Price Quotes, Order Inquires, Information & CA Residents
(714) 824-5555

TOLL FREE **ORDERS ONLY**
(800) 652-8391

PACKAGE DEALS!

No. 1 400/800 PRINTER SPECIAL
NEC 8023-AC Printer
Microbits MPP 1100 Printer
Interface with Cable
$539.95
Sorry, No Substitution On Package Deals.

No. 2 **400 Upgrade Package**
Inhome B Keyboard
Intec 48K RAM
$164.95
Sorry, No Substitutions On Package Deals.

INTEC RAM BOARDS
Lifetime Warranty

16K 800	**$ 49.95**
32K 400/800	**$ 59.95**
48K 400	**$ 84.95**
64K 400	**$109.95**

ATARI

800 48K	**$489.95**
1200 64K	**$649.95**

MONITORS

Amdek Color I	**$299.95**
NEC 12'' Hi-Res Green	**$164.95**
Cable 800/1200 to monitor	**$ 29.95**

DUST COVERS

800	**$9.95**
810	**$5.95**
NEC 8023, C. ITOH, Epson MX/FX	**$9.95**

PRINTERS

Epson FX80	**$549.95**
NEC 8023-AC	**$459.95**
C.ITOH 8510 P	**$399.95**
Okidata 92	**$529.95**
Gemini 10	**$329.95**

MODEMS

Microbits M.P.P. 1000 Includes Terminal Software	**$169.95**
Hayes Smart 300 Baud	**$199.95**

BLANK MEDIA

Elephant Disks 5¼''	**$21.95/10**
Maxell Disks 5¼''	**$25.95/10**

PRINTER RIBBONS

NEC 8023, C. ITOH 8510	**$12.95**	**3/29.95**
Epson MX/FX	**$9.95**	**3/24.95**

MICRO MERCHANT

290 North 10th Street, P.O. Box 1516, Colton, CA 92324

CA Residents add 6% Sales Tax. Credit Card orders add 3%.
UPS Shipping: 3% Hardware, $3 for Software. Shipping Minimum $3.00.
APO/FPO add 5% Shipping, Minimum $5.00.
International orders add 10% Shipping, Minimum $10.00. Sorry, No C.O.D. Orders.

MasterCard VISA

Easy VIC Machine Language Saves

Poul Christensen

The VIC and other Commodore machines allow you to place machine language routines next to your BASIC program. Once you know how, the method is simple and makes your program shorter and easier to load. This method is demonstrated with a simple example and a step-by-step description.

When you write in BASIC on your VIC, you will sometimes find that the resulting program isn't fast enough. With imagination and rewriting you can often make it faster, but sooner or later you may reach the point when only machine language will help.

Where do you place the machine code, and how do you load it in with its BASIC program as a single entity?

Placing And Loading

The most common method is to choose some unused area such as the tape buffers or the memory below the screen image. Of course, you cannot put your coding there directly, so you must write your machine code in DATA statements and include a routine to read your data and POKE the values in place during the program RUN.

If your only problem is speed, this method may work fine. It takes a little longer to load the program, and it takes time to POKE the machine code into the computer, but the main part of your program *will* run faster. But what if you also have memory constraints? You have extra DATA statements and extra code, so you are using up even more memory than before.

Fortunately, there is an easier and better way.

If you look in memory locations 45 and 46, you will find the "start of data" register. This is also the "end of the BASIC program" address. It's the address right after the last BASIC statement. (You can get the decimal number of the address in RAM where your program ends by: ?PEEK(45) + PEEK(46)*256.) When you save your program on tape or disk, *this "register" determines how much you are saving* and, therefore, how much you will load when you read your program in again.

Tricking The VIC

We can make VIC believe that the program extends past the last BASIC statement, and we can use the extra space for a machine program. Although we still have the problem of getting the machine language there in the first place, once it *is* there it will be saved with the program, so it becomes a permanent part of the program. If we add, delete, and change lines, we will change the length of the BASIC program, *but our machine code will stay right where it belongs, next to the last statement.*

A Practice Program

Let's put the theory into practice. This program has no serious purpose, but it serves well as a demonstration. The program simply shows three eight-letter words on the screen and, every three seconds, moves the words around. You'll see why we want to use machine language, and how we go about it.

```
10 PRINT"{CLEAR}{06 DOWN}";TAB(7);"ROTATI
   ON"
20 PRINT TAB(7);"CONFUSES"
30 PRINT TAB(7);"THE MIND"
40 PRINT"{WHT}";TAB(7);"XXXXXXXX{BLK}"
50 TI$="000000"
60 IF TI$<>"000003" THEN 60
70 GOSUB 100
80 GOTO 50
100 FOR I=8 TO 1 STEP -1
110 POKE 7818+66+I,PEEK(7818+I)
120 POKE 7818+I,PEEK(7818+22+I)
130 POKE 7818+22+I,PEEK(7818+44+I)
140 POKE 7818+44+I,PEEK(7818+66+I)
150 NEXT
160 RETURN
```

Cardco, Inc. announces five All-American ways to . . .

Expand your VIC® at affordable prices

card? A

A universal centronics parallel printer interface for the VIC-20 & C-64 computers. Obeys all standard VIC print commands.
Suggested Retail — $79.95

cardette 1

A universal cassette interface for the VIC-20 & C-64 computers. Emulates all functions of the data cassette.
Suggested Retail — $39.95

cardboard 3

The CARDBOARD 3 is a fuse protected, economy expansion interface designed to allow the user to access more than one of the plug-in-type memory or utility cartridges now available. It will accept up to three cartridges at once. This product includes reset button and switches.
Suggested Retail — $39.95

cardriter 1

A light pen for the VIC-20 and C-64 computers with a switch on the barrel and 6 good programs.
Suggested Retail — $39.95

cardboard 6

The CARDBOARD 6 is a fuse protected expansion interface designed to allow the user to access more than one of the plug-in-type memory or utility cartridges now available. Additionally it allows switch selection of games and other programs now available in the cartridge format, without the necessity of turning the computer off and on again, thereby saving a great deal of stress on your VIC-20 and on your television or monitor.
Suggested Retail — $99.95

All Cardco products are **Made in the U.S.A.** and are individually tested to ensure quality and reliability. Superior technological engineering optimizes the value/performance ratio of all of our products.

Specifications and prices subject to change.

Dealer inquiries invited.
United States: Cardco, Inc. • 313 Mathewson • Wichita, KS 67214 • (316) 267-6525
West Canada: LSI Distributing • Attn: Mr. Wong • 2091 W. 61st Avenue • Vancouver, BC. CA V6J 1Z2 • (604) 733-0211
England & Europe: Audiogenic • Martin Manary • 34-36 Crown St. • Reading, Berkshire England • (0734) 595647
East Canada: Hobby Craft Canada • 24 Ronson Drive • Rexdols Ontario M9W 1B4 • (416) 241-2661

When you run the program, you see the characters move. Let's speed up the program by programming the subroutine in machine language:

		Hex			Decimal		
LDX	#8	A2	08		162	8	
LDA	7818,X	BD	8A	1E	189	138	30
STA	7884,X	9D	CC	1E	157	204	30
LDA	7840,X	BD	A0	1E	189	160	30
STA	7818,X	9D	8A	1E	157	138	30
LDA	7862,X	BD	B6	1E	189	182	30
STA	7840,X	9D	A0	1E	157	160	30
LDA	7884,X	BD	CC	1E	189	204	30
STA	7862,X	9D	B6	1E	157	182	30
DEX		CA			202		
BNE	*-25	10	E5		16	229	
RTS		60			96		

We will first see where the program ends, so we PRINT PEEK(45) and PRINT PEEK(46); we should have 44 and 17, which means that the program ends at 17 x 256 + 44 or address 4396 (or hex address 112C). We will add 30 characters to the program, so we POKE 45, 74.

We now have 30 bytes available for the program, so we could start POKEing: POKE 4396,162; POKE 4397,8, etc.

This is not a very easy method, so let's add some lines to the program to read and POKE. But when we add lines, we change the location, so we must recompute the address.

```
1 OC=PEEK(46)*256+PEEK(45)-30
2 FOR I=0 TO 29
4 INPUT Q%
6 POKE OC+I,Q%
8 NEXT
9 STOP
```

Now we run the program, and input the 30 bytes as they are prompted. This little routine is good enough for our purpose, since we want to write only a small program. If you make an error, just start over. But if you have longer programs, you will probably want to add embellishments to your program so you can verify and correct your input.

When the program stops with a "break in 9," your program is in and, you hope, correct (otherwise, you would run the program again). Now is the time to delete all superfluous statements. We must leave line 1, but delete lines 2, 4, 6, 8, 9, and line 100 and on. Finally, change line 70 to:

```
70 SYS OC
```

Instant Changes

Now run the program, and you will see the difference in speed; the screen changes instantaneously.

Stop the program and PRINT OC; you should get 4284, so your program ends at 4314. Not only did we make the program faster, but we also saved 82 bytes.

You can now save the program, and when you load it again you will see that everything, machine language subroutine included, is still intact.

You can, of course, use the same method to place constants at the end of your program. That's useful if you want to write a melody or generate your own character set.

Two Hints

When you expand the program, be sure to allocate enough space – a few extra bytes at the end won't hurt you, and they'll make it much easier for you to change the machine language without having to make more changes in your program. In this example, I would normally expand the program by at least 40 bytes.

Make sure your program is *relocatable*. That means that you should make the program less than 128 bytes long and use branch commands only, not jumps.

If you have more than one machine language routine, you should create a branch table at the start of it and call your routines by SYS OC; SYS OC + 2; SYSOC + 4; etc. This also makes it easier to change your code.

It is easier to place your input routine at the end of your program and use a command like RUN900 to call it. That way you won't inadvertently end up in your input routine when you test your program, and you can leave the routine until the program is correct. Be sure to place a STOP between your program and the input routine.

Finally, let's recapitulate the steps.

1. Write your machine code, and determine how much expansion you need.

2. Print the contents of memory location 45.

3. Add the length of your routine (plus a little bit extra) to the contents of 45, and POKE this value into 45, provided the sum is less than 256.

4. If the sum is 256 or more, subtract 256 from the sum and POKE it into 45; read 46 and POKE a new value (1 higher) into 46.

5. Write an input routine at the end of your program. Make sure you precede the input routine by a STOP command, and that the first instruction computes the location of your expansion area. Also, compute the location of the expansion area in your main program.

6. Input and verify your code.

7. Change your program to call machine code, test the program, and change the machine code if necessary.

8. Delete your input routine and all unnecessary instructions.

9. Save your program on tape or disk. ©

ComputAbility™

Presents Software and Hardware For

commodore

VIC-64

Call for our package price on the VIC-64 System

VIC-20

VIC 1530 Datasette 60.00	VIC 1212 Programmers Aid Cart 45.00
VIC 1541 Disk Drive .... Call	VIC 1213 VICMON Machine Language Monitor 45.00
VIC 1525 VIC Graphic Printer 324.00	VL 102 Introdution To Basic 21.00
VIC 1111 16K Expander ... 94.00	VIC 1600 VIC Modem 92.00
VIC 1011 RS232 Terminal Interface 43.00	Kids & The VIC ... 17.95
VIC 1211 Super Expander ... 53.00	

JOYSTICKS

Starfighter 12.95	Slik Stik 9.95

Software for VIC 20

We carry a complete line of Hardware and Software for the VIC-20, VIC-64 and ATARI. Call for our free catalog.

VIC-20

Avenger	23.95
Super Slot	23.95
Super Alien	23.95
Jupiter Lander	23.95
Midnight Drive	23.95
Radar Rat Race	23.95
Raid on Fort Knox	23.95
Sargon II	28.95
Super Smash	23.95
Cosmic Cruncher	23.95
Gorf	28.95
Omega Race	28.95
Money Wars	23.95
Menagerie	23.95
Cosmic Jailbreak	23.95
Clowns	23.95
Sea Wolf	23.95
Adventureland	28.95
Pirate Cove	28.95
Mission Impossible	28.95
The Count	28.95
Vodoo Castle	28.95
The Sky Is Falling	23.95
Big Speed Math	23.95
Home Babysitter	23.95
Visible Solar System	23.95
Personal Finance	28.95
Tooth Invaders	23.95
Star Post	23.95
Cloud Burst-C	28.95
Satellites & Meteorites-C	34.95
Outworld-C	34.95
Video Vermin-C	34.95
Skibbereen-C	28.95
Grand Master-C	28.95

UMI

Siders of Mars-C	34.95
Meteor Run-C	34.95
Amok-T	18.95
Alien Blitz-T	18.95
Sky Math-T	12.95
Space Division-T	12.95
Super Hangman-T	15.95
Alien-T	18.95
3-D Maze-T	12.95
Kosmic Kamikaze-T	18.95
Sub Chase-T	18.95
Amok-C	28.95
Alien Blitz	28.95
Renaissance-C	34.95

TRONIX

Galactic Blitz-T	18.95
Swarm-T	20.95
Sidewinder	20.95

HES

VIC Trek-T	14.95
Simon-T	12.95
Fuel Pirates-T	12.95
Concentration-T	12.95
Torg-T	14.95
Raid on Islam-T	16.95
HES Games-T	16.95
VIC Forth-C	45.95
Hesmon-C	28.95
Turtle Graphics-C	28.95
Heswriter-C	28.95
Aggressor-C	28.95
Synthesound-C	48.95
Shamus-C	28.95
Protector-C	32.95
Robot Panic-C	28.95
Grid Runner-C	28.95

MISCELLANEOUS

Martian Raider-D	15.95
Shark Trap-D	15.95
Multi-Sound Synthesizer-D	15.95
Resuce at Rigel-T	21.95
Ricochet-T	15.95
Sword of Fargol-T	21.95
Monster Maze-C	28.95
Demon Attack-C	28.95
River Rescue-C	29.95
VIC Music Composer-C	29.95
Crossfire-T	21.95
Fast Eddy-C	29.95
Turmoil-C	29.95
Deadly Duck-C	29.95

800-558-0003

In Wisconsin Call: **414/351-2007**

MasterCard VISA — No Surcharge for Credit Cards

MASTERCARD/ VISA

ORDER HOURS

Mon-Fri 12-9 PM (CST)
Sat 12-5 PM (CST)

ORDERING INFORMATION

To order by mail, send money order, certified check, or personal check (allow 14 days to clear) to COMPUTABILITY. Include $2.00 shipping on all software orders. Include 3% shipping on all hardware orders, minimum $2.50. Mastercard and Visa please include card number and expiration date. WI residents please add 5% sales tax. Outside of Continental USA, please add 15% shipping, minimum $5.00 (US Funds only). Prices subject to change without notice.

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

COMPUTER SOFTWARE ASSOCIATES

"Sensational!"
"Magnificent!"
"Unbelievable!"

PRACTICALC

A comparative newcomer to the software market, PractiCalc has already elicited an overwhelming response. With PractiCalc, features that were once only associated with much larger computers are now available on the VIC-20 (with 16K Ram) and 64. PractiCalc's simple screen format allows easy entry and viewing of data, and its numerous mathematical functions allow for efficient solutions to the most advanced user problems. Easy to operate, PractiCalc offers an affordable alternative to large costly home computers.

- Over 20 Mathematical Functions
- Alpha/Numeric Sorting
- Saves & Stores Spreadsheet
- Available in both disc & cassette versions

$39.95

Order direct by calling 1-800-343-1078

Computer Software Associates products come from around the world and are priced for exceptional value. We offer a complete selection of software to meet your most demanding needs. Ask your local retailer for a catalog of over sixty programs or write us directly for more information.

CSA is searching for programmers who are creating software for Commodore and Timex. We offer a world-wide distribution network as well as generous royalties. Contact us in care of the address below.

Micro Software International, Inc. is the exclusive world-wide distributor of CSA products.

Micro Software International, Inc.
50 Teed Drive
Randolph, MA 02368

Goddard Road
Whitehouse Industrial Estate, Ipswich, IP15 NP
Suffolk, England

DEALER INQUIRIES WELCOME

TI Structured BASIC

Steven M. Ruhl

There has been a debate for years about the merits of "structured programming." In essence, this approach stresses certain rules and conventions which (according to its supporters) result in better, more easily understood program listings and more efficient programming in general. This discussion of structured programming, as applied to the TI-99/4A, should let you decide this issue for yourself.

Structured programming can help some programmers make fewer errors, and can make complex programs easier to modify. Structured programming involves planning and organization so that a program flows logically from one step to another. Some structured programming enthusiasts even outlaw the use of the GOTO statement, since GOTO interrupts the straightforward flow of a program, and may lead to confusing design.

Structured programming also makes liberal use of REM statements, so someone reading a program listing can understand the program's logic easily.

Structured programmers often employ *modular programming* – breaking a program into a series of problems, and solving each separately. Most programs, for example, can be broken down into four parts: initialization, input, processing, and output. Let's look at each of these parts in turn. The highest-level module in a program is the most general, and it controls the modules below it; as the program progresses, each succeeding module performs more specific tasks.

We can use a simple example to illustrate structured programming. Program 1 asks for seven numbers and prints their sum. Program 2 accomplishes the same task, but it does it according to the rules of structured programming. Let's see how it works.

Initialization

In the initialization module, the variables to be used in the program are defined in REM statements, and are initialized or dimensioned if necessary. The REM statements are indented to distinguish them from normal program statements. The blank REM lines separate program modules.

On some computers, variables must be set to zero at the beginning of a program – SUM = 0, for example. The TI-99/4A, however, clears all variables each time a RUN command is entered, so we needn't worry about that phase of initialization.

If you are using array variables, they may need to be DIMed, and the initialization module is the place to do it. DIM statements, which tell the computer how much space to reserve for your array, can be executed only once for each array variable, and must be executed before any other reference is made to the array.

Since we are adding seven numbers, we dimension a seven-element array in lines 180 and 190. When an array is DIMed, the computer sets the lower limit of the array subscript to zero. In other words, DIM N(7) is really an eight-element array composed of the variables N(0), N(1), N(2), N(3), N(4), N(5), N(6), and N(7).

The OPTION BASE 1 statement in line 180 is a feature of TI BASIC that tells the computer to make the lower limit of the array subscript one rather than zero. So, by using OPTION BASE 1, we eliminate the variable N(0) from our list and end up with a seven-element array.

Note that in Program 1, the variable N was not DIMed. In such cases, the TI automatically sets the upper limit of the array subscript to 10. Program 2 would have worked just as well without lines 180 and 190, but we include them to provide the documentation structured programming requires.

Input

Data can be passed to a program in a number of ways, including the INPUT, READ, DATA, and RESTORE statements. TI Extended BASIC offers a few other input possibilities: ACCEPT, SIZE, ERASE ALL, and VALIDATE.

In our example, a simple FOR/NEXT loop of INPUT statements is used to enter the seven numbers to be added. Structured programmers indent the lines within a FOR/NEXT loop to indicate (visually) what is being accomplished within the loop.

Once the INPUT is completed, control passes to the processing module.

Processing

Here again, a simple FOR/NEXT loop is used to add the values of the seven variables. Program 1 includes the processing statement in its INPUT loop, a perfectly valid way of handling the problem. The structured program separates the input and processing functions so that the tasks performed by each can be more easily understood.

Output

The output module takes the result of the processing module and, in this case, prints it on the screen. Output also can be sent to printers, tape, or disk.

Since the purpose of most programs is to provide some kind of computed information, or *output*, many programmers begin their program design with a definition of how that output will appear on the screen or the printer. After the form of the output has been determined, the input module can be tailored to produce the kind of information needed.

In TI BASIC, for example, the colon print separator can be helpful in formatting output.

```
PRINT "HELLO":"THERE"
```

will cause the two words to be printed on separate lines:

```
HELLO
THERE
```

Multiple colons can be used to print blank lines between output. For example,

```
PRINT "HELLO": :"THERE"
```

would insert a line of space between the words when they are printed. The same process can be used in TI Extended BASIC, but spaces must be left between the colons, because Extended BASIC interprets a double colon as a multistatement line.

Easy Modifications

One main purpose of following the rules of structured programming is to achieve clarity and understanding. It may take some rewriting to clear up any rough spots and make the documentation complete. A few months from now, you may want to use a modified version of your program to handle another task.

A clearly documented listing can save you the trouble of relying on your memory when you begin making changes. A program written in modules can also allow you to transfer these "subprograms" to your new program without much modification.

The rules are there for you to follow if you wish. The choice is yours.

Program 1:
Demonstration Program

```
100 REM          ADD SEVEN NUMBERS
110 REM
120      FOR I = 1 TO 7
130           PRINT "ENTER NUMBER ";I
140           INPUT N(I)
150           SUM = SUM + N(I)
160      NEXT I
170      PRINT "SUM OF SEVEN NUMBERS ";SUM
```

Program 2:
Structured Demonstration Program

```
100 REM      INITIALIZATION MODULE
110 REM
120 REM      N(1...7) LIST OF SEVEN NUMBERS
130 REM           TO BE INPUT IN LOOP.
140 REM      SUM IS THE SUM OF THE SEVEN
150 REM           NUMBERS N(1)...N(7).
160 REM      INDEX IS USED TO CONTROL LOOPING
170 REM
180      OPTION BASE 1
190      DIM N(7)
200 REM
210 REM      INPUT MODULE
220 REM
230      FOR INDEX = 1 TO 7
240           PRINT "ENTER NUMBER ";INDEX;
250           INPUT N(INDEX)
260      NEXT INDEX
270 REM
280 REM       PROCESSING MODULE
290 REM
300      FOR INDEX = 1 TO 7
310           SUM = SUM+N(INDEX)
320      NEXT INDEX
330 REM
340 REM      OUTPUT MODULE
350 REM
360      PRINT "SUM = ";SUM
370 END
```

©

Atari Formats

Sheldon Leemon

These programs are an excellent demonstration of Atari's deferred editing capabilities. They demonstrate a number of features which support neatly formatted output in Atari BASIC.

By this time, Atari owners probably know how easy and convenient Atari's superb editing features are. You can position the cursor anywhere on the screen, insert spaces and lines, or delete spaces and lines at will. But some people may not be familiar with the use of these same functions when they are to be executed within the course of the program itself.

Deferred Edit

A number of interesting effects can be achieved if you remember that any edit function that can be used in direct mode could also be used in deferred mode, with the aid of the Escape key. When the Escape key is pressed and then an edit command entered, the command is not immediately executed. Instead, an edit character, usually an arrow or a wedge, appears on the screen. Like any other ATASCII character, it can be put into a string, or used in a PRINT statement. When used in a PRINT statement, however, the edit character will not appear on the screen. Instead, the edit function represented by the character will be executed. For example, when you press Escape and then the Control and Clear keys, a crooked arrow (↰) appears. Whenever that symbol appears in a PRINT statement (e.g., 10 PRINT "↰ Where did they go?"), it clears the screen.

This feature gives us an easy means of formatting output. Program 1 shows how this approach may be applied. The example involves the creation of a table containing three columns. In the first column, there is a number from 0 to 15 (X), that number divided by 32 (X/32), and the sin of the quantity Pi times the number divided by 8 (SIN(PI*X/8)). Negative numbers are accommodated, and trailing zeros inserted to produce a uniform appearance. (This all builds on an earlier **COMPUTE!** article, "Formatted Output For Atari BASIC," March 1981.)

The Quick Way

Program 1 takes a direct approach to solving this problem. Because this program uses many edit characters, REMarks remind you of the sequence of keystrokes needed to produce these characters. But you should still reread Chapter 3 and Appendix F of the *Atari BASIC Reference Manual* to completely familiarize yourself with Atari editing.

Line 10 uses the symbols to clear the screen and move the cursor down one line. Notice how much easier it is to skip several lines of print by using down-arrows than to keep typing in PRINT:PRINT.

Line 20 sets up a string (TC$) that when printed will clear the tab. While not strictly necessary, this is done to show how several edit characters can be repeatedly executed by first putting them into a string, and then printing the string. Here, the tab has five default settings, so we Tab(CHR$(127)) and then Clear Tab(CHR$(158)) five times. This way, if the tab has to be changed later in the program, all we have to do to clear the tabs is print TC$ and set the new tab stops.

Line 30 prints TC$, which clears the tab stops, and then prints the Set Tab Character (CHR$(159)) at columns 7, 13, and 22.

Line 40 prints the headings. Note that we can use the tab characters to print all of the column headings using only one PRINT statement.

Line 50 sets up the FOR/NEXT loop and tabs to column 7.

Line 60 inserts a space if X is less than 10, so that the single-digit numbers line up at the right of the column.

Line 70 prints X and then tabs to column 13.

Line 80 rounds X/32 to three decimal places and prints the result.

Line 90 PEEKs memory location 85, which contains the column number of the present cursor location. This tells us where the cursor is located after print X/32. If it stops at column 14, we know that X/32 is an integer, and a decimal point is printed.

Line 100 uses the same technique to print trailing zeros until the cursor gets to column 18.

Line 110 tabs to column 22.

Line 120 moves the cursor left one space to accommodate a minus sign if the output is negative.

Line 130 prints SIN(PI*X/8) rounded to seven decimal places. To enter this line more easily,

ComputAbility™ presents the Newest in Software Products for ATARI

ATARI 800 48K	**CALL**
ATARI 810 DISK DRIVE	CALL
ATARI 850 INTERFACE	169.00
ATARI 410 PROGRAM RECORDER	77.00
ATARI 1010 PROGRAM RECORDER	77.00
EDUCATOR KIT	117.00
BOOKKEEPER KIT	179.00
PROGRAMMER KIT	55.00

We Carry The Complete PERCOM Line. Call For Items and Prices.

ENTERTAINER KIT	75.00
COMMUNICATOR KIT	CALL
HOME MANAGER'S KIT	CALL
EPSON FX-80	CALL

MOSAIC

64K RAM	149.00
32K RAM	87.95
48K RAM	CALL
16/32 EXPANDER	74.95

B-KEY 400 .... 84.95

SUNCOM JOYSENSOR 34.95

Touch Sensitive Joystick

Call For Our Special EPSON Printer Package

SUPER SPECIALS

Prices Effective June 1, Through June 30, 1983

Miner 2049'er	$33.95	Cartridge
Temple of Apshai	$26.95	Disk/Tape
Ali Baba	$21.95	Disk
Preppie	$19.95	Disk/Tape
Castle Wolfenstein	$19.95	Disk
Upper Reaches Of Apshai	$14.95	Disk/Tape

FREE* SOFTWARE AND ACCESSORIES

*** COUPON PROGRAM**

The purchase of each program (with the exception of Super Specials, Atari, and APX) will earn you 1 COMPUTABILITY DIVIDEND COUPON (CDC). Save 10 coupons and redeem them for your choice of any program we sell for $24.00 or less. Less than 10 coupons may be redeemed for premium items as indicated. You pay only a $2.50 shipping & handling charge.

ATARI

Conversational Languages - T	43.95
Programming 2 & 3 - T	21.95
Music Composer - C	32.95
My First Alphabet - D	26.95
Touch Typing - T	19.95
Home Filing Manager - D	37.95
Mailing List - T	19.95
Asteroids - C	26.95
Caverns Of Mars - D	28.95
Computer Chess - C	26.95
Missle Command - C	26.95
Super Breakout - C	26.95
Star Raiders - C	32.95
Assembly Editor - C	44.95
Basic - C	44.95
Macroassembler - D	65.95
Microsoft Basic - D	65.95
Pilot (Home Package) - C	58.95
Invitation To Programming I - T	18.95
Speed Reading - T	55.95
Basketball - C	26.95
Graph-It - T	15.95
Juggles House - D/T	22.95
Pilot (Educator) - C	97.95
Video Easel - C	26.95
Defender - C	32.95
Galaxian - C	32.95
Qix - C	32.95
Dig Dug - C	32.95
ET - C	38.95
Timewise - D	23.95
Atariwriter - C	61.95

ADVENTURE INTERNATIONAL

Preppie - D/T	23.95
S.A.G.A Adventures - D	31.95
Sea Dragon - D/T	27.95
Stratos - D/T	27.95
Bugoff - D/T	23.95
Preppie II - D/C	27.95
Stone of Sisyphus - D	27.95
Eliminator - D/T	20.95

COMPUTABILITY DIVIDEND COUPON PREMIUMS

3 M Blank Disk - Box of 10	27.00 or 10 CDC
Wabash Blank Disks - Box of 10	21.00 or 8 CDC
Flip 'n File Disk Holder	20.95 or 6 CDC
Flip 'n File Cartridge Holder	20.95 or 6 CDC
Monitor Stand	27.95 or 10 CDC
VU - Case Disk Holder W/Lock	24.00 or 10 CDC
Heavy Plastic Dust Cover 800 or 810	12.95 or 5 CDC
First Book of Atari Software 1983	17.95 or 6 CDC
Kids & The Atari	17.95 or 6 CDC
The Atari Assembler	12.95 or 5 CDC
Atari Games & Recreation	14.95 or 6 CDC
Atari Pilot for Beginners	14.95 or 6 CDC
The Visicalc Book Atari	14.95 or 6 CDC
Atari Basic	12.95 or 5 CDC
Atari Sound & Graphics	9.95 or 3 CDC
Starfighter	12.99 or 5 CDC
Slik Stik	9.95 or 3 CDC
Extension Cable - 6 ft	6.95 or 3 CDC
Lefty Adaptor	9.95 or 3 CDC
Stik Stand	6.99 or 3 CDC
Suncom Tack II	19.95 or 8 CDC
Suncom Game Switch	6.95 or 3 CDC
TI Adaptor	12.95 or 5 CDC
Kraft Switch Hitter Joystick	17.95 or 7 CDC
Kraft Joystick	15.95 or 6 CDC

ON-LINE

Jawbreaker - D/T	23.95
Luna Leeper - D	23.95
Ultima I - D	31.95
Threshold - D	31.95
Ultima II - D	44.95
Mouskattack - D	27.95
Frogger - D/T	27.95

ANALOG

Race In Space - D/T	20.95
Sunday Driver - D/T	23.95
Crash Dive! - D/T	23.95
Buried Bucks - D/T	23.95
Titan - D/T	23.95
Star Sentry - D/T	23.95

DATASOFT

Shooting Arcade - D/T	23.95
Pacific Coast Highway - D/T	23.95
Micropainter - D	27.95
Canyon Climber - D/T	23.95
Fathoms Forty - D	27.95
O'Riley's Mine - D/T	27.95
Rosen's Bridgade - D/T	27.95
Sands of Egypt - D	31.95
Zaxxon - D/T	31.95
Moon Shuttle - D/T	31.95

FIRST STAR

Cosmic Squeeze - D/T	23.95
Astro Chase - D/T	23.95

BIG FIVE

Miner 2049'er - Cart	39.95

I.D.S.I.

Pool 1.5 - D	27.95
Pool 400 - Cart	31.95
Speedway Blast - Cart	31.95
Juggler - D	23.95

SIRIUS

Bandits - D	27.95
Way Out - D	31.95
Repton - D	31.95
Twerps - D	23.95
Wavy Navy - D	27.95
Blade of Blackpoole - D	31.95

SENTIENT

Cyborg - D	27.95
Goldrush - D/T	27.95

BRODERBUND

Apple Panic - D/T	23.95
Stellar Shuttle - D/T	23.95
David's Midnight Magic - D	27.95
Star Blazer - D	25.50
Track Attack - D	23.95
Labyrinth - D/T	23.95
Serpentine - D	27.95
Dueling Digits - D	23.95
Choplifter - D	27.95
Genetic Drift - D/T	23.95
AE - D	27.95

EDU-FUN

Call For Items and Prices

MISCELLANEOUS

Night Mission - D/T	23.95
Jumpman - D	31.95
Airstrike - D	31.95
Castle Wolfenstein - D	23.95
Financial Wizard - D	49.95
Pinball Construction Kit - D	31.95
Pig Pen - D	23.95
Baja Buggies - D/T	25.50
Starbowl Football - D/T	25.50
Master Type - D	31.95
Ali Baba - D	26.50
Jeepers Creepers - D	23.95
Paint - D	33.95
Kid Grid - D/T	23.95

D - Disk T - Cassette C - Cartridge

We Carry Hundreds of Items for ATARI 400/800, Ask for Our FREE CATALOG.

Mastercard/VISA Order Toll Free 800-558-0003

No surcharge for credit cards

In Wisc. Call 414/351-2007

ORDERING INFORMATION: To order by mail, send money order, certified check, or personal check (allow 14 days to clear) to COMPUTABILITY. Include $2.00 shipping on all software orders. Include 3% shipping on all hardware orders, minimum $2.50. Mastercard and Visa please include card number and expiration date. WI residents please add 5% sales tax. Canadian, APO & FPO software orders include 5% shipping minimum $5.00. All other foreign software, please add 15% shipping, minimum $7.00 (US Funds only). Prices subject to change without notice.

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

10,000,000 can be entered using scientific notation (10E6).

Lines 140 and 150 duplicate the function of lines 90 and 100 to fill out the third column of print. Instead of typing in these duplicate lines, use the editing keys to change the line and column numbers of the existing lines 90 and 100, and re-enter them.

Line 160 moves the cursor to the next line and loops back for the next X.

Using Default Settings

In the above example, we didn't really have to go to the trouble of setting the tab. The default settings of the tab are at columns 7, 15, 23, 31, and 39. The default settings of the PRINT(,) statement are at columns 12, 22, and 32. So, using a combination of the two, we could have printed the output at columns 7, 12, and 22, by first pressing the tab, and then the PRINT(,) statement. Moreover, the width of the PRINT columns is adjustable. To change width, we need only POKE location 201 with the new width. If we put the statement POKE 201,5 at the beginning of the program, the PRINT (,) statement would then produce output at columns 7, 12, 17, 22, etc., and there would be no need for us to tab at all to produce the desired format.

Finally, we return to memory location 85. Not only can we PEEK this location to find the cursor, but we can also POKE 85 to position the cursor horizontally. This statement gives us the equivalent of the TAB statement found in other BASICs. To move the cursor to column 7, we could have used a POKE 85,7 with the same result as a TAB command.

The above is offered not as a subroutine of universal applicability, but as an example of the features that the Atari offers for formatting output. The approach shown here works in this particular situation because the output is fairly uniform. When only a little straightening up is needed, a more sophisticated method would be wasted. But this routine will not work with printed output as shown (the printer does not react to screen-editing commands, although it does have its own set of control characters that might be used). And if there is a greater variation in the types of output desired, a more integrated approach would be necessary.

Same Techniques, More Integration

Even in cases where a more organized approach is required, these techniques can be adapted to yield a fairly straightforward routine (Program 2). While basically an adaptation of the March 1981 article, it has the following important differences:

1. Decimal rounding and the addition of trailing zeros are accomplished by string manipulation rather than by mathematics. String manipulation is always faster, and here the difference *is* noticeable.

2. Rather than pack the strings with spaces between the variables to be printed, the POKE 201 and PRINT(,) commands are used as tabs. This allows formatting to take place on both screen and printer, limits the GOSUBs needed, and avoids having to set up the whole line before printing takes place.

3. A routine is added to accommodate numbers that start with -9. The Atari always rounds down, so that -9.5 rounds to -10. This means that INT(N) would have one more digit than the integer part of N does, and this throws off the decimal-rounding routine. The code in line 100 prevents this by adding one to all negative numbers in INT(N) except -1, which would round to 0, thus dropping the minus sign and losing a column space.

4. Commas are added for four-digit numbers. If numbers bigger than seven digits are used, another comma could be added by repeating the routine with the numbers representing the digits substituted accordingly.

Here's a brief explanation. The variables set up at random in line 20 are designed to give a wide range of outputs. Lines 30-60 set up the outputs for each column, with N being the variable to be formatted, ND the number of decimal places, and RC being the *width* of the column, rather than the right column position. This width should allow two or three spaces for print tabbing – if the column width is less than the length of the output, the computer will lock up.

The formatting subroutine starts at line 100, by setting up one string for N, and another for INT(N). Line 110 directs numbers which do not need decimal-rounding around the routine at: line 120, which deletes extra decimal places; line 130, which adds a decimal point to whole numbers; and line 140, which provides trailing zeros. Line 150 adds a comma for numbers with more than four whole digits, and line 160 prints the output, tabbing to the appropriate spot, in order to line up the right-hand columns.

These, then, are some of the exciting features that the Atari computers offer, and they're not only useful for print formatting. The screen editing functions, for example, might be used for simple animation. The graphics capabilities allow you to print alternating lines of regular and reverse video, for easy-to-read tables.

Program 1: Atari Formatting – A Table With Three Columns

```
10 ? "{CLEAR}{DOWN}":REM -[ESC][CTRL
   ][CLEAR];[ESC][CTRL][=]
20 DIM TC$(11):TC$="{TAB}{CLR TAB}
   {TAB}{CLR TAB}{TAB}{CLR TAB}{TAB}
   {CLR TAB}{TAB}{CLR TAB}":REM -[ES
   C][TAB];[ESC][CTRL][TAB];REPEAT 5
    TIMES
```

```
30 ? TC$:? "{6 SPACES}{SET TAB}{6 SPACES}{SET TAB}{9 SPACES}{SET TAB}{8 SPACES}":REM -TABSET=[ESC][SHIFT][TAB];6 SPACES;TABSET;6 SP.;TABSET;9 SP;T.S.
40 ? "{TAB} X{TAB} X/32{TAB}{LEFT}SIN(PI*X/8)":REM - [ESC][TAB];[ESC][CTRL][+]
50 FOR X=Ø TO 15:? "{TAB}";:REM -[ESC] [TAB]
60 IF X<10 THEN ? " ";
70 ? X;"{TAB}";:REM -[ESC] [TAB]
80 ? (INT(X/32*1000))/1000;
90 IF PEEK(85)=14 THEN ? ".";
100 IF PEEK(85)<=17 THEN ? "Ø";:GOTO 100
110 ? "{TAB}";:REM -[ESC] [TAB]
120 IF SIN(4*ATN(1)*X/8)<Ø THEN ? "{LEFT}";:REM -[ESC][CTRL][+]
130 ? (INT(SIN(4*ATN(1)*X/8)*10000000))/10000000;
140 IF PEEK(85)=23 THEN ? ".";
150 IF PEEK(85)<=30 THEN ? "Ø";:GOTO 150
160 ? :NEXT X
```

Program 2:

Atari Formatting – Integrated Approach

```
10 DIM N$(80),I$(80):POKE 82,1:GRAPHICS Ø:POKE 752,1
20 FOR X=1 TO 20:R1=RND(Ø)*100:R2=(-5*(RND(Ø)>Ø.5)+1)*RND(Ø)*100:Z=R1*R2:Z1=R2/R1:Z2=R2-R1
30 N=X:RC=4:ND=Ø:GOSUB 100
40 N=Z:RC=10:ND=Ø:GOSUB 100
50 N=Z1:RC=11:ND=2:GOSUB 100
60 N=Z2:RC=12:ND=3:GOSUB 100
70 ? :NEXT X
80 END
100 N$=STR$(N):I=INT(N):I$=STR$(I+((I<>-1) AND (SGN(N)=-1)))
110 IF ND=Ø THEN N$=I$:GOTO 150
120 IF LEN(N$)>LEN(I$)+ND+1 THEN N$=N$(1,LEN(I$)+ND+1)
130 IF LEN(N$)=LEN(I$) THEN N$(LEN(N$)+1)="."
140 IF LEN(N$)<LEN(I$)+ND+1 THEN N$(LEN(N$)+1)="Ø":GOTO 140
150 IF ((LEN(I$)>3 AND SGN(VAL(I$)>-1)) OR (LEN(I$)>4)) THEN L=LEN(I$)-2:I$=N$(L,LEN(N$)):N$(L)=",":N$(LEN(N$)+1)=I$
160 POKE 201,(RC-LEN(N$)):? ,N$;:RETURN
```
©

Joysticks For The Commodore 64

Michael A Tyborski

The Commodore 64 is, among other things, an excellent game machine. It features advanced graphics, realistic sound, joysticks, and paddles. Learning how to use the joysticks is one of the first steps toward making full use of your 64's entertainment capabilities.

The Commodore/Atari joystick is a very simple device consisting of four switches, as shown in Figure 1. When the control handle is moved, one or two switches close in various combinations. This provides an easy way to detect the joystick's position. There is also an independent switch called the fire button. It can be used to fire lasers, drop bombs, and select options.

How To Read Positions

Each joystick connects to a port on a 6526 "Complex Interface Adapter" (CIA). The back joystick uses port A, and the other uses port B. These ports are at addresses 56320 and 56321 respectively. This makes joystick selection extremely simple.

Since both ports (like those on the VIC) are also part of the keyboard scanning matrix, simultaneous use of the keyboard and joysticks is prevented. Fortunately, this is a minor problem.

The direction switches connect to port bits 0-3, but the ports return a value from 0-255 decimal when read. As a result, you should AND this value with 15 when reading the joystick direction.

Similarly, the fire buttons connect to bit 4 on the ports. To read them, AND the port value with 16. This returns zero when the button is pressed, and 16 otherwise.

For example, you can quickly test the back joystick with this program:

```
10 PRINT PEEK(56320)AND15,-((PEEK(56320)
   AND16)=16)
20 GOTO 10
```

You will read values like those in Figure 2a. Although usable, they are awkward to work with.

A better program would return easy-to-use direction codes. This would require more time, but it would simplify other programming. Figure 2b shows one possible pattern. The sequential values allow an ON-GOTO or ON-GOSUB statement to control program flow.

Joystick Read Subroutine

Now, let's tie all this together. The program below shows the necessary statements to read the 64 joystick. First, we will need to initialize a conversion array. We do this in a short initialization subroutine (lines 9000-9040). The routine also sets up the system constants: PA, JM, and FM.

Variable PA holds the joystick port base address; variables JM and FM are masks for future AND operations. These variables speed up the joystick read subroutine by eliminating floating point conversions. This is important for smooth graphic control.

