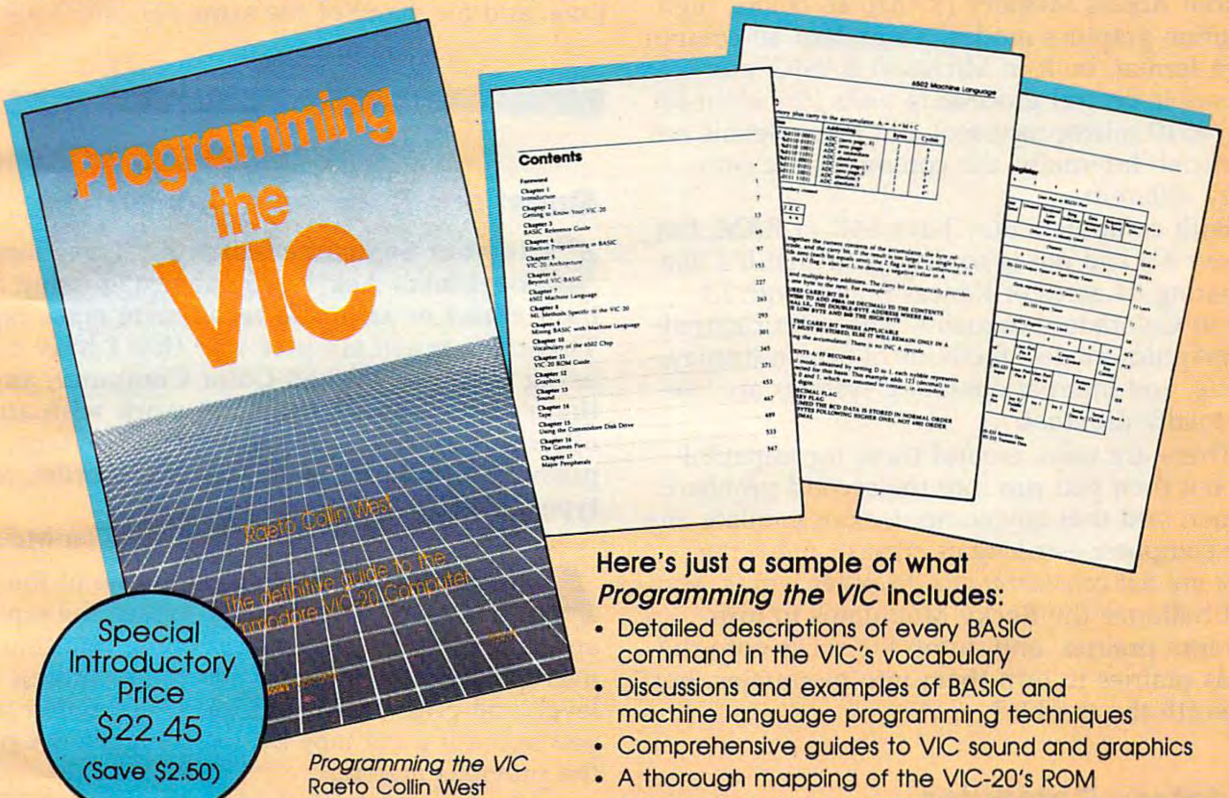
Over the years we've heard wishes, rumors, and announcements of other emulators, too: adapters to make Atari game machine cartridges compatible with Atari computers, or vice versa; a Commodore 64 emulator for the VIC-20; and even an IBM PC emulator for the Coleco Adam (actually announced by Coleco for tentative release in late 1984). But chances are you'll never

Vic-20 Owners

The ultimate reference book

The complete encyclopedia for the Commodore VIC-20 is now available. COMPUTE! Books, one of the leading publishers of application-oriented consumer computer books, has released *Programming the VIC* by Raeto Collin West. And we offer this extraordinary volume to you at a very special introductory price of $22.45 if ordered by December 15, 1984, a 10% savings off the regular price of $24.95.

For $22.45 you can own the *definitive* book on the VIC-20. There has never been a book published for the VIC-20 that gives you what this one does.



Programming the VIC
Raeto Collin West
$24.95
ISBN #0-942386-52-3
608 pgs.
paperbound

Here's just a sample of what *Programming the VIC* includes:

- Detailed descriptions of every BASIC command in the VIC's vocabulary
- Discussions and examples of BASIC and machine language programming techniques
- Comprehensive guides to VIC sound and graphics
- A thorough mapping of the VIC-20's ROM
- An annotated list of 6502 opcodes
- A practical guide to selecting and using printers, plotters, and modems

Programming the VIC's 17 chapters address virtually every programming situation that you, a VIC user, are likely to encounter. The book contains hundreds of examples and dozens of complete programs published in ready-to-type-in form.

Beginning to advanced programmers alike will find *Programming the VIC* to be an indispensable VIC resource. Whatever your programming level and whatever your VIC needs, this is a book that you'll refer to again and again.

Mail your prepaid coupon to: COMPUTE! Books, P.O. Box 5406, Greensboro, NC 27403 or call 1-800-334-0868

Yes! Send me ____ copies of *Programming the VIC* at ~~$24.95~~ $22.45* per copy.

*Limited offer. Save 10% if you order by December 15, 1984. ____

All orders must be prepaid in U.S. funds.

Add $3.00 shipping & handling per book. ____

N.C. Residents add 4.5% sales tax. ____

Total payment enclosed. $ ____

☐ Payment enclosed (check or money order).

☐ Charge my ☐ Visa ☐ MasterCard ☐ American Express

Acct. No. ____ Exp. Date ____ / ____

Name ____

Address ____

City ____ State ____ Zip ____

Please allow 4-5 weeks for delivery. 7VICGN4 see any of them. Or if you do, they'll scarcely be cost-effective.

Turning Mountains Into Prairies

There are two problems to overcome when designing an emulator: First, you have to exactly duplicate every hardware function of the computer you're trying to emulate (while avoiding patent- and copyright-infringement suits); and second, you have to match or exceed the performance and cost-effectiveness of the computer you're trying to emulate.

Let's tackle the first problem. It might seem that a Commodore 64 and an Apple IIe, for instance, have a lot in common: Both have 64K of Random Access Memory (RAM), 16 colors, high-resolution graphics modes, a standard 40-column screen format, built-in Microsoft BASIC, and compatible central processing units (the eight-bit 6502/6510 microprocessor). But these details are superficial. Internally, the computers are completely different.

Both computers may have 64K of RAM, but the way it's laid out is so dissimilar that it's like comparing 64 acres of Kansas prairie with 64 acres of Colorado mountains. The color capabilities, graphics modes, methods of screen display, BASICs, and internal operating systems are likewise totally different.

There are ways around these incompatibilities, but then you run into the second problem. It's been said that any computer can emulate any other computer—*as long as expense and performance are not considerations.* In other words, you could bulldoze the Rocky Mountains to turn them into prairies, and dump the rocks onto the Kansas prairies to turn them into mountains, but is it worth the trouble?

Emulator=Computer

The only practical way to build an emulator is to shrink the first computer down to a box or module that plugs into the second computer. Usually it's not worth it, because you could simply buy the first computer for not much more than the emulator would cost. For instance, once at a computer show we saw an Atari VCS game machine emulator for the VIC-20. It really worked, because essentially it was an Atari VCS in a plug-in module. However, it cost $89.95, and Atari game machines at that time were selling for $99.95. So the emulator cost $10 less, but didn't come with a pair of joysticks, paddle controllers, or a free game cartridge as the VCS does.

Similarly, one Apple emulator that was announced for the Commodore 64 was to be priced at about $800. But at some discount dealers, you can buy an Apple IIe and a disk drive for not much more. Why buy the emulator?

For an emulator to be worthwhile, it should provide at least 90 percent compatibility at a price significantly less than what the other computer would cost. Even then you should balance the cost of the emulator against the benefits of running the other computer's software.

Admittedly, it would be nice if we could buy inexpensive emulators that would let us run software written for everybody else's computers, because then our choice of which computer to buy wouldn't be so difficult. It would also be nice if we could buy a Datsun or Renault and repair it with Chevrolet or Ford parts, and vice versa. But realistically, neither is likely to happen for a long time, and for many of the same reasons.

Questions Beginners Ask

Q In your September column ["Questions Beginners Ask"] you suggested using a bulk eraser or an audio recorder to erase tapes. There is a much simpler way that I have been using with my TRS-80 Color Computer, and there's no reason it shouldn't work with any other micro. Why not just rewind the tape, press PLAY and RECORD on the recorder, and type CLOAD "X" ?

Ken McIsaac

A Essentially this is identical to one of the methods I recommended—place the tape in an audio recorder, insert a null plug into the microphone jack or turn down the recording level, and press PLAY and RECORD. Either way, you're erasing the tape by recording silence over the previous material.

However, your method works only on TRS-80 computers; Commodore, Apple, TI, and IBM computers don't have a CLOAD command. Atari BASIC uses CLOAD, but if you try typing CLOAD and pressing RECORD and PLAY, the tape stops after a short while and the computer reports ERROR 138—device timeout. When the Atari detects no program on the tape, it stops the recorder motor. As mentioned in September's column, you can get around this by pressing RECORD and PLAY and entering POKE 54018,52 to start the motor, and POKE 54018,60 to stop the motor.

In any event, these methods are inefficient ways to erase a cassette. It takes a half-hour to erase a C-30, an hour to erase a C-60, etc. A bulk tape eraser does the same thing in a few seconds, and saves wear and tear on your recorder. ©

When Jennifer's parents discovered DesignWare programs, they put Jennifer in a class by herself.

Because unlike most educational software, DesignWare gives Jennifer individualized attention for months on end.

For starters, DesignWare focuses on important skills—the ones Jennifer learns in school. In fact, all DesignWare programs are developed by educators. And tested by kids.

The graphics, sound and game play are unquestionably superior. If they weren't, they couldn't keep learners like Jennifer involved for as long as they do.

DesignWare programs provide multiple levels of challenges, which let Jennifer continue to grow long after she first starts to use each program.

But best of all, Jennifer won't really outgrow a DesignWare program. Because they're designed to let her type in her own questions and problems.

So Jennifer's parents can tailor her DesignWare program to match her homework assignments. Or Jennifer can change her program to challenge her parents.

Jennifer's parents think DesignWare is in a class by itself. So it's not surprising they give her DesignWare. Because they think Jennifer's in a class by herself.

FOR ALMOST EVERY AGE, SUBJECT AND COMPUTER.

DesignWare offers programs for children ages 4 to 16, and for parents of all ages.

They cover a wide range of important subjects including math, geometry, algebra, geography, vocabulary, spelling, grammar, history, computer literacy, and music.

And they run on Apple,® Atari,® Commodore 64,™ IBM® PC and IBM® PC Jr. For the name of your nearest dealer or to order our free catalog, call us toll-free at (800) 572-7767. (415-546-1866 in California.)

DesignWare®

LEARNING COMES ALIVE®

More Ways Computers Made Me Smarter After Only Thirteen Years Of Daily Use

Last month, on the occasion of my third anniversary as a COMPUTE! columnist, I recounted some of the blessings computers have brought me: *Cuisinart-brain thinking* (the ability to process facts by slicing them, dicing them, and mixing them together); an *algorithmic lifestyle* (applying patterned thinking to problems of everyday life, such as how to turn off an unfamiliar shower faucet in a hotel bathroom); *lightning-fast logic* (like the time it took me only 24 hours to realize I was wearing a name badge upside-down); and new-found *mechanical aptitude* (as evidenced by my futile attempts to open up a new portable computer until rescued by my seven-year-old daughter).

But the blessings don't end there. No, 13 years of working with computers have enhanced my life in other ways as well. For example...

Blessing 5: I've Become A Whiz With Robots

My family and I live an "Erma Bombeck lifestyle." That means our house is a mess, our lives are chaotic, and we struggle through each day doing our best just to cope.

But last week was even worse.

Last week a film crew from the PBS program *The New Tech Times* descended on our house to shoot a profile of me and my family (and our 14 robots and 23 computers), and a robot product review.

The film crew arrived Thursday morning and spent the entire day taping program segments all over the house. They filmed in my study, in the dining room, the rec room, the hallways, and in our bedrooms.

At one point, late in the day, my wife Janet came into the house and gasped. She had absolutely forbidden us to shoot in the living room, yet there we were, complete with a dozen robots, giant, aluminum umbrella reflectors for the camera lights, and thick, snakelike cables draped over our new couch.

In total despair, Janet dashed into the room, and swiped up the Christmas cards that had been sitting over the mantelpiece for at least seven months. "I can see there are no wives and mothers on this film crew," she muttered as she stalked out of the room.

Earlier in the week, to get ready for the program, I had panicked and gone "over the top" (as the English say), and tried to get everyone in my life into the show. I had my mother fly in from Pennsylvania to show how she and Catie have become computer pen pals on The Source. I had helped my five-year-old son Eric set up cubbyhole "offices" under his bed and underneath my desk in my study so he could show

how he uses a portable computer to do "gobbledygook processing."

I had organized two dozen neighborhood kids to try to teach one of our robots how to skateboard. We had bought Topo the robot a black cape and programmed him to breakdance with Eric, to the tune of Michael Jackson's "Beat It." We had enlisted the teachers and students in a preschool and two high schools to show how they were programming computers and robots and playing with them. And I had even managed to persuade Olga Pagenhardt, the 70-year-old director of Roanoke's "Programs for Retired People," to be present to show my concern for senior citizens and computers.

To get to all the schools and other sites for filming, we formed a caravan of vans and cars, loaded with people, cameras, computers, and robots, and we wound our way, in a big hurry, around the streets of Roanoke. Robots sat on car seats and on the floors of the vans, and peeked out of every window at fellow motorists and passersby. And each time we turned a sharp corner, a robot would tumble over and lose an arm or bend an antenna.

The house was literally crawling with robots. We had a HEROjr, we had a Talking Topo, we had a F.R.E.D. (Friendly Robotic Educational Device), a Maxx Steele, a Big Trak, an Armatron, and eight little crablike robots that bounced and hobbled their way across our kitchen floor.

The robots were the center of the show, but they were so *finicky* they almost caused me to have a nervous breakdown.

We had a HEROjr, for example, for two weeks before the program. He worked perfectly, he visited the kids at the preschool a couple of times, and he was a lovable addition to our family.

Then, inexplicably, he ceased to function.

To bring him back to life, we tried human-to-robot resuscitation. We tried pulling off his head, taking off his clothes, and everything else we could think of. But no luck. He was in the Robot Happy Hunting Ground, and we couldn't bring him back.

That's just when Topo the robot decided to become a problem. Topo, too, had been A-OK for over a month. Then, in quick succession, he suffered memory lapses, his infrared "eye" stopped working, and, worst of all, his recharger disappeared. Anybody who has ever hung around with robots knows how serious it is when a robot can't find his recharger.

Wednesday afternoon, the day before the TV film crew came, was the worst. Topo wasn't working. HEROjr wasn't working. And we had just gotten a shipment of little robots in the mail, and the most important little robot was broken.

"I give up!" I cried. "I hate robots! I never want to see another robot! Get them out of my sight. I'm going back to bed."

Then Eric came to the rescue.

Eric had just come home from school and walked in on my tirade. In his own breezy, take-charge manner, five-year-old Eric barged into a kitchen filled with computers, robots, and adults, sat down at the table, and began fooling with the broken robot.

A moment later it beeped!

Then its lights came on. And it beeped again.

Then it began jerkily moving around the kitchen table. It crawled. It stopped. It lurched. It stopped. It looked like a tipsy turtle ambling across a fishing boat in high seas.

When I saw the robot work I grabbed Eric and gave him a big kiss and a hug. "I don't believe it," I said. "You fixed it. How did you do it?"

"I just pushed the buttons," Eric said. "Do you have any more robots I can fix?"

Eric's dramatic rescue of the robot turned the whole day around. Within minutes we had found Topo's recharger, and we had lined up a new HEROjr to arrive before the TV crew showed up the following morning.

Eric and I sat on the kitchen floor having little robots bounce, jounce, and try to run up our pantlegs. Once again I was happy. Once again I felt like Fred D'Ignazio, Robot Tamer Extraordinaire.

Blessing 6: I Can Spot A Shortcut A Mile Away

Last spring, I took several computers and robots with me to London, England, to teach a "Robotics Literacy" course. I described my adventures in the October 1983 issue of COMPUTE!, in my column "There's A Robot In My Room."

In that column, I told how I tried to make a HERO robot I had taken with me into a robotic alarm clock that would wake me at 5:30 a.m. each morning so I could prepare the lectures for my students before class.

I went to extreme lengths to get HERO to become an alarm clock. I positioned him perfectly, right beneath my bedroom window. I wrote a program in hexadecimal and loaded it into HERO by punching the buttons on top of his head. I activated his light sensor, so he could look out the window and watch the sun coming up, then wake me just after dawn. The sunlight was supposed to trigger his light sensor, which in turn would trigger HERO to start talking. "Good morning, Fred. Time to wake up. Get out of bed, you sleepyhead," he was supposed to say. "It's 5:30 a.m."

And he did say it. But he didn't say it at 5:30 the next morning. He said it at 11:00 p.m.

You bought a computer to cultivate your kids' minds. Make sure it's bearing fruit, not growing vegetables.

Introducing a whole crop of Learning Adventure games from Spinnaker.

When it comes to cultivating adventurous young minds, the computer's potential is endless.

Unfortunately, the search for software that makes the most of that potential has been endless, too.

That is, until Spinnaker created the Learning Adventure Series. A unique collection of games that reward curiosity with hours of adventure and learning. So the time kids spend with our games will help them develop valuable skills. Instead of just tired thumbs.

But what really makes our Learning Adventure games unique – educational value aside – is how much fun they are. Which isn't too surprising when you consider you can do things like bargain with aliens, search a haunted house, or build your own railroad empire.

In fact, our games are so much fun, kids will really enjoy developing some very important skills. Deductive reasoning, note taking, and problem solving, for instance.

So, if you're in the market for software that will truly cultivate young minds, pick the Spinnaker Learning Adventure Series.

It's the best way to be sure your search will be fruitful.

Spinnaker Learning Adventure games are available for Apple,® Atari,® IBM® and Commodore 64™ home computers.

It's new! PRESIDENT'S CHOICE.™ Welcome to the White House – you're the President of the United States! Make the right decisions and you'll win re-election. **Ages 13-Adult.**

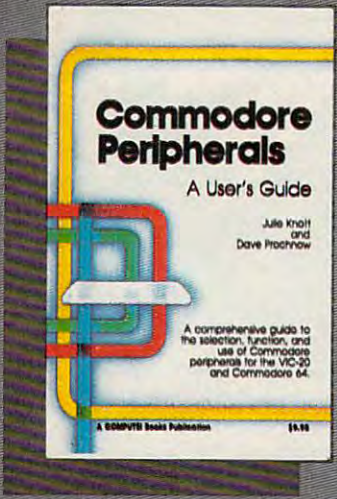
It's new! ROCK 'N' RHYTHM.™ It's your own recording studio, complete with instruments and equipment. Play and record existing music, or experiment with your own melodies and rhythms. **Ages 10-Adult.**

It's new! TRAINS.™ You're in charge of an old-time railroad – and whether it turns into a bonanza or a bust depends on how well you run it. But either way you'll find that working on this railroad is a challenge – and a lot of fun! **Ages 10-Adult.**

SPINNAKER™
We make learning fun.

Disks for: IBM (PRESIDENT'S CHOICE), Atari and Commodore 64 (ROCK 'N' RHYTHM and TRAINS).

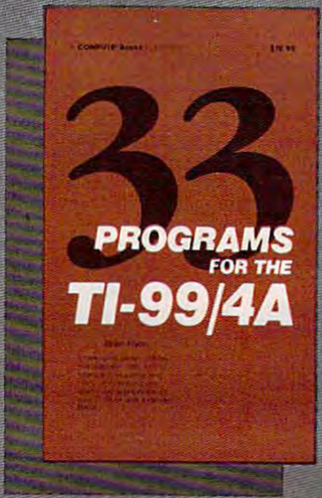
COMPUTE! BOOKS



Commodore Peripherals: A User's Guide
Julie Knott and David Prochnow

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

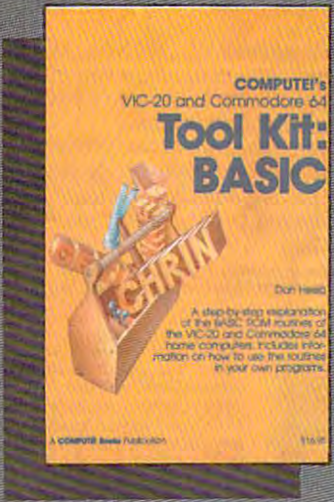
$9.95
ISBN 0-942386-56-6



33 Programs for the TI-99/4A
Brian Flynn

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

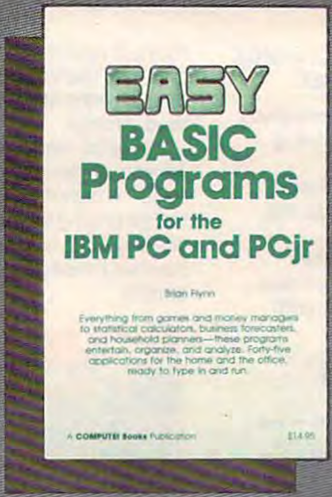
$12.95
ISBN 0-942386-42-6



VIC-20 and Commodore 64 Tool Kit: BASIC
Dan Heeb

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

$16.95
ISBN 0-942386-32-9



Easy BASIC Programs for the IBM PC and PCjr
Brian Flynn

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

$14.95
ISBN 0-942386-58-2

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

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

Finally, a computer keyboard kids can use.

A computer can help your child learn but the keyboard often gets in the way. It's a jumble of keys that's confusing and hard for little fingers to operate. And it's not much fun.

Introducing Muppet Learning Keys from Koala Technologies.

The first computer keyboard made especially for young children. And the *only* keyboard with Kermit, Miss Piggy and the Muppet gang right on it—ready to introduce your child to the magic of letters, numbers, and colors.

Imagine you're five years old. Now pick a keyboard.

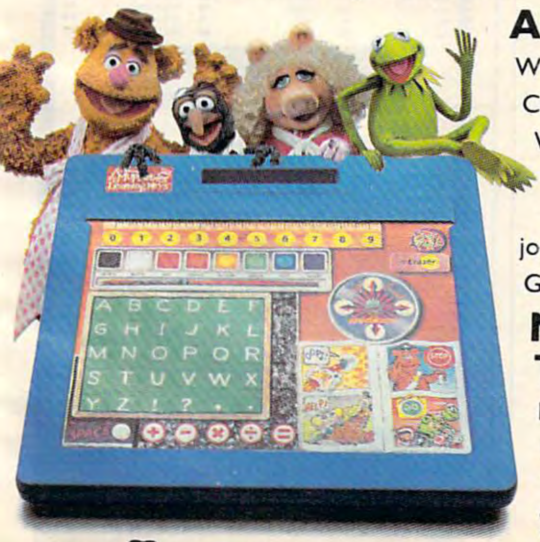
An easy choice, isn't it? That's because Muppet Learning Keys was created by education specialists to make learning an adventure for your child.

Unlike conventional keyboards, all the letters and numbers are in order. So a child can find A-B-C and 1-2-3 without hunting all over the keyboard.

Press any key on the keyboard and something always happens. Kermit flies a kite. Miss Piggy eats a pretzel. Fozzie puts out a fire.

That's how the fun begins. But soon, your child starts to explore and experiment. How many kites can Kermit fly? In how many colors? What do the other letters mean?

Muppet Learning Keys has things that every child knows and loves—a compass, a ruler, an eraser and a blackboard. Lots of stuff that a kid can't wait to get his hands on.



All-consuming tests.

When it came time to test this new marvel, we turned to the experts. Children.

We let them do their worst to it. Peanut butter. Teeth. Even Ketchup.

Then we wiped its washable mylar surface with a sponge, and plugged it in.

And those kids did their best with it—having fun while they experienced the joy and wonder of learning.

Give your child Muppet Learning Keys and make computer learning child's play.

Muppet Learning Keys. The hands-on keyboard for kids.

For the Apple IIc, Apple IIe, and Commodore 64 computers.

In-box software by Sunburst Communications. Muppet Learning Keys works with software that is designed or modified for it.

Jim Henson's Muppet Learning Keys™

and again at 11:45 p.m. The second time he said it, I was sound asleep. And I was not happy to be wakened—especially by a robot.

Of course it was not his fault. I had programmed his sensor to be so sensitive to light that even the tiniest amount of light would set off his robotic alarm clock. The first time he went off, his sensor had detected my bedside reading lamp. The second time, it responded to the headlight of a passing truck.

After these two experiences, I lowered the sensitivity of the sensor, went back to bed, and happily slept through the night.

The next morning I heard banging on my door. It was my colleague at the robotics course, and it was 8:30 in the morning. I had overslept by *three hours.*

Why hadn't HERO awakened me? Later in the day, when I had had time to get dressed, teach a course, and grab a late breakfast, I calmed down enough to realize that I had now erred in the opposite direction. This time I had set HERO's sensor *too low*. It was now so low that only a supernova would turn him on.

When my article appeared in COMPUTE!, I received dozens of nice letters. The readers loved my stories about living with robots, and they wanted to know more about the robotics literacy course.

Then, one day, I received a letter from a nine-year-old boy. He had enjoyed my article, too, he said. But he had one small question. Why, he asked, if HERO had a clock built inside him, couldn't I have written a little program to have HERO check his clock, and at 5:30 a.m. start talking and wake me up?

I wrote the boy a letter and answered his question simply and truthfully. "I didn't think of it," I said.

I showed my wife the boy's letter. That's when she christened me "Do It the Hard Way, Fred."

I wince when I admit it, but it is a perfect nickname. After all, I have spent thirteen years programming and working with computers and robots. I have been on intimate terms with machines of all sizes and personalities. But am I any smarter? Have computers made me a quicker, more logical thinker?

Judging from my experience with HERO in London and from all my other experiences, the answer is no. And it took a letter from a sharp little nine-year-old boy to make me realize it.

"Put your heart and soul into computing," I advised him in my letter, "and, one day, you may be smart just like me." ©

KUKULCAN™ ADVENTURE

A history "course" that's fun.

What an entertaining way to learn about Aztec history. The year - 1519. The place - Tenochtitlan, the City State of the Aztecs. You are one of the citizens, an Aztec scribe. Your leader - Montezuma II. The great city falls prey to evil omens. Montezuma calls for his scribes. You must go. This is your chance to earn a place in history.

$39.95

A Class 4 (experienced) adventure game with outstanding graphics. Created by Michael Glitzow. For 48K Apple][,][+, Commodore 64.

Available at your local computer or software dealer. Or direct from us. Also call or write for the complete catalog.

Apple, Apple][, Apple][+ are registered trademarks of Apple Computer, Inc. Commodore 64 is a registered trademark of Commodore Business Machines, Inc.

AMERICAN EAGLE SOFTWARE, INC.

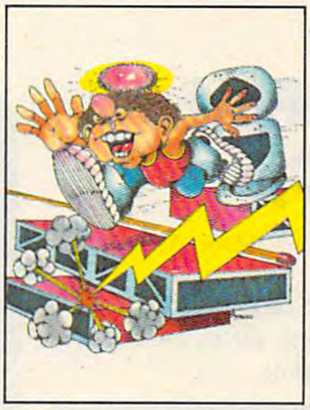
P.O. Box 46080
Lincolnwood, IL
60646
(312) 792-1227

©1984 American Eagle Software, Inc.

OLLIE'S FOLLIES™

You've enjoyed Jumpman, Donkey Kong, Lode Runner. Now comes Ollie's Follies™.

24 fast-action boards make you wonder if you'd ever master the game. Innovative, fun-filled tricks - elevators, blowers, teleporters, springboards, high-wire tricks, laser walls, lightning bolts, robots. Ollie's Follies can get your adrenalin going like any of the hit games.



$34.95

Created by Frank Cohen. For Commodore 64 and 48K Atari.

Available at your local computer or software dealer. Or direct from us. Also call or write for the complete catalog.

Jumpman is a registered trademark of Epyx Software. Donkey Kong is a registered trademark of Nintendo of America, Inc. Lode Runner is a registered trademark of Broderbund Software. Commodore 64 is a registered trademark of Commodore Business Machines, Inc. Atari is a registered trademark of Atari, Inc.

AMERICAN EAGLE SOFTWARE, INC.

P.O. Box 46080
Lincolnwood, IL 60646
(312) 792-1227

©1984 American Eagle Software, Inc.

LUCIFER'S REALM™ ADVENTURE

Pits you against history's most evil characters.

The good guy vs. the bad, and the bad don't come any badder. Hitler, Mussolini, Reverend Jim, Satan himself. You have to go against them all, just so that you can escape from hell (yes, the real thing).

$39.95

A Class 5 (expert) adventure game with outstanding graphics. Created by Jyym Pearson and Norm Sailer. For 48K Apple][,][+, Commodore 64, 48K Atari.

Available at your local computer or software dealer. Or direct from us. Also call or write for the complete catalog.

Apple, Apple][, Apple][+ are registered trademarks of Apple Computer, Inc. Commodore 64 is a registered trademark of Commodore Business Machines, Inc. Atari is a registered trademark of Atari, Inc.

AMERICAN EAGLE SOFTWARE, INC.

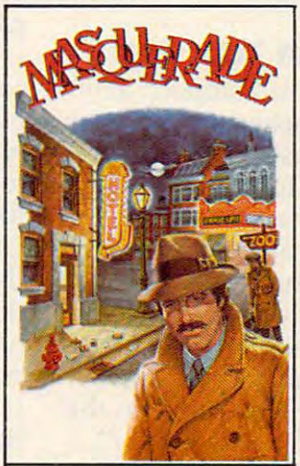
P.O. Box 46080
Lincolnwood, IL
60646
(312) 792-1227

©1984 American Eagle Software, Inc.

MASQUERADE™ ADVENTURE

"an adventure puzzle solver's piece de resistance...," Softalk, Nov. '83.

It turned out to be the toughest case of your detective career. Clues have led to nothing but dead ends. Meanwhile, the crime boss you are after is still operating from somewhere. You're about to throw in the towel. But wait! Something breaks. This could be the lead to solve the case. Maybe.



$34.95

A Class 5 (expert) adventure game with outstanding graphics. Created by Dale Johnson. For 48K Apple][,][+, Commodore 64.

Available at your local computer or software dealer. Or direct from us. Also call or write for the complete catalog.

Apple, Apple][, Apple][+ are registered trademarks of Apple Computer, Inc. Commodore 64 is a registered trademark of Commodore Business Machines, Inc.

AMERICAN EAGLE SOFTWARE, INC.

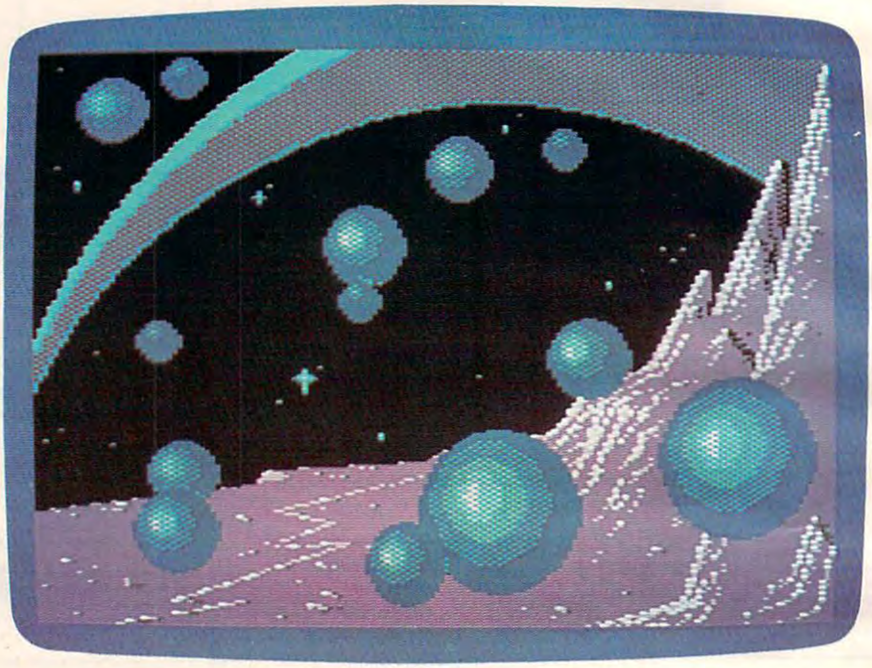
P.O. Box 46080
Lincolnwood, IL
60646
(312) 792-1227

©1984 American Eagle Software, Inc.

KoalaPad For PCjr

Lee Noel, Assistant Editor, Art & Design

Requirements: IBM PCjr with 128K RAM, disk drive, DOS 2.1, and color display.



"Bluemoon," an imaginary world made with the KoalaPad and KoalaPainter. *The blue spheres were designed on the alternate screen and then copied to the main screen. (All works reproduced here, except the fox, were created by author Lee Noel.)*

Long before I picked up the new PCjr KoalaPad, I was using the KoalaPad for the Commodore 64. In combination with its accompanying *KoalaPainter* program, it was—and still is—the most flexible and satisfying computer graphics tool I've ever used.

After testing the PCjr KoalaPad package for more than a month, I'm pleased to report that it's virtually identical to the Commodore 64 version. The essential features have all been retained, and the few differences are improvements.

A Screen In Your Lap

The KoalaPad is compact: about six by eight inches overall, with a pressure-sensitive drawing surface about four inches square. There are two large black buttons at the top. It's sturdily constructed of plastic and should withstand moderate abuse from children.

The KoalaPad plugs into one of the joystick ports on the rear of the PCjr with a 5-1/2 foot cord, and it works on the same principle as a game controller. But instead of reporting the position of a movable joystick, the KoalaPad reads the position of an object pressing against the pressure-sensitive membrane. (This can be the plastic stylus supplied with the package or even a finger.)

The program running in the computer then converts this input into some meaningful form. Not unexpectedly, the two buttons at the top of the pad act like fire buttons on a joystick. Unlike the joystick buttons, though, both buttons on the pad do exactly the same thing, so it never matters which one you press. Two buttons make the pad convenient for both left- or right-handed people.

It's up to the program to make use of the pad's readings. You could write your own programs to work with the KoalaPad, or even use it with some existing software. Of course, the pad comes with its own graphics program, *KoalaPainter*. It treats the pressure-sensitive surface as a representation of the computer's display screen. Pressing a certain point on the pad's surface causes a lighted arrow to appear at a corresponding point on the screen. Moving across the surface as you press also moves the arrow across the screen. Removing all pressure from the pad makes the arrow disappear. In computer jargon, the arrow is a cursor.

However, pressing two or more places at once does not make two arrows appear onscreen. The KoalaPad is designed to recognize only one contact at a time; more merely confuse it. Therefore, you have to keep stray fingers off the

Flight Simulator II

For Apple, Atari, & Commodore 64™

Put yourself in the pilot's seat of a Piper 181 Cherokee Archer for an awe-inspiring flight over realistic scenery from New York to Los Angeles. High speed color-filled 3D graphics will give you a beautiful panoramic view as you practice takeoffs, landings, and aerobatics. Complete documentation will get you airborne quickly even if you've never flown before. When you think you're ready, you can play the World War I Ace aerial battle game. Flight Simulator II features include ■ animated color 3D graphics ■ day, dusk, and night flying modes ■ over 80 airports in four scenery areas: New York, Chicago, Los Angeles, Seattle, with additional scenery areas available ■ user-variable weather, from clear blue skies to grey cloudy conditions ■ complete flight instrumentation ■ VOR, ILS, ADF, and DME radio equipped ■ navigation facilities and course plotting ■ World War I Ace aerial battle game ■ complete information manual and flight handbook.

See your dealer . . .

or write or call for more information. For direct orders please add $1.50 for shipping and specify UPS or first class mail delivery. American Express, Diner's Club, MasterCard, and Visa accepted.

Order Line: 800/637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

membrane. (This is a major difference between the KoalaPad and a competitive touch tablet, the PowerPad from Chalkboard, which can distinguish multiple contact points and report them all to the computer.)

The KoalaPad is easiest to use with its special stylus. The stylus moves smoothly across the surface and greatly aids accurate work. However, just about anything can be used as a stylus. Children often prefer just to press with a finger, in a fashion similar to fingerpainting. But even though the membrane is very durable, you should avoid sharp objects that might permanently damage it.

Easy To Run

Before using the *KoalaPainter* diskette for the first time, you must transfer parts of DOS 2.1 onto it. The procedure is very well explained in the manual, even for beginners. Once it's done, you'll never have to do it again; you boot up with this disk each time.

All you have to do is plug the pad into the rightmost (as you face the system unit) joystick port, then insert the prepared disk and turn on the computer. Everything runs automatically.

The *KoalaPainter* menu denotes various functions with tiny pictures on the screen, sometimes called *icons*. Even nonreaders can use this program. The menu is well-designed and extremely easy to understand. For example, a picture of two circles indicates the CIRCLE function, which lets you draw circles. It's that simple.

