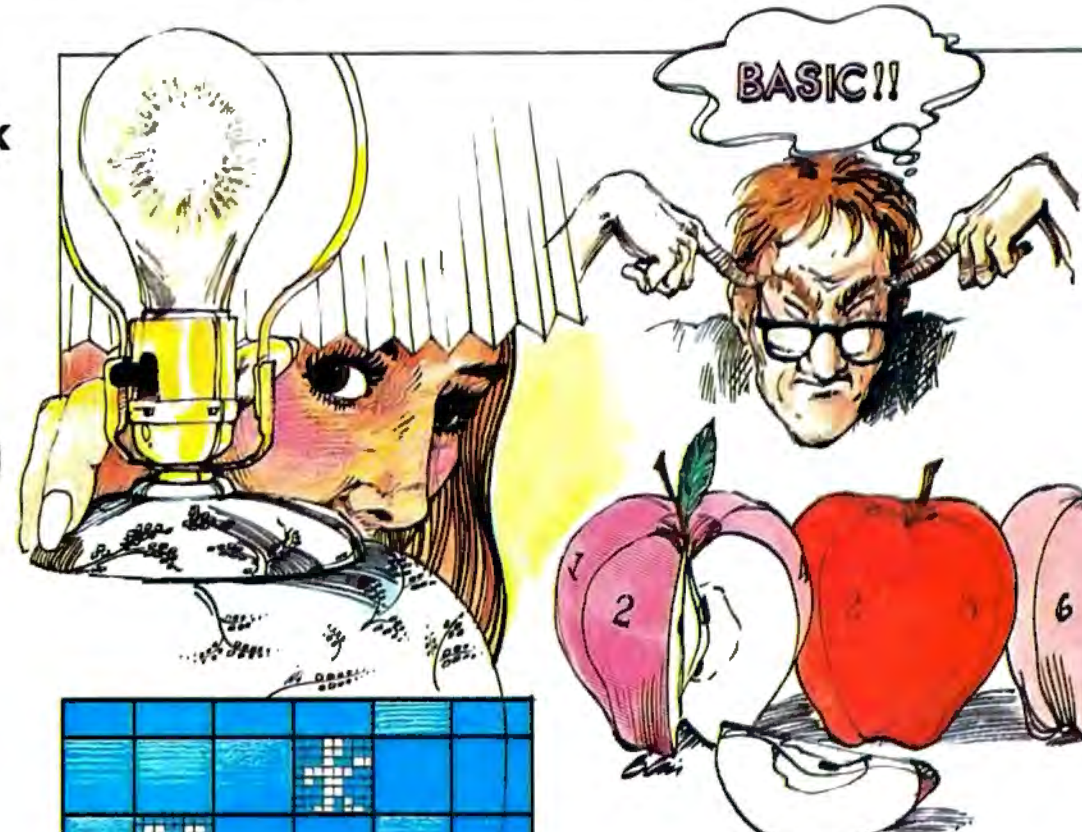
TWO NEW PERSONAL COMPUTERS FOR LESS THAN $200.00

COMPUTE!

$2.50
March, 1982
Issue 22
Vol. 4, No. 3
63379

The Journal For Progressive Computing™

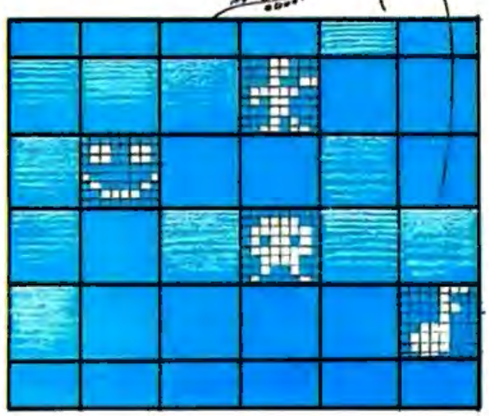


An Energy Workbook Program For Home Or Business

Two Programs To Help You Learn BASIC Programming

Organizing APPLE Data Storage

A Game Of Concentration For ATARI



Disk Drive Check-Out For PET/CBM

Starfight 3 For VIC-20

Reviews: Ricochet, Votrax Type 'n Talk, Atari's Microsoft BASIC And More...

READY TO USE BASIC PROGRAMS For Home, Education and Recreation

0 3
74470 63379

IF YOU'RE WAITING FOR THE PRICE OF WORD PROCESSORS TO FALL WITHIN REASON,

Everyone expected it would happen sooner or later...with WordPro PLUS™ it already has! Now all the marvelous benefits of expensive and advanced word processing systems are available on Commodore computers, America's largest selling computer line. WordPro PLUS, when combined with the new 80 column CBM 8032, creates a word processing system comparable to virtually any other top quality word processor available—but at savings of thousands of dollars!

New, low cost computer technology is now available at a fraction of what you would expect to pay. This technology allowed Commodore to introduce the new and revolutionary CBM 8032 Computer.

WordPro PLUS turns this new CBM 8032 Computer into a sophisticated, time saving word processing tool. With WordPro PLUS, documents are displayed on the computer's screen. Editing and last minute revisions are simple and easy. No more lengthy re-typing sessions. Letters and documents are easily re-called from memory storage for editing or printing with final drafts printed perfectly at over five hundred words per minute!

Our nationwide team of professional dealers will show you how your office will benefit by using WordPro PLUS. At a price far less than you realize.

Invest in your office's future... Invest in WordPro PLUS... Call us today for the name of the WordPro PLUS dealer nearest you.

Professional Software Inc.
166 Crescent Road
Needham, MA 02194
(617) 444-5224
TELEX: 95 1579

TM WordPro is a Registered Trademark of Professional Software, Inc. WordPro was written by Steve Punter.

CMS Softwa
General Accou

For the Commodore 8000

Now there is an interactive General Accounting System designed especially for the first time user. The user is led through each function by highlighted prompts explaining the required input at each point. And all input requests are prompted with complete verification. Plus, a step-by-step instruction manual guides the user through the program. The user prompts and the detailed manual make it virtually impossible for the first time user to get lost or to accidentally crash.

General Ledger

Cash Disbursements and Cash Receipts Journal as well as a General Journal. Maintains account balances for Present Month, Quarter to Date and Year to Date. Accepts postings from Accounts Receivable, Accounts Payable, Payroll and other programs. Also provides user customized financial statements. Up to 1000 accounts and up to 99 departments.

re Systems
nting System II

Series Computer System

Accounts Receivable

Invoices and Monthly Statements as well as Credit and Debit Memos. Accomodates full or partial payments. User sets finance charge rate and period. User also sets invoice aging reports and aging breaks. Invoices may be distributed among nine different General Ledger accounts with automatic updating to the General Ledger. During invoice data entry the invoice is displayed on the screen and typed exactly as if in a typewriter. File capacity allows 1800 customers and invoices. Five hundred customers allows room for 1300 invoices.

Accounts Payable

Accounts Payable checks with full voucher detail. Detailed check register. Automatic application of Credit Memos. User sets invoice aging reports and aging breaks. Invoices may be distributed among nine different General Ledger accounts with automatic updating to the General Ledger. File capacity allows 2500 vendors and invoices.

Payroll

Payroll checks with full deductions and pay detail. Pays regular, overtime, holiday and piece work hours. Accommodates Weekly, Bi-Weekly, Semi-Monthly and Monthly employees. Up to eight miscellaneous deductions or payments per employee. Prints Payroll Journal, Payroll Check Register and an Absentee Report as well as 941 information and W2 forms. Maintains Monthly, Quarterly and Yearly totals for up to 350 employees. Automatic General Ledger updating.

See your nearest Commodore dealer for a demonstration.

CMS Software Systems

2204 Camp David Mesquite, TX. 75149 214 285-3581

TASC™. The Applesoft* Compiler. It turns your Apple into a power tool.

Step up to speed. TASC, the Applesoft Compiler, converts a standard Applesoft BASIC program into super-fast machine code. By increasing program execution speed up to 20 times, Microsoft gives you a power tool for Applesoft BASIC programming.

Highest capacity available. TASC will compile and run larger programs than any other Applesoft Compiler. As a disk-based system, it doesn't require the simultaneous presence of compiler and program in memory. The memory you save allows you to compile significantly bigger programs.

Power without bulk. Code expansion of up to 100% severely restricts other compilers. TASC's special code compression schemes typically limit code expansion to only 25%. You'll really appreciate that with complex programs or programs that utilize Apple's hi-res graphic pages.

More BASIC power. TASC's powerful new commands increase Applesoft BASIC programming capability. Chain with COMMON allows compiled programs to share variables, so a main menu supports several programs in a single runtime environment.

TASC's True Integer Arithmetic and Integer FOR...NEXT capabilities maximize the execution speed of compiled programs.

TASC's near total compatibility with Applesoft speeds compilation of existing programs with little or no modification.

What about mistakes? You perfect your programs interactively with Applesoft. If something does slip by, TASC recovers from errors discovered in compilation and traps all runtime errors. It even permits graceful interruptions during compilation.

See for yourself. Ask for a demonstration of TASC at your Microsoft dealer. Discover the software package that turns your Apple into a power tool.

MICROSOFT
CONSUMER PRODUCTS

A Division of Microsoft Inc.
10700 Northup Way • Bellevue, WA 98004

*Applesoft is a trademark of Apple Computer, Inc.