Figure 1: Joystick Switch Arrangement

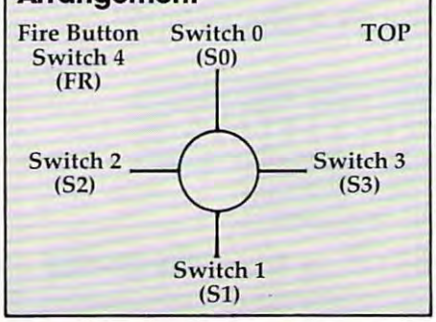
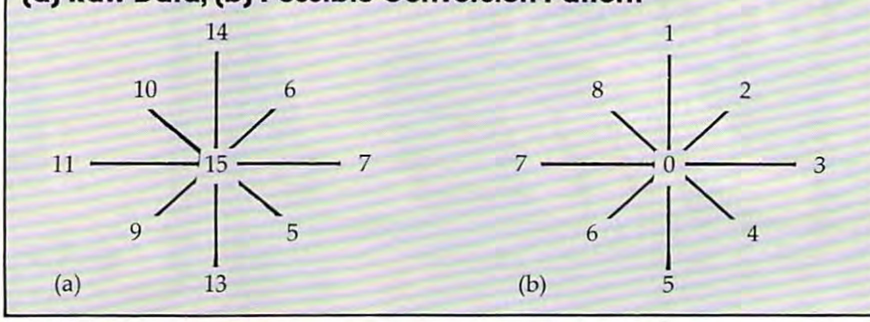


Figure 2: Joystick Direction Values: (a) Raw Data, (b) Possible Conversion Pattern



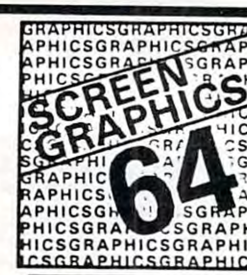
The joystick read subroutine (lines 1000-1040) performs the real work. It reads the status of the joystick selected by variable SN. It then sets variable JV to a direction code as shown in Figure 2b and tests the fire button. If the fire button is pressed, it sets variable FB to one.

The subroutine documentation explains the calling procedure. And more important, it shows which variables are reserved for joystick use.

You should eliminate the REMark statements when using lines 1000-1040. In addition, place them at the beginning of your program and the initialization routine at the end. This will speed things up a bit.

```
1000 REM JOYSTICK READ ROUTINE
1010 SN=SN AND 1:JS=PEEK(PA+SN):JV=JS
     AND JM
1020 FOR JI=1 TO 8:IF JV=JV(JI) THEN
     1040
1030 NEXT:JI=0
1040 JV=JI:FB=-((JS AND FM)=ZR):RETURN
9000 REM INITIALIZATION SUBROUTINE
9010 PA=56320:JM=15:FM=16:ZR=0
9020 FOR JI=1 TO 8:READ JV(JI):NEXT
9030 DATA 14,6,7,5,13,9,11,10
9040 RETURN
```

©



UNNEW
For The VIC And 64

Jim Wilcox

With this utility program on tape or disk, you'll never have to worry again if you type NEW by mistake and lose your BASIC program. Just load in UNNEW, type SYS (850), and there's your program back again. For VIC and 64.

Now and then it happens. You type NEW and then instantly regret it. Fortunately, the VIC and 64 don't erase the program; they only reset some "pointers." The machine language program below will set the pointers back to where they were before your impetuous NEW.

The first step is to type in the program, making sure each of the DATA statements is correct. RUN the program, then type NEW. Now insert a blank tape in the datassette and the statement below without any line numbers:

```
A$="":FORA=833TO938:A$=A$+CHR$(PEEK(A))
:NEXT:SAVEA$
```

When instructed, press PLAY and RECORD on the tape unit. Don't worry about the graphics characters while the SAVE is taking place.

Now to test the program. Type in or LOAD any BASIC program and give the NEW command. LOAD the UNNEW program and type in SYS(850). RUN or LIST the revived BASIC program to verify that it's there.

The routine resides in the cassette buffer. It may be reused (SYS 850) until another program is loaded from tape.

UNNEW For The VIC And 64

```
100 REM  UN-NEW FOR THE VIC & 64
110 FOR A=833 TO 938
120 READ B: POKE A,B: C=C+B: NEXT
130 IF C<>11380 THEN PRINT"CHECK DATA
    STATEMENTS FOR ERRORS":STOP
140 END
200 DATA 85,78,78,69,87,32,83,89
210 DATA 83,40,56,53,48,41,32,32
220 DATA 32,160,3,200,177,43,208,251
230 DATA 200,152,24,101,43,160,255,200
240 DATA 145,43,133,45,200,165,44,145
250 DATA 43,133,46,160,255,200,177,45
260 DATA 72,240,11,200,177,45,133,46
270 DATA 104,133,45,24,144,237,200,177
280 DATA 45,208,243,104,24,165,45,105
290 DATA 2,133,45,144,2,230,46,133
300 DATA 47,133,49,165,46,133,48,133
310 DATA 50,169,0,133,51,133,52,165
320 DATA 55,133,53,165,56,133,54,76
330 DATA 116,196
```

©

INSIGHT: Atari

Bill Wilkinson

This month Bill continues with the creation of the BAIT interpreter (Basic Almost InTerpreter). And he includes some comments from readers.

BAIT: Part 3

For those of you who may have missed Parts 1 and 2, let me give a brief description of this project. BAIT is an acronym for Basic Almost InTerpreter. It is a pseudo-BASIC actually written in Atari BASIC. It is slow. It uses one letter commands (for example, "P" for PRINT). It is simple. And its purpose is simply to give you an inkling of how a BASIC interpreter works. It is *not* a finished, usable language.

This month we will study Part 3 of this listing. We will publish only those lines which have changed from Parts 1 and 2. However, next month we will present Part 4, the last part, and we'll publish the entire listing.

Before starting on my own comments about and additions to BAIT this month, though, I would like to share some reader comments on Part 1.

First, Howard Fishman of Brooklyn, New York, pointed out that I could eliminate the question mark prompt from the INPUT statement by simply using OPEN #3,12,0,"E:" at the beginning of the program and then replacing INPUT with INPUT#3.

Sigh. How right you are, Howard. The funny thing is that I remember discovering this technique about three years ago on our Apple II version of OSS BASIC. How soon we forget. I will incorporate his suggestion in the finished version of BAIT.

Also, Howard protested my not including a facility to list BAIT programs to disk and retrieve them. Perhaps I might change my mind later, but for now I feel that adding that code is an excellent exercise for the reader.

The second letter was from Donald Biresch of Ottsville, Pennsylvania. His comment was that he wished I wouldn't "spend [my] time ... writing about creating BASIC interpreters (something ... less than 1 percent of the end user market has any interest in)." Is he right or wrong? Wrong, I hope, though I admit I have sometimes regretted starting this project, since it has proven to be a larger program than envisioned.

Still, I believe that the subject interests more than 1 percent of you, even if my readers aren't necessarily typical "end users." In particular, I think the BAIT articles are a good lead-in to a more serious study of a BASIC interpreter.

However, if Donald *is* correct, I apologize. Let me know how you feel.

New Features Of BAIT

As with the previous parts, I will describe this month's changes by line number or line number range.

1110. We set all variables to zero.

1515 to 1580. These are simply some line number equates for use as the objects of GOTOs or GOSUBs. Note, though, that they help produce readable code.

3060. Just centralizing some error messages.

4200 to 4250. A complete restructuring of the "Execute Next Statement" routines. Note that multiple statements per line are now legal. Also, note that pushing the START button now serves as a program break (the BREAK key still stops BASIC itself).

4610 to 4620. Sometimes when you generalize things, the program gets simpler. Direct and deferred execution are now virtually identical.

4700 to 4730. After executing a direct line, we wipe it out of the program memory.

4910 to 4960. Look at all the wonderful statements we can now use! They are in order here. Thus a statement "A" will cause DO ACCEPT to be called, etc.

8290. More clean up.

8400 to 8410. Ditto. See line 3060.

10190. Now, we exit from the statement "DO" routines only after getting the character which terminates the statement (that is, the colon or return character).

10250 to 10270. Ditto. Just making PRINT's code cleaner.

10400 to 10420. Look how easy BEGIN (same as BASIC's RUN) is! We zero out the variables, set the current line number to zero, say we found an end of line, and let execute-next-line (at 4600) start the program execution.

10500 to 10530. GOTO is almost as simple. Find what line number the user wants and fool execute-next-line into getting the next execution line from there.

10600 to 10650. LET is only a little more complex. It insists on a variable (10610) for a destination (10620), an equal sign (10630), and an expression (10640). Then it simply gives the destination variable the value of the expression.

10700 to 10730. IF is, I think, a little clever. It simply tells the get-next-statement code (4240 and 4250) that the next character is an end of line if the user's expression evaluates to zero. Otherwise, it does nothing, and the next statement (if any) gets executed.

10800 to 10910. ACCEPT and CALL will be implemented next month.

11000 to 11030. END simply forces an end of line character and an illegal next line number value. The direct statement test (line 4620) effectively ends the program.

11100 to 11410. FETCH, NEW, RETURN, and STORE are left for next month.

Well, there you have it. A functional, albeit minimal interpreter. If you have typed it all in properly, you might try the following program as a test of its logic.

```
1 P "N", : P "N+N", : P "N*N"
2 P N, : P N+N, : P N*N
3 L N=N+1
4 I N<20 : G 2
5 E
B
```

And, for those of you who have not followed BAIT up until now, that translates roughly into BASIC as:

```
1 PRINT "N","N+N","N*N"
2 PRINT N,N+N,N*N
3 LET N=N+1
4 IF N<20 THEN GOTO 2
5 END
RUN
```

And that's enough BAIT for this month. If you don't do anything else while waiting for next month's column, you might try writing the code to execute NEW. It will be *extremely* simple.

BAIT

```
1110 FOR ALPHA=Ø TO 26:VARIABLES(ALPHA)=
     Ø:NEXT ALPHA
1515 DIRECT=4700:BADLINE=8400
1560 DOBEGIN=10400:DOGOTO=10500:DOLET=10
     600:DOIF=10700
1570 DOACCEPT=10800:DOCALL=10900:DOEND=1
     1000:DOFETCH=11100
1580 DONEW=11200:DORETURN=11300:LET DOST
     ORE=11400
3060 GOTO BADLINE
     <<< DELETE LINE 3070 >>>
4200 REM EXECUTE A SINGLE STATEMENT
4230 IF PEEK(53279)<>7 THEN GOSUB DOEND
4240 IF C$=":" THEN 4200
4250 IF C>=Ø THEN GOTO SYNTAX
4610 CURLINE=CURLINE+1
4620 IF CURLINE>Ø AND CURLINE<=MAXLINE T
     HEN 4000
     <<< DELETE LINE 10280 >>>
4700 REM ===COME HERE ON END OF DIRECT L
     INE EXECUTE===
4710 IF LINES(Ø) THEN BUFFER$(INT(LINES(
     Ø)/1000))="*"
4720 LINES(Ø)=Ø
4730 GOTO PROMPT
4910 ERR$="BAD STATEMENT NAME"
4920 ON ALPHA GOTO DOACCEPT,DOBEGIN,DOCA
     LL,DODISPLAY,DOEND
4930 ON ALPHA-5 GOTO DOFETCH,DOGOTO,ERRO
     R,DOIF,ERROR,ERROR
4940 ON ALPHA-11 GOTO DOLET,ERROR,DONEW,
     ERROR,DOPRINT
4950 ON ALPHA-16 GOTO ERROR,DORETURN,DOS
     TORE
4960 GOTO ERROR
8290 GOTO DIRECT
8400 REM BAD LINE NUMBER
8410 ERR$="BAD LINE NUMBER":GOTO 8200
10190 GOTO GETNC
10250 IF C$=";"THEN GOTO GETNC
10260 IF C$=","THEN PRINT,:GOTO GETNC
10270 PRINT:RETURN
      <<< DELETE 4630 >>>
      <<< DELETE 4640 >>>
10400 REM ===EXECUTE BEGIN===
10410 FOR ALPHA=Ø TO 26:VARIABLES(ALPHA)
      =Ø:NEXT ALPHA
10420 CURLINE=Ø:C=-1:RETURN
10500 REM ===EXECUTE GOTO===
10510 GOSUB EXEXP
10520 IF LINES(EVAL)=ØTHEN ERR$="NO SUCH
       LINE":GOTO 8200
10530 CURLINE=EVAL-1:RETURN
10600 REM ===EXECUTE LET===
10610 GOSUB GETNC:IF NOT ALPHA THEN GOTO
       SYNTAX
10620 DESTVAR=ALPHA
10630 GOSUB GETNC:IF C$<>"=" THEN GOTO S
      YNTAX
10640 GOSUB EXEXP:VARIABLES(DESTVAR)=EVA
      L
10650 RETURN
10700 REM ===EXECUTE IF===
10710 GOSUB EXEXP
10720 IF NOT EVAL THEN C=-1:C$=""
10730 RETURN
10800 REM ===EXECUTE ACCEPT===
10900 REM ===EXECUTE CALL===
10910 GOTO ERROR
11000 REM ===EXECUTE END===
11010 PRINT"===END AT LINE";CURLINE;"===
      "
11020 C=-1:CURLINE=C:C$=""
11030 RETURN
11100 REM ===EXECUTE FETCH===
11200 REM ===EXECUTE NEW===
11300 REM ===EXECUTE RETURN===
11400 REM EXECUTE STORE===
11410 GOTO ERROR
```

©

Machine Language Saver

John O. Battle

Here is an easy way to save machine language programs to tape or disk from your VIC-20 or Commodore 64.

You've just written the ultimate character movement routine for your latest video game, and, of course, it's written in machine language for speed. Now you want to save it for future use. (You certainly don't want to type the routine in and debug it again!) But how do you get it onto tape or disk? The BASIC command SAVE works only for programs written in BASIC. You could load in a machine language monitor program and use its SAVE feature, but suppose you don't have a monitor, or that loading the monitor would overwrite the routine you want to save.

Here's the solution. ML Saver is a BASIC program which loads in a short machine language routine of its own. This routine allows you to easily save other machine language programs to tape or disk. And, since it is in machine language itself, it is very fast.

To use the program, simply type in and RUN the BASIC program. Since the numbers in the DATA statement in lines 1000-1300 make up a machine language program, they must be typed in *exactly*, no errors allowed. The program is self-prompting – simply press the letter T (for save to tape) or D (for disk) when asked. Then enter the beginning address for the save and press RETURN.

The program will next ask for the final address in the block of memory to be saved. If you press RETURN without entering an ending address, the program will ask instead for the total number of bytes you wish to save (beginning with the byte at the starting address). If your final address is not greater than your starting address, you will be asked to enter both addresses again.

Finally, the program will allow you to specify a filename for the saved program. This name can be no more than ten characters long.

In order to LOAD a machine language routine that was put on tape or disk by ML Saver, you use the standard BASIC command LOAD, but you must follow the device number with a comma and a one. For example:

TAPE	LOAD "filename",1,1
DISK	LOAD "filename",8,1

The one at the end of the LOAD command tells the computer to load the routine into the same memory locations from which it was saved. Without it, the auto-relocating feature of the VIC and 64 LOAD command would cause the routine to be stored beginning at the normal start-of-BASIC location.

```
40 POKE 52,29:POKE 56,29: REM USE THIS LI
   NE FOR THE VIC-20 ONLY
50 PRINT "{CLEAR}{09 DOWN}{RIGHT}{REV}MAC
   HINE LANGUAGE SAVE{REV}"
60 FOR I=7424 TO 7489
70 READ X
80 POKE I,X :NEXT I
90 FOR I=1 TO 3000:NEXT I
100 PRINT"{CLEAR}{10 DOWN}{06 RIGHT}"
110 PRINT "{REV}T{OFF}APE OR {REV}D{OFF}IS
    K"
120 GET D$:IF D$="" THEN 120
130 IF D$="T" THEN PRINT"{UP}TAPE SELECTED
    ":LF=1:DN=1:SA=2
140 IF D$="D" THEN PRINT"{UP}DISK SELECTED
    ":LF=15:DN=8:SA=15
150 IF D$<>"T" AND D$<>"D" THEN PRINT"{UP}
    ":GOTO 120
160 POKE 7661,LF
170 POKE 7662,DN
180 POKE 7663,SA
200 PRINT"{DOWN}STARTING ADDRESS FOR":INPU
    T"SAVE";S
210 S1=INT(S/256)
220 S2=S-S1*256
230 POKE 251,S2
240 POKE 252,S1
245 A$=""
250 PRINT"{DOWN}FINAL ADDRESS OF":INPUT"SA
    VE";A$
260 IF A$="" THEN 300
270 F=VAL(A$)
280 GOTO 320
300 PRINT "{02 UP}NUMBER OF BYTES TO BE":I
    NPUT"SAVED";N
310 F=S+N-1
320 F1=INT(F/256)
330 F2=F-F1*256
335 IF F<S THEN PRINT"{07 UP}":GOTO 200
340 POKE 7659,F2
350 POKE 7660,F1
400 INPUT"{DOWN}PROGRAM NAME";N$
410 NL=LEN(N$)
420 IF NL<10 THEN 460
430 PRINT"{DOWN}NAME TOO LONG"
440 GOTO 400
460 POKE 7648,NL
470 FOR I=1 TO NL
480 POKE 7648+I,ASC(MID$(N$,I,1))
490 NEXT I
500 IF D$="D" THEN PRINT "{DOWN}PRESS ANY ~
    KEY TO SAVE"
505 IF D$="T" THEN PRINT"{DOWN}REWIND TAPE
     AND PRESS ANY KEY{DOWN}"
510 GET A$
```

```
520 IF A$="" THEN 510
530 SYS 7472
560 END
1000 DATA 169,192,32,144,255,173,237,29,174
    ,238,29,172,239,29,32,186,255,173

1100 DATA 224,29,162,225,160,29,32,189,255,
    96,234,234,234,234
1200 DATA 169,0,32,144,255,96,234,234,234,2
    34,234,234,234,234,234,234
1300 DATA 32,0,29,169,251,174,235,29,172,23
    6,29,32,216,255,32,32,29,0
```
©

Automatic Atari DATA Statements

Richard Dorfner

Use these handy routines to merge machine language programs with BASIC. Never again will you need to manually convert machine language to DATA statements. Your Atari, using the auto-return mode, will do it for you.

Writing programs in BASIC is fine if you don't need great speed. But if you must have a subroutine that operates very quickly, then BASIC is not the answer. One solution is to whip out your assembler and write a machine language subroutine to get the job done. If you want to incorporate it into BASIC, you then need to transform all of these hexadecimal digits into a more useful form, namely DATA statements made up of *decimal* digits. This can be tedious. After once staying up into the wee hours of the morning, I decided there had to be a better way. Fortunately, there is.

Storage Technique

Before getting to the solution, we should first look at the storage technique used by the Editor Assembler cartridge. When the ASM saves the object code onto the disk, it first writes six bytes onto the beginning of the file. The first two are simply used to delimit the beginning of the header. The next two bytes determine the starting address. The last two bytes determine the ending address.

The difference between these two numbers represents the number of bytes to be POKEd into place – which presents a problem. What if there are several "chunks" of machine code to be POKEd into place? ASM handles this by writing another set of starting and ending addresses. This takes up four bytes. With this information, we can now begin the construction of a program which writes DATA statements using the object code file.

Actually, the program to create the DATA statements is rather simple. It fills an array with ten numbers at a time, getting the numbers directly from the object code file. The fun part is when it turns this information into a DATA statment.

To do this, we format a line of code on the screen in a manner that will be accepted by the BASIC interpreter. The next step is to write CONT on a line below the DATA statement.

Why? Well, for one thing, we're going to stop the program and send all the information that we wrote on the screen under program control to the BASIC interpreter, so we must somehow start it up again.

To accomplish this minor miracle, we first, under program control, position the cursor above the items on the screen and then type POKE 842,13:STOP. What happens next is that the operating system begins to send what's on the screen to the BASIC interpreter automatically. As soon as the line with DATA on it is reached, the BASIC interpreter takes it in and shoves it into its proper place in the listing. The next thing to be sent to the interpreter is the CONTinue statement. When BASIC sees this, it executes the command because there is no line number in front of it. It's an immediate mode command.

After the CONTinue statement is hit, the program starts executing on the next line of the BASIC program from where it stopped. In Program 1, this is line 280, which shuts off the automatic input feature.

This program will continue running until it runs out of data in the object code file. When this happens, the program dumps out all of the numbers held by DAT(1-10). This is done to avoid losing data, which might occur due to an End Of File error.

After all of the data has been READ in and LISTed to either the disk or cassette, the computer very kindly announces the file name under which the DATA statements were listed. This name will always be "D or C:DATASTAT.LST". It also includes an initial DATA statement which tells POKEIT how many bytes are to be POKEd into place.

Lyco Computer Marketing & Consultants

TO ORDER CALL US — **TOLL FREE 800-233-8760**
In PA 1-717-398-4079

FREE DUST COVER with Purchase of

ATARI 800 48K $489.00
ATARI 400 64K $349.00

FREE CATALOG with over 60 manuf. for ATARI

ATARI 1200 64K RAM...$CALL$

ATARI
A Warner Communications Company

ATARI HARDWARE

810 DISK DRIVE	$419.00
410 RECORDER	$75.00
1010 RECORDER	$75.00
850 INTERFACE	$164.00

PACKAGES

CX482 EDUCATOR	$119.00
CX 483 PROGRAMMER	$54.00
CX488 COMMUNICATOR	$219.00
CX419 BOOKEEPER	$189.00
KX7104 ENTERTAINER	$69.00

SOFTWARE

QIX	$32.75
CXL4012 MISSILE COMMAND	$28.75
CXL4013 ASTEROID	$28.75
CXL4020 CENTIPEDE	$32.75
CXL4022 PACMAN	$32.75
CXL4011 STAR RAIDER	$34.75
CXL4004 BASKETBALL	$26.75
CXL4006 SUPER BREAKOUT	$28.75
CXL4008 SPACE INVADER	$28.75
CX8130 CAVERNS OF MARS	$31.75
CX4108 HANGMAN	$12.75
CX4102 KINGDOM	$12.75
CX4112 STATES & CAPITALS	$12.75
CX4114 EUROPEAN COUNTRIES	$12.75
CX4109 GRAPHIT	$16.75
CX4121 ENERGY CZAR	$12.75
CX4123 SCRAM	$19.75
CX4101 PROGRAMMING I	$19.75
CX4106 PROGRAMMING II	$22.75
CX4117 PROGRAMMING III	$22.75
CXL4015 TELELINK	$21.75
CX4119 FRENCH	$39.75
CX4118 GERMAN	$39.75
CX4120 SPANISH	$39.75
CXL4007 MUSIC COMPOSER	$33.75
CXL4002 ATARI BASIC	$45.75
CX8126 MICROSOFT BASIC	$65.75
CXL4003 ASSEMBLER EDITOR	$45.75
CX8126 MACRO ASSEMBLER	$69.75
CXL4018 PILOT HOME	$65.75
CX405 PILOT EDUCATOR	$99.75
CX415 HOME FILING MANAGER	$41.75
CX414 BOOKKEEPER	$119.75

MONITORS

NEC JB1260	$125.00
NEC JB1201	$155.00
NEC TC1201	$315.00
AMDEK 300G	$159.00
AMDEK COLOR I	$329.00

MODEMS

ANCHOR MARK I	$79.00
ANCHOR MARK II	$79.00
HAYES SMART	$239.00
HAYES MICRO II	$309.00
CAT	$144.00
J-CAT	$ CALL $

64K RAM	$129.75
48K RAM	$ 99.75
128K RAM DISK	$399.75
80 Colum Screen Board	$279.75
400 KEY BOARD	$ 79.75
TECHNICAL NOTES	$29.75

COMPUTER COVERS

800	$6.99
810	$6.99
400	$6.99
410	$6.99

CALL for PRICES on RANA DISK DRIVES MICROMAINFRAME DRIVES

NEW RELEASES

MINER 2049er	$29.75
ZAXXON	$29.75
CHOPLIFTER ROM	$32.75
CRISIS MOUNTAIN	$25.95
WARLOCKS REVENGE	$25.95
HOME ACCOUNTANT	$59.75
BOOK of ATARI SOFTWARE 1983 346 pages	$16.75

BUSINESS SOFTWARE

VISICALC	$159.75
LETTER PERFECT	$115.75
LETTER PERFECT ...ROM	$159.75
DATA PERFECT	$75.75
TEXT WIZZARD	$79.75
SPELL WIZZARD	$64.75
FILE MANAGER 800+	$69.75
ATARI WORD PRO	$109.75
TAX ADVANTAGE	$35.75
HOME ACCOUNTANT	$59.75

WICO JOYSTICKS

COMMAND CONTROL	$23.75
RED BALL	$26.75
TRACK BALL	$52.75
EXTENSION CORD	$9.75

COMPUTER FURNITURE

GUSDORF	$59.75
BUSH CTA120	$69.75
add-on TV shelf	$17.95

POLICY

In-Stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit for COD orders. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash. Add 4% for Mastercard and Visa.

TO ORDER CALL TOLL FREE
800-233-8760
In PA 1-717-398-4079
or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

POKEIT

Program 2, POKEIT, must be loaded from wherever you have it saved. To use it, you also have to ENTER"Device.DATASTAT.LST" which was saved by the other program. After you type RUN, POKEIT begins to read the first item of information. This number that it gets is the number of bytes to be POKEd into place altogether.

It then reads the next six bytes. The first two of these are thrown away. The third and fourth bytes are used to calculate the starting memory location to POKE to. The fifth and sixth bytes are used to calculate the ending address to be POKEd.

The program starts to read and POKE until it runs out of memory locations to POKE to. And then compares the number of bytes it has POKEd to the total number of bytes it is required to POKE. If it has not yet completed its job, it's because there is another "chunk" of code to be placed in a starting and stopping set of addresses and cycles through the program once more. It will continue to do this until it has run out of data, at which point it will automatically jump to the machine language subroutine.

Program Description

Here's a play-by-play report on what Programs 1 and 2 are doing.

Program 1.

Line no.	
60	Gets the device and the file name that it will be working with.
70	Opens that file.
80-100	Initialize the array DAT to 0.
110-150	Are a WHILE loop which gets values for X and puts them in the array DAT(I).
170	Clears the screen.
180	Increments the line counter which will be used to give the DATA statement a line number.
190-250	Format the screen to look like a line of BASIC.
260	Positions the cursor above what was just written on the screen.
270	Turns on the automatic line-entering feature and stops the program so that it can take effect.
280	Turns it off. This is the line which BASIC will return to once it gets the CONTinue statement.
290	Checks to see if the trap flag was set. The only way it can get set is if the machine gets an End Of File error. If TR has been set, then it goes to line 325; otherwise, it goes to line 80.
310	Sets TR to 1 whenever it is reached.
325-370	Put the number of bytes read from the file into the first DATA statement and automatically enters it into the basic program.
380	Changes the file name to DATASTAT.LST.
390	LISTs the DATA statements to the device named in FILE$.
400	Tells the user by what file name the DATA statements were saved.

Program 2.

Line no.	
80	Gets the number of bytes to be read and the first two useless bytes saved from the file.
100 - 120	Calculate the starting and ending address which will be POKEd to.
140-170	Do the actual POKEing of the information held by the DATA statements.
180	Tests if all the information has been read. If it hasn't, then it goes back to line 100 and starts the process all over again.
190	Forces the machine to take the machine language plunge.
300-320	Get the starting address of the machine language subroutine for when the machine decides to go there.

Program 1: DATA Statements From Machine Code

```
10 REM DATA STATEMENTS FROM MACHINE
   CODE FILES
40 DIM DAT(10),FILE$(14)
50 POKE 710,0:POKE 709,14:REM make m
   y B$W T.V. look nice
60 ? CHR$(125):? :? "INPUT DEVICE:FI
   LENAME ";:INPUT FILE$:J=0:LINE=10
   000
70 OPEN #6,12,0,FILE$
80 FOR I=1 TO 10
90 DAT(I)=0:REM Initialize DAT to 0
100 NEXT I
110 I=0
120 I=I+1:J=J+1:TRAP 310:REM Begin t
    he while loop
130 GET #6,X:REM Get a byte
140 DAT(I)=X:REM Put it in DAT(I)
150 IF I=10 THEN GOTO 170:REM Done t
    en yet? If so GOTO 170
160 GOTO 120:REM Otherwise go get th
    e next byte
170 ? CHR$(125):REM Clear the screen
180 LINE=LINE+10:REM Increment the l
    ine counter
190 POSITION 2,4
200 ? LINE;"DATA";:REM Print the lin
    e number and then DATA
210 FOR I=1 TO 9
220 ? DAT(I);",";:REM Follow DATA wi
    th the various data items separa
    ted with commas
230 NEXT I
240 ? DAT(10):REM Finish the DATA st
    atement with a data item but don
    't put in a comma
250 ? :? "CONT":REM Skip a few lines
     then print CONT
260 POSITION 2,0:REM Position the cu
    rsor above it all
270 POKE 842,13:STOP :REM Then turn
    on the auto-entry feature and st
    op the program
280 POKE 842,12:REM When it has ente
    red the data statement, come bac
    k here and turn the feature off
290 IF TR=1 THEN GOTO 325:REM If the
     trap has occured, goto 325
300 GOTO 80:REM Otherwise goto 80
310 TR=1:REM Set the trap flag to 1
320 GOTO 170
325 CLOSE #6:REM close the file sinc
```

```
     e we're all done with it
330 ? CHR$(125):REM then clear the s
     creen
340 POSITION 2,4
350 ? 10005;"DATA";J-1:? :? :? "CONT
     ":REM Print a data statement tel
     ling how many bytes were read
360 POSITION 2,0:POKE 842,13:STOP :R
     EM Now enter it as a line of BAS
     IC
370 POKE 842,12:REM and turn the aut
     o-entry feature off
380 FILE$(3,14)="DATASTAT.LST":REM c
     hange the filename to DATASTA.LS
     T
390 LIST FILE$,10005,LINE:REM Then l
     ist it to the device under the n
     ame of DATASTAT.LST
400 PRINT "FILE LISTED UNDER":PRINT
     FILE$:REM and let the user know
     what to call it
```

Program 2: POKEIT

```
10 REM POKE IT IN PLACE
40 DIM FILE$(15)
50 POKE 709,14:POKE 710,0:REM MAKES
   MY B&W T.V. LOOK NICE
80 READ NUMBEROFBYTES,JUNK1,JUNK2
90 TRAP 190
100 READ STARTLO,STARTHI,LASTLO,LAST
    HI
110 START=STARTLO+256*STARTHI:REM CA
    LCULATE THE STARTING ADDRESS
120 LAST=LASTLO+256*LASTHI:REM CALCU
    LATE THE LAST ADDRESS
130 GOSUB 300:REM IS THIS THE FIRST
    ADDRESS TO BE CALCULATED?
140 FOR I=START TO LAST
150 READ X
160 POKE I,X
170 NEXT I
180 IF (LAST-START)<>NUMBEROFBYTES T
    HEN GOTO 100:REM GOT ALL OF THE
    DATA YET? IF NOT GO GET THE NEXT
     "CHUNK"
190 X=USR(BEGINNING)
200 END
300 IF FLAG<>1 THEN BEGINNING=START:
    REM HAVE WE FOUND A BEGINNING AD
    DRESS ALREADY? IF NOT THIS IS IT
310 FLAG=1:REM SET FLAG TO 1 TO INDI
    CATE WE HAVE ALREADY SET THE ENT
    RY POINT OF THE MACHINE CODE SUB
    ROUTINE
320 RETURN
```

©

Atari Fast Shuffle

James E. Korenthal

What's all this fuss about rearranging a few numbers? I've seen so many articles in various magazines about shuffling numbers in increasingly exotic ways that I'm thoroughly mixed up.

Here's the technique that I've been using for years. It's simple and fast, and it gives you an equal probability of any given permutation showing up. The program is written in Atari BASIC, but will work on, or can easily be converted to, any other BASIC. It's set up to shuffle 52 cards, represented as numbers from 0-51 in an array called DECK (with subscripts running from 0-51).

```
10 REM INITIALIZE
20 N=51:DIM DECK(N):REM SHUFFLE N+1 N
   UMBERS
30 FOR J=0 TO N:DECK(J)=J:NEXT J:REM
   FILL THE DECK
40 REM SHUFFLE THE DECK
50 FOR J=N TO 1 STEP -1:REM LOOP BACK
   WARDS THROUGH DECK
60 K=INT(RND(0)*(J+1)):REM PICK POSIT
   ION TO SWAP
70 TEMP=DECK(J):DECK(J)=DECK(K):DECK(
   K)=TEMP
80 NEXT J:REM AND THAT'S ALL THERE IS
    TO IT!
90 FOR J=0 TO N:PRINT DECK(J);" ";:NE
   XT J
```

You can easily set up the shuffling loop as a one-line subroutine, and then use a GOSUB when it's time to shuffle. Also, as long as you haven't changed the numbers in the array to be shuffled, you don't have to reinitialize (line 30 in the program) before shuffling. ©

VIC Contractor

Peter Lear

Using these two programs, you can make expansion memory invisible to the VIC so unexpanded programs can be run without removing the cartridge. There's also a short program to access the 3K expansion area usually lost when an 8 or 16K cartridge is employed.

Do you have some extra memory for your VIC that you don't always need? Are you tired of yanking out that cartridge and putting it back? Do you fear that your edge connector might eventually wear itself out? Here's the solution. Use these methods to "mask" cartridges from your VIC.

1. *The 3K expansion.* This is easy, because all that happens is that an empty memory block is filled. To forget this memory, just type in the following:

POKE 44,16:POKE 4096,0:NEW

2. *The 8K, 16K or 24K expanded VIC.* This is not as easy because not only has empty memory been filled in, but the VIC's screen has also moved. BASIC now begins at 4608 ($1200), and the screen is where BASIC once was, 4096 ($1000). Finally, the color locations for the screen now start at 37888 ($9400). To fix all these alterations so that the VIC is unexpanded, type in Program 1. Save a copy before running it (otherwise you will need to type it again to use it). To get the memory back, just do a cold start with SYS64802.

3. *The 3K in addition to 8K or 16K or 24K expanded VIC.* With this setup, you cannot see the 3K cartridge as BASIC memory. To make this arrangement into a nonexpanded VIC, just use Program 1. The 3K block will still be accessible for machine language. As with the other cartridge arrangements, you also have access to any filled-in memory blocks solely for machine language. Some applications require that the 3K block be part of BASIC. Program 2 accomplishes this. Rather than type in the whole thing, just modify Program 1.

For some VIC owners with expansion boards, these programs are not necessary since the boards themselves have switches which do the same thing. But these programs are useful for VIC owners whose boards do not provide such switches and for those without boards.

Note: Super Expander owners should change the following lines of Program 2:

```
130 POKE51,120:POKE52,29:POKE55,120:POKE56
    ,29:POKE646,6
150 PRINT"6519 BYTES FREE"
```

(This allows the function key labels to be protected.)

Program 1: Reset To Unexpanded VIC

```
100 POKE36866,150:POKE648,30:POKE36869,240
110 FORJ=217TO228:POKEJ,158:NEXT
120 FORJ=229TO250:POKEJ,159:NEXT
130 POKE51,0:POKE52,30:POKE55,0:POKE56,30:
    POKE646,6
140 PRINT"{CLEAR}**** CBM BASIC V2 ****"
150 PRINT"3583 BYTES FREE"
155 POKE243,154:POKE244,150:POKE642,16:POK
    E644,30
160 FORI=4096TO5000:POKEI,0:NEXT
170 POKE44,16:POKE46,16:POKE48,16:POKE50,1
    6:NEW
```

Program 2: Reset To VIC With 3K Expansion

```
100 POKE36866,150:POKE648,30:POKE36869,240
110 FORJ=217TO228:POKEJ,158:NEXT
120 FORJ=229TO250:POKEJ,159:NEXT
130 POKE51,0:POKE52,30:POKE55,0:POKE56,30:
    POKE646,6
140 PRINT"{CLEAR}**** CBM BASIC V2 ****"
150 PRINT"6655 BYTES FREE"
155 POKE243,110:POKE244,150:POKE642,4:POKE
    643,120:POKE644,29
160 FORI=1024TO1028:POKEI,0:NEXT
170 POKE44,4:POKE46,4:POKE48,4:POKE50,4
```

©

UMI Software is Making "Home" Work Fun

Wordcraft 20

UMI gives you sophisticated word processing software complete in one package! Wordcraft 20©, with a tutorial tape, contains 8K RAM, a unique automatic mail list feature, and everything else you'll need to create picture-perfect documents. This fully featured system lets you change a character, a word, an entire block of text; and sends encoded electronic mail. With 4-direction scrolling, you see it before you print; and it's compatible with any printer. With Wordcraft 20©, you'll never be at a loss for words again.

Viterm B

A sophisticated communications program that links you and your VIC™ to the world of information, VITERM B is compatible with virtually any modem. Your access to information banks and services over the telephone system is astonishing. At your fingertips, you'll have UPI news and features, information encyclopedias, discount buying services, the stock market and educational programs. And, VITERM B accesses CompuServe, THE SOURCE, and other similar computer services. You'll be able to send and receive personal electronic mail, set up personal finance programs, make travel reservations — all at electronic speed. The world is yours at the touch of a key with UMI's VITERM B.

BUTI

Improve your BASIC program with UMI's BUTI treatment. Adding 17 new commands to the BASIC language in your computer, BUTI formats the VIC™ to imitate 8K, 3K, or minimum memory configurations. BASIC program errors will stop program execution, list and mark the line of BASIC where the error occurred. Other features are single-step execution, renumbering, block search & replace, block line delete, tape append, and BASIC variable dump.

Simple . . . quick . . . and on command. That's the BUTI treatment for your VIC™.

VICEPS — Connects Epson MX100 or MX80 to your VIC20™ • Prints high-resolution graphics and character sets using Epson Graftrax • Does formatted BASIC program listings

VI-CALC — 10 memory registers and 4 stacked data • Registers always visible • Math function results visible at a single keystroke

VI-DATA — Powerful data base program on cassette or disk • User-defined screen format • Print screen format • Format print output • Alphanumeric sort

VI-CHECK — Manages checkbook • Lists accounts • Makes deposits • Keeps balance current • Lists transactions • Catches duplicate entries • Features calculator mode

FORTH 20
Structure of PASCAL or COMAL:
• Speed of machine code — 10 times faster than BASIC • Interactive; both a compiler and an interpreter • Transportable — based on FORTH 79-Standard • A language you tailor to your application by adding new commands • Comes complete with an extensive instuction manual and examples.

umi

United Microware Industries, Inc.
3503-C Temple Avenue
Pomona, CA 91768 (714) 594-1351

See us at CES, McCormick West

Custom Characters On Atari

Charles Delp

Custom character graphics is an easy way to program game animation, but sometimes it results in uneven motion. Smoother animation can be achieved by using custom characters to create the fixed playfield, and then using player/missile graphics to animate the players. The three programs here show you how.

One of the easiest ways to put colorful, high resolution playfields or special symbols on the screen is with character graphics, employing custom characters. A good example of custom character graphics is demonstrated by the game Gold Miner (**COMPUTE!**, July 1982).

Gold Miner also demonstrates one of the major drawbacks of using character graphics to animate a game: players can move only in large, character-sized jumps. When smoother action is desired, a better solution is to draw the fixed playfields using custom characters and then animate the players using player/missile graphics.

The advantages of using character graphics rather than bit mapped graphics to draw fixed playfields are:

1. Much less memory is required to achieve the same resolution.
2. More colors are available.
3. Less time is required to draw to screen memory.
4. Color fill is faster and easier.

The major disadvantage of using character graphics to draw fixed playfields is that only two colors (character color and background color) are available within any one character. Figure 1 shows the resolution, memory requirements, and colors available for various Atari BASIC character and bit mapped graphics modes.

How Characters Are Defined

Atari characters are defined by 64 pixels arranged in eight columns by eight rows. From right to left, the values of the columns are 1, 2, 4, 8, 16, 32, 64 and 128. If a particular pixel is turned on, the value of that column is added to the row total; if the pixel is turned off, zero is added to the row total. The total value of all the "on" pixels in a row forms a byte of data which defines that row. Each of the eight rows is defined by a byte of data, for a total of eight bytes per character. (See Figure 2 for a specific example.) Note how the row bytes are arranged in memory from the character start address (CHADD).

Character Editor

Program 1 is a character editor utility which will be a help in developing the DATA statements required to define each character. Draw the character using the joystick. Erase errors by holding the trigger button while drawing over the error. Press C (Clear character) at any time to clear the screen for another character. Press D (Demonstrate character) to see the character in all three of the character graphics modes, as well as the DATA statement required to produce the character. Press P (Print data) to print a hard copy of the character DATA statement. Press E (Enter data) to enter the character data as a program line beginning at line 9000. When all characters have been entered, typing LIST"D:CHAR",9000,9999 will save the data to disk or LIST"C",9000,9999, to cassette. The data may be merged into your graphics program using the ENTER command (see chapter 5, *Atari Basic Reference Manual*).

Locating The Custom Character Set In Memory

First, look at the memory map in Figure 3. The standard Atari character set is located in ROM beginning at address 57344 (CHORG). The location and size of screen memory including the display list will depend on how much RAM is installed in your computer and which graphics mode is called by your program. The new character set must be defined and stored in RAM in a location which does not interfere with screen memory, the display list, the player/missile display memory, or the BASIC program. The proce-

COMPUTE!'s First Book Of Atari Graphics

Authors: **COMPUTE!** Magazine editors and contributors
Price: $12.95
On Sale: Now

COMPUTE!, the leading magazine of home, educational, and recreational computing, has led the way for Atari owners since the computers were first introduced in 1979. **COMPUTE!** has published scores of articles on Atari graphics, and was the first to divulge many important details on such techniques as redefined characters, custom graphics modes, and player/missile graphics. But those articles are scattered across dozens of issues, many of which are scarce or out of print.

That's why the editors of **COMPUTE!** decided to gather the very best Atari graphics articles published over the past three years into *COMPUTE!'s First Book Of Atari Graphics*. From the fundamentals to advanced techniques, here are some of the most instructive articles ever published for the Atari.

But that's not all. *COMPUTE!'s First Book Of Atari Graphics* also presents articles never before published anywhere, and additional sections written especially for this book. These include "The Basics Of Atari Graphics," an introductory tutorial which prepares beginners for the rest of the book; "How To Design Custom Graphics Modes," which covers the fundamentals of mixing modes on a single screen; and "Introduction To Player/Missile Graphics," a guide to understanding one of the Atari's most advanced features, written by Bill Wilkinson, a **COMPUTE!** columnist and a creator of Atari BASIC and the Atari Disk Operating System.