To activate the CIRCLE function—or any similar option—you just move the stylus across the pad so the screen arrow points inside the box containing the appropriate icon. Then press either of the KoalaPad's fire buttons. At once, the box changes color, verifying

This finely crafted picture of a red fox is included on the KoalaPainter *disk.*

your choice. The function remains active until you select another one.

Of course, you don't want to draw pictures over the menu, so the program has a blank screen ready. To reach it, stop pressing against the pad and press one of the buttons. Instantly, the blank screen appears. You can return to the menu anytime by removing pressure and pressing the button again. That's basically all there is to operating the system.

The KoalaPad system shows how easy computers can be. There's rarely a need to touch the keyboard, and there's no mysterious new computer language to learn. Naturally, taking advantage of *KoalaPainter*'s many functions to create a detailed picture does take some practice, but you can start experimenting immediately. Thanks to the excellent manual (despite a few typographical errors), the task of learning the program is simplified considerably.

"Janus" shows one application of the KoalaPainter *mirror mode.*

Low-Resolution Graphics

For some reason, *KoalaPainter* only uses the PCjr's low-resolution 160 × 200 pixel graphics mode (SCREEN 3 in Cartridge BASIC). This mode does offer 16 simultaneous colors, but so does one of the medium-resolution (320 × 200) modes. The greater the resolution, the more detailed the pictures you can draw.

Koala Technologies says it chose low-resolution graphics to conserve memory and to insure a clear display on ordinary TV sets and composite color monitors. While the latter point makes some sense—TVs may indeed have difficulty with the finer display—it's hard to understand why memory was a factor. The PCjr version of *KoalaPainter* requires 128K RAM, more than enough memory to support all the requirements of the program. The Commodore version runs easily in 64K, and similar programs for other computers can run in as little as 16K. Koala says the decision to use the low-resolution mode was a judgment call, but it seems a shame that *KoalaPainter* cannot take full advantage of the PCjr's excellent display capabilities.

Of course, a lot can be done even with 160 × 200 graphics, as the accompanying screen photos prove. The picture of the fox, one of several demos on the *KoalaPainter* disk, contains plenty of detail. And the resolution is comparable to that offered by competing graphics packages for other home computers.

Still, it would be nice to have the highest full-color resolution the PCjr can support. Perhaps some RAM could be conserved if Koala supplied *KoalaPainter* on ROM cartridge instead of disk. That would, however, increase production costs, and therefore the retail

price. As Koala says, it's a judgment call.

A Hundred Shades

You can actually simulate more than 16 colors with *KoalaPainter*. Although the PCjr is limited to displaying only 16 true colors—or, more accurately, two shades of eight colors—you can mix any two colors into a halftone dot pattern. This creates the effect of different shades. Further, there's a special new rainbow color feature that mixes lots of colors together for you—children should love this. All in all, there are more than a hundred shades and combinations to choose from.

The border on the drawing screen changes to match the drawing color selected. Otherwise, it's easy to try to draw in, say, white on a white background. Without the border, you might wonder for a long time why nothing was happening. However, the border can display only a single solid color—not the rainbow color or the halftones.

In addition to all the colors, *KoalaPainter* gives you eight different types of brushes. You can use a fine brush for drawing, a broad brush for rapid painting, a brush that makes several parallel lines, a brush that letters in *italics*, and more. Of course, the brushes are just a software illusion—you always use the same stylus or fingertip.

Drawing Options

Once you've selected your color and brush, you choose one of the drawing commands from the menu. There's everything from a freehand drawing mode to a Zoom command that magnifies a section of your picture so you can make precise changes to any dot of color on the screen. Here's a summary of *KoalaPainter* drawing options:

• Line, Lines, and Rays let you draw perfectly straight lines in various ways. They use a technique called *rubber-banding*. First, you set an initial point by pressing the stylus against the pad and hitting the button. Then, maintaining pressure, you move the stylus. On the screen, you'll see a flashing line stretching back to the original point from the current stylus position. As you continue to move the stylus, the line moves to accommodate the new position. This looks a lot like a rubber band being stretched over the screen, hence the term.

When you're happy with the length and direction of your line, you press the button a second time. Then you can move the line (now of fixed length and direction) to any area of the screen. Pressing the button a third time locks the line in place. If you wish, you can continue putting identical lines all over the screen just by pressing the button repeatedly.

• Frame, Box, Circle, and Disk draw just what you'd expect. (A box is a solid frame and a disk is a solid circle.)

• Mirror lets you draw symmetrically in three different ways. For example, one of the mirror modes produces a top-bottom mirror image—anything you draw in the top half of the screen appears simultaneously, but inverted, at the bottom of the screen, and vice versa. You could draw a reflection in a pond or some other flat, shiny surface with about half the effort you might otherwise expect.

• Fill is another readily grasped command. If you've got a child who loves coloring books but is frustrated by not being able to color pictures evenly, Fill will come as a blessing. (They can't draw on walls with it, either.) You can fill any completely closed outline with any color. Using it is really child's play: Select the Fill command and a color from the menu, move to the picture screen, put the cursor inside the closed outline, and press the button.

The *KoalaPainter* Fill command is not only fast, but also the most thorough I've seen in any graphics package. While Fill (sometimes called Paint) commands are common to lots of graphics programs, they are sometimes fooled by complex

"Icarus," the mythical Greek who flew too close to the sun.

ATARI

ATARI is a trademark of ATARI, INC.

Atari Inc. has cut all hardware and software prices. Please call for latest current prices.

Bring the trivia craze home with P.Q. The Party Quiz Game for the Atari and the CBM 64 . . . . . Call

COMMODORE 64

CBM 64 . . . . Call
SX-64 Portable . . . . Call
1541 Disk Drive . . . . Call
1526 Printer 279.00
1530 Datasette 66.00
1702 Monitor . . . . Call
1650 AD/AA Modem 89.00
RS 232 Interface . . . . Call

SUPERPRINTER PACKAGES

Gemini 10X and Apeface 328
Prowriter and Apeface 409
Prowriter + Aid Interf. + Cable 485
Gemini 10X and Cardco + G 333
Prowriter and Cardco + G 414

No additional ship. charges on printer packages in Continental USA

STAR MICRONICS PRINTERS

Gemini 10X . . 259
Gemini 15X . . 389
Delta 10X 429
Delta 15X 579
Radix 10X 579
Radix 15X 699
Powertype . . . 339

PRINTERS

Alphacom 40C w/interface . 99.95
Alphacom 80C w/interface 189.95
Axiom AT-550 . . . . 279.00
Epson . . . . Call
Prowriter I . . . . Call
Riteman . . . . . . Call
Silver Reed . . . . Call
Toshiba 1351 . . Call
Toshiba 1340 . . Call

MODEMS

Hayes Smart Modem 300 . . . Call
Mark II 79.00
Mark VII/Auto Ans/ Auto Dial . . . . . Call
Mark XII/1200 Baud . . . . Call
MPP 1000 C . . . . Call
R-Verter Modem Adaptor 39.95
Prometheus . . . . Call

INTERFACES

Aid Interfast I . . . . . . Call
Ape Face . . . . Call
R-Verter Modem Adaptor 39.95
MPP 1150 . . . . Call

DISK DRIVES

Indus GT . . . . . . Call
Percom . . . . Call
Trak . . . . Call
Astra 2001 . . . . . Call

MOSAIC

48K RAM 94.00
64K RAM/400 . 149.00
64K RAM/800 + Cable Kit #1 . . 169.00
64K Expander for 600 XL 99.95

ATARI SOFTWARE

ACCESSORIES

Ape-Link 29.95
Big Foot 16K Buffer . . . . . Call
Big Foot 32K Buffer . . . . . Call
Big Foot 64K Buffer . . . . . Call
Gemini 10X 8K Upgrade . . Call
Koala Pad - D 69.95
Koala Pad - Cart 74.95
Humpty Dump - D 29.95
Monitors . . . . Call
Compuserve Starter . . . 27.95
Elephant SS/SD 17.00
Verbatim SS/DD 20.00
Elephant SS/DD 20.00
Triangle Replacement Keyboard for 400 . . . . . 54.95
Alien Voice Box II - D . . . 99.95
Omnimon 82.95

BRODERBUND

Bank Street Writer - D . 49.95
Choplifter - D 23.95
Drol - D 23.95
Gumball - D 23.95
Loderunner - D 23.95
Mask of the Sun - D . . . 27.95
Operation Whirlwind - D 27.95
Spelunker - D 23.95
Stelth - D 23.95
Whistler's Brother - D . . 23.95

DATASOFT

Bruce Lee - D/T 23.95
Dallas - D 23.95
Heathcliff - D/T 23.95
Letter Wizard - D 34.95
Micropainter - D 23.95
Spell Wizard - D 34.95
Juno First - D/T 20.95
Lost Tomb - D/T 27.95
Mancopter - D/T 27.95
Meridan III - D/T 27.95

ELECTRONIC ARTS

Archon - D 29.95
Pinball Construction - D 29.95
M.U.L.E - D 29.95
Murder/Zinderneuf - D 29.95
One On One - D 29.95
Archon II - D 29.95
Financial Cookbook - D . 37.95
Music Construction - D . 29.95

EPYX

Dragons/Pern - D 27.95
Fun With Art - Cart 27.95
Gateway to Apshai - Cart27.95
Jumpman Jr. - Cart 27.95
Jumpman - D/T 27.95
Pitstop II - Cart 27.95
Puzzlemania - D 23.95
Summer Games - D 27.95
Temple of Apshai - D/T27.95

INFOCOM

Deadline - D 29.95
Enchanter - D 23.95
Infidel - D 34.95
Planetfall - D 24.95
Sorcerer - D 34.95
Starcross - D 29.95
Suspended - D 29.95
Witness - D 34.95
Zork I, II or III - D 27.95
Sea Stalker - D 24.95
Cutthroats - D 24.95

MISCELLANEOUS ATARI

Zombies - D 23.95
Codewriter - D 69.95
Star League Baseball D/T 22.95
Star Bowl Football - D/T 22.95
Death in the Caribbean - D 27.95
Dino Eggs - D 27.95
The Heist - D 23.95
Gyruss - Cart 34.95
Star Wars - Cart 34.95
Master Type - D/Cart . 27.95
Flight Simulator II - D . . 37.95
S.A.M. - D 41.95
Castle Wolfenstein - D . 20.95
Compuserve Starter Kit 34.95
Home Accountant - D . . 49.95
Ken Uston's Blackjack - D 49.95
Megafont - D 19.95
Monkey Wrench II - Cart37.95
Movie Maker - D 41.95
Pogo Joe - D 20.95
Ultima III - D 41.95
Jupiter Mission - D 34.95
Boulder Dash - D/T 20.95
Scraper Caper - Cart . . . 34.95
Miner 2049'er - Cart . . . 34.95
Beachhead - D/T 23.95
Millionaire - D 37.95
Spy vs. Spy - D 23.95
Lifestyle - D 34.95
MPP Modem Driver - D . 19.95
Microfiler - Cart 34.95
Microcheck - D 34.95
Montezuma's Revenge - Cart 34.95
Mr. Do's Castle - Cart . . 34.95
Frogger II - Cart 34.95
Net Worth - D 54.95
Stickybear - D 27.95
Fischer Price - Cart 19.95
Windham Classics - D . . 34.95
Omnitrend Universe - D 69.95
Raid Over Moscow - D . 27.95
H.E.R.O. - Cart 31.95
Decathlon - Cart 31.95

MISC. ATARI (cont'd.)

Dark Crystal - D 27.95
Ultima I - D 23.95
Ultima II - D 41.95
Letter/Spell Perfect - D 74.95
Harcourt/Bruce S.A.T. - D 59.95

OPTIMIZED SYSTEMS

Action - Cart 69.95
Basic XL - Cart 69.95
MAC/65 - Cart 69.95
DOS XL - D 27.95
Action Aid - D 27.95
C65 - D 59.95
Handy-Writer - D 89.95
Postal Tool - D 41.95
Print Tool - D 41.95

SSI

Carrier Force - D 41.95
Combat Leader - D/T . . 27.95
Cosmic Balance II - D . . 27.95
Cosmic Balance - D . . . 27.95
Broadsides - D 27.95
War in Russia - D 55.95
50 Mission Crush - D . . 27.95
Questron - D 34.95
Rails West - D 27.95
Bomb Alley - D 41.95
Computer Ambush - D . 41.95
Galactic Adventures - D 41.95
Computer Baseball - D . 27.95
Reforger 88 - D 41.95
Objective Kursk - D . . . 27.95

SYNAPSE

Synfile - D 54.95
Syncalc - D 54.95
Syntrend - D 54.95
Synchron - D 29.95
Syncomm - D 29.95
Synstock - D 59.95
Relax - D 99.95
Alley Cat - D/T 23.95

D - Disk
T - Cassette
Cart - Cartridge

To Order Call Toll Free
800-558-0003
For Technical Info, Order Inquiries, or for Wisc. Orders -
414-351-2007

COMMODORE 64 SOFTWARE

ACCESS

Neutral Zone - D/T . . . 23.95
Spritemaster - D/T . . . 23.95
Beachhead - D/T 23.95
Master Composer - D . . 27.95
Raid Over Moscow - D/T27.95
Scrolls of Abadon - D/T 23.95

ACTIVISION . . . . Call

BATTERIES INCLUDED

Consultant - D 69.95
Paperclip w/Spellpak - D 84.95
Super Busscard II . . . . . . Call
Home Inventory - D 23.95
Recipe - D 23.95
Audio/Video Cat - D . . . 23.95
Mail List - D 23.95
Stamps - D 23.95
B.I. 80 Card . . . . Call

CARDCO

Cardprint/B 47.95
Cardco * G 64.95
Cardboard/5 . . . . 59.95
Cardkey 39.95
Cassette Recorder 37.95
Printer Utility - D/T . . . 19.95
Write Now - Cart 34.95
Mail Now - D 29.95
File Now - D 27.95
Graph Now - D 27.95
Spell Now - D 27.95
LQ-1 Printer . . . . Call
LQ-2 Printer . . . . Call
LQ-3 Printer . . . . Call
RS-232 Interface . . . . . . Call

COMMODORE

Assembler - D 39.95
Easy Finance I, II, III, IV - D 19.95
Easy Calc - D 64.95
Easy Mail - D 19.95
Easy Script - D 44.95
Easy Spell - D 19.95
Logo - D 57.95
The Manager - D 39.95
General Ledger - D 39.95
Accts. Rec. - D 39.95
Accts. Pay - D 39.95
Magic Desk - D 42.95
Zork I, II or III - D 29.95

COMMODORE (cont'd.)

Suspend - D 29.95
Starcross - D 29.95
Deadline - D 29.95
Soccer - Cart 29.95
Magic Voice 54.95
International Soccer - Cart 22.95
Cutthroats - D 29.95
The Suspect - D 29.95

ELECTRONIC ARTS — See Atari section for items and prices

EPYX

Barbie - D 27.95
G.I. Joe - D 27.95
Hot Wheels - D 27.95
Impossible Mission - D . 27.95
World's Greatest Baseball - D 27.95

HANDIC

64 Forth - Cart 29.95
64 Graf - Cart 23.95
Stat 64 - Cart 23.95
Calc Result Easy - Cart . . 34.95
Calc Result Adv - Cart . . 69.95
The Diary - Cart 23.95
The Tool - Cart 29.95

INSTA (CIMMARON)

Insta-Writer - Cart 39.95
Insta-Mail - D 24.95
Insta-File - D 49.95
Management Combo . . . 64.95
Insta-Calc - Cart/D . . . 31.95
Insta-Graph - D 24.95
Insta-Vestor - D 31.95
Insta-Speed - D 99.95
Insta-Sched - Cart/D . . 49.95
Insta-Music - Cart/D . . 79.95
Invest Combo 74.95

MISC. COMMODORE

Ken Uston's Blackjack - D 49.95
Quick Brown Fox - D/Cart34.95
Ultima III - D 41.95
Flight Simulator II - D . . 37.95
Night Mission/ Pinball - D/T 20.95
Home Accountant - D . . 49.95
Step By Step - D/T 44.95
Barron's Sat. - D 67.95
Bristles - D/T 20.95
Telestar 64 - Cart 37.95
Star League Baseball - D/T 22.95
Castle Wolfenstein - D . 20.95
Mastertype - D/Cart . . 27.95
Aztec - D 27.95
Strip Poker - D 23.95
Astro Chase - D/T 20.95
Flip Flop - D/T 20.95

MISC. COMM. (cont'd.)

Beyond Wolfenstein - D 23.95
Sam - D 41.95
Mae Assembler - D 49.95
Jupiter Mission - D 34.95
Spelunker - D 23.95
Stelth - D 23.95
Bungling Bay - D 20.95
Dr. Creep - D 20.95
Adventure Writer - D . . . 37.95
Speedwriter - D 49.95
Bruce Lee - D/T 23.95
On-Field Tennis - D/T . 22.95
Graphics Basic - D 23.95
Air Rescue I - D/T 23.95
Challenger - D/T 23.95
F-15 Strike Eagle - D/T 23.95
Movie Maker - D 41.95
Data Manager II - D . . . 34.95
Word Writer - D 34.95
Millionaire - D 37.95
Boulder Dash - D 23.95
VIP Terminal - D 39.95
Westridge Modem 79.95
Doodle - D 27.95

PARKER BROS. . . . . . Call

SSI . . . . Call

SPINNAKER

Adventure Creator - Cart 27.95
Alf in the Color Caves - Cart 22.95
Alphabet Zoo - Cart 23.95
Delta Drawing - Cart . . . 27.95
Facemaker - Cart 22.95
Fraction Fever - Cart . . . 22.95
Grandma's House - D . . 23.95
Kids on Keys - Cart 22.95
Kidwriter - D 22.95
Kindercomp - Cart 22.95
Search/ Amazing Thing - D . . . 27.95
Snooper -1 - D 30.95
Snooper -2 - D 30.95
Story Machine - Cart . . 22.95
Trains - D 27.95

SYNAPSE

Blue Max - D/T 23.95
Fort Apocalypse - D/T . 23.95
Necromancer - D/T . . . 23.95
New York City - D/T . . 23.95
Quasimodo - D/T 23.95
Rainbow Walker - D/T . 23.95
Relax Stress Reduction Sys. 99.95
Zaxxon - D/T 27.95

TIMEWORKS . . . . Call

TOUCH TABLETS

Koala Pad - D 69.95
Koala Pad - Cart 74.95
Animation Station - D . . 59.95

NEW LOWER PRICES

ComputAbility™

NEW PRODUCTS

NO SURCHARGE FOR MASTERCARD OR VISA

ORDERING INFORMATION. Please specify system. For fast delivery send cashier's check, money order or direct bank transfers. Personal and company checks allow 2 weeks to clear. Charges for COD are $3.00. School Purchase Orders welcome. In CONTINENTAL USA, include $3.00 shipping per software order. Include 3% shipping on all Hardware orders, minimum $3.00. Mastercard & Visa please include card # and expiration date. WI residents please add 5% sales tax. HI, AK, FPO, APO, Canadian orders — add 5% shipping, minimum $5.00. All other foreign orders, please add 15% shipping, minimum $10.00. All goods are new and include factory warranty. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call 414-351-2007 to obtain an RA# or your return will NOT be accepted for replacement or repair. Prices and availability are subject to change without notice.

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

ORDER LINES OPEN
Mon-Fri 11 AM - 7 PM CST
Sat 12 PM - 5 PM CST

outlines and fail to fill them completely. With *KoalaPainter*, as long as the outline is fully closed and the fill color doesn't conflict with the background color, Fill works flawlessly.

There are limitations, though. Remember that Fill actually "looks" for the border around the space it's supposed to fill. So you can't refill an area already filled with a halftone, for example; every point in a halftone pattern is surrounded by four others in a contrasting color. However, there are ways to get around this (see below).

• Copy lets you select a portion of your picture and transfer it anywhere on screen. You can draw a detail once and use it over and over again.

• Swap lets you flip between two drawing screens, completely independent of each other. Apart from the obvious fun of working on two pictures at once, there are a couple of interesting applications that come to mind for this command. For one thing, you could reserve one screen for your picture, keeping the other as a work area where you create details for the main picture. Once the details are complete, the Swap and Copy commands can transfer the images from the work screen to the main screen. You can test and perfect certain elements of your picture before committing them to the main screen.

I've used some expensive graphics software that makes a big deal out of allowing you to create individual picture/detail libraries for various purposes. *KoalaPainter* provides the same capability with Swap and Copy, especially when used with the Zoom and Mirror modes.

Another (as yet untested) application for Swap is cartoon-like animation. You could copy an entire picture from one screen to the other, making the slight changes required to produce the illusion of movement. This process could be repeated until the desired sequence was completed. The resulting frames could be filmed or videotaped.

"Moons." The night has a thousand eyes.

Oops-Proof

• The Erase command blanks out a hopelessly botched screen, but only the one you were looking at most recently. The second screen is safe and can be reached with the Swap command.

• Oops fixes your minor mistakes. It nullifies the effect of your last menu selection. This is a very useful feature, and it's missing from lots of graphics software. Let's say you want to test a certain fill color in part of your picture. Just go ahead. If you don't like the way it looks, *immediately* go back to the menu and select Oops. The picture is restored exactly as it was before. But you have to use Oops immediately. If you fill, then select Draw, and suddenly decide you don't like the fill color, Oops can't rescue you. It would work only on the more recent Draw selection.

By the way, this is one method of remedying an unsatisfactory halftone fill. You just use Oops and try again with a different color.

• X-Color may seem a little weird at first, but it's a command I use a great deal. Amazingly, it will change *all* areas of a certain color in your picture to any other color. This is possible because the PCjr is the only home computer besides the Atari that has a color-indirection system (although this feature can be simulated in software on other computers, as *KoalaPainter* does on the Commodore 64). Let's say you suddenly realize outer space isn't really black, it's purple. Just select X-Color and purple, go to the picture screen, place the cursor on any black area of the picture, and all the black turns purple. (That's all the black, so if something should have remained black, it's time to use Oops!)

X-Color can also extricate you from a halftone fill problem, especially if Oops won't work. You just use X-Color to make one of the tone colors the same as the other. Then you've got a solid color which can be filled in the normal way.

• Storage lets you name, save, and recall your pictures with the disk drive. It calls up a special disk menu which requires elementary reading skills. Picture files must be named and typed in on the keyboard. Parents may need to help young children at this point.

Jittery Drawing

• The Draw option lets you draw freehand, just like you would with a pen or brush. Marks appear on the screen only when the pad button is pressed. This mode ought to be simple and straightforward, but it's not.

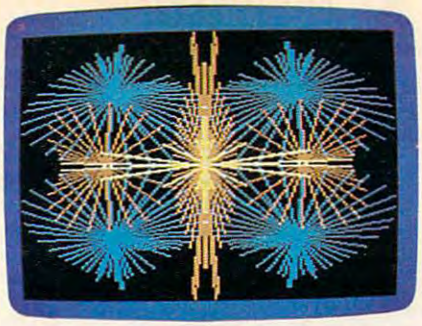
First, the cursor is always jiggling around with *KoalaPainter*. It's almost impossible to draw a smooth line with Draw. Instead, Draw makes a spontaneous, lively line. This is fun for some purposes, but frustrating for others. When rigid control is required, you must switch to one of the precise drawing commands, such as Line.

More importantly, there is a relatively minor but annoying bug in the KoalaPad system. The pad keeps track of the stylus's position by constantly comparing it with a theoretical pressure point at the upper-left corner of the pad. Sometimes, mostly when using the Draw command, if your stylus pressure is a little light, the pad suddenly decides it's *you* up there in the left-hand corner. Consequently, a line instantly splashes up to that corner of the screen. If this happens, go straight to Oops, unless you've drawn a great deal since the last menu selection. If that's the case, it's better to choose a thick brush and erase the pesky line by painting over it with the background color.

Be alert for this problem. It's built-in, but won't be a danger unless you forget what's happening.

• Zoom is the ultimate command for the detail-minded. When you select Zoom, a small rectangle appears on your drawing screen. It can be moved to any area of your picture that needs close inspection or detailed revision. Then press the button. You'll see not only a greatly enlarged view of the area under the rectangle, but also a normal-sized view of that general part of the screen. A special palette display lets you change colors without going to the menu. You can use the magnified cursor to change any of the individual dots of color that make up your picture. The changes appear instantly in the normal-sized section of the picture. The only *KoalaPainter* command supported in Zoom is Draw. This means you are pretty well restricted to changing just one block of color at a time.

Other graphics packages offer Zoom commands that support more complex drawing functions, but you usually can't even tell what you're looking at unless you call back the main picture screen. The KoalaPad Zoom is far superior to any other I've seen or used.



"Geo," a symmetrical geometric abstraction created with rays, lines, and four-way mirroring.

Art Or Doodling?

As an instrument for creating art and graphics, the KoalaPad and *KoalaPainter* are a boon to the serious designer as well as the computer doodler.

While personal computers are now widely recognized for their serious financial and communications applications, there seems to be some reluctance to acknowledge them as a viable medium for the graphic arts. Artists who make full use of the power of the KoalaPad system may help turn this around.

It would also help if *KoalaPainter* provided a way to make hardcopy printouts of its screen displays. At the moment, you can permanently save your creations only by storing them on disk, recording them on videotape, or photographing them off the screen. Maybe someday a good programmer will write a utility to dump the images to a color printer.

There will always be skeptics who think the home computer is, at best, an art toy. After all, what do those dots of color on a TV screen have to do with real art? But traditional artists have created great works using similar techniques. One example is Georges Seurat (1859–1891), the famous French artist and champion of the Neo-Impressionist movement. His significant contribution was largely based on a system of isolated dots of color used to build up entire paintings. Although his unfortunately brief working life spanned a number of years, Seurat's demanding, precise technique limited his lifelong output to seven major paintings. I suspect that Georges Seurat might have been very interested in the efficient and powerful KoalaPad and *KoalaPainter.*

KoalaPad
Koala Technologies Corporation
3100 Patrick Henry Drive
Santa Clara, CA 95052-8100
$124.95

OmniWriter & OmniSpell

Joseph R. Sutton

Requirements: Commodore 64, a disk drive, and a printer on either the serial bus or user port (parallel only). See text for the printers supported.

OmniWriter is a page-oriented word processor for the Commodore 64 which includes *OmniSpell*, a 30,000-word spelling checker. The package is best suited for page-oriented writing applications, such as letters.

A bit of explanation: Word processors are either page-oriented, line-oriented, or character-oriented. For the sake of convenience, most word processors these days are character-oriented; you can move a cursor anywhere in the document, and all the writing and editing is done on the same screen. In other words, a character-oriented word processor does not treat a document traditionally as a group of pages during the writing and editing phases. Instead, the document is written and edited as if it were one very long page. Only when the document is printed (or print-previewed, if the word processor has such a feature) is the text broken up into separate pages.

Page-oriented word processors work quite differently. Usually they treat a document as separate pages onscreen. Often there are separate screens for various functions. For instance, *OmniWriter* has a work page, header page, footer page, and text pages. The work page can be used for such things as comments and rearranging text. The header and footer pages, obviously, hold the header and footer information. The text pages hold your document. In addition, the first text page has a format line where you can specify right and left margins, tab settings, and other formatting options.

Any number of format lines are allowed on any of the pages. Lines can be up to 240 characters long, scrolling horizontally across the 40-column screen. Each file can hold up to 34,000 characters, and files can be linked together for printing out larger documents.

OmniWriter constantly displays useful information at the top of the screen: the title *OmniWriter*, your filename for the document in memory, the page number, number of pages, line number, and column number. Activating the page-width command highlights the *W* in the title *OmniWriter*. Prompts for other commands are just below the filename of the document. To use any command, you press the Commodore logo key and the appropriate letter. You can embed formatting commands in the text by pressing the CONTROL key, lighting up a small white box in the lower-left corner of the screen. Then you select the appropriate letter. This inserts a symbol in the document and performs the proper function, such as line centering, tabbing, and so on.

To help you remember all the commands, *OmniWriter* comes with a quick reference card and a function key overlay.

Merging And Printing

Like most word processors, *OmniWriter* lets you merge other documents or files into your text from disk or tape. It allows two types of merges: pasting text into the document with an editing command, and mail merging. The merged document can be created with *OmniWriter*, *HESwriter*, *EasyScript*, *Wordcraft*, *WordPro*, *SuperScript*, *MicroScript*, or *Busicalc*—that is, it can be any standard Commodore sequential file. You can also merge disk directories. Mail merge takes place at the time of printing and is activated by the embedded merge command. The merged text (such as an address from a mailing list) is taken from either the work page or a disk file. This feature can be used for producing form letters.

The print command brings up a screen of options to make *OmniWriter* compatible with a number of different printers. It supports VIC, Epson, Que/Diablo, New Spinwriter, Triumph Adler TRD170S, Ricoh Flowriter, and ASCII printers. Unfortunately, *OmniWriter* does not support RS-232 serial printers.

As a page-oriented word processor, *OmniWriter* presents some advantages and disadvantages when printing out documents. Among the advantages: You can specify the starting and ending pages of the document to be printed. This can save you lots of paper, trouble, and time when you have to reprint only a portion of a document—after making a minor last-minute change, for example. Also, the screen shows formatted text at all times, except for multiple-line spacing. This too can save paper and labor.

But there are also some disadvantages. For one thing, page endings are not automatic. If you don't specify the page breaks, you'll have one long page. If you decide later that you want double-spacing, you have to do some arithmetic to rearrange the page breaks again. (Practically all character-oriented word processors calculate page breaks automatically.)

Headers and footers cannot be turned on and off from within the text, so if you want a header on all pages except the first, you must print the first page separately with the header turned off. Footers cannot be turned off at all. To remove a footer you must delete all information from the footer page.

A minor problem is that if you separate sentences by typing two spaces after periods, and if a sentence ends at the end of a line, *OmniWriter* prints the second space at the beginning of the next line. The solution is to type only one space after periods, although this runs counter to some typists' training.

Fast Spell-Checking

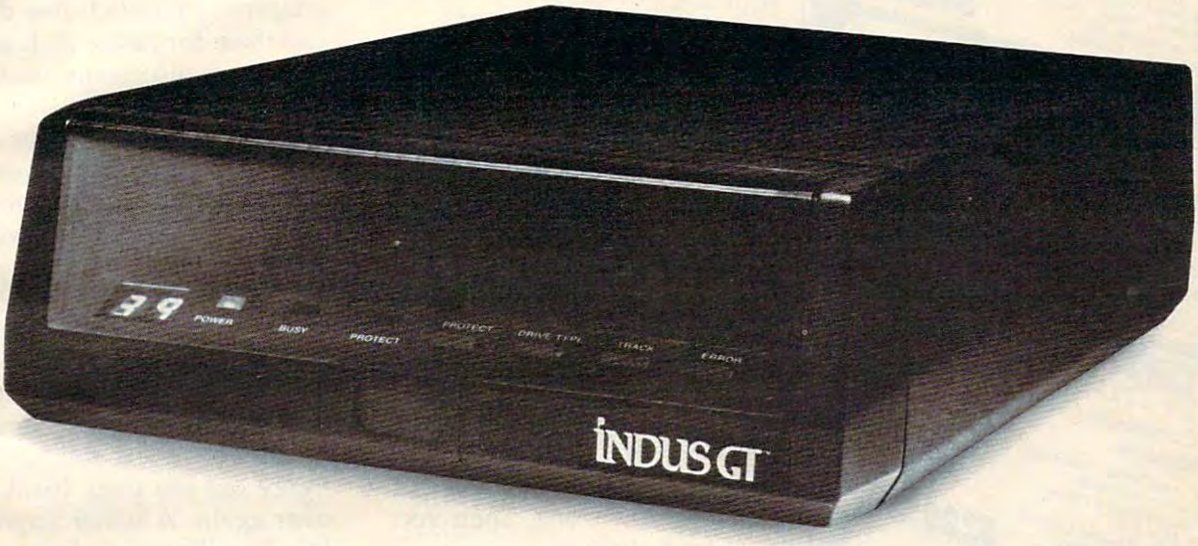
OmniSpell is included with *OmniWriter* to check for spelling errors. I was pleasantly surprised with its speed of operation. After it's loaded, it arranges all the unique words in the text in alphabetical order, then presents a menu:

F1—Spell-Check Document
F3—Alphabetical Word List
F5—High-Usage Word List
F7—Dictionary Search
F8—Return to *OmniWriter*

When you check the spelling, the words in the ordered list are displayed on the screen in two columns. The words are checked against a dictionary on

Now Available For
COMMODORE

Looks like a Ferrari. Drives like a Rolls. Parks like a Beetle.



Ask your computer dealer to let you test drive the all new 1984 Indus GT.™

The most advanced, most handsome disk drive in the world.

Flip its power switch and...
Turn your Atari into Ferrari.
Unleash your Apple.
And now turbocharge your Commodore.

Looks like a Ferrari.

The Indus GT is only 2.65″ high. But under its front-loading front end is slimline engineering with a distinctive European-Gran flair.

Engaging its AccuTouch™ buttons lets you control the LED-lit CommandPost.™ Marvel at how responsive it makes every Commodore, Apple and Atari personal computer.

Drives like a Rolls.

Nestled into its soundproofed chassis is the quietest and most powerful disk drive system money can buy. At top speed, it's virtually inaudible...whisper quiet.

Built into each Indus GT is a perfect combination of craftsmanship and advanced engineering. Luxurious styling reflects the personal tastes of each GT owner.

And each GT comes with the exclusive *GT DrivingSystem*™ of software programs.* World-class word processing is a breeze with the *GT Estate WordProcessor.*™ Your dealer will describe the two additional programs that allow GT owners to accelerate their computer driving skills.

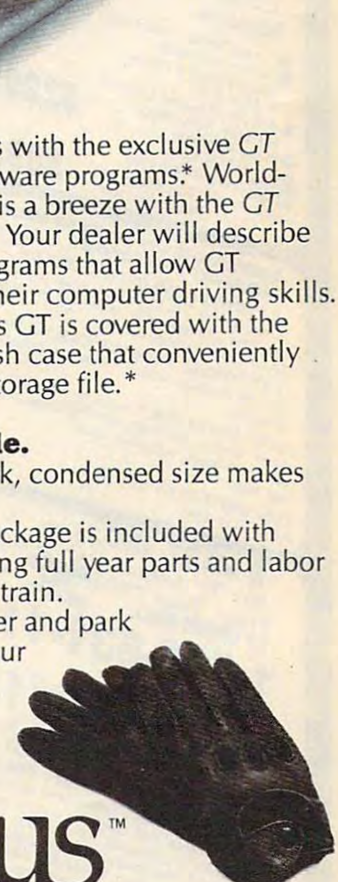
Also, the 1984 Indus GT is covered with the GT PortaCase.™ A stylish case that conveniently doubles as a 80-disk storage file.*

Parks like a Beetle.

The GT's small, sleek, condensed size makes it easy to park.

A WarrantyPlus™ package is included with every Indus GT, featuring full year parts and labor on the complete drive train.

Drive home a winner and park an Indus GT next to your personal computer.



INDUS™

The all-new 1984 Indus GT Disk Drive.

The most advanced, most handsome disk drive in the world.

*Included as standard equipment.

For dealer information, call 1-800-33-INDUS. In California, 1-800-54-INDUS, (213) 882-9600.

the disk which has a file for each letter of the alphabet, and a user dictionary for your own new words. As each word is checked, it is highlighted. When no match is found, the word remains highlighted.

The alphabetical and high-usage word lists, as well as various statistics about your document compiled beneath the menu, are useful when studying your writing style.

The dictionary search option is used to look up words. If you're not exactly sure how to spell a word—after all, that's why you're looking it up, right?—you can type just a portion of the word with some wild card symbols (*,?). The wild cards are similar to those used when specifying filenames with some Disk Operating Systems.

To correct spelling errors, you return to *OmniWriter* and use the verify command. Verify scans the text and highlights the unrecognized word. Then you have four choices:

EDIT—Change text.
SKIP—Continue scanning without taking any action.
ACCEPT—Treat the word as if it is recognized.
LEARN—Add the word to a list to update the user dictionary.

All the commands, except EDIT, return to verify mode when done.

One note: It's best to switch the screen into the 40-column mode when using verify; otherwise, some highlighted words may be partially off the screen, and there's no way to scroll them into view except to exit the verify mode.

When you're finished correcting errors, you should return to *OmniSpell* if you've used the LEARN command. Then you can add all new words to the user dictionary.

OmniSpell comes on a single disk with all the programs and files. Unlike *OmniWriter*, it may be copied for backup purposes. Duplicate copies of the entire disk are available from HES (for a nominal fee) if something happens to the original.

The *OmniSpell* disk also contains several utility programs: dictionary maintenance, interface software so you can convert the user port into a parallel printer port with BASIC, a backup program, and a program that makes the software work with the MSD IEEE-488 bus adapter (a parallel disk drive interface for faster disk access). There are also some useful example files.

The manual is clear and easy to understand. Overall, the software does a good job and is easy to use. However, when using *OmniWriter* to write this review, I found the lack of automatic page breaks to be a very real problem (especially with double-spacing). Rearranging text from one page to another is simple, but then you have to figure out the page breaks all over again. A better application for *OmniWriter*, perhaps, would be short letters and form letters.

OmniWriter & OmniSpell
Human Engineered Software
150 North Hill Drive
Brisbane, CA 94005
$59.95

©

WizType

James V. Trunzo

WizType requires an Atari, Commodore 64, or Apple II-family computer with at least 32K RAM and a disk drive. The Atari version is reviewed here; other versions are similar.