Table of Contents

March, 1982 Vol. 4 No. 3

Features

The Winter Of Our Content: A Report On The January Consumer Electronics Show .... David Thornburg, 26
Twenty Questions Revisited .... Elizabeth Deal, 38
Energy Workbook .... David E. Pitts, 44
Two Short Programs of CAI For Teaching BASIC ...... R. Hiatt, 56
Infinite Precision Multiply .... G. H. Watson, 62
More About Numbers .... Edmund Ricchezza, 65
Word Hunt .... Robert W. Baker, 68
Count The Hearts .... Christopher J. Flynn, 78

Education And Recreation

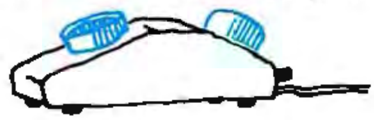
Family: A Simulation In Genetics .... Andy Gamble, 88
Large Alphabet For The VIC .... Doug Ferguson, 104
Concentration .... Charles Brannon, 106
Comment Your Catalog .... Richard Cornelius, 110
Starfight3 .... David R. Mizner, 112
Swirl And Scribble .... Matt Giwer, 115
WEBS .... Loran Gruman, 117

Reviews

Votrax Type 'N Talk: TNT .... Charles Brannon, 118
Olympia's ES 100 KRO Typewriter/Printer .... Richard Mansfield, 119
RPL: A FORTH Sequel? .... Jim Butterfield, 120
Ricochet .... Richard Mansfield, 121
Atari Microsoft BASIC (Part I) .... Jerry White, 122

Columns/Departments

The Editor's Notes .... Robert Lock, 6
Ask The Readers .... Robert Lock, Richard Mansfield, and Readers, 12
Computers And Society .... David Thornburg, 16
The Beginner's Page: How Computers Remember .... Richard Mansfield, 20
Basically Useful BASIC: Tabulation ...... Paul Lilly, 30
COMPUTE!'s Listing Conventions 43
Learning With Computers: Word Processing In The Classroom .... Glenn Kleiman and Mary Humphrey, 96
Friends Of The Turtle .... David Thornburg, 100
Telecommunications: MODEM Applications .... Michael Day, 124
Machine Language: Loops and Quality .... Jim Butterfield, 128
Insight: Atari .... Bill Wilkinson, 130



The Journal

Disk Checkout For 2040, 4040, and 8050 Disks .... Jim Butterfield, 139
Organizing Data Storage .... John Hudson, 141
Machine Language Sort Utility .... Ronald and Lynn Marcuse, 144
Dynamic Renumber .... R. D. Young, 152
Disk Data Structures: An Interactive Tutorial .... David Young, 156
Apple Addresses .... Bill Grimm, 163
More VIC Maps .... Jim Butterfield, 168
EPROM Reliability .... Michael Day, 176
Random Music Composition On The PET .... Alfred J. Bruey, 182
Ghost Programming .... Aric Wilmunder, 190
BASIC 4.0 To Upgrade Conversion Kit .... Elizabeth Deal, 194
Screen Save Routine .... Joseph Trem, 201
A Superboard II Monitor (Part I) ...... Frank Cohen, 203

New Products 209
CAPUTE! Corrections And Amplifications 222
Advertisers Index 224

Computer Guide To Articles And Programs By Page Number

Apple 6, 12, 16, 20, 26, 30, 38, 43, 44, 56, 62, 65, 68, 78, 96, 100, 110, 118, 119, 121, 124, 128, 141, 163, 176, 203
Atari 6, 12, 16, 20, 26, 30, 38, 43, 44, 56, 62, 65, 68, 78, 96, 100, 106, 115, 118, 119, 121, 122, 124, 128, 130, 144, 156, 176, 190, 201, 203
PET/CBM 6, 12, 16, 20, 26, 30, 38, 43, 44, 56, 62, 65, 68, 78, 88, 96, 100, 117, 118, 119, 120, 124, 128, 139, 152, 176, 182, 194, 203
VIC-20 6, 12, 16, 20, 26, 30, 38, 43, 44, 56, 62, 65, 68, 78, 96, 100, 104, 112, 118, 119, 124, 128, 176, 203
Other Computers 6, 12, 16, 20, 26, 30, 38, 43, 44, 56, 62, 65, 68, 78, 96, 100, 118, 119, 124, 128, 176, 203

COMPUTE! The Journal for Progressive Computing (USPS: 537250) is published 12 times each year by Small System Services, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 625 Fulton Street, Greensboro, NC 27403. Domestic Subscriptions: 12 issues, $20.00. Send subscription orders or change of address (P.O. form 3579) to Circulation Dept., **COMPUTE!** Magazine, 515 Abbott Drive, Broomall, PA 19008. Controlled circulation postage paid at Greensboro, NC 27403 and additional mailing offices. ISSN 0194-357X.

The Editor's notes...

Robert C. Lock
Publisher/Editor

COMPUTE!'s New Format

As we indicated last issue, the format of the magazine has been revised to allow us greater flexibility in article placement. In the front sections of the magazine you'll find general interest articles, including applications programs and games that have been developed to run on numerous computers. More advanced material, both applications and programming support, will be found in the *Journal* section of the magazine,

Each column, article, program, etc. is keyed by page number in the Table of Contents to relevant microcomputers.

On The Importance Of Feedback

On our direct mail response cards at the end of this issue, you'll find a special Editor's Feedback card. Take a look at our new format for the magazine, and *please* take a few minutes to fill out and return the card.

We've tried to accomplish the changes as a constructive enhancement, and rely on your collective opinion as well. Let us know what you think.

Two New Personal Computers For Less Than $200.00

The Winter Consumer Electronics show was impressive. Commodore was showing a startling array of VIC-20 software. Atari introduced some new twists as well. For full details on the show and the new computers, see David Thornburg's article in this issue.

Home Applications

Being firmly convinced that the myth of the utility of home computers is truly a myth, we've been excited by the range of useful software you've been sending in. With everything from our continuing series of Energy Conservation and Analysis programs by David Pitts, to Real Estate Investment Analysis and sophisticated financial modeling, we've been proving by example that home computers are being used for education and applications as well as recreation. Keep your efforts coming.

COMPUTE! Books

Our Book Division is now off and rolling, with the release (finally) of our first two titles, **COMPUTE!'s First Book of ATARI®** and **COMPUTE!'s First Book of PET™/CBM™**. We have three more books scheduled for release in the first and second quarters of 1982. Having learned our lesson the first time around, these won't be announced until scheduled at our printers, but we're sure you'll find them equally useful resources.

The West Coast Computer Faire

Once again we expect to see hundreds of you among the 30,000 to 40,000 attendees at the Sixth West Coast Computer Faire. Jim Warren's annual extravaganza is a looked-for meeting place for old and new friends. Stop by our booths, numbers 1543 and 1642, and say hello.

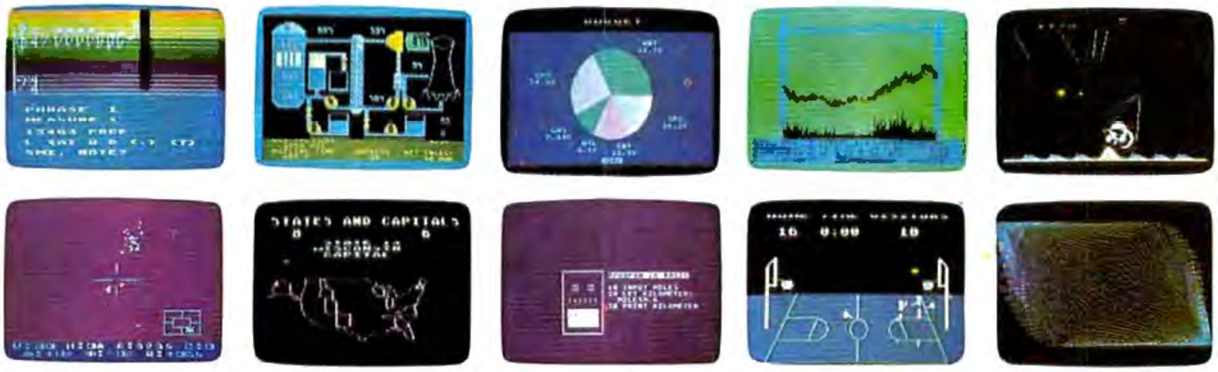
An Important Message To Our Retailers

As you've all noticed, we recently moved to a new billing system. By now, our scheduling and updating problems have been resolved. Your new invoices will be showing a message "Payments received after ___________ are not reflected in this invoice." Interim shipments of back issues or **COMPUTE! Books** will be billed as shipped, and *also* reflected on your monthly statement-invoice.

©

THE GRAPHIC DIFFERENCE

BETWEEN ATARI® COMPUTERS AND ALL OTHERS.



3.7 million reasons why the ATARI Home Computer is something to see.
The display screen used with our computers is composed of 192 horizontal lines, each containing 320 dots. Delivering color and luminosity instructions to each dot for a second requires 3.7 million cycles... a lot of work for the normal 6502 processor.

That's why the ATARI computer has equipped its 6502 with its own electronic assistant. It's called ANTIC, and it handles all the display work, leaving the 6502 free to handle the rest. What this means to you is uncompromisingly spectacular display capabilities without loss of computer power needed to carry out the demands of your program.

That's a quality you just don't find in ordinary personal computers. And it's one of the reasons some computer experts say that ATARI computers are so far ahead of their time.

There's more... which is what you'd expect from ATARI.

Language. The ATARI Personal Computer uses several programming languages to give the user maximum control of its extraordinary capabilities. PILOT, Microsoft BASIC,* and ATARI BASIC are understood and spoken by the ATARI computer. You'll also find our Assembler Editor cartridge indispensable for machine language programming.

Sound. An ATARI computer has four sound generators, or voices, activated by a separate microchip. This leaves the principal microprocessor chips free to perform other tasks. And you can take full advantage of this capability which is designed for easy programming.

Change. ATARI Home Computers have been designed to make change and expansion easy. The ATARI computer has a modular operating system* that can be easily replaced as new technology develops. If you need it, memory expansion requires no more than inserting additional RAM modules.* And the ATARI ROM cartridge system also makes it easy to change languages. In short, your ATARI computer won't be obsoleted by future developments... *beause it already incorporates the future.*

Sharing. To learn more about the amazing capabilities of ATARI computers, visit your local computer store for a demonstration. Or send for our Technical User's Notes, intended for the serious programmer. They are only $27 and contain a lot more information about our computers' special capabilities than most companies could tell. See your ATARI dealer, or send $30 ($27 plus $3 postage and handling), payable to ATARI, to Technical User's Notes, c/o ATARI Customer Service, 1340 Bordeaux Avenue, Sunnyvale, CA 94086.