Numerous other articles include "Designing Your Own Character Sets," a new and improved "SuperFont," "High Speed Animation With Character Graphics," "Animation And Player/Missile Graphics," "The Collision Registers," and "GRAPHICS 8 In Four Colors Using Artifacts." There's even a brand new article by Wilkinson, "The Priority Registers," which for the first time shows how to use player/missile graphics to create a *fifth* player.

In the **COMPUTE!** tradition, *Atari Graphics* is crisply written and edited to be useful to beginners and experts alike. And it's spiral-bound for easy access to its dozens of ready-to-type program listings.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2 shipping and handling. Outside the U.S. add $5 for air mail, $2 for surface mail. All orders prepaid, U.S. funds only.

dure described below and illustrated in Figure 3 will keep everything nicely separated.

1. Find MEMTOP on your computer by entering the following line: PRINT PEEK (106) * 256.

2. Decide whether your program using the custom characters will be written in graphics mode 0, 1, or 2. For your information, the bottom of screen memory, including display list, will be located at MEMTOP - X
where X=992 for graphics mode 0
X=674 for graphics mode 1
X=424 for graphics mode 2

3. The starting address of the custom character set, CHBASE, must be located on a 1K memory boundary, so CHBASE should start 2K below MEMTOP (1K for screen memory, 1K for the character set); therefore, CHBASE = MEMTOP - 2048.

4. If player/missile graphics are to be used, PMBASE must be located on a 2K boundary (for single line resolution), so P/M Base should start 4K below MEMTOP (1K for screen memory, 1K for the character set, 2K for P/M Display Area); therefore, PM BASE = MEMTOP - 4096.

Developing A Custom Character Set

Normally a character set consists of 128 different characters in graphics mode 0, and 64 different characters in graphics modes 1 and 2. However, a character set need not be full and may contain only as many characters as needed to meet the requirements of your program. The first character in the set must always be a space (DATA statement filled with zeros).

Program 2 demonstrates how to set up and use a custom character set containing only custom characters. To keep things simple, the set contains only eight characters.

Lines 10-30 Initialize and find CHBASE

Lines 50-70 Clear space in memory for the custom character set

Lines 90-130 POKE the new characters into memory beginning at CHBASE

Lines 200-280 Contain the character data

Lines 300-360 Print the characters on the screen

The simplest way to print the custom characters to the screen is with the PRINT #6 statement; however, the custom characters are not shown on the keyboard, so the following correlation must be performed:

Note: Refer to Table 9.6 – The Internal Character Set, in the *Atari BASIC Reference Manual,* page 55.

1. Assign a character number to each of your custom characters, beginning with zero for your first character, number 1 for your second character, number 2 for your third character, etc.

2. Correlate your character numbers, one for one, with the Atari internal character set numbers in Table 9.6.

3. To print your custom character, enter the corresponding Atari character in your print statement. For example, the Atari character number 4 is the dollar sign ($). PRINT #6; "$" will print your custom character number 4 on the screen.

It is necessary to skip the third character of your set. (See line 220 of Program 2.) The third character corresponds to the Atari internal character quotation mark ("). It is not possible to print a quotation mark to screen using the PRINT #6 statement.

The color of a character is selected by its form in the print statement. If the custom character corresponds to an Atari alphabetical letter, the color is determined by entering the corresponding Atari letter in the print statement in upper- or lowercase, or inverse upper- or lowercase. Four colors are available for characters corresponding to Atari alphabetical letters.

If the custom character corresponds to an Atari number, punctuation mark, etc., the color is determined by entering the corresponding Atari number in the print statement in standard or inverse video. No upper- or lowercase numbers and punctuation marks are possible, so only two colors are available for these characters when using the PRINT #6 method. (See the table for character color information.)

The PRINT #6 method of putting custom characters on the screen has some serious drawbacks. The method used in Program 3 may not be as easy to understand, but has fewer limitations, particularly for drawing entire playfields.

Mixing Standard And Custom Characters

In addition to colorful playfields, most games print numbers and specific letters on the screen to display such things as score, time, fuel, hits, etc. The standard Atari character set already contains these characters, so it would be pointless to develop custom characters for this purpose. The solution is to develop a custom character set containing all the necessary standard numbers and letters, but to replace all unneeded standard characters with custom characters.

The procedure for developing a mixed character set is described below:

Note: Refer to Table 9.6 in the *Atari BASIC Reference Manual*, page 55.

1. Determine which standard characters will be needed in your program.

2. Form a string variable which contains the unneeded standard characters. The string may include any unneeded characters with numbers between zero and 127 for graphics 0, or between zero and 63 for graphics 1 and 2. The only exception in either case is the quotation mark, for reasons explained before. (Try putting a quotation mark in your string.) An example string:

CHNEW$="!#*<BFGJLMPQZ"

3. Copy the standard character set from CHORG (57344) to CHBASE by PEEK and POKE statements.

4. Modify the unneeded standard characters into custom characters by POKEing custom character data into the character address (CHADD) of each character in the string. (See Program 3 for specific details of the procedure.)

Printing Complete Playfields

Program 2 places the custom characters on screen with PRINT #6 statements. A better, though more difficult, method is plotting the character on the screen using color data to designate which character is to be plotted and in what color the character will appear. The color data to define a character contains two élements: the character number (the Atari internal character set number from table 9.6), and a plus or minus offset which determines the color of the character. The offsets may be obtained from Figure 9.7 on page 56 of the *Atari BASIC Reference Manual*. The easiest way to explain this concept is with examples.

Example 1: Suppose you want to display the standard character "K" in Graphics mode 1 with color 0:

1. From table 9.6, the internal character number for "K" is 43. Note that the "K" is from column 2.

2. From table 9.7, the offset to produce a column 2 character in color 0 is +32.

3. The color data to plot "K" in color 0 would be 43+32=75.

4.
```
210 ...
220 Color 75
230 Plot 5,7
240 ...
```
The program lines above will print a "K" in color 0 at X=5, Y=7.

Example 2: Suppose you want to display your custom character number 19 in graphics mode 2 with color 3. Your character number 19 corresponds to the standard character ";":

1. From table 9.6, the internal character number for ";" is 27 from column 1.

2. From table 9.7, the offset to produce a column 1 character in color 3 is +128.

3. The color data to plot your custom character in color 3 would be 27+128=155.

4.
```
150 ...
160 Color 155
170 Plot 7,1
180 ...
```
The program lines above will print your custom character in color 3 at X=7, Y=1.

A complete playfield may be drawn using the color/plot method by implementing a nested row, column loop which reads the color numbers from DATA statements and plots the characters to the screen (see lines 550 through 610 of Program 3 for a method).

Program 3 is a full screen, graphics 2, fixed playfield demonstration using 31 custom characters:

Lines 30-80	Initialize, define string, and find CHBASE.
Lines 110-130	Move standard character set down to CHBASE.
Lines 150-210	Modify the characters in the string into custom characters. Line 160 locates the correct addresses to modify. The -32 is an offset to change ATASCII to Atari internal character numbers.
Lines 301-331	Custom character data.
Line 420	Select split screen mode; kill cursor.
Line 510-530	Change character set pointer; select colors.
Lines 550-610	Read color data and plot characters on screen.
Line 630	Print standard characters in text window.
Lines 650-680	Flicker engine exhaust.
Lines 700-709	Color data for ten rows of 20 characters.

Figure 1: Atari Display Mode Facts

Graphics Mode	Graphics Type	Resolution H x V	Colors Available (Including Background)	Bytes of Memory/ Screen
0	Character	320 x 192	2	960
1	Character	160 x 192	5	480
2	Character	160 x 96	5	240
5	Bit Mapped	80 x 48	4	960
7	Bit Mapped	160 x 96	4	3840
8	Bit Mapped	320 x 192	2	7680

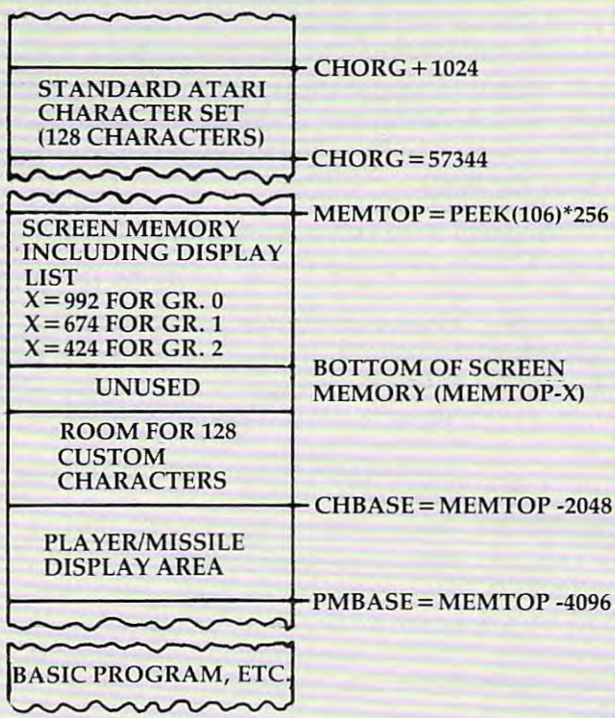
Character Color Information

Character Type	Color Register
Uppercase alphabetical	0
Lowercase alphabetical	1
Inverse uppercase alphabetical	2
Inverse lowercase alphabetical	3
Numbers, punctuation marks, etc.	0
Inverse numbers, punctuation marks, etc.	2

Figure 2: Typical Custom Character

Row Byte Memory Location	Rows	Column Values (128 64 32 16 8 4 2 1)	
CHADD	Byte 1 = 24		8+16
CHADD+1	Byte 2 = 36		4+32
CHADD+2	Byte 3 = 66		2+64
CHADD+3	Byte 4 = 255		1+2+4+8+16+32+64+128
CHADD+4	Byte 5 = 0		0
CHADD+5	Byte 6 = 27		1+2+8+16
CHADD+6	Byte 7 = 24		8+16
CHADD+7	Byte 8 = 64		64

Figure 3: Memory Map



Program 1: Character Editor

```
5 CLR :? "{CLEAR}":OPEN #1,4,0,"K:":
  OPEN #6,4,0,"S:":SETCOLOR 2,9,2:SE
  TCOLOR 4,9,2:POKE 752,1
10 DIM C$(1),STORE(8):N=0
20 GOSUB 6000
40 ? :? :? "PLUG JOYSTICK INTO JACK
   1"
50 ? "DRAW CHARACTER WITH JOYSTICK"
60 ? "HOLD TRIGGER BUTTON TO ERASE":
   ? :?
70 ? :? :? "{8 SPACES}PLEASE WAIT .
   . ."
90 CHBASE=(PEEK(106)-8)*256:CHORG=57
   344
100 FOR I=0 TO 1023:POKE CHBASE+I,PE
    EK(CHORG+I):NEXT I
105 C$="&"
110 CHADD=CHBASE+(ASC(C$)-32)*8
120 POKE 756,CHBASE/256
200 ? "{CLEAR}":POKE 752,1:GOSUB 600
    0
205 FOR I=0 TO 7:STORE(I)=0:NEXT I
210 ?
220 ? "{5 SPACES}87654321"
230 ? "{4 SPACES}{Q}{8 R}{E}"
240 ? "{4 SPACES}|{8 SPACES}| 1"
250 ? "{4 SPACES}|{8 SPACES}| 2"
260 ? "{4 SPACES}|{8 SPACES}| 3
    {3 SPACES}C = CLEAR CHAR"
270 ? "{4 SPACES}|{8 SPACES}| 4
    {3 SPACES}D = DEMO CHAR"
280 ? "{4 SPACES}|{8 SPACES}| 5
    {3 SPACES}P = PRINT DATA"
290 ? "{4 SPACES}|{8 SPACES}| 6
    {3 SPACES}E = ENTER DATA"
300 ? "{4 SPACES}|{8 SPACES}| 7"
310 ? "{4 SPACES}|{8 SPACES}| 8"
320 ? "{4 SPACES}{Z}{8 R}{C}"
499 REM MAIN LOOP
500 X=7:Y=6
510 K=PEEK(764)
512 IF STRIG(0)=0 THEN 700
513 IF K=18 THEN 1000
514 IF K=58 THEN 2000
515 IF K=10 THEN 3000
516 IF K=42 THEN 5000
518 POSITION X,Y:? " "
520 FOR DELAY=1 TO 15:NEXT DELAY
530 POSITION X,Y:? "■"
540 FOR DELAY=1 TO 15:NEXT DELAY
550 ST=STICK(0)
560 IF ST=15 THEN 510
570 IF ST=6 OR ST=14 OR ST=10 THEN Y
    =Y-1
580 IF ST=5 OR ST=9 OR ST=13 THEN Y=
    Y+1
590 IF ST=5 OR ST=6 OR ST=7 THEN X=X
    +1
600 IF ST=9 OR ST=10 OR ST=11 THEN X
    =X-1
610 IF X>14 THEN X=14
620 IF X<7 THEN X=7
630 IF Y>13 THEN Y=13
640 IF Y<6 THEN Y=6
650 GOTO 510
700 POSITION X,Y:? "■"
710 FOR DELAY=1 TO 15:NEXT DELAY
720 POSITION X,Y:? " "
730 FOR DELAY=1 TO 15:NEXT DELAY
732 IF K=18 THEN 1000
734 IF K=58 THEN 2000
736 IF K=10 THEN 3000
738 IF K=42 THEN 5000
740 GOTO 550
999 REM CLEAR CHAR
1000 POKE 764,255
1010 GOTO 200
1999 REM DEMO CHAR
2000 BYTE=0:BIT=0
2005 GOSUB 4000
```

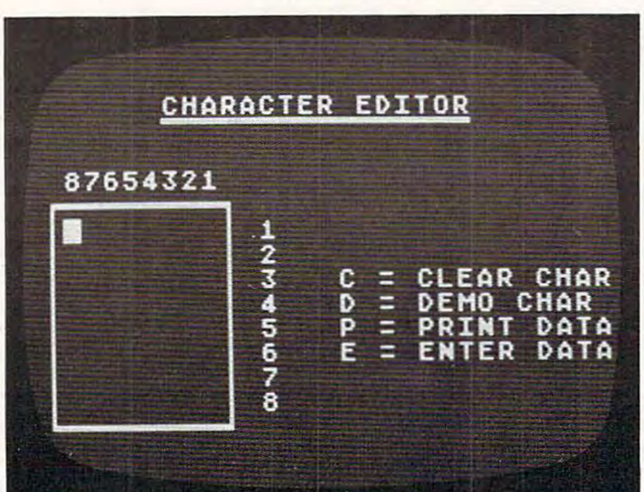
```
2008 REM DETERMINE DATA VALUES
2010 FOR Y=0 TO 7
2020 FOR X=7 TO 0 STEP -1
2030 LOCATE (X+7),(Y+6),PIX
2035 POSITION (X+7),(Y+6):PUT #6,PIX
2040 IF PIX=160 THEN PIX=1
2050 IF PIX=32 THEN PIX=0
2060 IF X=7 THEN BIT=PIX
2070 IF X=6 THEN BIT=PIX*2
2080 IF X=5 THEN BIT=PIX*4
2090 IF X=4 THEN BIT=PIX*8
2100 IF X=3 THEN BIT=PIX*16
2110 IF X=2 THEN BIT=PIX*32
2120 IF X=1 THEN BIT=PIX*64
2130 IF X=0 THEN BIT=PIX*128
2140 BYTE=BYTE+BIT
2150 NEXT X
2160 STORE(Y)=BYTE
2165 BYTE=0
2170 NEXT Y
2180 POSITION 2,16:? "DATA ";
2190 FOR Y=0 TO 6
2200 STORE=STORE(Y)
2210 ? STORE;",";
2220 NEXT Y
2230 STORE=STORE(7)
2240 ? STORE;
2242 FOR J=0 TO 7:STORE=STORE(J)
2244 POKE CHADD+J,STORE:NEXT J
2248 REM ALTER DISPLAY LIST
2250 A=PEEK(560)+PEEK(561)*256
2260 POKE A+25,6:POKE A+26,6:POKE A+
     27,7:POKE A+28,PEEK(A+29):POKE
     A+29,PEEK(A+30):POKE A+30,PEEK(
     A+31)
2265 REM PRINT CHAR TO SCREEN
2270 POSITION 2,18:? "{3 SPACES}GR 0
     :  & & & & & & & & & & & & &";
2280 POSITION 0,20:? #6;"GR 1: & & &
      & & & &";
2290 POSITION 0,21:? #6;"GR 2: & & &
      & & & &";
2345 POKE 764,255
2350 GOTO 500
2999 REM PRINT DATA TO PRINTER
3000 TRAP 3100
3005 POKE 559,0
3030 GOSUB 3200
3040 LPRINT "DATA ";S0;",";S1;",";S2
     ;",";S3;",";S4;",";S5;",";S6;",
     ";S7
3050 POKE 559,34
3060 POKE 764,255
3070 GOTO 200
3100 GOSUB 4000
3110 POKE 559,34
3120 POSITION 2,17
3130 ? "  PRINTER NOT CONNECTED"
3140 ? "{9 SPACES}- OR -"
3150 ? "{3 SPACES}PRINTER TURNED OFF"
3160 FOR DELAY=1 TO 400:NEXT DELAY
3165 GOSUB 4000
3170 POKE 764,255
3180 GOTO 200
3200 S0=STORE(0):S1=STORE(1):S2=STOR
     E(2):S3=STORE(3):S4=STORE(4):S5
     =STORE(5):S6=STORE(6):S7=STORE(
     7)
3210 RETURN
3999 REM CLEAR DATA SUB
4000 POSITION 2,16
4010 FOR Y=16 TO 19
4020 ? "{37 SPACES}"
4030 NEXT Y
4040 POSITION 0,20:? "{19 SPACES}"
4050 POSITION 0,21:? "{19 SPACES}"
4200 RETURN
4999 REM ENTER DATA INTO PROGRAM
5000 POKE 559,0
5010 GOSUB 3200
5020 GOSUB 5200
5030 ? 9000+N;" DATA ";S0;",";S1;","
     ;S2;",";S3;",";S4;",";S5;",";S6
     ;",";S7
5040 GOSUB 5210
5050 N=N+1
5060 POKE 764,255
5070 POKE 559,34
5080 GOTO 200
5200 ? CHR$(125):? :RETURN
5210 ? :? :? "CONT":POSITION 0,0:POK
     E 842,13:STOP
5220 POKE 842,12:? CHR$(125):? :RETU
     RN
6000 ? "{10 SPACES}CHARACTER EDITOR"
6010 ? "{10 SPACES}{16 M}"
6020 RETURN
```

Program 2: Custom Characters

```
10 N=0
20 MEMTOP=PEEK(106)*256
30 CHBASE=MEMTOP-2048
40 REM CLEAR MEMORY FOR NEW CHARACTE
   R SET
50 FOR I=CHBASE TO CHBASE+1024
60 POKE I,0
70 NEXT I
80 REM POKE NEW CHARACTER SET INTO M
   EMORY
90 READ A
100 IF A=999 THEN 300:REM 999 IS END
     OF DATA FLAG
110 POKE CHBASE+N,A
120 N=N+1
130 GOTO 90
190 REM DATA STATEMENTS FOR SPACE,6
    CHARACTERS AND FLAG. FIRST CHARA
    CTER MUST BE A SPACE
195 REM LINE 220 IS A SPACE TO SKIP
    THE QUOTATION MARKS
200 DATA 0,0,0,0,0,0,0,0
210 DATA 32,33,35,35,35,35,255,255
220 DATA 0,0,0,0,0,0,0,0
230 DATA 112,112,112,112,248,248,248
    ,248
240 DATA 248,252,254,254,86,6,255,25
    5
250 DATA 0,0,32,32,32,32,112,240
260 DATA 41,38,32,32,32,32,32,32
270 DATA 0,0,0,0,0,32,32,48
280 DATA 999
290 REM SET GRAPHICS MODE
300 GRAPHICS 2
310 REM TELL COMPUTER WHERE TO FIND
    NEW CHARACTER SET
320 POKE 756,CHBASE/256
324 REM PRINT NEW CHARACTERS
325 POSITION 9,7
330 ? #6;"'%"
335 POSITION 9,8
340 ? #6;"&#"
345 POSITION 9,9
350 ? #6;"!$"
360 GOTO 360
```

Program 3: Fixed Playfield Demonstration

```
10 CLR
20 REM N = NUMBER OF CHARACTERS IN CHNEW$ STRING
30 N=31:CHORG=57344
40 REM DEFINE STRING
50 DIM CHNEW$(N)
60 CHNEW$="!#$%&'()*+,-./;<=>?@BGHJKMNPQVW"
70 REM FIND CHBASE
80 CHBASE=(PEEK(106)-8)*256
90 ? :? "  PLEASE WAIT, 760 NUMBERS TO MOVE"
100 REM COPY STANDARD CHARACTER SET FROM CHORG TO CHBASE
110 FOR I=0 TO 511
120 POKE CHBASE+I,PEEK(CHORG+I)
130 NEXT I
140 REM READ AND POKE CUSTOM DATA INTO THE CHARACTERS IN STRING CHNEW$"
150 FOR I=1 TO N
160 CHADD=CHBASE+(ASC(CHNEW$(I))-32)*8
170 FOR J=0 TO 7
180 READ A
190 POKE CHADD+J,A
200 NEXT J
210 NEXT I
300 REM CUSTOM CHARACTER DATA
301 DATA 0,0,0,128,0,0,0,0
302 DATA 0,0,0,0,0,0,0,16
303 DATA 0,0,0,0,1,0,0,0
304 DATA 8,0,0,0,0,0,0,0
305 DATA 0,0,0,0,0,0,31,127
306 DATA 0,0,0,0,0,0,255,255
307 DATA 0,0,0,0,0,0,248,254
308 DATA 0,0,0,0,7,15,31,31
309 DATA 1,7,31,24,255,255,255,219
310 DATA 255,231,255,0,255,255,255,219
311 DATA 128,224,248,24,255,255,255,219
312 DATA 0,0,0,0,224,240,248,248
313 DATA 31,31,15,7,1,1,3,2
314 DATA 219,255,255,255,127,16,32,64
315 DATA 219,255,255,255,255,24,60,60
316 DATA 219,255,255,255,254,8,4,2
317 DATA 248,248,240,224,128,128,192,64
318 DATA 6,5,6,12,127,0,0,0
319 DATA 128,0,0,0,0,0,0,0
320 DATA 60,126,126,126,60,60,60,60
321 DATA 1,0,0,0,0,0,0,0
322 DATA 96,160,96,48,254,0,0,0
323 DATA 24,24,24,24,24,0,0,0
324 DATA 128,192,240,240,248,252,254,255
325 DATA 129,195,231,255,255,255,255,255
326 DATA 128,192,192,224,224,224,248,255
327 DATA 1,3,7,31,63,63,127,255
328 DATA 1,3,7,7,15,31,63,255
329 DATA 255,255,255,255,255,254,249,7
330 DATA 252,251,247,207,191,127,255,255
331 DATA 255,255,255,255,255,255,255,255
400 REM PUT PLAYFIELD ON SCREEN
420 GRAPHICS 2:POKE 752,1
500 REM TELL COMPUTER WHERE TO FIND NEW CHARACTER SET
510 POKE 756,CHBASE/256
530 SETCOLOR 0,3,6:SETCOLOR 1,8,6:SETCOLOR 2,1,10:SETCOLOR 3,0,10
540 REM PLOT CHARACTERS USING COLOR DATA
550 FOR ROW=0 TO 9
560 FOR COLUMN=0 TO 19
570 READ CHAR
580 COLOR CHAR
590 PLOT COLUMN,ROW
600 NEXT COLUMN
610 NEXT ROW
620 REM PRINT STANDARD NUMBERS AND LETTERS IN TEXT WINDOW
630 ? :? "FUEL:2568  STARDATE:174  ALTITUDE:390";
```



This simple utility program allows you to design your own Atari custom characters.

An example of the game playfields you can create with custom characters. This screen was generated by one of the following sample programs.

```
640 REM BLINK ENGINE EXHAUST
650 FOR LUM=0 TO 8 STEP 2
660 SETCOLOR 0,3,LUM
670 NEXT LUM
680 GOTO 650
699 REM CHARACTER COLOR DATA
700 DATA 0,129,0,0,0,0,131,0,132,0,1
    33,0,0,132,0,0,0,0,0,131
701 DATA 0,0,0,133,0,0,0,0,6,7,8,0,1
    29,0,0,129,0,133,0,0
702 DATA 0,133,0,0,132,0,9,10,11,11,
    11,12,13,129,0,0,131,0,129,0
703 DATA 0,0,132,0,0,0,14,15,27,27,2
    7,28,29,0,133,0,0,0,0,132
704 DATA 202,133,0,132,0,133,30,31,6
    4,64,64,98,103,0,129,0,206,202,1
    32,0
705 DATA 215,203,202,0,0,0,129,133,7
    2,72,72,0,133,0,206,203,215,215,
    205,0
706 DATA 215,215,215,205,133,0,206,2
    03,215,202,133,0,0,208,215,215,2
    15,215,215,202
707 DATA 215,215,209,214,215,215,215
    ,215,215,215,215,209,214,215,215
    ,215,215,215,215,215
708 DATA 215,215,215,215,215,215,215
    ,215,215,215,215,215,215,215,215
    ,215,215,215,215,215
709 DATA 215,215,215,215,215,115,99,
    111,114,101,26,21,16,19,23,215,2
    15,215,215,215
```

©

Sinclair/Timex Screen Splitter

Harold Miller

This short, clever technique allows you to put text on screen and then clear the screen below at will, leaving your messages intact.

After you've owned your Sinclair/Timex computer for a few minutes, you know that the bottom two lines on the screen are not available to you. Your computer needs them to send you report codes and to display the program line you are editing.

Here's a way to reserve as many lines as you need at the top of the screen for your own words. You can put instructions, reminders, warnings, and other messages there. Then, instead of using CLS in your program, you can use the USR command (a call to a six-byte machine language routine) to clear the screen of all but those top lines.

The machine language routine is easy to add to any program in the form of a REM statement that *must* be the first line in your program:

1 REM *T*?LN G*S*TAN

The six keystrokes after REM are: graphic T/?/function LN/G/graphic S/function TAN.

Before you can use this routine, you must POKE into it information on how many lines you want to protect at the top of the screen:

POKE 16515, 24-N

where N is the number of protected lines (and 24-N is the number of lines that are cleared).

Then, each time you want to clear only the lower portion of the screen, use:

RAND USR 16514

To see how easily this works, add these lines *after* the REM line and RUN:

```
10 FOR I=0 TO 20
20 PRINT AT I,0;I
30 NEXT I
40 PRINT AT 21,0;"N=?"
50 INPUT N
60 POKE 16515,24-N
70 PRINT AT 21,0;"PRESS ENTER TO CLEAR"
80 PAUSE 4E4
90 RAND USR 16514
```

Here is the assembly listing of the machine language subroutine:

Location	Decimal	Hex	Opcode	Comment
16514	6	06	ld b,N	loads N into the B register
16515	N_D	N_H		$N_H \leq 24$
16516	205	CD	call 0A2C	calls CLS routine in ROM, but skips 0A2A and 0A2B, which is ld b,24
16517	44	2C		
16518	10	0A		
16519	201	C9	ret	

©

VIC Power Spirals

Bruce Jordan

Computer art in the form of "spiro-graphic" designs. Spiro is an excellent demonstration of the high resolution color graphics capability of the VIC 20 Superexpander module. You'll see almost anything – from geodesic domes to impossible bottles.

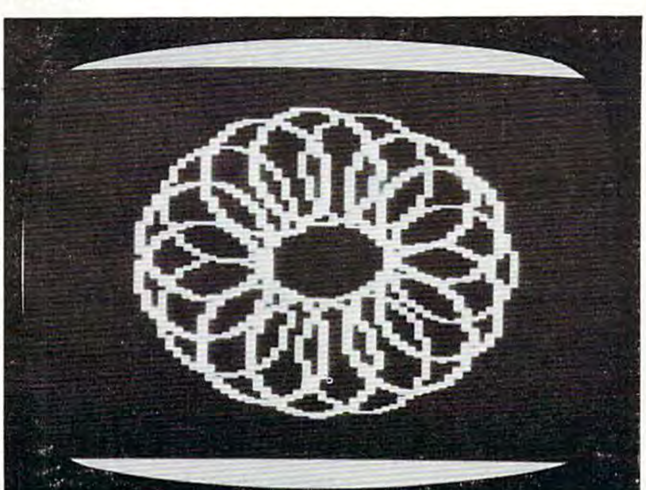
"Spiro" works by choosing a screen location based on a value of a circular function; then, it uses the CIRCLE command of the Superexpander to draw a circle (I call them spiros) around the screen location. Next, the program loops back to the function, takes a new value, and the process is repeated until the design is complete.

When the program is run, it asks for information that gives you control over the final design.

Creating Your Design

The first thing that Spiro asks you is "How many spiros?" This determines how many times the spiro will be repeated across the screen. The higher the number, the more complex the design will be. Try 30 to start with.

Second, it asks "Alternate?" This allows you to create a design consisting of two different spiros. If you type Y, Spiro will ask you for the height and width of each of the two spiros to be used in your design. But if you type N, you will be asked for the height and width of a single spiro. Try N, and then 200 for the height and 150 for the width.



One of the designs possible with a VIC-20 and "Power Spirals."

The last thing you are asked is which graphics mode you want your design to be drawn in. If you type 1, the design will be drawn in the multicolor mode. If you choose 2, the design will be drawn in the high-resolution mode, and in black and white.

After the design is finished, you can return to the beginning of the program by typing R. This lets you alter your design by changing the parameters. If you type F after your design is complete, the program will automatically list the plotting function to allow for quick and easy editing.

Changing the plotting function will radically alter how your design will look. Depending on what function you use, you can wind up with anything from a geodesic sphere to a sort of improbable bottle that has its insides opening outwards to become its outsides! The best thing about Spiro is that it's unpredictable. Below is a list of some of the possible plotting functions. Try these, or make up your own and see what you get.

Plotting Functions For Spiro

1. X=511+200*COS(A), Y=511+250*SIN(A)
2. X=511+200*COS(A), Y=511+250*SIN(A)↑3
3. X=511+200*COS(A)↑3, Y=511+250*SIN(A)↑3*COS(A)
4. X=511+150*SIN(A)+200*COS(A), Y=511+325*SIN(A)
5. X=511+200*SIN(A)*COS(A), Y=511+250*SIN(A)

```
10 PRINT"{CLEAR}"
20 INPUT"HOW MANY SPIROS";Z
30 INPUT"ALTERNATE";A$
40 IFA$="N"THEN90
50 INPUT"#1 HEIGHT";H1
60 INPUT"#1 WIDTH";W1
70 INPUT"#2 HEIGHT";H2
80 INPUT"#2 WIDTH";W2:GOTO110
90 INPUT"HEIGHT OF SPIRO";H1
100 INPUT"WIDTH OF SPIRO";W1
110 INPUT"GRAPHICS MODE";G
120 IFG=1THENP=3:K=0
130 IFG=2THENP=0:K=1
140 GRAPHIC G:COLOR K,4,P,7
150 A=0:DA=2*π/Z
160 I=1:DI=1
170 GOSUB 280
180 CIRCLE I,X,Y,W1,H1
190 IF A$="Y" THEN GOSUB 350
200 A=A+DA:IF A=2*π THEN 230
210 I=I+DI:IFI=>4THEN160
220 GOTO170
230 GETB$:IFB$=""THEN230
240 IFB$="R"THEN260
250 IFB$="F"THEN270
```

```
260 GRAPHIC 0:COLOR 1,2,0,4:GOTO 20
270 GRAPHIC 0:COLOR 1,2,0,4:LIST 280-290
280 X=511+200*COS(A)
290 Y=511+250*SIN(A)
300 IFX<=0THENX=0
310 IFX=>1023THENX=1023
320 IFY<=0THENY=0
330 IFY=>1023THENY=1023
340 RETURN
350 L=I:IFL<=2THENL=L+1:GOTO370
360 IFL=>3THENL=1
370 CIRCLE L,X,Y,W2,H2
380 RETURN
```

©

Standard VIC 20

no additional memory needed

(CG008) Alien Panic $12.95
Race against time as your guy digs holes to trap aliens in 4 floor laddered, brick construction site. Requires joystick.

(CG096) Antimatter Splatter $24.95
This game is as good as its name. Another pure machine code game, this one is fast! The alien at the top of the screen is making a strong effort to rid the world of humankind by dropping antimatter on them. The splatter cannon and you are our only hope as more and more antimatter falls. Joystick again is optional equipment.

(CG026) Collide $12.95
"Vic" controls one, you the other as cars go opposite directions on 4 lane track. Requires joystick.

(CG094) Exterminator $24.95
Recently scoring a rating of 10 out of a possible 10 this game was praised as "one of the best I've seen on any computer" by a prominent reviewer in a leading magazine. The idea is to shoot a centipede before it overuns you, the problem being every time you hit it, it divides into two separate shorter ones. Several other little creatures bounce around during this struggle. All of them lethal. 100% machine language makes the rapid fire action very smooth. A joystick is optional, but as always, recommended, (a trac ball is also very nice!).

(CG054) Krazy Kong $12.95
Three screens, a gorilla, barrels, and changing difficulty levels help to make this one of our most popular. Joystick optional.

(CG098) Racefun $19.95
Extensive use of multicolored character capabilities of the "Vic" make this one very appealing to the eye. Fast all machine language action, quick response to the stick or keyboard controlled throttle, combine with the challenge of driving in ever faster traffic to make it appeal to the rest of the body. Joystick controlling is an option.

(CG058) Rescue From Nufon $12.95
Must find 30 hostages in this 100 room, 5 story, alien infested, graphic adventure game. A continual big seller. Keyboard only (n. = north w = west etc.)

(CG068) The Catch . . . $12.95
Another all machine language game based on the principle that one person with one joystick guiding one catch/shield can catch everything that one alien can throw at one. The action comes slowly at first but by the fourth wave you'll be aware of . . . "The Catch" . . .

Expanded Memory Vic 20 Games

(CG090) Defender On Tri $19.95
Pilot a defender style ship on mission to save trapped scientists from a fiery fate (they are aboard an alien vessel deep in the gravity well of sol). Excellent graphics. Short scene setting story in the instructions. "Defender On Tri" requires at least 3K added memory.

(CG092) 3D Man $19.95
The maze from probably the most popular arcade game ever, with perspective altered from overhead to eye level. The dots, the monsters, the power dots, the side exits, the game is amazing. "3D Man" requires at least 3K added memory.

(CG088) Space Quest $19.95
Our first 8K memory expander game and its a beauty. The scene (a short story is included) is far in the future, a time when man's knowledge has reduced an entire galaxy into a mapped series of quadrants. This game has stratagy (you plot your own hyperspace jumps on Galaxy map), action (against a starry background you find yourself engaged in a dogfight, laser style), exploration (you must fly your ship deep into caverns to pick up necessary fuel). "Space Quest" requires at least 8K memory expansion and a joystick.

Commodore 64

(CG602) 3D-64, Man $19.95
This available on the expanded "Vic 20" game, has been completely rewritten for the 64 and uses sprites, sounds, and other features not available on the "Vic". This one requires a joystick.

NÜFEKOP

P.O. Box 156, Shady Cove, Oregon 97539-0156 Mastercard and Visa cards accepted C.O.D. Orders...call (503) 878-2113

VIC is a trademark of Commodore Business Machines, Inc. Games will be on tape unless you request disk. Ask for our FREE catalog!

Using The Atari Timer

Stephen Levy

Because FOR/NEXT loops are not accurate timers, the solution is to incorporate Atari's internal counters into programs where you want something delayed or timed reliably.

Have you ever written a program and wanted a specific time delay? What did you do? Some of us figured a FOR/NEXT loop was the answer, so we set to work with our stopwatches until we found that the following takes about three seconds to write "STOP":

```
10 PRINT "BEGIN"
20 FOR X=1 TO 1000
30 NEXT X
40 PRINT "STOP"
```

Then we went along and wrote our programs and found that our three-second delay had become five, six, or even ten seconds. Why? Because the Atari FOR/NEXT loops take longer as you add lines of code to the program.

There is a better way. Yes, machine language routines are great for timing on the Atari, especially if you know how to use locations 536 to 558 ($218 to $22E). But it can be most disconcerting if you allow some of those registers to drop to zero unchecked.

Accurate Delays

BASIC programmers, there is a way. Use memory locations 18, 19, and 20. (In the May 1981 issue of **COMPUTE!**, Richard Bills shows how to use these locations for timing in "Real-Time Clock on the Atari.")

These timers work like the mileage gauge on a car's speedometer: one counter counts up and then sets the one next to it which, in turn, sets the next one. Each counter on the speedometer goes up when the one to its right hits ten. In the computer, they count up to 255 before going back to zero.

Register number 20 counts at the rate of 60 numbers per second up to number 255, then increments register 19 by one and starts over. When register 19 reaches 255, it increments register 18 by one. If you POKE zero into all three registers, it will take about 1092 seconds before a one appears in register 18 (more than 18 minutes). The table gives some times (it assumes all three registers began with zero). Notice that it would take more than 77 hours for memory location 18 to reach 255.

Well, how does all this help? Let's look at our short program again. We can rewrite it this way:

```
10 PRINT "BEGIN": POKE 20,0
20 IF PEEK(20)<180 THEN 20
30 PRINT "STOP"
```

This routine will continue to take three seconds no matter how long your program. Well, not exactly; since it is written in BASIC, the longer the program, the longer the routine will take. But the influence of the program length will usually be negligible.

Included here are three programs which demonstrate a much more functional use of this timer. Type in Program 1, leaving out the REM statements. This program tells the user the time interval between the pressing of RETURN after typing RUN and the pressing of RETURN a second time. Notice that if you press another key the computer goes back to line 130.

This short program demonstrates several useful concepts. First, the computer is looking for a particular input, in this case the RETURN key (ATASCII 155). Second, line 150 PEEKs at registers 18, 19, 20. Notice we POKEd location 20 last on line 110 and PEEKed at it first on line 150. Third, line 160 contains the important formula for converting the information in locations 18, 19, 20 to seconds. Why 4.267? Because 256 divided by 60 numbers per second equals 4.267. Fourth, lines 170 to 190 convert the total number of seconds to minutes and seconds.

Program 2 is a bit more useful. It is a timed math quiz in which the user is allowed eight and one half seconds to answer. Line 55 is used to check if a key has been pressed. If no key has been pressed, then the program goes back to check how much time has elapsed. Once a key is pressed, the computer GETs the ATASCII code and calls it A1. At lines 70 and 80, A1 is converted to its CHR$ and placed in its proper place in ANS$. If A1 equals 155 (ATASCII code for the RETURN key), then the program moves to line 200, where the value of ANS$ is put into variable ANS.

The final illustration, Program 3, is also a math quiz. In this case the user is given unlimited time. This program combines elements of both programs 1 and 2.

If you are serious about personal finance...

- **Budget Forecast** – 26 expense categories
- **Check Entry** – easy data entry – scan & modify 26 major & 36 sub-categories – information block
- **Check Search** – single or multiple parameters – (up to seven) to search entries
- **Tabulations** – detailed expense vs. budget comparisons by month, year-to-date, category
- **Bar Graphs** – screen displays in graph form expenses vs. budget – by month or category – printing with graphic capable printers
- **Check Reconciliation** – fast clearing of resident checks & deposits, complete summary report
- **Checkwriter** – prints your custom checks
- **Printouts** – most popular printers
- **Multi-Colored Graphics • Audio Enhancements**
- **7 Utility Programs • User-Friendly Operation**
- **Easy To Use Instruction Manual • Audit Report**
- **Handsome Tinted Plastic Storage Case**

COMPUTE!
The Leading Magazine Of Home, Educational, And Recreational Computing

In a feature editorial.

"If you want to use a finance system, but don't want to spend several days trying to learn how to use one, then A Financial Wizard by Computari may be just what you need."

"The illustrated manual that comes with this program is clear, direct, and very thorough."

"It appears that this finance system was designed to achieve the best and most comfortable working relationship between the user and the program."

"The check entry routine is the most attractive feature of this finance system. Data prompts are very clear and the category item names are displayed at all times during data entry for your convenience."

"The file search capabilities of this program are superior. You are offered seven ways to look up the checks."

"The system is disk intensive. All data is saved automatically and immediately following all routines that either enter data or modify it."

"Scanning your entries is made possible by pressing START. You can see records very quickly this way."

"This is an excellent finance system – entertaining, accurate, and fun to use."

ANALOG COMPUTING
THE MAGAZINE FOR ATARI COMPUTER OWNERS

Analog Magazine in a comprehensive study of personal finance systems for Atari* computers.

"A Financial Wizard from Computari is by far the best of these programs and will be the standard of comparison for the others."

"The check entry mode is easy to use..."

"The way a Financial Wizard handles your tabulations is excellent. You can chart your actual expenses vs. your budget by month, by category or year to date."

"...where it really outshines the rest is in the check reconciliation."

"In effect it gives you your bank statement on the screen, a complete list by month of all your checks and deposits."

"A Financial Wizard has one disk that does everything..."

"Graphics, while really not a factor in the quality of programs of this type, do make your budgeting chores a little more pleasant. Again A Financial Wizard comes out on top."

"Everything about this program is excellent..."

Antic
The ATARI Resource

In a Report from Antic.

"Like most Atarians, I am captivated by the graphic, color and sound capabilities of my machine. Nothing quite discourages me more than to boot up an applications program (personal, business, etc.) and to be presented with the standard graphic 'o' white characters on a blue screen.

Of course the usefulness and effectiveness of a program is of primary importance. However, enhancing the dullest of applications programs with some of Atari's charms, is a great asset. A Financial Wizard, a personal finance program by Computari's Bill McLachlan, is an excellent example of an applications program that integrates many of the Atari's features into a well conceived and executed program."

"The use of color and sound in the data input prompts and error checking routines are so well done that it's quite simple to boot up the disk, follow along with the very clear documentation, and be 'up and running' in short order."

"I give A Financial Wizard high marks in ease of use, documentation and performance. If a disk-based home finance package is in your future, The Wizard should get serious consideration."