Not so long ago you had to search through the fine print in computer software ads to find a program that would help teach you how to type. Now, practically any respectable computer dealer can show you an entire shelf of such products. A new program from Sierra (formerly Sierra On-Line) adds yet another program to this selection, and it just might be the best one yet. It's called *WizType*, and it's certainly a wizard of a product.

WizType is based on the characters who frequent the popular comic strip "The Wizard of Id." These include the Wizard himself; the spirit he conjures from the well; and Bung, the frequently inebriated court jester.

After booting the program disk, you are asked to enter your name, a standard feature of many programs. Next, however, a bar graph appears on the screen, illustrating the progress you've made during your lessons. Next there's a menu which shows the variety of options available in *WizType*:

1. Game
2. Drill
3. Practice
4. Words
5. Own Lesson
6. Paragraphs

Entertaining Graphics

The Game option, for example, is not just fun and play. It's a good example of the graphics and animation built into *WizType*. Three-fourths of the screen shows the Wizard of Id facing the spirit that has emerged from the well. Letter combinations begin to appear on the screen midway between the two characters. As you correctly type the letters, the Wizard zaps the spirit with a lightning bolt, keeping it docile. If you make errors, the spirit begins to metamorphose into a ferocious dragon's head which eventually starts breathing fire at the Wizard, reducing him to a pile of ashes. The combination of smooth animation, facial expressions, and instructional lessons is hard to describe; you must see this program to appreciate it fully.

As you progress, the material you're supposed to type becomes more difficult and appears more frequently, demanding more speed and accuracy if you are to continue playing.

Even in the Game section, *WizType* does not simply display random letter combinations on the screen. Each level gears itself to a different set of keys in order of difficulty (that is, home row keys at the easiest, and numbers and symbols at the hardest). You're always shown which keys will be used, the correct finger positioning, and the finger reaches that will be exercised at each level of play.

Sierra has a good reputation for not cutting corners in its products. The Game section of *WizType* includes such touches as bonus rounds, multiple lives, and comic strip-type balloons in which the Wizard cracks jokes after you successfully complete each level.

A Little Literature

While Games contains some surprises and challenges, the Drill and Practice sections are exactly what they sound like. You can work on areas of weakness for as long as you want and select a comfortable typing speed, varying from 10 to 60 words per minute at 10-word intervals.

You can also control the typing speed after selecting the Words option from the menu. In Words, however, *WizType* assumes you are familiar with the keyboard. You start out by typing two- and three-letter words and progress to longer, more difficult words as your skill improves.

The two options which really set *WizType* apart from most typing-practice programs are Paragraphs and Own Lesson.

With Paragraphs, you type parts of eight literary works saved on the program disk. You can type the introductory paragraphs from Charles Dickens's *A Tale Of Two Cities* or a few scenes from William Shakespeare's *Hamlet*. Or perhaps you'd prefer The Gettysburg Address or selections from Mother Goose. You can also choose to have Bung serve as a pacer, hopping along on his pogo stick above the sentence being typed, at a rate you select.

In any case, typing entire paragraphs helps you attain well-rounded typing skills and arrive at a true idea of your typing speed and competency. Typing letters and words is fine for the purpose it serves, teaching the keyboard. But typing lengthy paragraphs adds new elements—fatigue, consistency, and smoothness. You can learn to establish a rhythm (and see the importance of rhythm) that cannot be achieved merely by typing single words.

If you tire of typing "To be or not to be," no problem: *WizType* also lets you create your own lessons. There are two ways to do this. First, you can simply select the Own Lesson option. This accepts individual words only. Or you can select the Paragraph option and then choose Create Paragraph from a submenu.

There are many fine typing programs currently on the market. Sierra's *WizType* certainly meets their standards, and surpasses most of them.

WizType
Sierra
P.O. Box 485
Coarsegold, CA 93614
$34.95

©

Computers And Society

David D. Thornburg

Over the past several months a considerable amount of mail has come in regarding a few of my columns. Although it isn't possible to respond to all of you individually, hearing from you is appreciated.

The column on the PCjr and Macintosh stimulated a large response. Judging from many of the letters, PCjr owners should be careful not to move their computer with the cables plugged in. Apparently, this can break the connectors and require expensive repairs of the mother board. While many of you shared some sympathy with my views of IBM's entry into the consumer marketplace, IBM's new version of the PCjr addresses at least some of Junior's problems.

Some of you wrote to express concern that the Macintosh may not make it in the marketplace. If Macs have indeed been selling at an alleged monthly rate well in excess of the total installed base of PCjr's, it is logical to conclude that it already is a success. Of course, it takes more than machines to make a market—the software from third parties is an essential component of any computer system. While a recent check of local computer stores showed less than 20 Mac titles in stock, this computer is so popular that one has to drive as far as 90 miles from the San Francisco Bay area to buy blank disks for it. Since the 3.5-inch disks are made by Sony, Apple, Hewlett-Packard, BASF, and Memorex, the scarcity of these disks in the retail outlets is a pretty good measure of the Mac's popularity, at least locally.

Some of you felt that there wasn't enough sensitivity in the critique of Craig Brod's book, *Technostress*. It was never my intention to claim that technological change couldn't induce stress in people; clearly it can. But this stress is not technology-specific. We need only look at the attempts of the weavers to kill Jacquard, the inventor of the automatic loom, or the attempts of the Luddites to thwart the industrial revolution, to see that major societal changes induce stress in some people. My concern about Brod's book was that he directed it to the general public rather than to his fellow health-care professionals. As a result of being directed to a larger audience, his book has been used by some people as further support for their own belief that computer technology is intrinsically evil.

Technology is neutral. The computer that is used to help a handicapped author write a novel can be used to help rob a bank. The computer that allows one businessperson to spend more time with the family can be used by another as an excuse for withdrawal from society. Unless a single piece of technology can be shown to induce the same stress reaction in everybody, we would perform a greater service to our species by looking at the causes of stress within the human psyche rather than in the artifacts of man. This is not to say that technology can't be abused. It can, and it has. What is of critical importance is to realize that the source of the abuse is human, not mechanical.

Videogame Microworlds

All of which brings us to a perennial topic: videogames. In his book *Mindstorms*, Seymour Papert, the father of Logo, talks about the samba schools in Brazil where children learn the dance from their peers, selecting the group that best represents their skill levels. This environment of peer teaching suggests to Papert a model of educational reform in our own classrooms, a model where children are free to explore computer-based microworlds and to acquire skills and knowledge in the context of these microworlds.

Of course, Papert had Logo in mind as the computational language to be used by the children. But, independently of our schools, and without the benefits of Logo, child-centered computational environments have sprung up in our society almost spontaneously. These are, of

David Thornburg is an author and speaker who has been heavily involved with the personal computer field since 1978. His main interest is in making computers responsive to people's needs. He is the inventor of the KoalaPad graphics tablet and is the author of nine books about programming, including Computer Art and Animation: A User's Guide to Atari Logo, The KoalaPad Book, *and* Exploring Logo Without a Computer *(Addison-Wesley). His* 101 Ways to Use a Macintosh *will appear soon from Random House. He has been called "an enthusiastic advocate for a humanistic computer revolution," and his editorial opinions have appeared in* COMPUTE! *since its inception.*

course, the game arcades.

Whenever parents and teachers talk about videogames, there are always several strong opponents to the arcades. They talk about the arcades as hangouts for delinquents, they talk about the addictive nature of the games themselves, and they talk about the violence and mayhem represented by the nature of the games themselves. What they don't talk about is their own experience in the arcades because, almost without exception, the most vocal detractors of the arcades *have never been inside one!* Before giving views on what is happening in the arcades, let's explore the research of some people who *have* taken the time to study what is going on there.

Sherry Turkle, whose doctorate is in sociology and psychology from Harvard, has spent a lot of time in game arcades, including over 100 hours carefully studying 30 game-players of all ages. Her results are included in her book *The Second Self—Computers and the Human Spirit* (Simon and Schuster, $17.95). She acknowledges the ambivalence felt by adults toward the game arcades. Their children are coming home from school with new skills, they are learning how to program,. and they take computer technology for granted. Parents want their children to have these skills, but they also realize that their expertise in the computer world may create a new generation gap. Consequently, when a game arcade applies for a business license, this is a chance for the parents to say,"Let's wait." As Turkle says, "It feels like a chance to buy time against more than a video game. It feels like a chance to buy time against a new way of life." With respect to the commonly expressed belief that game players are caught in a "mindless addiction," she replies:

> There is nothing mindless about a video game. The games demand skills that are complex and differentiated. Some of them begin to constitute a socialization into the computer culture: you interact with a program, you learn how to learn what it can do, you get used to assimilating large amounts of information about structure and strategy by interacting with a dynamic screen display. And when one game is mastered, there is thinking about how to generalize strategies to other games. There is learning how to learn.

It is this epistemological aspect of videogaming that gives it the power to become a good educational medium, if anyone wanted to really explore that field. But this still doesn't address the issue of "addiction." Turkle points out that, yes, some players can become addicted to their games. But she also points out, "Most people don't become addicted to video games just as most people who diet don't become anorexic."

A Man's World

In another in-depth study of videogaming, psychology professors Geoffrey and Elizabeth Loftus have examined many aspects of these games and their influence on players in their book *Mind at Play—The Psychology of Video Games* (Basic Books, $14.95). Among many other things, they point out that, in addition to the eye-hand coordination skills acquired through playing these games, there are other indirect benefits as well. One of these is the development of intense interest in computers which can lead the game players into the computer field as a profession.

Another criticism leveled against arcades is that they appear to be male-dominated. Clearly the content of the games themselves has something to do with this. *Dragon's Lair*, a videodisc based game, has a male player that you control to help save the princess. If this game had the roles reversed, or otherwise took into consideration the types of fantasies that might appeal more to women, it would perhaps encourage more women to visit the arcades.

Of course, there have been some games that seem to appeal equally well to men and women. Among these are *Centipede* (which was written by a woman), *Pac-Man* (and its offspring), *QIX*, *Tempest*, and several other recent games. If the arcades are the breeding ground for interest in employment in the information sector, then we should do everything in our power to insure that the arcades attract men and women alike.

Of course, there is the additional argument that the games are too violent. But violence has to be viewed in a context. We don't have a videogame in which we see someone go into a restaurant and kill babies in their mother's arms, but we can see that on television if we watch the evening news. We don't have videogames in which the goal is to demoralize the opponent by killing his livestock, laying waste to his property, and killing his children, but we can read all about it in the Bible. If violence is bad in games, then violence is bad, period. Ban the violent games if you wish, but then ban the Bible, ban the news, ban Hans Christian Andersen's fairy tales, ban Prokofiev's "Peter and the Wolf."

It is one thing to say that our society is too violent, and another to say that videogames are too violent. One can make the argument that we should do something about violence in our society, but violence won't be reduced by removing the videogames. My opinion is that anyone who lets their children read fairy tales or the Bible, but who becomes concerned when the child plays a game defending the earth from alien invaders, is a hypocrite.

Some videogames may be violent, but at least they aren't hypocritical. ©

All About The Status Register

Part 2

Louis F. Sander

Beginning machine language programmers are often confused by the 6502's status register, a collection of eight bits which act as status flags. Part 1, published last month, briefly described the function of each flag. Part 2 picks up with a more detailed explanation and includes a sample program for the Commodore 64, PET/CBM, Apple, and Atari.

Last month's short description of the 6502's eight-bit processor status register and the seven status flags it contains may have cleared some mystery away, but it surely wasn't comprehensive. That sort of description is found in most machine language programming books, to which you are now referred, and which will be much easier to understand once you've mastered what is presented here. Let's gain that mastery by running a simple test program, using a *machine language monitor* to observe its effects on the status register.

6502 Monitors

A monitor is nothing more than a machine language program that makes it easier to work with other ML programs, and there are many, many different monitors available for the 6502. Sometimes monitors are called debuggers. PET/CBMs and Apples have simple monitors built into their ROMs, while Ataris have one in the *Assembler Editor* cartridge. Commodore 64s can use "Supermon64," a program that appeared in COMPUTE! (January 1983). Our test program works identically on all those machines, but since their monitors are somewhat dissimilar, we'll show the screen dialogue for each one.

The monitor is a wonderful tool for the beginning ML programmer, and if you've dabbled with ML, you've at least used it to examine memory locations and to save ML programs on tape or disk. I used mine for those things for many months, but never paid much attention to the registers display. That's the line of labeled numbers the monitor sometimes prints on the screen, and it looks like this on Commodore machines:

```
  PC   IRQ   SR   AC   XR   YR   SP
 0005  72E8  30   00   5E   04   F8
```

(If you have an Apple or Atari, please follow the Commodore explanation anyway. Your monitor is only slightly different.)

PC shows the address in the 6502's program counter, which is nothing more than the address of the next instruction to be executed. Because of various quirks, monitor PC displays are not all alike. Supermon64 and the Atari monitor show the address of the BRK instruction which activated them; the Commodore monitor shows the address one byte *after* the BRK; and the Apple shows the address two bytes after the BRK. Some monitors don't label the address, but all of them display it.

IRQ (not shown on many monitors) gives the so-called interrupt vector, a very important address, but one beyond the scope of our interest here.

SR gives the contents of the processor status register, expressed in hexadecimal form. (The

NRI gives you ALL the training you'll need to repair ALL microcomputers.

When you've learned the basics the NRI way, you can troubleshoot the entire system—and earn good money doing it!

For business owners, lost computer time means lost money. For customers, it usually means frustration, delay and a strong temptation to take a walk to the nearest competitor.

But for the NRI-trained computer service technician, a down computer can mean higher earnings or even the opportunity to start a profitable business.

Fixing computers: fastest growing occupation in the U.S.

Whether the flaw is in a circuit board, a disk drive or a printer, *everybody* wants it fixed—fast.

The U.S. Department of Labor recently projected that the number of computer service jobs wil *double* before 1995. Median earnings of full-time computer service technicians are $430 per week.

The trained computer technician can choose between: working for a large corporation or an independent; making office calls or staying in the shop; working for a retailer or for a specialized service firm—even starting his own computer repair business.

Total System Training from NRI.

As an NRI graduate, you'll be qualified to fix just about everything that can go wrong, for any major brand of desktop microcomputer. Only a person who knows and fully understands all the underlying fundamentals of microcomputers can hope to be able to tackle all microcomputers. NRI has known the need for thoroughly understanding fundamentals since 1914.

NRI's training is hands-on training. You get practical experience in writing programs in BASIC, testing and debugging systems. You'll learn how to install an expansion board, how to troubleshoot pesky circuit flaws. Using NRI's exclusive Discovery Lab®, you'll perform over 60 experiments. You'll learn how to fix the disk drives and printers.

You learn at your convenience, at your own most comfortable pace. Without classroom pressures, without rigid night-school schedules, without wasted gasoline. Your personal NRI instructor and the NRI staff will answer your questions, give you guidance—even give special help if you need it.

IBM is a Registered Trademark of International Business Machine Corporation.
Epson is a Registered Trademark of Epson America, Inc.
Apple and the Apple logo are Registered Trademarks of Apple Computer, Inc.
Compaq is a Registered Trademark of COMPAQ Computer Corporation.
©1984 AT&T Technologies, Inc.

Computer, disk drive and printer—all yours to keep

As part of your training, you work with a TRS-80 Model 4, a powerful microcomputer with many of the features and capabilities of machines costing three times as much.

You'll install a double-density disk drive and a dot-matrix printer. And the entire system—computer, drive, printer and manuals—is yours to keep, as part of your training.

100-page FREE Catalog Tells More

Send the coupon today for NRI's big 100-page free color catalog, which gives you all the facts about NRI training in Microcomputers, Robotics, Data Communications, TV/Video/ Audio Servicing and other high-tech career fields. (If the coupon is missing, write to NRI, 3939 Wisconsin Ave., Washington, DC 20016.)

Your NRI course includes this modern 64K RAM microcomputer, dual-density disk drive, dot matrix printer... plus a professional LCD multimeter, NRI Discovery Lab and hundreds of demonstrations and experiments. It's all yours to keep.

NRI SCHOOLS
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue, Washington, DC 20016
We'll give you tomorrow.

All Career courses approved under GI bill.
☐ Check for details.

CHECK ONE FREE CATALOG ONLY

- ☐ Computer Electronics with Microcomputers
- ☐ Data Communications
- ☐ Robotics & Industrial Controls
- ☐ Color TV, Audio, and Video System Servicing
- ☐ Electronic Design Technology
- ☐ Digital Electronics
- ☐ Communications Electronics
- ☐ Industrial Electronics
- ☐ Basic Electronics
- ☐ Telephone Servicing
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction
- ☐ Locksmithing & Electronic Security

Name (Please Print) Age

Street

City/State/Zip Accredited by the National Home Study Council 198-114

TRS-80 is a trademark of the Radio Shack division of Tandy Corp.

NEW 128K —MEGA BYTE DUAL DISK DRIVE—80 COLUMN

COMPUTER SYSTEM SALE!

HOME • BUSINESS • WORD PROCESSING

LOOK AT ALL YOU GET FOR ONLY **$895.**

$895 SALE PRICE

	LIST PRICE
(1) B128 COMMODORE 128K 80 COLUMN COMPUTER	$ 995.00
(2) 4023 - 100 CPS - 80 COLUMN BIDIRECTIONAL PRINTER	499.00
(3) 8050 DUAL DISK DRIVE (over 1 million bytes)	1795.00
(4) 12" HI RESOLUTION 80 COLUMN MONITOR	249.00
• BOX OF 10 LORAN LIFETIME GUARANTEED DISKS	49.95
• 1100 SHEETS FANFOLD PAPER	19.95
• ALL CABLES NEEDED FOR INTERFACING	102.05
TOTAL LIST PRICE	**$3717.95**

PLUS YOU CAN ORDER THESE BUSINESS PROGRAMS AT SALE PRICES

	LIST	SALE
Professional 80 Column Word Processor	$149.95	**$99.00**
Professional Data Base	$149.95	**$99.00**
Accounts Receivable	$149.95	**$99.00**
Accounts Payable	$149.95	**$99.00**
Payroll	$149.95	**$99.00**
Inventory	$149.95	**$99.00**
General Ledger	$149.95	**$99.00**
Financial Spread Sheet	$149.95	**$99.00**

PRINTER REPLACEMENT OPTIONS
(replace the 4023 with the following at these sale prices)

	LIST	SALE
☆ Olympia Executive Letter Quality Serial Printer	$699.00	**$399.00**
☆ Comstar Hi-Speed 160 CPS 15½" Serial Business Printer	$779.00	**$499.00**
☆ Telecommunications Deluxe Modem Package	$199.00	**$139.00**

15 DAY FREE TRIAL. We give you 15 days to try out this SUPER SYSTEM PACKAGE!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!
90 DAY IMMEDIATE REPLACEMENT WARRANTY. If any of the SUPER SYSTEM PACKAGE equipment or programs fail due to faulty workmanship or material we will replace it IMMEDIATELY at no charge!!

Add $50.00 for shipping and handling!!
$100.00 for Alaska and Hawaii orders.
WE DO NOT EXPORT TO OTHER COUNTRIES
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Commodore - 64

WORD PROCESSING BREAKTHROUGH!

SCRIPT-64 EXECUTIVE WORD PROCESSOR

(80 Columns in Color)

40 or 80 columns in color or black and white; turns your computer into a Business Machine!

This is the finest word processor available. Features include line and paragraph insertion/deletion, indentation, right and left justification, titles, page numbering, characters per inch, etc. All features are easy to use and understand. With tabs, etc. SCRIPT-64 even includes a 250 word dictionary/spelling checker to make sure your spelling is correct. The dictionary is user customizable to any technical words you may use. Furthermore, all paragraphs can be printed in writing and everyday letters are a snap. To top things off, there is a 100 page manual and help screens to make learning how to use SCRIPT-64 a snap. This word processor is so complete we can't think of anything it doesn't have. When combined with the complete database you have a powerful mailmerge and label program that lets you customize any mailing list with personalized letters. List $99.95. **Sale $49.95**. *Coupon Price $39.00. (Disk only.)

SCRIPT-64 20,000 WORD DICTIONARY

Allows you to check spelling on 20,000 most often mispelled words! List $29.95 **Sale $19.95.** *Coupon Price $10.00. (Disk only.)

SCRIPT-64 COMPLETE DATABASE

(PLUS MAIL MERGE AND LABELS)

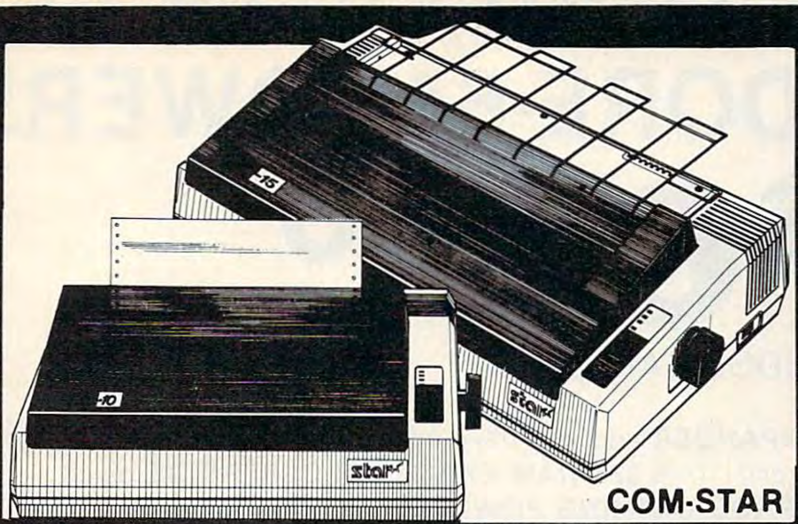
This powerful DATABASE is user friendly and makes any information easy to store and retrieve. The user defines the fields and then can add, change, delete, and search for any category wanted! When combined with the Executive Word Processor you can search out any category (zip codes, even hair color, etc.) and print super personalized letters! 600 names can be sorted and formulated on each disk in any order or category! Will handle any size mailing list by changing or adding disks! List $69.00. **Sale $34.00**. *Coupon Price $24.00.

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

WE SHIP C.O.D HONOR VISA AND MASTER CHARGE
ADD $3.00 SHIPPING FOR C.O.D. ADD $2.00 MORE
SPECIAL SERVICES:
One Day — Express Mail add $10.00

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

FANTASTIC COMPUTER PRINTER SALE!!!



COM-STAR T/F

Tractor
Friction
Printer

only $ 169**

- Lowest Priced, Best Quality, Tractor-Friction Printers in the U.S.A.
- Fast 80-120-160 Characters Per Second
- 40, 46, 66, 80, 96, 132 Characters Per Line Spacing
- Word Processing
- Print Labels, Letters, Graphs and Tables
- List Your Programs
- Print Out Data from Modem Services
- "The Most Important Accessory for Your Computer"

**** DELUXE COMSTAR T/F**
80 CPS Printer — $169.00

This COMSTAR T/F (Tractor Friction) PRINTER is exceptionally versatile. It prints 8½" x 11" standard size single sheet stationary or continuous feed computer paper. Bi-directional, impact dot matrix, 80 CPS, 224 characters. (Centronics Parallel Interfact).

Premium Quality 120-140 CPS
10" COM-STAR PLUS+
Printer $249.00

The COM-STAR PLUS+ gives you all the features of the COMSTAR T/F PRINTER plus a 10" carriage, 120-140 CPS, 9 x 9 dot matrix with double strike capability for 18 x 18 dot matrix (near letter quality), high resolution bit image (120 x 144 dot matrix), underlining, back spacing, left and right margin settings, true lower decenders with super and subscripts, prints standard, italic, block graphics and special characters. It gives you print quality and features found on printers costing twice as much!! (Centronics Parallel Interface) (Better than Epson FX80). List $499.00 **SALE $249.00**

Premium Quality 120-140 CPS
15½" COM-STAR PLUS+
Business Printer $349.00

Has all the features of the 10" COM-STAR PLUS+ PRINTER plus 15½" carriage and more powerful electronics components to handle large ledger business forms! (Better than Epson FX 100) List $599
SALE $349.00

Superior Quality 140-160 CPS
10" COM-STAR PLUS+ IBM
IBM Pers/Bus Printer $369.00

Has all the features of the 10" COM-STAR PLUS+ PRINTER! It is especially designed for all IBM personal computers! 140-160 CPS HIGH SPEED PRINTING 100% duty cycle, 2K buffer, diverse character fonts, special symbols and true decenders, vertical and horizontal tabs.
A RED HOT IBM personal business printer at an unbelieveable low price of $369.00 (centronics parallel interface)
List $699 **SALE $369.00**

Superior Quality 160-180 CPS
10" COM-STAR PLUS+HS
Business Printer $369.00

The **Super Com-Star+ High Speed Business Printer** 160-180 CPS has a 10" carriage with all the Com-Star+ features built in! The 15½" **High Speed Business Printer** is especially designed with more powerful electronics to handle larger ledger business forms! Exclusive bottom feed! (Centronics parallel interface) 15½" printer is also compatable with IBM Personal/Business Computers! 15½" Printer List $799.00 ***SALE* $469.00**

Olympia
Executive Letter Quality
DAISY WHEEL PRINTER $379.00

This is the worlds finest daisy wheel printer **Fantastic Letter Quality**, up to 20 CPS bidirectional, will handle 14.4" forms width! Has a 256 character print buffer, special print enhancements, built in tractor-feed (Centronics Parallel and RS232C Interface) List $699 **SALE $379.**

• *15 Day Free Trial - 1 Year Immediate Replacement Warranty*

PARALLEL INTERFACES

For VIC-20 and COM-64 — $49.00 For Apple computers — $79.00 Atari 850 Interface — $79.00 For ALL IBM Computers — $89.00

Add $14.50 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $29.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA—MASTER CARD—We Ship C.O.D. to U.S. Addresses Only

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

COM-STAR PLUS+ ABCDEFGHIJKLMNOPQRSTUVWXYZ
Print Example: ABCDEFGHIJKLMNOPQRSTUVWXYZ 1234567890

label is P in Apple/Atari, but what it displays is exactly, absolutely, positively identical in all the monitors.) People with 6502's in their cerebral cortices may be able to determine individual flag statuses from a hex display, but it's a burdensome interpretation for the rest of us. Who can figure out whether $FB means the Z flag is set or clear? Not me, I can guarantee you. The table is a handy guide for interpreting that byte. With it, you can tell at a glance which flags are set or cleared in a given status byte, and just what each flag means. And that ability can be a golden key to better machine language programming.

AC, XR, and YR show the contents of the accumulator, X, and Y registers, respectively, at the moment the monitor was activated. They're labeled A, X, and Y in some monitors, but mean the same thing, regardless of the label.

SP (or S) gives the value of the stack pointer, which is yet another useful value that's beyond our present scope. The value will vary from time to time and from machine to machine.

Figure 1: Apple Screen Dialogue

Step 1	To activate the monitor, type CALL−151, then press RETURN.	`]CALL-151`

Step 2	Put the program into memory by making these entries *exactly* as shown. Press RETURN at the end of each line.	`*3300:D8 18 A9 00 AA A8 C9 FF`
		`*3308:00 A9 80 00 A9 7F 00 A9`
		`*3310:00 00 A9 FF 00 69 01 00`
		`*3318:69 01 00 C9 02 00 00 00`

Step 3	Check your work by entering this command and comparing your screen display with the program.	`*3300L`
Step 4	Type the G command, then press RETURN. When this line appears, return to the text.	`*3300G` `330A-   A=00 X=00 Y=00 P=30 S=F0`
Step 5	This and the following steps are identical to Step 4, except for the numbers entered and displayed.	`*3309G` `330D-   A=80 X=00 Y=00 P=B0 S=EE`
Step 6	As above.	`*330CG` `3310-   A=7F X=00 Y=00 P=30 S=EC`
Step 7	As above.	`*330FG` `3313-   A=00 X=00 Y=00 P=32 S=EA`
Step 8	As above.	`*3312G` `3316-   A=FF X=00 Y=00 P=B0 S=E8`
Step 9	As above.	`*3315G` `3319-   A=00 X=00 Y=00 P=33 S=E6`
Step A	As above.	`*3318G` `331C-   A=02 X=00 Y=00 P=30 S=E4`
Step B	As above. This is the last step in our demonstration.	`*331BG` `331F-   A=02 X=00 Y=00 P=33 S=E2`

Stepping Through Flags

Now that you've seen a description of the register display, plus that handy table, let's use them to experiment with the important flags. Our experiment will have the dual benefit of making us more fluent in ML, and giving us practice using the register display.

The program at the end of this article is an instructive, but do-nothing, ML program that occupies an innocuous corner of memory. From left to right, each line shows a memory address, the bytes held by it and maybe its upward neighbor, and the mnemonic for the instruction that those bytes represent. The program's first seven lines set all the 6502's flags and registers to zero, then break to the monitor, where we can review their status.

Single, Simple Operations

The rest of the program is a series of single, simple operations, each followed by a break to the monitor. We're about to go through them one by one, and see what happens to the negative, break, zero, and carry flags. We'll leave V, D, and I for another day, for the reasons previously mentioned.

Figures 1–3, the different versions for PET/CBM, Apple, and Atari, will be used to track our demonstration. Find the version which applies to you, and follow it as you read on.

Steps 1–3. Our first step will be to put the ML demo program into memory. Do it now, by carefully following Steps 1, 2, and 3 of the appropriate version of Figure 1, 2, or 3. If you've never worked with ML before, don't worry—the process is easy, and we'll take you through it step by step. When you finish Step 3, come back here for further instructions.

At the end of Step 3, the monitor should still be active, and your screen should be showing you its distinctive monitor prompt. You're now ready to run the ML demo program, which you do by executing your monitor's G command.

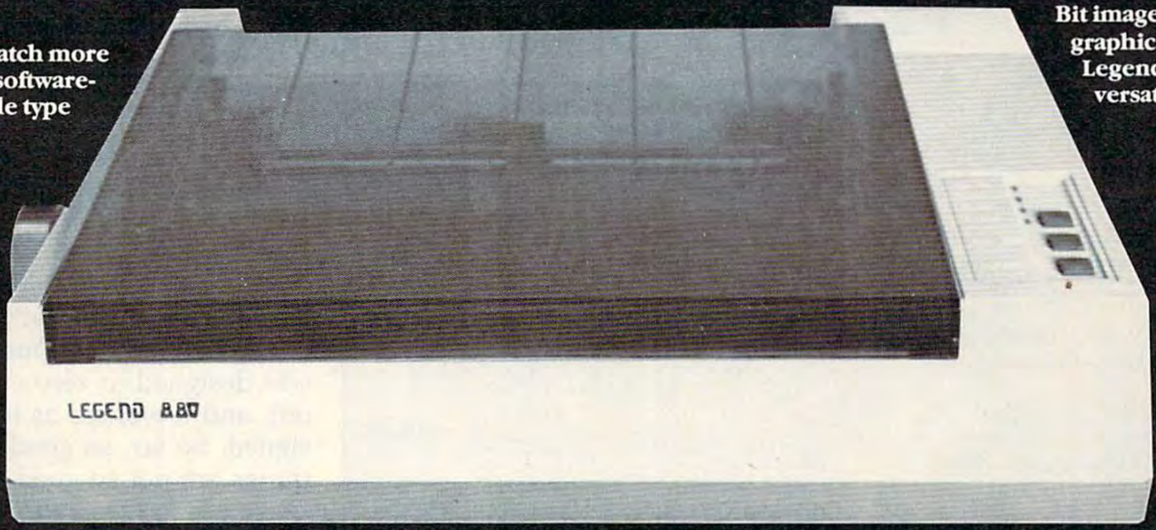
The Making Of A Legend.

Both tractor & friction feed are standard.

Prints a crisp original, plus up to three copies.

Mix & match more than 40 software-selectable type styles.

Bit image graphics add to Legend's versatility.



Centronics parallel interface standard; Serial optional.

Serial impact dot matrix output.

New square dot technology for higher resolution & near letter quality print.

Up to 142 columns in compressed printing mode.

Bi-directional logic seeking for fast output.

Check Legend 880's performance, and you'll look twice at the price. With new square dot technology and bi-directional logic seeking, this impact printer turns out near letter-quality work at 80 cps. Prints in 80-column–or 142-column–with compressed print. Does the full ASCII character set and high-resolution bit image graphics in the bargain. Lets you mix and match more than 40 software-selectable type styles. Produces a crisp original, plus three copies. Includes tractor and friction feed to handle fanfold paper and single sheets. All with Legendary reliability and a lifetime warranty on the print head.

With standard Centronics® and optional RS232C serial interfaces, it's a great match with any micro. See a Legend 880 in action.

For more information, contact:
CAL-ABCO Peripherals Division
6041 Variel Avenue, Woodland Hills
CA 91367. Telephone (818) 704-9100
Outside CA call toll free 1-800-321-4484
Telex 662436. Dealer inquiries invited

LEGEND
PERIPHERAL PRODUCTS

Legend is a trademark of CAL-ABCO.
Centronics is a registered trademark of Centronics Data Computer Corp.

Figure 2: Atari Screen Dialogue

Step 1 To activate the monitor, insert the *Assembler Editor* cartridge, and answer the EDIT prompt by typing BUG, then pressing RETURN.

Step 2 Put the program into memory by making entries *exactly* as shown. Press RETURN at the end of each line.

```
DEBUG        DEBUG     DEBUG
C3300<D8     C<00      C<01

DEBUG        DEBUG     DEBUG
C<18         C<A9      C<00

DEBUG        DEBUG     DEBUG
C<A9         C<7F      C<69

DEBUG        DEBUG     DEBUG
C<00         C<00      C<01

DEBUG        DEBUG     DEBUG
C<AA         C<A9      C<00

DEBUG        DEBUG     DEBUG
C<A8         C<00      C<C9

DEBUG        DEBUG     DEBUG
C<C9         C<00      C<02

DEBUG        DEBUG     DEBUG
C<FF         C<A9      C<00

DEBUG        DEBUG     DEBUG
C<00         C<FF      C<00

DEBUG        DEBUG     DEBUG
C<A9         C<00      C<00

DEBUG        DEBUG     DEBUG
C<80         C<69      C<00

                       DEBUG
```

Step 3 Check your work by entering this command and comparing your screen display with the one shown.

```
D 3300,331F

3300 D8 18 A9 00 AA A8 C9 FF
3308 00 A9 80 00 A9 7F 00 A9
3310 00 00 A9 FF 00 69 01 00
3318 69 01 00 C9 02 00 00 00

                       DEBUG
```

Step 4 Type the G 3300 command, then press RETURN. When DEBUG appears, return to the text.

```
G 3300
3308    A=00 X=00 Y=00 P=30 S=00
DEBUG
```

Step 5 This and the following steps are identical to Step 4, except for the numbers entered and displayed.

```
G 3309
330B    A=80 X=00 Y=00 P=B0 S=00
DEBUG
```

Step 6 As above.

```
G 330C
330E    A=7F X=00 Y=00 P=30 S=00
DEBUG
```

Step 7 As above.

```
G 330F
3311    A=00 X=00 Y=00 P=32 S=00
DEBUG
```

Step 8 As above.

```
G 3312
3314    A=FF X=00 Y=00 P=B0 S=00
DEBUG
```

Step 9 As above.

```
G 3315
3317    A=00 X=00 Y=00 P=33 S=00
DEBUG
```

Step A As above.

```
G 3318
331A    A=02 X=00 Y=00 P=30 S=00
DEBUG
```

Step B As above. This is the last step in our demonstration.

```
G 331B
331D    A=02 X=00 Y=00 P=33 S=00
DEBUG
```

Each monitor has its own syntax for this; yours is illustrated in the appropriate figure. Monitor commands are fussy about spaces, etc., so pay close attention to details at this point. Now go do Step 4, which will start execution of the machine language routine at address $3300. That routine will run until a BRK instruction is executed, at which point processing will stop and the monitor's register display will appear on the screen. When that happens, which should be immediately, come back here.

Step 4. Study the register display, disregarding IRQ and SP, and observe that AC, XR, and YR are all set to $00. (Non-Commodore people observe the same thing, labeled A, X, and Y.) Use the table to confirm that $30 means that all SR (or P) flags are clear, except for the B and the meaningless bit that's always set. Remember what the B flag is for, and it will be easy to see why it's set. Our program was designed to zero everything out, and it worked as it was designed. So far, so good. (If things are not so good, you've made a mistake. Repeat your work from the beginning.)

Nothing Has Changed

Step 5. Now perform Step 5, and notice what has happened. The program has loaded $80 (1000 0000) into the accumulator, and the monitor AC display so indicates. Since the high-order bit of $80 is a 1, the computer set its own N flag. The program counter has advanced, but nothing else has changed. (If your stack pointer changed, never mind—the monitor, not our program, changed it.) The BRK brought us back to the monitor. Simple, isn't it?

Step 6. The LDA has loaded $7F (0111 1111) into the accumulator, setting N to match its highest bit. The register display shows the $7F, and proves

"My dear, now that dinner is complete, let's do something that will work us into a passionate, writhing, intellectual frenzy. My mind is swirling with words I cannot explain. Please spend the night with me and Compuzzler."