ATARI®

*ATARI 800™ computer only.

A Warner Communications Company

Staff of COMPUTE!

Robert C. Lock, Publisher/Editor
Kathleen Martinek, Managing Editor
Richard Mansfield, Assistant Editor
Alice S. Wolfe, Publisher's Assistant
Georgia Papadopoulos, Art Director/ Production Manager
Harry Blair, Advertising Sales Manager
Terry Cash, Typesetting/Production Assistant
Margret Jackson, Typesetting
De Potter, Production Assistant
Dai Rees, Production Assistant
Kate Taylor, Production Assistant
Charles Brannon, Editorial Assistant
Sonja Whitesell, Dealer Coordinator
Bonnie Valentino, Accounting Coordinator
Fran Lyons, Circulation Coordinator

Associate Editors

Jim Butterfield, Toronto, Canada
Harvey Herman, Greensboro, NC

Contributing Editors

Marvin DeJong, Dept. of Mathematics -Physics, The School of the Ozarks Pt. Lookout, MO 65726
David Thronburg, P.O. Box 1317, Los Altos, CA 94022
Bill Wilkinson, Optimized Systems Software, 10379-C Lansdale, Ave. Cupertino, CA 95014
Gene Zumchak, 1700 Niagara St., Buffalo, NY 14207

Subscription Information

COMPUTE! Circulation Dept.
515 Abbott Drive
Broomall, PA 19008 USA

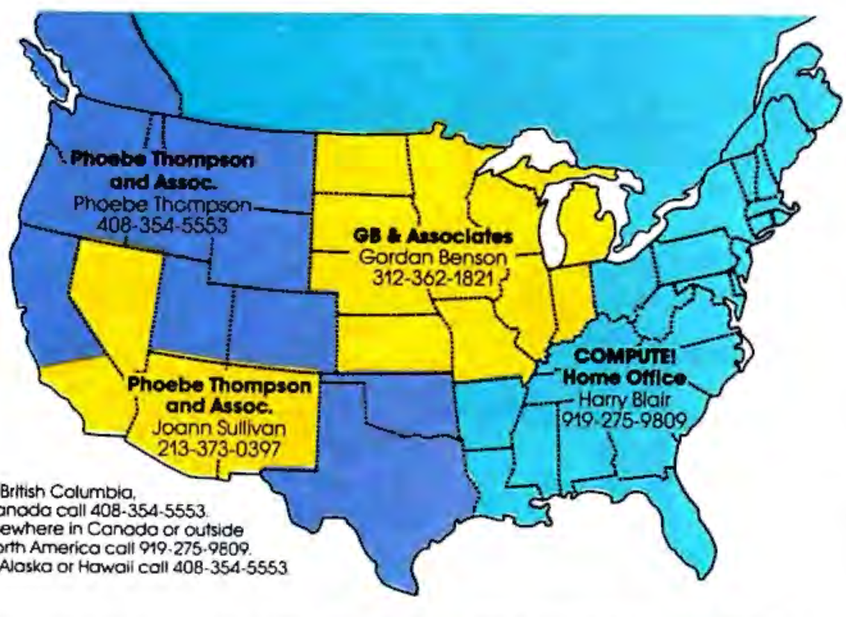
TOLL FREE
Subscription Order Line
800-345-8112
In PA 800-662-2444

COMPUTE! Subscription Rates (12 Issue Year):

US	(one yr.)	$20
	(two yrs.)	$36
	(three yrs.)	$54
Canada and Foreign Surface Mail		$25
Air		
Europe, US Possessions		$38
Middle East, Central America and North Africa		$48
South America, South Africa, Far East and Australia		$88

Canadian Retail Dealers should contact:
Micron Distributing
409 Queen Street West
Toronto, Ontario M5V 2A5
416-361-0690

Advertising Sales



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

Phoebe Thompson and Associates	Phoebe Thompson and Associates	GB & Associates	COMPUTE! Home Office
101 Church Street, Suite 13	P.O. Box 700-462	P.O. Box 335	625 Fulton Street
Los Gatos, CA 95030	Redondo Beach, CA 90277	Libertyville, IL 60048	Greensboro, NC 27403
PHOEBE THOMPSON	JOANN SULLIVAN	GORDON BENSON	HARRY BLAIR

Address all advertising materials to: Alice S. Wolfe/Advertising Production Coordinator, **COMPUTE!** Magazine, 625 Fulton Street, Greensboro, NC 27403 USA.

Small System Services, Inc. publishes:

COMPUTE!
The Journal For Progressive Computing

COMPUTE! Books

Corporate office:
625 Fulton Street,
Greensboro, NC 27403 USA

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

Telephone: 919-275-9809

Robert C. Lock, President
W. Jerry Day, Vice-President, Comptroller
Kathleen Martinek, Assistant To The President
Ellen Day, Accounting Manager
Carol Holmquist Lock, Research Assistant

Coming In The April Issue Of COMPUTE!:

Tracking Down Memory Bugs

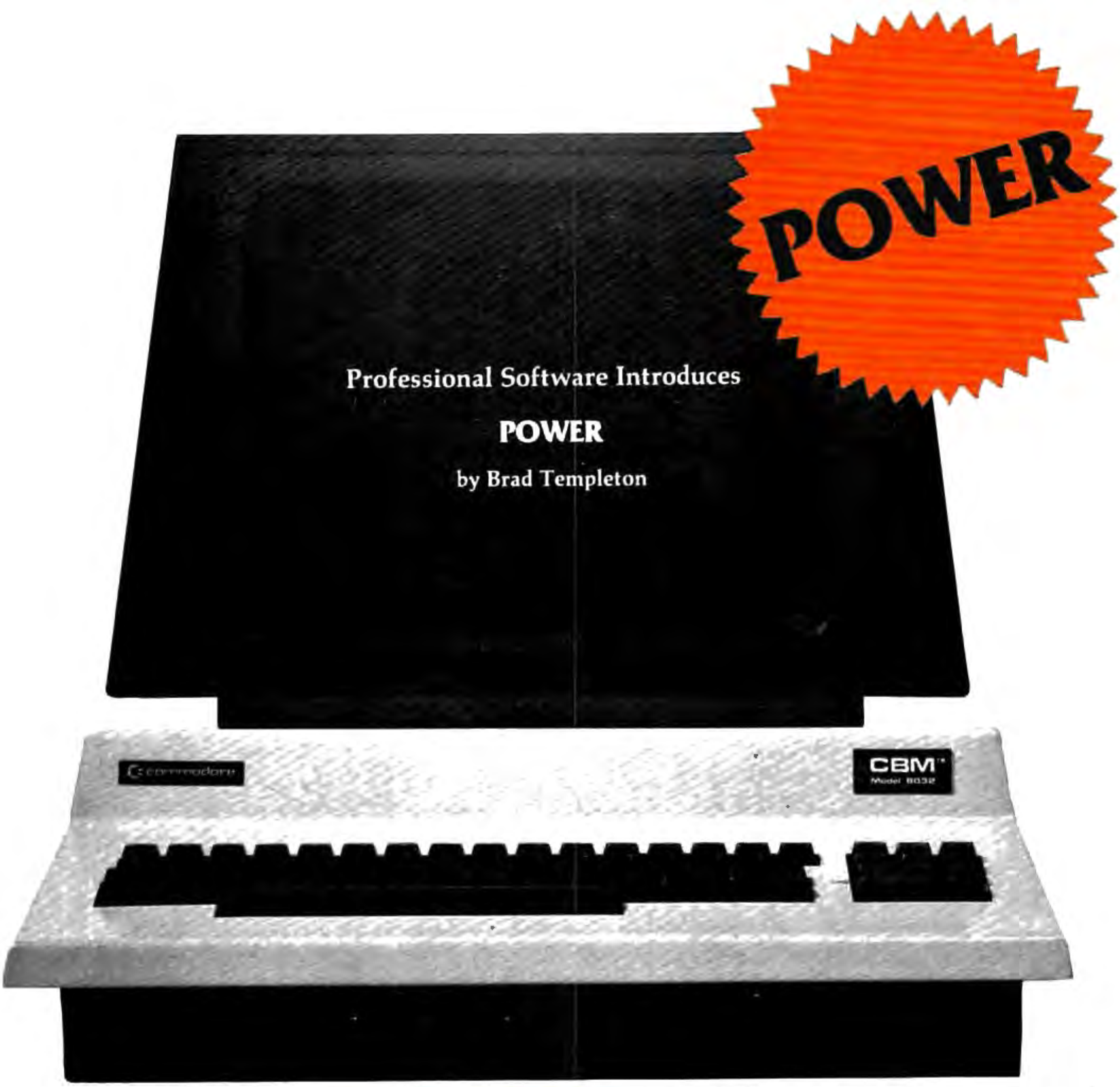
Shooting Stars Game

User Defined Functions: Defined for PET/CBM

Using Textplot For Animated Atari Games

A VIC-20 Terminal Program

Customizing Apple's Copy Program



ADD POWER™ TO YOUR COMMODORE COMPUTER

$89.95

POWER produces a dramatic improvement in the ease of editing BASIC on Commodore's computers. POWER is a programmer's utility package (in a 4K ROM) that contains a series of new commands and utilities which are added to the Screen Editor and the BASIC Interpreter. Designed for the CBM BASIC user, POWER contains special editing, programming, and software debugging tools not found in any other microcomputer BASIC. POWER is easy to use and is sold complete with a full operator's manual written by Jim Butterfield.