Computari's
A Financial Wizard 1.5
The logical choice.

The system is designed for Atari computers having a minimum of 32K and operating from a disk drive. The cost is only $59.95 plus $3 for handling/postage.

If your dealer does not have **A Financial Wizard** . . . Telephone orders are accepted on Mastercharge or Visa credit cards. Mail order must be accompanied by check or money-order or credit card #.

Dealer inquiries invited.

*trademark of Atari Inc.

OK residents ad 4% Tax.

Exclusively thru **ON LINE** COMPUTER CENTERS of OKC™ (405)751-7925

10944 North May, Oklahoma City, OK 73120

☐ CHECK ENCLOSED ☐ VISA ☐ MASTERCHARGE

NAME (Print) ______________________

Address ______________________

City ______________________

State ______________ Zip ________

Card # ______________ Exp. ________

Signature ______________________

This Atari timing device should be beneficial whether you wish to impose a time limit, simply time answers, or have users compete against each other or themselves. The timer has applications for both educational programming and games. With some experimentation you should be able to adapt this timing device for use with your own programs.

Sample Times

LOC.20	LOC.19	LOC.18	TIME MIN:SEC
60	0	0	0:01
60	1	0	0:05
0	2	0	0:08
100	2	0	0:10
0	3	0	0:12
100	4	0	0:18
21	14	0	1:00
42	28	0	2:00
84	56	0	4:00
176	112	0	8:00
0	255	0	18:08
0	60	2	40:40
0	0	16	291:17
0	0	100	1820:35
0	0	150	2730:52
0	0	255	4642:29

Program 1: Atari Timer

```
10 REM PROGRAM 1
15 REM
20 REM THIS PROGRAM DEMONSTRATES HOW
30 REM TO USE ATARI TIMER:
40 REM ADDRESS 18,19,20
50 REM IT FIGURES HOW LONG IT TAKES
60 REM YOU TO PRESS THE <RETURN> KEY.
70 REM RUN THE PROGRAM THEN PRESS
80 REM <RETURN>
90 REM PROGRAM RUNS BETTER WITHOUT TH
   E
95 REM REMARK STATEMENTS OR GOTO 100
100 OPEN #1,4,0,"K:"
110 FOR Z=18 TO 20:POKE Z,0:NEXT Z
130 GET #1,D:IF D=155 THEN 150
140 GOTO 130
150 A=PEEK(20):B=PEEK(19):C=PEEK(18)
160 SEC=INT((4.267*256*C)+(B*4.267)+(
    A/60))
170 MIN=INT(SEC/60)
180 M=MIN*60
190 SEC=SEC-M
200 PRINT MIN;" MINUTES ";SEC;" SECON
    DS"
```

Program 2: Timed Math Quiz

```
1 REM PROGRAM 2
2 REM
3 REM THIS IS A TIMED MATH QUIZ
4 REM CHANGE LINE 50 TO A=1
5 REM ALLOWS 4 1/4 SECOND
6 REM A=2 ALLOWS 8 1/2 SECONDS
7 REM A=3 ALLOWS 12 3/4 SECONDS, ETC.
10 OPEN #1,4,0,"K:":DIM ANS$(10)
15 PRINT :Q1=INT(RND(0)*20):Q2=INT(RN
   D(0)*20):X=1
20 PRINT Q1;" + ";Q2;"=";
25 POKE 18,0:POKE 19,0:POKE 20,0
45 A=PEEK(19):B=PEEK(20)
50 IF A=2 THEN 100:REM 8 1/2 SECONDS
55 IF PEEK(764)=255 THEN 45
60 GET #1,A1:IF A1=155 THEN 200
70 ANS$(X,X)=CHR$(A1)
80 PRINT ANS$(X,X);:X=X+1:GOTO 45
100 PRINT :PRINT "TIME'S UP"
110 PRINT "THE ANSWER IS ";Q1+Q2
115 FOR W=1 TO 400:NEXT W
120 ANS$=" ":GOTO 15
200 ANS=VAL(ANS$):PRINT
210 IF ANS=Q1+Q2 THEN PRINT :PRINT "C
    ORRECT":GOTO 115
220 PRINT :PRINT "SORRY":PRINT :GOTO
    110
```

Program 3: Revised Math Quiz

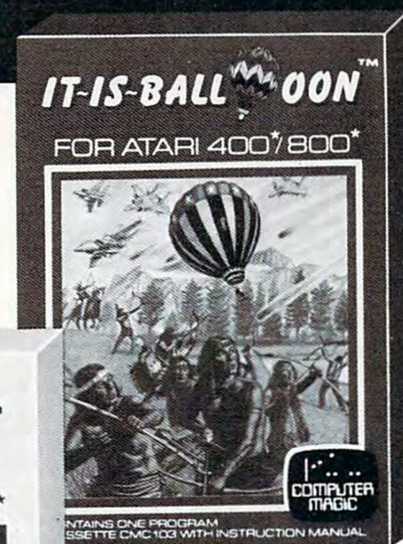
```
1 REM PROGRAM 3
2 REM
3 REM THIS PROGRAM COMBINES ELEMENTS
4 REM OF PROGRAMS 1 AND 2.
5 REM IT GIVES MATH QUIZ AND TELL HOW
6 REM LONG IT TOOK YOU TO DO EACH
7 REM PROBLEM.
10 OPEN #1,4,0,"K:":DIM ANS$(10)
15 PRINT :Q1=INT(RND(0)*20):Q2=INT(RN
   D(0)*20):X=1
20 PRINT Q1;" + ";Q2;"=";
25 POKE 18,0:POKE 19,0:POKE 20,0
60 GET #1,A1:IF A1=155 THEN 190
70 ANS$(X,X)=CHR$(A1)
80 PRINT ANS$(X,X);:X=X+1:GOTO 60
110 PRINT "THE ANSWER IS ";Q1+Q2
115 FOR W=1 TO 1000:NEXT W
120 ANS$=" ":GOTO 15
190 A=PEEK(20):B=PEEK(19):C=PEEK(18)
200 ANS=VAL(ANS$):PRINT
210 IF ANS=Q1+Q2 THEN PRINT :PRINT "C
    ORRECT":GOTO 230
220 PRINT :PRINT "SORRY"
230 SEC=INT((4.267*256*C)+(B*4.267)+(
    A/60))
240 MIN=INT(SEC/60)
250 M=MIN*60
260 SEC=SEC-M
270 IF MIN<>0 THEN 290
280 PRINT "THAT TOOK YOU ";SEC;" SECO
    NDS":GOTO 300
290 PRINT "THAT TOOK YOU ";MIN;" MINU
    TES":PRINT "AND ";SEC;" SECONDS"
300 GOTO 115
```

©

Introducing our exciting new family of video computer games from COMPUTER MAGIC™

IT-IS-BALLOON™

ENJOY THE GOOD LIFE WHILE YOU PILOT YOUR BALLOON OVER TREES, MOUNTAINS, FORESTS AND INDIANS. WATCH YOUR FUEL, LOOK FOR METEOR STORMS AND JET PLANE ATTACKS!

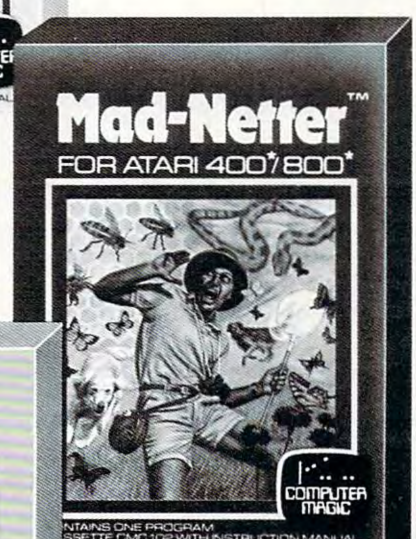


KAYOS™

WELCOME TO THE WORLD OF KAYOS. ALL GROUND LIFE HAS BEEN DESTROYED. YOU ARE UNDER CONSTANT ATTACK BY DEADLY FORCES. HOW LONG WILL YOU SURVIVE.

Mad-Netter™

YOU ARE THE MAD NETTER BEAUTIFUL BUTTERFLIES ARE EVERYWHERE! COLLECT AS MANY AS YOU CAN BUT DON'T GET STUNG BY THE BEES, CHASED BY DOGS, OR BIT BY A SNAKE!



SEE US AT BOOTH #5039, McCORMICK WEST, THE SUMMER CES IN CHICAGO, FOR NEW EXCITING RELEASES!

COMPUTER MAGIC

COMPUTER MAGIC LTD.
18 EAST MALL, PLAINVIEW,
N.Y. 11803/PHONE (516) 694-8960

FOR ATARI 400*/800*

Cassette or Diskette
Requires 16K

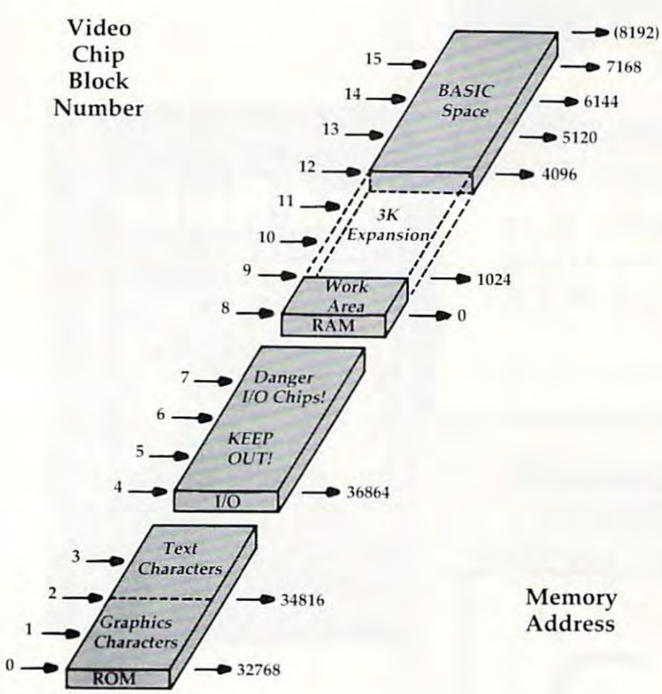
Part II

Visiting The VIC-20 Video

Jim Butterfield, Associate Editor

This is the second part in our exploration of the VIC-20 video, in which the traveler discovers that a character set is less important for its footage than its mileage.

It's worthwhile making an observation about the "minimum VIC" configuration here. We know that the video chip sees memory in an unusual way:



How the video chip sees memory.

Some users have memory expansion permanently connected to their VIC machines. They don't want to plug and unplug the memory units. Yet some programs call for a "minimum VIC with only 5K." A few POKEs can reconfigure any machine to this minimum configuration.

First, we set the Limit-of-BASIC:

```
POKE 55,0:POKE 56,30:CLR
```

And then put the screen into place (Block 15.5):

```
POKE 36869,240:POKE 36866,150:PRINT CHR$(147)
```

This takes care of the high end of memory. It's not always necessary, but we can also set up the low end:

```
POKE 4096,0:POKE 43,1:POKE 44,16:NEW
```

Small Character Sets

A full character set, 256 characters, takes up 2048 bytes of memory; there are eight bytes for each character. We have tried copying over the whole set. On a small VIC, it takes up a lot of our available RAM and starts to cramp our program space. Can we omit some of the characters and save space? Yes, we can.

Our program may not need the reverse video characters. If so, there's a savings of 1024 bytes. Be careful: reverse video is used for flashing the cursor. If you give it up, the cursor may not work in quite the same way.

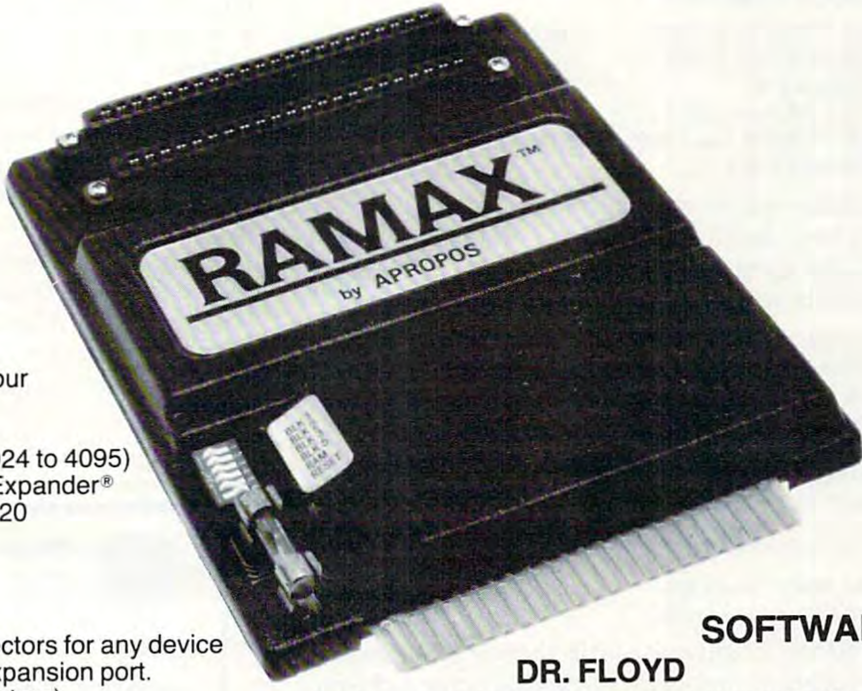
But there's more. Which are the characters that we use the most? Well, the alphabetic characters A to Z, the space character, of course, and the numbers 0 to 9. What luck! These characters are bunched together within the first 58 of the character set, including a few spares. 58 times 8 gives us 464 bytes of storage – not bad for a functional character set.

We could do better than this if we had a specialized display that could work from very few characters. For example, a game might use only four characters: a ball, a ninepin, a "gutter," and the all-important space character to give us blank

RAMAX™

by APROPOS

The ONLY MEMORY your VIC-20® will need



FEATURES

- A full 27k bytes of RAM (added to VICs 5k equals 32k.)
- Fully switchable in sections:
  - BLK 1 switches 8k (Adr. 8192 to 16383)
  - BLK 2 switches 8k (Adr. 16384 to 24575)
  - BLK 3 switches 8k (Adr. 24576 to 32767)
  - BLK 5 allows/disallows your 8k ROM (games) (Adr. 40960 to 49152)
  - RAM switches 3k (Adr. 1024 to 4095)
- May be used with Super Expander® games or **ANY** other VIC-20 compatible cartridge.
- Built in RESET switch.
- Fuse protected.
- Totally self-contained.
- 2 duplicate extension connectors for any device normally plugged into the expansion port. (BLK 5 is switched to connectors)
- Very low power usage. (.150 amp max.)
- High reliability gold plated connectors.
- 6 month parts and labor warranty.
- Factory service. - Extended service always available.

THIS SUPERB PLUG-IN GIVES YOUR VIC-20 REAL POWER AND EXPANDABILITY

FOR ONLY $149.00 Shipping included

10 DAY SATISFACTION OR YOUR MONEY BACK GUARANTEE

WE ARE NOW OFFERING "RAMAX Jr." (19k), which is identical to RAMAX in EVERY way, except the top 8k (BLK 3) is not incorporated. Our introduction price is **$129.00**, shipping included.

WE SERVICE WHAT WE SELL

TO ORDER:

Send Check or Money Order For the Total
Calif. residents add 6% tax.
Phone orders: CALL **(805) 482-3604 24 HRS.**
For credit card orders, include all information on card.
or **contact your local dealer.**

VISA Foreign orders, add $15.00. MasterCard
All items shipped from stock.
DEALER INQUIRIES WELCOME

SOFTWARE

DR. FLOYD

Psychoanalysis by computer? — well, not quite, but Dr. Floyd will carry on a conversation with you using psychoanalytic techniques giving the appearance of artificial intelligence. Requires 16k RAM or more.
$14.95 shipping included.

WORD PLAY

"WORDPLAY" is a collection of programs which allow the user to make original stories, write a form of Japanese poetry, play the fun game of Animal (children love this one), and create jargon. A bonus secret message (cypher) program is also included. In a word, "WORDPLAY" is a bargain.
Requires 16k RAM or more.
$14.95 shipping included.

TYPE FOR YOUR LIFE

With more challenge than an arcade game, learn to type up to 75+ words/min. (User selectable, but no FOOLING AROUND allowed). TEXT IS WIDELY VARIED SINCE IT COMES FROM THE PROGRAM TAPE. Action color graphics with sound fix your eyes to the screen (away from your fingers - clever!) Your man rows your boat up stream as fast as you can type. Maintain speed and destroy the Sea Monster; slow down and he will get you. Runs on the unexpanded VIC.
$14.95 shipping included.

All software is on high quality cassettes and is replacement guaranteed.

VIC-20 & SUPER EXPANDER are registered trademarks of Commodore Business Machines, Inc.

APROPOS TECHNOLOGY

350 N. Lantana Ave., Suite 821
Camarillo, CA 93010

space. Even so, we might be tempted to go the whole alphanumeric set – to display scores, instructions, and the like.

A little arithmetic shows us a convenient arrangement. The character set must start on a block boundary. Screen memory may start on a half-block boundary. If we put them one behind the other, this would give us 512 bytes for the character set, enough for 64 characters.

In fact, let's try this, with the partial character set at block 15 and the screen at its usual block 15.5. We can write a simple graphics demonstration program.

A Little Program

```
100 POKE 56,28:CLR          (lower Limit-of-BASIC)
110 FOR J=0 TO 63           (copy 64 characters)
120 J1=J*8                  (8 bytes per character)
130 FOR K=0 TO 7            (copy each byte)
140 POKE J1+K+7168,PEEK(J1+K+32768)
150 NEXT K,J
```

Here come our custom characters. We'll draw a ship in two characters: the left half in the character 27 and the right half in character 28. The "pixels" of the drawing are in the DATA statements:

```
160 DATA 0,0,4,4,127,63,31,0
170 DATA 0,0,192,192,252,248,240,0
180 FOR J=27 TO 28   (two specials)
190 J1=J*8
200 FOR K=0 TO 7
210 READ X:POKE J+K+7168,X
220 NEXT K,J
```

Now we put our new character set in gear:

```
230 POKE 36869,255
240 POKE 36866,150
```

And we'll draw our little ship with a simple demonstration program. Note that screen character 27 corresponds to ASCII character 91.

```
300 PRINT CHR$(147);"SHIP GRAPHIC"
310 FOR J=2 TO 18   (left to right)
320 PRINT CHR$(19)
330 PRINT TAB(J);CHR$(32);CHR$(91);CHR$(92)
340 FOR K=1 TO 99
350 NEXT K,J
360 GET X$:IF X$="" GOTO 300
```

The program ends here. Restore the regular character set:

```
370 POKE 36869,240
```

Run the program. After the initial pause for generating the new character set, a ship will move across the screen. You can adjust its speed by changing the value of 99 in line 340. The program will terminate if you hold down any key.

If you press RUN/STOP, the program will break with the custom character set still in place. You'll notice the lack of a cursor; apart from that, you can type most alphanumeric characters without problems. You might like to look around to find out which keys now represent left and right halves of the ship. When you are finished playing, type CONT to allow the program to continue, and then terminate by holding a key down.

You may notice that the program does not restore the 512 bytes that it takes for the character generator. So the character set is protected, and you can try going back to it if you wish with a POKE 36869,255. Eventually, you may wish to make the program complete by adding line 380, with a POKE 56 and a CLR. I'll leave you to work out the proper details.

Here's a question that may cross your mind: if the character generator starts at block 15, where would the video chip go for the reverse characters? They would be in the next block, but we don't have a block 16. What happens? The video chip address "wraps around," and we go to block 0. The characters in block zero are not reversed, of course, and that's why the cursor doesn't flash.

We can do some good work with a very small character set. It doesn't necessarily have to be big to be useful.

Another thing that we have spotted in this episode: we can build effective graphic pictures by using more than one character. Our program used two separate characters which together drew a ship, but we can use three, four, six, or more as needed.

COMPUTE!'s First Book Of VIC

Authors: **COMPUTE!** Magazine contributors
Price: $12.95
On Sale: Now

Finally, it's VIC's turn!

Users of other popular personal computers have been enjoying their **COMPUTE! Books**: *COMPUTE!'s First Book Of PET/CBM, Programming The PET/CBM*, and others.

Now, there's a book devoted exclusively to the Commodore VIC-20™ Computer: *COMPUTE!'s First Book Of VIC.*

The editors of **COMPUTE!** Magazine – the leading resource for the VIC-20 – gathered together the best VIC-20 articles published since the summer of 1981 and added some new material. The result is more than 200 pages of valuable information – information that goes beyond the instruction manuals. In the **COMPUTE!** tradition, it is carefully edited to be easily understood and useful for beginners and experts alike.

COMPUTE!'s First Book Of VIC is spiral-bound to lie flat, and includes ready-to-type program listings and articles such as "The Joystick Connection: Meteor Maze," "STARFIGHT3," "Train Your PET To Run VIC Programs," "Renumber BASIC Lines The Easy Way," "High Resolution Plotting," "Custom Characters For The VIC," "VIC Memory – The Uncharted Adventure," and "A Simple Monitor For The VIC."

At only $12.95, less than most computer manuals, *COMPUTE!'s First Book Of VIC* is among the best resources a VIC user can own.

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2 shipping and handling. Outside the U.S. add $5 for air mail, $2 for surface mail. All orders prepaid, U.S. funds only.
VIC-20 is a trademark of Commodore Electronics Limited.

PET Relative File Field Separator

T. A. Zucal

A few lines of BASIC and a small machine language routine solve some speed and syntax problems faced when accessing relative records. For PET/CBMs with 4.0 BASIC and disk.

Often relative files can become more efficient if each record is broken down into fields. When of a fixed length, these fields are easily handled at reasonable speed. But when you attempt to bring fields of variable length in from the file, the coding can get messy. It is necessary to use the LEN( ) function to count bytes, always remembering to add the extra byte for the field delimiter. In addition, this method will not adequately handle commas or colons.

An alternative is to use one string for the entire record, and then to separate the fields using the MID$( ) function in conjunction with the field delimiters. But this method works only if the record length is 80 bytes or less. This is not too bad, but again commas and colons present problems. A third alternative is to use GET#, which will handle colons and commas, but is extremely slow.

Solving The Problem

All of this can be bypassed with a minor addition to the machine code of Jim Butterfield's "String Thing" (**COMPUTE!**, November 1982) and a couple of lines of BASIC. This method will handle all characters at a reasonable speed and with no confusion.

The BASIC loader program moves the machine language code to high memory and works with any size memory. You should copy for later use the SYS memory locations which the loader displays on screen. However, the test program will also adjust itself to any size machine. The delimiter used in this version is the backslash – the slash located above the "I" on graphic keyboards. If you prefer, you can choose any other character by changing the 92 in DATA line number 270 to the PETASCII value of the character desired.

The test program will create a relative file with two records. Record #1 is used by the field separator routine; record #2 is used by the byte count and GET# methods. Each record is exactly the same number of bytes to permit a fair speed test.

When this test is run on all three options, you will notice that the Field Separator Method is approximately 12 jiffies faster than byte counting and 85 jiffies faster than the GET# method. However, the GET# is the only other method which will handle colons and commas without a hassle. So, this method will save you one minute for every 42 records read and will handle all types of input. Even if your fields don't contain the problem characters, you still save 20 seconds for every 100 records.

Program 1: Loader For Field Separator Routine

```
120 REM ** FIELD SEPARATOR FOR RELATIVE FI
    LES
125 REM
130 PRINT"{CLEAR}{02 DOWN}NOW LOADING"
135 REM *** FIND CURRENT TOP OF MEMORY
140 MM=PEEK(52)+PEEK(53)*256
145 REM *** CALCULATE & SET NEW TOP OF MEM
    ORY
150 M=MM-129:GOSUB315
155 POKE52,L%:POKE53,H%
160 REM *** LOAD PROGRAM FROM DATA
165 FORI=MM-128TOMM-53
170 READA%
175 REM *** ADJUST DATA FOR MEMORY SIZE
180 IFI=MM-110THENM=MM-52:GOSUB315:A%=L%
185 IFI=MM-109THENA%=H%:LS%=L%:HS%=H%
190 IFI=MM-68THENA%=L%
195 IFI=MM-67THENA%=H%
200 IFI=MM-63THENM=MM-51:GOSUB315:A%=L%
205 IFI=MM-62THENA%=H%
210 IFI=MM-59THENA%=LS%
215 IFI=MM-58THENA%=HS%
220 IFI=MM-54THENM=MM-79:GOSUB315:A%=L%
225 IFI=MM-53THENA%=H%
230 POKEI,A%
235 POKE32862,A%
240 NEXTI
245 DATA160,2,177,42,153,134,0,200
250 DATA192,6,208,246,152,72,160,0
255 DATA152,153,204,127,200,192,51,208
260 DATA248,104,168,162,1,32,198,255
265 DATA32,228,255,201,13,240,15,164
270 DATA139,145,137,200,132,139,201,92
275 DATA240,7,196,136,208,234,76,204
```

JINI MICRO-SYSTEMS, Inc.

DATABASE MANAGEMENT SYSTEM DESIGN

BOX 274 KINGSBRIDGE STN., RIVERDALE, N.Y. 10463 (212) 796-6200

```
280 DATA255,152,72,172,204,127,165,139
285 DATA153,205,127,200,140,204,127,104
290 DATA168,76,177,127
295 PRINT"{CLEAR}"
300 PRINT"FIELD SEPARATOR LOADED"
305 PRINT"{DOWN}CALL WITH SYS"PEEK(52)+256
    *PEEK(53)+1
310 END
315 H%=M/256:L%=M-H%*256:RETURN
```

Program 2: Test Of Separator Routine

```
100 REM ** FIELD SEPARATOR TEST PROGRAM
105 REM ** STRING MUST BE FIRST VARIABLE
110 A$="ABCDEFGHIJKLMNOPQ"
115 A$=A$+A$+A$+A$+A$
120 A$=A$+A$+A$
125 REM ** FOR RELATIVE FILES STRING IS SA
    FE SET AT MAXIMUM, (255) AS ABOVE

130 REM ** SET UP RELFILE WITH 2 RECORDS
135 REM ** BOTH RECORDS HAVE SAME CHARACTE
    R COUNT
140 PRINT"{CLEAR}{03 DOWN}CREATING RELATIV
    E FILE, 'RELFILE'"
145 DOPEN#1,"RELFILE",D0,L254
150 P$="RECORD1**FIELD-1\THIS IS FIELD #2\
    FIELD #3\PET/CBM ++++\FIELD #5\"
155 P$=P$+"###FIELD 6###\YOU CAN HAVE UP T
    O\50 FIELDS BUT ARE LIMITED\"
160 P$=P$+"TO 254 CHARACTERS, PLUS\THE RET
    URN AT THE END."
165 RECORD#1,1
170 PRINT#1,P$
175 R$=CHR$(13)
180 P$="RECORD1**FIELD-1"+R$+"THIS IS FIEL
    D #2"+R$+"FIELD #3"+R$+"PET/CBM +
    +++"
185 P$=P$+R$+"FIELD #5"+R$+"###FIELD 6###"
    +R$+"YOU CAN HAVE UP TO"+R$
190 P$=P$+"50 FIELDS BUT ARE LIMITED"+R$+"
    TO 254 CHARACTERS, PLUS"+R$
195 P$=P$+"THE RETURN AT THE END."+R$+"END
    "
200 RECORD#1,2
205 PRINT#1,P$
210 DCLOSE#1
215 REM
220 DIMF$(50)
225 REM ** JUMP TO MENU FOR TESTING
230 GOTO320
235 REM ** THE FOLLOWING IS THE ACTUAL USA
    GE ROUTINE
240 ML=PEEK(52)+1+256*PEEK(53)
245 DT=ML+76
250 DOPEN#1,"RELFILE"
255 TM=TI
260 RECORD#1,1
265 SYSML
270 PRINT"{CLEAR}{DOWN}STRING THING METHOD
     RUNNING{DOWN}"
275 NF=PEEK(DT)+1:POKEDT+NF,PEEK(139)+1
280 FORI=1TONF
285 F$(I)=MID$(A$,PEEK(DT+I-1)+1,PEEK(DT+I
    )-PEEK(DT+I-1)-1)
290 PRINTF$(I)
295 NEXTI
300 REM ** END OF ACTUAL ROUTINE
305 PRINT"{DOWN}THE TIME FOR INPUT, SEPARA
    TION, AND      PRINTING OF THE ABO
    VE WAS";
310 PRINT"{REV}"TI-TM"{OFF}JIFFIES."
315 DCLOSE#1
320 PRINT"{DOWN}TO TIME THESE OPTIONS HIT ~
    NUMBER DESIRED"
325 PRINT"{REV}BUT WAIT FOR DRIVE TO STOP ~
    FOR FAIR TEST{OFF}"
330 PRINT"{DOWN} 1-GET#"
335 PRINT" 2-BYTE COUNTING"
340 PRINT" 3-STRING THING WITH FIELD SEPAR
    ATOR"
345 PRINT" 4-QUIT"
350 GETR$:IFR$=""THEN350
355 IFVAL(R$)<1ORVAL(R$)>4THEN320
360 ONVAL(R$)GOTO370,440,240,495
365 REM ** STANDARD GET#
370 PRINT"{CLEAR}{DOWN}STANDARD GET#{DOWN}
    "
375 DOPEN#1,"RELFILE"
380 TM=TI
385 I=1
390 Q$=""
395 RECORD#1,2
400 GET#1,R$:IFR$=CHR$(13)THEN410
405 Q$=Q$+R$:GOTO400
410 IFQ$="END"THEN420
415 PRINTQ$:F$(I)=Q$:Q$="":I=I+1:GOTO400
420 PRINT"{DOWN}THE TIME WAS{REV}"TI-TM"{O
    OFF}JIFFIES FOR GET#."
425 DCLOSE#1
430 GOTO320
435 REM ** BYTE COUNT
440 PRINT"{CLEAR}{DOWN}BYTE COUNT METHOD N
    OW RUNNING{DOWN}"
445 DOPEN#1,"RELFILE"
450 TM=TI
455 BY=1:I=1
460 RECORD#1,2,(BY)
465 INPUT#1,F$(I)
470 BY=BY+LEN(F$(I))+1
475 IFF$(I)<>"END"THENPRINTF$(I):I=I+1:GOT
    O460
480 PRINT"{DOWN}THE TIME WAS{REV}"TI-TM"{
    OFF}JIFFIES AND THE COMMA  CAUSED
     A PROBLEM"
485 DCLOSE#1
490 GOTO320
495 STOP
```

©

MACHINE LANGUAGE

Jim Butterfield, Associate Editor

Part II

NUMERIC OUTPUT

This is the second in a three-part series on techniques of handling numeric displays or printouts in machine language.

Preparing decimal output can be done in a number of ways. The methods for converting binary integers to decimal can be summarized by direction: right-to-left or left-to-right. In both cases, there is usually a need to perform division. And don't forget that each digit must be converted to ASCII before it is output.

No matter which way we do the job, we need to plan the output format. A one-byte number might require three decimal digits to be printed (e.g., 255), but a two-byte number might need five digits (e.g., 65535). It's often a good idea to plan to output a fixed number of digits, since numbers may need to be printed neatly into columns or onto specific parts of the screen. We might also find it desirable to suppress leading zeros on a number so that 00307 becomes 307, with leading spaces.

Right-To-Left

The method goes something like this: divide by ten. The remainder is the rightmost digit. If the quotient is non-zero, repeat. Thus, a binary value of 287 is calculated: divide by 10, remainder 7; divide quotient 28 by 10, remainder 8; divide quotient 2 by 10, remainder 2. The quotient becomes zero at this point, so we have the three digits – 2, 8, and 7.

The digits come out backwards, however. In the above example, we can't print the 7 the moment we calculate it, since we must work out two earlier digits. That's not a problem, since the digits can be placed into a buffer area – or on the stack, for that matter.

Right-to-left is attractive because it automatically finds the number of digits that need to be printed; the procedure stops when a quotient of zero is reached. You can immediately spot numbers that are too big. It's also very easy to insert leading spaces to fill out the number to any desired length. You'll need a good divide-by-ten routine, of course.

Left-To-Right

This method takes a little more effort to set up, but generates digits in the "normal" order, which allows you to output them directly. Zero suppression adds a little extra code.

We must start by assuming the number of digits that we wish to output. Let's say, for example, that we expect up to three digits. We would follow roughly the following procedure:

Set FACTOR to 100;

Divide the number by FACTOR;

The quotient is the next digit;

Take the remainder, set FACTOR to 10, and repeat;

Then set FACTOR to 1 and repeat; or for that matter, the remainder from the last calculation will be your last digit.

To convert 287, we divide by 100; the quotient of 2 is our first digit. Take the remainder (87) and divide by 10; the quotient of 8 is the next digit. Finally, the remainder of 7 is our last digit whether or not we divide it by 1.

We can achieve this without a formal division routine; repeated subtraction will work efficiently enough for most purposes. We might change our algorithm to read:

Set FACTOR to 100;

Set COUNTER to 0;

If the number is greater than or equal to FACTOR, then subtract FACTOR from the number, add 1 to COUNTER, and repeat this step;

COUNTER now contains the first digit; you may print it.

Now set FACTOR to 10, COUNTER to 0, and repeat.

Our example of 287 would have 100 subtracted from it until it reached 87. The counter would have counted 2 subtractions, so we can

Products for Commodore, Atari, Apple, and others!

NEW

THE MONKEY WRENCH II

A PROGRAMMERS AID FOR ATARI 800

NEW AND IMPROVED — 18 COMMANDS

PLUGS INTO RIGHT CARTRIDGE SLOT

$59.95

If you are a person who likes to monkey around with the ATARI 800, then THE MONKEY WRENCH II is for you!! Make your programming tasks easier, less time-consuming and more fun. Why spend extra hours working on a BASIC program when the MONKEY WRENCH can do it for you in seconds. It can also make backup copies of boot type cassette programs. Plugs into the right slot and works with ATARI BASIC cartridge.

The MONKEY WRENCH provides 18 direct mode commands. They are: AUTO LINE NUMBERING — Provides new line numbers when entering BASIC program lines. RENUMBER — Renumbers BASIC's line numbers including internal references. DELETE LINE NUMBERS — Removes a range BASIC line numbers. VARIABLES — Display all BASIC variables and their current value. Scrolling — Use the START & SELECT keys to display BASIC lines automatically. Scroll up or down BASIC program. FIND STRING — Find every occurrence of a string, XCHANGE STRING — Find every occurrence of a string and replace it with another string. MOVE LINES — Move lines from one part of program to another part of program. COPY LINES — Copy lines from one part of program to another part of program. FORMATTED LIST — Print BASIC program in special line format and automatic page numbering. DISK DIRECTORY — Display Disk Directory. CHANGE MARGINS — Provides the capability to easily change the screen margins. MEMORY TEST — Provides the capability to test RAM memory. CURSOR EXCHANGE — Allows usage of the cursor keys without holding down the CTRL key. UPPER CASE LOCK — Keeps the computer in the upper case character set. HEX CONVERSION — Converts a hexadecimal number to a decimal number. DECIMAL CONVERSION — Converts a decimal number to a hexadecimal number. MONITOR — Enter the machine language monitor.

In addition to the BASIC commands, the Monkey Wrench also contains a machine language monitor with 16 commands used to interact with the powerful features of the 6502 microprocessor.

NEW

VIC RABBIT CARTRIDGE AND CBM 64 RABBIT CARTRIDGE

"High-Speed Cassette Load and Save!"

$39.95 (includes Cartridge and Manual)

Expansion Connector on the VIC Cartridge

"Don't waste your Life away waiting to LOAD and SAVE programs on Cassete Deck."

Load or Save 8K in approximately 30 seconds! Try it — your Un-Rabbitized VIC takes almost 3 minutes. It's not only Fast but VERY RELIABLE.

Almost as fast as VIC Disk Drive! Don't be foolish — Why buy the disk when you can get the VIC Rabbit for much, much less!

Easy to install — it just plugs in.
Expansion Connector on rear.
Works with or without Expansion Memory.
Works with VIC Cassette Deck.
12 Commands provide other neat features.

Also Available for 2001, 4001, and 8032

Now for the "64"

STCP — 300/1200 Baud

Standard Terminal Communications Package

PFO IOD OOA CP<D1>D2 BELL = 12:30:00 10:14:36

Don't settle for non-standard Communications Protocol! Access Micro Net, Source, Bulletin Boards, Local Main-frame, etc.

- Complete Package — Includes RS232 Interface Board and software (does not include modem)
- Communicates in Industry Standard ASCII
- Upload/Download to/from Disk
- Automatic File Translation
- Can be controlled from keyboard or user supplied basic or machine language program

Specify 3.0 or 4.0 ROMS or 8032 Commodore Computer 4040 or 8050 or PEDISK II Disk or CBM64 on 1541.

Price: $129.95

PET BASIC SCROLL PROGRAM

Scroll thru Basic Programs using cursor up/down keys. Specify computer. $6.00 on cassette, $9.00 on diskette.

65C02 MAE

Same as our MAE but enhanced for the new 65C02 Opcodes. Turns your computer into a development system for the new ROCKWELL 65C02 Microprocessor. $200.00 — Specify Computer.

6800 CROSS ASSEMBLER

A Cross Assembler based on the MAE that runs on the PET, Apple, or Atari but assembles opcodes for the Motorola 6800 microprocessor. Turns your computer into a development system for the Motorola 6800 Microprocessor. $200.00 — Specify Computer.

ATARI and VIC Cartridges

EHS can supply large quantities of ATARI and VIC Cartridges for software developers. If you need cartridges, call for pricing.

More than just an Assembler/Editor!

Now for the "64"

It's a Professionally Designed Software Development System

MAE

for PET APPLE ATARI ~~$169.95~~ New Price $99.95

Blast off with the software used on the space shuttle project!

- Designed to improve Programmer Productivity.
- Similar syntax and commands — No need to relearn peculiar syntaxes and commands when you go from PET to APPLE to ATARI.
- Coresident Assembler/Editor — No need to load the Editor then the Assembler then the Editor, etc.
- Also includes Word Processor, Relocating Loader, and much more.
- Options: EPROM Programmer, unimplemented opcode circuitry.
- STILL NOT CONVINCED: Send for free spec sheet!

ATARI AND PET EPROM PROGRAMMER

Programs 2716 and 2532 EPROMs. Includes hardware and software. PET = $75.00 — ATARI (includes sophisticated machine language monitor) = $119.95

TRAP 65

TRAP 65 is a hardware device that plugs into your 6502's socket. Prevents execution of unimplemented opcodes and provides capability to extend the machines' instruction set. For PET/APPLE/SYM. Reduced from $149.95 to $69.95

5¼ INCH SOFT SECTORED DISKETTES

Highest quality. We use them on our PETs, APPLEs, ATARIs, and other computers. $22.50/10 or $44.50/20

Prowriter Printer - Excellent dot matrix print. Parallel = $489.00 Serial = $600.00 IEEE = $589.00

DC Hayes Smart Modem = $235.00
DC Hayes Micro Modem II = $289.00

Rana Disk Drive - 375
4 Drive Controller - 114

EPROMS 2716 = $4.50 2532 = $7.50
Over 40 Commodore Programs by Baker (on 4040) = $25.00

Eastern House

3239 Linda Dr.
Winston-Salem, N.C. 27106
(919) 924-2889 (919) 748-8446
Send for free catalog!

VISA MasterCard

send the digit 2 to output.

The various factors (1000, 100, 10, 1, or whatever is needed) may be stored in a table for quick reference rather than calculated. Using true division would be faster than our subtraction algorithm. But since we'll never need to subtract more than nine times for each digit (and since we're likely to spend much more time delivering the output digit to its destination), it's not much of a worry.

Mathematics fiends will tell you that the left-to-right procedure may be easily extended to generate decimal fractions. Useful, but only if you are using binary numbers with fractional parts in the first place.

An Example

Let's do some very quick code to output a dozen numbers from memory in decimal. We'll use the left-to-right method. Zero suppression won't be used. Address FFD2 will be used for output (PET/CBM/VIC/C64 compatible).

```
OUTPUT LDX  #$00      (number counter)
       STX  COUNT
 NXNUM LDA  $0350,X   (get mem value)
       LDY  #$02      (2+1 digits)
  LOOP CMP  TABLE,Y
       BCC  DONE
       SBC  TABLE,Y
       INC  COUNT
       BNE  LOOP
  DONE PHA            (add seven)
       LDA  COUNT
       ORA  #$30
       JSR  $FFD2
       LDA  #$00
       STA  COUNT
       PLA
       DEY
       BPL  LOOP
       LDA  #$0D
       JSR  $FFD2
       INX
       CPX  #$0A
       BCC  NXNUM
       RTS
 TABLE .BYTE          1,10,100
```

It's fun to do this in a practical example. Let's POKE it from BASIC:

```
100 DATA 162,0, 142,144,3, 189,80,3
110 DATA 160,2, 217,132,3, 144,8
120 DATA 249,132,3, 238,144,3
130 DATA 208,243, 72, 173,144,3, 9,48
140 DATA 32,210,255, 169,0, 141,144,3
150 DATA 104,136, 16,225, 169,13
160 DATA 32,210,255, 232, 224,10
170 DATA 144,210, 96, 1,10,100
200 FOR J=848 TO 902:READ X
210 T=T+X:POKE J,X
220 NEXT J
230 IF T<>6199 THEN STOP
300 SYS 848
```

It will take a few moments to POKE the program in place; after that, the decimal numbers come out with blinding speed (especially if you have cleared the screen so that there is no need for scrolling). The numbers, by the way, are the same values as in the DATA statements in line 100 and part of 110.

But there's more.

These are the conventional methods, and they have a number of variations that we haven't mentioned.

But there's a very fast and radically different method available on the 6502. It uses Decimal mode in an unusual way to generate decimal number output super fast.

More on that the next time around. ©

COMPUTE!'s
Machine Language For Beginners

Author: Richard Mansfield
Price: $12.95
On Sale: Now

One of the most exciting moments in computing is when a beginner writes his or her first program which actually works... usually after hours of effort. A new world opens up.