Compuzzler.™ From Uptown Software,™ Inc. The "5 O'Clock" software that's proper for any social encounter.

Compuzzler. A strategy game based upon crossword puzzles. For people who have a love affair with crosswords. And for those who have never enjoyed the thrill.

Compuzzler comes with 70 competitive crossword puzzles. You can play them by yourself. Or against one or more people. There are five different ways to play. Once you've enjoyed these 70 puzzles, you can purchase 50 more at a time on additional disks.

Compuzzler. From Uptown Software. The first of many new ways to fall in love with your computer.

Ask your dealer about the upcoming National Compuzzler Tournament. And be sure to try our sizzling Uptown Double-Crostics™ and Uptown Trivia.™

For the dealer nearest you, call 1-800-824-7888, Operator 584. (In Hawaii and Alaska 1-800-824-7919, Operator 584)

UPTOWN Software™

that N is now clear, while all other flags remain the same. Now do Step 7.

Step 7. Putting $00 (0000 0000) in the accumulator sets the Z bit, since zeros beget zeros. Notice how the PC is stepping right along with us, and do Step 8.

Step 8. $FF (1111 1111) is *not* a zero, so the zero flag is cleared. Its high bit is a 1, so the N flag is set. Move on to the next step.

Step 9. The ADC instruction adds 1 to the accumulator. Like driving another mile when the speedometer reads 99999, this rolls the accumulator over to $00 (0000 0000). We can tell when this happens, because the rollover automatically sets the carry flag. The carry bit is often used in just this way, to tell when a counter has reached its maximum. In our example, Z is also set, since the operation resulted in a zero. When you've absorbed those simple details, go on to Step A.

Bump A Counter

Step A. The last operation did not roll over the accumulator, so the carry bit was cleared. What it did was to add 1 to the zero in the accumulator, giving a result of 2. How on earth does 1 + 0 = 2? The answer is in the carry bit. An ADC adds its operand plus the carry bit to the contents of the accumulator, then reconditions C based on the result. That's very useful, because often when a counter rolls over, we want to increment a higher-order counter, so nothing gets lost in the counting. Many programs look for the carry bit, and bump a counter if it's set. Our own little program didn't go that far, but it did show us how such things can be done. Now do the next G.

Step B. What's this? We compared a 2 to a 2, and the zero and carry flags got set. That's a special use of flags in comparing numbers. CMP and

Figure 3: Commodore Screen Dialogue

Step 1	To activate the monitor, type SYS 4, then press RETURN. (If you have a PET with original ROM, you must first load the monitor program from tape.)	`SYS4`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  0005 E62E 30 00 5E 04 F8`

Step 2	Type this command, then press RETURN.	`.M 3300 331F`

Step 3	You will see a display like this, but with different two-digit numbers. Carefully change them to these numbers by typing over them. Press RETURN at the end of each line.	`.:  3300 D8 18 A9 00 AA A8 C9 FF`
		`.:  3308 00 A9 80 00 A9 7F 00 A9`
		`.:  3310 00 00 A9 FF 00 69 01 00`
		`.:  3318 69 01 00 C9 02 00 00 00`

Step 4	Type the G 3300 command, then press RETURN. When this line appears, return to the text.	`.G 3300`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  3309 E62E 30 00 00 00 F8`

Step 5	This and the following steps are identical to Step 4, except for the numbers entered and displayed.	`.G 3309`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  330C E62E B0 80 00 00 F8`

Step 6	As above.	`.G 330C`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  330F E62E 30 7F 00 00 F6`

Step 7	As above.	`.G 330F`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  3312 E62E 32 00 00 00 F6`

Step 8	As above.	`.G 3312`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  3315 E62E B0 FF 00 00 F6`

Step 9	As above.	`.G 3315`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  3318 E62E 33 00 00 00 F6`

Step A	As above.	`.G 3318`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  331B E62E 30 02 00 00 F4`

Step B	As above, but when the registers display appears, type X and press RETURN. Then go back to the text.	`.G 331B`
		`B*`
		`    PC   IRQ  SR AC XR YR SP`
		`.;  331E E62E 33 02 00 00 F4`

LET'S CLEAR UP A 'BASIC' MISUNDERSTANDING ABOUT COMPUTERS.

YOU DON'T HAVE TO BE INTIMIDATED BY YOUR COMPUTER ANY LONGER.

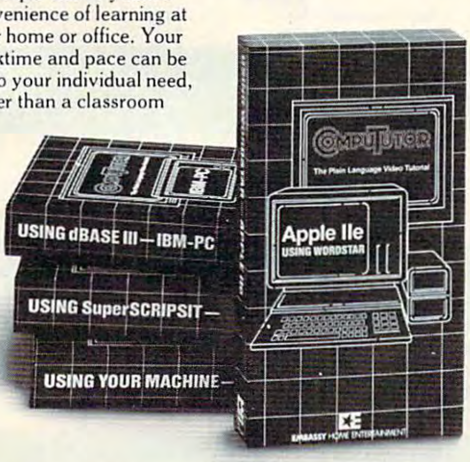
If you're having a misunderstanding with your computer, it's probably not your computer's fault. It's the complicated owner's manual or software instructions that are hard to comprehend. But now you can learn to operate, program and use software easier and faster than ever.

INTRODUCING COMPUTUTOR™

CompuTutor, is a remarkable series of 90-minute plain language computer tutorials on pre-recorded videocassette. It will make learning to operate your personal computer a matter of hours, rather than days or weeks. No more wading through complex instruction manuals.

NOW YOU'RE IN CONTROL

CompuTutor is designed specifically for the IBM®-PC, Apple® IIe, the Radio Shack® TRS-80®, Model 4, and over forty compatibles. It differs from manuals and other on-disk or on-tape tutorials in three very important ways. First, CompuTutor presents clear, "non-computerese," step-by-step instruction. You'll learn specific information about your machine, rather than catch-all generalizations. Second, CompuTutor is both entertaining and imaginative. And third, CompuTutor's self-teaching technique allows you the convenience of learning at your home or office. Your worktime and pace can be set to your individual need, rather than a classroom schedule. Whether you're a novice or an experienced end user, CompuTutor is a personal instructor, a private coach anytime, as close as your videocassette recorder.



WE'RE TEACHING YOUR FAVORITE PROGRAMS

Now popular software for your personal computer doesn't have to be complicated. We've made CompuTutor for the most popular business and home computing programs available for your machine:

"Using Your Machine" gets you started from assembly or hook-up, to learning to operate your machine and its disk operating system. You'll even write a short program in "BASIC" language.

"Using Word Processing"—WORDSTAR® and SuperSCRIPSIT™ shows the use and application of popular word processing programs currently sold.

"Using VisiCalc®" teaches basic properties and applications of VisiCalc, the electronic spread sheet.

"Using Data Base Management"—dBASEII®, dBASEIII™ and Profile 4™, teaches the most popular of the relational data base programs currently sold.

CompuTutor makes the best software for your personal computer perform even better, because we've made it easier to learn and understand. After all, isn't that the bottom line?

WHAT YOU CAN DO RIGHT NOW

Get the most out of your personal computer. Start with the right learning system! CompuTutor, the power of simplicity.

To find out more, ask your computer dealer or contact Jim Brown, Director of Sales/Marketing, Embassy Home Entertainment, 1901 Avenue of the Stars, Los Angeles, California 90067.

COMPUTUTOR™

developed by Chase Scientific

The Plain Language Video Tutorial

Marketed Exclusively by EMBASSY HOME ENTERTAINMENT®

IT ALL ADDS UP...

IBM

NEC PRINTERS
NEC 2050..............................$899.00
NEC 3550..............................$1599.00

TANDON
5¼" 320K Floppy....................$189.00

VISICORP
VisiCalc IV..............................$159.00
VisiWord +..............................$249.00
Visi-on 'Series'..........................CALL
Optical Mouse..........................$189.99

IDEAssociates
5MB to 45MB Hard drives with removable Cart. back up as low as.... $1399.00

AST RESEARCH
Six Pak Plus.........from.........$249.00
Combo Plus II........from........$279.00
Mega Plus..........from...........$299.00
I/O Plus............from............$139.00

QUADRAM
Quadlink 64K..............................$479.00
Quadboard II.....as low as.....$309.00
Quad 512 Plus....as low as....$259.00
Quadcolor I..............................$219.00
Chronograph..............................$89.99
Parallel Interface Board..........$89.99
64K RAM Chips Kit..................$49.99

SPI
Open Access..............................$369.00

HAYES
Please (Data Base)....................$269.00

MAI
General Ledger..............................CALL
Accounts Payable..........................CALL
Accounts Receivable......................CALL
Payroll..CALL
Inventory...CALL

MICROPRO
WordStar Professional Pack..$339.00

MICROMIM
R:Base 4000..............................$279.00

MULTIMATE INT.
Multi Mate..............................$289.00

MICROSTUF
Crosstalk..............................$105.00

MICROSOFT
MultiPlan..............................$139.00

ASHTON-TATE
Framework..............................$399.00
cBASE II..........upgrade..........$149.00
dBASE II..............................$299.00
dBASE III..............................$399.00
Friday!..............................$179.00

IUS
EasyWriter II..............................$249.00
EasySpeller..............................$119.00
EasyFiler..............................$229.00

CONTINENTAL SOFTWARE
1st Class Mail/Form Letter......$79.99
Home Accounting Plus............$88.99

LOTUS
Symphony..............................$499.00
1-2-3..............................$319.00

PROFESSIONAL SOFTWARE
PC Plus/The Boss..............................$279.00

SYNAPSE
File Manager..............................$69.99

SOFTWARE

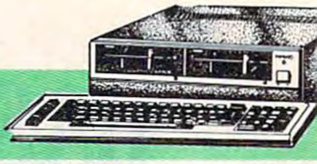
	C-64	Atari	IBM	Apple
ELECTRONIC ARTS				
One on One	$29.99	N/A	$29.99	$29.99
Music Construction	$29.99	$29.99	$29.99	$29.99
Pinball Construction	$29.99	$29.99	$29.99	$29.99
Cut & Paste	$39.99	$39.99	$39.99	$39.99
Hard Hat Mack	$27.99	$27.99	$27.99	$27.99
INFOCOM				
Witness	$29.99	$29.99	$29.99	$29.99
Infidel	$29.99	$29.99	$29.99	$29.99
Deadline	$29.99	$29.99	$29.99	$29.99
Planetfall	$29.99	$29.99	$29.99	$29.99
Enchanter	$29.99	$29.99	$29.99	$29.99
Suspended	$29.99	$29.99	$29.99	$29.99
Sorcerer	$29.99	$29.99	$29.99	$29.99
Zork 1,2,3 ea	$27.99	$27.99	$27.99	$27.99
ATARISOFT				
Joust	$34.99	N/A	$34.99	$34.99
Moon Patrol	$34.99	N/A	$34.99	$34.99
Pole Position	$34.99	N/A	$34.99	$34.99
PacMan	$16.99	N/A	$16.99	$16.99
Ms. PacMan	$34.99	N/A	$34.99	$34.99
Donkey Kong	$34.99	N/A	$34.99	$34.99
DYNATECH				
Code Writer	$75.99	$79.99	$175.99	$155.99
VisiCorp				
VisiCalc	$159.99	$139.99	$159.99	$159.99
PFS:				
Write	N/A	N/A	$89.99	$89.99
Graph	N/A	N/A	$89.99	$79.99
Report	N/A	N/A	$79.99	$79.99
File	N/A	N/A	$89.99	$79.99
Solutions:* as low as	N/A	N/A	$16.99	$16.99
THE LEARNING COMPANY				
Magic Spells	$22.99	$22.99	$22.99	$22.99
Word Spinner	$22.99	$22.99	$22.99	$22.99
Bumble Games	$22.99	$22.99	N/A	$22.99
Addition Magician	$22.99	N/A	$22.99	$22.99
Reader Rabbit	$24.99	N/A	$24.99	$24.99

*Call on Titles

ZENITH data systems

PC COMPATIBLE 16 BIT SYSTEMS
Call for price and configurations

SANYO



MCB 550..............................$699.
MBC 550-2..............................$749.
MBC 555..............................$949.
MBC 555-2..............................$1099.

APPLE

APPLE IIe STARTER PACK
64K Apple IIe, Disk Drive & Controller, 80 Column Card, Monitor II & DOS 3.3 ....CALL

APPLE IIe..............................CA
APPLE IIc..............................CA
MacINTOSH..............................CA

MONITORS

AMDEK
300 Green..............................$129.00
300 Amber..............................$149.00
310 Amber IBM Plug..............$169.00
Color 300/audio......................$269.00
Color 500 Composite/RGB/VCR$389.00
Color 600 Hires(640 × 240).....$549.00
Color 700 Hires(720 × 240).....$639.00
Color 710 Long Phosphor......$679.00

BMC
1201 (12" Green)......................$88.99
1201 Plus (12" Green Hi-Res).$98.99
9191 Plus..............................$249.00

GORILLA
12" Green..............................$88.99
12" Amber..............................$95.99

NEC
JB 1206 Green......................$109.00
JB 1201 Green......................$139.99
JB 1205 Amber......................$149.99
JB 1215 Color......................$269.00
JC 1216 RGB......................$429.00
JC 1460 Color......................$359.00

PRINCETON GRAPHICS
MAX-12 Amber......................$199.00
HX-12 RGB......................$489.00
SR-12 RGB......................$629.00

SAKATA
SC-100 Color......................$269
SG-1000 Green......................$129
SA-1000 Amber......................$139

TAXAN
100 12" Green......................$125
121 IBM Green......................$149
100 12" Amber......................$1
122 IBM Amber......................$159
210 Color RGB......................$269
400 Med-Res RGB......................$319
415 Hi-Res RGB......................$439
420 Hi-Res RGB (IBM)..........$469

USI
Pi 1, 9" Green......................$99
Pi 2, 12" Green......................$119
Pi 3, 12" Amber......................$129
Pi 4, 9" Amber......................$119
1400 Color......................$269

QUADRAM
Quadchrome 8400 Color........$499

ZENITH
ZVM 122 Amber......................$99
ZVM 123 Green......................$89
ZVM 135 Color/RGB..............$469

MODEMS

ANCHOR
Volksmodem..............................$59.99
Mark IL Serial..............................$79.99
Mark VII (Auto Ans/Auto Dial)$99.99
Mark XII (1200 Baud)...........$259.99
Mark TRS-80..............................$99.99
9 Volt Power Supply................$9.99

HAYES
Smartmodem 300.................$199.00
Smartmodem 1200...............$499.00
Smartmodem 1200B.............$419.00
Micromodem IIe....................$269.00
Micromodem 100..................$299.00
Smart Com II..........................$75.99
Chronograph.........................$199.00

NOVATION
J-Cat..............................$99
Cat..............................$139
Smart Cat 103......................$179
Smart Cat 103/212................$399
AutoCat..............................$219
212 AutoCat......................$549
Apple Cat II......................$249
212 Apple Cat......................$449
Apple Cat 212 Upgrade........$259
Smart Cat Plus......................$369

ZENITH
ZT-1..............................$339
ZT-10..............................$309
ZT-11..............................$369

KOALA
Atari (ROM)..............................$79.99
C-64..............................$79.99
IBM..............................$99
Apple/Franklin......................$85

COMPUTER MAIL ORDER

west
MasterCard
800-648-3311
In NV call (702)588-5654
Order Status Number: 588-5654
P.O.Box 6689, **Dept.105**
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces800-268-4559
In Toronto call (416) 828-0866
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B, **Dept.105**
Mississauga, Ontario, Canada L5L1T1

east
VISA
800-233-8950
In PA call (717)327-9575
Order Status Number: 327-9576
Customer Service Number: 327-1450
477E.3rdSt., **Dept.105** Williamsport,PA 1770

No risk, no deposit on C.O.D. orders and no waiting period for certified checks or money orders. Add 3% (minimum $5) shipping and handlin on all orders. Larger shipments may require additional charges. NV and PA residents add sales tax. All items subject to availability an price change. Call today for our catalog.

...THE BEST PRICES

PRINTERS

AXIOM
- AT-100 Atari Interface Printer $199.00
- AT-550 Atari Bidirectional $319.00
- GP-100 Parallel Interface $189.00
- GP-700 Atari Color Printer $489.00
- GP-550 Parallel Printer $269.00

BMC
- 401 Letter Quality $589.00
- BX-80 Dot Matrix $269.00

C.ITOH
- Gorilla Banana $149.00
- Prowriter 8510P $339.00
- Prowriter 1550P $599.00
- A10 (18 cps) Son of Starwriter $569.00
- Hot Dot Matrix CALL
- F10-40 Starwriter $949.00
- F10-55 Printmaster $1249.00

COMREX
- ComWriterII Letter Quality $149.00

DIABLO
- 620 Letter Quality $949.00
- 630 API Letter Quality $1699.00

DAISYWRITER
- 2000 $999.00

EPSON
- RX-80, RX-80FT, RX-100 CALL
- FX-80, FX-100 NEW CALL
- LQ 1500 LOW CALL
- JX-80 Color PRICES CALL

JUKI
- 6100 $469.00

MANNESMAN TALLY
- 160L $589.00
- 180L $749.00
- Spirit 80 $259.00

NEC
- 2010/15/30 $749.00
- 3510/15/30 $1369.00
- 7710/15/30 $1799.00
- Pinwriter CALL

OKIDATA
- 82, 83, 84, 92, 93, 2350, 2410 CALL

OLYMPIA
- Compact 2 $469.00
- Compact RO $499.00
- ESW 3000 $1399.00
- Needlepoint Dot Matrix $329.00

SMITH CORONA
- TP-1000 $449.00
- Tractor Feed $119.00

SILVER REED
- 500 Letter Quality $379.00
- 550 Letter Quality $469.00
- 770 Letter Quality $869.00

STAR
- Gemini 10X $279.00
- Gemini 15X $389.00
- Serial Board $75.00
- Radix 10 $599.00
- Radix 15 $699.00

TOSHIBA
- 1340 $799.00
- 1351 $1369.00

TRANSTAR
- 120P $469.00
- 130P $649.00
- 315 Color $459.00

HOME COMPUTERS

ATARI

CALL WHILE SUPPLIES LAST
600XL, 800XL, 1200XL

- 850 Interface $139.00
- 1010 Recorder $59.99
- 1020 Color Printer $89.99
- 1025 Dot Matrix Printer $229.99
- 1027 Letter Quality Printer $269.99
- 1030 Direct Connect Modem $69.99
- 1050 Disk Drive $259.99
- 64 Memory Module (600) $119.99
- Touch Table/Software $64.99
- Light Pen/Software $72.99
- CX22 Track Ball $39.99
- 7097 Atari Logo $74.99
- 4018 Pilot (Home) $57.99
- 405 Pilot (Educ.) $99.99
- 8036 Atari Writer $77.99
- 5049 VisiCalc $79.99
- CX30Paddles $11.99
- CX40 Joystick $7.99
- 4011 Star Raiders $12.99
- 4022 Pac Man $16.99
- 4025 Defender $32.99
- 8026 Dig Dug $32.99
- 8031 Donkey Kong $32.99
- 8034 Pole Position $32.99
- 8040 Donkey Kong Jr. $32.99
- 8043 Ms Pacman $32.99
- 8044 Joust $32.99
- 8045 Pengo $16.99
- 8052 Moon Patrol $32.99
- 4003 Assembler $34.99
- 8126 Microsoft Basic I or II $64.99
- 488 Communicator II $119.99

MEMORY BOARDS

ATARI
- Axlon 32K $49.99
- Axlon 48K $74.99
- Axlon 128K $279.99

Apple/Franklin
- Axlon 128K $299.00
- Axlon 320K $849.00

DISK DRIVES FOR ATARI

INDUS
- GT Drive (Atari) $379.00

RANA
- 1000 $299.00

TRAK
- AT-D2 $389.00
- AT-D4 $539.00

HEWLETT PACKARD

- **41CV $189.99**
- **41CX $249.99**
- **HP 71B $419.99**
- HP 11C $62.99
- HP 12C $92.99
- HP 15C $92.99
- HP 16C $92.99
- HP 75D $999.00
- HPIL Module $98.99
- HPIL Cassette or Printer $359.99
- Card Reader $143.99
- Extended Function Module $63.99
- Time Module $63.99

NEC

- **PC-8201 Portable Computer $449.00**
- PC-8221A Thermal Printers $149.99
- PC-8281A Data Recorder $99.99
- PC-8201-06 8K RAM Chips $105.99
- PC-8206A 32K RAM Cartridge $329.00

SHARP

- **PC-1500A $165.99**
- **PC-1250A $88.99**
- CE-125 Printer/Cassette $128.99
- CE-150 Color Printer Cassette $171.99
- CE-151 4K RAM $29.99
- CE-155 8K RAM $49.99
- CE-161 16K RAM $134.99
- CE-500 ROM Library ea $29.99

commodore

- **SX-64 Portable $839.00**
- **CBM 8032 $639.00**
- CBM 8096 $869.00
- CBM 9000 $999.00
- B128-80 $769.00
- 8032 to 9000 Upgrade $499.00
- 2031 LP Disk Drive $299.00
- 8050 Disk Drive $999.00
- 8250 Disk Drive $1249.00
- 4023 Printer $399.00
- 8023 Printer $589.00
- 6400 Printer $1449.00
- Z-RAM $369.00
- Silicon Office $499.00
- The Manager $199.00
- SoftROM $125.00
- VisiCalc $159.00
- **CBM 64 $199.00**
- C1541 Disk Drive $249.00
- C1530 Datasette $69.99
- C1520 Color Printer/Plotter $129.00
- M-801 Dot Matrix Printer $219.00
- C1526 Dot Matrix/Serial $299.00
- C1702 Color Monitor $259.00
- C1311 Joystick $4.99
- C1312 Paddles $11.99
- C1600 VIC Modem $59.99
- C1650 Auto Modem $89.99
- Logo 64 $49.99
- Pilot 64 $39.99
- Word Pro 64 Plus $59.99
- Calc Result 64 $65.99
- Calc Result Easy $39.99
- MCS 801 Color Printer $499.00
- DPS 1101 Daisy Printer $459.00
- Magic Voice Speech Module $54.99
- Desk Organizer Lock $49.99
- Vidtex Telecommunications $34.95

MSD DISK DRIVES
- SD1 Disk Drive $349.00
- SD2 Disk Drive $599.00

PROFESSIONAL SOFTWARE
- Word Pro 2 Plus $159.00
- Word Pro 3 Plus $189.00
- Word Pro 4 Plus/5 Plus each $279.00
- Info Pro $179.00
- Administrator $399.00
- Power $79.99

PRECISION SOFTWARE
- Superbase 64 $59.99

PERSONAL PERIPHERALS
- Super Sketch Graphics Pad $39.99

maxell

- 5¼" MD-1 $19.99
- 5¼" MD-2 $26.99
- 8" FD-1 $39.99
- 8" FD-2 $49.99

VERBATIM
- 5¼" SS/DD $26.99
- 5¼" DS/DD $34.99

BIB
- 5¼" Disk Head Cleaner $14.99

Dennison

- Elephant 5¼" SS/SD $15.99
- Elephant 5¼" SS/DD $17.99
- Elephant 5¼" DS/DD $24.99
- Elephant EMSP 5¼ $34.99

DISK HOLDERS
INNOVATIVE CONCEPTS
- Flip-in-File 10 $3.99
- Flip-in-File 50 $17.99
- Flip-in-File 50 w/lock $24.99
- Flip-in-File (400/800 ROM) $17.99

COMPUTER MAIL ORDER

west
800-648-3311
In NV call (702)588-5654
Order Status Number: 588-5654
P.O.Box 6689, **Dept.105**
Stateline, NV 89449

canada
Ontario/Quebec 800-268-3974
Other Provinces 800-268-4559
In Toronto call (416) 828-0866
Order Status Number: 828-0866
2505 Dunwin Drive, Unit 3B, **Dept.105**
Mississauga, Ontario, Canada L5L1T1

east
800-233-8950
In PA call (717)327-9575
Order Status Number: 327-9576
Customer Service Number: 327-1450
477E.3rdSt., **Dept.105** Williamsport,PA 17701

CANADIAN ORDERS: All prices are subject to shipping, tax and currency fluctuations. Call for exact pricing in Canada.
INTERNATIONAL ORDERS: All orders placed with U.S. offices for delivery outside the Continental United States must be pre-paid by certified check only. Include 3% (minimum $5) shipping and handling.

the other comparison instructions don't store their results anywhere, but they *do* condition the N, Z, and C flags in a special way that facilitates branching after the comparison. Read up on the CMP, CPX, and CPY instructions for full information on how they set the flags.

We're now at the end of our flag-waving tour. If you kept with us this far, you're in the know about some elementary but important attributes of the processor status register, and you may have improved your knowledge of your monitor. Dig into those ML texts that you didn't understand last time, and you'll be surprised how easy they've become. If you're really feeling like an expert, come up with a branch instruction to take our program back to $3300.

Decoding Status Displays

First Digit		Second Digit	
0		0	D I Z C
1		1	D I Z **C**
2	N V - B	2	D I **Z** C
3	N V - **B**	3	D I **Z** **C**
4		4	D **I** Z C
5		5	D **I** Z **C**
6	N **V** - B	6	D **I** **Z** C
7	N **V** - **B**	7	D **I** **Z** **C**
8		8	**D** I Z C
9		9	**D** I Z **C**
A	**N** V - B	A	**D** I **Z** C
B	**N** V - **B**	B	**D** I **Z** **C**
C		C	**D** **I** Z C
D		D	**D** **I** Z **C**
E	**N** **V** - B	E	**D** **I** **Z** C
F	**N** **V** - **B**	F	**D** **I** **Z** **C**

This table decodes two-digit hex displays of the processor status register. Bold face indicates bit set; regular face, bit clear.

Machine Language Demonstration Program

```
3300 D8       CLD
3301 18       CLC
3302 A9 00    LDA #$00
3304 AA       TAX
3305 A8       TAY
3306 C9 FF    CMP #$FF
3308 00       BRK
3309 A9 80    LDA #$80
330B 00       BRK
330C A9 7F    LDA #$7F
330E 00       BRK
330F A9 00    LDA #$00
3311 00       BRK
3312 A9 FF    LDA #$FF
3314 00       BRK
3315 69 01    ADC #$01
3317 00       BRK
3318 69 01    ADC #$01
331A 00       BRK
331B C9 02    CMP #$02
331D 00       BRK
```

©

INSIGHT: Atari

Bill Wilkinson

Comparing Printers

After disk drives, probably the most frequently purchased peripheral for personal computer systems is a printer. But buying a printer is a lot harder than buying a disk drive. Usually your choice of drives is limited to the computer manufacturer's own unit plus a few produced by third-party companies. And despite some slight differences, they all deliver similar performance.

But printers are another story. There are hundreds of printers on the market for personal computers. Most of them can be made to work with your Atari. And they vary widely in terms of price, performance, features, and compatibility.

One of the main differences between printers is their printing speed. Usually this is measured in characters per second, abbreviated cps. By comparing the speed ratings, you can decide whether a certain printer is fast enough for your applications. But recently I discovered how misleading those speed ratings can sometimes be. It all started when those of us at Optimized Systems Software (OSS) began looking around for a new printer.

To begin, let me tell you that we have a rather unique requirement for a printer: We needed a good, fast, reliable printer which we could hook up to any of several computers. And, of course, it had to be compatible with all our software: several languages, four different operating systems, and a couple of word processors.

It is also time for a bit of history. For the last couple of years, our mainstay printer has been a venerable DEC LA-120 Decwriter. This is actually a printing terminal (remember, from the days of mainframe timesharing?) which operates via a serial RS-232-C connection at 120 cps. As reliable as this beast has proven to be, it has a few problems: Its print quality is marginal at best, without even descenders on lowercase letters; because it uses a serial instead of the more standard parallel interface, much software simply will not work with it; although it is rated at 120 cps, it is actually capable of only about 105 to 110 cps when printing typical documents.

At the time, the only other printers we had (or had significant experience with) were a Diablo daisywheel (also serial, at 30 cps), an Atari 825 (rated at 60 cps), and a C. Itoh Prowriter (rated at 120 cps). All had performed adequately (or, in the case of the Prowriter, more than adequately), but all were too slow for our purposes.

And, of course, software compatibility was another big issue. Our primary problem in the past had been that some of our computers transmitted a linefeed after a carriage return (for example, the CP/M based machines), while others (our Atari computers) did not. We were well aware, also, that more problems would be coming as we acquired more software and wanted more capabilities.

Instantaneous Vs. Continuous Speed

For the sake of compatibility then, the first printer that came to mind was the Epson MX-80. Why? Simply because it is used on so many machines with so much software. Yet we immediately rejected the MX-80. Rated at only 80 cps, it is simply way too slow for our applications.

So we started looking for a *fast* printer which would be largely compatible with the MX-80. To make a long story short, we bought an Epson FX-100, a wide-carriage version of the FX-80. Imagine our surprise when this printer, rated at 160 cps, was only marginally faster than the Prowriter and actually *slower* than the Decwriter!

It turns out that with few exceptions, the printer speeds published by manufacturers and often faithfully reported by magazines are the *maximum instantaneous* speeds of which a machine is capable. This instantaneous speed rarely correlates to the actual number of lines a printer will produce in a minute.

What's more, even those companies which do admit that speed ratings are maximum values employ other claims to suggest that their printer is faster than the competition. For example, many claim that because their printers are *bidirectional* or *logic-seeking*, they are faster than the old-fashioned machines which print in only one direction (*unidirectional*).

Let me describe how the FX-100, for example, prints a typical program listing. First, it receives and prints a line (say, 50 characters), moving the print head from left to right, stopping at the end of the line. Then, it receives the command to print the next line (say, 70 characters). It moves the print head to the seventieth column, stops, advances the paper to the next line, and prints backward from right to left. If the next line is indented (mine often are), it goes through the same sequence of stopping, moving the head, and advancing the paper once again.

But stopping, starting, moving paper, and starting again all take time. A lot of time compared to the actual printing time. Printers like the Prowriter, on the other hand, contain an internal buffer which they use intelligently. After printing a 50-character line, it checks to see where the right end of the next line needs to be and automatically continues to move the head to that position. One stop-and-start sequence eliminated. The results? See for yourself in the following chart, which records the time it took for three different printers to print the same moderate-length program listing:

Printer	Rated Speed (cps)	Time Required	Approx. Actual Speed (cps)
Decwriter	120	6 min 30 secs	110
Prowriter	120	7 min 45 secs	90
FX-100	160	7 min 30 secs	95

Oh, yes. Did I forget to mention that the Decwriter has no logic-seeking and prints unidirectionally only? That's a lot of stopping and starting. Sometimes raw power can accomplish what "logic" can't.

Throughput: True Speed

Well, I would like to report that we ran out and bought 30 or 40 different printers and tested them, too, just so I could bring you a full comparison chart. But our budget at OSS won't stretch that far.

I did, however, go to several dealers and informally time the speed of various printers. Since I had a couple of reference points (the speeds of the Prowriter and FX-100), it wasn't too hard to get a fair idea of true *throughput* figures: the printing speeds they could actually sustain.

Then I discovered another trick used by a few manufacturers. Many printers are capable of two or three (or more) character widths or fonts (typically 10, 12, and 17 characters per inch). It seems to me that at least a few printers are rated only for their smallest (and hardest to read) fonts.

Luckily we had an understanding dealer who allowed us to "trade up" our FX-100. And what printer did we then buy? Actually, we ended up buying two.

Because of our need for a printer capable of using the vast library of MX-80-compatible software, we got an Epson MX-100 (simply a wide-carriage MX-80). We have been very happy with it, though I am sure any of several MX-80-compatible printers would have done as well. True, the MX-80 is slow. But its throughput rate seems to be around 50 to 60 cps, which is respectable compared to its rated speed.

Because we needed speed, though, we disregarded MX-80 compatibility for our other new printer, an Okidata 2350 (the model number seems to reflect its retail price). It is rated at over 300 cps and surprised us by performing our little speed test in 1 minute 55 seconds, for a throughput rate of over 360 cps. However, sometimes it gets too hot while printing long listings and stops to wait for the head to cool off. Even so, it probably has a throughput rate of 300 cps or more.

So, did you learn anything from our experiences? I sincerely hope so.

When shopping for a printer, ask to see a demonstration of its speed. Many printers perform better with uniform-length lines (such as those produced by a word processor), so ask to see a program listing also. And make your own time trials.

Judge the print quality for yourself. Ask about replacement ribbon costs. (We found one printer that worked only with carbon ribbons. $$$$! But if you need good print quality, it might be worth it.)

Above all, be certain a particular printer is compatible with your computer *and* software. Few things are worse than saving $50 on a printer only to find out you have to spend another $100 because your current word processor isn't compatible with your new printer.

Of Memory And Machines

We've received a few letters recently on seemingly different subjects, but which all relate to what is obviously some confusion and uncertainty about the Atari XL computers. Let's address these letters and, at the same time, shed some light on the workings of these little gems.

First, Jacqueline Patton of San Antonio, Texas, asks whether she is "stuck with a problem computer [1200XL] and an unreliable disk drive [Atari 1050]." We'll discuss the 1200XL's compatibility problems in a moment. First, a few words about the 1050.

I have not personally observed the 1050 to be any more or less reliable than any other drive on the market. Disk drives, in general, tend to be like automobiles: Sometimes you get one which goes 100,000 miles with no maintenance, and sometimes you get a lemon, but most often you get one which will last a reasonable time with reasonable care and regular checkups. This is not surprising: Disk drives and cars are both mechanical nightmares, subject to extremely close manufacturing tolerances and acute material stresses.

If the 1050 has a problem, it may be simply that it cannot read all of the more strangely protected software disks that are flooding the market. There are antipiracy measures in use today that try the limits of many drives and their controllers. Yet most programs will load fine on any good Atari-compatible drive, including the 1050.

My objections to the 1050 are centered around only one point: Although every other Atari-compatible drive manufacturer has complied with the Percom-standard double-density format (derived in turn from Atari's defunct double-density 815 drive), only Atari chose to be different. Further, Atari's method gives you a maximum of 128K bytes per disk. The others get 180K bytes. There is no excuse for this. It results from Atari's typical blindness when it comes to outside vendors.

All this does not mean the 1050 is no good. It just means that, on a bytes per dollar basis, it is overpriced.

Use Your Options

Another letter, from Shahid Ahmal of London, England, was actually a complaint to OSS about the fact that some programs (including our disk-based *MAC/65*) would not load and run properly on his 800XL. The problem is that these programs require you to remove the BASIC cartridge before booting up—impossible on the 800XL and 600XL, since the BASIC "cartridge" is built into the newer computers as a standard feature. His solution was to write a program which switched off the built-in BASIC, changed RAMTOP, and closed and reopened the screen driver.

Whew! I am impressed. Doing all that in the proper order is not easy. But there really is a much simpler way.

This discussion applies only to disks containing programs which *do not* use Atari BASIC. Obviously, such things as assemblers, compilers, and utility programs fit this category. Not so obviously, many game disks will not run if Atari BASIC is present. In any case, if you own an 800XL (or, I assume, a 600XL with expanded memory), and the directions for a disk or program tell you to remove your BASIC cartridge, try this:

Turn on power to all devices *except* the computer. Insert the disk you wish to boot. Push *and hold down* the OPTION button. Turn on the computer's power. When the disk starts to load, you can release the OPTION button.

This has the effect of disabling the built-in BASIC. Atari's manuals tell you all this. But they don't emphasize enough that you should try this with any disk/program if it otherwise doesn't work. And they don't tell you about the OPTION button when used with the *Translator Disk*. "What's that?" Glad you asked . . . .

I have mentioned the *Atari Translator Disks* before in this column, but only part of what I'll add is repetition.

If you own an Atari 1200XL, 800XL, or expanded 600 XL with a disk drive, run—do not walk—to your nearest Atari users' group and purchase (usually for about $10) the pair of *Atari Translator Disks*. (You may still be able to get them from Atari directly.)

The instructions tell you to boot the version A disk first, wait for it to give you a message, insert your otherwise unbootable disk, and push the SELECT key. If that doesn't work, you are supposed to try the version B disk. (Both disks actually load an old Atari 800-style operating system into your XL machine's memory, thus hopefully assuring compatibility with programs that rely on the older operating system.)

What the instructions don't say is that you may also need to hold down the OPTION button. Why? Because otherwise, good old Atari BASIC is still there, messing up the memory address space.

Six Ways To Boot

There are, then, no less than six ways to try booting a disk on an XL machine: with or without holding down the OPTION button alone or in combination with either of the two *Translator Disks*. This sounds like a real pain, but once you find the method that works with a given disk, you can write it down for future reference.

I should note that all of these methods still result in compatibility with only about 97 percent of all software (85 percent of heavily protected software). Is there anything you can do if your favorite piece of software won't boot using any of these methods? Yes, two things.

First, you can write, phone, telex, or otherwise cajole and threaten the software manufacturer. I have said before, and I am sure I will go hoarse saying again, that I believe the responsibility for the lack of compatibility does *not* rest with Atari. No other manufacturer has ever produced a series of computers with as many changes and improvements as the XL line and yet maintained as much compatibility as has Atari.