POWER's special keyboard 'instant action' features and additional commands make up for, and go beyond the limitations of CBM BASIC. The added features include auto line numbering, tracing, single stepping through programs, line renumbering, and definition of keys as BASIC keywords. POWER even includes new "stick-on" keycap labels. The cursor movement keys are enhanced by the addition of auto-repeat and text searching functions are added to help ease program modification. Cursor UP and cursor DOWN produce **previous** and next lines of source code. COMPLETE BASIC program listings in memory can be displayed on the screen and scrolled in either direction. POWER is a must for every serious CBM user.

Call us today, for the name of the Professional Software dealer nearest you.

Professional Software Inc.
166 Crescent Road
Needham, MA 02194
Tel: (617) 444-5224 Telex #951579

TM POWER is a Registered Trademark of Professional Software, Inc.

Ask The Readers

Robert Lock, Richard Mansfield, And Readers

If you have any questions (or answers to the questions printed below) please write to: Ask The Readers, **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403.

Answers

On page 10 of **COMPUTE!** *#19, a reader asked about, and you commented on, COM as an Atari BASIC keyword.*

I very recently bought an Atari and as yet have not delved extensively into the programming aspects of it since most of the time it has Missile Command in the left slot, but, in other computer systems I have used with either interpreter or compiler BASIC, COM or COMMON defines common variables, either string or numeric, for programs which are to be chained instead of POKEing the information into protected memory by one program, then PEEKing it out by a subsequent one. This is usually accomplished under DOS control.

A question that I have, and perhaps another reader can answer, is how I could build an interface or two to use the Radio Shack (Tandon 40 track) drives and Centronics 737 parallel printer I have for my TRS-80 Model I with the Atari serial port. I realize this identifies me as a "Hardware Nut," but for personal satisfaction, I would rather build, than buy, an interface. William E. Allen

---

Our thanks to the others who sent in responses to this question.

I am writing in response to columnist Zumchak's comments in the "Ask The Readers" section of **COMPUTE!** *#16.*

The question of a next generation 6502 microprocessor is certainly not a dead issue. However, this is a market-driven decision. Synertek would readily undertake the development and marketing of a sequel to the 6502 if there were reasonable expectations that this would be justified by adequate business in the new part.

As to the SYM board, the SYM-2 version is laid out to accept the 6502, 6800, or 6809. There is also a plug-in module for the SYM-1 to adapt it to those processors. These new products were introduced to widen the market for the SYM, not as a move away from the 6502.

Synertek is continuing to support the 6502, as we have in the past. New peripheral products are being designed to work with the 6502. We believe that the 6502 is one of the best 8-bit processors available. Although many applications use 1MHz parts, the 6502 is available in versions that run with clocks of 2, 3, and 4 MHz.

Dr. Michael Smolin
Director, Strategic Planning
Synertek Inc.

In his excellent article Machine Language: What's Your Sign? *(***COMPUTE!** *#17), Jim Butterfield discusses how the Overflow (V) flag will be set if the addition or subtraction of two signed numbers causes a "sign switch."*

A rarely publicized fact is that a sign switch, or overflow, can only *occur if you are adding two numbers that have the* same *sign (both positive or both negative) or subtracting two numbers that have* opposite *signs. Moreover, because a sign switch indicates that the result exceeds the legal limits of a signed number, -128 or +127, V = 1* always *constitutes an error condition...*

One further point of interest is that CMP, unlike SBC, does not affect the contents of the Accumulator. This allows you to compare N1 to some other number (N3) immediately, without another LDA N1 instruction.

Leo J. Scanlon

For several years I have been dealing with the CRASH of the INPUT statement on our 2001 and 4016 COMMODORE Computers when the RETURN key is depressed with no input! I have read of several fairly short routines that overcome this problem and have developed my own favorite, as have most users (Our students develop software for the Elementary School – kids, and their teachers, will *crash programs).*

Anyway – this evening, quite by accident, I stumbled upon a quick way to protect the RETURN key on INPUT! At least I have never heard of it. [Lines 70, 80, 160, 170, and 180 will do.] *If you type in the enclosed program and RUN same, you will notice that the INPUT at line 70 (numeric) and line 160 (string) are protected under input/return (and, of course RUN/STOP)!!*

When you look at the listing, I'm sure your reaction will be the same as mine was when I looked at the listing and said to myself and all within earshot, "What's going on here? I tried this years ago! It didn't work then, so why is it WORKING now???."

The key lies in the length of the input prompt – exactly 38 characters. *This configuration puts the INPUT question mark at the end of the input line and the cursor all by itself on the left of the next line. Bingo – RETURN = null. The conditional picks up the pieces and we're in business.*

John Taylor

To answer John Fry's question [in **COMPUTE!** *#18]:*


NASIR GEBELLI PRESENTS:
PATHFINDER
PATHFINDER
Never before released, a super arcade-adventure game for Atari . . . alone in a vast transparent maze inhabited by radioactive monsters and nuclear waste, you must blast the waste and free the maze of mutant monsters. By Solitaire Group, requires 24K Atari with disk drive.
MATCH RACER
MATCH RACER
Brand new for the Atari, this one or two player game tests your driving skills on a race track filled with treacherous obstacles. See how far you can get before your luck and skill run out! By Solitaire Group, requires 16K Atari with disk drive.
ANDROMEDA
ANDROMEDA
Introducing a spectacular, original arcade-adventure for the Atari . . . a battle in inner space . . . you must invade a living organism and destroy its tissues before its antibodies destroy you. By Solitaire Group, requires 16K Atari with disk drive.
HOLLYWOOD
GEBELLI SOFTWARE INC
Gebelli Software Inc.
1771 Tribute Rd. Suite A, Sacramento, CA
1981 Gebelli Software, Inc. Atari is a registered trademark of Atari, Inc.

sequential fata files: you have two problems.

Statement in line 40 is wrong. It should read 40 DISK CLOSE,6 not *40 DISK CLOSE,6,A$.*

Also, don't forget to run change program to shift BASIC work space to allow for your 1 buffer (step 3 on page 24 in my system's manual).

OSI puts the buffers used for device 6 and 7 before the BASIC program. If you don't run change to tell BASIC to allow for these, it will start in this area for program storage. So your first print #6, A starts writing over your program and, thus, garbage on LIST.

Alex J. Kowalski, Jr.

I am writing with a few odds and ends, corrections/modifications to a couple of programs, and a question that "Ask the Readers" may be able to answer.

To begin with, users of "Keyword" as revised in **COMPUTE!** *#17 may have advised you by now that this very nice utility crashes when used with Upgrade (3.0 BASIC) ROMs. I found that the following line fixed the problem:*

```
215 POKE BASE+8,46:POKE BASE+12,230
```

While on the subject of "Keyword," it may be worth mentioning to all that this and any program that alters the interrupt vector must be turned off *prior to initiating tape SAVEs or LOADs. I also found that "Keyword" should be turned OFF prior to running a BASIC program to avoid printing a collection of keywords where graphics or lower case should be printed. Perhaps someone can explain this problem.*

I am using an Original ROM PET with Upgrade ROMs installed. The Upgrade ROM Monitor, available all the time, is great. I have found at least one anomaly with this Monitor, however: it refuses to accept program names that are more than 15 characters long. Perhaps occasionally, an "End-of-Tape" header is recorded at the end of a machine language program SAVE using the Monitor. Are there any explanations for this out there?

Lastly, I finally got to try the program "Basic Math for Fun and Profit" from **COMPUTE!** *#9. It's a great program! I did make one significant change to it, however:*

```
922 IFZ$(T)=CHR$(13)THENZ$(T)=STR$(0):I=9
923 PRINTZ$(T);" ";
924 Y$=Z$(9)+Z$(8)+Z$(7)+Z$(6)+Z$(5)+Z$(4)+
    Z$(3)+Z$(2)+Z$(1):W=VAL(Y$)
925 NEXT:GOTO357
```

(Changes at lines 922 and 925.)

R. D. Young

I would like to share some lessons learned about the VIC 20. In obtaining a TV for use with the VIC 20 I learned several items I thought would be of interest to all VIC 20 owners. Zenith TVs models L, M, N, can have a jump in the picture because of the difference in the sync rate between the VIC 20 and the TV's. This jump can be taken out by your service man by having him connect the jumper (sometimes called non standard vertical or cable standard jumper) on the vertical output board. If your TV has a crystal controlled tuner and there is no fine tuning, you can adjust your modulator with the adjustment screw closest to the input end so you can get color on both channels 3 and 4 on the TV.

If you are having wavy lines on your TV you could be picking up RF interference signals on the wires from the TV computer switch box. If your TV has cable input capability, use of coaxial cable from the modulator to the input for cable can eliminate this interference. If your picture is not the right spot on the TV POKE 1 through 14 into 36864 will move it right and left, and 0 through 125 into 36865 will move it up and down.

Want to use your "f" keys? You can if you GET A$ and use the ASC(A$) to create a value 133 for f1 to 140 for f8. This CHR$ value can then be used in the program. Jim Turrentine

---

Questions

I want to thank you and Dub Scroggin for the fine game program in the January 82 issue for the VIC 20. Instead of using the keyboard, I made a change in "ZAP!!" to use a joystick and it works great.

I have a problem with my recorder. I have to turn it upside down to load a program. I have taken the back off, but can't find anything loose. If I try a load it finds a program then doesn't load and prints "out of memory" or loads a scrambled up program. Any ideas? Don Dudley

---

The school at which I teach has just purchased the VIC 20. Software is scarce at the moment. Could you suggest or indicate programs of an educational nature that I might use in an elementary school setting?

I'm really keen on getting a good start with the VIC 20 in my school. Michael Moher

---

Hallo there in USA!

I've opened an Atari User Club here in West Germany. I would appreciate it if you could send me the addresses of Atari Programmers. Here in Germany you can only buy some games like Star Raiders and some more, but they are few. My Club wants to make it possible for the members to buy programs from the USA.

Also, could you send me the addresses of Atari User Clubs there in the USA?