But as beginners grow into intermediate programmers and become more fluent in BASIC, they realize the language's limitations – slow speed, and the lack of total control over the inner operations of the computer. They often develop an admiration for the fast, smoothly running machine language programs that mark commercial software. Unfortunately, too many people view machine language as mysterious and forbidding, and they are reluctant to tackle it themselves.

COMPUTE! Books' latest release, *Machine Language For Beginners*, by Richard Mansfield, introduces newcomers to the challenges of machine language with a unique approach. Aimed at people who understand BASIC, *Machine Language For Beginners* uses BASIC to explain how machine language works. A whole section of the book explains machine language in terms of equivalent BASIC commands. If you know how to do it in BASIC, you can see how it's done in machine language.

Machine Language For Beginners is a general tutorial for all users of computers with 6502 microprocessors – with examples for the Commodore 64, VIC-20, Atari 400/800/1200XL, Apple II, and PET/CBM. The numerous machine language programs will work on all these computers.

As a bonus, *Machine Language For Beginners* includes something that all fledgling machine language programmers will need to get started – an assembler. The "Simple Assembler," written in BASIC for the various computers, takes the tedium out of entering and assembling short machine language programs. The book even explains how to use the built-in machine language monitors on several of the computers. And it includes a disassembler program and several monitor extensions.

This book fills the need for a solid, but understandable, guide for personal computing enthusiasts. Mansfield is Senior Editor of **COMPUTE!**. His monthly column, "The Beginner's Page," has been one of **COMPUTE!**'s most popular features.

In the **COMPUTE!** tradition, *Machine Language For Beginners* has been written and edited to be straightforward, clear, and easily understood. It is spiral-bound to lie flat to make it easier to type in programs.

Table of Contents

Available at computer dealers and bookstores nationwide. To order directly call TOLL FREE 800-334-0868. In North Carolina call 919-275-9809. Or send check or money order to **COMPUTE! Books**, P.O. Box 5406, Greensboro, NC 27403.

Add $2 shipping and handling. Outside the U.S. add $5 for air mail, $2 for surface mail. All orders prepaid, U.S. funds only.

Atari Reset Run

Thomas A. Marshall

You can easily disable your BREAK key, but here's how to prevent accidental crashes caused by the system RESET key. This short routine also illustrates some uses of the "dynamic keyboard" technique.

This "RESET RUN" program was developed from a need to disable the system reset button. My two-year-old son kept stopping his alphabet teaching program by hitting the system reset button. I had disabled the break key with POKE 16,64: POKE 53774,64 (back on with POKE 16,192:POKE 53774,192).

The program changes the system reset button vector so that the program merely *reruns* when the reset button is pressed. The program uses Atari's ability to vector (point) to where the computer should go when the system reset key is pressed. This is accomplished by POKEing the address of our machine language program at memory locations 12 and 13. The least significant byte goes to memory location 12, and the most significant byte goes to memory location 13. The program also uses Atari's ability to read from the screen with a POKE 842,13. With this POKE, the cursor will do continual RETURNs down the screen.

The trick is to place a RUN statement on the screen so that the cursor will do a RETURN on it. Then a POKE 842,12 in the BASIC program will turn off the returning action. This *dynamic programming* technique has many applications.

I decided to use this technique to convert the DATA statements into a single string and then delete the DATA statements. In this string form, the three lines can be easily entered at the beginning of any program. There are other ways to accomplish a program rerun, but the advantage of RESET RUN is that the machine language code is relocatable.

To place the RUN statement on the screen, the machine language program first hides all the action by setting the character color and intensity the same as the background (lines 100 and 110). It then clears the screen (lines 120 and 130), sets the cursor row position (lines 140 and 150), reads "RUN two up arrows" and places it on the screen (lines 160 to 210). Finally, it turns on the dynamic programming with POKE 842,13 (lines 220 and 230), and jumps to the BASIC cartridge.

The BASIC program POKEs the disk boot initialization vector locations 12 and 13 with the address of the R$ string where the machine language routine resides and turns off Atari's dynamic programming feature with the POKE 842,12. At the end, the GRAPHICS 0 statement returns the character intensity to normal. If you do *not* want the program to delete the DATA statements, *do not* type in lines 330 to 370, and in line 320 type "RUN" instead of "GOTO 330".

Note that once the system reset button is pressed, you cannot immediately use the SAVE command. When any disk operation is attempted, an ERROR 130, nonexistent device, results. To save the program, the initialization vector has to be reset with POKE12,64: POKE13,21. (If you have typed in the program, simply type GOTO 10 and press return.) Then press the system reset button and SAVE"D:filename". Also, note that if you immediately list the program after the initial run and then press the system reset button, the program will crash. In any case, it is always advisable to save any program prior to running it.

Reset Run

```
0 DIM R$(60)
1 GOTO 250
2 I=ADR(R$):H=INT(I/256):L=I-H*256:P
  OKE 12,L:POKE 13,H:POKE 842,12:GRA
  PHICS 0
5 ? "HI THERE, ATARI ENTHUSIAST!":EN
  D
10 POKE 12,64:POKE 13,21:? "NOW PRES
   S THE SYSTEM RESET BUTTON":? "AND
    SAVE THIS PROGRAM":END
100 DATA 169,148
110 DATA 141,197,2
120 DATA 169,125
130 DATA 32,164,246
140 DATA 169,2
150 DATA 133,84
160 DATA 169,0
170 DATA 32,164,246
180 DATA 169,28
```

```
190 DATA 32,164,246
200 DATA 169,28
210 DATA 32,164,246
220 DATA 169,13
230 DATA 141,74,3
240 DATA 76,0,160
250 DIM INSTR$(3):INSTR$="RUN"
260 S=LEN(INSTR$):FOR I=1 TO 16:READ
    T:R$(I,I)=CHR$(T):NEXT I
270 K=0:FOR I=17 TO 17+5*(S-1) STEP
    5:RESTORE 160
280 FOR J=0 TO 4:READ T:R$(I+J,I+J)=
    CHR$(T):NEXT J
290 K=K+1:R$(I+1,I+1)=INSTR$(K,K):NE
    XT I
300 FOR I=22+5*(S-1) TO 39+5*(S-1):R
    EAD T:R$(I,I)=CHR$(T):NEXT I
310 ? CHR$(125):POKE 766,1:POSITION
    2,5:? "1 R$(1,";I-1;")=";CHR$(34
    );R$;CHR$(34):POKE 766,0
320 ? "POKE 842,12:GOTO 330":POKE 76
    6,0:POSITION 0,0:POKE 842,13:END

330 IMAX=90:GOLINE=340:LMAX=360
340 IMIN=IMAX+10:IMAX=IMAX+140:IF IM
    AX>=LMAX THEN IMAX=LMAX:GOLINE=3
    70
350 ? CHR$(125):? :FOR I=IMIN TO IMA
    X STEP 10:? I:NEXT I
360 ? "POKE 842,12:? :? :GOTO";GOLIN
    E:POSITION 0,0:POKE 842,13:END
370 ? CHR$(125):? :? "370":? "POKE 8
    42,12:? :RUN":POSITION 0,0:POKE
    842,13:END
```

©

Soft-16 For Commodore 64

Douglas D. Nicoll

This program, "USR(PEEK)", demonstrates several interesting concepts about managing the memory of the 64. BASIC programs can be run essentially without BASIC, and you can switch between ROM and RAM during a program RUN to access an additional 16K of RAM for data storage. You'll also see how to use the USR() statement.

An inexpensive 16K RAM expansion for the Commodore 64? Run BASIC programs without BASIC or the kernal? Well, almost. The 6510 has the capability of exchanging RAM data banks with the three ROM banks (BASIC [AB] $A000-$BFFF; characters [D] $D000-$DFFF; and kernal [EF] $E000-$F000). It switches between ROM and RAM with the use of a control port located at $0001. Bit zero in $0001 controls EF, bit one controls AB, and bit two controls D. Setting the bit to one switches in ROM (the normal state), and zero switches in RAM memory.

In normal BASIC operation, it is possible to POKE values to AB and EF locations, but PEEKing these locations will show *only* the ROM data. POKEs and PEEKs to D work fine, but you can't PEEK the character ROM without setting a number of switches so the system won't crash. Thus, without the ability to PEEK the hidden RAM memory, AB and EF locations are effectively eliminated from use in BASIC programs.

USR(PEEK) is a valuable machine language utility program that opens up the hidden RAM for use in BASIC programs, giving the user 16K more memory cells for data storage. The program is loaded into $C001-$C0E4, and uses $C000 as a temporary storage cell. The vector for the USR() function is set (POKE 785,1:POKE 786,192). BASIC programs are loaded normally, and any RAM location can be PEEKed by using X = USR(N), where X is any variable and N is any number from 1 to 65535. Any number less than 0.5 will set X to -1, 0.5 to 1.9 evaluate as 1, and all other decimal numbers are truncated to the integer. If a negative number is given for N, the value returned is for ABS(N). If a number is greater than 65535, then X is -1. If N is between 53248 and 57343, X is the value of data stored in character ROM (D).

Automatic Switching

How does USR(PEEK) work? The statement X = USR(N) in a BASIC program loads N into the floating point accumulator and sends the computer to the machine language program pointed to by the USR vector. The machine language program evaluates the number in the FP accumulator, switches out BASIC and kernal ROM, loads the desired RAM data into the FP accumulator, switches BASIC and kernal ROM back in, and finally sets up the FP accumulator so that X contains the values on return to the BASIC program. When character ROM is desired, it is switched in for the manipulation.

The techniques used to dynamically switch between RAM and ROM have many other uses for hybrid programmers (people who use both BASIC and machine language). For example, machine language programs can be loaded under BASIC or kernal ROM and run with BASIC programs – this leaves more space for BASIC programs and variable storage. It is possible to envision loading a BASIC program editor under BASIC ROM and calling it for renumbering, searching, etc.

Type in the program and, after saving a copy, RUN it to see a demonstration of how easy it is to use. Then eliminate lines 10-540 and save it with the name USR(PEEK). To use with your programs, LOAD and RUN USR(PEEK) and then LOAD and RUN your own BASIC programs that can be constructed to utilize the additional 16K of RAM data storage.

```
1 GOSUB1000:REM SET UP USR(PEEK)
10 PRINT"{CLEAR}USR(PEEK) AT CHARACTER ROM"
20 V$="{HOME}{24 DOWN}"
30 H$=""+"{39 RIGHT}"
40 UC=53248:LC=55296:GC=53760
50 H=0:V=10:L=83*8+UC:GOSUB500
60 H=8:V=10:L=3*8+UC:GOSUB500
70 H=14:V=5:L=85*8+UC:GOSUB500:H=14:V=14:
   L=74*8+UC:GOSUB500
```

```
80 H=22:V=10:L=54*8+UC:GOSUB500
90 H=30:V=10:L=52*8+UC:GOSUB500
100 PRINTLEFT$(V$,5);LEFT$(H$,18);"SC{UP}U
    {02 DOWN}{LEFT}J{UP}64";LEFT$(V$,22)
110 PRINT"PRESS ANY KEY TO CONTINUE";
120 GETA$:IFA$=""THEN120
130 PRINT"{CLEAR}USR(PEEK) INTO BASIC HIDD
    EN RAM"
140 PRINTLEFT$(V$,5);"INPUT 10 NUMERS(0-25
    5) TO STORE IN $A000TO $A00A :"
150 FORI=1TO10
160 PRINT"NUMBER ";I;": ";:INPUT"";X
170 IFINT(X)<>XORX<0ORX>255THENPRINT"INVAL
    ID ENTRY...":GOTO160
180 POKE40959+I,X:NEXT
190 PRINT"{CLEAR}USR(PEEK) INTO HIDDEN BAS
    IC RAM"
200 PRINT:PRINT:PRINT"LOCATION   PEEK   US
    R(PEEK)"
205 PRINT"----------------------------"
210 FORI=1TO10:PRINTI+40959,PEEK(I+40959),
    USR(I+40959):NEXT
220 PRINTLEFT$(V$,22);"PRESS ANY KEY TO CO
    NTINUE ";
230 GETA$:IFA$=""THEN230
240 PRINT"{CLEAR}USR(PEEK) INTO KERNAL HID
    DEN RAM"
250 PRINTLEFT$(V$,5);"INPUT 10 NUMERS(0-25
    5) TO STORE IN $F000TO $F00A :"
260 FORI=1TO10
270 PRINT"NUMBER ";I;": ";:INPUT"";X
280 IFINT(X)<>XORX<0ORX>255THENPRINT"INVAL
    ID ENTRY...":GOTO160
290 POKE61439+I,X:NEXT
300 PRINT"{CLEAR}USR(PEEK) INTO HIDDEN KER
    NAL RAM"
310 PRINT:PRINT:PRINT"LOCATION   PEEK   US
    R(PEEK)"
320 PRINT"----------------------------"
330 FORI=1TO10:PRINTI+61439,PEEK(I+61439),
    USR(I+61439):NEXT
340 END
500 FORJ=LTOL+7:X$="":X=USR(J)
510 FORI=7TO0STEP-1:IFX=>2↑ITHENX=X-2↑I:X$
    =X$+"{WHT}{REV} {OFF}":GOTO530
520 X$=X$+"{RIGHT}"
530 NEXTI:IFJ=LTHENPRINTLEFT$(V$,V);
540 PRINTLEFT$(H$,H);X$:NEXT:RETURN
1000 POKE785,1:POKE786,192:REM USR VECTOR
1010 FORI=49153TO49380:READX:POKEI,X:NEXT
1015 RETURN
1020 DATA173,97,0,201,144,208,3,76,188,192,
     56,201,128,176,3,76,163,192,201,145
1030 DATA144,3,76,163,192,73,128,141,97,0,5
     6,169,16,237,97,0,240,13,170,24
1040 DATA78,98,0,110,99,0,202,224,0,208,244
     ,173,98,0,141,78,192,173,99,0
1050 DATA141,77,192,173,1,0,141,0,192,120,7
     3,7,141,1,0,173,255,255,141,98,0
1060 DATA173,0,192,141,1,0,88,173,98,0,201,
     0,208,3,76,140,192,162,8,173,98,0
1070 DATA24,42,176,5,202,224,0,208,247,106,
     141,98,0,73,128,141,102,0,138
1080 DATA9,128,141,97,0,169,0,141,99,0,141,
     100,0,141,101,0,96,169,0,141,97,0
1090 DATA141,99,0,141,100,0,141,101,0,141,1
     02,0,169,128,141,98,0,96,169,129
1100 DATA141,97,0,169,128,141,98,0,141,102,
     0,169,0,141,99,0,141,100,0,141,101,0
1110 DATA96,56,173,98,0,201,224,144,3,76,22
     3,192,201,208,176,3,76,223,192,169,4
1120 DATA141,72,192,173,97,0,32,26,192,169,
     7,141,72,192,96,173,97,0,76,11,192
```

©

Minefield

Sean Igo

Your job is to get your trucks in quickly, defuse the bombs (especially the flashing ones which are about to go off), and get out as fast as you can. This game has four skill levels, and there are versions for VIC, 64, Apple, and PET.

In this game, you drive a truck around to gather and defuse time bombs before they explode – all the while avoiding mines and bomb craters.

Playing The Game

You find yourself in the center of a small minefield with several bombs, represented by circles, and a generous number of mines, shown as X's. Your truck is a diamond. To defuse the bombs, just run over them with the truck.

When the bombs first appear, they are innocent-looking little circles. After a short time – the rate varies from bomb to bomb – they turn reverse-field. This means *watch it*. Soon they begin to blink and you have only a few blinks to defuse them before they explode. Any mines (or heroic defusing teams) caught in the explosion will be instantly lost. Bombs caught in the explosion will explode, whether they were ready to or not.

Notes For VIC-20, C-64, And Apple Versions Of "Minefield"

The VIC version will run on any VIC with at least a 3K expansion. Both the VIC and 64 versions are designed to be used with a joystick although the 64 also can be played using certain keys on the keyboard (see the instructions within the program). Since the Apple computer doesn't have a realtime clock, a loop is incremented and checked in this version to provide the necessary delay before the bombs explode. The familiar "I," "K," "J," and "M" keys are used in the Apple version to move the truck around the screen.

Your truck can move in only four directions. It can "wrap around" all four edges of the screen. Don't run it into the mines or the craters (*) left by the bombs or your truck will be destroyed. Once you begin moving, your truck cannot stop until it is blown up or until the current minefield is cleared of bombs.

Skill Levels And Scoring

Minefield has four skill levels. Skill levels differ only in the number of trucks you get. Level 0, the easiest, has four trucks. Level one has three. Level two has two and level three has one.

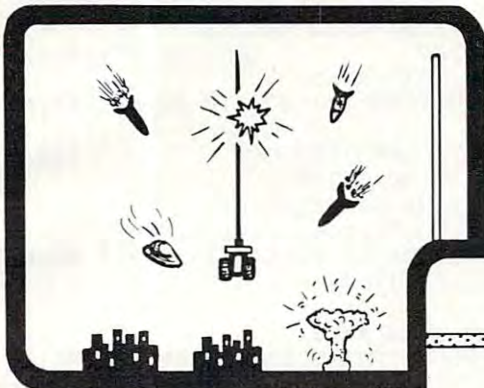
Scoring: **10 points for a normal bomb**
20 points for a reverse-field bomb
30 points for a blinking bomb
-10 points at the end of an explosion for every bomb that went off. This is incentive to defuse more than one or two bombs in the later explosions.

Program 1: Minefield – PET/CBM Version

```
10 REM   *** MINEFIELD FOR 40 COLUMN PETS
   ***
50 GOSUB 1130
60 REM ---INITIALIZE VARIABLES---
70 DIM BT(37),B3(37),B4(37),BP(37),BS(37)
   ,XM(4),YM(4),BC(25)
80 DEF FNY(X)=INT((X-32768)/40)
90 DEF FNX(X)=(X-40*FNY(X))-32768
100 DEF FNS(X)=32768+PX+40*PY
110 DEF FNP(X)=33051+INT(34*RND(1))+40*INT
    (15*RND(1))
120 DEF FNN(X)=PEEK(FNS(X))
130 FORJ=1 TO 4:READ XM(J),YM(J):NEXT
140 DATA Ø,1,-1,Ø,1,Ø,Ø,-1
150 SC=Ø:BT=168Ø:NB=4:NW=Ø
160 PRINT"{CLEAR}";:POKE 59468,12
170 PRINT"{REV}MINE****] SCORE: Ø"
180 PRINT"{REV}********]{RIGHT}HI SCORE:";
    HS
190 PRINT"{REV}***FIELD]{RIGHT}WAVE: 1"
200 PRINT"{REV}          ]{RIGHT}";:IF NL<>1
     THEN FORJ=1 TO NL-1:PRINT"Z";:NE
    XT
210 FORJ=32768 TO 32927:IFPEEK(J)=32 THEN ~
    POKE J,16Ø
220 NEXT
230 XP$="U]I{DOWN}{Ø4 LEFT}UU]II{DOWN}{Ø6
    LEFT}UUU]III{DOWN}{Ø7 LEFT}@@@*@@
    @{DOWN}{Ø7 LEFT}JJJ]KKK"
235 XP$=XP$+"{DOWN}{Ø6 LEFT}JJ]KK{DOWN}{Ø4
     LEFT}J]K"
240 S$="{HOME}{24 DOWN}"
250 Q$="{4Ø RIGHT}"
260 XR$="   {DOWN}{Ø4 LEFT}     {DOWN}{Ø6
    LEFT}       {DOWN}{Ø7 LEFT}   *  ~
     {DOWN}{Ø7 LEFT}       "
```

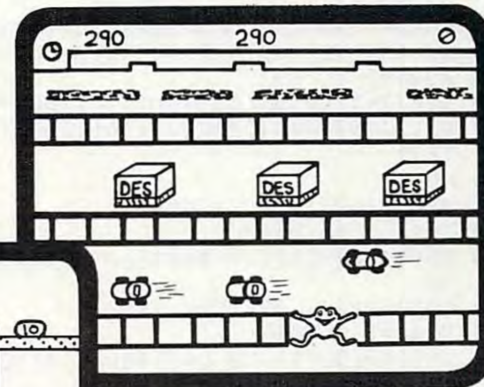
D.E.S.-SOFT ™
a division of
DES-Data Equipment Supply Corp.

LASER COMMAND by Bob Burnett



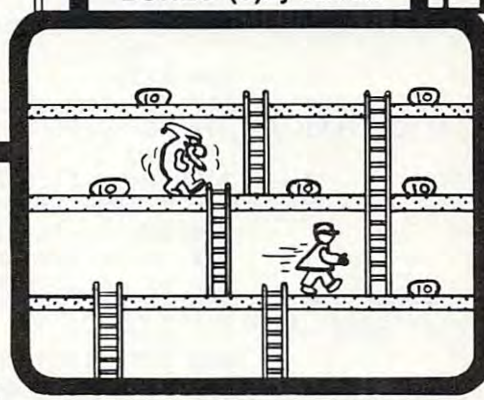
You are the commander of a squadron of laser ships. It is your duty to defend the cities of Earth against the alien onslaught. Spectacular graphics and machine code for super fast arcade fun. VIC-20 and cassette, joystick.
$20.00

HOPPER by Thomas Kim



Rated a Five Star game by Creative Computing. Avoid the cars, buildings, logs and other obstacles to bring the frog home. Machine language. VIC-20, cassette and joystick.
$20.00

BONZO (c)by Kavan



One of the most popular games in Europe. You control BONZO as he climbs the ladders and picks up point blocks. Watch out for the alien guards. Excellent graphics & sound. 100% machine code. VIC-20 w/8K expander, cassette. Joystick or keyboard.
$20.00

SHIFTY (c)by Kavan

Watch the maze change as you pass thru the revolving doors. This is a really cute one. Machine language. VIC-20 w/8K expander, cassette. Joystick and keyboard.
$20.00

SPACE TRADER
by Doug Caruthers

Search for the lost planet of Alantia. 1 to 4 players. Text adventure. VIC-20 w/16K expander and cassette.
$39.95

Many exciting titles to choose from.
New Software for the CBM B500/700 series available now !!
More to Come !!!

PAL 20 ™ **$10.00**

- 95 pages of aids, worksheets & logs
- Something for every VIC-20 programmer
- Coming soon for the Commodore 64

NEW for the VIC-20 & 64
electronic ab VIC-20 products

VIC FORTH **$59.95**
This is a highly capable language that operates from cartridge. It is based on Fig-Forth. Disk and cassette compatible. Vic Forth will work with any memory expansion. 3K of RAM is included in this cartridge.

VIC GRAPH **$49.95**
The intention of this program is to serve as a mathematical and pedagogic aid for studying complicated equations and functions by their graphs. Plots graphics in high resolution within an x-axis range defined by you. You can also "blow up" parts of a graph in detail by a specified range.

VIC STAT **$49.95**
Vic Stat is a cartridge which will simplify your work with statistics and graphic displays. It will add approximately 15 commands to BASIC. For example, bar chart, horizontal or vertical, plotting with 2024 points, printout of screen. Statistical commands for calculations of, for example, mean value, standard deviation, variance, etc.

VIC REL **$59.95**
The purpose of this cartridge is to simplify control of, for example, burglar alarms, garage doors, door locks, heating elements, lamps, radios, remote controllers, valves, pumps, telephones, accumulators, irrigation systems, electrical tools, stop watches, ventilators, humidifiers, etc., etc. This cartridge contains 6 relay outputs and 2 inputs of type optocoupler. For the VIC-20 and Commodore 64.

SEE YOUR LOCAL DEALER TO SEE OUR FINE PRODUCTS

Dealer Inquiries Invited *Software Distribution Available* *Programs Wanted*

(714) 778-5455
DES
Data Equipment Supply Corp.
8315 Firestone Blvd., Downey, CA 90241
(213) 923-9361

```
265 XR$=XR$+"{DOWN}{06 LEFT}       {DOWN}{04
    LEFT}   "
270 REM ---SET UP NEXT WAVE---
280 BG=0:NW=NW+1:IF NW>11 THEN 310
290 NB=NB+1.5:IF NW=1 THEN 330
300 IF NW<6 THEN BT=BT-180
310 PRINT"{HOME}{02 DOWN}{REV}";TAB(15);NW

320 POKE FNS(1),32:FORJ=1 TO NB:POKEBP(J),
    32:NEXT
325 FORJ=1 TO 25:POKE BC(J),32:NEXT
330 BN=INT(NB):FORJ=1 TO NB:BS(J)=1:NEXT
340 FORJ=1 TO NB
350 BT(J)=(.4+INT(61*RND(1))/100)*BT
360 B3(J)=BT(J)+.5*BT(J):B4(J)=B3(J)+.25*B
    T(J)
370 NEXT
380 PX=19:PY=15:POKE FNS(1),90
390 FORJ=1 TO NB
400 BP(J)=FNP(1):IF PEEK(BP(J))<>32 THEN 4
    00
410 POKE BP(J),87:NEXT:NN=0
415 FORJ=1 TO 25
416 BC(J)=FNP(1):IF PEEK(BC(J))<>32 THEN 4
    16
417 IF PEEK(BC(J)+1)=87 THEN 416
419 POKEBC(J),86:NEXT
420 GET R$:IF R$<>"" THEN 420
430 DR=0:TX=TI
440 REM ---GET COMMANDS---
450 GET R$:IF R$<"2" OR R$>"8" THEN 490
460 IF R$="3" OR R$="5" OR R$="7" THEN 490

470 DR=VAL(R$)/2
480 REM ---MOVE TRUCK---
490 IF DR=0 THEN 600
500 POKE FNS(1),32:PX=PX+XM(DR):PY=PY+YM(D
    R)
510 IF PX<0 THEN PX=39
520 IF PX>39 THEN PX=0
530 IF PY<4 THEN PY=24
540 IF PY>24 THEN PY=4
550 X=FNN(1)
560 IF X=32 THEN POKE FNS(1),90:GOTO 600
570 IF X=42 OR X=86 THEN 960
580 GOTO 890
590 REM ---UPDATE BOMBS---
600 NN=NN+1:IF NN>INT(NB)THEN NN=1
610 IF BS(NN)=0 THEN 600
620 TG=TI-TX
630 IF TG>B4(NN) THEN N1=NN:GOTO 720
640 IF BS(NN)>2 THEN 690
650 IF TG>BT(NN) THEN BS(NN)=2
660 IF TG>B3(NN) THEN BS(NN)=3
670 IF BS(NN)=1 THEN 450
680 IF BS(NN)=2 THEN POKE BP(NN),215:GOTO
    450
690 IF BS(NN)=3 THEN POKE BP(NN),87:BS(NN)
    =4:GOTO 450
700 IF BS(NN)=4 THEN POKE BP(NN),215:BS(NN
    )=3:GOTO 450
710 REM ---BOMB EXPLODES---
720 TQ=TI:PD=0
725 X$="{OFF}"+LEFT$(S$,FNY(BP(N1))-2)+LEF
    T$(Q$,FNX(BP(N1))-1)
730 BS(N1)=0:N2=0:PRINTX$;XP$;
740 FORJ=1 TO NB:X=PEEK(BP(J)):IF BS(J)=0
    THEN 760

750 IF X<>87 AND X<>215 AND X<>218 THEN N2
    =J
760 NEXT:IF FNN(1)<>90 AND FNN(1)<>218 THE
    N PD=1
770 PRINTX$;XR$;
780 FORJ=1TONB:IF PEEK(BP(J))=32 AND BS(J)
    <>0 THEN POKE BP(J),87-128*(BS(J)
    >1)
790 NEXT:BN=BN-1
800 IF PD=1 THEN 960
810 IF BN=0 THEN 840
820 IF N2=0 THEN TX=TX+(TI-TQ):GOTO 450
830 N1=N2:GOTO 725
840 PRINT"{HOME}{02 DOWN}{REV}";TAB(20);
850 FORJ=1 TO 20:PRINT"{REV}COMPLETED{09
    LEFT}";:FORK=1 TO 100:NEXT
860 PRINT"{REV}          {09 LEFT}";:FORK=1
     TO 100:NEXT:NEXT
870 SC=SC-10*(INT(NB)-BG):IF SC<0 THEN SC=
    0
880 PRINT"{04 LEFT}{03 UP}          {10
    LEFT}";SC:GOTO 280
885 REM ---BOMB GATHERED---
890 BG=BG+1:TQ=TI:POKE FNS(1),218
895 FORJ=1 TO NB:IF PEEK(BP(J))=218 THEN A
    J=BS(J):BS(J)=0
900 NEXT
910 IF AJ=4 THEN AJ=3
920 SC=SC+10*AJ:PRINT"{HOME}{REV}";TAB(16)
    ;SC
930 BN=BN-1:IF BN=0 THEN 840
940 TX=TX+(TI-TQ):GOTO 450
950 REM ---PLAYER DESTROYED---
960 TQ=TI:FORJ=1 TO 20:POKE FNS(1),42:FORK
    =1 TO 25:NEXT:POKE FNS(1),170
970 FORK=1 TO 25:NEXT:NEXT:POKE FNS(1),32:
    NL=NL-1
980 POKE 32897+NL,160:DR=0:PX=19:PY=15
990 IF NL=0 THEN 1045
1000 IF BN=0 THEN 840
1010 GET R$:IF R$<>"" THEN 1010
1020 FORJ=1TONB:IF PEEK(BP(J))=32 AND BS(J)
     <>0 THEN POKE BP(J),87-128*(BS(J)
     >1)
1030 NEXT
1040 POKE FNS(1),90:TX=TX+(TI-TQ):GOTO 450
1045 IF SC>HS THEN HS=SC:PRINT"{HOME}{DOWN}
     {REV}";TAB(19);HS
1050 FORJ=1 TO 1500:NEXT:PRINT"{HOME}{02
     DOWN}{REV}";TAB(20);"GAME OVER{
     DOWN}{09 LEFT}PLAY AGAIN?";
1060 PRINT"(Y/N){04 LEFT}";
1080 PRINT"{REV}Y/{OFF}N{03 LEFT}";
1081 FORJ=1 TO 99:NEXT
1082 PRINT"{OFF}Y{REV}/N{03 LEFT}";
1083 FORJ=1 TO 99:NEXT
1084 GET R$:IF R$="Y" THEN 1110
1090 IF R$<>"N" THEN 1080
1100 PRINT"{CLEAR}LATER ON!":END
1110 GOSUB 1130:GOTO 150
1120 REM ---INSTRUCTIONS---
1130 PRINT"{CLEAR}{REV}M I N E F I E L D":P
     OKE 59468,14
1140 PRINT"DO YOU NEED INSTRUCTIONS (Y/N)"
1150 GET R$:IF R$="N" THEN 1410
1160 IF R$<>"Y" THEN 1150
1170 PRINT"{CLEAR}{REV}M I N E F I E L D"
1180 PRINT"{DOWN}THE OBJECT OF THIS GAME IS
      TO PICK UP"
1190 PRINT"AS MANY BOMBS AS YOU CAN BEFORE
     THEY"
1200 PRINT"EXPLODE. TO PICK UP A BOMB, JUST
      RUN"
1210 PRINT"OVER IT WITH YOUR TRUCK."
1220 PRINT"BOMBS WILL EXPLODE AFTER A SHORT
      TIME."
1230 PRINT"IF A BOMB TURNS REVERSE-FIELD, B
     E CARE-"
```

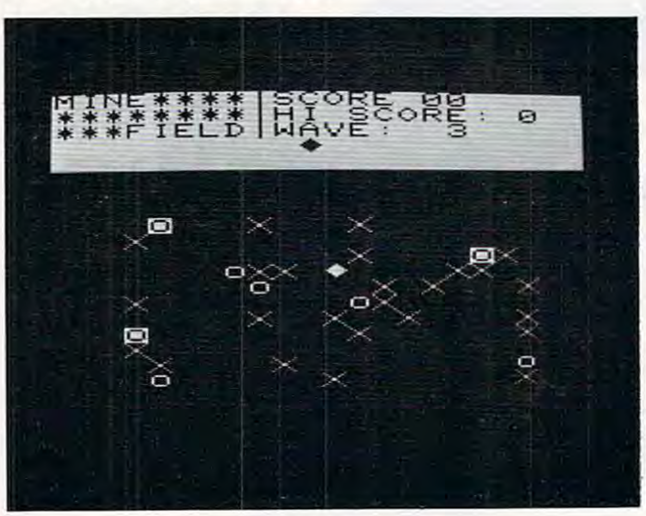
```
1240 PRINT"FUL WITH IT. IF IT STARTS TO BLI
    NK, IT"
1250 PRINT"WILL VERY SHORTLY EXPLODE-WATCH ~
    OUT!!"
1260 PRINT"BOMBS WILL CHAIN-REACT; ONE BOMB
     CAUGHT"
1270 PRINT"IN ANOTHER'S EXPLOSION WILL ALSO
     BLOW"
1280 PRINT"UP. IF YOU ARE CAUGHT IN A BOMB'
    S "
1290 PRINT"EXPLOSION, YOU WILL BE BLOWN UP.
    "
1300 PRINT"ALSO, DO NOT RUN INTO BOMB CRATE
    RS (*)"
1310 PRINT"OR MINES (X) OR YOU'LL BE TOTALL
    ED."
1320 PRINT"THE CONTOLS ARE: 8 TO GO UP"
1330 PRINT"                 4 TO GO LEFT"
1340 PRINT"                 6 TO GO RIGHT"
1350 PRINT"                 2 TO GO DOWN"
1360 PRINT"YOUR TRUCK CANNOT STOP ONCE YOU ~
    BEGIN"
1370 PRINT"MOVING. IT CAN WRAP-AROUND BOTH ~
    THE"
1380 PRINT"THE TOP AND SIDES OF THE SCREEN.
    "
1390 PRINT"{DOWN}PRESS RETURN TO CONTINUE";

1400 GET R$:IF R$<>CHR$(13) THEN 1400
1410 PRINT"{CLEAR}SELECT SKILL SETTING (0-3
    )"
1420 GET R$:IF R$<"0" OR R$>"3" THEN 1420
1430 NL=4-VAL(R$):RETURN
```

Program 2: Minefield – VIC Version

```
45 POKE36879,8
50 GOSUB1410
60 REM ---INITIALIZE VARIABLES---
70 DIM BT(37),B3(37),B4(37),BP(37),BS(37)
   ,XM(4),YM(4),BC(25)
80 DEF FNY(X)=INT((X-FNV(X))/22)
85 DEF FNV(X)=4*(PEEK(36866)AND128)+64*(P
   EEK(36869)AND120)
90 DEF FNX(X)=(X-22*FNY(X))-FNV(X)
95 DEF FNC(X)=37888+4*(PEEK(36866)AND128)
98 DEF FND(X)=FNC(X)-FNV(X)
100 DEF FNS(X)=FNV(X)+PX+22*PY
110 DEF FNP(X)=FNV(X)+178+INT(18*RND(1))+2
    2*INT(11*RND(1))
120 DEF FNN(X)=PEEK(FNS(X))
130 FORJ=1 TO 4:READ XM(J),YM(J):NEXT
140 DATA 1,0,-1,0,0,-1,0,1
150 SC=0:BT=(FNV(1)+360):NB=4:NW=0
160 PRINT"{CLEAR}";CHR$(142);
170 PRINT"{REV}{WHT}MINE****]SCORE: 0"
180 PRINT"{REV}{WHT}********]HI SCORE:";HS
    ;
190 PRINT"{HOME}{02 DOWN}{REV}***FIELD]WAV
    E: 1"
200 PRINT"{REV}{WHT}            ]{RIGHT}";:IF
    NL<>1 THEN FORJ=1 TO NL-1:PRINT"Z
    ";:NEXT
210 FORJ=FNV(1)TO FNV(1)+109:IFPEEK(J)=32
    THEN POKE J,160:POKEJ+FND(1),1
220 NEXT
230 XP$="{RED}{DOWN}{08 RIGHT}{07 LEFT}U]I
    {DOWN}{03 LEFT}U]I{DOWN}{07 LEFT}
    {04 RIGHT}{LEFT} @*@{DOWN}{03
    LEFT}J]K"
240 S$="{HOME}{22 DOWN}"
250 Q$="{WHT}{22 RIGHT}"
260 XR$="{WHT}    {DOWN}{04 LEFT}      {
    DOWN}{06 LEFT}       {DOWN}{07
    LEFT}    *  {DOWN}{05 LEFT}      ~
    "
270 REM ---SET UP NEXT WAVE---
280 BG=0:NW=NW+1:IF NW>11 THEN 310
290 NB=NB+1.5:IF NW=1 THEN 330
300 IF NW<6 THEN BT=BT-99
310 PRINT"{HOME}{02 DOWN}{REV}";TAB(15);NW
320 POKE FNS(1),32:FORJ=1 TO NB:POKEBP(J),
    32:NEXT
325 FORJ=1 TO 25:POKE BC(J),32:NEXT
330 BN=INT(NB):FORJ=1 TO NB:BS(J)=1:NEXT
340 FORJ=1 TO NB
350 BT(J)=((.4+INT(61*RND(1))/100)/.75)*BT
360 B3(J)=BT(J)+.3*BT(J):B4(J)=B3(J)+.20*B
    T(J)
370 NEXT
380 PX=11:PY=11:POKE FNS(1),90:POKE(FNS(1)
    +FND(1)),1
390 FORJ=1 TO NB
400 BP(J)=(FNP(1)):IFPEEK(BP(J))<>32 THEN ~
    400
410 POKE BP(J),87:POKEBP(J)+FND(1),1:NEXT:
    NN=0
415 FORJ=1 TO 25
416 BC(J)=FNP(1):IF PEEK(BC(J))<>32 THEN 4
    16
417 IF PEEK(BC(J)+1)=87 THEN 416
419 POKEBC(J),86:POKEBC(J)+FND(1),5:NEXT
420 GET R$:IF R$<>"" THEN 420
430 DR=0:TX=TI
440 REM ---GET COMMANDS---
445 DR=2
450 POKE37154,127:R=NOTPEEK(37151)AND60-((
    PEEK(37152)AND128)=0)
453 IFPEEK(37152)=119THENR=1
454 POKE37154,255:R=R*2
455 DR=INT(R):IFR<>0THENDR=INT(LOG(DR)/LOG
    (2)):IFDR=5THENDR=2
460 IFPEEK(37152)=119THENDR=2
470 IFR=0THEN600
475 IFDR>4THENDR=0
480 REM ---MOVE TRUCK---
500 POKE FNS(1),32:PX=PX+XM(DR):PY=PY+YM(D
    R)
510 IF PX<0 THEN PX=21
520 IF PX>21 THEN PX=0
530 IF PY<5 THEN PY=22
540 IF PY>22 THEN PY=5
550 X=FNN(1)
560 IF X=32 THEN POKE FNS(1),90:POKE(FNS(1
    )+FND(1)),1:GOTO 600
570 IF X=42 OR X=86 THEN 960
580 GOTO 890
590 REM ---UPDATE BOMBS---
600 NN=NN+1:IF NN>INT(NB)THEN NN=1
610 IF BS(NN)=0 THEN 600
620 TG=INT(TI*1.03)-TX
630 IF TG>B4(NN) THEN N1=NN:GOTO 720
640 IF BS(NN)>2 THEN 690
650 IF TG>BT(NN) THEN BS(NN)=2
660 IF TG>B3(NN) THEN BS(NN)=3
670 IF BS(NN)=1 THEN 450
680 IF BS(NN)=2 THEN POKE BP(NN),215:POKEB
    P(NN)+FND(1),1:GOTO 450
690 IF BS(NN)=3 THEN POKE BP(NN),87:POKEBP
    (NN)+FND(1),1:BS(NN)=4:GOTO 450
700 IF BS(NN)=4 THEN POKE BP(NN),215:POKEB
    P(NN)+FND(1),1:BS(NN)=3:GOTO 450
710 REM ---BOMB EXPLODES---
720 TQ=TI:PD=0
725 X$="{OFF}"+LEFT$(S$,FNY(BP(N1))-2)+LEF
    T$(Q$,FNX(BP(N1))-1)
```

```
730 BS(N1)=Ø:N2=Ø:PRINTX$;XP$;
740 FORJ=1 TO NB:X=PEEK(BP(J)):IF BS(J)=ØT
    HEN76Ø
750 IFX<>87ANDX<>215ANDX<>218THENN2=J
760 NEXT:IFFNN(1)<>9ØANDFNN(1)<>218THENPD=
    1
770 PRINTX$;XR$;
780 FORJ=1TONB:IF PEEK(BP(J))=32 AND BS(J)
    <>Ø THEN POKE BP(J),87-128*(BS(J)
    >1)
790 NEXT:BN=BN-1
800 IF PD=1 THEN 96Ø
810 IF BN=Ø THEN 84Ø
820 IF N2=Ø THEN TX=TX+(TI-TQ):GOTO 45Ø
830 N1=N2:GOTO 725
840 PRINT"{HOME}{Ø2 DOWN}{REV}";TAB(2Ø);
850 FORJ=1 TO 11:PRINT"{Ø2 RIGHT}{REV}COMP
    LETED{11 LEFT}";:FORK=1 TO 1ØØ:NE
    XT
860 PRINT"{REV}            {11 LEFT}";:FORK
    =1 TO 1ØØ:NEXT:NEXT
870 SC=SC-1Ø*(INT(NB)-BG):IF SC<Ø THEN SC=
    Ø
880 PRINT"{HOME}{REV}";TAB(14);SC;:GOTO 28
    Ø
885 REM ---BOMB GATHERED---
890 BG=BG+1:TQ=TI:POKE FNS(1),218:POKE(FNS
    (1)+FND(1)),1
895 FORJ=1 TO NB:IF PEEK(BP(J))=218 THEN A
    J=BS(J):BS(J)=Ø
900 NEXT
910 IF AJ=4 THEN AJ=3
920 SC=SC+1Ø*AJ:PRINT"{HOME}{REV}";TAB(14)
    ;SC
930 BN=BN-1:IF BN=Ø THEN 84Ø
940 TX=TX+(TI-TQ):GOTO 45Ø
950 REM ---PLAYER DESTROYED---
960 TQ=TI:FORJ=1 TO 2Ø:POKE FNS(1),42:FORK
    =1 TO 25:NEXT:POKE FNS(1),17Ø
970 FORK=1 TO 25:NEXT:NEXT:POKE FNS(1),32:
    NL=NL-1
980 POKE FNV(1)+75+NL,16Ø:DR=Ø:PX=11:PY=11
990 IF NL=Ø THEN 1Ø45
1000 IF BN=Ø THEN 84Ø
1010 GET R$:IF R$<>"" THEN 1Ø1Ø
1020 FORJ=1TONB:IF PEEK(BP(J))=32 AND BS(J)
     <>Ø THEN POKE BP(J),87-128*(BS(J)>1)
1030 NEXT
1040 POKE FNS(1),9Ø:TX=TX+(TI-TQ):GOTO 45Ø
1045 IF SC>HS THEN HS=SC:PRINT"{HOME}{DOWN}
     {REV}";TAB(17);HS;
```