Second, you can try one of the commercial translator programs. I am aware of two at this time: *XL BOSS* from Allen Macroware and *XL FIX* from Computer Software Services. I have used neither, so I cannot comment on them. However, I recommend that to avoid unnecessary expense you should certainly seek verification from these manufacturers that the particular software package you want to use will work correctly with their product.

And More Memory

The commercial translator programs do have one interesting bonus: They give your XL computer an extra 4K of memory. Let's see why.

The original Atari 400 and 800 computers had a 10K operating system and a 2K input/output space. Since the maximum RAM they supported was 48K, that left 4K unused in the total address space of 64K (unless you bought a third-party RAM board—such as those from Mosaic—which placed RAM in this unused space). The empty 4K was located at address $C000 (49152), just above the normal 48K RAM.

When the XL computers arrived, they sported more graphics modes, device downloaders, parallel bus support, self-diagnostics, and more, all of which pushed the size of the operating system up to 14K. Guess where Atari got the extra 4K from? Yep. No more "unused" space.

However, the commercial translators effectively emulate the original 10K operating system, leaving that 4K free again. But since an XL machine has 64K of RAM, the unused space becomes free RAM. If you are using a cartridge-based program (even the built-in Atari BASIC), this isn't a real big help. The 4K of RAM is still at address $C000, above the cartridge address space. You could install machine language routines here, use it as a buffer for disk I/O or player/missile graphics, or even use it for any graphics screen up to the size of that of GRAPHICS 7. But the average beginner will have a hard time using this space.

On the other hand, programs which don't use a cartridge don't have this restriction. For example, if you use one of these translators to load *VisiCalc* into an XL machine, you'll gain 4K of valuable spreadsheet space. Try it sometime. It's easy.

And one more comment before we pause until next month: Since the *Atari Translator Disks* work much like the commercial translators, it may just be possible to modify them and gain the same 4K of RAM. I have not had the time to investigate this, but if any COMPUTE! readers discover anything in this regard, we'd be happy to hear about it. ©

MACHINE LANGUAGE

Jim Butterfield, Associate Editor

Stack Tricks

The 6502 stack sits quietly in page 1 (typically addresses $01FA down to about $0140) and works behind the scenes. If you call a subroutine using JSR, a couple of entries push their way onto the stack; they pop back off when RTS is used. Everything is tidied up, and we don't need to think about the stack workings most of the time.

Once in a while, however, we want to squeeze a little more performance out of the stack. We may read the stack pointer by transferring it to the X register with TSX, or even set it by transferring the other way with TXS. We may set up a dummy return address by pushing values to the stack before an RTS. Often such tricks are more trouble than they are worth, but sometimes they can be useful.

A Subroutine Limitation

An early 6502 text suggested that an easy way to pass data to a subroutine would be to place it on the stack. It can be done, but it's not easy; I tend to discourage this kind of coding for beginners.

Here's the problem: You take one or more values and place them on the stack using the PHA (PusH A) command, then call a subroutine. The idea is that the subroutine can simply pull these values from the stack with PLA (PuLl A) and use them, but that won't work. When the subroutine is called with JSR, the last two values placed on the stack are the subroutine return address (to be exact, the address minus 1). So the pull command gets, not the data, but the return address. Annoying.

There are a couple of ways around the problem, but they are clumsy. First, you can pull the return address (two bytes) from the stack and save them. Then the data bytes are pulled and saved. Finally the return address is recalled and put back on the stack. That's a lot of work. It would be easier to have the calling routine store the data somewhere.

The second method is a little more workable, but still clumsy. If the stack pointer is transferred to the X register with TSX, we may now look directly at the stack as it lies in page 1. An instruction such as LDA $0100,X would look at the stack memory area, but would miss the real stack: The effective address would be of the first "empty" stack location. We'll have to climb a little higher to see the "live" stack. For example, LDA $0101,X would look at the last item on the stack; LDA $0102,X would look at the previous item, and so on.

Back to our original problem. There's a byte of data on the stack, behind a subroutine call. We can read it with TSX followed by LDA $0103,X. But we can't remove it from the stack without setting up a loop to repack everything. We can also change this stack item with a STA command. When the subroutine returns, the main routine must pull the extra item back from the stack.

It's often more trouble than it's worth, but it does work. A small example will illustrate.

This routine prints a triangle of asterisk signs. There are better ways to do the job, but it does illustrate moderately advanced stack work.

```
033C A9 01     LDA #$01    ;start count at 1
033E 48        PHA         ;pass to the stack
033F 20 4B 03  JSR $034B   ;call print subrtn
0342 68        PLA         ;get back the count
0343 18        CLC
0344 69 01     ADC #$01    ;add one to count
0346 C9 10     CMP #$10    ;stop at 16
0348 90 F4     BCC $033E   ;else do it again
034A 60        RTS
          ; SUBROUTINE TO CHECK STACK
034B BA        TSX         ;get the pointer
034C BD 03 01  LDA $0103,X ;dig out the count
034F A8        TAY         ;put it in Y
0350 A9 2A     LDA #$2A    ;asterisk character
0352 20 D2 FF  JSR $FFD2   ;print it
0355 88        DEY         ;count down
0356 D0 FA     BNE $0352   ;if more, go back
0358 A9 0D     LDA #$0D    ;carriage return
035A 20 D2 FF  JSR $FFD2   ;print it
035B 60        RTS         ;quit
```

Call the above program from BASIC with SYS 828.

If you'd rather enter the program as BASIC DATA statements, the following program will do the job:

```
100 DATA 169,1,72,32,75,3,104,24
110 DATA 105,1,201,16,144,244,96
120 DATA 186,189,3,1,168,169,42
130 DATA 32,210,255,136,208,250
140 DATA 169,13,32,210,255,96
200 FORJ=828TO861
210 READX
220 T=T+X
230 POKEJ,X
240 NEXT J
```

```
250 IFT<>3911THENSTOP
260 SYS828
```

More Muscle

Perhaps a more useful task for the stack is to streamline frequently used subroutines. For example, if there's a popular subroutine that I call a dozen times or more, it will be in my interest to make the calling sequence as brief and easy as possible.

Here's a common one. I often need to print various messages, and expect to use a subroutine to do it. The normal calling sequence would be to load the address of the particular message into a couple of registers—say, A and Y—and then have the subroutine use this address to print the message. This means that the subroutine will have an overhead of two instructions: the LDA and LDY before the call. The overhead might in fact be greater: I might need to save previous values in A and Y in order to continue my program after the message is printed.

Suppose I could do this: just call the subroutine, and leave the *message itself* behind the calling routine. I could flag the end of the message text with a zero byte. Now, if I could make the subroutine smart enough to go after this message, I could save a lot of setup coding.

Not too hard. The subroutine would need to pull the return address from the stack and set it into an indirect address. The return address would need to be adjusted by a value of 1, since it has a built-in offset. Now the subroutine could walk through the message, printing out the characters as it found them. When it finds a zero, it's time to return; but we must adjust the return address so that we'll go to the address *behind the message*. All this takes a little careful work, but we can do it.

More Complex

Now let's make the task a little more complicated. Not only do we want our subroutine to print the message located behind the JSR instruction; we want it to do this without affecting any registers—A, X, or Y.

The natural thing to do is to push A, X, and Y to the stack, using the sequence PHA:TXA:PHA:TYA:PHA; just before we return, we'll pull everything back and restore the original register values. If we do this, however, we can't pull the return address from the stack, since it's buried beneath the new stuff we have just stacked. If we go this way, we must dig out the return address from midstack, using TSX and so on.

This kind of coding has been seen in various application programs; it's not new and revolutionary, just a little more careful work.

Commodore is using this technique for the first time in the ROM of its new computer series, the Commodore 16 and the Plus/4. You can track the coding in one of the machines by using the built-in machine language monitor. Start the disassembler at address $FBD8 with command DFBD8. You'll see code along the following lines:

Save all registers to the stack:

```
PHA:TYA:PHA:TXA:PHA
```

Copy the stack pointer, and adjust it to match the return address:

```
TSX:INX:INX:INX:INX
```

Copy the return address to zero page, so that it can be used as an indirect address:

```
LDA $0100,X:STA $BC:INX:LDA $0100,X:STA $BD
```

The indirect address in $BC and $BD is one too low, since a JSR return is offset by one. Add one to it:

```
BUMP INC $BC:BNE PASS:INC $BD
```

Get a character—it will come from behind the calling JSR instruction. If it's zero, we're finished and go to EXIT:

```
PASS LDY #$00
GETCH LDA ($BC),Y:BEQ EXIT
```

If it's not zero, print it; then go back to bump the address and get another one:

```
JSR $FFD2:INY:BNE GETCH
```

Y will never reach 255 (no messages are that long), so the BNE is an "always" branch. If we reach EXIT, we must get the count of characters from Y:

```
EXIT TYA
```

Now we recompute the position of the return address in the stack:

```
TSX:INX:INX:INX:INX
```

We add the count to the indirect address, and put the new return address directly into its place in the stack:

```
CLC:ADC $BC:STA $0100,X
LDA #$00:ADC $BD:INX:STA $0100,X
```

And finally, we restore our three registers and return:

```
PLA:TAX:PLA:TAY:PLA:RTS
```

For many of us, this type of stack manipulation is overkill. It makes programs hard to disassemble for study purposes, and the memory saving on small programs is negligible. For that matter, what are you going to do with the few dozen bytes you save?

Nevertheless, it can be a great coding convenience to allow a programmer to simply "drop" his data in line with the coding. This can save extra coding for setup, extra labels—and possible mistakes.

And it can be satisfying and fun to know that you can get that extra ounce of control over the workings of your computer. ©

C. Regena

Algebra Tutorial Part 2

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

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

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

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

Helping Where It Is Needed

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

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

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

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

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

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

Presenting Problems

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

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

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

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

Customizing The Tutorial

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

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

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

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

Algebra Tutorial, Part 2

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

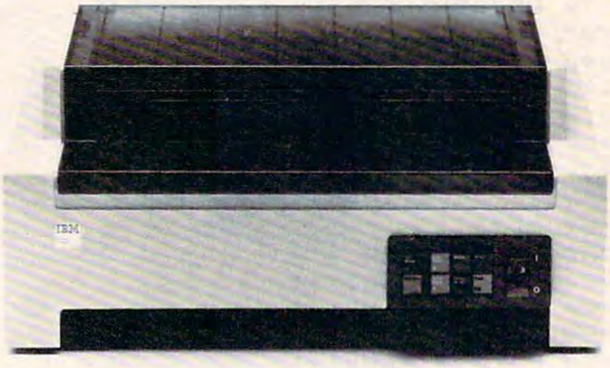
```
1740 L=LEN(P$)
1750 IF L=2 THEN 1770
1760 CC=CC+1
1770 C$=""
1780 FOR I=1 TO L
1790 C=CC+I
1800 GOSUB 1620
1810 C$=C$&CHR$(K)
1820 NEXT I
1830 IF C$=P$ THEN 1870
1840 GOSUB 1580
1850 CALL HCHAR(23,CC+1,32,L)
1860 GOTO 1770
1870 RETURN
1880 SD=(-1)^(INT(2*RND)+1)
1890 SD$="+"
1900 IF SD=1 THEN 1920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

©

IBM Personal Computing

Donald B. Trivette

Editor's Note: We're pleased to welcome a new column to COMPUTE! *this month—Donald B. Trivette's "IBM Personal Computing." Trivette will be covering topics of interest to users of all IBM microcomputers, including the IBM PC, PCjr, PC-XT, and Portable PC. (Much of the information will be useful to owners of compatibles, too.) Based in North Carolina, Trivette is a freelance writer, author, and consultant whose work has appeared in such magazines as* INC, Business Computer Systems, PC World, *and* Softalk. *He also wrote the "Getting Down To Business" column in* COMPUTE!'s PC & PCjr *magazine. From 1969 to 1981 he taught computer science courses and was the director of computing services for the University of North Carolina at Wilmington, and founded its Computing Center. His most recent book* A BASIC Primer for the IBM PC *(Scott, Foresman & Co.).*



If you need a hard copy of your color graphics, the IBM Color Printer can print at speeds up to 200 characters per second in eight colors.

Hard Copy Color Graphics

There are two basic ways to generate a hard copy of the color graphics on your display screen: photographic techniques and color printers. Photography is ideal if you need a slide to project at a business meeting or presentation, but it requires special equipment and at least some photographic skill. That's why many people opt for the more direct method, color printers.

One of two technologies is used to print a color image. The printer either sprays colored ink through a tiny jet onto the paper (ink-jet technology), or it hammers the image onto the paper with a multicolored ribbon (impact technology). Although there are several IBM-compatible color printers made by independent companies—including Quadram's Quadjet (ink-jet) and Centronics Data Computer's Model 358 (impact)—this month we'll discuss the IBM Personal Computer Color Printer, Model 5182. It's a dot-matrix impact printer that plugs into the parallel interface of a PC or PC-XT. (It also works with a PCjr equipped with a parallel interface, but not all software supports the Color Printer when attached to a Junior.)

The IBM Color Printer produces color printouts with a cartridge ribbon that has four bands. It works like the old two-band black and red typewriter ribbons, except the printer automatically switches from one color to another and, of course, there are more colors. The Process Ribbon cartridge ($19.95) has yellow, magenta (red), cyan (blue), and black bands. By printing one color on top of another, it can produce four additional hues—orange, green, violet, and brown. For example, when yellow is printed over magenta, the result is orange.

The Primary Ribbon cartridge (also $19.95) is suitable for less colorful printouts. Its four bands are also red, blue, green, and black, but no overprinting is allowed.

Finally, for everyday correspondence, you can buy an all-black cartridge ribbon ($12.95). It has automatic band shifting to extend the ribbon's life and snaps into place just like the other cartridges.

The IBM Color Printer is more than just a dot-matrix printer with a color ribbon. It has three speeds for printing text in various qualities. At 200 characters per second (cps) the print quality is good enough for preliminary drafts, informal notes, and program listings. At 110–150 cps the dot matrix is printed more densely to improve the character images, resulting in what is sometimes called correspondence quality. At 30–40 cps the printing approaches letter quality. In fact, the dot matrix pattern is hardly visible unless you examine the characters very closely.

Figure 1: Text Samples Generated On The IBM Color Printer

```
DATA-PROCESSING QUALITY:
   Printing Speed:        200 characters per second
   Print Head Travel:     Bi-directional, one pass
   Printing Sizes:        10 characters per inch (fixed spacing)
                          12 characters per inch (fixed/proportional spacing)
                          17.1 characters per inch (fixed/proportional spacing)
TEXT QUALITY:
   Printing Speed:        110 to 150 characters per second
   Print Head Travel:     Bi-directional, one pass
   Printing Sizes:        10 characters per inch (fixed/proportional spacing)
                          12 characters per inch (fixed/proportional spacing)
                          17.1 characters per inch (fixed/proportional spacing)
NEAR-LETTER QUALITY:
   Printing Speed:        30 to 40 characters per second
   Print Head Travel:     Uni-directional, two passes
   Printing Sizes:        10 characters per inch (fixed/proportional spacing)
                          12 characters per inch (fixed/proportional spacing)
                          17.1 characters per inch (fixed/proportional spacing)
```

The IBM Color Printer can also print in boldface type, automatically justify text, underline, print subscripts and superscripts, and space proportionally. Three pitches are selectable to print 10, 12, or 17.1 characters per inch. In graphics mode the printer has a resolution of 168 dots per horizontal inch and 84 dots per vertical inch—more than enough to represent the circle of a pie chart or the bud of a rose.

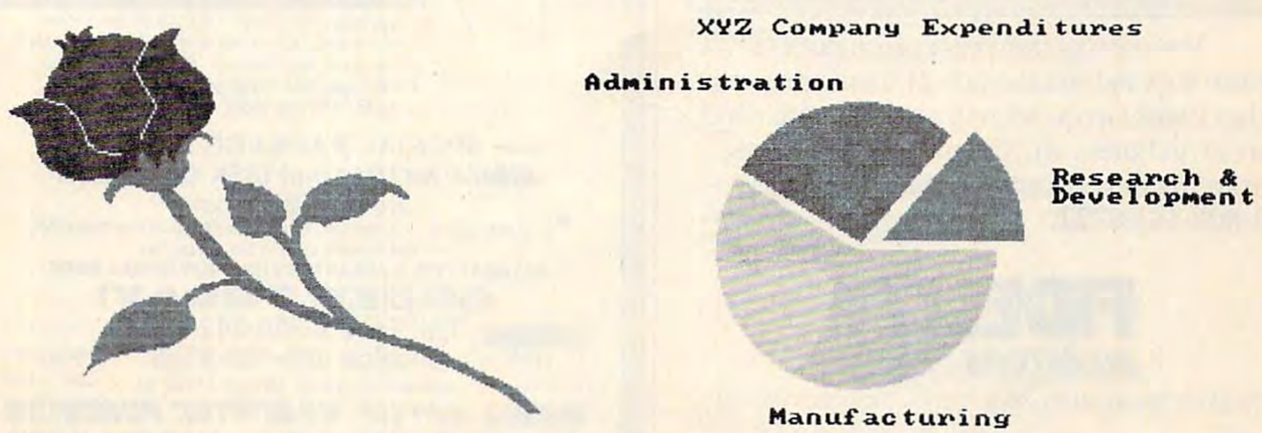
Color Limitations

Since the PC can generate 16 colors and the Color Printer can reproduce only eight, there is a slight incompatibility. We say "slight" because actually the PC generates only eight basic colors; colors 8–15 are simply brighter luminances of colors 0–7. Therefore, the Color Printer reproduces the high-intensity colors as their low-intensity equivalents. Unfortunately, background colors are not reproduced. The colorful bar graph you've created with a blue background will have a white background unless you can find blue computer paper.

Enough about the technology of color printing—the real question is how *easy* is it to get an image from screen to paper? By using a little-known IBM utility program, it's as easy as pressing two keys. The *IBM Personal Computer Print Screen Utility Program* (product #6024186, $35) sends a screen of text or graphics, color or monochrome, from the display buffer to the Color Printer when you press the Shift-PrtSc keys. You run the utility program once when starting up DOS, and then it remains quietly in memory until it's called forth to print a screen. It must be reloaded each time DOS is rebooted. Unfortunately, it works only with the eight-color Process Ribbon.

Some software, particularly graphics software, has color printer options built-in. *IBM Graphing Assistant* is one of these—it directly supports the IBM Color Printer. Simply issuing

Figure 2: Graphics Dumps Made With The IBM Color Printer



the print command reproduces the graph on the screen in color. Two excellent graphics programs from Digital Research, *DR Draw* and *DR Graph*, also support color printers, but not the IBM Color Printer. At least, not yet. A spokesperson for Digital Research says the company plans to add an IBM Color Printer option for both of these popular programs.

Color printers vary in price from $900 to $9000; the IBM Color Printer retails for $1995. The quality, as the accompanying printout shows, is quite good. (In fact, this is the printer that COMPUTE! recently started using to make program listings for IBM, Atari, Apple, Texas Instruments, and the TRS-80 Color Computer.) If you frequently need color graphics hardcopy, the output from a color printer may be better than a screen photograph.

Tougher Rules For Computer Deductions

Congress needs (wants) more money, so they recently changed the rules for deducting the purchase of personal computer equipment as a business expense. It used to be fairly easy to qualify for this deduction. But the 1984 Tax Bill limits the tax write-off for an employee who buys a computer to do company work at home.

Under the old rule, you could deduct the full cost of the computer if it were used exclusively for business, and a proportional amount if it were used for both business and pleasure. Formerly I advised people that keeping a log of use would help document their claims should the Internal Revenue Service (IRS) come calling.

It's no longer that simple. According to a story in *The Wall Street Journal*: "For purchases made after June 18, employees won't be able to claim any business deduction unless the computer is 'required for the convenience of the employer and as a condition of employment.' " The story goes on to say that a letter from your employer stating that you are required to have a computer at home *will not* be sufficient to support a tax deduction.

It seems Congress anticipated that your employer might be *too* willing to provide such a letter. So even with a letter, you'll have to prove to the IRS that you really need a computer at home. A log of personal and business use is now *required*.

Two questions an IRS auditor would likely raise: "Why didn't you stay late at the office and use the equipment there?" and "If this is really a condition of employment, why didn't the employer purchase the computer for you?" Have your answers ready—April 15, 1985 isn't that far away. ©

Enhanced Commodore 64 DOS Support

Stephen S. Melsheimer

The wedge program that comes with every Commodore disk drive makes input/output much simpler. "Enhanced Commodore DOS Support" takes the wedge a few steps further by adding APPEND and VERIFY commands, allowing the use of wedge commands within a BASIC program, and several other features. There are also instructions on how you can further customize your wedge.

The Commodore 1541 disk drive used with the Commodore 64 and VIC-20 is *intelligent*—the disk unit contains a 6502 microprocessor, 16K of ROM holding the disk operating system (DOS), and 2K of RAM that serves as a buffer for information going to or from the disk. Thus, the computer is freed from the chore of managing the disk operations, and no computer memory is appropriated for a disk operating system when a disk drive is added.

Unfortunately, there are no simple commands in the *computer's* operating system to provide simple communication with the disk and its DOS. For example, there is no SCRATCH command to delete a file from the disk. Instead, you must use a cumbersome statement like:

OPEN 1,8,15,"S0:*filename*":CLOSE 1

To make up for this, Commodore provides a DOS support program named "DOS 5.1" (and a simpler "VIC-20 Wedge" for use with the VIC) to facilitate use of the disk. These programs appear on the *TEST/DEMO* disk which comes with the drive. This DOS wedge program is not needed to operate the disk, and adds no extra capabilities beyond those already present in the disk drive DOS ROM. What it provides is a set of shorthand commands that make things easier for the user. These commands do provide features that are vast improvements over what is possible directly from BASIC. In particular, the ability to display the disk directory without disturbing the program currently in memory is a great convenience.

The Theory Of Wedging

Programs like "DOS 5.1" are called *wedge* programs because they are wedged into the stream of BASIC interpreter processing. Central to the operation of BASIC is a subroutine called CHRGET (located at addresses \$73–\$8A in the 64 and VIC). This subroutine gets characters from a BASIC statement and delivers them to the interpreter. A wedge program intercepts each character and inspects it to see if it is a symbol recognized by the wedge. If not, control immediately returns to BASIC.

The "DOS 5.1" wedge also looks in the microprocessor stack and checks the return address of the BASIC routine which called CHRGET. If it is not an address which indicates the start of a new statement, control returns to BASIC. This allows the symbol characters to have their normal meaning in the middle of a statement. Finally, "DOS 5.1" checks whether BASIC is in direct or program mode, and exits to BASIC if in program mode. Obviously, all of this takes time—a couple of simple benchmark programs took about 15 percent longer to run when the wedge was active, even though it has no useful effect in program mode. Thus, the wedge should be deactivated before running any program where execution speed is important. For those who are curious about the details of "DOS 5.1," Table 1 gives an abbreviated memory map to facilitate exploring it with a machine language monitor.

Extending "DOS 5.1"

"DOS 5.1" provides a table of symbols, and a list of associated vectors that point to the routines for the various functions. Functions can thus be added by altering a vector to point to the new function, and changing the corresponding symbol to the desired character. Since "DOS 5.1" has seven distinct functions, but provides eleven symbols (several are redundant), it is not necessary to delete any existing functions to add new ones. While "DOS 5.1" is very handy, I found that I wanted a few features that were not provided. The resulting "Enhanced DOS Support" program includes APPEND and VERIFY commands, provides a safety feature requiring user confirmation before erasing information on a disk, permits use of DOS commands within BASIC programs, and adds several other features. Still more commands could be added, and procedures for doing this are described below.

APPEND And VERIFY Commands

This revision of "DOS 5.1" originated because I wanted a simple procedure to append a BASIC program on disk to a BASIC program resident in memory. This would allow linking library subroutines or utility programs to other BASIC programs. Mark Niggemann presented an appending technique for the VIC-20 (COMPUTE!, March 1983) which is also applicable to the 64. All that is necessary to append a new program to an existing program is to determine the end address of the program in memory, and alter the starting address of the relocating loader accordingly. A LOAD command then executes as an APPEND. While this method is simple, it uses several direct-mode commands.

To automate the process, I made a patch in the LOAD section of the DOS support program at $CE26 (see the memory map, Table 1). The patch causes the jump to a new routine at $CF5F–$CF7E which alters the starting address in a manner similar to that presented by Niggemann. & was selected as the symbol to be replaced, so to append a program on disk with the name PROG2 to a program in memory, one simply enters

```
&PROG2
```

It should be noted that this is not a *merge*: All statement numbers in the appended program must be higher than those in the program already in memory.

The ease with which APPEND was added encouraged me to add further enhancements to "DOS 5.1." Since the same 64 ROM routine handles both LOAD and VERIFY, it seemed reasonable to add a VERIFY command. The = character was assigned as the symbol. In order to produce the OK or VERIFY ERROR messages, a routine ($CE36–$CE6B) was written to check the error status byte for LOAD/VERIFY errors, and issue the appropriate messages. A typical application of VERIFY is

```
=PROG3
```

which compares PROG3 on disk with the program currently in memory.

The *disk* error status read by the @ command of "DOS 5.1" indicates errors detected within the 1541 disk drive and is completely independent of the computer. Although it is essential to check the 64's status byte on VERIFY (and tape LOAD) operations, there is no great reason to check the status byte on disk LOAD operations, since most potential errors are monitored by the disk unit. However, Enhanced DOS Support reads the Commodore 64 error status on all LOAD/VERIFY operations, and does provide some useful messages even on disk LOADs (for example, BREAK ERROR).

Destruction Warnings

Certain disk commands may be disastrous if executed inadvertently—NEW erases an entire diskette, and SCRATCH deletes a file. BASIC 4.0 (used on larger Commodore PET/CBM computers) issues a warning on such commands, and requires user confirmation before executing them. Since "DOS 5.1" makes issuing commands to the disk very easy (and thus increases the prospect of careless errors), Enhanced DOS Support provides this safety feature for destructive commands issued via the DOS support program.

The portion of "DOS 5.1" ($CD48–$CD6E) that sends such commands to the disk was rewritten, and a new section was added at $CFBD–$CFEF that prints an ARE YOU SURE (Y/N)? message to the screen, and waits for a keyboard response. If N, the command is aborted. This revision also changed the scheme for decoding the commands for deactivating the wedge and for setting the device number. The effect was to free two vectors previously used, compensating for the two used for VERIFY and APPEND. In addition, the command for changing device numbers was simplified to @*n* rather than @#*n*.

Since SAVE&REPLACE also deletes a file, provision was made to issue the warning when the replace option is specified with SAVE operations. The revised SAVE routine is located at $CFF0–$CFFF.

Enabling Program Mode

While Enhanced DOS Support is mainly intended for direct-mode use, a few of the commands can be quite valuable in program mode. Thus, the portion of "DOS 5.1" ($CED1–$CED6) that prevented use of wedge commands within a program was deleted. Several other small changes were also needed to enable the commands to be used in program mode, and the LOAD/VERIFY routine ($CE36–$CE6B) was revised to facilitate program mode use of the % and & commands. With these changes some of the Enhanced DOS Support commands may be used within a program. The symbols =, ↑, and / are *not* allowed as DOS commands. In addition,

the * and ? symbols cannot be used as wild cards in filenames while Enhanced DOS Support is active. In program mode these symbols are all tokenized by BASIC with codes other than the ASCII codes, and thus are not recognized by Enhanced DOS Support.

Of the commands that are operable in program mode, % is especially useful for loading machine language routines or screen images since it does not cause BASIC to restart as a LOAD command does. Thus, constructions like line 100 of Program 1 or line 10 of Program 2, which prevents repeated LOADing, are not needed. The & command was also designed to permit a program to automatically append BASIC subroutines, though using it is a little tricky. A procedure like

```
10 IF(PEEK(2))<>1THENPOKE2,1:&SUBPROG
20 POKE2,0
```

must be used since & restarts BASIC after it is executed, and also CLRs all BASIC variables. Obviously, this should normally be done at the very beginning of the program. Memory location 2 is convenient to use as a flag since it is not used by the computer and has a value of zero on power-up. By using each bit of location 2 as a separate flag, up to eight subprograms could be appended in this manner. Another command that is valuable in program mode is @Q which deactivates the DOS wedge, thus speeding up program execution by about 15 percent. SYS52222 can activate the wedge from within a program, so it can be turned on only when needed.

Note that Enhanced DOS Support commands can be placed in multiple-statement lines, as illustrated in line 10 above. However, they must appear last on any line in which they are used, and thus only one DOS command can be placed on a line. Further, the Enhanced DOS Support symbol must be the first character of a statement for it to be recognized as such.

Other Changes

After SAVE operations, "DOS 5.1" reads the disk error status in order to verify a successful SAVE. A minor annoyance was that the disk status message appeared after the filename without any spacing or punctuation. Revision of $CD9C–$CDAE added a carriage return to provide a neater and cleaner error report.

The symbol table and list of vectors, located from $CC03 to $CC26, were revised considerably. The number of symbols was increased from 11 to 12, making the start of the symbol table $CC1B rather than $CC19 as given in the DOS 5.1 memory map of Table 1. Including the new & and = commands, seven distinct symbols have been used. For various reasons, > and ., both synonyms for @, were added, thus using two more symbols. Since "Supermon64" (COMPUTE!, January 1983) was used extensively in developing Enhanced DOS Support, a symbol (!) was added that branches to Supermon64 (or to any other monitor that is entered via the break vector). Of course, the monitor must have previously been loaded. With ten symbols now assigned, two remain available for future use.

Command Summary

Table 2 lists the commands currently available in Enhanced DOS Support. Note that the . symbol is indicated for most disk command functions, while > is shown for device code changes and @ for reading the disk error status. This is strictly a matter of taste since all three (@, >, and .) are synonyms that perform exactly the same functions. Program 2 is a BASIC boot program that writes a command summary to the screen as well as LOADing and activating Enhanced DOS Support (assuming you saved it with the program name used in line 10). Use Program 2 as you used "C-64 Wedge" to activate the original "DOS 5.1."

Entering The DOS 5.1 Enhancements

Program 1 will make all the necessary modifications to "DOS 5.1" to create the Enhanced DOS Support program. Before you run Program 1, make sure you have a disk containing the "DOS 5.1" program in the drive; otherwise, you will get the message FILE NOT FOUND ERROR IN 100. If any errors are detected in the DATA statements, the program will stop and report which lines contain errors. When Program 1 has finished, the cursor should be resting on a line of POKE statements. At this point you should insert the disk on which you want to store Enhanced DOS Support. Press RETURN, and the cursor should move to the line with the SAVE statement. Press RETURN again, and Enhanced DOS Support will be stored on the disk with the filename DOS 5.1E. At this point you will need to turn the computer off and back on (or reset with SYS 64738) to return the system to its normal configuration.

If you want to take the easy way out, send a diskette containing "DOS 5.1" with $3 and a stamped, self-addressed mailer, and I will add the revisions for you.

Stephen S. Melsheimer
Clemson University
Department of Chemical Engineering
Clemson, SC 29631

To use Enhanced DOS Support, load it from disk by running Program 2. The original "C-64 Wedge" supplied with "DOS 5.1" can also be

FOR COMMODORE-64 HACKERS ONLY!

The ultimate source for Commodore-64 Computer information

OTHER BOOKS AVAILABLE SOON

THE ANATOMY OF THE C-64

is the insider's guide to the lesser known features of the Commodore 64. Includes chapters on graphics, sound synthesis, input/output control, sample programs using the kernal routines, more. For those who need to know, it includes the complete disassembled and documented ROM listings.

ISBN-0-916439-00-3 300pp $19.95

THE ANATOMY OF THE 1541 DISK DRIVE

unravels the mysteries of using the misunderstood disk drive. Details the use of program, sequential, relative and direct access files. Include many sample programs FILE PROTECT, DIRECTORY, DISK MONITOR, BACKUP MERGE, COPY, others. Describes internals of DOS with completely disaddembled and commented listings of the 1541 ROMS.

ISBN-0-916439-01-1 320pp $19.95

MACHINE LANGUAGE FOR C-64

is aimed at those who want to progress beyond BASIC. Write faster, more memory efficient programs in machine language. Test is specifically geared to Commodore 64. Learns all 6510 instructions. Includes listings for 3 full length programs: ASSEMBLER, DISASSEMBLER and amazing 6510 SIMULATOR so you can "see" the operation of the '64.

ISBN-0-916439-02-X 200pp $14.95

TRICKS & TIPS FOR THE C-64

is a collection of easy-to-use programming techniques for the '64. A perfect companion for those who have run up against those hard to solve programming problems. Covers advanced graphics, easy data input, BASIC enhancements, CP/M cartridge on the '64, POKEs, user defined character sets, joystick/mouse simulation, transferring data between comuters, more. A treasure chest.

ISBN-0-916439-03-8 250pp $19.95

GRAPHICS BOOK FOR THE C-64

takes you from the fundamentals of graphic to advanced topics such as computer aided design. Shows you how to program new character sets, move sprites, draw in HIRES and MULTICOLOR, use a lightpen, handle IRQs, do 3D graphics, projections, curves and animation. Includes dozens of samples.

ISBN-0-916439-05-4 280pp $19.95

ADVANCED MACHINE LANGUAGE FOR THE C-64

gives you an intensive treatment of the powerful '64 features. Author Lothar Englisch delves into areas such as interrupts, the video controller, the timer, the real time clock, parallel and serial I/O, extending BASIC and tips and tricks from machine language, more.

ISBN-0-916439-06-2 200pp $14.95

IDEAS FOR USE ON YOUR C-64

is for those who wonder what you can do with your '64. It is written for the novice and presents dozens of program listing the many, many uses for your computer. Themes include: auto expenses, electronic calculator, recipe file, stock lists, construction cost estimator, personal health record diet planner, store window advertising, computer poetry, party invitations and more.

ISBN-0-916439-07-0 200pp $12.95

PRINTER BOOK FOR THE C-64

finally simplifies your understanding of the 1525, MPS/801, 1520, 1526 and Epson compatible printers. Packed with examples and utility programs, you'll learn how to make hardcopy of text and graphics, use secondary addresses, plot in 3-D, and much more. Includes commented listing of MPS 801 ROMs.

ISBN-0-916439-08-9 350pp. $19.95

SCIENCE/ENGINEERING ON THE C-64

is an introduction to the world of computers in science. Describes variable types, computational accuracy, various sort alogrithms. Topics include linear and nonlinear regression, CHI-square distribution, Fourier analysis, matrix calculations, more. Programs from chemistry, physics, biology, astronomy and electronics. Includes many program listings.

ISBN-0-916439-09-7 250pp $19.95

CASSETTE BOOK FOR THE C-64

(or Vic 20) contains all the information you need to know about using and programming the Commodore Datasette. Includes many example programs. Also contains a new operating system for fast loading, saving and finding of files.

ISBN-0-916439-04-6 180pp. $12.95

DEALER INQUIRIES ARE INVITED

IN CANADA CONTACT:
The Book Centre, 1140 Beaulac Street
Montreal, Quebec H4R1R8 Phone: (514) 322-4154

AVAILABLE AT COMPUTER STORES, OR WRITE:

Abacus Software

P.O. BOX 7211 GRAND RAPIDS, MI 49510

Exclusive U.S. DATA BECKER Publishers

For postage & handling, add $4.00 (U.S. and Canada), add $6.00 for foreign. Make payment in U.S. dollars by check, money order of charge card. (Michigan Residents add 4% sales tax.)

VISA MasterCard

FOR QUICK SERVICE PHONE (616) 241-5510

Commodore 64 is a reg. T.M. of Commodore Business Machines

Table 1: "DOS 5.1" Memory Map

$CC00	Wedge activation entry (52224)
CC03	Symbol vectors
CC19	Symbol table
CC27	Text buffer
CC77	Device number
CC78	Filename suffix
CC7A	Current symbol
CC7B	"DOS MANAGER 5.1 ..." text
CCE1	Wedge activation routine
CCF3	Normal entry point
CCF8	CHRGET call address validation ($A7E6 and $A48C are allowed)
CD0E	Check character against symbol table
CD30	Branch to execute routine if symbol matched
CD48	Execute @ commands
CD71	Send command string to disk (for example, @S0:filename)
CD90	Read disk error status (@)
CDB2	List directory (@$)
CE22	Execute LOAD (including /, %, and ↑)
CE6C	Disable wedge (@Q)
CE79	Execute SAVE (←)
CE7E	Set device code (@#N)
CEA3	Process line into text buffer
CF4B	Print DOS MANAGER message
CF5B	End of "DOS 5.1"

Table 2: Enhanced DOS Support Commands

.$	Directory Listing
/ *filename*	Load
↑ *filename*	Load and run
& *filename*	Append
← *filename*	Save
← @0:*filename*	Save and replace
= *filename*	Verify
% *filename*	Absolute Load
.S0:*filename*	Scratch
.R0:*newname*=0:*oldname*	Rename
.C0:*newfile*=0:*oldfile* ,*old2,...old4*	Copy/Concatenate up to four files
.N0:*diskname,id*	Format disk
.I	Initialize disk
.V	Validate disk
@	Read disk error status
!	Break/Activate monitor
>*n*	Change disk device number to *n*
.Q	Disable DOS Support
SYS 52224	Reactivate DOS Support

(Note: ., >, and @ can be used interchangeably in any command beginning with one of these symbols.)

used, if the filename in line 10 is changed from DOS 5.1E to DOS 5.1.

One caution: Commodore may in the future issue revisions of the DOS wedge. The enhancements given here may not work with versions different from the one on which it is based. If you get the message DOS MANAGER V5.1/071382 when you activate the original DOS 5.1, you have the correct version. If Program 1 checks out okay, but your program will not work, that may be the problem. The "easy out" mentioned above is still available in that case, of course.

Further Extensions

Two symbols remain unassigned in Enhanced DOS Support, and two of the three redundant command codes (@, >, and .) could be reassigned for other uses if needed. Thus, additional functions can easily be added to the program. One possibility is a command to link other utility programs (for example, a programmer's aid package) in a manner similar to the ! monitor link. Another handy addition might be a help routine that could display a summary of the Enhanced DOS Support commands on the screen without disrupting the program in memory. This could also include monitor or even BASIC commands as well.

To add a command, put the ASCII value of the desired symbol character in location $CC26. Then, put the high byte of the starting address of the new routine in $CC0E, and put the low byte of the target address *less one* in $CC1A. For a second command, decrement each of these addresses by one. The only other thing you must do is the hard part—writing the routine that will accomplish the new function. Locations $CF7F–$CFB1 are unused in DOS 5.1E, but extensive routines would have to be located outside the $CC00–$CFFF block. Remember that if a routine is to be used in program mode, the symbol must not be tokenized by BASIC (for example, do not use * as a symbol).

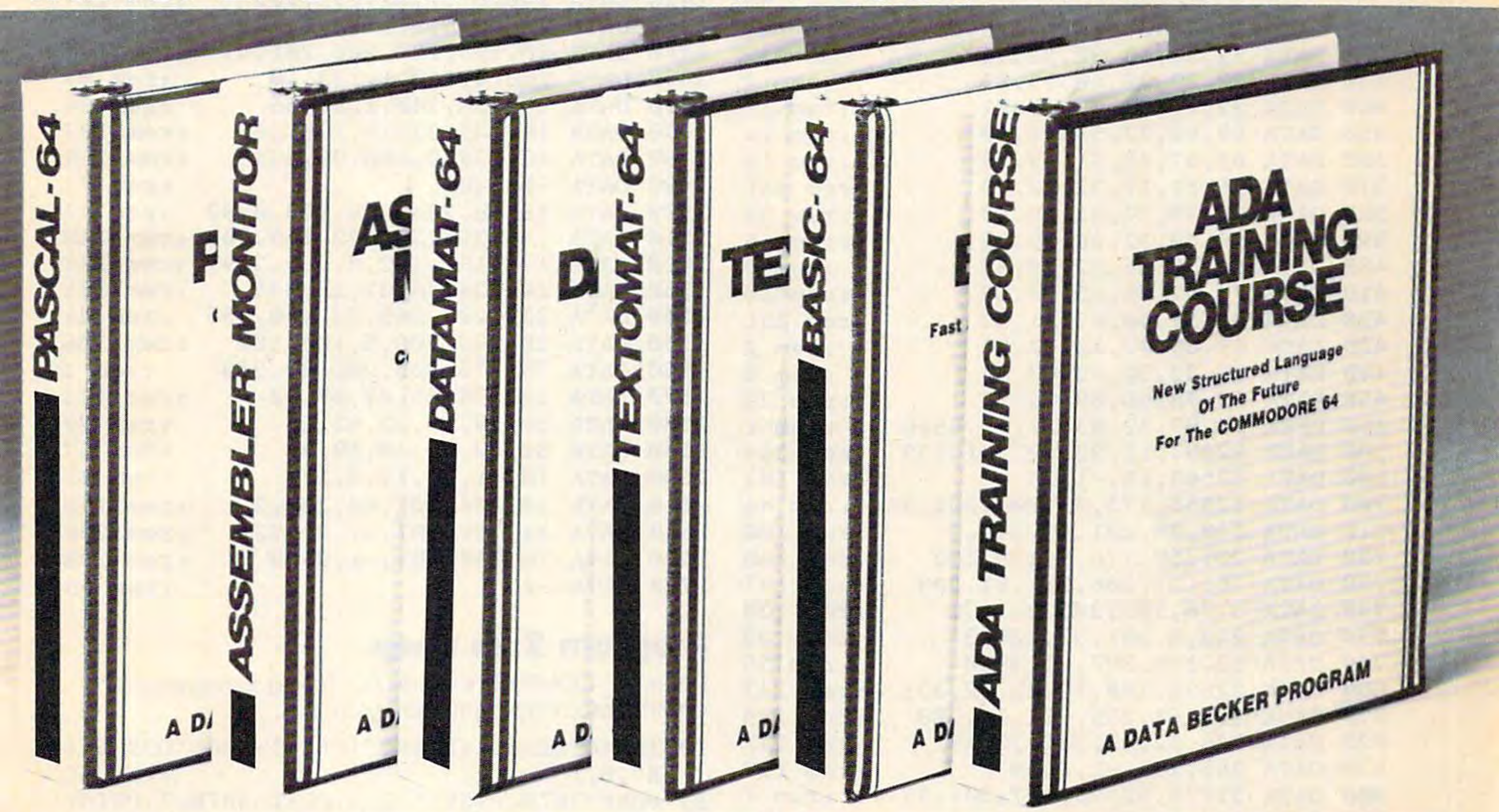
Program 1: Enhanced DOS Support

Refer to "COMPUTE!'s Guide For Typing In Programs" article before typing this program in.