I hope you can help me.
Knut Hermann

Merscheider Strasse 130
5650 Solingen-Merscheid
Germany

ATARI OWNERS, WE'VE OPENED UP A WHOLE NEW WORLD FOR YOU.

At SSI, we established our reputation in quality computer games by producing some of the finest strategy simulations for the Apple® and TRS-80.®

Well, Atari owners, this is your lucky day. Because we've just introduced our first game for the Atari® 800 computer: THE SHATTERED ALLIANCE.™

With it, we've opened up for you SSI's world of fun and excitement in a big way — by literally creating an entire planet called *Osgorth* to serve as your personal playground!

OSGORTH! It is a wonder world of fantasy and adventure populated with creatures as varied as they are enchanted: dwarves, unicorns, Valkyries, centaurs, lizardmen and zorgs (whatever they are!)... just to name a few. They are preassembled into different armies, categorized into five classes of infantry and three of cavalry. They are meticulously rated for armor, speed, strength, and morale. Their weapons may be the bow, javelin, spears, pikes, cavalry lance, swords, axes, clubs, and rocks. They can even call on magic spells to enhance their speed, attacking or defensive power, or courage.

THE SHATTERED ALLIANCE is much more than a sophisticated strategy game. It is a rich tale of magic, rivalry among realms, and war among exotic beings. Best of all, these fantasy soldiers are *yours* to command as you roam Osgorth in search of glory and conquest.

As an added bonus for the Ancient-battles enthusiasts, we have included scenarios such as *Carthage vs. Alexander the Great.*

THE COMPUTER PROGRAM is a veritable wizard. With color Hi-Res graphics, it paints a beautiful Osgorthian landscape that is your battleground. It keeps track of all details so you can concentrate solely on strategy planning while its *Rapidfire* movement system makes for smooth and fast action. It even provides a powerful and intelligent computer opponent for solitaire play.

All you need to play this game is a 48K Atari 800 with mini floppy disc drive. For only $39.95, you can own this fantastic piece of real estate. So hurry on down to your local game/computer store and get THE SHATTERED ALLIANCE today!

To order with your VISA or MC, call toll free **800-227-1617, ext. 335** (800-772-3545, ext. 335 in California). To order by mail, send your check to Strategic Simulations Inc, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043. All our games carry a 14-day money back guarantee.

Also available for 48K Apple II with ROM for $59.95.

SSI

Atari is a registered trademark of Atari, Inc.

STRATEGIC SIMULATIONS INC

As part of our demanding standards of excellence, we use maxell floppy discs.

WE'RE LOOKING FOR A FEW GOOD DESIGNERS. We're always on the lookout for high-quality games from independent designers. If you have game programs that you'd like SSI to publish, give us a call at (415) 964-1353 or write to SSI, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043.

Apple is a registered trademark of Apple Computer Inc.

TRS-80 is a registered trademark of the Tandy Corporation.

Computers And Society

David D. Thornburg
Innovision
Los Altos, CA 94022

Artists have always been among the first to explore new technologies. Many times this exploration begins before the technology leaves the laboratory. For example, long before the laser found its way into supermarket check stand scanners, it was being used as an expressive medium by artists like Robert Whitman who received major technical assistance from Bell Laboratories and the optical scientist Eric Rawson.

The collaboration between artist and technologist requires extraordinary sensitivity on the part of both people. Historically, this collaboration has also required that the artist have access to a large university or industrial research laboratory where the needed technical resources could be found.

The innovative use of movie cameras in the 1940's benefited as much from technological collaboration as does the use of computers by artists today. Whenever a new technology offers an expressive medium to the artist, some technologists are more than happy to lend their assistance when needed. But one must wonder if the need for collaboration aids or inhibits the flow of creative ideas. Some artists are happy to work with others, and some are loners – preferring the solitude of an isolated studio to the constant interchange of ideas with co-workers.

One artist who has spent many years making innovative use of technology is John Whitney. Over the span of several decades, Whitney evolved the idea that the dynamic visual media (such as motion pictures, videodisks, and real-time computer animation) open the door to the creation of visual harmonies which are the counterparts of musical harmonies. His book, *Digital Harmony: On the Complementarity of Music and Visual Art* (Byte Books, 1980) expresses this philosophy in great detail. In it, Whitney traces his own development as an artist involved with technology.

In the late 1930's, he saw the motion picture camera as a tool which would allow him to create visual images which somehow "looked" the way music sounded. During his early experiments he learned something very important. To suggest that a motion picture camera can contribute to a new visual art makes about as much sense as suggesting that a tape recorder can contribute to the art of music.

Music Into Visual Symmetries

In the computer, on the other hand, Whitney saw a new medium – a medium of creative expression in which his quest for the dynamic display of visual harmonic symmetries could be satisfied.

...we are on the threshold of an era when isolated artists will ... be able to use the computer with as much ease as... pastels, oils, or clay.

Thus began a period in which his images were created first on a cathode ray tube and then captured on film for others to see. This work benefited from collaboration with technologists. For example, Larry Cuba assisted in the making of Whitney's film *Arabesque*. (Cuba's software talents in computer animation have probably been seen by many **COMPUTE!** readers – he created the Death Star animation sequence in *Star Wars*.)

Arabesque is occasionally shown on public television, and is a film worth seeing. Each time I have seen it I am once again impressed with its fluidity of geometric form. Not surprisingly, Whitney devotes a chapter of his book to this film.

Books of this sort are inspiring to artists and non-artists alike. It is exciting to see the art which springs out of centers of excellence, such as universities.

But it is even more exciting to realize that we are on the threshold of an era when isolated artists will themselves be able to use the computer with as much ease and familiarity as they presently use pastels, oils, or clay. While the early phases of computer-based art required a high level of technical and financial support, the implementation of easy-to-use graphics environments on low-cost computers (e.g., Atari PILOT on the Atari 400 or 800, TI LOGO on the 99/4A, and Apple SuperPILOT or LOGO on the Apple II) places this medium within the comprehension and budget of many artists.

This adoption of the computer as a new medium will not occur overnight, however. Most of

SOFTWARE FOR
ATARI™*
400/800

MarTesoro

The lure of sunken treasure has been an intriguing adventure for centuries. Now you can embark on your own expedition for fame and fortune.

With your initial financing you obtain necessary equipment and boat to explore off the coral reefs and around the shores of an island to accumulate treasures, some are worthless, others are worth a small fortune.

Joystick Required	With Graphics
24k Tape $16.95	32k Disk $21.95

Maze of Death

As a Freelance Agent, you try to stop the Cosmitron Device from exploding and destroying the city. The sinister scientist has allowed the device to become critical by removing the radiation control cubes. Your assignment is to find and replace the control cubes before it is too late. Five levels of play in Real Time.

Joystick Required	All Color Graphics
16k Tape $12.95	24k Disk $17.95

SYNCRO INC.
SOFTWARE DIVISION
31332 Via Colinas, Suite 107
Westlake Village, Calif. 91361
(213) 889-9508

ORDER FROM YOUR FAVORITE DEALER OR YOU MAY ORDER DIRECT FROM SYNCRO, INC.
ADD $1.50 FOR SHIPPING AND HANDLING — CALIF. RESIDENTS ADD SALES TAX.

the serious artists I know have two concerns about computers. First, for many, the cost of a $3,000 system is prohibitive. Second, there seems to be no easy way for these people to bridge the software barrier. Artists who really want to use computers as more than playthings know that, sooner or later, they will have to become programmers. How sad it is to see these people attending evening courses at the local community college learning BASIC or (Heaven forbid!) COBOL, in the hopes that by learning how to write bubble sort routines, the computer will become less mysterious.

Most intoductory programming courses don't satisfy any of these students' real needs. The fact that at least *some* artists (Whitney, for example) have been able to bridge this gap is heartening news, but this is likely to be cold comfort to those art/computer students who think a string variable refers to the color of a length of twine.

Sooner or later, artists in great numbers will start using computers on their own. Whitney's book shows us the promise of this tool and he encourages others to follow their own path with this technology.

All we need now are more practitioners. ©

JINSAM

Data Manager selected by NASA, Kennedy Space Center

SAVE TIME! SAVE MONEY!
JINSAM saves taxpayers 100's of thousands of $

OFFICIALLY APPROVED BY COMMODORE

"Much more powerful than you can imagine!"
Robert Baker
Kilobaud Microcomputing

JINSAM is an integrated system. It makes it easy to use your information to its fullest. No more will hundreds of valuable hours be spent searching or analyzing needed information nor re-entering information for various reports.

JINSAM transforms your desk-top computer into the "state of the art" data processing machine with features and accessories found nowhere, even at 10 times the price. NASA, Kennedy Space Center selected JINSAM 8.0 and saved approximately $95,000 over other software/hardware costs. Riley County, Kansas also selected JINSAM 8.0 and saved approximately $90,000 over other software/hardware costs.

JINSAM is designed for you. It is forgiving. It has help commands for every option, available at the touch of a button. The amount of information you store, its structure and/or your hardware can change but your data won't have to be re-entered. Recovery utilites are included even for catastrophes, security passwords are built in for privacy, simple editing and entry includes auto recall, and deleting records is easy and the space is reclaimed. JINSAM includes TWO FREE accessories for reports and labels. You have unlimited report formats with summing and lined up decimals and the label printer prints up to 5 across – any size and even allows single envelopes or index cards.

JINSAM 1.0 allows fast and easy file handling, manipulation and report generation for any CBM computer with CBM 2040 disk drive. It features a menu for ease, has encrypted passwords, 3 deep sorts, .5 to 3 second recall.

JINSAM 4.0 for CBM 4000 series adds JINSORT, user accessible machine sort of 1000 records in 15 seconds; compaction/expansion of information, automatic list maintainance, unlimited number of fields, unlimited record length and much more.

JINSAM 8.0 for CBM 8000 series has all 4.0 features plus unlimited sort, horizontal format, and search by key or record number.