"Minefield," VIC-20 version.

```
1050 FORJ=1 TO 15ØØ:NEXT:PRINT"{HOME}{WHT}{
     Ø4 DOWN}{REV}";"GAME OVER{DOWN}{
     WHT}{Ø9 LEFT}PLAY AGAIN?";
1060 PRINT"(Y/N){Ø4 LEFT}";
1080 PRINT"{REV}Y/{OFF}N{Ø3 LEFT}";
1081 FORJ=1 TO 99:NEXT
1082 PRINT"{OFF}Y{REV}/N{Ø3 LEFT}";
1083 FORJ=1 TO 99:NEXT
1084 GET R$:IF R$="Y" THEN 111Ø
1090 IF R$<>"N" THEN 1Ø8Ø
1100 PRINT"{CLEAR}{WHT}LATER ON!":END
1110 GOSUB141Ø:GOTO 15Ø
1400 GET R$:IF R$<>CHR$(13) THEN 14ØØ
1410 PRINTCHR$(14);"{CLEAR}S{WHT}ELECT SKIL
    L SETTING (Ø-3)"
1420 GET R$:IF R$<"Ø" OR R$>"3" THEN 142Ø
1430 NL=4-VAL(R$):RETURN
```

Program 3: Minefield – 64 Version

```
30 REM MINEFIELD FOR C-64
45 POKE5328Ø,Ø:POKE53281,Ø
50 GOSUB 113Ø
60 REM ---INITIALIZE VARIABLES---
70 DIM BT(37),B3(37),B4(37),BP(37),BS(37)
   ,XM(4),YM(4),BC(25)
80 DEF FNY(X)=INT((X-1Ø24)/4Ø)
90 DEF FNX(X)=(X-4Ø*FNY(X))-1Ø24
100 DEF FNS(X)=1Ø24+PX+4Ø*PY
110 DEF FNP(X)=13Ø7+INT(34*RND(1))+4Ø*INT(
    15*RND(1))
120 DEF FNN(X)=PEEK(FNS(X))
130 FORJ=1 TO 4:READ XM(J),YM(J):NEXT
140 DATA Ø,-1,Ø,1,-1,Ø,1,Ø
150 SC=Ø:BT=168Ø:NB=4:NW=Ø:D=54272
160 PRINT"{CLEAR}";:POKE 53272,21
170 PRINT"{REV}{WHT}MINE****] SCORE: Ø"
180 PRINT"{REV}{WHT}********]{RIGHT}HI SCO
    RE:";HS
190 PRINT"{REV}{WHT}***FIELD]{RIGHT}WAVE: ~
    1"
200 PRINT"{REV}{WHT}           ]{RIGHT}";:IF ~
    NL<>1 THEN FORJ=1 TO NL-1:PRINT"Z
    ";:NEXT
210 FORJ=1Ø24 TO 1183:IFPEEK(J)=32 THEN PO
    KE J,16Ø:POKEJ+D,1
220 NEXT
230 XP$="{RED}U]I{DOWN}{Ø4 LEFT}UU]II{
    DOWN}{Ø6 LEFT}UUU]III{DOWN}{Ø7 LE
    LEFT}@@@*@@@{DOWN}{Ø7 LEFT}JJJ]KK
    K"
235 XP$=XP$+"{RED}{DOWN}{Ø6 LEFT}JJ]KK{
    DOWN}{Ø4 LEFT}J]K"
240 S$="{HOME}{24 DOWN}"
250 Q$="{WHT}{4Ø RIGHT}"
260 XR$="{WHT}   {DOWN}{Ø4 LEFT}     {
    DOWN}{Ø6 LEFT}       {DOWN}{Ø7
    LEFT}   *   {DOWN}{Ø7 LEFT}      ~
     "
265 XR$=XR$+"{DOWN}{Ø6 LEFT}     {DOWN}{Ø4
     LEFT}   "
270 REM ---SET UP NEXT WAVE---
280 BG=Ø:NW=NW+1:IF NW>11 THEN 31Ø
290 NB=NB+1.5:IF NW=1 THEN 33Ø
300 IF NW<6 THEN BT=BT-18Ø
310 PRINT"{HOME}{Ø2 DOWN}{REV}";TAB(15);NW
320 POKE FNS(1),32:FORJ=1 TO NB:POKEBP(J),
    32:NEXT
325 FORJ=1 TO 25:POKE BC(J),32:NEXT
330 BN=INT(NB):FORJ=1 TO NB:BS(J)=1:NEXT
340 FORJ=1 TO NB
350 BT(J)=(.4+INT(61*RND(1))/1ØØ)*BT
360 B3(J)=BT(J)+.5*BT(J):B4(J)=B3(J)+
```

```
    T(J)
370 NEXT
380 PX=19:PY=15:POKE FNS(1),90:POKEFNS(1)+
    D,1
390 FORJ=1 TO NB
400 BP(J)=FNP(1):IF PEEK(BP(J))<>32 THEN 4
    00
410 POKE BP(J),87:POKEBP(J)+D,8:NEXT:NN=0
415 FORJ=1 TO 25
416 BC(J)=FNP(1):IF PEEK(BC(J))<>32 THEN 4
    16
417 IF PEEK(BC(J)+1)=87 THEN 416
419 POKEBC(J),86:POKEBC(J)+D,5:NEXT
420 GET R$:IF R$<>"" THEN 420
430 DR=0:TX=TI
440 REM ---GET COMMANDS---
450 R=(15-(PEEK(56321)AND15))*2
460 IFR<>0THENDR=LOG(R)/LOG(2)

470 IFR=0THEN490
480 REM ---MOVE TRUCK---
490 IF DR=0 THEN 600
500 POKE FNS(1),32:PX=PX+XM(DR):PY=PY+YM(D
    R)
510 IF PX<0 THEN PX=39
520 IF PX>39 THEN PX=0
530 IF PY<4 THEN PY=24
540 IF PY>24 THEN PY=4
550 X=FNN(1)
560 IF X=32 THEN POKE FNS(1),90:POKEFNS(1)
    +D,1:GOTO 600
570 IF X=42 OR X=86 THEN 960
580 GOTO 890
590 REM ---UPDATE BOMBS---
600 NN=NN+1:IF NN>INT(NB)THEN NN=1
610 IF BS(NN)=0 THEN 600
620 TG=TI-TX
630 IF TG>B4(NN) THEN N1=NN:GOTO 720
640 IF BS(NN)>2 THEN 690
650 IF TG>BT(NN) THEN BS(NN)=2
660 IF TG>B3(NN) THEN BS(NN)=3
670 IF BS(NN)=1 THEN 450
680 IF BS(NN)=2 THEN POKE BP(NN),215:POKEB
    P(NN)+D,1:GOTO 450
690 IF BS(NN)=3 THEN POKE BP(NN),87:POKEBP
    (NN)+D,1:BS(NN)=4:GOTO 450
700 IF BS(NN)=4 THEN POKE BP(NN),215:POKEB
    P(NN)+D,1:BS(NN)=3:GOTO 450
710 REM ---BOMB EXPLODES---
720 TQ=TI:PD=0
725 X$="{OFF}"+LEFT$(S$,FNY(BP(N1))-2)+LEF
    T$(Q$,FNX(BP(N1))-1)
730 BS(N1)=0:N2=0:PRINTX$;XP$;
740 FORJ=1 TO NB:X=PEEK(BP(J)):IF BS(J)=0 ~
    THEN 760
750 IF X<>87 AND X<>215 AND X<>218 THEN N2
    =J
760 NEXT:IF FNN(1)<>90 AND FNN(1)<>218 THE
    N PD=1
770 PRINTX$;XR$;:GR=129     :GOSUB2000
780 FORJ=1TONB:IF PEEK(BP(J))=32 AND BS(J)
    <>0 THEN POKE BP(J),87-128*(BS(J)>1)
790 NEXT:BN=BN-1
800 IF PD=1 THEN 960
810 IF BN=0 THEN 840
820 IF N2=0 THEN TX=TX+(TI-TQ):GOTO 450
830 N1=N2:GOTO 725
840 PRINT"{HOME}{02 DOWN}{REV}";TAB(20);
850 FORJ=1 TO 20:PRINT"{REV}COMPLETED{09
    LEFT}";:FORK=1 TO 100:NEXT
860 PRINT"{REV}                {09 LEFT}";:FORK=1
     TO 100:NEXT:NEXT
870 SC=SC-10*(INT(NB)-BG):IF SC<0 THEN SC=0
880 PRINT"{04 LEFT}{03 UP}           {10
    LEFT}";SC:GOTO 280
885 REM ---BOMB GATHERED---
890 BG=BG+1:TQ=TI:POKE FNS(1),218
895 FORJ=1 TO NB:IF PEEK(BP(J))=218 THEN A
    J=BS(J):BS(J)=0
900 NEXT
910 IF AJ=4 THEN AJ=3
920 SC=SC+10*AJ:PRINT"{HOME}{REV}";TAB(16)
    ;SC
930 GR=33:GOSUB2000:BN=BN-1:IF BN=0 THEN 8
    40
940 TX=TX+(TI-TQ):GOTO 450
950 REM ---PLAYER DESTROYED---
960 GR=129:GOSUB2000
961 TQ=TI:FORJ=1 TO 20:POKE FNS(1),42:FORK
    =1 TO 25:NEXT:POKE FNS(1),170
970 FORK=1 TO 25:NEXT:NEXT:POKE FNS(1),32:
    NL=NL-1
980 POKE 1153+NL,160:DR=0:PX=19:PY=15
990 IF NL=0 THEN 1045
1000 IF BN=0 THEN 840
1010 GET R$:IF R$<>"" THEN 1010
1020 FORJ=1TONB:IF PEEK(BP(J))=32 AND BS(J)
     <>0 THEN POKE BP(J),87-128*(BS(J)>1)
1030 NEXT
1040 POKE FNS(1),90:TX=TX+(TI-TQ):GOTO 450
1045 IF SC>HS THEN HS=SC:PRINT"{HOME}{DOWN}
     {REV}";TAB(19);HS
1050 FORJ=1 TO 1500:NEXT:PRINT"{HOME}{WHT}{
     03 DOWN}{REV}";TAB(20);"GAME OVER
     {DOWN}{WHT}{09 LEFT}PLAY AGAIN?";
1060 PRINT"(Y/N){04 LEFT}";
1080 PRINT"{REV}Y/{OFF}N{03 LEFT}";
1081 FORJ=1 TO 99:NEXT
1082 PRINT"{OFF}Y{REV}/N{03 LEFT}";
1083 FORJ=1 TO 99:NEXT
1084 GET R$:IF R$="Y" THEN 1110
1090 IF R$<>"N" THEN 1080
1100 PRINT"{CLEAR}{WHT}LATER ON!":END
1110 GOSUB 1130:GOTO 150
1120 REM ---INSTRUCTIONS---
1130 PRINT"{CLEAR}{REV}{WHT}M I N E F I E L
     D":POKE 53272,23
1140 PRINT"{WHT}DO YOU NEED INSTRUCTIONS (Y
     /N)"
1150 GET R$:IF R$="N" THEN 1410
1160 IF R$<>"Y" THEN 1150
1180 PRINT"{CLEAR}{WHT}{DOWN}THE OBJECT OF ~
```

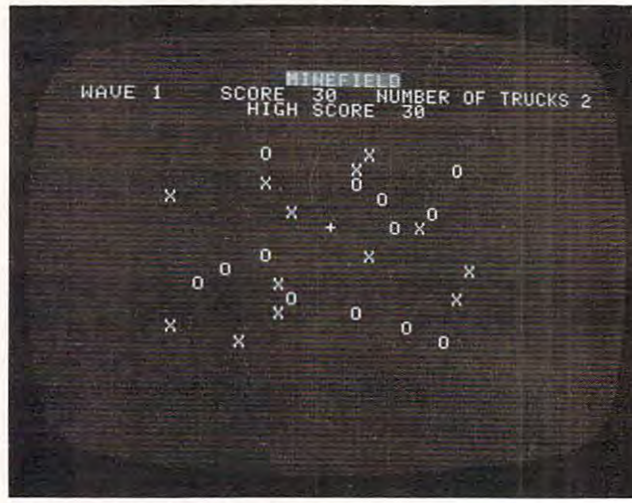
A mine explodes in the Commodore 64 version of "Minefield."

```
     THIS GAME IS TO PICK UP"
1190 PRINT"{WHT}AS MANY BOMBS AS YOU CAN BE
     FORE THEY"
1200 PRINT"{WHT}EXPLODE. TO PICK UP A BOMB,
     JUST RUN"
1210 PRINT"{WHT}OVER IT WITH YOUR TRUCK."
1220 PRINT"{WHT}BOMBS WILL EXPLODE AFTER A ~
     SHORT TIME."
1230 PRINT"{WHT}IF A BOMB TURNS REVERSE-FIE
     LD, BE CARE-"
1240 PRINT"{WHT}FUL WITH IT. IF IT STARTS T
     O BLINK, IT"
1250 PRINT"{WHT}WILL VERY SHORTLY EXPLODE-W
     ATCH OUT!!"
1260 PRINT"{WHT}BOMBS WILL CHAIN-REACT; ONE
     BOMB CAUGHT"
1270 PRINT"{WHT}IN ANOTHER'S EXPLOSION WILL
     ALSO BLOW"
1280 PRINT"{WHT}UP. IF YOU ARE CAUGHT IN A ~
     BOMB'S "
1290 PRINT"{WHT}EXPLOSION, YOU WILL BE BLOW
     N UP."
1300 PRINT"{WHT}ALSO, DO NOT RUN INTO BOMB ~
     CRATERS (*)"
1310 PRINT"{WHT}OR MINES (X) OR YOU'LL BE T
     OTALLED."
1320 PRINT"{WHT}THE CONTOLS ARE: 1 TO GO UP
1330 PRINT"{WHT}                 CTRL TO GO
     LEFT"
1340 PRINT"{WHT}                 2 TO GO RI
     GHT"
1350 PRINT"{WHT}                 _ TO GO DO
     WN"
1355 PRINT"{WHT}OR YOU CAN USE A JOYSTICK I
     N PORT 1."
1360 PRINT"{WHT}YOUR TRUCK CANNOT STOP ONCE
     YOU BEGIN"
1370 PRINT"{WHT}MOVING. IT CAN WRAP-AROUND ~
     BOTH THE"
1380 PRINT"{WHT}THE TOP AND SIDES OF THE SC
     REEN."
1390 PRINT"{DOWN}{WHT}P{WHT}RESS RETURN TO ~
     CONTINUE";
1400 GET R$:IF R$<>CHR$(13) THEN 1400
1410 PRINT"{CLEAR}S{WHT}ELECT SKILL SETTING
     (0-3)"
1420 GET R$:IF R$<"0" OR R$>"3" THEN 1420
1430 NL=4-VAL(R$):RETURN
1900 END
2000 REM SOUND OF EXPLOSION
2010 QW=54272
2020 FORS=QWTOQW+24:POKES,0:NEXT
2025 POKEQW+24,47
2030 POKEQW+5,64+7 :POKEQW+6,240
2050 POKEQW+4,GR :POKEQW+1,36:POKEQW,85
2060 FORT=1TO250:NEXT
2070 FORT=15TO0STEP-1 :POKEQW+24,INT(T):NEX
     T
2080 RETURN
```

Program 4: Minefield – Apple Version

```
80  REM  MINEFIELD FOR APPLE
90  GOSUB 2000
100  REM  SETUP VARIABLES
105 NT = 3
110  DIM BO(23),XL%(23),MO(23)
125 H = 20:U = 12
130  HOME : GOSUB 600: REM  HEADINGS
139  REM  SET UP ARRAYS
140  FOR X = 6 TO 19:BP =  INT ( RND (1
    ) * 25 + 5):BO(X) = BP: NEXT
150  FOR X = 6 TO 19
160 MP =  INT ( RND (1) * 30 + 5):MO(X)
     = MP: IF MO(X) = BO(X) THEN 160
165  NEXT
170  REM  PRINT SCREEN
180  FOR X = 6 TO 19: VTAB (X): HTAB (B
    O(X)): PRINT "O";: VTAB (X): HTAB
    (MO(X)): PRINT "X";: NEXT
190  REM  SET UP ARRAY FOR POSITION
200  FOR I = 0 TO 7:XL%(I) = 1024 + 128
     * I:XL%(I + 8) = 1064 + 128 * I:X
    L%(I + 16) = 1104 + 128 * I: NEXT
210 U = 12:H = 20:NB = 10
215  HTAB (H): VTAB (U): PRINT "+";
220  REM  GET COMMANDS
225 TI = TI + 1: GOSUB 800
230 P =  PEEK ( - 16384) - 128: IF P <
    73 OR P > 77 THEN 230
235 B$ =  CHR$ (P)
245 ODDU = U:OLDH = H
250  IF B$ = "I" THEN U = U - 1
260  IF B$ =  CHR$ (3) THEN  END
270  IF B$ = "M" THEN U = U + 1
280  IF B$ = "K" THEN H = H + 1
290  IF B$ = "J" THEN H = H - 1
300  IF H < 1 THEN H = 39
310  IF H > 39 THEN H = 1
320  IF U < 6 THEN U = 22
330  IF U > 22 THEN U = 6
333  REM   PICK UP BOMBS
335  IF  PEEK (XL%(U - 1) + H - 1) = 20
    7 THEN  POKE XL%(U - 1) + H - 1,42
    :NB = NB - 1:SC = SC + 10:BO(U-1)=0
336  IF  PEEK (XL%(U - 1) + H - 1) = 79
     THEN  POKE XL%(U - 1) + H - 1,42:
    NB = NB - 1:SC = SC + 20:BO(U - 1)= 0
340  REM CHECK FOR COLLISION
345  IF NB < 0 THEN W = W + 1:TI = TI -
    25: GOTO 130
350 K =  PEEK (XL%(U - 1) + H - 1): IF
    K = 207 OR K = 224 OR K = 160 OR K
     = 32 OR K = 176 OR K = 42 OR K =
    79 OR K = 238 THEN 360
355  VTAB (ODDU): HTAB (OLDH): PRINT "
    ";:NT = NT - 1: GOSUB 1000:W = W +
    1: GOTO 130
360  REM  UPDATE BOMBS
380  REM  SCORE
390  VTAB (2): HTAB (6): PRINT W;: HTAB
    (18): PRINT SC;:IF HS< SC THEN HS=SC
400  VTAB (3): HTAB (25): PRINT HS;: VTAB
    (2): HTAB (40): PRINT NT;
410  IF NT < 1 THEN W = W + 1: GOTO 700
580  REM  MOVE TRUCK
585  VTAB (ODDU): HTAB (OLDH): PRINT "
    ";: VTAB (U): HTAB (H): PRINT "+";
590  GOTO 220
600  HTAB (16): FLASH : PRINT "MINEFIEL
    D": NORMAL : VTAB (2): PRINT "WAVE
         ";"SCORE        ";"NUMBER OF
    TRUCKS                            "
    : VTAB (3): HTAB (13): PRINT "HIGH
     SCORE";: RETURN
700  VTAB (4): HTAB (4): FLASH : PRINT
    "PLAY AGAIN?";: NORMAL
710  VTAB (4): HTAB (17): PRINT "Y OR N";
730  GET A$: IF A$ = "Y" THEN SC = 0:NT
     = 3:TI = 0:W = 0: GOTO 130
740  IF A$ = "N" THEN  HOME : INVERSE :
     PRINT "LATER ON": NORMAL : END
760  GOTO 700
800  IF TI > 100 THEN X =  INT ( RND (1
    ) * 14 + 6): IF BO(X - 1) <  > 0 AND
    BO(X) <  > 0 AND XL%(X - 1) > 6 THEN
```

```
     POKE XL%(X - 1) + BO(X) - 1,79
810  IF TI > 300 AND BO(X - 1) <  > 0 AND
     BO(X) <  > 0 THEN  POKE XL%(X - 1)
      + BO(X),160: POKE XL%(X - 1) + BO
     (X) - 1,188: POKE XL%(X - 1) + BO(
     X) + 1,188
820  IF TI > 300 AND BO(X - 1) > 5 < 30
      AND BO(X) <  > 0 THEN  POKE XL%(X
      - 1) + BO(X) - 1,160: POKE XL%(X -
     1) + BO(X) + 1,160: POKE XL%(X - 1
     ) + BO(X) + 2,188: POKE XL%(X - 1)
      + BO(X) - 2,188: POKE XL%(X - 1) +
     BO(X) - 2,160: POKE XL%(X - 1) + B
     O(X) + 2,160
825  IF TI > 325 THEN TI = 0: GOTO 130
830  RETURN
1000  REM EXPLOSION
1010  FOR G = 1 TO 3: POKE XL%(U - 1 -
     G) + (H - 1 - G),239: POKE XL%(U -
     1 + G) + (H - 1 - G),220: POKE XL%
     (U - 1 - G) + (H - 1 + G),220: POKE
     XL%(U - 1 + G) + (H - 1 + G),239: NEXT
1020  FOR G = 1 TO 3: POKE XL%(U - 1 -
     G) + (H - 1 - G),224: POKE XL%(U -
     1 + G) + (H - 1 - G),224: POKE XL%
     (U - 1 - G) + (H - 1 + G),224: POKE
     XL%(U - 1 + G) + (H - 1 + G),224: NEXT
1025  GOSUB 3000
1030  RETURN
2000  HOME : PRINT " THE OBJECT OF MINE
     FIELD IS TO  COLLECT AS MANY BOMBS
      AS POSSIBLE BEFORE THEY EXPLODE.I
     F THE BOMBS(O) FLASH THEN THEY WIL
     L SOON EXPLODE. YOU GET THREE TRUC
     KS(+) PER GAME.
2010  PRINT "POINTS ARE AWARDED AS FOLLOWS:
2020  PRINT "TEN POINTS FOR A NORMAL BOMB (O)"
2030  PRINT "TWENTY POINTS FOR A FLASHI
     NG BOMB ": FLASH : PRINT "O": NORMAL
2040  PRINT "POINTS ARE DEDUCTED IF YOU
      ARE EXPLODED."
2050  PRINT "USE THE I J K AND M KEYS T
     O MOVE YOUR TRUCK"
2060  FOR T = 1 TO 15000: NEXT
2070  RETURN
3000  FOR SR = 1 TO 60:XF =  PEEK ( - 1
     6336): NEXT : RETURN
```



"Minefield," Apple version.

©

apple®

Apple is a Registered Trademark of Apple Computer, Inc.

ATARI®

commodore

APPLE® COMPATIBLE COMPUTER
AN ECONOMICAL ALTERNATIVE

SYSCOM 2
$649

DISK DRIVES

MICRO-SCI
A2 (w/cont) $369
A40 (w/cont) $435
A70 (w/cont) $555

RANA
Elite I (w/cont) $399
Elite II (w/cont) $539
Elite III (w/cont) $679

FOURTH Super (w/cont) $385

APPLE SOFTWARE

ARTSCI
Magic Window $65
Magic Window II $99
Magic Words $45
AVANT GARDE
Zero Grav. Pinball $20
Hi-Res Golf $20
Hi-Res Secrets $82
BEAGLE BROS.
Dos Boss $17
Alpha Plot $28
Utility City $21
Tip Disk #1 .... $15
Apple Mechanic $21
Pronto Dos $21
Flex Text $21
Frame Up $21
Type Faces $14
BRODERBUND
Choplifter $23
Star Blazer $22
Davids Midnight $23
Apple Panic $20
Alien Rain $17
Arcade Machine $41
CONTINENTAL
Home Accountant $52
1st Class Mail $53
DATAMOST
Snack Attack $21
Swashbuckler $23
EDUWARE
Spelling Bee w/Read ... $27
Alg. 1, 2 or 3 .... $27
Rendevous $27
Prisoner II $23
HAYDEN SOFTWARE
Sargon II $25
Piewriter $95
INFOCOM
Zork I, II or III $27
Starcross $27
Deadline $34
INSOFT
Graforth II $50
Electric Duet $20
LOTUS
Exec. Brief Sys. $126
MICROSOFT
Basic Compiler $270
Multiplan $190
Olympic Decathalon ... $20
MUSE
Robot War $27
Castle Wolfenstein $20
PENGUIN
Graphic Magician $42
Comp. Graphics Sys. .. $49
Special Effects $29

ON-LINE
Screenwriter $89
Wiz & Princess $22
Mystery House $17
Time Zone $65
Maurauder $24
Frogger $23
Cannonball Blitz $23
Screenwriter Prof. $135
QUALITY
Bag of Tricks $27
Ali Baba & 40 Thieves . $22
SENSIBLE SOFTWARE
Sensible Speller $85
Super Disk Copy III $21
Multi Disk Catalog $18
SIRTECH
Wizardry $35
Knight of Dia. $24
SOFTWARE PUBLISH.
PFS: Report $65
PFS: Filing $87
PFS: Graph $87
SPINNAKER
Snooper Troops # $29
Snooper Troops #2 ..... $29
Story Machine $23
Face Maker $23
STONEWARE
D.B. Master $158
D.B. Master Util #1 ..... $70
Graphics Proc. Sys. $46
STRATEGIC SIM.
Computer Baseball $27
Cytron Masters $27
Guadalcanal $42
Galactic Gladiators $27
Battle of Shiloh $27
Tigers In Snow $27
Cosmic Balance $27
Computer Qrtrback $27
SUBLOGIC
Flight Simulator $25
Space Viking $34
Saturn Navigator $24
SYNERGISTIC
Atlantis $27
Global Prog. Line Ed. .. $44
VISICORP
Visiterm $72
Visicalc 3.3 .... $175
MISCELLANEOUS
Bandits $23
Human Fly $21
Master Type $27
New Step by Step $57
Sam (w/DAC) $83
Ultima $27
Zoom Graphics $27

PRINTERS

CITOH
Prowriter $379
Prowriter II $649
Starwriter $1379
Printmaster $1599

GEMINI 10 .. $319
GEMINI 15 .. $485

NEC
8023 A-C $465
3510 .... $1375
3530 .... $1595
3550 (IBM) $1829
7710/7730 .... $2319

SMITH CORONA ... $589
AXIOM GP-100 ..... $269

MONITORS

NEC
12" GRN (JB1260) ... $115
12" GRN (JB1201M) . $155
12" Color Composite $329
12" Color RGB $689

USI (Amber) $169

AMDEK
V300 .... $139
V310 (GRN-IBM) $169
V310-A (Amber-IBM) . $169
COLOR I $298
COLOR II $650

MODEMS

HAYES
Micromodem II $269
Stack Smartmodem . $215
Smartmodem 1200 .. $499

NOVATION
Apple-Cat II $299
212 Apple-Cat $589
D-Cat $155

ANCHOR AUTOMATION Signalman I or II $79

ACCESSORIES

Wico (Joy) $23
Wico (T-Ball) $49

800 (48K) $479.95
NEW 64K ATARI 1200 . CALL

RAMDISK (128K) $378
BIT 3 - 80 COL. BOARD $279
BOOKKEEPER KIT $169

810 DISK DRIVE $419
410 RECORDER $74
850 INTERFACE $164
400 COMPUTER $219
Entertainer $66
Communicator $298

32K RAM (Mosaic) ... $99
32K RAM (Intec) $69
64K (Intec/400) $129
48K (Intec/400) $95
Educator $112
Programmer $48

PERCOM

Single Density Master $395
Single Density Dual $639
Double Density Master $515
Double Density Dual $859
Dbl Sided Dbl Density Master $639
Dbl Sided Dbl Density Dual $949

ADVENTURE INT'L
Rear Guard (D) $17
Adv. 1-12 each (C) $18
Preppie (C/D) $20
Diskey (D) $33
APX
Eastern Front (C/D) $23
Fam. Cash Flow (D) $17
747 Land Sim. (C/D) $17
ATARI INC.
Fig-Forth (C) $30
Galaxian $32
Defender $32
ET $36
Microsoft Basic (D) $65
Macro Ass. & Edit. (D) . $65
Assembler Editor (R) ... $45
Basic Cartridge (R) $45
Pac Man (R) $32
Centipede (R) $32
Caverns of Mars (D) $28
Missile Command (R) .. $27
Star Raiders (R) $32
Conv. Lang. Ea. (C) $44
Music Composer (R) $31
Super Breakout (R) $27
My First Alphabet (D) .. $26
Prog. 2 & 3 (ea.)(C) $21
Word Processor (D) ... $107
Pilot (Educ) $98
Touch Typing (C) $19
Home File Mngr (D) $37
AUTOMATED SIMUL.
Monster Maze (R) $27
Invasion Orion (C/D) ... $18
Temple of Aps.(C/D) $28
Star Warrior (C/D) $28
Dragon's Eye (D) $20
Crush Crumble (C/D) ... $20
AVALON HILL
Empire of Over (D) $23
B-1 Nuc. Bomber (C) ... $12
BRODERBUND
Apple Panic (C/D) $20
Star Blazer $22
Choplifter ... (D) $23 (R) $29
DATA SOFT
Text Wizard (D) $65
Graphics Gen. (D) $17
Basic Compiler (D) 165
Zaxxon (C/D) $27
EDU-WARE
Compu-Read (D) $21
Compu-Math Fr. (D) $28
Compu-Math Dec. (D) .. $28
INFOCOM
Zork I, II or III (D) $27
Starcross $27
Deadline (D) $34
JV SOFTWARE
Action Quest (C/D) $20
Ghost Encount. (C/D) .. $20

K-BYTE
Krazy (each) $34
LJK
Letter Perfect (D) $104
Data Perfect (D) $74
ON-LINE
Wiz & Princess (D) $22
Crossfire (C/D) $20
Frogger (C/D) $23
Jawbreaker (C/D) $21
Crossfire (R) $28
OPTIMIZED SYSTEMS
Max-65 (D) $58
Basic A + (D) $58
ROKLAN
Gorf (D) $27
Gorf (R) $30
Wizard of Wor (D) $27
Wizard of Wor (R) $30
SIRIUS
Space Eggs (D) $20
Sneakers (D) $20
Way Out (D) $27
Bandits (D) $23
Fast Eddy (R) $27
SPINNAKER
Snooper Troops #1 (D) . $30
Snooper Troops #2 (D) . $30
Storm Machine (D) $23
Face Maker (D) $23
STRATEGIC SIM.
Shattered Alliance (D) . $28
Tigers In Snow (C/D) ... $28
Battle of Shiloh (C/D) .. $28
Battle of Norm. (C/D) .. $27
Galactic Gladiator (D) . $27
Cytron Masters (D) $27
SYNAPSE SOFTWARE
File Mngr 800 + ... $65
Protector II .. (D) $23 (R) $29
Shamus ...... (D) $23 (R) $29
Nautilus ..... (D) $23 (R) $29
Claim Jump (C/D) $23
THORN EMI
Jumbo Jet (R) $34
Submarine Comm. (R) . $34
MISCELLANEOUS
Ali Baba (D) $22
Miner 2049er (R) $35
Kid Grid (C/D) $20
Pool 1.5 (D) $23
Raster Blaster (D) $20
Sam (D) $41
Galactic Chase (C) $17
Warlocks Revenge (D) . $24
Visicalc (D) $169
3-D Supergraph (C/D) .. $29
Starbase Hyp (D) $17
Pogo Man (C/D) $27
Airstrike (C/D) $28
Sammy Sea Serp. (C) ... $13
Pinball (D) $20

EAGLE

64K RAM
780 KB Disk Storage
Word Processor
Ultracalc CP/M
C-Basic Software

EAGLE II $2329
EAGLE IV (10M Byte Hard Disk) $4299

VIC 64 .... CALL

VIC 20 .... $149
1541 DISK DRIVE ... CALL
1525 PRINTER ..... CALL

1530 RECORDER $64
1600 MODEM $92
16K RAM $88

VIC SOFTWARE

Avenger $23
Superslot $23
Super Alien $23
Jupiter Lander $23
Draw Poker $23
Midnight Drive $23
Radar Rat Race $23
Raid on Fort Knox $23
Sargon II Chess $29
Super Smash $23
Cosmic Cruncher $23
Gorf $29
Omega Race $29
Money Wars $23
Menagerie $23

Cosmic Jailbreak $23
Clowns $23
Garden Wars $23
Sea Wolf $23
Adventureland $29
Pirate Cove $29
Mission Impossible $29
The Count $29
Voodoo Castle $29
The Sky Is Falling $23
Mole Attack $23
Bingo Speed Math $23
Home Babysitter $23
Visible Solar System ... $23
Personal Finance $29

HES Software
VIC Forth (R) $45
HES Mon (R) $29
Turtle Graphics (R) $29
HES Writer (R) $29
Aggressor (R) $29
Shamus (R) $29
Protector (R) $33
Synthesound (R) $49
Skier (C) $15
Maze of Mikor (C) $15
Tank Wars (C) $15
Victrek (C) $15
Pinball (C) $13
Simon (C) $13
Fuel Pirates (C) $13
Pak Bomber (C) $13
Laser Blitz (C) $15
Tank Trap (C) $15
Concentration (C) $13
Dam Bomber (C) $13

UNITED MICROWARE
Spiders of Mars (R) $34
Meteor Run (R) $34
Amok (C) $17
Alien Blitz (C) $17
Skymath (C) $12
Space Division (C) $12
Super Hangman (C) $14
The Alien (C) $17
3D Maze (C) $12
Kosmic Kamikaze (C) .. $17
Sub Chase (C) $17
Amok (R) $27
Renaissance (R) $34
Alien Blitz (R) $27
Cloud Burst (R) $27
Satellites (R) $34
Outworld (R) $34

COMPUTERS

TELEVIDEO
802 .... $2595   802H $4445

NEC
PC-8001 Computer $709
PC-8012 I/O Unit $469
PC-8031 (Dual Drive) $709

NORTH STAR
Advantage $2595
Advantage (5M Byte H.D.) $3799

TRS-80
MOD III (2 Drvs, 48K) $1649

COSMIC COMPUTERS
UNLIMITED

THE ABOVE PRICES ARE FOR PREPAID ORDERS

ORDER LINES OPEN MON-SAT 8 am - 8 pm

(714) 861-1265

228 N. PROSPECTORS RD.
DIAMOND BAR, CA 91765

Add $2.00 Shipping per software order anywhere in U.S. Add $5.00 Shipping per software for non-U.S. orders, P.O. Box or FPO-APO. Call for cost of Hardware shipping. Calif. residents add 6½% sales tax. Cashiers Checks or Money Orders filled same day. Personal checks require 4 weeks to clear. **Master Card and Visa OK for software only, add 3% surcharge.** Include card no., expiration date and signature. Prices subject to change.

Relocation Calculator

Linton S. Chastain

This automatic analysis of the effects of switching jobs might reveal some hidden economic factors in making career choices. The program will work with either standard or Extended BASIC and either 16 or 32K.

Those of you who have been asked to relocate or may be considering relocating for your own reasons, might be interested in this program called "Salary Comparison." The program, which requires 5055 bytes to run, compares your present salary and cost with the new salary and cost. It gives you the bottom line of either a profit or loss based on the input of the old and new salaries and costs. You have a choice of outputting the old and new salaries and costs with results to the screen or printer.

The input for your old salary and cost should be readily available from your paycheck stubs and household budget. The input for the new salary can be obtained through calculations based on payroll formulas or from interpolations based on percentages of deductions of your old salary, or you may be able to obtain the input needed from the payroll department. The input for the new OUTGO can be obtained from national tax accountants, realtors, and the chamber of commerce in the new location. You can also use an almanac and indexing of the old to the new cost.

The more accurate the inputs, of course, the more accurate the results. (Recall the old computer saw: "Garbage in, garbage out.") Regardless of whether or not you decide to take the new position, at least you have an idea of what it will cost. Your decision may well be a more objective one and, in the long run, a more satisfying one.

```
40 A=0:B=0:C=0:D=0:E=0:F=0:G=0:H=0:I=
   0:J=0:K=0:L=0:M=0:N=0:O=0:P=0:Q=0:
   R=0:AA=0:AB=0:U=0:V=0:Y=0:Z=0:AC=0
   :AD=0:AE=0:AF=0:AJ=0:AK=0:AL=0:AM=
   0:AN=0:AO=0:AP=0:AQ=0:AR=0:AZ=0:AT
   =0:AU=0:AV=0:AW=0:AY=0:W=0:X=0
70 CLS:PRINT"INCOME  COMPARISON":PRIN
   T:PRINT"COMMAND LIST # 1"
80 PRINT" 1-INPUT SALARY"
90 PRINT" 2-INPUT OUTGO"
100 PRINT" 3-DISPLAY SALARIES"
110 PRINT" 4-DISPLAY OUTGOES"
120 PRINT" 5-DISPLAY DIFFERENCE"
130 INPUT"ENTER COMMAND BY NUMBER";S:
    IFS<1 OR S>5 THEN70
140 ON S GOTO150,505,370,760,910
150 PRINT
160 PRINT:PRINT"ENTER THE FOLLOWING D
    ATA AS REQUESTED"
180 PRINT"-SALARY A   ;B"
190 PRINT"-FED TAX C   ;D"
200 PRINT"-FICA E   ;F"
210 PRINT"-STATE TAX G   ;H"
220 PRINT"-SAVING BOND I   ;J"
230 PRINT"-LIFE INS. K   ;L"
240 PRINT"-GP. INS. M   ;N"
250 PRINT"-SAVING O   ;P"
260 PRINT"-LTD Q   ;R"
280 INPUT"SALARY A";A:INPUT"SALARY B"
    ;B
290 INPUT"FED TAX C";C:INPUT"FED TAX
    D";D
300 INPUT"FICA E";E:INPUT"FICA F";F
310 INPUT"STATE TAX G";G:INPUT"STATE
    TAX H";H
320 INPUT"SAVING BOND I";I:INPUT"SAVI
    NG BOND J";J
330 INPUT"LIFE INS. K";K:INPUT"LIFE I
    NS. L";L
340 INPUT"GP. INS. M";M:INPUT"GP. INS
    . N";N
350 INPUT"SAVING O";O:INPUT"SAVING P"
    ;P
360 INPUT"LTD Q";Q:INPUT"LTD R";R
365 GOTO70
370 AA=A-(C+E+G+I+K+M+O+Q):AB=B-(D+F+
    H+J+L+N+P+R)
375 INPUT"INPUT PRINT TO CRT(1) OR PR
    INT TO PRINTER(2)";S:PX=S
376 IFS<1 OR S>2 GOTO375
377 IFPX=2 THENPP=-2 ELSE PP=0
380 CLS:PRINT#PP,"SALARY COMP. INCOME
     1  INCOME 2"
385 PRINT#PP,CHR$(10)
390 PRINT#PP,"SALARY";TAB(13);A;TAB(2
    3);B
400 PRINT#PP,"FED TAX";TAB(13);C;TAB(
    23);D
410 PRINT#PP,"FICA";TAB(13);E;TAB(23)
    ;F
420 PRINT#PP,"STATE TAX";TAB(13);G;TA
    B(23);H
430 PRINT#PP,"SAV. BOND";TAB(13);I;TA
    B(23);J
440 PRINT#PP,"LIFE INS.";TAB(13);K;TA
    B(23);L
450 PRINT#PP,"GP. INS.";TAB(13);M;TAB
    (23);N
460 PRINT #PP,"SAVING";TAB(13);O;TAB(
    23);P
470 PRINT#PP,"LTD";TAB(13);Q;TAB(23);
    R
480 PRINT#PP,"----------------------
    --------"
490 PRINT#PP,"DIF.";TAB(13);AA;TAB(23
    );AB
495 PRINT#PP,CHR$(10)
500 INPUT"HIT ENTER TO CONTINUE";R$:I
```

```
     FR$=INKEY$ THEN 70
505  PRINT
510  PRINT:PRINT"ENTER THE FOLLOWING D
     ATA AS REQUESTED"
530  PRINT"-PROP. TAXES U  ;V"
540  PRINT"-CAR & HOME INS. W  ;X"
550  PRINT"-FOOD Y  ;Z"
560  PRINT"-CLOTHING AC  ;AD"
570  PRINT"-MORTGAGE AE  ;AF"
580  PRINT"-WATER AG  ;AI"
590  PRINT"-N. GAS AJ  ;AK"
600  PRINT"-ELECTRICITY AL  ;AM"
610  PRINT"-MED. & DENT. AN  ;AO"
620  PRINT"-GAS VEHICLES AP  ;AQ"
630  PRINT"-EDUCATION AR  ;AZ"
650  INPUT"PROP. TAXES U";U:INPUT"PROP
     . TAXES V";V
660  INPUT"CAR & HOME INS. W";W:INPUT"
     CAR & HOME INS. X";X
670  INPUT"FOOD Y";Y:INPUT"FOOD Z";Z
680  INPUT"CLOTHING AC";AC:INPUT"CLOTH
     ING AD";AD
690  INPUT"MORTGAGE AE";AE:INPUT"MORTG
     AGE AF";AF
700  INPUT"WATER AG";AG:INPUT"WATER AI
     ";AI
710  INPUT"N. GAS AJ";AJ:INPUT"N. GAS
     AK";AK
720  INPUT"ELECTRICITY AL";AL:INPUT"EL
     ECTRICITY AM";AM
730  INPUT"MED. & DENT. AN";AN:INPUT"M
     ED. & DENT. AO";AO
740  INPUT"GAS VEHICLES AP";AP:INPUT"G
     AS VEHICLES AQ";AQ
750  INPUT"EDUCATION AR";AR:INPUT"EDUC
     ATION AZ";AZ
755  GOTO70
760  AT=U+W+Y+AC+AE+AG+AJ+AL+AN+AP+AR:
     AU=V+X+Z+AD+AF+AI+AK+AM+AO+AQ+AZ
770  CLS:PRINT#PP,"SALARY COMP.  OUTGO
      1{3 SPACES}OUTGO 2"
780  PRINT#PP,CHR$(10):PRINT#PP,"PROP.
      TAXES";TAB(13);U;TAB(23);V
790  PRINT#PP,"C & H INS.";TAB(13);W;T
     AB(23);X
800  PRINT#PP,"FOOD";TAB(13);Y;TAB(23);Z
810  PRINT#PP,"CLOTHING";TAB(13);AC;TA
     B(23);AD
820  PRINT#PP,"MORTGAGE";TAB(13);AE;TA
     B(23);AF
830  PRINT#PP,"WATER";TAB(13);AG;TAB(2
     3);AI
840  PRINT#PP,"N. GAS";TAB(13);AJ;TAB(
     23);AK
850  PRINT#PP,"ELECT.";TAB(13);AL;TAB(
     23);AM
860  PRINT#PP,"MED.&DENT.";TAB(13);AN;
     TAB(23);AO
870  PRINT#PP,"GAS VEH.";TAB(13);AP;TA
     B(23);AQ
880  PRINT#PP,"EDUC.";TAB(13);AR;TAB(2
     3);AZ
885  PRINT#PP,"----------------------
     --------"
890  PRINT#PP,"TOTALS";TAB(13);AT;TAB(
     23);AU
900  INPUT"HIT ENTER TO CONTINUE";R$:I
     FR$=INKEY$ THEN 70
910  CLS:PRINT#PP,CHR$(10):PRINT#PP,"S
     ALARY COMP. INCOME 1  INCOME 2"
920  PRINT#PP,CHR$(10):PRINT#PP,"NET I
     NCOME";TAB(13);AA;TAB(23);AB
930  PRINT#PP,"OUTGO";TAB(13);AT;TAB(2
     3);AU
940  PRINT#PP,"----------------------
     --------"
950  AV=AA-AT:AW=AB-AU
960  PRINT#PP,"NET MONEY";TAB(13);AV;T
     AB(23);AW
970  AY=AW-AV
980  IF AW=>AV THEN PRINT#PP,CHR$(10):
     PRINT#PP,"A PROFIT OF ";AY;"DOLLARS"
990  IF AW<AV THEN PRINT#PP,CHR$(10):P
     RINT#PP,"A LOSS OF ";AY;"DOLLARS"
1010 INPUT"HIT ENTER TO CONTINUE";R$
1020 IFR$=INKEY$ THEN70
```
©

COMPUTE! Subscriber Services

Please help us serve you better. If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine
P.O. Box 5406
Greensboro, NC 27403

or call the Toll Free number listed below.