```
100 IF A=0 THEN A=1:LOAD "DOS 5.1",8,1
                                   :rem 133
110 READ AD:IF AD=-2 THEN 180      :rem 229
120 CK=0:FL=PEEK(64)*256+PEEK(63) :rem 72
130 READ DT:IF DT=-1 THEN 150       :rem 9
140 POKEAD,DT:CK=CK+DT:AD=AD+1:GOTO 130
                                    :rem 71
150 LL=PEEK(64)*256+PEEK(63)        :rem 28
160 READ CS:IF CS<>CK THEN PRINT"ERROR IN
     DATA: LINES";FL;"-";LL:STOP :rem 108
170 GOTO 110                        :rem 99
180 PRINT "{CLR}ENHANCEMENTS ADDED":PRINT
    "{2 DOWN}PO43,0:PO44,204:PO45,0:PO46,
    208"                            :rem 1
190 PRINT "{2 DOWN}SAVE"+CHR$(34)+"DOS 5.
    1E"+CHR$(34)+",8":PRINT"{HOME}":END
                                   :rem 180
200 DATA 52232,207,205,205,205,204
                                   :rem 122
210 DATA 205,205,33,33,33,33        :rem 79
```

SERIOUS 64 SOFTWARE

INDISPENSIBLE TOOLS FOR YOUR COMMODORE 64



PASCAL-64

This full compiler produces fast 6502 machine code. Supports major data Types: REAL, INTEGER, BOOLEAN, CHAR, multiple dimension arrays, RECORD, FILE, SET and pointer. Offers easy string handling, procedures for sequential and relative data management and ability to write INTERRUPT routines in Pascal! Extensions included for high resolution and sprite graphics. Link to ASSEM/MON machine language. DISK $39.95

DATAMAT-64

This powerful data base manager handles up to 2000 records per disk. You select the screen format using up to 50 fields per record. DATAMAT 64 can sort on multiple fields in any combination. Complete report writing capabilities to all COMMODORE or ASCII printers. DISK $39.93
Available November

TEXTOMAT-64

This complete word processor displays 80 columns using horizontal scrolling. In memory editing up to 24,000 characterrs plus chaining of longer documents. Complete text formatting, block operations, form letters, on-screen prompting.
Available November DISK $39.95

ASSEMBLER / MONITOR-64

This complete language development package features a macro assembler and extended monitor. The macro assembler offers freeform input, complete assembler listings with symbol table (label), conditional assembly.

The extended monitor has all the standard commands plus single step, quick trace breakpoint, bank switching and more. DISK $39.95

BASIC-64

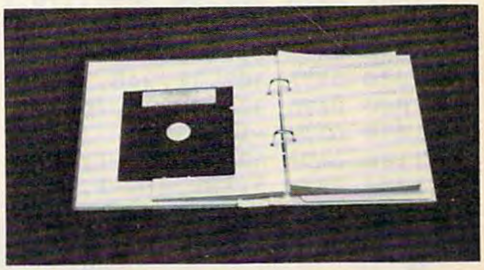
This is a full compiler that won't break your budget. Is compatible with Commodore 64 BASIC. Compiles to fast machine code. Protect your valuable source code by compiling with BASIC 64.
Available December DISK $39.95

ADA TRAINING COURSE

This package is an introduction to ADA, the official language ot the Department of Defense and the programming language of the future. Includes editor, syntax checker/compiler and 110 page step by step manual describint the language.
Available November DISK $79.95

OTHER NEW SOFTWARE COMING SOON!

All software products featured above have inside disk storage pockets, and heavy 3-ring-binder for maximum durability and easy reference.



DEALER INQUIRIES INVITED

AVAILABLE AT COMPUTER STORES, OR WRITE:

Abacus Software
P.O. BOX 7211 GRAND RAPIDS, MI 49510
Exclusive U.S. DATA BECKER Publishers

For postage & handling, add $4.00 (U.S. and Canada), add $6.00 for foreign. Make payment in U.S. dollars by check, money order of charge card. (Michigan Residents add 4% sales tax.)

VISA MasterCard

FOR QUICK SERVICE PHONE (616) 241-5510

```
220 DATA 33,239,71,71,71,220      :rem 90
230 DATA 71,71,37,47,38,94         :rem 8
240 DATA 61,95,64,46,62,33         :rem 3
250 DATA 64,64,83,-1,3203        :rem 198
300 DATA 52334,170,170,170,170,170
                                 :rem 130
310 DATA 170,170,170,170,8,56    :rem 141
320 DATA 52,33,13,32,32,32       :rem 235
330 DATA 32,32,32,68,79,83         :rem 4
340 DATA 32,77,65,78,65,71        :rem 13
350 DATA 69,82,32,53,46,49        :rem 11
360 DATA 69,47,48,51,49,53        :rem 16
370 DATA 56,51,13,32,32,70       :rem 246
380 DATA 82,79,77,32,86,53        :rem 20
390 DATA 46,49,32,66,89,32        :rem 16
400 DATA 66,79,66,32,70,65         :rem 9
410 DATA 73,82,66,65,73,82        :rem 10
420 DATA 78,32,40,67,41,32       :rem 251
430 DATA 67,66,77,13,32,32         :rem 2
440 DATA 32,32,32,69,88,84         :rem 8
450 DATA 69,78,68,69,68,32        :rem 29
460 DATA 66,89,32,83,32,-1,6546    :rem 1
500 DATA 52497,11,221,27,-1,259  :rem 244
600 DATA 52548,15,-1,15          :rem 102
700 DATA 52555,173,39,204,201,36  :rem 44
710 DATA 240,96,201,48,144,9     :rem 100
720 DATA 201,58,176,5,198,183    :rem 160
730 DATA 76,127,206,201,81,208   :rem 197
740 DATA 3,76,108,206,201,78     :rem 100
750 DATA 240,4,201,83,208,3       :rem 39
760 DATA 32,189,207,-1,4950       :rem 55
800 DATA 52636,169,13,32,22,231  :rem 247
810 DATA 32,165,255,201,13,208   :rem 189
820 DATA 246,32,22,231,32,171    :rem 137
830 DATA 255,234,-1,2564         :rem 157
900 DATA 52774,32,95,207,201,37    :rem 1
910 DATA 208,3,169,1,44,169       :rem 54
920 DATA 0,133,185,165,10,32      :rem 88
930 DATA 213,255,144,3,76,249    :rem 156
940 DATA 224,165,10,240,3,76      :rem 94
950 DATA 126,225,32,183,255,41   :rem 200
960 DATA 191,240,3,76,156,225    :rem 155
970 DATA 173,122,204,201,37,208  :rem 243
980 DATA 1,96,134,45,132,46       :rem 56
990 DATA 32,94,166,173,122,204   :rem 205
1000 DATA 201,94,208,3,76,178    :rem 145
1010 DATA 207,76,161,225,234,162  :rem 33
1020 DATA 2,189,171,227,149,124  :rem 246
1030 DATA 202,16,248,76,121,0    :rem 132
1040 DATA 32,89,225,76,144,164   :rem 203
1050 DATA -1,11372               :rem 104
1100 DATA 52874,48,-1,48         :rem 160
1200 DATA 52882,48,-1,48         :rem 160
1300 DATA 52945,234,234,234,234,234
                                 :rem 192
1310 DATA 234,-1,1404            :rem 247
1400 DATA 53067,165,123,201,2,208 :rem 81
1410 DATA 13,162,0,189,123,204   :rem 183
1420 DATA 240,6,32,22,231,232    :rem 127
1430 DATA 208,245,96,169,0,133   :rem 201
1440 DATA 10,173,122,204,201,61  :rem 224
1450 DATA 208,4,169,1,133,10      :rem 87
1460 DATA 201,38,240,1,96,56      :rem 96
1470 DATA 165,45,233,2,170,165   :rem 197
1480 DATA 46,233,0,168,96,234    :rem 156
1490 DATA -1,6489                 :rem 77
1500 DATA 53168,234,234,169,0,32  :rem 42
1510 DATA 144,255,32,142,166,76  :rem 248
1520 DATA 174,167,162,0,189,219  :rem 255
1530 DATA 207,240,6,32,22,231    :rem 131
1540 DATA 232,208,245,32,228,255  :rem 41
1550 DATA 201,78,208,5,104,104   :rem 188
1560 DATA 76,175,205,201,89,208    :rem 1
1570 DATA 240,96,65,82,69,32     :rem 113
1580 DATA 89,79,85,32,83,85       :rem 79
1590 DATA 82,69,32,40,89,47       :rem 67
1600 DATA 78,41,63,13,0,173       :rem 41
1610 DATA 39,204,201,64,208,3    :rem 138
1620 DATA 32,189,207,32,89,225   :rem 206
1630 DATA 76,144,205,-1,9669     :rem 109
1700 DATA -2                      :rem 65
```


Program 2: 64 Wedge

Refer to "COMPUTE!'s Guide For Typing In Programs" article before typing this program in.

```
10 IFA=0THENA=1:PRINT"{CLR}":LOAD"DOS 5.1
   E",8,1                         :rem 56
15 POKE53070,8:SYS52224:POKE53070,2:PRINT
   "{DOWN}{10 SPACES}{RVS} COMMAND SUMMAR
   Y "                            :rem 81
20 PRINT".$ DIRECTORY{8 SPACES}@
   {2 SPACES}ERROR STATUS        :rem 198
25 PRINT".V VALIDATE{8 SPACES}>N
   {2 SPACES}DEVICE# = N         :rem 206
30 PRINT".I INITIALIZE{7 SPACES}!
   {2 SPACES}MONITOR/BREAK        :rem 88
35 PRINT".Q{2 SPACES}DISABLE DOS SUPPORT
                                 :rem 173
40 PRINT"SYS 52224{2 SPACES}REACTIVATE DO
   S SUPPORT                      :rem 28
45 PRINT"/FILENAME{14 SPACES}LOAD:rem 168
50 PRINT"↑FILENAME{14 SPACES}LOAD & RUN
                                 :rem 238
55 PRINT"&FILENAME{14 SPACES}APPEND
                                  :rem 56
60 PRINT"←FILENAME{14 SPACES}SAVE:rem 228
65 PRINT"←@0:FILENAME{11 SPACES}SAVE & RE
   PLACE                         :rem 181
70 PRINT"=FILENAME{14 SPACES}VERIFY
                                 :rem 105
75 PRINT"%FILENAME{14 SPACES}LOAD ABSOLUT
   E                               :rem 0
80 PRINT".S0:FILENAME{11 SPACES}SCRATCH
                                  :rem 75
85 PRINT".R0:NEWNAME=0:OLDNAME{2 SPACES}R
   ENAME                         :rem 112
90 PRINT".C0:NEWFILE=0:OLDFILE{2 SPACES}C
   OPY/MERGE 1-4                  :rem 15
92 PRINT"{2 SPACES},0:OLD2,---0:OLD4
   {6 SPACES}FILES               :rem 100
95 PRINT".N0:DISKNAME,ID{8 SPACES}FORMAT
   {SPACE}NEW DISK               :rem 230
100 NEW                          :rem 123
```
©

IBM Screen Formatter

David Leithauser

Here's a simple programming trick that will help you write programs to be compatible with either 40- or 80-column text screens. It works with all versions of IBM BASIC.

IBM Personal Computers have the option of using a 40-column or 80-column display in text mode. This is done primarily because you have a choice of various types of monitors (screens), ranging from special RGB (Red-Green-Blue) computer displays to ordinary TV sets. The 40-column option is necessary because on most TV sets the letters of an 80-column display are too small and fuzzy to be seen clearly. The resolution on TV screens is not as good as on special video monitors. For people who have invested in a dedicated computer monitor, however, the 80-column display allows twice as much information to be displayed on the screen.

While these options make IBM computers more versatile for users, they cause some problems for programmers. If a program is written in the 80-column mode, some of the words that are printed on the screen may be split when the program is run in the 40-column mode. For example, if your program displays the message PRESS M FOR MENU, R TO REPEAT COMPUTATIONS, the word COMPUTATIONS will be split between the O and the N when the program is run in the 40-column mode. One way to avoid this problem is to insert enough spaces before the word that would be split so it starts on the next line in 40-column mode. Unfortunately, this sometimes looks very strange in the 80-column mode. When printing PRESS M FOR MENU, R TO REPEAT COMPUTATION, for example, you would have to insert ten spaces before the word COMPUTATION, which would look odd on an 80-column screen.

A Better Solution

A more effective way to solve this problem is to take advantage of a feature built into the PRINT statement in IBM BASIC. When the PRINT statement is printing a series of strings separated by semicolons, it checks to see if each string will fit in the space remaining in the current screen line. If it won't, the string automatically starts on the next line.

For example, if A$ and B$ are both 30 characters long, the statement:

```
PRINT A$;B$
```

prints A$ and B$ on the same line if the screen is in the 80-column mode, but on different lines on a 40-column screen. B$ starts printing on the next line because it won't fit completely on the same line with A$.

Therefore, by splitting the text in your PRINT statement after a space within the first 40 characters, you can be sure the words will not be broken. Just count the characters until you get to the fortieth, then backtrack until you get to a space and split the text after the space into two sections separated by a semicolon. For example, the statement:

```
PRINT "PRESS M FOR MENU, R TO REPEAT
COMPUTATION."
```

becomes

```
PRINT "PRESS M FOR MENU, R TO REPEAT
";"COMPUTATION."
```

You may need to split the text in the PRINT statement in several places if the second portion of the string is more than 40 characters long.

In some cases, you may be printing out a string variable (such as A$) rather than a string literal (characters enclosed in quotes). Sometimes you may not even know the length of the string, such as when the string was input by the user. In these instances, the following subroutine will print the contents of the string variable (in this case A$) without splitting any words—regardless of the screen width or the length of the string (provided there's at least one space per 40 characters). The line numbers in this subroutine are arbitrary, so use whatever line numbers you like (omit lines 10–30 when using this as a subroutine). Just assign the text you want printed to A$ and GOSUB 65000. (Be sure to put an END statement after your main code so the subroutine isn't accidentally executed twice.)

IBM Screen Formatter

Refer to "COMPUTE!'s Guide To Typing In Programs" before typing in this program.

```
HO 10 CLS:A$="THE IBM PERSONAL COMPUTE
      R HAS THE OPTION OF USING A 40-C
      OLUMN OR 80-COLUMN DISPLAY"
MB 20 GOSUB 65000
CF 30 END
LI 65000 WS=1
NJ 65010 WE=INSTR(WS,A$," ")
KA 65020 IF WE>0 THEN PRINT MID$(A$,WS
      ,WE-WS+1);:WS=WE+1:GOTO 65010
JC 65030 PRINT MID$(A$,WS)
JK 65040 RETURN
```

©

Apple Disk Verify

Ilan Reuben

Here's a short but useful verification utility for checking BASIC programs saved on disk. It works with all Apple II-family computers.

The VERIFY command in Apple DOS 3.3 indicates only whether a saved program is legible. Sometimes this isn't enough. If you need to be absolutely sure that the program you just saved is safely stored on the disk, "Verify+" is the answer. It's a utility written in machine language which insures that the BASIC program saved on disk is exactly the same as the program in memory. Verify+ is only about 400 bytes long and uses two 256-byte buffers.

The program following this article is a BASIC loader which creates Verify+ by encoding the machine language (ML) in DATA statements. Type in the program and run it. If any errors are detected in the data, the program stops. When you see the message ML LOADED, save the ML by typing:

BSAVE VERIFY+,A$8E00,L$175

Now that you have saved the machine language for Verify+, you can reload it whenever it's needed by typing:

BLOAD VERIFY+

To run it, you can type this command:

CALL 36352

If you plan to run Verify+ frequently, there's an even easier way. First, load the utility by typing BLOAD VERIFY+ as before. Then enter these two statements:

POKE 1014,0:POKE 1015,142

Then all you'll have to type to perform the verification is the ampersand symbol, &.

A Simple Test

After typing in Verify+ and saving it as described above, try this easy test. First, create a one-line BASIC program:

10 PRINT "HELLO"

and save it on disk with the filename TRIAL. Then, run Verify+ by typing:

CALL 36352 "TRIAL"

(or, if you entered the appropriate POKEs, run Verifyby typing & "TRIAL"). If the program was properly verified, you'll see the message FILE OK.

Now, slightly modify the BASIC test program:

10 PRINT "JELLO"

and try verifying it again: CALL 36352 "TRIAL" or & "TRIAL". You should get the message VERIFY ERROR.

Unfortunately, Verify+ works only with BASIC programs. This is because of the different ways that BASIC, binary, and text files are stored on the disk. But if Verify+ ever keeps you from losing even one important BASIC program because of a faulty disk or other problem, you'll be glad you kept it handy.

Apple Disk Verify

```
10   FOR ADRS = 36352 TO 36724
20   READ BYTE:X = X + BYTE
30   POKE ADRS,BYTE
40   NEXT
50   IF X < > 47562 THEN  PRINT "ERR
     OR IN DATA STATEMENTS": END
100  DATA 160,31,169,160,153,43,143,
     136,16,250,200,32,177,0,166,184
     ,177,184,201,34
110  DATA 240,12,9,128,153,43,143,20
     0,232,192,31,144,239,96,232,134
     ,184,169,140,141
120  DATA 98,143,169,17,141,93,143,1
     69,0,141,94,143,32,79,143,173,1
     ,140,141,93
130  DATA 143,173,2,140,141,94,143,1
     3,93,143,208,3,76,210,142,32,79
     ,143,169,11
140  DATA 133,6,169,140,133,7,160,32
     ,185,40,143,209,6,208,7,136,192
     ,2,208,244
150  DATA 240,11,165,6,24,105,35,133
     ,6,208,231,240,198,160,0,177,6,
     141,1,140
160  DATA 200,177,6,141,2,140,200,16
     9,7,133,9,169,255,133,8,169,141
     ,133,7,169
170  DATA 0,133,6,169,140,141,98,143
     ,173,1,140,141,93,143,173,2,140
     ,141,94,143
180  DATA 13,93,143,240,49,32,79,143
     ,169,141,141,98,143,162,11,232,
     240,221,189,0
190  DATA 140,141,93,143,232,189,0,1
     40,141,94,143,13,93,143,240,18,
     32,79,143,177
200  DATA 6,209,8,208,13,200,208,247
     ,240,221,169,0,240,29,169,30,20
     8,25,24,152
210  DATA 101,8,133,8,144,2,230,9,56
     ,229,175,201,2,176,6,165,9,197,
     176,240
220  DATA 229,169,16,168,169,141,32,
     240,253,185,5,143,200,32,240,25
     3,201,141,208,245
230  DATA 96,135,198,201,204,197,160
     ,206,207,212,160,198,207,213,20
     6,196,141,135,214,197
240  DATA 210,201,198,217,160,197,21
     0,210,207,210,141,198,201,204,1
     97,160,207,203,141,197
250  DATA 160,207,203,141,203,141,20
     0,160,160,160,160,160,160,160,1
     60,160,160,160,160,160
260  DATA 160,160,160,160,160,160,16
     0,160,160,160,160,160,160,160,1
     60,169,143,160,89,32
270  DATA 217,3,96,0,0,1,96,1,0,17,1
     5,111,143,0,140,0,0,1,0,0
280  DATA 96,1,0,0,0,0,0,0,1,239,216
     ,0,254,255
```

©

Commodore Potpourri

COMPUTE! Readers

Compiled by Todd Heimarck, Assistant Editor

Who hasn't felt the thrill of finding a new programming technique that's a little faster, takes less memory, or somehow seems to be a more elegant way of doing things? Here are a few such techniques discovered by readers of COMPUTE! *and* COMPUTE!'s GAZETTE. *For Commodore 64, VIC-20, and PET.*

Random Access DATA Statements

Ian Adam

DATA statements are a handy way of feeding information to a program. You don't have to fool around with opening, reading, and closing tape or disk files. The information you need is right there in the program, waiting for a READ statement.

But there are two disadvantages to using DATA. First is that the program reads each item only once. After you use a piece of data, you can't go back and read it a few more times. Second is that, like tape files, DATA statements are sequential. You begin at the beginning and end at the end. Try to go past the last item and you get an ?OUT OF DATA error.

Of course, there is a way to solve the first problem. RESTORE resets the data pointer to the beginning of the list. RESTORE isn't too flexible, though. What we really need is a command to go back to a specific line number, like RESTORE to 1000. Some versions of BASIC, such as Atari BASIC, have this feature built-in.

Here's a way to do almost the same thing, with just a couple of POKEs:

```
POKE65,PEEK(61):POKE66,PEEK(62)
```

This resets the DATA pointer (at locations 65 and 66) to the current position of the program counter (at locations 61 and 62, also used for the CONTinue command). In other words, it tells the computer, "Please start reading right here."

If you want to play a specific tune encoded in the DATA lines, for example, you would make the POKEs the first line in a subroutine, followed by the DATA statements and the READ, POKE section. (Note to PET users: Depending on which ROM you have, the technique is the same, but the zero-page locations will be different—62, 63 and 58, 59 for BASIC 4.0 PET/CBM machines.)

Graphic REMarks

Daniel Shaffer

If you've ever tried to use capital letters or graphics symbols in a REMark, you know it can be frustrating. For example, when you enter:

```
10 rem Designed by Frank Dow
```

What you see after LISTing is:

```
10 rem str$esigned by ascrank str$ow
```

The computer interprets the shifted letters as BASIC tokens and prints them in full. Sometimes this quirk can be useful (SHIFT-L is interpreted as SYNTAX ERROR and stops people from LISTing past the line containing it), but usually it's annoying. A simple way to get around the problem is to enter quote mode. After the REM, type a quotation mark. The rest of your message will appear as you typed it, graphics and all.

Embedded Carriage Returns

Hla T. Thein

It's common knowledge that a carriage return is built into the PRINT statement unless you follow it with a semicolon or comma. To print three different things on three different lines, you would have to use three PRINTs:

```
10 PRINT"QUICK": PRINT"BROWN": PRINT"FOX"
```

But most people don't know there's a way of embedding carriage returns in a string. First type this line:

```
10 PRINT"QUICK BROWN FOX"
```

If you run the program, everything goes on the same line. Now list the program and cursor up to the space between QUICK and BROWN. Press RVS ON (CTRL-9) and type a SHIFT-M. Move the cursor right and do the same thing between BROWN and FOX.

The program now thinks there is a SHIFT-RETURN between the letters, and it will print the words on three different lines. Note that this trick also affects the way the program lists.

(Editor's Note: The two hints above can be combined for some interesting effects. Enter this line: 10 REM". Press RETURN, move the cursor up, turn on reverse, and right after the quotation mark put a REVERSE-SHIFT-M followed by a REVERSE-SHIFT-S. Now try to list line 10. The screen clears. The line is, in effect, unlistable. The SHIFT-M forces a carriage return, which turns off quote mode. The reverse heart then causes the screen to clear. Adding such a REM to every program line can help you hide the listing from snoopy users. You can also include cursor movements with REMarks.)

Machine Language Backup On Tape

Willem Schaaij

A broken or worn-out tape can be a disaster if you don't have a backup. Duplicating BASIC programs is easy enough, but machine language (ML) programs are a headache to copy.

If you have a machine language monitor, and know the starting and ending addresses of the ML program, it's easy to make backups. But what if you don't have a monitor, or don't know where the program begins or ends?

Looking through a memory map suggests an answer. The tape header contains the information we need. And after a LOAD, the header information is stored in the cassette buffer at locations 829–832. BASIC expects to find the program's beginning and ending addresses at locations 43–46. The solution:

1. LOAD "program name",1,1
2. Type NEW
3. POKE 43, PEEK(829)
   POKE 44, PEEK(830)
   POKE 45, PEEK(831)
   POKE 46, PEEK(832)
4. SAVE "program name" (using a new tape)

Because you've changed the pointers, you'll have to cold start the computer after the SAVE. Either turn it off and then on again, or use the SYS described below.

On PETs, the cassette buffer is in the same place (location 828), but the pointers to the beginning and end of the program may vary.

Saving Wear On The On/Off Switch

Shawn McDonald

A *cold start* (turning your computer off and then on) quickly clears memory and resets everything. But does it do any harm to the computer if you do it frequently?

The good news is, it doesn't do any significant harm to the circuits or chips, although it does cause minimal wear to the power switch. One way to do the same thing is to use this line: SYS 64802 (VIC), SYS 64738 (64), SYS 64790 (PET/CBM). After entering this line, you should see the usual opening message.

This can be a useful way to end a game—for example, if the user answers *no* to PLAY AGAIN (Y/N)?—or to reset the computer if you are working with custom characters or a high-resolution screen.

There are a few things which may not be reset. If you have POKEd 128 into 650, to make the keys repeat, you will find that the keys still repeat after the cold start.

Slightly different from the cold start SYS is a warm start SYS, which preserves the pointers to the beginning and end of memory.

If you have partitioned a section of memory to be used for custom characters or machine language, SYS 58232 (VIC) or SYS 58260 (64) will simulate a warm start. If you want to set the pointers before this SYS, POKE the beginning of BASIC (in low-byte, high-byte format) into 641 and 642, and the top of BASIC into 643 and 644.

One more tip: Certain televisions, when connected to a VIC, display a wavy, jumpy picture. I've discovered that POKE 36864,133 corrects the problem, although RUN/STOP– RESTORE causes the picture to start bouncing again. Zenith TVs seem to be most affected by the bouncing.

Unlistable Programs

Shawn K. Smith

On the VIC and 64, locations 774 and 775 contain a vector pointing to the LIST routine. If you change the values with a POKE, the program in memory becomes unlistable. What I do is POKE 774,255. It's a good idea to use a PEEK to learn what number *should* be in address 774 in case you want to reenable the LIST command.

If a program containing the POKEs is loaded, but not run, it can be listed, so this method can be circumvented. But in combination with other tamperproofing methods (*like REM SHIFT-L, described above*), you can keep most prying eyes out of your programs.

If you look at a good memory map, you can find some other useful vectors in the same area of memory. Locations 808 and 809 point to the STOP routine (called when unshifted RUN/STOP is pressed); put some new values there and you can disable the STOP key.

Defining A Joystick Function

Richard Mehalick

DEFine FuNction can be very useful in a program that frequently reads the joystick. For example, to read the joystick in port 1 of the Commodore 64, use DEF FNJO(Y) = 15 – (PEEK (56320) AND 15).

To make it even easier to use, combine it with the ON-GOTO command:

```
10 DEFFNJO(Y)=15-(PEEK(56320)AND15)
20 ON FNJO(Y) GOTO 50,60,20,70,30,30,30,80
30 GOTO20
50 PRINT"NORTH":GOTO20
60 PRINT"SOUTH":GOTO20
70 PRINT"WEST":GOTO20
80 PRINT"EAST":GOTO20
```

To include the fire button, define a separate function, or change every 15 in the function above to 31.

Since a defined function can include PEEKs, you can take this idea a step further and use it to check current screen position, watch for collisions, or read the jiffy clock. ©

Atari Easy Scroll

Eugene D. McMillin

These short, simple BASIC scrolling routines demonstrate a method for scrolling using the Atari computer's string variables. For beginners, there is a detailed explanation of how both programs work.

Sooner or later most BASIC programmers find that they would like to set up multiple screens and scroll through them. For the advanced programmer who understands the inner workings of the machine, this usually isn't too difficult. But for most BASIC programmers, struggling with such things as bytes per line, display lists, write and screen memory locations, pointers, and interrupts can be confusing.

Fortunately the Atari offers a simple way to scroll vertically. This can be done in BASIC without even one PEEK or POKE. The method involves using string variables. With the Atari we have the ability to dimension string variables to almost any size and then to access any portion of them we want. For instance, a string variable named NAME$, dimensioned to a size of 10, could contain two names of five characters each. If we want to view the first name, we PRINT NAME$(1,5) and all the letters between the first and fifth are displayed. Or, if we want to view the last name, we PRINT NAME$(6,10).

Scrolling An Entire Screen

Program 1 demonstrates how to use a string variable to simulate vertical scrolling of the entire screen. Here's how it works:

Line 10: Sets up a full GRAPHICS 1 screen. This technique will work with any graphics mode. However, in the higher resolution modes the memory requirements are unrealistic.

Line 20: Dimensions a string variable SC$ large enough to accommodate three full screens. A GRAPHICS 1 + 16 screen contains 24 lines with 20 characters on each line. However, it seems that some Ataris won't print to position 19,23 without getting a CURSOR OUT OF RANGE error. In order to get around this, the screen size is reduced to 23 lines. With this in mind, each screen requires 20 characters per line multiplied by 23 lines for a total of 460 characters per screen. As a result, the three screens require a string variable consisting of three screens multiplied by 460 characters per screen, or 1380 characters.

Line 30: Sets the first 460 characters of the string variable to C. In other words, the first screen will consist entirely of the letter C.

Line 40: Sets the second screen to the letter J.

Line 50: Sets the third screen to the letter W.

Line 60: POS is a variable which designates the first position of the character string SC$ that we will print to the screen. In this case it is the first character in the string.

Line 70: Sets a variable equal to the position of the joystick.

Line 80: If the joystick is forward, the screen will scroll down. As there are 20 characters in each line of GRAPHICS 1, the program subtracts 20 from the variable set up in line 60.

Line 90: The same as line 80 except the joystick is in the opposite direction. Therefore, we add 20 rather than subtract.

Line 100: Tests the variable POS to see if it's less than 1. If it is, resets it to 1 to avoid trying to print a portion of our string variable that is 0 or less, which would result in an error.

Line 110: We test the variable POS to see if it's greater than 921. If it's greater, reset it to avoid printing past the end of the dimensioned string.

Line 120: In order to print the entire screen, this line sets the cursor to the upper left-hand corner.

Line 130: This prints a full screen of 460 characters. The exact portion of the string variable SC$ printed depends on the value of POS.

Line 140: Back up and sample the joystick and try it all over again.

Type in Program 1, hook up your joystick, and see what happens. It isn't quite as smooth or fast as machine language scrolling, but it gets the job done.

Scrolling Part Of A Screen

Program 2 demonstrates this same scrolling technique over a small portion of the screen. In a spreadsheet program or game, you might want stationary text at the top or bottom of your screen while the rest scrolls. For example, when you're looking out the window of an airplane cockpit, the horizon would rise or fall as you dive or climb, but the instrument panel would stay stationary on the screen. Here are the significant changes to Program 1.

Line 20: Sets up three screens; however, each screen will only have ten lines. Twenty characters per line multiplied by 10 lines multiplied by 3 screens gives us a variable size of 600.

Lines 30–50: The size of the FOR-NEXT loops is reduced to take into account the reduced size of the screens.

Lines 51–52: These are new lines. They provide a stationary text for the screen. This stationary text is positioned above and below the portion of the screen that will scroll.

Line 110: This line is changed to account for the reduced size of the string variable SC$.

Line 120: The cursor is positioned part of the way down the screen. This is the top left position of the scrolling portion of the screen.

Line 130: Again the only change is to accommodate the reduced size of the screen.

Program 1: Vertical Scroll

```
HF 10 GRAPHICS 1+16
BD 20 DIM SC$(1380)
MH 30 FOR I=1 TO 460:SC$(I,I)="C":NE
      XT I
DK 40 FOR I=461 TO 920:SC$(I,I)="J":
      NEXT I
HK 50 FOR I=921 TO 1380:SC$(I,I)="W"
      :NEXT I
MG 60 POS=1
EK 70 ST=STICK(0)
BP 80 IF ST=14 THEN POS=POS-20
BN 90 IF ST=13 THEN POS=POS+20
AO 100 IF POS<1 THEN POS=1
OH 110 IF POS>921 THEN POS=921
JE 120 POSITION 0,0
JN 130 PRINT #6;SC$(POS,POS+459)
DF 140 GOTO 70
```

Program 2:
Vertical Scroll With Stationary Section

```
HF 10 GRAPHICS 1+16
NN 20 DIM SC$(600)
LP 30 FOR I=1 TO 200:SC$(I,I)="C":NE
      XT I
CL 40 FOR I=201 TO 400:SC$(I,I)="J":
      NEXT I
DN 50 FOR I=401 TO 600:SC$(I,I)="W":
      NEXT I
GO 51 POSITION 0,0:? #6;"SCROLL DEMO
      NSTRATION"
CK 52 POSITION 0,5:? #6;"SPLIT SCREE
      N"
MG 60 POS=1
EK 70 ST=STICK(0)
BP 80 IF ST=14 THEN POS=POS-20
BN 90 IF ST=13 THEN POS=POS+20
AO 100 IF POS<1 THEN POS=1
NJ 110 IF POS>401 THEN POS=401
JK 120 POSITION 0,6
JO 130 PRINT #6;SC$(POS,POS+199)
DF 140 GOTO 70
```

©

FOR-NEXT Loop Etiquette

Jim Butterfield, Associate Editor

If you've ever run into problems with FOR-NEXT loops, maybe it's because you weren't minding your programming manners. This explanation of FOR-NEXT rules should clear things up. Although written for Commodore BASIC, the advice applies to nearly all versions of BASIC.

The FOR-NEXT loop structure is one of the foundations of efficient BASIC programs. It seems to be surrounded by a mystique: Can you or can't you exit a loop before it completes its allotted repetitions?

There's a maxim gaining ground which says: "Never jump out of a FOR-NEXT loop, or sooner or later you'll get an OUT OF MEMORY error." Partly right, partly wrong. You *can* jump out of a FOR-NEXT loop, but you must understand the rules.

The Problem

Let's suppose you have a list of 1000 cities around the world. You're writing a program to give the distance between any two cities. The list of city names is in an array called C$, dimensioned to hold 1000 names.

The coding would start by asking the user to enter a city name. Then there would be a search through the table for a name match. The program would partly look like this:

```
INPUT "ENTER CITY NAME";N$
FOR J=1 TO 1000
IF C$(J)=N$ ...
...
NEXT J
```

If the user typed in PARIS, and it happened that PARIS was the second city in array C$, it would seem to be a waste if the program was forced to look at the remaining 998 table entries. On the other hand, if we're forbidden to jump out of the loop (to a statement following NEXT J), we seem to have no choice but to allow the extra 998 iterations.

What Are Our Options?

First, we can indeed exercise the loop over its entire range. The coding would look something like this:

```
K=0
FOR J=1 TO 1000
IF C$(J)=N$ THEN K=J
NEXT J
```

At this point, K will hold the city number; if the city is not found in the list, K will equal zero. It will work, but the loop will be slow; there will be a significant pause for each city, even if the name is found at the top of the list. It seems inefficient.

Second, we can force the loop variable outside its range on the assumption that this will cause the loop to terminate. The coding would look like this:

```
K=0
FOR J=1 TO 1000
IF C$(J)=N$ THEN K=J:J=1001
NEXT J
```

This works, but it seems to me to be dangerous. If the city list were expanded to 2000 items, it might be easy to overlook the change that would be needed to the J=1001 statement. We could fix that part by changing it to J=1E20, a high number we never expect to reach.

Changing the value of a loop variable is bad practice. Some languages (even some BASIC implementations) forbid this, and may even stop with an error such as LOOP VARIABLE CHANGED WITHIN LOOP. Here's the problem: The FOR-NEXT loop was designed to allow strict control over the number of repetitions made by the loop. Once we play around with the variable, we endanger the integrity of the loop. Doing this might create a situation where the loop will never end or will behave unpredictably.

Third, we can jump out of the loop when it has done the job we want: found the matching item, or whatever. The coding in this case might go:

```
K=Ø
FOR J=1 TO 1ØØØ
IF C$(J)=N$ GOTO 31Ø
NEXT J
PRINT "NOT FOUND":GOTO ...
31Ø continue....
```

It seems natural, and in many languages it's heartily encouraged. Structured purists might look down their noses at the GOTO statement that gets you out of the loop, but it would be a syntax complaint rather than an objection to leaving the loop early. Very structured language might offer an EXIT command to escape the loop.

Departing from an incompleted loop has developed a bad reputation. The rumor has gotten around that if you do this, the loop will never go away and eventually you'll hang up on an OUT OF MEMORY error. Not true. There is, however, a slight chance that naive coding might produce a baffling NEXT WITHOUT FOR halt; in this case, a little understanding or application of good programming habits will eliminate the danger.

Some Theory

We don't want unclosed loops to hang around forever and clutter up our computer. There are four ways that a FOR-NEXT loop can be retired from service—apart from obvious extreme measures such as turning the computer off or typing NEW.

1. When the loop goes through its complete range, it will be scratched from the active loop list.

2. If a loop is within another loop, the inner loop will be scratched whenever the outer loop reaches a NEXT statement. Note that this doesn't mean the outer loop must complete its range; if it goes back for another repetition, that too will cancel the inner loop.

3. If a loop is opened within a subroutine, RETURN from that subroutine will automatically scratch the loop.

4. If a FOR statement is encountered, any existing loop using the same variable name will be scratched, together with any other loops nested within.