JINSAM 8.2, NEW FOR '82 expands 8.0 capabilities by adding information search by word, key or record number and machine language print, format and manipulation routines.

- ★ CUSTOM DATA FILES
- ★ CUSTOM REPORTS/LABELS
- ★ KEYED RANDOM ACCESS
- ★ FAST/EASY/MENU DRIVEN
- ★ MULTIPLE SEARCH KEYS
- ★ PRIVACY ACCESS CODES
- ★ WILD CARD SEARCH

JINSAM EXECUTIVE version (soon to be released) is our most powerful professional system for the CBM 8000 and 9000 series. Executive will have 8.2 extended features plus allow multiple users with in-use lockout protection, executive command files, automatic math relations, join, merge or link files, greatly increased record capacity and machine information search by word, as well as by key or record number and many, many more features.

There are currently 7 more interfacing modules – and more under development, including independent interfaces between JINSAM and business packages for your G/L, A/R, A/P needs. We announce the availability of modules and enhancements in JINSAM's quarterly newsletter.

WORDPROPACK - Intelligent interface for WordPro 3, 3+, 4, 4+, creates lists of information from JINSAM files. It allows up to 10 conditions based on each item of information. Produce individualized letters, report cards, special reports, checks, invoices, etc.

MULTI-LABEL - Prints multiple labels per record with up to 2 lines for messages and consecutive numbering. Produce inventory, caution labels, bulk mail labels, etc.

MATHPACK - global calculator/editor +, −, *, /, by another field or constant; null (remove contents) of a field or replace contents of a field with any word, number or phrase. Sum multiple fields in each record or running sum of single field in all records. Extract information or effect permanent change. Replace in the same field or place in a waiting field.

DESCRIPTIVE STATPACK - Determine MEAN, MEDIAN, MODE, STANDARD DEVIATION, VARIANCE, RANGE. Generate HISTOGRAMS from 1 to 25 steps, and produce Z-SCORE reports.

ADVANCED STATPACK - (You must also acquire DESCRIPTIVE STATPACK) Generate one, two or three way CROSSTABS (number of occurance) CHI SQUARE, LINEAR REGRESSION, with graphic representation and prediction, LINEAR CORRELATION and SIMPLE ANALYSIS OF VARIANCE.

CALCPACK - 2 way interface to VisiCalc or any user program. It lets you use VisiCalc for complex manipulation, editing, placing results in JINSAM for sorting, storing or moving data to WordPro as well as giving the ability for exchange with your own applications.

INTERAC - Interface which can read VisiCalc files, WordPro files and almost any sequential files to build JINSAM databases automatically. For example: You could "download" information on holerith cards to sequential files and INTERAC would place them into JINSAM files.

All accessories are accessed thru the JINSAM menu and require security password to gain entrance.

JINSAM gives you FREEDOM OF CHOICE. Start with JINSAM 1.0 and upgrade hardware and data at any time. Choose from accessories at any time. The JINSAM Newsletter brings the latest updates, user input and uses and keeps an eye on the future.

JINSAM stands alone by placing "a lot of potential computing power in one integrated program package" (Fred Klein, Ferson div. of Bausch & Lomb). "The JINSAM package is justification for buying a system no matter what the hardware, be it Vector or Commodore or whatever the system" (Larry Colvin, Micro Computer Systems). It is a "grandfather" in this young field. JINSAM EXECUTIVE will be the third generation in development. All JINSAM systems are sophisticated and flexible yet easy to use. JINSAM is saving its users valuable time and money in government, educational and research institutions, business and industry worldwide.

JINSAM is a Commodore approved product. See your local dealer for a demonstration.

JINSAM Data Manager

- .... Additional Information and nearest dealer
- .... Newsletter Subscription ($5 US/$8 Worldwide)
- .... JINSAM Demo Disk ($15, plus shipping & tax)
- .... User's Guide ($40.00, plus shipping & tax)

Please send to:

Name __________

Position __________

Company __________

Address __________

City, State, Zip __________

Phone (   ) __________

Computer, Disk __________

Application __________

JINI MICRO-SYSTEMS, Inc.
Box 274C • Kingsbridge Station
Riverdale, New York 10463
(212) 796-6200

Dealer Inquiry Welcome

The Beginner's Page:

How Computers Remember

Richard Mansfield
Assistant Editor

Computers fascinate us for many reasons. We have only recently built this machine and it is fast and complex, but the same could be said of a jet airplane. One reason for our fascination might be that, after years of passively watching TV, we can finally take control of what appears on the screen. Beyond this, perhaps, is the fact that the computer is the thing we've built which comes closest to being a copy of our own mind.

There are significant differences, though, between the way we think and the way the computer thinks. For complex, abstract thinking, a child can easily outdo the finest computer available today. On the other hand, the computer is far faster than any human. For one thing, the computer was *designed* to be efficient when dealing with numbers. The engineers realized from the start that the computer would be much better off if it avoided using our "decimal" system.

Let's look into some reasons why thinking in decimal is less effective than *binary*, the computer's way. This will help us understand how a machine can "remember."

A computer does two major things: it solves problems (computes) and it *remembers*. To add 2+2, the computer first remembers (stores) your instructions which contain the problem. 10 X = 2+2. When you type RUN, it looks for the lowest line number in its memory and solves whatever it finds. Then it remembers the answer (elsewhere in its RAM, its temporary memory cells). If there is another line number (20 PRINT X) it will put the number 4 on your screen, showing that it not only computed the problem in line 10, but also remembered the result.

How does it remember?

Let's try an experiment. Type: POKE 0,1 and notice that nothing happens (at least nothing we can see). In fact, the "lowest" cell in your computer's memory just changed. POKE is a way that you can directly change a memory cell. Each cell has its own *address*, just as if each cell were a house in a city with 65536 houses. (The addresses go from 0 to 65535.) So, when you POKE, you are replacing the number in that cell with whatever number follows the comma in your POKE command. (If your machine does not have the maximum memory, some zones will not respond to POKE. Also, you can't expect to POKE into the ROM memory because it never changes – its contents are carved in it forever.)

Now type: ? PEEK (0) and notice that our "1" is printed (? is short for PRINT). The "1" is still in the 0 address cell and it will stay there until you POKE something else in, or the computer, in the course of performing a task, needs to use that cell to remember a number of its own. It will also "forget" if you turn off the power. But we still don't yet know just *how* this number is remembered in that cell.

Why We Count By Tens

Another name for a cell in your computer's memory is *byte*. If you have 16K of free memory (a *K* is 1024 bytes, called a *kilobyte*) this means that you have 16 X 1024 cells for a total of 16384 cells for your use. What are these cells? Each one is able to "remember" numbers. How high can a cell count? A byte can hold the numbers from 0 up to 255.

Nobody knows for sure, but it seems a good guess that we humans count in groups of 10's (*decimal*) because long ago, in caves, when the hunters were asked how many bison they saw that morning, the leader would likely hold up a certain number of fingers in the air. On a good day, the leader might say, "We saw many. My hands plus Joe's hands and one hand more!" The people would nod and vaguely understand the concept of 25 bison.

We tend to think that 10's are logical, even a natural, way to count. After all, so many things seem to fall into groups of tens: a dime, a quarter (four hands plus one), a dollar, a $10 bill, etc. Nevertheless, if lobsters ran the world, it is quite possible that things would be counted in fours. If you're curious, the most "natural" number grouping (found in shells, feathers, flowers and so forth) is called the Fibonacci series. The next number in this series is equal to the sum of the two previous ones: 0,1,1,2, 3,5,8 and so on.

So, since 10's are not natural except to creatures with that many fingers, there is no reason why a computer should count by 10's. For something that runs on electricity, the easiest way to count is by 2's (called *binary*, meaning "can be counted using only two symbols, the zero and one"). The computer finds binary counting very simple since this lets it represent numbers by merely turning something on (like a lightbulb) to stand for "1" or leaving it off to mean "0". If it had to deal with 3's, and 7's, and the rest, it would need to recognize and juggle ten different symbols, 0 through 9, like we do. This

POOL 1.5™

ATARI™ OWNERS,

NOW YOU TOO CAN PLAY POOL 1.5 ON YOUR ATARI 48K DISK SYSTEM.

ALSO AVAILABLE FOR THE APPLE II™ 48K DISK SYSTEM.

PRICE: $34.95

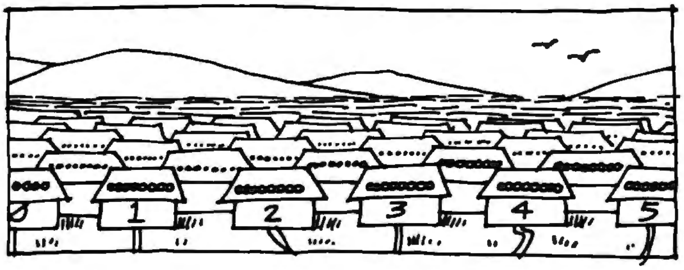
- **High resolution graphics- Life like motion.**
- **Instant replay on any shot.**
- **Superfine aiming.**
- **Pool room sound effects**
- **Choice of 4 popular games**

See your local computer dealer or order directly from IDSI. Specify ATARI or APPLE.

IDSI
P.O. BOX 1658
LAS CRUCES, NM
88004
Ph. (505) 522-7373

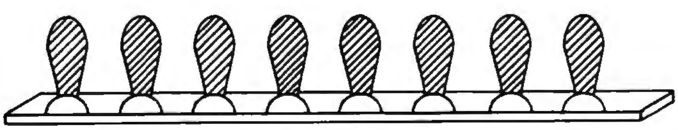
is what we mean when we say that the decimal system has ten *digits* (0-9). The binary system has only two digits, zero and one. By the way, *digit* also means *finger*.