Change Of Address. Please allow us 6-8 weeks to effect the change; send your current mailing label along with your new address.

Renewal. Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

New Subscription. A one year (12 month) US subscription to **COMPUTE!** is $20 (2 years, $36; 3 years, $54. For subscription rates outside the US, see staff page). Send us your name and address or call the Toll Free number listed below.

Delivery Problems. If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

COMPUTE!
800-334-0868
In NC 919-275-9809

CAPUTE!

Modifications Or Corrections To Previous Articles

VIDEO 80 For Atari

Brad Brooks points out that the 80-column software from the April issue (p. 170) can be restarted after a SYSTEM RESET without having to power up again. Simply type ?USR(9013). You'll get an ERROR 9 message, but this does not affect program operation.

Match-Em

This game for the TI in the April 1983 issue (p. 123) has a minor flaw. Hitting the same key twice will register as a valid match. Our thanks to reader James Alessio, for suggesting the following fix:

```
1085 IF A2=A1 THEN 1030
```

TI Air Defense

The confusing characters at the beginning of line 1950 of the TI version of this game (page 46 of the April issue) should be replaced with the command PRINT.

Atari CRAB

This Atari BASIC cross-reference program from the April issue (p. 188) has problems when handling inverse video characters or USR codes in quotes. To prevent this, reader David Butler suggests adding the following line:

```
245 IF C=15 THEN GOSUB GC FOR J=I TO C:GE
    T #I,G:NEXT J:GOSUB GC
```

Also, some printers add a carriage return after LPRINT;. To correct this, David offers the following modifications:

```
390 OPEN #I,8,Z,"P:":PRINT #I;"XREF FOR "
    ;A$
420 D=INT(LN/H):M=LN-H*D:IF NOT M THEN PR
    INT #I:PRINT #I
430 PL=PEEK(X+T):FOR J=5 TO PL-T:PRINT #I
    ;CHR$(PEEK(X+J));:NEXT J
440 PRINT #I;" ";:IF NOT M THEN PRINT #I
460 PRINT #I:PRINT #I:PRINT #I;D-L+I;" VA
    RIABLES"
```

Apple Subroutine Capture

In the text for the EXEC file (Program 1, page 171) of this article from the March issue, be sure that the word "RUN" appears (without a line number) as the last line in the file or the program will not operate properly.

VIC Data Acquisition

In the program on page 248 of the May 1983 issue, the POKE 37166,128 in line 20 should be replaced with POKE 37166,64.

ZX-81/TS-1000 Data Management

Line 2065 of this program from the March 1983 issue (p. 230) should read:

```
2065 IF N$(S,1 TO 30)=S$(1,1 TO 30) THEN
     GOTO 2140
```

Color Computer Version Of Vehicle Cost Performance

In line 770 (February 1983, p. 164), the WRITE#1 should be PRINT#1. In line 1160 the CLD should be CLS. In line 1250, the "GALLONS";Y should be "GALLONS",Y.

Commodore Maze Generator Enhancement

The maze generating program by Charles Bond reprinted in the February issue (p. 106) has a shortcoming. The fixed order of the elements in the A(3) coordinate array generates mazes that almost invariably spiral counterclockwise around the screen. Neil Murray suggests stirring in a little randomness by adding the following line:

```
115 FOR I=3 TO 1 STEP -1:K=INT(RND(I)*I):
     SV=A(K):A(K)=A(I):A(I)=SV:NEXT I
```

Automatic Commodore Program Selector

The programs which accompany this article from the March 1983 issue (p. 156) require modification to work with the VIC-20 and 1540/1541 disk drive. All output to the screen should be adjusted for the 22-column display. Line 290 of Program 1 and line 470 of Program 2 should be changed to:

```
PRINT"{4 DOWN}RUN":PRINT"RUN":PRINT"
{HOME}{7 DOWN}";
```

In Program 3, delete line 420 and make the following changes:

```
210 DIM AE$(150)
240 PRINT:PRINT:PRINT"READING{2 SPACES}D
    RIVE 0"
250 PRINT"{DOWN}PROGRAMS FOUND: 0"
290 IF C$=CHR$(220) THEN 410
350 PRINT"{UP}"TAB(17)AN-A0
440 MM=9:PRINT"{CLR}{RVS}PROGRAM MENU #"
    STR$(MN+1)"{OFF}{DOWN}"
460 PRINT TAB(3)"{RVS}"RIGHT$(STR$(I),1)
    "{OFF} "MID$(AE$(MN*9+I),3,16)"
    {DOWN}"
470 NEXT:PRINT"{RVS}MAKE A CHOICE OR
    {OFF}"
480 PRINT"{RVS} PRESS 'RETURN' {OFF}"
590 PRINT"RUN":PRINT"RUN":PRINT"{HOME}
    {7 DOWN}";
```

For the Commodore 64 with 1541 drive, delete line 420 and make the following changes to Program 3:

```
210 DIM AE$(150)
290 IF C$=CHR$(199)THEN 410
```

©

ATTENTION
VIC 20/TI-994/A
USERS

NEED A PROGRAM?
HAVING TROUBLE WITH ONE?
DON'T KNOW HOW TO
TRANSLATE YOUR IDEA?

WE CAN HELP

Send us a description of the program you want. We will send you more information and the cost for us to create your program.

Starting at $24.95

DYTEK
P. O. Box 241
Pinellas Park, FL 33565
PH: 393-3597

We also buy programs.
SEND NAME AND PHONE NO.

CASSETTES ! ! !

FOR YOUR COMPUTER

- Computer Grade • Wide Dynamic Range
- 100% Error Free • 5 Screw Housing
- Fully Guaranteed • Carefully Packed

All Prices Include Shipping
** Phone Orders Add $1.50 C.O.D. Fee **

COMPUTER TAPE PRICES

Length	12 LOT	24 LOT	100 LOT
C-5	.52/6.24	.38/9.12	.35/35.00
C-10	.55/6.60	.40/9.60	.35/35.00
C-20	.60/7.20	.45/10.80	.40/40.00

BASF DPS Tapes Add .05 Cents Per Tape
— Custom Lengths Available —
... Write For Volume Prices...

— Norelco Cassette Cases and Labels —
(with Cassette Orders Only)

12-249 Cases/ .20 Ea.	250- .13 Ea.
12 Labels for .20	120 for 1.70
1000 Pinfeed Labels	14.50

SEND MONEY ORDERS OR CHECKS TO:

CASS-A-TAPES
Box 8123-C
Kansas City, Mo. 64112
816-444-4651

Eric Martin's
Where prices are born, not raised!

ATARI

Atari 1200XL 64K CALL

810 Disc Drive	CALL
1010 Program Recorder (NEW)	$79.00
1025 Printer (NEW)	$410.00
CX85 Numeric Keyboard	$109.00
CX419 Bookkeeper Kit	CALL
CX488 Communicator Kit	CALL
ATARI Software	
CXL4025 Defender	$39.00
RX8026 Dig Dug	$39.00
RX8039 Eastern Front (ROM)	$44.00
CLX 4027 Qix	$39.00
RX8037 Star Trux	$39.00
DX5049 Visicalc	$159.00
CA01655 Technical Reference Notes	$29.95
Third Party Software	
Miner 2049	CALL
Zaxxon	$27.95
Eastern Front 1941 (cassette)	$29.95
Wizard of War	$30.00
Gorf	$30.00
Frogger	$30.00
Chop Lifter	$29.00
Apple Panic	$23.75
Baha Buggies	$23.75
Submarine Commander (ROM)	$35.95
Jumbo Jet Pilot (ROM)	$35.95
Soccer (ROM)	$36.95
Gamestar Football	CALL

SCOTCH maxell

Diskettes & Tape	
SCOTCH 5¼" Single sided	$26.95
SCOTCH 5¼" Double sided	$29.95
SCOTCH Tape C-10 (lot of 10)	$24.90
SCOTCH Tape C-30 (lot of 10)	$26.90
SCOTCH T-120 VHS	$9.95
MAXELL 5¼" Single sided	$31.00
MAXELL 5¼" Double sided	$39.00
MAXELL VHS Tape (SUPER PRICE)	CALL

We take TRADE-INS.
Call for your price.

ATARI SERVICE
FACTORY AUTHORIZED NETWORK

Orders shipped in 24 hours

For fast delivery, send certified or cashier checks, money orders, or direct bank wire transfers. Personal checks allow 2 to 3 weeks to clear. Prices reflect cash discount and are subject to change. Add 2% for credit card purchases. Shipping—Software $2 Minimum. Hardware—call. Foreign inquiries invited —add 15% for shipping. Ohio residents add 6.5% sales tax.

Eric Martin's

5485 Warrensville Center Road
Maple Heights, Ohio 44137
Call Toll Free
1-800-482-7254
In Ohio 216/663-2032
Mon.-Sat. 10-6 EST

TRAIN AT HOME IN SPARE TIME!
NO PREVIOUS EXPERIENCE NEEDED!

FOR HOME OR BUSINESS!
WHAT...WHEN...HOW

LEARN HOW TO USE A COMPUTER

Learn how and why a computer can help you. Learn to write your own computer programs. See how easy it is to use different programs already available. See how they fit into your home or business operations...budgeting, real estate, bookkeeping, inventories, expenses, pricing, profit margins, investments, interest, taxes, shopping lists, vacation planning, addresses, phone numbers, routing...hundreds more including foreign languages, computer games and graphics. Never again be at the mercy of a so-called "computer expert." Know what really happens when you get a computer problem from a bank, store, loan company, oil company, utility or anyone else. You'll be able to talk their language...understand why and how things happen...be able to take the offensive when you're the victim of a computer error.

EXPERTS SHOW YOU WHAT TO DO, HOW TO DO IT...TO MAKE YOUR LIFE EASIER

Everything is explained in easy-to-understand language with plenty of examples. Step-by-step directions take you through basic computer programming. You learn everything you need to know to use the computer. You'll be able to understand computer experts and talk their language...storage, systems, terminals...you'll learn it all and much, much more.

NOW...ALL THIS IN ONE COURSE!

- Computer Training • Computer Programming
- Computer Applications • Computer Games

Find out what you need to know about computer operations

TIMEX COMPUTER INCLUDED WITH YOUR TRAINING
Plugs into any TV!

SEND FOR FREE FACTS!

ICS SINCE 1891 COMPUTER TRAINING, Dept. DZ063
Scranton, Pennsylvania 18515

Please rush me free facts and color brochure that tells how I can learn computer applications, programming and operation at home in spare time. NO SALESMAN WILL CALL.

Name ______________ Age ____
Address ______________
City/State/Zip ______________

COMPUTE! Back Issues

Here are some of the applications, tutorials, and games from available back issues of **COMPUTE!**. Each issue contains much, much more than there's space here to list, but here are some highlights:

February 1981: Simulating PRINT USING, Using the Atari as a Terminal for Telecommunications, Attach a Printer to the Atari, Double Density Graphing on C1P, Commodore Disk Systems, PET Crash Prevention, A 25¢ Apple II Clock.

May 1981: Named GOSUB/GOTO in Applesoft, Generating Lower Case Text on Apple II, Copy Atari Screens to the Printer, Disk Directory Printer for Atari, Realtime Clock on Atari, PET BASIC Delete Utility, PET Calculated Bar Graphs, Running 40 Column Programs on a CBM 8032.

June 1981: Computer Using Educators (CUE) on Software Pricing, Apple II Hires Character Generator, Ever- expanding Apple Power, Color Burst for Atari, Mixing Atari Graphics Modes 0 and 8, Relocating PET BASIC Programs, An Assembler In BASIC for PET, QuadraPET: Multitasking?

July 1981: Home Heating and Cooling, Animating Integer BASIC Lores Graphics, The Apple Hires Shape Writer, Adding a Voice Track to Atari Programs, Machine Language Atari Joystick Driver, Four Screen Utilities for the PET, Saving Machine Language Programs on PET Tape Headers, Commodore ROM Systems, The Voracious Butterfly on OSI.

August 1981: Minimize Code and Maximize Speed, Apple Disk Motor Control, A Cassette Tape Monitor for the Apple, Easy Reading of the Atari Joystick, Blockade Game for the Atari, Atari Sound Utility, The CBM "Fat 40," Keyword for PET, CBM/PET Loading, Chaining, and Overlaying.

October 1981: Automatic DATA Statements for CBM and Atari, VIC News, Undeletable Lines on Apple, PET, VIC, Budgeting on the Apple, Switching Cleanly from Text to Graphics on Apple, Atari Cassette Boot-tapes, Atari Variable Name Utility, Atari Program Library, Train your PET to Run VIC Programs, Interface a BSR Remote Control System to PET, A General Purpose BCD to Binary Routine, Converting to Fat-40 PET.

December 1981: Saving Fuel $$ (Multiple Computers: versions for Apple, PET, and Atari), Unscramble Game (multiple computers), Maze Generator (multiple computers), Animating Applesoft Graphics, A Simple Printer Interface for the Apple II, A Simple Atari Wordprocessor, Adding High Speed Vertical Positioning to Atari P/M Graphics, OSI Supercursor, A Look At SuperPET, Supermon for PET/CBM, PET Mine Maze Game.

January 1982: Invest (multiple computers), Developing a Business Algorithm (multiple computers), Apple Addresses, Lowercase with Unmodified Apple, Cryptogram Game for Atari, Superfont: Design Special Character Sets on Atari, PET Repairs for the Amateur, Micromon for PET, Self-modifying Programs in PET BASIC, Tinymon: a VIC Monitor, Vic Color Tips, VIC Memory Map, ZAP: A VIC Game.

February 1982: Insurance Inventory (multiple computers), Musical Transposition (multiple computers), Multitasking Emulator (multiple computers), Disassemble Apple Programs from BASIC, Plotting Polar Graphs on Apple, Atari P/M Graphics Made Easy, Atari PILOT, Put A Rainbow in your Atari, Marquee for PET, PET Disk Disassembler, VIC Paddles and Keyboard, VIC Timekeeping.

May 1982: VIC Meteor Maze Game, Atari Disk Drive Speed Check, Modifying Apple's Floating Point BASIC, Fast Sort For PET/CBM, Extra Atari Colors Through Artifacting, Life Insurance Estimator (multiple computers), PET Screen Input, Getting The Most Out Of VIC's 5000 Bytes.

July 1982: Gold Miner Game (Atari and VIC), IRA Planner (multiple computers), Atari Video Graphics, Apple DOS Changer, Super QuadraPET, VIC Overview, Maze Race (multiple computers), Direct Access File Editor (PET and Atari), VIC Super Expander Memory Map, Using The 6560 Video Interface Chip, PET Compactor, Headless FORTH Metacompilation, Test RAM Nondestructively (multiple computers).

August 1982: The New Wave Of Personal Computers, Household Budget Manager (multiple computers), Word Games (multiple computers), Color Computer Home Energy Monitor, Intelligent Apple Filing Cabinet, Guess That Animal (multiple computers), PET/CBM Inner BASIC, VIC Communications, Keyprint Compendium, Animation With Atari, VIC Curiosities, Atari Substring Search, PET and VIC Electric Eraser.

September 1982: Apple and Atari and the Sounds of *TRON*, Commodore Automatic Disk Boot, VIC Joysticks, Three Atari GTIA Articles, Color Computer Graphics, The Apple Pilot Language, Sprites and Sound on the Commodore 64, Peripheral Vision Exerciser (multiple computers), Banish INPUT Statements (multiple computers), Charades (multiple computers), PET Pointer Sort, VIC Pause, Mapping Machine Language, Editing Atari BASIC With the Assembler Cartridge, Process Any Apple Disk File.

January 1983: Sound Synthesis And The Personal Computer, Juggler And Thunderbird Games (multiple computers), Music And Sound Programs (multiple computers), Writing Transportable BASIC, Home Energy Calculator (multiple computers), All About Commodore WAIT, Supermon64, Perfect Commodore INPUTs, Atari Autonumber, Copy VIC Disk Files, Commodore 64 Architecture.

February 1983: How The Pros Write Computer Games, 12 Joysticks Compared, Slalom (a game in 3-D for multiple computers), Super Shell Sort For PET, Atari SuperFont Plus, Creating Graphics On The VIC, Joysticks And Sprites On The 64, Bi-Directional VIC Scrolling, Commodore 64 Video: A Guided Tour, The Atari Cruncher, Easy Apple Editing, VIC Custom Characters For Games.

Home and Educational COMPUTING! (Fall 1981 and Summer 1981 – count as one back issue): Exploring The Rainbow Machine, VIC As Super Calculator, Custom Characters, Alternate Screens, Automatic Line Numbers, Using The Joystick (Spacewar Game), Fast Tape Locater, Window, VIC Memory Map.

Back issues are $3 each or six for $15. Price includes freight in the US. Outside the US add $1 per magazine ordered for surface postage. $4 per magazine for air mail postage. All back issues subject to availability.

In the Continental US call
TOLL FREE 800-334-0868
(In NC Call 919-275-9809)

Or write to **COMPUTE!** Back Issues, P.O. Box 5406, Greensboro, NC 27403 USA. Prepayment required in US funds. MasterCard, Visa and American Express accepted. North Carolina Residents add 4% sales tax.

Don't Buy Anything In Computer Hardware or Software Till You've Checked Out

N.Y.C.D.

New York Computer Discounters

Orders Only Call - 1-800-424-4212
Other Info Call - 212-964-5886

HARDWARE ATARI

NEW Atari 1200XL CALL FOR PRICE

800 48K 478.99
400 16K 198.99
810 Disk Drive 418.99
410 Cassette Recorder .. 71.99
850 Expansion Interface . 158.99

TARA Expansion Boards

64K RAM (400) 158.99
48K RAM (400) 128.99
32K RAM (400/800) 83.99
16K RAM (400/800) 63.99
Keyboard for 400 .... 98.99

Software for ATARI

ADV. INTL.

Preppie (C/D) 19.99
Sea Dragon (C/D) 22.99
S.A.G.A. #1,2,3 (D) .. 27.99/ea
Stratos (C/D) 22.99
Bug Off (C/D) 22.99
Tutti Frutti (C/D) 18.99
All others at 25% off List

SYNAPSE Software

Slime (R) 28.99
Slime (C/D) 22.99
Shamus (R) 28.99
Shamus (C/D) 22.99
Chicken (R) 28.99
Chicken (C/D) 22.99
Picknick Paranoia (R) 28.99
Picknick Paranoia (C/D) . 22.99
Protector II (R) 28.99
Nautilus (C/D) 22.99
Apocalypse (C/D) 22.99
File Manager 800+ 69.99
Disk Manager (D) 21.99

VISICORP

Visicalc (D) 169.99

Don't see what you want?? CALL US, we carry almost every major brand of Computer Hardware and Software. Chances are we have what you want at an amazingly low price

LJK ENTERPRISES

Letter Perfect (D) 109.99
Letter Perfect (R) 144.99
Mail Merge/Utility (D) ... 20.99
Data Perfect (D) 74.99
Edit 6502 (R) 149.99

ATARI

Pac Man (R) 31.99
Centipede (R) 31.99
Missile Command (R) 31.99
Star Raiders (R) 31.99
Asteroids (R) 25.99
Microsoft Basic (D) 63.99
Defender (R) 29.99
Dig Dog (R) 29.99
Galaxian (R) 30.99
ET Phone Home (R) 33.99
Caverns of Mars (D) 27.99
Atari Writer (R) 56.99
All other Atari Software . CALL

APX

Eastern Front (C/D) 21.99
Family Cash Flow (D) 16.99
747 Landing Sim. (C/D) .. 16.99

DATA SOFT

Zaxxon (C/D) 29.99
Canyon Climber (C/D) ... 20.99

EPYX

Temple of Apshai (D/C) .. 26.99
Invasion Orion (D/C) 16.99
Rescue at Rigel (D/C) 19.99
Star Warrior (C/D) 27.99
Escape from Villans Isle (C/D) 20.99
Hellfire Warrior (D) 26.99
Monster Maze (R) 26.99
Platter Mania (R) 26.99
All other Epyx Software CALL

K-Byte/CBS

All K-byte 31.99/ea

ON-LINE

Frogger (C/D) 22.99
Jawbreaker (C/D) 19.99
Threshold (D) 27.99

ROKLAN

Wizard of Wor (D) 27.99
Gorf (D) 26.99
Deluxe Invaders (D) 24.99
Wizard of Wor (R) 29.99
Gorf (R) 29.99
Deluxe Invaders (R) 27.99

AVALON HILL

B-1 Nuclear Bomber (C) .. 11.99
Empire of Overmind (D) .. 22.99

ATTENTION ALL TRS-80 OWNERS

N.Y.C.D. carries a huge selection of TRS-80 software and peripherals!! Please call for further info and prices.

APPLE

Bell & Howell Computer by Apple II-e .... CALL

ADV. INTL.

Rear Guard (D) 19.99
S.A.G.A. #1-12 (D) ... 27.99/ea
Stone of Sisyphus (D) 19.99

BRODERBUND

Choplifter (D) 23.99
Apple Panic (C/D) 19.99
Sea Fox (D) 23.99
Arcade Machine (C/D) ... 43.99
David's Midnight Magic (C/D) 23.99
Star Blazer (C/D) 22.99
Back Street Warrior (C/D) 43.99

JOYSTICKS

WICO Joystick 21.99
WICO Trackball 56.99
WICO Apple Adapter 17.99
WICO Joystick Deluxe ... 26.99
T.G. Trakball 44.99
T.G. Joystick 42.99
T.G. Paddles 27.99
T.G. Select-A-Port 42.99
Pointmaster 12.99
Slik Stik 8.99
Starfighter 11.99
Le Stick 29.99

AUTOMATED SIMULATIONS (EPYX)

Temple of Apshai (D) 27.99

DATASOFT

Zaxxon (D) 26.99
Canyon Climber (D) 19.99

DATAMOST

Snack Attack (D) 20.99
Aztec (D) 27.99
Tubeway (D) ....

EDU WARE

Compu-Read (D) 20.99
Compu-Read (C) 13.99
Compu-Math/Fractions(D) . 27.99
Compu-Math/Decimals(D) . 27.99
SAT Word Attack(D) 33.99
PSAT Word Attack (D) 35.99
Empire I, II, III (D) 23.99/ea

LJK Enterprises

Letter Perfect w/Mailmerge (D) 109.99
Data Perfect (D) 79.99
Edit 6502 (D) 79.99

INFOCOM

Zork 1, 2, or 3 (D) 26.99/ea
Deadline (D) 33.99
Starcross 26.99

ON LINE

Threshold (D) 27.99
Jawbreaker (D) 20.99
Crossfire (D) 20.99
Frogger (D) 24.99
Cannonball Blitz (D) 24.99

SIRIUS SOFTWARE

FANTASTIC VOYAGE

Wavy Navy (D) 24.99
Pascal Graphics Editor (D) 74.99
E.Z. Draw 3.3 (D) 36.99
Joyport (C/D) 34.99
Fast Eddy (R) 25.99
Deadly Duck (R) 25.99
Beany Bopper (R) 25.99
Worm War I (R) 25.99
All other Sirius Software CALL

MICROPRO

Wordstar (C/D) 198.99
Mailmerge (C/D) 74.99
Spell Star (C/D) 119.99

SOFTWARE PUBLISHING

PFS: File (C/D) 87.99
PFS: Graph (C/D) 87.99
PFS: Report (C/D) 66.99

SPINNAKER

Snooper Troops 1/2 (D) . 32.99/ea
Face Magic (D) 25.99
Story Machine (D) 25.99

STONEWARE PRODUCTS

D.B. Master (D) 144.99
D.B. Master Utility Pak #1 or #2 (D) 68.99
D.B. Master Stat Pak (D) .. 68.99

VISICORP

Visicalc (D) 164.99
Visiplot (D) 148.99
Desktop Plan II (D) 187.99
Visitrend/Plot (D) 198.99
Visidex (D) 187.99
Visiterm (D) 78.99
Visifile (D) 187.99
Visischedule (D) 198.99
Business Forecast (D) ... 78.99
Visilink (D) 187.99

ALSO FOR APPLE:

Avalon Hill, Budgeco, SirTech, Silicon Valley, Avant-Garde, and many more; please call or write for info.

COMMODORE

Commodore-64 .... 398.99
VIC 20 .... 138.99

SOFTWARE

AUTO. SIMULATIONS

Sword of Fargoal (C) 26.99
Monster Maze (R) 26.99

THORN/EMI

River Rescue (C) 27.99
Vic Music Composer (C) .. 27.99

COMMODORE

Gorf (R) 27.99
Omega Race (R) 27.99
Vic Avenger 21.99
Scott Adams Adventures for Vic 20
Pirate Cove (C/D) 27.99
Mission Impossible(C/D) . 27.99
Voodoo Castle (C/D) 27.99
Adventure Land (C/D) ... 27.99
Sargon II Chess (C/D) 28.99

Also Available - All products from American Peripherals, APX, Avalon Hill, Avante Garde, Compumax.

PRINTERS

C. ITOH

Prowriter (120 CPS) 398.99
Prowriter II (120 CPS) ... 648.99
Starwriter (40 CPS) 1298.99
Printmaster F-10 (55 CPS) 1548.99

ANADEX

All Anadex products .... CALL

EPSON

MX-80 Graftrax .... CALL
MX-100 Graftrax .... CALL
FX-80 .... 648.99
All Epson options, interfaces & cables .... CALL
HX-20 Computer .... CALL
QX-10 Computer .... CALL

STAR MICRONICS

Gemini 10 .... CALL
Gemini 15 .... CALL

NEC

All models .... CALL

Franklin Ace 1000 ...... CALL
Franklin Ace 1200 ...... CALL

DISKETTES

Maxell SSSD MDI .. 28.99 for 10
BASF .... 18.99 for 10
Elephant #1 .... 20.99 for 10
#2 .... 28.99 for 10

MODEMS

ANCHOR AUTOMATION

ANC-1 Signalman (RS-232) 78.99
ANC-2 (Atari 850) 78.99
Hayes .... Call for prices

DISK DRIVES

Rana Elite II (Apple II) ... 488.99
Elite I (Apple II) 288.99
Controller Can 98.99
Percom Single Side for Atari .... CALL

MONITORS

AMDEK

Video 300 12" Green 138.99
Video 300A 12" Amber .. 157.99

BMC

12" Green or black 148.99

NEC

JB 1201 12" grn w/sound 158.99
JC 1201 same composite 298.99
JB 1260 12" Grn for Apple 114.99

All above prices are for prepaid orders. MasterCharge & Visa add 4%: American Exp. add 5%: N.Y. Residents add Sales Tax. All software orders must include $2.00 for shipping. Hardware orders call for shipping info. All in stock items shipped promptly. Personal checks require 3-4 weeks clearance. Prices subject to change without notice. Not responsible for typographical errors. All Mdse. carries Mfrs. Guarantee. Hours M-Sat. 9:30-6:30 Eastern Time.

NEW YORK COMPUTER DISCOUNTERS
38 PARK ROW
N.Y.C., N.Y. 10038

NEWS&PRODUCTS

Arcade-Style Joystick

Newport Controls has released its Prostick II arcade-style replacement joystick for Atari 400 and 800, Commodore VIC-20, and various other videogame/computer applications.

The Prostick II is as durable and responsive as its predecessor, the Prostick Model 150. However, the case has been redesigned for easy hand-held operation, and the solid steel, short-throw control shaft with an arcade-style ball on the top gives the user precise, fast action.

The Prostick II includes a switchable 4-way/8-way gateplate. Setting the gateplate indicator to the 4-way position allows only horizontal and vertical motion, improving response for maze-type games. Setting the gateplate indicator to the 8-way position offers the regular precise 8-position action.

In addition, the Prostick II comes equipped with two "soft-touch" firing buttons which have been moved to the top end of the base, allowing either right- or left-hand play.

Newport Controls' Prostick II features a switchable gameplate, allowing 4-way or 8-way movement.

The $24.95 Prostick II is backed by a five-year limited warranty.

Newport Controls
15425 Los Gatos Boulevard
Los Gatos, CA 95030
(408)358-3439

Personal Fantasy Adventure

A combination adventure, strategy, and arcade game, *In Search Of The Most Amazing Thing*, has been released by Spinnaker Software. The game is designed to create an environment in which children are encouraged to experiment and discover, to negotiate with aliens, not destroy them.

The user begins his quest for The Most Amazing Thing by seeking the advice of Old Smoke Bailey, who explains the rules and outlines the hazards ahead. As the game progresses, the child learns decision making, note taking, map reading, trading and bargaining, music writing,

and drawing in his search for The Most Amazing Thing.

The game, written by Tom Snyder, comes with Jim Morrow's novel, *The Adventures of Smoke Bailey*, which was written as an accompaniment.

The $39.95 program is available for the Apple, Atari, and Commodore 64 computers.

Spinnaker Software
215 First Street
Cambridge, MA 02142
(617)868-4700

Space Game For Atari

First Star Software has released *Astro Chase*, a game by Fernando Herrera, the first winner of the Atari Star Award. *Astro Chase* is available for the Atari 400/800, and will be converted for use on other computers.

The game features high-resolution graphics, seven animated intermissions, 34 levels of play and Single Thrust Propulsion, a feature that allows a pilot to lock his craft on course and then fire independently in any direction. Suggested retail price is $29.95.

First Star Software
22 East 41st St.
New York, NY 10017
(212)532-4666

Commodore 64 To RS232 Interface Cable

Connecticut microComputer has introduced the ADA 6410, an RS232 interface cable for the Commodore 64.

The cable plugs into the 64's RS232 port, and provides voltage conversion to drive standard RS232 printers, terminals, and mainframes. The unit includes a six-foot cable, and all electronics are completely enclosed. Power is received from the computer. There is no special software needed. Address # 2 is used.

The ADA 6410 RS232 interface cable for the Commodore 64.

The ADA 6410 retails for $79.

Connecticut microComputer
36 Del Mar Drive
Brookfield, CT 06804
(203)775-4595

Keyboards For The Timex/Sinclair

Two companies have developed keyboards for the Timex/Sinclair

computers.

KOPAK Creations has introduced the $99.95 KOPAK Keyboard. Its features include: Sinclair/Timex markings on keys, five single-stroke keys, shift lock, a full-size space bar, additional ENTER and extra shift keys, sculptured keys, an optional numeric keypad, and an optional aluminum case.

KOPAK Creations, Inc.
448 West 55th Street
New York, NY 10019
(212)757-8698

The E-Z Key 60 keyboard is described as a "tactile feel" keyboard that plugs into the same connectors as the existing keyboard. It includes 60 keys, molded legends on key tops, keys for edit, delete, single and double quotes, colon, semicolon function, and stop, two shift keys, a numeric keypad, and an optional mounting base.

The E-Z Key 60 sells for $84.95.

E-Z Key
Suite 75A
711 Southern Artery
Quincy, MA 02169
(617)773-1187

Parallel Interface Cables For TI

TENEX Computer Marketing Systems has developed an interface cable making the Texas Instruments 99/4A computer compatible with several Centronics standard printers such as the Okidata and Smith-Corona TP-1. The parallel output from the TI's RS232 Interface is modified within the connector housing of the new TI/CEN cable. The cable is priced at $37.95.

TI/CEN parallel interface cable for the TI-99/4A.

TENEX Computer Marketing Systems
Box 6578
South Bend, IN 46660
(219)277-7726

Color Author For TRS-80

Radio Shack is introducing a courseware development system for the TRS-80 Color Computer 32K disk system.

Color Author allows educators with no previous experience to create instructional materials for delivery on the computer. The menu-driven system guides the user through the lesson-creating process. Lessons may contain tutorial text, questions, and graphics. The instructor creates lesson frames

PLAY **WORDRACE** FOR EXCITEMENT
COMPETITION
STRATEGY
DELIGHT . . .

AND YOUR VOCABULARY WILL GROW BY ITSELF.

Kids and adults, boost your verbal knowledge while you enjoy this fast-action word game. You do the playing — **WORDRACE** does the educating.

- FOR 1-4 PLAYERS
- 2000 WORDS AND DEFINITIONS
- CHOICE OF LEVELS
- APPLE VERSION SPEAKS
- AVAILABLE FOR ATARI AND APPLE COMPUTERS
- **NEW:** NOW AVAILABLE FOR COMMODORE 64

Disk version — **$24.95**
3 levels of play: Beginner, Regular, Challenge
For Atari 400/800/1200XL, requires 32K, disk drive, BASIC
For Apple II/II+/IIe, requires 48K, disk drive, Applesoft
For Commodore 64, requires disk drive, BASIC

Cassett version — **$19.95**
2 levels of play: Beginner, Intermediate
For Atari 400/800/1200XL, requires 16K, cassette drive, BASIC
For Commodore 64, requires cassette drive, BASIC

See **WORDRACE** and **ACCESSORY #1** at your software dealer. Or order by mail from Don't Ask — add $2.00 for shipping and handling to your check or money order. California residents add 6% sales tax (6.5% for L.A. County residents)

WORDRACE ACCESSORY #1 —
Use it with your WORDRACE disk to play 3 new games:

- **CLAIM TO FAME** — 600 FAMOUS PEOPLE IN HISTORY
- **SPORTS DERBY** — 600 SPORTS TRIVIA STUMPERS
- **INTERMEDIATE WORDRACE** — 600 NEW WORDS AND DEFINITIONS (AGES 11-16)

For Atari and Apple computers, disk only.
$19.95. Requires WORDRACE disk.

DON'T ASK COMPUTER SOFTWARE INC.
2265 Westwood Bl., Ste. B-150
Los Angeles, CA 90064
(213) 477-4514

Dealer inquiries invited.

Apple, Applesoft are trademarks of Apple Computer, Inc. Atari is a trademark of Atari, Inc. Commodore 64 is a trademark of Commodore Business Machines, Inc.

COMSTAR

INTRODUCING SOMETHING NEW AND USEFUL

THE ALOG PAGEWRITER

FOR ATARI 800/400

A Fast, Simple, Easy To Use and Inexpensive
WORD PROCESSOR

- Average Time To Master — 5 min
- Full Page Display Guide
- Help Screen With Command Summary
- Uses Standard ATARI Editing Keys
- Visible Adjustable Margins
- Stores Ten Pages on a Disc with No Swapping

Other Features Include:
Centering of Text • Set Tabs • Right Justification • Easy Text Insertion • Accepts Printer Control Commands

Requires at least 32k of memory and a 80 column printer.

List price, 39.95. Introductory price **$34.00.**
Manual included.

DEALER INQUIRIES INVITED

AIR* SHIPPING WITHIN 2 DAYS

ATARI

48K RAM (FOR 400)	$115
64K RAM (FOR 400)	135
ALIEN GROUP VOICE BOX (D,T)	119
S.A.M. (D) 8K	46
BIT 3 80 COL. BOARD	289
TECHNICAL NOTES	25
BOX OF DISKS (10)	19
PROWRITER PRINTER	419
B KEY 400 (KEYBOARD)	99
NEWPORT PROSTICK	27
MICROBITS MODEM	159
PRINTER INTERFACE	89
THE ATARI ASSEMBLER (BOOK)	12
VAL FORTH (D) 24K	45
PREPPIE (D,T) 16K	19
EASTERN FRONT (D,T) 16K	24
MINER 2049ER (C)	35
STARBOWL FOOTBALL (D,T) 24K	25
PILOT (C)	59
BASIC A + WITH OS/A + (D)32K	60
ASTRO CHASE (D) 32K	24
BAJA BUGGIES (D,T) 16K	23
JUMP MAN (D) 32K	29
CHOPLIFTER (C)	32
ZAXXON (D,T)	29

VIC-20

CARDBOARD (3 SLOT EXP.)	$ 33
VIDEOPAK WITH 16K (40/80 COL)	250
VIDEOPAK WITH 64K (40/80 COL)	319
PRINTER INTERFACE (PARALLEL)	55
KIDS AND THE VIC (BOOK)	17
16K RAM	85
8K RAM	55
HES MON (ASSEMBLER) (C)	29
HES WRITER (WORD PROC.) (C)	29
TURTLE GRAPHICS (C)	29
VIC FORTH (C)	49
QUICK BROWN FOX(WORD PROC.)(C)	54
SHAMUS (C)	29
PROTECTOR (C)	33
CHOPLIFTER (C)	34
APPLE PANIC (C)	34
VIC RABBIT (C)	35
UNWORD PROCESSOR (T) 5K	19
STARFIGHTER JOYSTICK	14
CARDETTE (CASS. INTERFACE)	33
CRUSH, CRUMBLE & CHOMP (T)21K	24
SWORD OF FARGOAL (T) 21K	23
VICAT (T) 8K	20
DEADLY DUCK (C)	28
TOTL MAIL LIST (T) 13K	19

HEAR ATARI SOUNDS THROUGH YOUR STEREO SPEAKERS WITH
STEREODAPTER — FOR ATARI 800
• NO ASSEMBLY REQUIRED • CAN USE STEREO HEADPHONES
• SHIELDED CABLE • ADJUST TONE & VOLUME WITH STEREO CONTROLS
STEREODAPTER WITH 16 FT CABLE $8 WITH 26 FT CABLE $10
DEALER INQUIRIES INVITED

C = CARTRIDGE D = DISK T = CASSETTE * MOST ITEMS

COMSTAR
P.O. BOX 1730
GOLETA, CA 93116
(805) 964-4660

ORDERS ONLY: CALL TOLL FREE 800-558-8803
IN CALIF. (805) 964-4660 or send check, money order or credit card number and exp. date. Include $2.00 for shipping. Add 3% for Visa or MC (except Calif.). Calif. add 6% tax. There is a $2.50 charge for COD. Please include type of computer. (Checks—10 days to clear.)

one at a time by filling in a form on the screen.

Display features include normal and double-sized text, reverse video, underlining, boldface and special graphics characters.

Color Author offers three delivery modes – instruction, review, and test. Student records can be kept and printed out for each mode.

Tandy Corporation/Radio Shack
1800 One Tandy Center
Fort Worth, TX 76102

Speech Synthesizer For Timex/Sinclair

The Parrot, a plug-in speech module for Timex/Sinclair computers, has been introduced by R.I.S.T. The Parrot is supplied with cassette software allowing the user to select, string together, and pronounce all 64 allophones in the English language, generating words, sentences and sound effects.

The Parrot plugs directly into the Sinclair ZX80/81 and the Timex/Sinclair 1000, and is piggyback expandable so other modules can be plugged in behind it. It includes an audio output jack for any 4 or 8 ohm speaker and volume control. The $89.95 Parrot also includes a power-input jack to provide additional power if modules other than The Parrot are being used.

The Parrot is a 3" by 3" by 1½" speech module for Timex/Sinclair computers.

R.I.S.T. Inc.
P.O. Box 499
Fort Hamilton Station
Brooklyn, NY 11209
(212)259-4934

Mail List For Commodore 64

A mailing list management program for the Commodore 64 is available from RAK Electronics. *Commodore 64 Maillist* constructs, sorts, maintains, and prints a mailing list of more than 300 names. The computer and cassette deck are the only requirements, but the system will work with a disk drive and printer.

Maillist is available on tape for $14.95 or disk for $17.95, plus a $2 shipping charge with each order.

RAK Electronics
P.O. Box 1585
Orange Park, FL 32073

Data Base Manager For Apple

Multi-Trieve, a data base management system for the Apple II+ and Apple IIe, has been released by Multisoft. Instead of displaying data in a traditional way, one record per screen, *Multi-Trieve* displays records in the form of a table, with field left/right justification, column headings, decimal point alignment, and running totals for numeric fields.

The horizontal scroll can shift a table to the right or left to view the fields that are outside of the physical boundaries of the screen.

Multi-Trieve supports the 80-column Text Card on Apple IIe, significantly increasing the number of fields (columns) that can be seen at any given time. *Multi-Trieve* is menu-driven, and it includes file Search/Edit, indexes, sorting at data entry, file reorganizing, multiple fields in selection criteria, and the ability to split a single-diskette file into two diskettes.