```
1ØØ FOR J=1 TO 5Ø STEP 3
11Ø T=T+J
12Ø NEXT J
```

After these lines are executed, loop J will no longer be active. It has completed its range.

```
1ØØ FOR I=1 TO 1
11Ø FOR J=1 TO 5Ø STEP 3
12Ø T=T+J
13Ø IF T>1ØØ GOTO 16Ø
14Ø NEXT J
15Ø STOP
16Ø NEXT I
```

After these lines are run, loops I and J will no longer be active. Why not? I has gone through its entire range. J has not; but the J loop was scratched from the active list the moment NEXT I was encountered. At first glance, the I loop seems to have no purpose, since there is no repetition of the lines between FOR and NEXT; but it does serve to clean away the J loop.

```
1ØØ GOSUB 2ØØ
11Ø END
2ØØ FOR J=1 TO 5Ø STEP 3
21Ø T=T+J
22Ø IF T>1ØØ THEN RETURN
23Ø NEXT J
24Ø STOP
```

After these lines are executed (reaching line 110), loop J will no longer be active. Why not? Because it was opened in subroutine 200, and the RETURN at line 220 canceled its status. Program style experts might criticize the subroutine at line 200 because RETURN is not at the end; put it there if you like.

```
1ØØ FOR J=1 TO 5Ø STEP 3
11Ø T=T+J
12Ø IF T>1ØØ GOTO 15Ø
13Ø NEXT J
14Ø STOP
15Ø K=J
16Ø FOR J=1 TO 1:NEXT J
```

When these lines are done, loop J will be inactive, even though the FOR-NEXT at 100–130 was not completed over its range. The opening of a new J loop in line 160 cancels the previous J loop.

Self-Repair

In most cases, rule 4 saves most programs from encountering loop problems. Opening a new loop cancels the old one even when we jump out of it. All we need to do is use the same variable name. Often, programs go back and repeat an early section; and the same loops are opened again, with old loops scrapped as the new ones come into force. We hardly need think about the question.

We can make this almost rigorous if we apply a simple rule: Give all your major loops the same loop variable name, and inner loops similar consistent variables. Any big loop will then automatically cancel the previous big loop, and so on.

But if we indulge in "barefoot" coding and pick variables according to the way that letters of the alphabet pop into our heads, we can run into trouble on rare occasions.

Horrible Example

Here's a horrible sample program. It doesn't do anything useful, but illustrates the puzzling problem that can occur when we let loops run free.

```
100 INPUT "YOUR AGE";A
110 FOR J=1 TO 99
120   A=A-J
130   IF A<0 GOTO 200
140 NEXT J
150 STOP
200 R=J:T=0
210 FOR M=1 TO R
220   T=T+M
230   FOR J=1 TO T
240     V=V+J
250   NEXT J
260 NEXT M
270 PRINT "I WISH YOU";V;"JOYS"
```

Here's the puzzling thing. This program stops with a NEXT WITHOUT FOR error in line 260. It's baffling to the programmer: The NEXT M is clearly matched with the FOR M in line 210. How dare the computer say they don't match?

Let's carefully trace what happens here, and how an open loop gets us into peculiar trouble.

The FOR-NEXT loop in lines 110 to 140 is not completed; line 130 exits directly to line 200. There's still a live J loop when line 200 is reached.

Line 210 opens a new FOR loop using variable M. Since the J loop is still active—the new one doesn't cancel it—we now have two loops. The outer loop uses variable J and the inner loop uses variable M.

Line 230 wants to open another FOR loop, this time using variable J. But wait a minute; we have an active loop still in existence that uses J. Fine. Cancel the old J loop; that's what rule 4 says. And since the M loop is inside the J loop, it gets canceled too. What do we have now? A brand-new J loop and nothing else. The old J and the M are scrapped.

Line 250 finds the NEXT J statement quite acceptable. There's a J loop active, and it will be exercised however many times the values call for. When the loop completes going through its range of values, it is retired from duty. Now there are no active loops, and we may proceed to line 260.

Line 260 says NEXT M, but the computer doesn't have an active M loop anymore. It was canceled back at line 230, remember? So the computer stops and reports NEXT WITHOUT FOR, causing the programmer to tear out his or her hair.

Fixing It

How do we fix it? Let me count the ways:

1. We fix it by rule 1. We change the FOR-NEXT loop at 110 to 140 to exercise its entire range. Line 130 would change to something like: IF A<0 THEN RJ; we'd eliminate the STOP in line 150 and change line 200 to just T0. Slower, but OK.

2. We fix it by rule 2. We insert the two lines:

```
105 FOR I=1 TO 1
205 NEXT I
```

The extra loop does nothing but cancel the J loop, but that makes everything OK.

3. We fix it by rule 3. The coding from line 110 to 150 is changed to a subroutine. RETURN cancels the open J loop.

4. We fix it by rule 4. We could insert a new line 205 that said FOR J=1 TO 1:NEXT J; this would certainly cancel the active J loop. It might be better, however, to use our variable hierarchy rule, making J the outer loop through the entire program. Lines 210 to 260 become:

```
210 FOR J=1 TO R
220   T=T+J
230   FOR M=1 TO T
240     V=V+M
250   NEXT M
260 NEXT J
```

Now the FOR J loop at line 210 immediately cancels the open J loop from earlier lines.

But perhaps it's not so much a problem of fixing a program gone wrong. If we develop good programming habits, using systematic variable names, there will never be anything to fix.

Conclusion: You can jump out of FOR-NEXT loops and still be considered a good person. It's sound programming. But you'd be well-advised to understand a little more about how these loops work, and to develop good habits in choosing loop variable names, to banish the possibility of these annoying—and puzzling—program halts. ©

CAPUTE!

Modifications Or Corrections To Previous Articles

VIC Canyon Runner ✓

Many readers have reported that parts of line 830 in Program 2 (p. 62) of this game from the October issue were blurred. The line reads:

```
830 DATA156,30,31,158,169,128,141,19,145,
    169,0,133,1,133,2,169,127,141,34,145,
    162,119                         :rem 141
```

Commodore Autoboot

One step in the process for creating programs that run automatically was not made immediately clear in this article from the September issue (p. 130). After typing the POKEs and saving your original program with a new filename, you must reset the computer by turning it off and back on before loading and running "Autoboot."

Programs created by Autoboot may not run when loaded immediately after the computer is turned on or immediately after a cold start reset. The solution is to load *anything*—even the disk directory—before loading the autorun program. Once any other file has been loaded, the autorun programs will behave as expected.

READing DATA

Our mail indicates that many readers have problems typing DATA statements. Some letters insist that programs containing DATA statements never work properly. Typing mistakes in DATA lines often result in cryptic error messages or in mysterious program crashes, which often baffle the beginning programmer.

If any program stops with a SYNTAX ERROR, ILLEGAL QUANTITY, or OUT OF DATA message (ERROR 3, ERROR 6, or ERROR 8 on an Atari), check the line where the program was reported to have stopped. If that line contains a READ statement, *the error is probably not in the line for which the error was reported.* Instead, it's almost certain that you've typed something incorrectly in one or more of the DATA statements.

Errors in DATA can be hard to find. A common mistake is typing a period for a comma. These two characters appear side by side on the keyboard, and they can be difficult to distinguish on the screen. This mistake would, for example, cause the two DATA items 165,15 to be interpreted as one item, 165.15. Other frequent problems are transposing digits in DATA items (for example, typing 236 as 263) and adding extra commas at the ends of lines.

Other messages may signal a flaw in the program, but any of the error situations mentioned above point toward a typing mistake. Check carefully before you blame the program.

Proofreader And MLX Caveat

Many readers may be unaware that BASIC programs entered with the aid of "The Automatic Proofreader" or machine language programs entered with "MLX" can still contain typing errors. While these two utilities greatly reduce the chances of making a typing mistake, there are errors that are not detected.

The Commodore and Atari Proofreader programs check only that the correct characters are present, not whether they are in the correct order. The following line has a Commodore Proofreader checksum of 117:

```
100 POKE 214,12:A=19.7:PRINT A
```

If you scramble the line and type:

```
100 PKOE 241,21:A=17.9:PNIRT A
```

the Proofreader still reports a checksum of 117. This is an extreme case; more subtle mistakes are more difficult to detect. One reader insisted that COMPUTE! had made an error, since his program didn't work even though the Proofreader showed his typing to be correct. However, the listing he sent along showed that in one line he typed GOTO 535 where he should have had GOTO 355. That mistake was invisible to the Proofreader—all the right characters were present—but it made the program crash without warning. The Proofreader can also cause you to overlook typing errors in DATA statements. The numbers 169 and 196 look the same to the Proofreader, but in the DATA for a machine language program such a difference will cause a prompt crash.

MLX, on the other hand, can detect the transposition of individual digits within a single number—if you type 196 where you mean 169, MLX will report the mistake—but it cannot detect transposition of entire numbers. For example, given the following line from an MLX-format listing:

```
49188 :160,000,185,071,201,201,086
```

MLX will *not* notice the mistake if you instead type:

```
49188 :160,185,000,201,071,201,086
```

©

COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program *exactly* as listed, including any necessary punctuation and symbols. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Since programs can contain some hard-to-read (and hard-to-type) special characters, we have developed a listing system that spells out in abbreviated form the function of these control characters. You will find these special characters within curly braces. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. A symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. Commodore machines have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a new kind of special bracket. A graphics character can be listed as [<A>]. In this case, hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as $\underline{S}$.

If a number precedes a symbol, such as {5 RIGHT}, {6 $\underline{S}$}, or [<8 Q>], you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered with the Atari logo key. Since spacing is sometimes important, any more than two spaces will be listed, for example, as: {6 SPACES}. A space is never left at the end of a line, but will be moved to the next printed line as {SPACE}. There are no special control characters found in our IBM PC/PCjr, TI-99/4A, and Apple program listings. For your convenience, we have prepared this quick-reference key for the Commodore and Atari special characters:

Atari 400/800/XL

When you see	Type	See	
{CLEAR}	ESC SHIFT <		Clear Screen
{UP}	ESC CTRL -	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE		Backspace
{DELETE}	ESC CTRL DELETE		Delete character
{INSERT}	ESC CTRL INSERT		Insert character
{DEL LINE}	ESC SHIFT DELETE		Delete line
{INS LINE}	ESC SHIFT INSERT		Insert line
{TAB}	ESC TAB		TAB key
{CLR TAB}	ESC CTRL TAB		Clear tab
{SET TAB}	ESC SHIFT TAB		Set tab stop
{BELL}	ESC CTRL 2		Ring buzzer
{ESC}	ESC ESC		ESCape key

Commodore PET/CBM/VIC/64

When You Read:	Press:		See:	When You Read:	Press:		See:
{CLR}	SHIFT	CLR/HOME		{GRN}	CTRL	6	
{HOME}		CLR/HOME		{BLU}	CTRL	7	
{UP}	SHIFT	↑ CRSR ↓		{YEL}	CTRL	8	
{DOWN}		↑ CRSR ↓		{F1}	f1		
{LEFT}	SHIFT	← CRSR →		{F2}	f2		
{RIGHT}		← CRSR →		{F3}	f3		
{RVS}	CTRL	9		{F4}	f4		
{OFF}	CTRL	0		{F5}	f5		
{BLK}	CTRL	1		{F6}	f6		
{WHT}	CTRL	2		{F7}	f7		
{RED}	CTRL	3		{F8}	f8		
{CYN}	CTRL	4		←	←		
{PUR}	CTRL	5		↑	SHIFT	↑	

The Automatic Proofreader

Also, we have developed a simple, yet effective program that can help check your typing. Type in the appropriate Proofreader program for your machine, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader will still be active, hidden in memory, as a machine language program). Pressing RUN/STOP–RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reenable the Atari Proofreader. The IBM Proofreader is a BASIC program that lets you enter, edit, list, save, and load programs that you type. It simulates the IBM's BASIC line editor.

Using The Automatic Proofreader

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a number (on the Commodore) or a pair of letters

(Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need *not* be typed in. It is just there for your information.

In Atari and IBM listings, the checksum is given to the left of each line number. Just type in the program, a line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore and Atari Proofreader, spaces are not counted as part of the checksum, and no check is made to see that you've typed in the characters in the right order. If characters are transposed, the checksum will still match the listing. Because of the checksum method used, do not use abbreviations, such as ? for PRINT. However, the Proofreader does catch the majority of typing errors most people make. The IBM Proofreader is even pickier; it *will* detect errors in spacing and transposition. Also, be sure you leave Caps Lock on, except when you need to enter lowercase characters.

Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in the cassette buffer, which is used during tape LOADs and SAVEs. Be sure to press RUN/STOP–RESTORE before you save or load a program, to get the Proofreader out of the way. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines *exactly* as shown, pressing RETURN after each one:

```
A$="PROOFREADER.T":B$="{10 SPACES}"
   :FORX=1TO4:A$=A$+B$:NEXT

FORX=886TO1018:A$=A$+CHR$(PEEK(X))
   :NEXT:OPEN 1,1,1,A$:CLOSE1
```

Then press RECORD and PLAY on a blank tape, and a special version of the Proofreader will be saved to tape. Anytime you need to reload the Proofreader after it has been erased, just rewind the tape, type OPEN1:CLOSE1, then press PLAY. When READY comes back, enter SYS 886.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader will prompt you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program into the normal BASIC environment (this will replace the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE "filename",A.

Program 1: VIC/64 Proofreader

```
100 PRINT"{CLR}PLEASE WAIT...":FORI=886TO10
    18:READA:CK=CK+A:POKEI,A:NEXT
110 IF CK<>17539 THEN PRINT"{DOWN}YOU MADE
    {SPACE}AN ERROR":PRINT"IN DATA STATEMEN
    TS.":END
120 SYS886:PRINT"{CLR}{2 DOWN}PROOFREADER A
    CTIVATED.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003
```

Program 2: Atari Proofreader

```
100 GRAPHICS 0
110 FOR I=1536 TO 1700:READ A:POK
    E I,A:CK=CK+A:NEXT I
120 IF CK<>19072 THEN ? "ERROR IN
     DATA STATEMENTS   CHECK TYPI
    NG.":END
130 A=USR(1536)
140 ? :? "AUTOMATIC PROOFREADER N
    OW ACTIVATED."
150 END
```

```
1536 DATA 104,160,0,185,26,3
1542 DATA 201,69,240,7,200,200
1548 DATA 192,34,208,243,96,200
1554 DATA 169,74,153,26,3,200
1560 DATA 169,6,153,26,3,162
1566 DATA 0,189,0,228,157,74
1572 DATA 6,232,224,16,208,245
1578 DATA 169,93,141,78,6,169
1584 DATA 6,141,79,6,24,173
1590 DATA 4,228,105,1,141,95
1596 DATA 6,173,5,228,105,0
1602 DATA 141,96,6,169,0,133
1608 DATA 203,96,247,238,125,241
1614 DATA 93,6,244,241,115,241
1620 DATA 124,241,76,205,238,0
1626 DATA 0,0,0,0,32,62
1632 DATA 246,8,201,155,240,13
1638 DATA 201,32,240,7,72,24
1644 DATA 101,203,133,203,104,40
1650 DATA 96,72,152,72,138,72
1656 DATA 160,0,169,128,145,88
1662 DATA 200,192,40,208,249,165
1668 DATA 203,74,74,74,74,24
1674 DATA 105,161,160,3,145,88
1680 DATA 165,203,41,15,24,105
1686 DATA 161,200,145,88,169,0
1692 DATA 133,203,104,170,104,168
1698 DATA 104,40,96
```

Program 3: IBM Proofreader

```
100 DIM L$(500),LNUM(500):COLOR 0,7,7:KEY OFF:CLS:MAX=0:LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,CHR$(4)+CHR$(70):ON KEY(15) GOSUB 640:KEY (15) ON:GOTO 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:PRINT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-INT(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:POKE 1052,34:POKE 1054,0:POKE 1055,79:POKE 1056,13:POKE 1057,28:LINE INPUT L$:DEF SEG:IF L$="" THEN 150
170 IF LEFT$(L$,1)=" " THEN L$=MID$(L$,2):GOTO 170
180 IF VAL(LEFT$(L$,2))=0 AND MID$(L$,3,1)=" " THEN L$=MID$(L$,4)
190 LNUM=VAL(L$):TEXT$=MID$(L$,LEN(STR$(LNUM))+1)
200 IF ASC(L$)>57 THEN 260 'no line number, therefore command
210 IF TEXT$="" THEN GOSUB 540:IF LNUM=LNUM(P) THEN GOSUB 560:GOTO 150 ELSE 150 'delete line
220 CKSUM=0:FOR I=1 TO LEN(L$):CKSUM=(CKSUM+ASC(MID$(L$,I))*I) AND 255:NEXT:LOCATE Y,1:PRINT CHR$(65+CKSUM/16)+CHR$(65+(CKSUM AND 15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM THEN L$(P)=TEXT$:GOTO 150 'replace line
240 GOSUB 580:GOTO 150 'insert the line
250 'command processor. step 1: convert to uppercase
260 TEXT$="":FOR I=1 TO LEN(L$):A=ASC(MID$(L$,I)):TEXT$=TEXT$+CHR$(A+32*(A>96 AND A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "):COMMAND$=TEXT$:ARG$="":IF DELIMITER THEN COMMAND$=LEFT$(TEXT$,DELIMITER-1):ARG$=MID$(TEXT$,DELIMITER+1) 'separate command from argument
280 IF COMMAND$<>"LIST" THEN 410
290 OPEN "scrn:" FOR OUTPUT AS #1
300 IF ARG$="" THEN FIRST=0:P=MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-"):IF DELIMITER=0 THEN LNUM=VAL(ARG$):GOSUB 540:FIRST=P:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELIMITER)):LAST=VAL(MID$(ARG$,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRST=P:LNUM=LAST:GOSUB 540:IF P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(STR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GOTO 370
360 CKSUM=0:A$=N$+L$(X):FOR I=1 TO LEN(A$):CKSUM=(CKSUM+ASC(MID$(A$,I))*I) AND 255:NEXT:A$=CHR$(65+CKSUM/16)+CHR$(65+(CKSUM AND 15))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN OPEN "lpt1:" FOR OUTPUT AS #1:GOTO 300
420 IF COMMAND$="CHECK" THEN CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN 450
440 GOSUB 600:OPEN ARG$ FOR OUTPUT AS #1:ARG$="":GOTO 300
450 IF COMMAND$<>"LOAD" THEN 490
460 GOSUB 600:OPEN ARG$ FOR INPUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INPUT #1,L$:LNUM(P)=VAL(L$):L$(P)=MID$(L$,LEN(STR$(VAL(L$)))+1):P=P+1:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN INPUT "Erase program - Are you sure";L$:IF LEFT$(L$,1)="y" OR LEFT$(L$,1)="Y" THEN MAX=0:GOTO 130:ELSE 130
500 IF COMMAND$="BASIC" THEN COLOR 7,0,0:ON ERROR GOTO 0:CLS:END
510 IF COMMAND$="FILES" THEN FILES:GOTO 130
520 PRINT"Syntax error":GOTO 130
530 'find line
540 P=0:WHILE LNUM>LNUM(P) AND P<MAX:P=P+1:WEND:RETURN
550 'delete line
560 MAX=MAX-1:FOR X=P TO MAX:LNUM(X)=LNUM(X+1):L$(X)=L$(X+1):NEXT:RETURN
570 'insert line
580 MAX=MAX+1:FOR X=MAX TO P+1 STEP -1:LNUM(X)=LNUM(X-1):L$(X)=L$(X-1):NEXT:L$(P)=TEXT$:LNUM(P)=LNUM:RETURN
590 'filename adjustments
600 IF LEFT$(ARG$,1)<>CHR$(34) THEN 520 ELSE ARG$=MID$(ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34) THEN ARG$=LEFT$(ARG$,LEN(ARG$)-1)
620 IF INSTR(ARG$,".")=0 THEN ARG$=ARG$+".BAS"
630 RETURN
640 PRINT"Stopped.":RETURN 150
650 PRINT "Error #";ERR:RESUME 150
```

©

MLX Machine Language Entry Program For Commodore 64 And Unexpanded VIC-20

Charles Brannon, Program Editor

MLX is a labor-saving utility that allows almost fail-safe entry of machine language programs published in COMPUTE!. You need to know nothing about machine language to use MLX—it was designed for everyone.

MLX is a new way to enter long machine language (ML) programs with a minimum of fuss. MLX lets you enter the numbers from a special list that looks similar to BASIC DATA statements. It checks your typing on a line-by-line basis. It won't let you enter illegal characters when you should be typing numbers. It won't let you enter numbers greater than 255 (forbidden in ML). It won't let you enter the wrong numbers on the wrong line. In addition, MLX creates a ready-to-use tape or disk file.

Using MLX

Type in and save the appropriate version of MLX (you'll want to use it in the future). When you're ready to type in an ML program, run MLX. MLX for the 64 asks you for two numbers: the starting address and the ending address. These numbers are given in the article accompanying the ML program. Tiny MLX for the unexpanded VIC *does not* ask for the starting and ending address of the program to be entered. Instead, this information must be included in line 210. The values in line 210 of Tiny MLX as listed here are for the "Spiders" program in this issue. Line 100 will also have to be adjusted for each program you type in with Tiny MLX. Refer to the program's article for details.

When you run MLX, you'll see a prompt corresponding to the starting address. The prompt is the current line you are entering from the listing. It increases by six each time you enter a line. That's because each line has seven numbers—six actual data numbers plus a *checksum number.* The checksum verifies that you typed the previous six numbers correctly. If you enter any of the six numbers wrong, or enter the checksum wrong, the computer rings a buzzer and prompts you to reenter the line. If you enter it correctly, a bell tone sounds and you continue to the next line.

MLX accepts only numbers as input. If you make a typing error, press the INST/DEL key; the entire number is deleted. You can press it as many times as necessary back to the start of the line. If you enter three-digit numbers as listed, the computer automatically prints the comma and goes on to accept the next number. If you enter less than three digits, you can press either the space bar or RETURN key to advance to the next number. The checksum automatically appears in inverse video for emphasis.

To simplify your typing, 64 MLX redefines part of the keyboard as a numeric keypad (lines 581–584):

	U	I	O			7	8	9
H	J	K	L	become	0	4	5	6
	M	,	.			1	2	3

64 MLX Commands

When you finish typing an ML listing (assuming you type it all in one session), you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you've made a typo when entering the MLX program itself.

You don't have to enter the whole ML program in one sitting. MLX lets you enter as much as you want, save it, and then reload the file from tape or disk later. 64 MLX recognizes these commands:

SHIFT-S: Save
SHIFT-L: Load
SHIFT-N: New Address
SHIFT-D: Display

When you enter a command, MLX jumps out of the line you've been typing, so we recommend you do it at a new prompt. Use the Save command to save what you've been working on. It will save on tape or disk as if you've finished, but the tape or disk won't work, of course, until you finish the typing. Remember what address you stop at. The next time you run MLX, answer all the prompts as you did before, then insert the disk or tape. When you get to the entry prompt, press SHIFT-L to reload the partly completed file into memory. Then use the New Address command to resume typing.

To use the New Address command, press SHIFT-N and enter the address where you previously stopped. The prompt will change, and you can then continue typing. Always enter a New Address that matches up with one of the line numbers in the special listing, or else the checksum won't work. The Display command lets you display a section of your typing. After you press SHIFT-D, enter two addresses within the line number range of the listing. You can abort the listing by pressing any key.

Tiny MLX Commands

To squeeze Tiny MLX into the unexpanded VIC and still leave room for the ML program being entered, all the special commands of the 64 version had to be omitted, as well as the provision for the redefined keyboard. Since Tiny MLX has no provisions for reloading a partially completed program, you must enter all the ML data in one sitting. When you finish typing an ML listing, you can then save the completed program on tape or disk. Follow the screen instructions. If you get any errors while saving, you probably have a bad disk, or the disk is full, or you made a typo when entering the MLX program itself.

Program 1: 64 MLX

```
10 REM LINES CHANGED FROM MLX VERSION 2.00
   {SPACE}ARE 750,765,770 AND 860    :rem 50
20 REM LINE CHANGED FROM MLX VERSION 2.01 I
   S 300                             :rem 147
100 PRINT"{CLR}[6]";CHR$(142);CHR$(8);:POKE
    53281,1:POKE53280,1              :rem 67
101 POKE 788,52:REM DISABLE RUN/STOP
                                     :rem 119
110 PRINT"{RVS}{39 SPACES}";         :rem 176
120 PRINT"{RVS}{14 SPACES}{RIGHT}{OFF}[*]£
    {RVS}{RIGHT} {RIGHT}{2 SPACES}[*]{OFF}
    [*]£{RVS}£{RVS}{14 SPACES}"; :rem 250
130 PRINT"{RVS}{14 SPACES}{RIGHT} [G]
    {RIGHT} {2 RIGHT} {OFF}£{RVS}£[*]
    {OFF}[*]{RVS}{14 SPACES}";       :rem 35
140 PRINT"{RVS}{41 SPACES}"          :rem 120
200 PRINT"{2 DOWN}{PUR}{BLK} MACHINE LANGUA
    GE EDITOR VERSION 2.02{5 DOWN}":rem 238
210 PRINT"[5]{2 UP}STARTING ADDRESS?
    {8 SPACES}{9 LEFT}";             :rem 143
215 INPUTS:F=1-F:C$=CHR$(31+119*F)  :rem 166
220 IFS<256OR(S>40960ANDS<49152)ORS>53247TH
    ENGOSUB3000:GOTO210              :rem 235
225 PRINT:PRINT:PRINT                :rem 180
230 PRINT"[5]{2 UP}ENDING ADDRESS?
    {8 SPACES}{9 LEFT}";:INPUTE:F=1-F:C$=CH
    R$(31+119*F)                     :rem 20
240 IFE<256OR(E>40960ANDE<49152)ORE>53247TH
    ENGOSUB3000:GOTO230              :rem 183
250 IFE<STHENPRINTC$;"{RVS}ENDING < START
    {2 SPACES}":GOSUB1000:GOTO 230 :rem 176
260 PRINT:PRINT:PRINT                :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S       :rem 56
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),2)
    ,5);":";                         :rem 33
315 FORJ=ATO6                        :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                     :rem 228
390 IFN=-211THEN 710                 :rem 62
400 IFN=-204THEN 790                 :rem 64
410 IFN=-206THENPRINT:INPUT"{DOWN}ENTER NEW
     ADDRESS";ZZ                     :rem 44
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"{RVS}
    OUT OF RANGE":GOSUB1000:GOTO410:rem 225
417 IFN=-206THENAD=ZZ:PRINT:GOTO310:rem 238
420 IF N<>-196 THEN 480              :rem 133
430 PRINT:INPUT"DISPLAY:FROM";F:PRINT,"TO";
    :INPUTT                          :rem 234
440 IFF<SORF>EORT<SORT>ETHENPRINT"AT LEAST"
    ;S;"{LEFT}, NOT MORE THAN";E:GOTO430
                                     :rem 159
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("0000"
    +MID$(STR$(I),2),5);":";         :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00"+
    MID$(STR$(N),2),3);",";          :rem 66
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310
                                     :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRINT:
    GOTO310                          :rem 50
480 IFN<0 THEN PRINT:GOTO310        :rem 168
490 A(J)=N:NEXTJ                     :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CKSU
    M=(CKSUM+A(I))AND255:NEXT        :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146);
                                     :rem 94
511 IFN=-1THENA=6:GOTO315           :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530  :rem 122
520 PRINT:PRINT"LINE ENTERED WRONG : RE-ENT
    ER":PRINT:GOSUB1000:GOTO310     :rem 176
530 GOSUB2000                        :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE5427
    2,0:POKE54273,0                  :rem 227
550 AD=AD+6:IF AD<E THEN 310        :rem 212
560 GOTO 710                         :rem 108
570 N=0:Z=0                          :rem 88
580 PRINT"[£]";                      :rem 81
581 GETA$:IFA$=""THEN581             :rem 95
582 AV=-(A$="M")-2*(A$=",")-3*(A$=".")-4*(A
    $="J")-5*(A$="K")-6*(A$="L")    :rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"):
    IFA$="H"THENA$="0"               :rem 134
584 IFAV>0THENA$=CHR$(48+AV)        :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44OR
    A=32THEN670                      :rem 229
590 IFA>128THENN=-A:RETURN           :rem 137
600 IFA<>20 THEN 630                 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
    {OFF}{LEFT} {LEFT}";:GOTO690    :rem 62
620 GOTO570                          :rem 109
630 IFA<48ORA>57THEN580              :rem 105
640 PRINTA$;:N=N*10+A-48             :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                     :rem 229
660 Z=Z+1:IFZ<3THEN580               :rem 71
670 IFZ=0THENGOSUB1000:GOTO570      :rem 114
680 PRINT",";:RETURN                 :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                     :rem 149
691 FORI=1TO3:T=PEEK(S%-I)           :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                     :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                     :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                     :rem 236
715 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF} A
    LONE TO CANCEL SAVE){DOWN}"     :rem 106
720 F$="":INPUT"{DOWN} FILENAME";F$:IFF$=""
    THENPRINT:PRINT:GOTO310          :rem 71
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
    {RVS}D{OFF}ISK: (T/D)"           :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740:rem 36
```

```
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:OP
EN15,8,15,"S"+F$:CLOSE15                  :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$):
POKE782,ZK/256                            :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(T$
):SYS65469                                :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS65466
                                          :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)*2
56:POKE780,253                            :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(782)
*256:SYS65496                             :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
                                          :rem 111
775 PRINT"{DOWN}DONE.{DOWN}":GOTO310
                                          :rem 113
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}TRY
 AGAIN.":IFDV=1THEN720                    :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$;E
2$:CLOSE15:GOTO720                        :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***{2 DOWN}"
                                          :rem 212
795 PRINT"{2 DOWN}(PRESS {RVS}RETURN{OFF} A
LONE TO CANCEL LOAD)"                     :rem 82
800 F$="":INPUT"{2 DOWN} FILENAME";F$:IFF$=
""THENPRINT:GOTO310                       :rem 144
810 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
{RVS}D{OFF}ISK: (T/D)"                    :rem 227
820 GETA$:IFA$<>"T"ANDA$<>"D"THEN820:rem 34
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                          :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$):
POKE782,ZK/256                            :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(T$
):SYS65469                                :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS65466
                                          :rem 70
850 POKE780,0:SYS65493                    :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
                                          :rem 111
865 PRINT"{DOWN}DONE.":GOTO310            :rem 96
870 PRINT"{DOWN}ERROR ON LOAD.{2 SPACES}TRY
 AGAIN.{DOWN}":IFDV=1THEN800              :rem 172
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$;E
2$:CLOSE15:GOTO800                        :rem 102
1000 REM BUZZER                           :rem 135
1001 POKE54296,15:POKE54277,45:POKE54278,16
5                                         :rem 207
1002 POKE54276,33:POKE 54273,6:POKE54272,5
                                          :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE5427
3,0:POKE54272,0:RETURN                    :rem 202
2000 REM BELL SOUND                       :rem 78
2001 POKE54296,15:POKE54277,0:POKE54278,247
                                          :rem 152
2002 POKE 54276,17:POKE54273,40:POKE54272,0
                                          :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
                                          :rem 57
3000 PRINTC$;"{RVS}NOT ZERO PAGE OR ROM":GO
TO1000                                    :rem 89
```

Program 2: VIC Tiny MLX