In The City Of Bytes

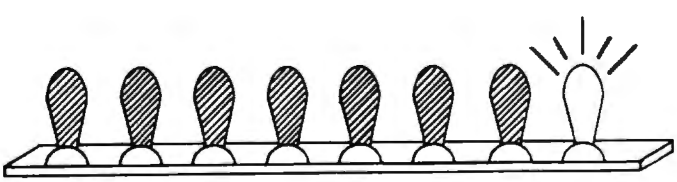


In our computers, there are thousands of rows of bytes lined up. Each of these bytes has an "address" of its own which is just its position in the city of bytes. POKE 5,1 puts the number one into the fifth byte and POKE 8001,0 changes the 8001th byte to zero. We can easily visualize bytes if we think of each one as a house with an odd decoration on the roof: a row of eight lightbulbs. Let's imagine that we have just turned on our computer and all our free bytes are "empty" – we haven't yet typed in a program or loaded anything into them from a tape or disk.

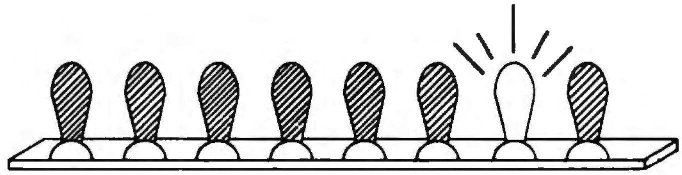
If we fly over the city of bytes, we see only darkness. Each byte contains nothing (zero) so all eight of its bulbs are off. (On the horizon we can see a glow, however, because the computer has memory up there, ROM memory, which is very active with its built-in programs.) But we are down in RAM (our free user-memory) and every house is dark. Let's observe what happens to an individual byte when different numbers are stored there; we can randomly choose byte 1504.



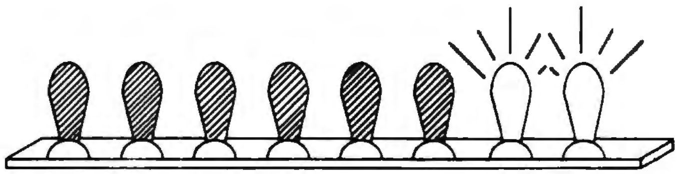
Like all the rest, this byte is dark. Each bulb is off. Observing this, we know that the byte here is holding a zero. If someone types in POKE 1504,1 – suddenly the rightmost lightbulb goes on and the byte holds a one instead of a zero:



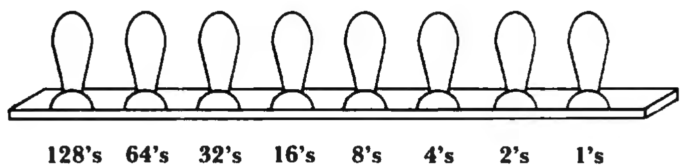
This rightmost bulb is the 1's column (just as it would be when counting by tens in our *decimal* system.) But the next bulb is in the 2's column, so POKE 1504, 2 would be:



And three would be one and two:



In this way – by checking which bulbs are turned on and then adding them together – the computer can look at a byte and know what number is there. These lightbulbs are called *bits* which is short for *binary digit*. Each bit is in its own position and has a value twice the value of the one just before it:



The Columns

Here is a program which will show you a byte as it looks in binary. You then try to give the number in decimal. It has been designed to run on Atari, PET, or Apple. Don't type in any line which has REM in it *unless* it's followed by the name of your computer.

```
100 REM BINARY QUIZ
110 C1=20:C0=111: REM FOR ATARI ONLY
120 C1=88:C0=79:  REM FOR APPLE ONLY
130 C1=209:C0=215:REM FOR COMMODORE ONLY
140 X=INT(256*RND(1)): D = X: P = 128
150 PRINT CHR$(125);: REM ATARI ONLY
160 PRINT CHR$(147);: REM COMMODORE ONLY
170 HOME: REM APPLE ONLY
180 FOR I = 1 TO 8
190 IF INT(D/P) = 1 THEN PRINT CHR$(C1);:
    D = D-P: GOTO 210
200 PRINT CHR$(C0);
210 P = P/2: NEXT I: PRINT
220 PRINT " WHAT IS THIS IN DECIMAL?"
230 INPUT Q: IF Q = X THEN PRINT
    "CORRECT": GOTO 250
240 PRINT "SORRY, IT WAS";X
250 FOR T = 1 TO 1000: NEXT T
260 GOTO 140
```

©

SYSRES™

THE ULTIMATE RESIDENT PROGRAM MANIPULATION SYSTEM FOR PET™/CBM™ MICROCOMPUTERS

SYSRES™ EXTENDED DOS SUPPORT COMMANDS

@ (type "N" keyboard)
← (type "B" keyboard)
] (original keyboard)
> (for 'wedge' users)

These commands may be used interchangably, to perform the following dos support functions.

Disk	Printer	Tape	Directory	Modes	Command	Function
x				3	@	Display disk status / send command
x					@N	Format (header) a new diskette
x					@I	Force initialize diskette
x					@V	Validate diskette (collect)
x					@D	Duplicate diskette
x			x	4	@C	Copy or concatenate disk file(s)*
x					@R	Rename file
x			x	3	@S	Scratch file(s)*
x					@$	List directory**
x					@U:	Reset disk drive
x	x	x	x	6	@L	List disk file or BASIC program**

Note: Some of the disk utility command set may also be used, if an appropriate direct access channel has been opened.

* Standard command with added options.
** Added disk command.

SYSRES™ EXTENDED EDITOR COMMANDS

Disk	Printer	Tape	Directory	Modes	Command	Function
x			x	4	/	Quick load from disk
x			x	4	↑	Quick load from disk with auto run
x			x	2	APPEND	Append from disk to end of current program
				4	AUTO	Auto line number (allows header)
x			x	3	BLOAD	Load machine language (binary) file
x			x	3	BRUN	Load and execute machine language program
	x			776	CHANGE	Change pattern to another pattern
				2	CLOSE	Close one or all files
				1	CMD	Set output to file (does not send "READY.")
				4	DELETE	Delete a range of lines from program
	x			1	DUMP	Dump all scalar variables to screen or file
x			x	2	EXEC	Execute a file as keyboard commands
	x			240	FIND	Find occurances of a pattern
x		x	x	3	GET	Read a sequential file into editor
				7	KEY	Define a key as a special function
				1	KEYS	Turn key functions on
				1	KILL	Disable SYSRES™
				1	KILL*	Disable SYSRES™ and unreserve memory
	x			10	LIST	Improved BASIC LIST command
x		x	x	3	LOAD	Defaults to disk drive
x			x	2	MERGE	Merge from disk into current program
	x			1	MON	Break to current machine language monitor
				1	OLD	Restore program after "NEW"
x	x	x	x	24	PUT	Send program to disk as text file
				6	RENUMBER	Renumber all or part of program
				2	RUN	Run current program, ignores screen garbage
x		x	x	3	SAVE	Defaults to disk drive, allows replace
x		x		1	SETD	Set disk device #, allows multiple drives
	x			4	SETP	Set printer channel, format mode, paging
				4	TRACE	Select 1 of 3 trace/step modes and speed
x		x	x	3	VERIFY	Compare current program against disk/tape
				1	WHY	Print position of last error
				1	WHY?	List line of break or error
x	x				*	Send output to printer
	x			1	#	Display current version of SYSRES™

JUST A FEW OF THE FEATURES OF SYSRES™

* Fast up/down scrolling which works!
* Advanced repeat–key routine!
* Re–define any or all keys as any keyword (full or short form) or as any string up to 255 charactors long!
* Auto line numbering which can feed a string of up to 127 charactors as well!
* Extended DOS support (requires DOS 2A or greater)!
* Never enter another file name! All file commands work from the directory!
* Supports multiple disk drives!
* List BASIC programs, sequential and relative files without loading them into memory!
* TRUE PROGRAM MERGE (overlay). Supports subroutine libraries!
* Load and run machine language programs with parameter passing!
* Supports multiple printers!
* Automatic printer output with paging plus formatted listings with full ASCII code conversion including cursor control and special charactors for non–CBM™ printers!
* Edit text files and assembler source code without leaving BASIC!
* Renumber part of a program or even change the order of lines!
* Over 700 FIND/CHANGE commands including variable names ("A$" will not match "BA$"), pattern matching with "wild–cards", and even commands to remove spaces and REM's!
* Three TRACE modes including trace variables!
* Does not affect BASIC program operation!
* One AUTO–BOOT DISKETTE works for ALL PET™ or CBM™ computers (BASIC 2.0 or greater with at least 16k of RAM.). SYSRES™ requires NO ROM SPACE or extra boards, so you can take it with you if you want to use another computer. It may be put above the screen if you have RAM there. It boots automatically without disturbing any program in RAM!
* Diskette and Extensive Manual – only $75 (cdn)
Documented Source Code (YES!) – $150 (cdn)
Custom versions are available for unique systems!

Dealer enquiries are welcome.

VISA MasterCard

CANSOFT DATA INC.

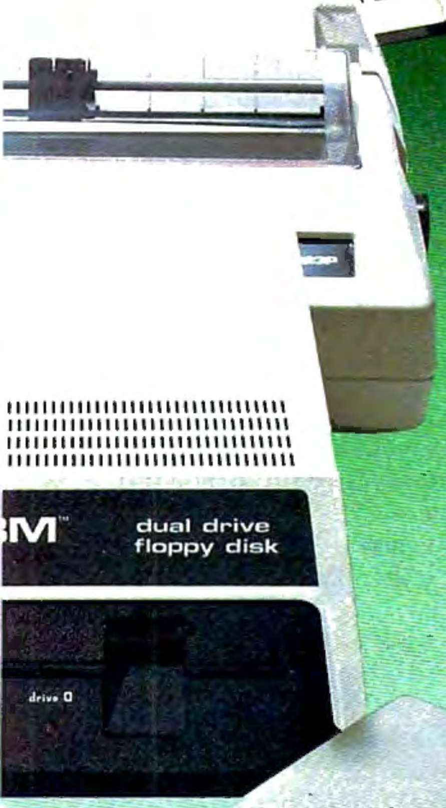
MICROCOMPUTER SERVICES

PROGRAMMING * CONSULTING * DESIGNING * EVALUATING

CANSOFT

BOX 76752, STN.S, VANCOUVER, B.C. V5R 5S7 TEL:(604)437-6667

leads to another..