Multi-Trieve offers three printing methods, either in normal or condensed character set:

- Default Report Writer for printing of reports automatically formatted by the program.
- Form/Label Writer.
- A single-stroke snapshot of any data screen.

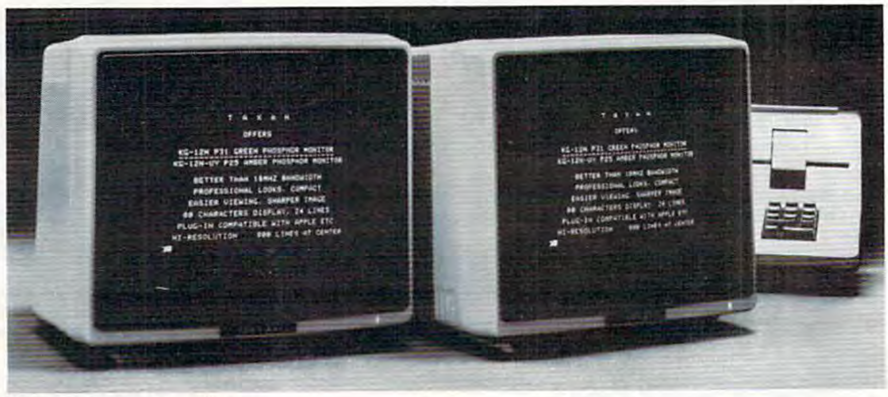
Multi-Trieve requires 48K RAM, at least one disk drive, and Applesoft in ROM. It will sell for $199.

Multisoft
120 East 90th Street
New York, NY 10028
(212) 534-0602

Resource Guide For Handicapped

The International Council for Computers in Education at the University of Oregon has published a resource guide on computers for the handicapped. The $7 guide, titled *Computer Technology for the Handicapped in Special Education and Rehabilitation: A Resource Guide,* is a comprehensive bibliography of 191 annotated references on computers for the handicapped. The references, drawn from books, periodicals, reports, and conference proceedings, cover a broad range of topics on physical and developmental handicaps.

International Council for Computers in Education
135 Education
University of Oregon
Eugene, OR 97403
(503) 686-4414



Taxan amber monitor.

Amber Monitor Available

TSK Electronics Corporation is introducing a 12-inch amber monochrome monitor, model KG-12NUY. The Taxan monitor features 800 lines at center, 18 MHz bandwidth, and a black, glare-proof screen.

Suggested retail price of the KG-12NUY is $179.

TSK Electronics Corporation
18005 Cortney Court
City of Industry, CA 91748
(213) 810-1291

Family Game For Atari

Avalon Hill has released a computer version of its popular *Facts In Five* game. *Computer Facts In Five* is a game of knowledge which pits players against time and each other. It includes options for solitaire play, doubles play, and party play.

In each round of the game, five subject categories are selected from a list of more than 1000. Five letters are associated with each category, and the players supply answers that start with the designated letters.

Game difficulty can be controlled, and modes for family and education use are available. The sand clock timer and scoring system add to the challenge.

Computer Facts In Five is available for the 48K Atari 800. Retail price is $26.

Avalon Hill Game Company
4517 Hartford Road
Baltimore, MD 21214
(301) 254-5300

Voice Input Module For Apple

MCE has announced a voice input

module for Apple Computers.

The MCE Voice Input Module is designed to operate virtually any Apple II or Apple IIe software by voice without software modification.

The MCE Voice Input Module will recognize anybody's voice with 98% + accuracy. It has an unlimited vocabulary using 80 word/phrase subsets at a time, and it allows for simultaneous input of data by voice and keyboard. Included with the module is utility software for building, editing, and training vocabulary. The vocabularies for BASIC, *VisiCalc*, *Wordstar*, *Magic Window*, and other popular programs are predefined in this software. 83 pages of documentation and a microphone complete the $825 system.

MCE Inc.
157 South Kalamazoo Mall
Kalamazoo, MI 49007
(800) 421-4157

Micro Math Educational Programs

Micro Math, a new series of educational math programs for 12-16 year olds, has been launched by PM International.

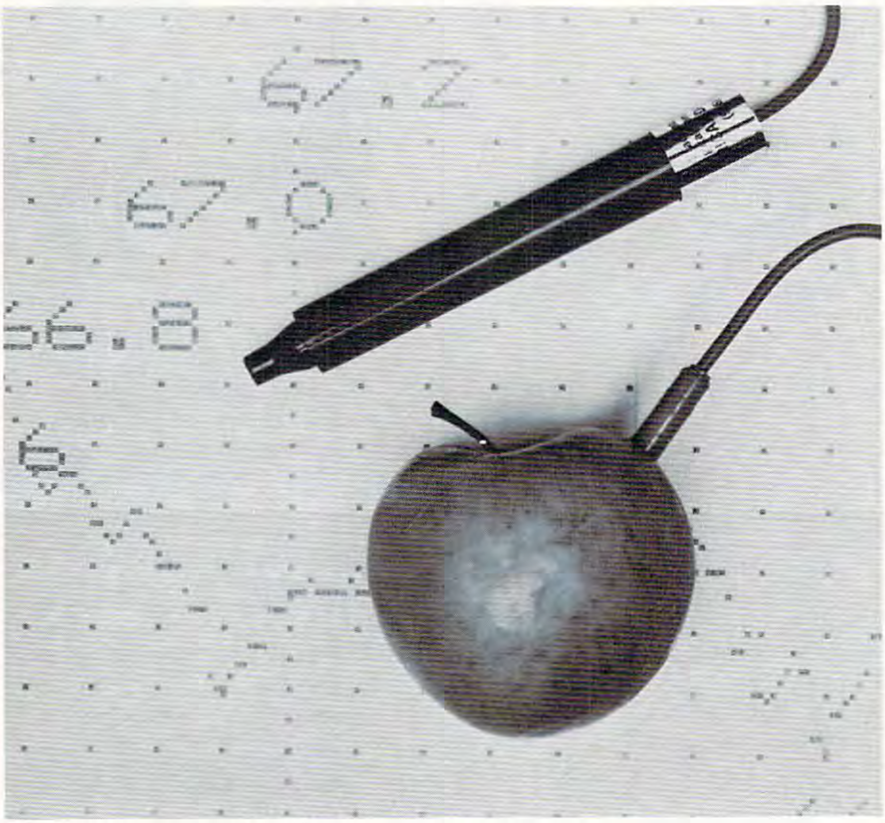
Part of the more extensive *Master Math* package, *Micro Math* is intended for use on the Sinclair ZX81, Timex 1000, Commodore PET, and VIC-20 in schools and colleges as a computer-assisted learning aid and at home as a self-tuition course.

A random number generator originates problems, and the student is prompted to answer.

If an incorrect answer is keyed in, the student is given a clue about the theory behind the problem, and is prompted to answer again. If an incorrect answer is again keyed in, the correct answer is displayed together with a complete explanation.

Micro Math, priced at $50, comprises 6 program suites, each containing 4 programs which are available on two cassettes. Subjects covered include algebra, geometry, differentiation-calculus, and statistics.

PM International
P.O. Box 87
Buckfield, ME 04220
(207) 336-2500



The Nth Degree digital temperature probe for Apple II.

Measure Temperatures With Apple

A temperature measuring and control system is now available as a peripheral to the Apple II computer.

The device accurately measures temperatures between -60 and +105 degrees Celsius. Changes of temperature as small as 1/100th of a degree may be detected. The system, produced by American Data Cable, uses a hand-held probe the size of a marking pen and attaches to the Apple II computer by a 6-foot cable. The system software is on a DOS 3.3-compatible, 5.25-inch diskette.

The system, called the Nth Degree, displays temperature readings in either Celsius, Farenheit or Kelvin. An optional program displays or can print out a continuous record of temperature changes in a "strip chart" format.

Accuracies of .01 degree may be obtained, and the probe can be re-calibrated for maximum accuracy over any temperature range. In addition, the system may be programmed for alarms and set points.

The probe can measure reaction rates and temperatures of reactants, and can monitor heating and cooling apparatus. The disk accepts up to 16 years' worth of temperature readings (taken one time an hour) or can store continuous samplings of temperatures taken once a second for 36 hours.

As many as 256 probes may be attached to one system. Temperature readings are made simply by placing the lightweight probe against the surface to be measured. When not in contact with a specific material, the probe measures the temperature of the air.

The model 551A probe with software, Apple II interface adapter and user's manual sells for $129.

American Data Cable, Inc.
2864 Ray Lawyer Dr., No. 205-352
Placerville, CA 95667
(916) 622-3465

English As A Second Language

The Soft Spot is now marketing *Teachers' Friend*, a program that teaches English as a second language (ESL) to students who can read English at the second-grade level. This 80-lesson curriculum, developed and refined over a two-year period, sells for $15 per lesson. The lessons can be used independently of one another. Students can go right to the lesson they need, when they need it, without going through all lessons in sequence.

Teachers' Friend also serves as an anticipatory instructional device (AID) that allows the student to prepare for difficult lessons in advance. For instance, a student who is weak in verb conjugation can prepare before the class and reinforce learning after class.

Each lesson takes about 10-15 minutes to complete. Each lesson teaches a basic English skill, such as spelling, punctuation, parts of speech, verb tenses, syntax, pronunciation, drawing conclusions, making generalizations, developing vocabulary, alphabetical order, and others.

Using six question formats, *Teachers' Friend* works with Apple or Radio Shack Computers. Students do not need to be proficient at running a microcomputer, or even know how to type.

The Soft Spot, Ltd.
800 East Arapaho, Suite 110
Richardson, TX 75081
(214)669-1779

Everything You Always Wanted To Know About Home Computers

PolyGram Records recently released *Everything You Always Wanted To Know About Home Computers*, an album narrated by Steve Allen and Jayne Meadows. The album is accompanied by a 16-page instruction booklet with a glossary, computer literacy quiz, diagrams, and a simulated computer keyboard with sample programs.

The album answers such questions as: Why do I need a computer? How hard is it to use? What can I do with one? How does a computer do all that? And just what do the arcane terms of computer language mean?

Everything You Always Wanted To Know About Home Computers was written by Arnold Friedman, Department Chairman of Computer Studies in the Great Neck Public Schools Adult Program of Great Neck, New York. Friedman directs a private school for computer studies and has written computer programming textbooks, magazine articles, and audio-visual materials for classroom use.

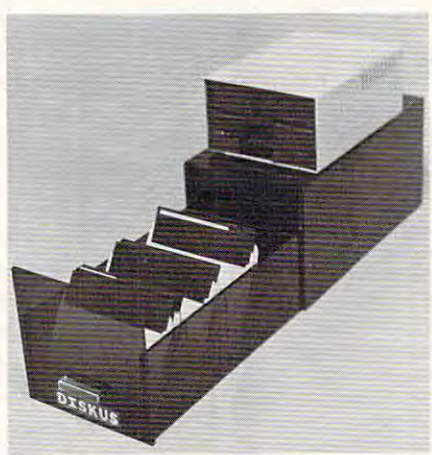
PolyGram Records
810 Seventh Avenue
New York, NY 10019
(212)399-7067

or

8255 Sunset Blvd.
Los Angeles, CA 90046
(213)650-8300

Diskette Storage

The *Diskus*, a storage unit for up to 125 diskettes, is now available from Diskus. Five disk compartments are separated by removable clear inserts. Index tab dividers are also included.



The Diskus storage file.

The unit is compatible with most disk drives and computer desk setups. Made of a dark, see-through acrylic, the storage unit permits easy access, allowing the user to have a consolidated disk library close at hand. The price is $75.95.

Diskus
7051 Hanna Ave.
Canoga Park, CA 91304

Folklife Club Software Available To Commodore Users

The Folklife Terminal Club, a Commodore users group, has announced it is making its software library available to other users of Commodore equipment. The club's archives contain more than 5000 public domain programs. The programs are stored on diskettes and are usable on various configurations of PET, CBM, and VIC computers. The software itself is free. There is a copying and mailing fee of $10 per diskette. The first diskette that should be ordered is the *Catalog Disk*, which contains an Automatic Disk Cataloging Program, a listing of all the available programs and complete instructions.

Folklife Club
Box 2222
Mt. Vernon, NY 10551

Math Concepts For Atari

Mind Movers has released a mathematics program called *Secret Formula*. Users discover and understand math concepts and relationships through a sequenced developmental program for all ability levels.

The user controls the level of difficulty, and can create his own program. The program can be used by individuals or by small groups for interaction and competition.

Secret Formula is available for the Atari 800 with 48K and requires a disk drive and BASIC cartridge. The cost is $59.95.

Mind Movers Inc.
4286 Redwood Hwy., Suite 245
San Rafael, CA 94903
(415)499-8281

Hewitt's Computer Shop

COMMODORE

CBM 64	$425.00
VIC-20	$149.95
1525 Printer	$319.95
1541 Disk Drive	$325.95
VIC Modem	$90.00
1520 Color Printer/Plotter	$179.95
Color Monitor	$269.95

CARDCO

Card Board	$27.95
Cardette	$27.95
Cardwritter	$27.95
Card ?	$59.95

DATA 20

16K RAM Expansion	$74.95
Video Pak w/16 K RAM	$215.95
Printer Interface	$55.95

MIDWEST MICRO

Terminal 40	$26.95
Un-word Processor 2	$16.95
Printer Interface	$49.95

SCIENTIFIC MICROTRONICS

6 slot expansion motherboard Buffered w/power supply	$89.95
Color monitor cable	$19.95

INDEPENDENT FOR ATARI

BIT-3

Full view 80	$299.95
32K RAM expansion	$149.95

MICRO CONNECTION

Direct connect modem w/tp-850 not needed	$199.95
Auto dial auto answer opt.	$65.00

MICROTEK

16K RAM expansion	$79.95
32K RAM expansion	$108.00
Centronics printer cable	$29.00
Serial cable	$29.00

MOSAIC

32K RAM expansion	$130.00
Expander	$98.00
Adapter	$70.00

MPC

16K RAM expansion	$84.95
32K RAM expansion	$141.95

SYMTEC

Light pen	$115.95

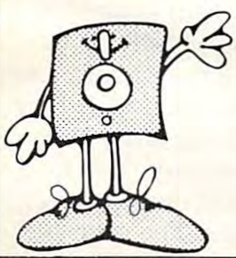
SPECIALS

AMDEC COLOR I MONITOR
$299.95

"THE DATA BASE"
FOR VIC-20, 64 -- BY JOHN CALHOUN. REQUIRES DISK DRIVE AND 16K.
$69.95

Hewitt's Computer Shop
2448 Menaul NE, Albuquerque NM 87107, (505) 883-0984
HOT LINE: 1-800-821-9287

Verbatim
flexible disks



Call Free **(800) 235-4137** for prices and information. Dealer inquiries invited. C.O.D. and charge cards accepted.

PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. In Cal. call
(800) 592-5935 or
(805) 543-1037

VISA

Cassette/Magazine/Newsletter
A Monthly Tradition
in the World of Computing for

20 Load™

VIC 20™ USERS

June is a'bustin' out all over for **20 LOAD**. Is your June bride crying because of the high cost of software? Boo Hoo. Well, dry those tears. Receive each month a bouquet containing a cassette magazine and a newsletter, by subscribing to **20 LOAD**. Turn those tears into happy times — NOW. Subscription rates are: 1 yr. - $50.00, 6 mo. - $30.00, or $6.00 for a single issue. Write to **20 LOAD**, 550 Grant Ave., Junction City, Kansas 66441. For more info, our phone number is (913) 762-4730.

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

New Commodore 64 Games!

An introductory offer from

ADVANCED ARCADEWARE

HEAD ON
A daring and dangerous death race.

TARGET COMMAND
Projectiles destroy a city.

ALIEN INVASION
4 players, creatures descend.

SERPENTS
2 players, snakes entangle each other.

4 Game Pack suggested list price $59.95

NOW $29^95

To order: Mail address along with check or money order to:

Advanced Arcadeware
P.O. Box 845
Thomasville, NC 27360
(919) 431-3231

Specify Disk-Pack or Cassette Pack
Add $2.00 for shipping

Allow 2 weeks for delivery

New Epson FX-80 Printer

The FX-80, a high-performance bi-directional printer with a printing speed of 160 characters per second, is now available from Epson America.

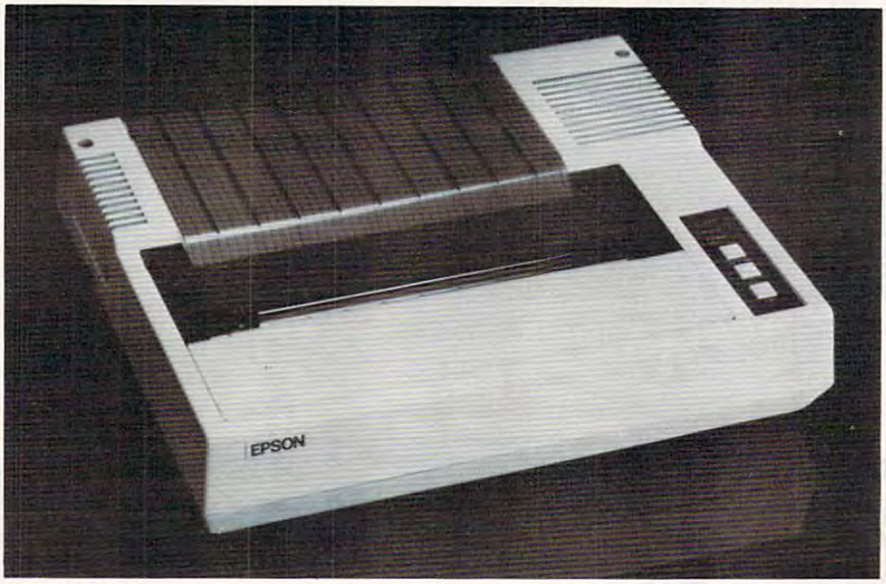
The FX-80 offers a software-selectable choice of elite (12 cpi) or pica (10 cpi) print spacing. In addition, users can send their own special fonts from their computer system to the printer, downloading the font into the printer's memory. This feature will be of particular help to those using math, engineering, foreign language, or medical applications.

The FX-80 also features a one-to-one graphics ratio – the dot-matrix has the same scale vertically as horizontally; accurate graphics, including true circles, can be drawn with the dot-addressable graphics capability. Also incorporated into the new printer is a 2K-byte buffer, which allows buffered printing on longer productions.

The new Epson printer provides 9x9 dot-matrix characters with full descenders and is downward compatible with the Epson MX Series of printers. Also featured is proportional spacing, pin- and friction-feeds, and a standard parallel communications interface, with serial or IEEE 488 interfaces also available as options.

Four different printing densities – normal, emphasized, bold, and double-emphasized – are available.

The short-form tear-off bar will aid in the production of computer-generated paychecks and other pre-printed forms. The tear-off bar separates the paper from the printer one-inch from the last printed line.



Epson's dot-matrix printer.

The Epson FX-80 printer retails for $699.

Epson America, Inc.
3415 Kashiwa Street
Torrance, CA 90505
(213)539-9140

Statistical Graphs For The Timex/Sinclair

Practical Computer Products has announced the release of their statistical aid, *Graphpak,* for the Timex/Sinclair 1000. The program presents numerical information in a visual format: bar graphs, line graphs, pie charts, and area graphs (rectangles divided to show percentages).

Graphpak plots scientific data and assists in business and family financial planning. It is a 10K BASIC program which is listable. Available on cassette for $14.95. Shipping and handling is included, but California residents should include 6% sales tax. Send an SASE for details.

Practical Computer Products
21111 Strathmoor Lane
Huntington Beach, CA 92646

EPROM Programmer For Atari

Creative Firmware of Dallas, Texas, recently introduced a low-cost EPROM development system for Atari computers. This system includes the following items:

1. The Memory-Maker EPROM Programmer comes in kit form and permits programming of 2716 and 2532 EPROMs. The machine language software is available for either cassette- or disk-based systems and includes EPROM blank checking, programming, and verifying. Also included are the capabilities to read most 2K and 4K ROMs, edit any loaded software, list this software to the screen or printer, examine any portion of computer memory, and save any portion of computer memory to disk or tape. The save and load functions are compatible with DOS binary files. Assembly time for this kit is approximately one hour. The kit includes all parts, including power supply and zero-insertion force socket. The case is optional.

2. For those desiring to program 2732 or 2732A EPROMs, the Creative Firmware 2732/2732A Programming Adapter kit. This unit drops into the zero-insertion force socket on the Memory-Maker and permits switch selection of 2732 or 2732A EPROMs. It contains its own zero-insertion force socket. Assembly time for this kit is approximately 15 minutes.

3. A Cartridge EPROM Board kit. This unit permits up to 8K of EPROM space on a board which plugs into the Atari cartridge slots. This board takes one or two 2716, 2532, 2732, or 2732A EPROMs. The connector is gold-plated. This kit can be assembled in 15 minutes.

Pricing for the above items:

* Memory-Maker EPROM Programmer: $79
* Custom Memory-Maker case: $15
* 2732/2732A Programming Adapter: $17.50
* Cartridge EPROM Board: $14.89

These kits are available from:

HARDSEL
P.O. Box 565
Metuchen, NJ 08840
1(800)835-5465

Microcomputers In Education

The Oryx Press announces publication of *Microcomputers in Education: A Handbook of Resources.* The book, edited by Katherine Clay, is a guide to the literature, materials, and resources on the use of microcomputers in schools.

Microcomputers in Education covers materials published from the beginning of the microcomputer revolution in 1976 to March 1982. The major portion of the book is an annotated bibliography of books, journal articles, and microfiche covering the state of the art, from Philosophy and Futures/Trends to Computer Literacy, Classroom Applications, Teacher/Administrator Education, and more. Included is a section on references and resources, including bibliographies, glossaries, sources of funding, and book reviews. In addition, there is an appendix covering sources of information and assistance such as computer journals, user groups, associations, microcomputer centers, and software vendors.

The price is $18.50, and there is no charge for postage and handling on prepaid orders.

Microcomputers in Education:
A Handbook of Resources
The Oryx Press
2214 North Central
Phoenix, AZ 85004

Heathcliff, Banjo, And Terrytoons Software

Datasoft recently obtained the rights to market home computer software based on Heathcliff, the syndicated cartoon cat; Banjo, the Woodpile Cat; and approximately 200 Terrytoons cartoon characters, including such favorites as Heckle and Jeckle, Pearl Pureheart, and Deputy Dog.

Heathcliff, now nine years old, was created by George Gately. Along with a cast of supporting characters, Heathcliff is now also a Saturday morning TV star. Datasoft expects to introduce its Heathcliff product in the second quarter of this year.

Banjo the Woodpile Cat, the first film done by Don Bluth Productions, was aired last year on ABC-TV. The film presents the whimsical story of Banjo, a country cat who wanders off to the big city. Although Banjo discovers excitement and adventure, and learns the value of friendship, he is only too glad to return home. Datasoft plans to introduce Banjo software during the spring, possibly in April.

The Terrytoons software should be introduced around mid-year.

Datasoft, Inc.
16 East 52 Street
New York, NY 10022
(212)355-5049

CALENDAR

June 7-11, University of Rochester, Rochester, NY. The third annual Rochester Forth Applications Conference. This year's conference will focus on Forth and robotics. Other sessions will cover Forth and education, VLSI design, graphics, and finite state machines. For more information, contact Diane Ranocchia, Institute for Applied Forth Research, Inc., 70 Elmwood Ave., Rochester, NY 14611; (716) 235-0168.

June 20 - July 15, Rutgers University, Rutgers, NJ. Computers in Education '83 – a three-day conference and a four-week summer institute. The conference, which will discuss "Necessary Direction for Computer Education; Navigational Aids for the 80s," begins June 27. The summer institute offers 40 professional development courses ranging from one to twelve days. Continuing Education Units and Graduate Credit are available. Preregistration is necessary. Additional information is available from Dr. Mitchell E. Batoff, Director, CE '83, Institute for Professional Development, 245 Nassau St., Suite D, Princeton, NJ 08540; (609) 924-8333.

June 27, Stanford University, Stanford, CA. International Institute on Microcomputers in Education, sponsored by Interactive Sciences, Inc., and Stanford University's School of Education. Five-week course offers exposure to computer technology and a discussion of ways to integrate computers into the schools. For more information, write the School of Education, Stanford University, Stanford, CA 94305; (415) 497-2102.

June 27-29, New York Hilton, NY. Videotex '83, an international conference and exhibition on the technology, current use and projections for videotex. For information, write Pam Fendel, London Online, Inc., 1133 Avenue of the Americas, 33rd Floor, New York, NY 10036; (212) 692-9003.

July 20-22, Eugene Conference Center, Eugene, OR. "The Computer: Extension of the Human Mind II" will explore the classroom applications of computers and will discuss current research on computers in education. For information, contact Jude Ridge, College of Education, University of Oregon, Eugene, OR 97403; (503) 686-3405.

August 10-12, Madison, WI. The second annual Microcomputers and High Technology Conference in Vocational Education. Beginning and advanced classes on programming, PILOT, CAD, courseware design, and demonstrations of vocational education programs not in use. For information, contact Dr. Judith Rodenstein, Vocational Studies Center, University of Wisconsin-Madison, 1025 W. Johnson St., 964 Educational Science Building, Madison, WI 53706; (608)263-4367.

COMPUTE! *welcomes notices of upcoming events and requests that the sponsors send a short description, their name and phone number, and an address to which interested readers may write for further information. Please send notices at least three months before the date of the event, to: Calendar, P.O. Box 5406, Greensboro, NC 27403.*

New Product releases are selected from submissions for reasons of timeliness, available space, and general interest to our readers. We regret that we are unable to select all new product submissions for publication. Readers should be aware that we present here some edited version of material submitted by vendors and are unable to vouch for its accuracy at time of publication. ©

PRODUCT MART

Forth

Commodore 64

on disk or tape

- $37 full fig-Forth with assembler, editor, floating point, graphics, sound, tutorial and support
- $13 Games, Graphics, Sounds tutorial
- $13 Artificial Intelligence tutorial

future extensions

for details try...

MINDWORKS
Rte. 2, Box 28A
Winters, CA 95694
(916) 795-2322

VIC-20
CBM-64

SOFTWARE
AT
UNBEATABLE $$

Send for Free Catalogue:

SOFTWARE UNLIMITED
P.O. Box 582
Bethpage, New York 11714

PALACE SOFTWARE

GAMES PROGRAMMERS & DISTRIBUTORS

Palace Software, part of a leading British film and video company, is looking for games for Atari 400/800, TI99/4A, VIC 20 and CBM 64 for distribution in the UK and Europe. High royalties will be paid for top quality and highly original material. Send samples to: Pete Stone, Palace Software, 62-64 Kensington High Street, London W8, England (Tel: London 937 6258)

WHAT'S
SII
?

$$$$
SOFTWARE

For Most Personal Computers Available At

WHOLESALE

For Use Or

PROFIT

Earn $$$ Full or Part Time.

Send $1.00 to Unisoft
PO Box 60104
Duluth, MN

or Call 218-727-1175 or 218-525-6882

DEALER INQUIRIES INVITED

RUSSIAN

FOR THE

VIC-20

BEGINNING RUSSIAN

LEARN THE ALPHABET AND BASIC VOCABULARY ON THE UNEXPANDED VIC. INCLUDES CYRILLIC 'TYPING' FUNCTION.

tape 23.95 *IL res. add 7% tax*

Send check or m.o. to:

Russian Software
Box 36, 1744 W. Devon
Chicago IL 60660

VIC IS A REGISTERED TRADEMARK OF COMMODORE BUSINESS MACHINES

STOP LOOKING!

THIS IS

✱ Ware It's At! ✱

FOR THE

COMMODORE 64

Luna	Victory
Firefighter 64..........c15.95	Grove Robbers...........c12.50
Pedestrian.............. "	Annihilator.............c16.50
Money Minefield......... "	Galactic Conquest....... "
Hyper Hen............... "	Adventure Pack 1 or 2...c12.50
Comm.Data	Tronix
Escape MCP.......c14.00/d18.50	Sidewinder..............c23.95
Centropods....... " "	Swarm................... "
Pakacuda......... " "	Gold Fever.............. "
Nüfekop	For Vic-20
3-D Man 64..............c15.95	Gold Fever...............23.95
Exterminator 64......... "	Deadly Skies............ "
Defender on Tri......... "	Scorpion................ "

Also: Word processing, accounting, educational, and more games - Lowest prices on Totl., Pixell, Programmers Institute, Synapse, Sirius. The Most, the Newest, the Lowest!

Send for Free Catalog

Ware It's At!

Send Check or MO. Include $1.50 Shipping

P.O. Box 27
Galveston, TX 77553

(*) (/)

UNICALC

(-) (+)

SCIENTIFIC CALCULATOR PROGRAM FOR

VIC-20

- CALCULATOR DISPLAY
- TWO MEMORY REGISTERS
- ARITHMETIC, TRIG, ETC.

5k, CASSETTE $7.95 + 75¢ SHIPPING
MN RESIDENTS ADD 6% TAX

ALLEGIANCE ENTERPRISES
P.O. BOX 8939
WHITE BEAR LAKE, MN 55110

LIST OF OTHER SOFTWARE FREE
DEALERS SOUGHT

VIC-20? Stock Portfolio?

Do you know todays value of your portfolio?
Do you know the profit or loss on your portfolio?

Get it together with:

"Portfolio Manager"

Requires 16K RAM Expansion
Program on Cassette Tape: **$29.95**

Stock	# Shares	Cost / Share	% Change	Dollar Change
IBM	200	97¾	18.5	3,616.75
GM	100	95⅛	29.3	2,801.08

SEND CHECK OR MONEY ORDER TO:

BASIC BYTE, INC.

P.O. BOX 924, SOUTHFIELD, MI 48037-0924

Phone Orders: 1-800-835-2246 Ext. 237
Kansas Residents: 1-800-362-2421 Ext. 237

VISA AND MASTERCARD ACCEPTED
Michigan Residents add 4% Sales Tax.

- DEALER INQUIRIES INVITED -

REFILL YOUR OWN PRINTER RIBBON CARTRIDGES

$$ save $$ save $$ save $$ save $$ save$$

NEC 8023A	Refills 3 for	$ 8.95
NEC SPINWRITER	" 3 for	9.75
COMMODORE PET 8023P	" 3 for	8.95
EPSON MX70,80	" 3 for	9.95
EPSON MX100	" 3 for	12.95
CENTRONICS 704	" 3 for	15.50
DIABLO HYTYPE I	" 3 for	8.49
DIABLO HYTYPE II (sp'fy 1/4,5/16)	3/	9.45
DIABLO MATRIX	Refills 3 for	8.95
RADIO SHACK DAISYWHEEL II	" 3 for	8.95
RADIO SHACK LP III,V	" 3 for	9.95
ANADEX 9500	" 3 for	12.95
QUME	" 3 for	8.95

ALL REFILLS ARE BLACK NYLON

ESD p.o.box 952
cleveland, oh 44120

NO Shpg Chgs 3 or more. See our ads APR/MAY COMPUTE! for complete ribbons. Ask for YOUR PRINTER/OTHER COLORS! (OH Incl. sls tx)

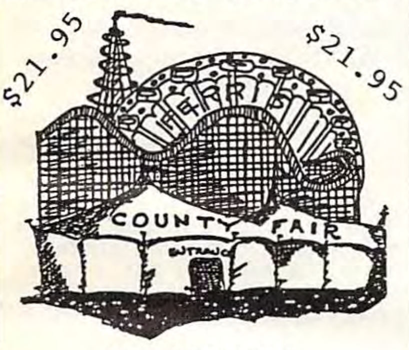
COMPUTER ACCESSORIES

- Perfectdata Floppy Disk Drive Cleaning System. For 5.25" or 8" disk drives. Specify size. **$22.75.**
- Organize your diskettes with an Innovative Concepts Flip "n" File. Holds up to 50 Diskettes in a handsome smoke colored transparent plastic case. **Case for 5.25" disks. $23.75. Case for 8" disks. $29.75.**
- **WHAT'S YOUR FAVORITE NUMBER?** New for spring. Top quality shirts with "64" or "20" printed in large numerals on both front and back. Specify color and size and number choice. Red or blue, S-M-L-XL **$7.50 each.**

- **ORDERING INFORMATION** Telephone and mail orders accepted. Phone (301) 994-1122.
- **SHIPPING** Add $1.50 to all orders for shipping. We pay balance for UPS service on all orders. Add $2.00 for COD. Maryland residents please add 5% state sales tax.
- WRITE OR CALL FOR FREE COMPUTER CATALOG: SURGE PROTECTORS, DUST COVERS, BOOKS, DISK MAINTENANCE, BLANK MEDIA, AND MUCH MORE.

Mercury Micro
Dept. C
Cherry Field Road
Drayden, MD. 20630

COMMODORE 64
SHOOTING GALLERY



100% MACHINE CODE
DEALER DISCOUNTS
AVAILABLE

EMERALD SOFTWARE
122 BANGOR STREET
LINDENHURST, NY 11757
(516) 226-5849

for use with the
COMMODORE 64

PRO BOWL

PRO BOWL

New FROM DATA PLUS

PROVIDES ACCURATE RECORD KEEPING
& HANDICAPPING FOR BOWLING LEAGUES !
REQUIRES 1541 DISC DRIVE & PRINTER

VISA ☐ MASTER CHARGE & C.O.D.'s ACCEPTED **$199.95**

*** AVAILABLE FROM A-1 SERVICES ***

15 DAY F R E E HOME TRIAL ON ALL MERCHANDISE

FREE COMMODORE 64 CATALOG WITH ORDER
OR SEND $1.00 TO BE REFUNDED WITH FIRST ORDER

A-1 COMPUTER SERVICES (509) 783-4980
7103 W. CLEARWATER - H- 227 or
KENNEWICK, WA 99336 (509) 783-5966

☐ DEALER INQUIRIES WELCOME ☐

VIC-20

Dealers call 212-499-5400 direct for pricing.

NEW ITEMS!

HARDWARE

UPA-20 Standard Centronics Cable $19.95
Driver Listing Included!
UCA-20 Universal Cassette Cable $19.95
Use any cassette machine!
16K RAM/ROM Board Jumpered for any 8K block
Socketed Board with support chips $24.95
Populated with 8K $49.95
Populated with 16K $79.95
2716 EPROMs for RAM/ROM Board ... @ $7.50
3 Slot Memory Port Expander $19.95

SOFTWARE

WORDWIZ for the Unexpanded VIC-20 ... $14.95
A nice small Word Processor
Mailing List requires 8K minimum $14.95
Sorts, selects, and prints labels

2 FREE DISASSEMBLERS
when you send $1 for our CATALOG.

WORLD ELECTRONICS
177 27th Street
Brooklyn, N.Y. 11232
A Division of World International Trading Corp.

Vic-20 ROMS at cassette prices!

thru July 5 **$24.95**

Aggressor (H.E.S.)	Music Composer (Thorn)
Apple Panic (Creative)	River Rescue (Thorn)
Astroblitz (Creative)	Serpentine (Creative)
Black Hole (Creative)	Terraguard (Creative)
Choplifter (Creative)	Trashman (Creative)
Gridrunner (H.E.S.)	Video Mania (Creative)

Huge savings on all programs from Commodore, Creative, Epyx, H.E.S., Thorn-EMI and UMI. Write or call for FREE catalog.

Mail order prices do not apply at our retail location.

$2.50 shipping & handling per order. Add 4% for credit card orders. Mass. residents add 5% sales tax.

SOFTWARE PLUS

Orders and info.: **(617) 369-1992**
Phone: 10-6 EST Mon. thru Sat.

45 Walden St., Box AA, Concord, MA 01742

COMMODORE 64 VIC 20

PROTECTION + PRIDE

Protect your investment while enhancing your Vic-20 or Commodore 64 with our handcrafted acrylic covers. This is not a thin vinyl or cloth cover but a full 1/8 inch thick acrylic with chemically bonded seams. All handmade and guaranteed not to break or we will repair at no charge for one year. Beautifully enhances the looks of your system while providing protection from dust, spills, unauthorized fingers, ashes, and other mishaps.

CBM 64/VIC20 CLEAR COVER $17.95

Send check or money order to:
T. G.'s Computer Covers
c/o Dennis Graham
P. O. Box 1594
Maryville, TN 37801
Please add $1.75 for shipping/handling.

Commodore 64 and Vic 20
Products of Commodore Business Machines, Inc.

LARGEST SELECTION OF COMPUTER BOOKS ON THE EAST COAST

Books shipped within 24 hours
Toll-free order line
Retail discounts + terms
Individual orders accepted
(prepayment required)

We stock the latest books from:

Addison-Wesley

Alfred	Dilithium
Arcsoft	W.H. Freeman
Ballinger	Harper & Row
Birkhauser	Hayden
Brady	McGraw-Hill
William C. Brown	Osborne
CBI	Que
Chilton	Howard W. Sams
Compusoft	Sybex
Compute!	TAB
Creative Computing	Weber
Design Enterprise	John Wiley

Call or write for our complete catalog

THE BOOK CARRIER
9121 Industrial Court • Gaithersburg, MD 20877
301/258-1177 800/638-4108

RUNWAY 64

A FLIGHT SIMULATOR GAME FOR THE COMMODORE 64* COMPUTER. CASSETTE OR DISK. **$25.00**

RUNWAY 20

A FLIGHT SIMULATOR GAME FOR THE VIC-20* COMPUTER. REQUIRES 16K EXPANSION. COLOR, SOUND & JOYSTICK OPTION. CASSETTE OR DISK. **$25.00**

SKY PILOT

A FLIGHT SIMULATOR GAME FOR THE COMMODORE VIC-20* COMPUTER. CASSETTE OR DISK. REQUIRES 8K EXPANSION. JOYSTICK OPTION **$18.00**

SUSIE SOFTWARE
709 WILSHIRE DRIVE
MT. PROSPECT, IL 60056
312/394-5165

*REGISTERED TRADEMARK OF COMMODORE BUSINESS MACHINES

NEW NEW

NEC® 3500 Multistrike Replacement Cartridge Ribbon*

The NEC® 3500 Multistrike Replacement Ribbon, manufactured by Aspen Ribbons, Inc., is now available for delivery.

Prices range from $7.50 each to $3.75 each depending on the quantity ordered. Colors are available, by special request, for an additional $.50 per ribbon—choose from red, green, blue, brown, or purple.

Call or write for YOUR FREE, NEW 1983 CATALOGUE.

*Aspen Ribbons, Inc. is not affiliated with NEC (Nippon Electric Company).

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

VIC-20 | NO MORE | VIC-20
VIC-20 | CHECK | VIC-20
VIC-20 | WRITING | VIC-20

BILL WRITER/SUMMARY

BILL WRITER/SUMMARY was designed for home accounts on the VIC-20 COMPUTER. Four(4) options are provided ranging from viewing monthly account data/checks on the screen to printing monthly account data/checks. **NO SPECIAL CHECKS ARE NEEDED** if your printer can take double 18 pound paper. Accounts paid data can be written to tape for use with BILL SUMMARY to provide yearly summaries of accounts for tax purposes. Tape drive, extra 16K memory and 80 column printer required. To order, mail check you will use in printer to **UHL RESEARCH ASSOCIATES, INC.**, 7926 Berner St., Long Beach, CA 90808 for $49.95 plus $3.00 shipping.

TIMEX SINCLAIR 1000

SINCLAIR ZX81 SOFTWARE

CALL OR WRITE FOR FREE CATALOG

SOFTSYNC, INC.
14 E. 34th St. NY, NY 10016
212-685-2080

EUCHRE FANS!!!

YOU'RE GONNA LOVE OUR FIRST OFFERING

"EUCHRE FOR FOUR"

(For You Alone)

* * * * * *

EXCELLENT 4-HAND EUCHRE FOR YOU AND YOUR COMPUTER

* * * * * *

AVAILABLE ON DISKETTE OR CASSETTE FOR COMMODORE 64* OR EXPANDED VIC-20* (NEED 24K EXP.)
(PLEASE SPECIFY WHICH)

VISA **$29.95** MC
(NY State Residents add 7% Sales Tax)

HUTCHCRAFT ASSOCIATES
1132 Dryden Road Ithaca, NY 14850
607 347-4782
DEALER INQUIRIES WELCOME

*TRADEMARKS OF COMMODORE BUSINESS MACHINES

ATARI OWNERS REJOICE

Quality software at affordable prices
Want to learn more than hand-eye coordination?
Try our educational-games.
Have fun and learn too.
Ages 8 to adult

CRYPTOGRAMS, KNOW YOUR STATE, STAR GAZER, PRESIDENTS, LOAN AM, FLAGS AND FACTS OF THE AMERICAS, FLAGS AND FACTS OF EUROPE

MAIL ORDER DISCOUNT – $14.95 each (cassette or diskette)

DACUS SOFTWARE
19900 SW 92nd Ave
Miami, Fl 33157
(305) 251-0392

VIC-COMMODORE-HOBBYIST

VIC 20 4 slot expander board $44*

COMMODORE 64 7 slot expander board $69*

VIC 20 24 static RAM with slots for up to 8K EPROM **$159***

COMMODORE 64 EPROM card slots for up to 4 2732 EPROMs **$59***

VIC 20/COMMODORE 64 300 baud modem with terminal emulator software **$89***

8085 based CRT electronics $249*

*plus shipping and handling
Washington residents add sales tax

To order: phone toll free
1-800-858-8020

BAZ Electronics
P.O. Box 4895 Federal Way, WA 98003
(206) 874-3029

VISA MasterCard

INTRODUCING **"BUDGET AIDE"**
FOR COMMODORE 64

Family budget assistance in money management, menu driven, chart of accounts, user define categories, integrated checkbook with reconciliation, color graph anaylisis of budget management and more. **$69.95**

Also:

- Checkbook Accounting (an unlimited check book) $24.95
- Home Inventory list all items including insurance needs $29.95
- Small Business Inventory $35.95
- Utility Tracker & Forecaster $29.95

(send check or money order and $1.50 for handling)

To: **MicroTechniques**
P.O. Box 192
Chesterfield, MO 63017

FREE

Write for your free catalog of

Software and Books

for your **PET, VIC** and **Commodore 64**

TIS, inc.
Box 921 Dept. C
Los Alamos, NM 87544