```
100 POKE55,0:POKE56,25:CLR                :rem 8
210 S=6405:E=7676                         :rem 136
300 PRINT"{CLR}";CHR$(14):AD=S            :rem 56
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5);
":";:FORJ=1TO6                            :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                          :rem 228
480 IFN<0THENPRINT:GOTO310                :rem 168
490 A(J)=N:NEXTJ                          :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CKSU
M=(CKSUM+A(I))AND255:NEXT                 :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                          :rem 234
515 IFN=CKSUMTHEN530                      :rem 255
520 PRINT:PRINT"LINE ENTERED WRONG":PRINT"R
E-ENTER":PRINT:GOSUB1000:GOTO310
                                          :rem 129
530 GOSUB2000                             :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT        :rem 80
550 AD=AD+6:IFAD<ETHEN310                 :rem 212
560 GOTO710                               :rem 108
570 N=0:Z=0                               :rem 88
580 PRINT"{+}";                           :rem 79
581 GETA$:IFA$=""THEN581                  :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44OR
A=32THEN670                               :rem 229
590 IFA>128THENN=-A:RETURN                :rem 137
600 IFA<>20 THEN 630                      :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
{LEFT} {LEFT}";:GOTO690                   :rem 172
620 GOTO570                               :rem 109
630 IFA<48ORA>57THEN580                   :rem 105
640 PRINTA$;:N=N*10+A-48                  :rem 106
650 IFN>255 THEN A=20:GOSUB1000:GOTO600
                                          :rem 229
660 Z=Z+1:IFZ<3THEN580                    :rem 71
670 IFZ=0THENGOSUB1000:GOTO570            :rem 114
680 PRINT",";:RETURN                      :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
                                          :rem 149
692 FORI=1TO3:T=PEEK(S%-I)                :rem 68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
                                          :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1);:RETURN
                                          :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***{3 DOWN}"
                                          :rem 236
720 INPUT"{DOWN} FILENAME";F$             :rem 228
730 PRINT:PRINT"{2 DOWN}{RVS}T{OFF}APE OR
{RVS}D{OFF}ISK: (T/D)"                    :rem 228
740 GETA$:IFA$<>"T"ANDA$<>"D"THEN740:rem 36
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
                                          :rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$):
POKE782,ZK/256                            :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(T$
):SYS65469                                :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS65466
                                          :rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256:P
OKE780,253                                :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256:S
YS65496                                   :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780
                                          :rem 111
775 PRINT"{DOWN}DONE.":END                :rem 106
780 PRINT"{DOWN}ERROR ON SAVE.{2 SPACES}TRY
 AGAIN.":IFDV=1THEN720                    :rem 171
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$;E
2$:CLOSE15:GOTO720                        :rem 103
782 GOTO720                               :rem 115
845 POKE780,1:POKE781,DV:POKE782,1:SYS65466
                                          :rem 70
1000 REM BELL TONE                        :rem 250
1001 POKE36878,15:POKE36874,190           :rem 206
1002 FORW=1TO300:NEXTW                    :rem 117
1003 POKE36878,0:POKE36874,0:RETURN :rem 74
2000 REM BELL SOUND                       :rem 78
2001 FORW=15TO0STEP-1:POKE36878,W:POKE36876
,240:NEXTW                                :rem 22
2002 POKE36876,0:RETURN                   :rem 119
```

ATARI®

600XL .... CALL
800XL .... CALL

DISK DRIVES
- Rana 1000 .... $298
- Astra 2001 .... $549
- Indus GT $309
- Trak AT-D2 .... $329
- Trak AT-1 .... $319
- Trak AT-D4 .... Call
- Astra 1620 (Dual) ... $499
- Percom .... Call
- Atari 1050 .... $249

INTERFACES
- Axiom 846 .... Call
- Ape Face .... Call
- Atari 850 (In Stock) .. $169
- Interfast 1 .... $150
- Microbits 1150 ...... Call
- R-Verter .... Call

MEMORIES
- Microbits 64K (XL) .. $115
- Mosaic 48K (400) $98
- Mosaic 64K (400/800) Call
- Mosaic 32K $68
- Atari 64K (600XL) .... Call

DIRECT PRINTERS
- Axiom AT-100 ...... $195
- Atari 1027 .... $269
- Axiom 550 AT $259
- Axiom 700 AT $469
- Atari 1025 .... $299

DIRECT MODEMS
- Microbits 1000C $128

OTHER ATARI
- 400 Keyboard .... Call
- Koala Pad $67
- Chalkboard Pad $75
- Bit-3 80 Column $228
- ATR-8000 (64K) $489
- ATR-8000 (16K) $359
- Alien Voice Box $98
- 1010 Recorder $74

ATARI SOFTWARE

MISCELLANEOUS
- Syn Calc (D) $59
- Syn File (D) $59
- Syn Trend (D) $48
- Syn Com (D) $29
- Syn Chron (D) $29
- Decathlon (R) $29
- Drols (D) $23
- Gyruss (R) $31
- Heist (D) $23
- Bruce Lee (C/D) $27
- Universe (D) .... Call
- Questron (D) $34
- Koala Coloring I $20
- Koala Logo Design $27
- Bumble Games (D) $27
- Miles Accounting ..... Call
- World Gtst. Baseball .. $29
- Gridrunner (R) $20
- Sargon II (C/D) $23
- Millionaire (D) $34
- Castle Wolfenstein (D) $20
- Odesta Chess (D) $46
- Financial Wizard (D) .. $41
- Ultima III (D) $39

ADVENTURE INT'L
- Ultra Disassembler (D) $33
- Diskey (D) $33
- Adv. 1-12 (each) (C) ... $18
- Saga 1-12 (each) (D) .. $27

ATARI
- Atari Writer (R) $68
- Paint (D) $30
- Microsoft Basic II (R) . $64
- Visicalc (D) $139
- Home File Mgr (D) ... $36
- Assembler Editor (R) . $44
- Qix (R) $32
- Dig Dug (R) $32
- Atari Logo (R) $72
- Ms. Pac Man (R) $33
- Joust (R) $33
- Donkey Kong Jr. (R) . $35
- Computer Chess (R) . $24

AVALON HILL
- Telengard . (C) $16 (D) $19
- Close Assault (C) 20 (D) 23
- TAC (D) $27

BRODERBUND
- Arcade Machine (D) .. $39
- Bank St. Writer (D) ... $46
- Oper. Whirlwind (D) .. $27
- Choplifter . (D) $23 (R) $29

CBS SOFTWARE .... Call

CONTINENTAL
- Home Accountant (D) $48
- Tax Advantage (D) ... $45

COUNTERPOINT SW Call

DATASOFT
- Pooyan (C/D) $20
- Teletalk (D) $33
- Graphic Master (D) ... $23
- Micropainter (D) $23
- Zaxxon (C/D) $27

EASTERN HOUSE
- Monkey Wrench II ... $51

EDUCATIONAL SW
- Tricky 1,2,3 or 4 ...... $15
- Tricky 5-13 .... $22

EPYX
- Dragon Riders (C/D) .. $27
- Temple APS (C/D) $27
- Jumpman (C/D) $27

FIRST STAR
- Boulder Dsh (C/D) 20 (R) 27
- Bristles (C/D) $20
- Flip Flop (C/D) $20

GAMESTAR
- Football (C/D) $21
- Baseball (C/D) $21

INFOCOM
- Zork I, II or III (D) $27
- Deadline (D) $34
- Starcross (D) $27
- Suspended (D) $34
- Witness (D) $34
- Planetfall (D) $34
- Enchanter (D) $34
- Infidel (D) $34

KRELL SAT .... Call

INTELL. STATEMENTS
- Prof. Blackjack (D) ... $46

LJK
- Letter Perfect (D) $74
- Data Perfect (D) $74
- Spell Perfect (D) $56
- Letter Perfect (R) $74

MICROPROSE
- Solo Flight (D) $26
- Hellcat Ace (C/D) $23

MONARCH
- ABC Compiler (D) $55

OPTIMIZED SYSTEMS
- Action (R) $65
- Basic XL (R) $65
- Mac 65 (D) $58
- C-65 (D) $58
- Bug 65 (D) $23

PARKER BROS
- Astrochase (R) $33
- Death Star (R) $33
- Q-Bert (R) $33
- Popeye (R) $33

QUALITY SW
- Return of Hercules (D) $22
- Ali Baba (D) $22

RESTON
- Moviemaker (D) $45

SCARBOROUGH SYS.
- Mastertype (D/R) $27
- Songwriter (D) $27

SCHOOL WIZWARE . Call

SIERRA ON-LINE
- Homeword (D) $46
- Ultima II (D) $39
- Dark Crystal (D) $26
- Wiz. & Princess (D) ... $22

SPINNAKER
- Snooper Troop 1,2 (D) . $30
- Most Amazing (D) $27
- Kids on Keys (D) $20
- Trains (D) $27
- Delta Drawing (R) $27
- Aerobics (D) $34

STRATEGIC SIM.
- Broadsides (D) $27
- Carrier Force (D) $39
- Combat Leader (D) ... $27
- Rails West (D) $27
- Epidemic (D) $23
- Eagles (D) $27
- Cosmic Bal I or II (D) . $27

SUBLOGIC
- Flight Simulator II (D) . $36
- Pinball (C/D) $20

SYNAPSE
- File Manager (R) $54
- Fort Apocalypse (C/D) $23
- Dimension X (C/D) $23
- Blue Max (C/D) $23
- Encounter (D/R) $23
- Zepplin (C/D) $23
- Pharoah's Curse (C/D) $23

TRONIX
- S.A.M. (D) $39
- P.M. Animator (D) $29
- Juice (C/D) $20
- Chatterbee (D) $27

Printers/Etc.

DIABLO
- 630 Letter Qual. $1648

SILVER REED
- EXP 400 Ltr. Qual. .. $310
- EXP 500 Ltr. Qual. .. $379
- EXP 550 Ltr. Qual. .. $449
- EXP 770 Ltr. Qual. .. $888

STAR
- Gemini 10X $259
- Gemini 15X $378
- Delta 10 .... $378
- Delta 15 .... $519
- Radix 10 .... $549
- Radix 15 .... $645
- Power Type .... Call

TOSHIBA
- 1340 .... $788
- 1351 .... $1299

NEC
- 3510 .... $1259
- 3530 .... $1259
- 3550 .... $1598
- 7710/7730 .... $1719

CITOH
- Prowriter $335
- Prowriter II $569
- Starwriter $929
- Printmaster $1198

OKIDATA
- 82A .... Call
- 84P .... Call
- 92 .... $409
- 93 .... Call

DAISYWRITER
- 2000 .... $995

MANNESMANN
- 160L $559
- Spirit $299

JUKI
- 6100 .... $419

ABATI
- LQ 20P Ltr. Qual. $378

PANASONIC
- 1090 .... $255
- 1091 .... $309
- 1092 .... $458

MONITORS

AMDEK
- V300 G $139
- V300 A $149
- V310 G (IBM) $155
- V310 A (IBM) $165
- Color I+ $269
- Color II+ $399
- Color III $349
- Color IV (IBM) $699

PRINCETON GRAPHICS
- MAX 12 (Amber) $178
- HX 12 (RGB) $489
- SR 12 (RGB) $619

SUPER 5
- 100A (Amber) $99
- 500G (IBM with tilt) . $126
- 500A (IBM with tilt) . $126

SAKATA
- SC100 (Color) $239
- 1000G (Green) $99

TAXAN
- 100 Green $115
- 105 Amber $125
- 210 RGB/Composite . $269
- 400 RGB Med-Res. . $296
- 415 RGB Hi-Res. $429
- 420 RGB Hi-Res.(IBM) $479

NEC
- JB 1260 (Grn) $99
- JB 1201 (Grn) $145
- JB 1205 (Amber) $145

ZENITH
- Green $85
- Amber $95

MODEMS

NOVATION
- J-Cat $99
- Apple Cat II $259
- D-Cat $149

HAYES
- Smartmodem $209
- Smartmodem 1200 .. $498
- Micromodem II $259
- Micromodem IIe $239

PROMETHIUS
- Promodem 1200 .... $349

D = DISK
R = ROM (Cartridge)
C = CASSETTE

OPEN MON-SAT

COSMIC™

COSMIC COMPUTERS

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

IN CALIF. **(714) 594-5204**
FOR TECHNICAL INFO, ORDER INQUIRIES,

Add $2.50 shipping per software order in continental U.S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10.00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock. Personal checks require 4 weeks to clear. MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change.

COMMODORE

INTERFACES
- The Connection $85
- Bus Card $149
- Cardco G+ $69
- Cardco B .... Call
- MSD (IEEE) $98
- Cardco 5 Slot $48
- Grappler C D $109

DIRECT MODEMS
- Hesmodem $53
- 1650 Automodem $99
- 1600 Modem .... Call
- Westridge Modem ... Call

80 COLUMN BDS
- Batteries 80 Col $149
- Video Pak 80 .... $129
- Z80 Video Pak $209

DISK DRIVES
- MSD (170K) $349
- MSD (Dual) (170Kx2) . $539
- Laser (170K) $325
- Commodore 1541 ... $239
- Concord (170K) ..... Call

RECORDERS
- Cardco Recorder $48
- 1530 Commodore .... Call
- Cassette Interface ... $29
- Phonemark Rec. $37

DIRECT PRINTERS
- MPS 801 .... $219
- Commodore 1525 ... $288
- Cardco LQ/I $498
- 1520 Color Printer ... $129

64 SOFTWARE 64

MISCELLANEOUS
- MAE Assembler (D) .. $47
- VIP Terminal (D) $38
- Star Wars (R) $33
- Super Base 64 (D) ... $68
- Doodle City (D) $27
- Summer Games (D) .. $27
- Pitfall II (D) $25
- Decathlon (D) $25
- 50 Mission Crush (D) . $27
- IFR. (C/D) $20
- Master Composer (D) . $27
- Donkey Kong (R) $29
- Bruce Lee (D) $23
- Pro Football Stat. (D) . $56
- Seastalker (D) $27
- Koala Coloring I $20
- Koala Logo Design ... $27
- Rockys Boots (D) $33
- Bumble Games (D) .. $27
- Beyond Wolfenstein ... $23
- Peachtree Account .. Call
- Odesta Chess (D) $46
- Ultima III (D) $39
- Prof. Blackjack (D) ... $46
- Homeword (D) $46
- Pers. Accountant (D) . $23
- Karate Devils (D) $27
- Final Flight (D) $22
- Diskey (D) $33
- Barrons Sat (D) $59
- Millionaire (D) $39
- Sargon II (D) $23
- B-Graph (D) $59
- Castle Wolfenstein (D) $20

ACCESS SOFTWARE
- Beached (D) $24
- Neutral Zone (C/D) ... $24
- Spritemaster (D) $25

AVALON HILL
- Nuke War (C) $12
- Androm. Conquest (C) $14
- Midway Campaign (C) $13
- Computer Football (C) $13
- Telengard . (C) $16 (D) $19
- Flying Ace (C) $15
- Moon Patrol (C) $17

BATTERIES INCLUDED
- Paper Clip (D) $69
- Consultant (D) $75
- Paper Clip w/Spell (D) $85
- Spell Pack (D) $37
- Organizer Series (Ea) . $22

BLUE SKY
- Calc Result Adv. $99
- Calc Result Easy $57

BRODERBUND
- Bank St. Writer (D) ... $46
- Operat. Whrlwnd (D) .. $27
- Choplifter (R) $27
- Lode Runner . (D) 23 (R) 27

CBS SOFTWARE
- Success with Math (D) $17
- Wbstr Word Game (D) $20

COMMODORE
- Simons Basic (R) .... Call
- Magic Desk (R) $48
- Logo (D) $50
- Assembler 64 (D) $36
- Easy Script 64 (D) ... $52

CONTINENTAL S.W.
- Home Accountant (D) $48
- Tax Advantage (D) ... $45
- FCM (D) $34

COUNTER POINT SW
- Early Games (Ea) $20

CREATIVE SOFTWARE
- Moondust (R) $23
- Save New York (R) ... $23
- Pipes (R) $23
- Househld Finance (C) $20

DATASOFT
- Pooyan (C/D) $20
- Moon Shuttle (D) $20

ENTECH
- Studio 64 (C/D) $28
- Database 64 (D) $45

EPYX
- Temple of APS (C/D) .. $27
- Jumpman (C/D) $27
- Dragonriders (C/D) ... $27
- Gateway to APS (R) ... $27

FUTURE HOUSE
- Comp. Pers. Account. $56

HES
- Omniwriter (D) $45
- Hesmon (D) $27
- 64 Forth (R) $39
- Multiplan (D) $65
- Turtle Graphics (R) ... $39

INFOCOM
- Planet Fall (D) $34
- Enchanter (D) $34
- Infidel (D) $34

JINSAM
- Mini Jini (R) $75

MICROSOFTWARE INT'L
- Practicalc 64 . (C) 34 (D) 36
- Spreadsheet . (C) 49 (D) 52
- Practifile 64 (D) $36

MIRAGE CONCEPTS
- Data Base (D) $68
- Word Processor (D) .. $68

MICROSPEC
- Database (D) $44
- Mailing List (D) $32
- Checkbook Mgr (D) ... $47
- G/L (D) $44
- A/P (D) $47
- A/R (D) $47

PARKER BROS
- Q-Bert (R) $33
- Popeye (R) $33
- Frogger (R) $33

PROFESSIONAL SW
- Wordpro W/Spell (D) .. $68
- Spellright (D) $45

QUIKTEX
- Quick Br. Fox (R) $49

RAINBOW
- File Assistant (D) $46
- Writers Assistant (D) . $46
- Spreadsht Assist. (D) . $56

SCARBOROUGH
- Mastertype (D/R) $27
- Song Writer (D) $27

SOFTSMITH
- Touch Typing (C/D) ... $21

SOUTHERN SOLUTIONS
- Businessman (D) $48
- Bill Payer (D) $48
- Bill Collector (D) $48
- Paymaster (D) $48

SPINNAKER
- Snooper 1 or 2 (D) $27
- Aerobics (D) $34
- Kids on Keys (D) $20
- Most Amazing (D) $27
- Kindercomp (D) $20
- Alphabet Zoo (D) 20
- Trains (D) $27
- Delta Drawing (R) $27

SUBLOGIC
- Flight Simulator II (D) . $36
- Pinball (C/D) $22

SYNAPSE
- Ft. Apocalypse (C/D) . $23
- Necromancer (C/D) ... $23
- Zaxxon (D) $23
- Blue Max (D) $23
- Pharoah's Curse (C/D) $23

SSI
- Combat Leader (C/D) . $27
- Computer Baseball (D) $27
- Eagles (D) $27
- Ringside Seat (D) $27
- Tigers In Snow (C/D) .. $27
- Battle Normandy (C/D) $27

TIMEWORKS
- Dungeons of Alg. (C/D) $17
- Robbers Lost (C/D) ... $17
- Money Mgr. (C/D) $17
- Wall Street (C/D) $17
- Data Manager (C/D) .. $17
- Elec. Checkbook (C/D) $17

TOTL
- Totl Text .. (C) $32 (D) $34
- Label ..... (C) $15 (D) $17
- Time Mgr. .. (C) $24 (D) $27
- Rsrch Asst. (C) $24 (D) $27

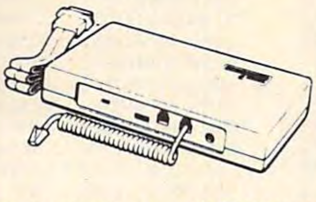
TRONIX
- S.A.M. (D) $39
- Juice (D) $23
- Chatterbee (D) $27

WE TAKE THE BYTE OUT OF HIGH COMPUTER SOFT & HARDWARE$!

CALL 1-800-372-0214 AND SAVE!

CBS SOFTWARE · ATARI · IBM · SHARP · COMMODORE · apple · SANYO · COLECO · AND MANY, MANY MORE!

MODEMS

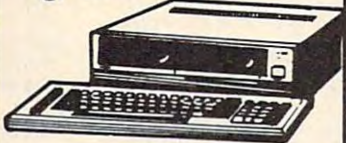


Auto Modem (Commodore) $69.95

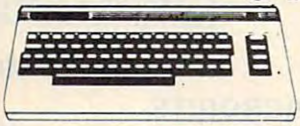
Anchor Mark VI	$159.00
Anchor Mark XII	$289.00
Hayes Smart Com 11	$ 75.00
Hayes 300	$209.00
Anchor Mark VII	$119.00
Anchor Volksmodem	$ 59.00
MPP 1000C (Atari)	$109.95

COMPUTERS

SANYO



COMMODORE 64



Commodore 64 CALL

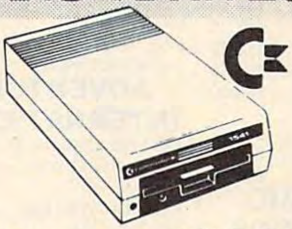
Atari 800XL	NEW LOW PRICE
SANYO 550, 550-2	CALL
SANYO 555, 555-2	CALL

SOFTWARE

Zork I, II or III $29.95

Flight Simulator II	$36.95
Super Base 64	$69.95
Paper Clip w/spell	$79.95
Home Accountant	$59.95
Bank Street Writer	$49.95

DISK DRIVES



Commodore 1541 $232.00

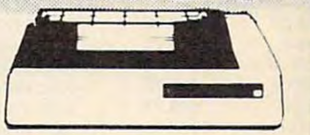
New Commodore Drive	CALL
Atari 1050	NEW LOW PRICE
Rana 1000	$289.00
Trak At-D2	$369.00
Indus GT	$369.00
MSD (Commodore)	$319.00
MSD Dual Drive	$569.00

DISKETTES

Computer Warehouse ss/dd $15.95

Elephant ss/sd	$16.95
Elephant ss/dd	$19.95
Maxell MDI	$24.95
Verbatim ds/dd	$24.95
Ultra Magnetics ss/sd	$15.95

PRINTERS



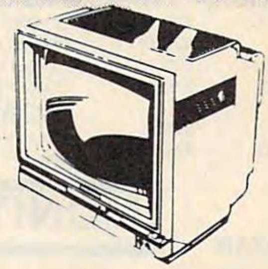
Commodore 801 $209.00

Gemini 10X	$259.00
Gemini 15X	$385.00
Delta 10	$489.00
Power Type	$399.00
AXIOM AT 700 (color)	$499.00

AXIOM AT 550 $259.00

Atari 1025	$229.00
Atari 1027	$249.00
Commodore 1526	$279.00
Okidata 82, 83, 84 92, 93	CALL

MONITORS



Commodore Color $249.00

BMC 13" color	$229.00
Panasonic (composite & RGB)	$329.00
SAKATA 13 color	$235.00
BMC (green)	$ 69.00
BMC (yellow)	$ 99.00

COMPUTER WAREHOUSE

Computers and Software at warehouse prices.

Send for free monthly special sheet.
$3 for software, hardware & educational catalogs

7222 S.W. 117th Avenue
Miami, FL 33183
In Florida call:
1-800-432-0368

FREE Computer Warehouse Diskette with any purchase.

Lyco Computer Marketing & Consultants

TO ORDER CALL US **TOLL FREE 800-233-8760**
In PA 1-717-398-4079

apple

KOALA
- KOALA TABLET $84.75
- PROG. GUIDE $12.75
- GEOMETRIC DES $22.75
- LOGO DESIGN $27.75
- ILLUSTRATOR $99.75
- SONG WRITER D. $27.75
- MASTER TYPE $27.75

CONTINENTAL
- HOME ACCOUNT D $44.75
- TAX ADVANTAGE $35.75
- BOOK OF APPLE SOFTWARE $16.95
- GEN. LEDGER $179.95
- ACCOUNTS PAY $179.95
- ACCOUNTS REC $179.95
- PAYROLL $179.95
- PROP. MGMT $329.95

BRODERBUND
- BANK STREET D $44.75
- CHOPLIFTER D $24.95
- DAVID'D MAGIC $24.96
- DROL D $24.95
- LODE RUN D. $24.95
- SEAFOX D $22.95
- SPARE CHANGE D $24.95
- AE D $24.95

SSI
- KNIGHTS OF DES $26.75
- EAGLES $26.75
- TIGERS IN SNOW $26.75
- GERMANY 1985 .... $36.75
- BATTLE FOR NORMANDY $26.75
- SHATTERED ALL $39.75

ATARI

INFOCOM
- 800XL COMPUTER ...... CALL
- 1050 DRIVE .... CALL
- 1010 RECORDER $55.00
- 1020 PRINTER $59.00
- 1025 PRINTER $189.00
- 1027 PRINTER $249.00
- 1030 MODEM $59.00

KOALA
- SUPER SKETCH $69.95
- KOALA TOUCH TABLET $69.95
- THE ILLUSTRATOR $99.95
- SPIDER EATER $22.50
- SPEEICOPTER $27.75
- PROG. GUIDE $12.75
- GEOMETRIC DESIGN ... $22.50
- CRYSTAL FLOWERS ... $22.50
- KOALA GRAMS SPELLING $27.50
- LOGO DESIGN MASTER $27.50

STRATEGIC SIMULATIONS
- EAGLES $26.75
- THE SHATTERED ALL ... $26.75
- TIGERS IN THE SNOW ... $26.75
- BATTLE FOR NORMANDY $26.75
- KNIGHTS OF DESERT ... $26.75
- COSMIC BALANCE $26.75
- COSMIC BALANCE II ... $26.75
- CYTON MASTERS $26.75
- COMBAT LEADER $26.75

SCARBOROUGH
- SONG WRITER D. $27.75
- MASTER TYPE $27.75

ATARI
- DEADLINE $34.75
- ENCHANTER $34.75
- INFIDEL $34.75
- PLANETFALL $34.75
- STAR CROSS $34.75
- SUSPENDED $34.75
- WITNESS $34.75
- ZORK I $34.75
- ZORK II $34.75

BUSINESS
- VISICALC $159.75
- LETTER PERFECT $89.75
- LETTER PERFECT R ... $89.75
- DATA PERFECT $89.75
- FILE MANAGER $69.75
- HOME FILE MGR $69.75

ADVENTURE INTERNATIONAL
- PREPPIE 16KC/32KD ... $24.85
- RALLY SPEEDWAY ROM) $32.75
- REAR GUARD 16KC $14.95
- SEA DRAGON 16KC/32KD $25.00
- STRATOS 16KC/32KD $25.00
- DISKEY 32KD $34.75

CONTINENTAL
- HOME ACCOUNT D $44.75
- TAX ADVANTAGE $35.75
- BOOK OF ATARI SOFTWARE $16.95

EASTERN HOUSE
- MONKEYWRENCH II ... $52.75

COMMODORE

COMMODORE
- C64 COMPUTER .... CALL
- SX 64 COMPUTER .... CALL
- C1541 DISK DRIVE ... $239.00
- C1526 PRINTER $269.00
- MPS801 PRINTER $215.00
- C1702 MONITOR $249.00
- C1650 AUTO MODEM ... $84.00
- C1530 DATASETTE $65.00
- C64101 ASSEMBLER .. $29.00
- C64105 LOGO 64 ...... $45.00
- C64106 PILOT 64 ...... $35.00
- SIMON'S BASIC $29.00

HES
- 64 FORTH $39.95
- HESMON $22.75
- HESWRITER $28.75
- OMNI WRITER $45.75
- TYPE N WRITER $24.75
- HOME MANAGER $28.75
- TIME MONEY MANAGER $44.75
- OMNI CALC $33.75
- HES MODEM $49.96
- M. MULTIPLAN $65.75

TIMEWORKS
- INVENTORY $39.75
- ACCOUNTS REC $39.75
- ACCOUNTS PAT $39.75
- GEN. LEDGER $39.75
- PAYROLL $39.75
- INVENTORY MAN $39.75
- CASHFLOW $39.75
- DATA MANAGER $19.75
- MONEY MGR. c/d $19.95

KOALA 64
- KOALA TABLET $69.75
- PROG. GUIDE $22.75

CARDCO
- LIGHT PEN $29.75
- 5 SLOT EXPAN. 64 ..... $54.00
- 64 WRITE NOW $39.00
- 64 MAIL NOW $29.00
- 20 WRITE NOW $29.00
- 64 KEYPAD $64.00
- UNIVERSAL CASS. INT. $29.75
- PRINTER UTILITY $19.75
- 6 SLOT EXPAN. $79.96
- 3 SLOT EXPAN. $24.95

SPINNAKER 64
- KINDERCOMP D/R $21.75
- STORY MACH ROM $24.75
- FACE MAKER D/R $24.75
- SNOOPER TR. D $26.75
- SNOOPER T2 D $26.75
- DELTA DRAW ROM $26.75

CONTINENTAL 64
- HOME ACCOUNT D $44.75
- TAX ADVANTAGE $35.75
- F.C.M. $35.75
- 64 USERS OF ENCYC ... $12.75
- 20 ENCYCLOPEDIA $12.75

SCARBOROUGH 64
- SONG WRITER D $27.75
- PHI BETA FILER $32.75
- MASTER TYPE $27.75

BATTERIES INCLUDED
- PAPERCLIP 64 .... $49.95
- PAPERCLIP 64W/SPELLPAK $69.75
- THE CONSULTANT 64 ... $54.95
- BI-80 COL CARD $139.95

MONITORS

TAXAN
- 210 Color RGB $259.00
- 100 Green $115.00
- 105 Amber $125.00
- 400 Color RGB $295.00
- 410 Color RGB $349.00
- 420 Color RGB-IBM .. $459.00
- 121 Green-IBM $145.00
- 122 Amber-IBM $149.00

ZENITH
- ZVM122A Amber ..... $ 95.00
- ZVM123G Green ..... $ 85.00
- ZVM124 Amber-IBM . $129.00
- ZVM131 Color $275.00
- ZVM133 RGB $389.00
- ZVM135 Composite $449.00
- ZVM136 HI RES Color $589.00

GORILLA
- 12" Green .... $ 82.00
- 12" Amber .... $ 89.00

AMDEK
- 300 Green $139.00
- 300 Amber $149.00
- 310 Amber-IBM $159.00
- Color I Plus $259.00
- Color 4T-IBM $589.00

NEC
- JB 1260 Green .... $ 99.00
- JB 1201 Green $145.00
- JB 1205 Amber $145.00
- JC 1215 Color $255.00
- JC 1216 RGB $399.00
- JC 460 Color $349.00

SAKATA
- SC-100 Color $229.00
- STSI Tilt Stand .... $ 35.00
- SG 1000 Green ...... $ 99.00
- SA 1000 Amber $109.00

MODEMS

MITEY MO 79.95

MICROBITS
- MPP1000C $109.00

NOVATION
- J-Cat $89.00
- Cat $129.00
- Smart Cat 103/ $169.00
- Smart Cat 103/212 ... $389.00
- AutoCat $209.00
- 212 Auto Cat $539.00
- Apple Cat II $239.00
- 212 Apple Cat $439.00
- Apple Cat 212 (Upgrade) $249.00
- Smart Cat Plus $359.00

Hayes
- Smartmodem 300 .... $189.00
- Smartmodem 1200 ... $489.00
- Smartmodem 1200b ... $439.00
- Micromodem IIe. $259.00
- Micromodem 100 ..... $289.00
- Chronograph $179.00

ANCHOR
- Volksmodem $55.99
- Mark VII (auto ans/dial) $95.99
- Mark VII (1200 band) $259.00

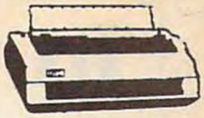
Lyco Computer

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

AMERICA'S MAIL ORDER HEADQUARTERS
LYCO COMPUTER
WORLD'S LEADER IN SALES & SERVICE

TO ORDER CALL TOLL FREE
800-233-8760
or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

SAVE ON THESE IN-STOCK PRINTERS



PRINTER INTERFACING Available for Commodore, VIC, Apple, Atari, IBM-PC, TRS 80 TI, and Others.

APEFACE $59.95
TYMAC CONNECTION ... CALL
AXION .... CALL
MICROBITS .... CALL

BLUE CHIPS
M12010 .... $279.00
M12010 C-64 .... $279.00
D4015 .... $1399.00

Citoh
Gorilla GX100 .... $119.00
Gorilla Serial $129.00
Prowriter 8510 .... $315.00
Prowriter II $575.00
8600 .... $899.00
Starwriter $929.00
Printmaster $1199.00
Sheet feeder $425.00
620 .... $929.00
630 .... $1699.00
8510Sp $499.00
8510SCP $559.00
A10 Letterqual $499.00

PANASONIC
1090 .... $229.00
1091 .... $299.00
1092 .... $439.00

EPSON
RX-80 .... $259.00
RX 80FT $299.00
FX-80 .... $439.00
FX-100 .... $639.00

OKIDATA
80 .... CALL
82A $299.00
83A $569.00
84 .... $959.00
92 .... $399.00
93 .... $685.00

LEGEND
880 .... $259.00
1000 .... $279.00
1200 .... CALL
1500 .... CALL
1081 .... CALL

STAR MICRONICS
GEMINI 15X $359.00
DELTA 10 .... $375.00
DELTA 15 .... $475.00
RADIX 10 .... $509.00
RADIX 15 .... $559.00
POWERTYPE .... CALL
SWEET P (MODEL 100) $549.00
STX 80 .... $149.00

CARDCO
LQ1 .... $449.00
LQ2 .... $279.00
LQ3 .... $339.00
PRINTER INTERFACE ... $39.75
PRINTER INTERFACE W/ FULL GRAPHICS $65.75

GEMINI 10X $249.00

MANNESMANN TALLY
SPIRIT 80 .... $299.00
MTL-160L $559.00
MTL-180L $775.00

NEC
NEC 8025 .... $699.00
NEC 8027 .... $359.00

IBM-PC COMPATABLE

CORONA IBM PC Compatable
PPC22A Portable 256K-Amber ... $1995
PPC22G Portable 256K-Green ... $1995
PPCXTA Portable 256K-10Meg ... $3295
COR128K 128K RAM ... $ 159

NEC
PC8201 Portable $429
NECB1 64K Computer System $1049
NECB2 128 K Computer System $1299
PC8221 Thermal Printer ... $ 139
PC8281 Data Recorder ... $ 89
PC8201 8K RAM Chip ... $ 99
PC8206 32K RAM Cart ... $ 299
PC300 Modem .... $ 65
PC8801 MSDOS 16 Bit Card .... $ 339
PC8025 132 Column Printer .... $ 639
PC8027 80 Column Printer .... $ 349
15LQ Letter Quality ..... $ 479

OVER 2000 SOFT-WARE TITLES IN STOCK

HARD DISK DRIVE

IBM, APPLE TRS80
10 MEG $975.00
20 MEG $1399.00
30 MEG .... CALL
*D.O.S. EXTRA

INNOVATIVE CONCEPTS
DISK STORAGE (10) $4.95
DISK STORAGE (15) $9.95
DISK STORAGE (25) $19.95
DISK STORAGE (50) w/lock $25.75
DISK STORAGE $17.95
ROM STORAGE $24.75

COMPUTER CARE

BIB
DISK DRIVE CLEANER $12.75
COMPUTER CARE KIT ... $19.75

NORTRONICS DISK DRIVE CLEANER with software for IBM-PC, Atari, Vic,
DISK DRIVE CLEANER with software for IBM-PC, Atari, Vic, Apple, TI $29.75
DISK CLEANER REFILL $14.75
CASSDRIVE CLEANER ... $9.95
MEDIA BULK ERASER ... $46.75

PRINTING PAPER
3000 SHEETS FANFOLD $42.75
1000 SHEETS FANFOLD $19.75
1000 SHEET LETTER ... $21.95
200 SHEETS LETTER ... $8.99
150 RAG STATIONARY ... $10.99
MAILING LABELS (1in.) ... $9.95
14 x 11 1000 FANFOLD $24.75

INNOVATIVE CONCEPTS
FLIP-N-FILE 10 .... $3.75
FLIP-N-FILE 15 .... $8.95
FLIP-N-FILE 25 .... $18.95
FLIP-N-FILE 50 .... $17.75
FLIP-N-FILE (ROM HOLDER) $17.75

DISKETTES

SKC
SKC-SSSD $14.75
SKC-SSDD $17.75
SKC-DSDD $21.75

ELEPHANT
5¼"SSSD $15.99
5¼"SSDD $17.99
5¼"DSDD $22.99

MAXELL
5¼"MD-1 .... $19.95
5¼"MD-2 .... $24.99

DISK DRIVES

MSD
SD1 DRIVE $309.00
SD2 DRIVE $499.00

PERCOM
AT88S1 .... $249.00
AT88S1 PD $299.00
ADD-ON DRIVES .... CALL
AT 88 DOUBLER $119.00

RANA 1000 .... $299.00
INDUS GT $325.00

CONCORD
ATARI 176K MASTER $289.00
ATARI 348K MASTER $369.00
ATARI ADD-ON DRIVE .... CALL

TRACK DRIVES
AT D2 .... $329.00
AT-D2 TURBO PAK $22.96
AT-DH .... CALL
PRINTER CABLE $22.95

Lyco Computer

800-233-8760

Lyco Computer
Marketing & Consultants

TO ORDER
CALL TOLL FREE
800-233-8760

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore PA 17740

CUSTOMER SERVICE
1-717-327-1825

POLICY

Risk Free -
No deposit on C.O.D. orders; free shipping on prepaid cash orders within the continental U.S.
APO, FPO, and international orders add $5.00 plus 3% for priority mail service. PA residents add sales tax. Advertized prices show 4% discount for cash; 4% for Master Card or Visa ...

Advertisers Index

COMPUTE!'s
FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166
167	168	169	170	171	172	173	174	175	176	177
178	179	180	181	182	183	184	185	186	187	188
189	190	191	192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221
222	223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242	243
244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265
266	267	268	269	270	271	272	273	274	275	276
277	278	279	280	281	282	283	284	285	286	287
288	289	290	291	292	293	294	295	296	297	298
299	300	301	302	303	304	305	306	307	308	309
310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331
332	333	334	335	336	337	338	339	340	341	342
343	344	345	346	347	348	349	350			

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please print or type your full name and address. Limit one card per person.

Name ______________________________

Address ______________________________

City ______________________________

State/Province ______________ Zip ______________

Country ______________________________

Please include zip code. Expiration 1/31/84. C0184

Place
Postage
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089