MUPET II
saves money and time too!

Mupet II offers all the advantages of Mupet, plus:

- Will store commands if disk and printer are being used allowing continued use of computer and completion of stored command in turn.
- Protects against incorrect use of harmful disk commands.
- No more printer interface - background printing supported on IEEE, RS 232C and parallel ports included in controller.
- Programmable 16K RAM controller, expandable to 32K provides user defineable operating systems.
- Up to 16 computers may be connected to central printer and disk drives.

Integrate intra-office terminals for common data access allowing independent analysis for such functions as:

- Accounting
- Sales forecasts, budgets and orders
- Production scheduling
- Word processing

The only product available capable of serving Commodore Computers up to 100 feet apart.

MUPET II a small price for huge capabilities, from:

$995.00

The Winter Of Our Content

A Report On The January Consumer Electronics Show

David D Thornburg
Innovision
Los Altos, CA

At a time when most of us are contemplating sugar plums, planning New Year's parties, or replacing shingles kicked loose by Santa's reindeer, a few hardy souls are preparing for a major event – the Winter Consumer Electronics Show held each January in Las Vegas.

In the area of personal computers, this year's show will be long remembered. Formal announcements by Commodore, Panasonic, Astrovision, Sharp, and Toshiba, more than justified leaving the remains of our sudden 12" rainstorm for the sunny (if somewhat cold) desert of Nevada.

Here, then, are the highlights of some of the more exciting products for 1982. As you read this, remember that many of the products displayed were prototypes, and that they may undergo some modification before going to market.

For sheer impact, Commodore stole the show with the announcement of two new color computers! Imagine a color computer with separate graphics and sound chips, two memory-mapped graphics modes, the traditional PET 40 column display, three voice music and sound effects, and cartridge programmability. Sound interesting? Got your checkbook ready? The price for the Commodore Ultimax is $149.95. Yes, the decimal point is in the right place. For less than two-hundred dollars, you will be able to do wondrous things which previously required much more expensive hardware.

What's the catch? Well, the Ultimax doesn't have a full stroke keyboard; it uses the membrane type instead. Also it doesn't have a lot of RAM. However, you do get access to a 300 x 200 dot graphics display (two color mode), or a 160 x 200 screen (four colors at a time out of 16). Also (for you Atari and TI users), Commodore gives you access to up to 255 animated "sprites" per screen (up to 8 per line). Each sprite (or player, for Atari fans) resides in a 63 byte field and is composed of 24 x 21 dots.

The Ultimax uses the VIC peripherals (joysticks, cassette, etc.).

But what about those of us who want more RAM? Well, Commodore has us in mind too. The Commodore-64 comes with 64K of RAM. Using the same chip set as the Ultimax, and equipped with a full-stroke keyboard, this gem sports a $595 price tag.

Both new computers use the 6510 processor (a stretch 6502, I guess), the 6566 graphics controller, a 6581 sound interface device, and a 6526 PIA. Don't expect to see these chips at your local parts shop for awhile. Commodore plans to use every one they can make. While neither new computer will be on the market immediately (about June for the Ultimax – so *please* don't bug your dealer yet!), Commodore's marketing wizard, Kit Spencer, boldly predicted that Commodore would sell more computers in 1982 than were sold by all manufacturers in 1981.

While the Ultimax (with its game-like price) may lead the sales into the homes, both machines will probably penetrate the school market as well. The reason for this is that they both have a serial port capable of communicating with up to 30 other computers. Just what the teachers ordered!

Commodore wasn't the only booth sporting ear-to-ear grins. Panasonic introduced the 6802-based JR-100 – a $199.95 compact computer which comes with 16K of RAM and an 8K BASIC. The system can be expanded to 48K RAM. While this monochrome computer has only a character-based display (32 characters by 24 lines), the user is free to define his or her own character set. A color version of this computer will be available for $299.95 at some future date. While I was impressed by the overall physical layout of this computer, one aspect of the keyboard bothered me. The space bar was located where the right SHIFT key should be. Perhaps this will be changed by the time the product gets to market. The JR-100 was designed (and will be built) by a different group than the one respon-

DTL BASIC COMPILER

- Compatible with all existing Basic interpreter commands
- Accepts extensions to Basic implemented in RAM or ROM
- Provides demanding two pass syntax and logic analysis
- Improved Performance based on faster execution times
- Large Programs (16K+) will benefit from memory savings
- Security Key attaches to either cassette port

A Basic COMPILER for your Commodore Microcomputer by Drive Technology

DTL-BASIC is a Basic compiler for Commodore machines designed to convert existing programs to machine code and run them without modification. Compiled programs will run much faster and operate in exactly the same way as the un-compiled versions. Compiled code is typically 20 to 50% smaller than source code. For large programs this saving will more than offset the 4K run-time library appended to each compiled program, providing additional internal memory space.

The compiler implements true integer arithmetic as well as real arithmetic. Use of integers can lead to significant speed improvements. Special compile time options make identification and conversion of real variables to integers a simple task.

A 'Compiler' security key, which plugs into either cassette port, is supplied together with the DTL-BASIC compiler. This key must be used in order to compile a program or to run the compiled version. In order to allow for the distribution of compiled versions of user developed programs, a second type of key known as a 'Run-Time' key is available in any required quantities. Software developers can obtain private security key sets with unique serial numbers providing comprehensive protection of their products while allowing customers to make backup copies of compiled programs.

DTL-BASIC is a disk based system requiring a 32K PET/CBM and comes complete with an in-depth user manual and a Compiler Security Key. Three versions of the compiler exist for CBM 3032, CBM 4032, and CBM 8032 machines. Please specify machine type and disk type (4040 or 8050) on which compiler is to be supplied.

DTL BASIC WITH MANUAL AND SECURITY KEY $350.00

RUN-TIME KEYS . . . . $50.00 EACH

FROM

CMD
CANADIAN MICRO DISTRIBUTORS LTD.

365 Main St., Milton, Ontario L9T 1P7
416 878-7277

THE COMPANY THAT BROUGHT YOU
MUPET • DOUBLE-MUPET • SPOOLER
THE MANAGER • I/O PRODUCTS

sible for the 6502-based hand held computer being sold by Panasonic and Quasar. The JR-100 will be sold to the home market – primarily the hobbyist and home study market at first. A full line of peripherals is planned – modems, floppy disks, etc.

Astrovision demonstrated the Bally Arcade with a plug-in expansion unit housing a full-stroke keyboard and running the language ZGRASS. This graphics language was developed (if memory serves) by Tom Defanti at the University of Illinois in Chicago. The goal is to allow artists and other non-technical types to create breathtaking real-time animations without having to learn machine language programming. The result is beautiful. Since the Z80-based Astrovision game machine has the same multi-chip display controller used in Bally's coin operated arcade games, high quality graphics is possible. What makes it practical is the ease with which figures drawn on a tablet can be saved as procedures, located anywhere on the screen, and can be magnified and rotated at will.

Astrovision seems intent on aggressively pursuing this project, but it isn't clear how effectively they will be able to compete. As with the other machines mentioned so far, don't expect to see the ZGRASS machine next week.

For those who take their graphics seriously, Toshiba was pleased to announce the T100 computer. When used with a high resolution color monitor, the user has access to 8 colors on a 640 by 200 dot array. A multi-line liquid crystal display was shown as an alternative. The packaging of this CP/M machine reminded me of the NEC PC-8000 – very stylish and business-like. Equipped with a Z80, 32K of ROM and 64K of RAM, this computer should appeal to those who are interested in doing some serious work.

Sharp had a surprise of its own. The original PC-1200 hand-held computer (available from Radio Shack for some time) has been joined by an 8-bit brother: the PC-1500. While this new machine is much faster than its predecessor, the most amazing feature was the availability of a color graphic printer. For a total price of $550, Sharp users will have access to a computer with a pocket-sized drum plotter that lets you draw pictures with any of four automatically selected pens (red, green, blue, black). This is one product that has to be seen to be believed – and even then you won't believe it.

The PC-1500 is available now. (Of course I ordered one!)

At long last, the Casio FX-9000 desktop computer is on the market. This computer has a built-in 5" monochrome display with the ability to show graphics images with a 256x128 resolution. The sleek styling is reminiscent of the Hewlett Packard HP-85, but the price (under $1800, fully loaded) is more in keeping with Casio's products. The FX-9000 uses a Z80 compatible processor and gives the user access to two types of RAM. Front panel accessibility to RAM cartridges lets the user choose between 16K dynamic RAM cartridges, and 4K CMOS cartridges. The 4K byte RAM module contains its own battery, so that programs which are saved on this module can be removed and saved for instant reloading later.

Texas Instruments developed and showed a similar 4K RAM cartridge for the 99/4. Since TI also showed a new Assembler/Editor system, the ability to let users make their own cartridges is quite appealing. My, what a difference a year makes.

TI also showed a nicely designed expansion box which does much to reduce the clutter associated with fully loaded systems.

Atari's booth was almost impossible to enter. It was as if everyone who attended the show decided to check them out at the same time. The official authorized Atari Pac Man program was introduced. The action is a little different from the previous version done for the Atari computer (Jawbreaker) by OnLine Systems. Compared to the Pac Man games also shown by Magnavox, Astrovision, Texas Instruments, and others, Atari did a very nice job at software development. Coleco's hand-held version of the game was not as exciting, but then it doesn't use a color display screen either.

While my emphasis has been on hardware, it should be noted that software was on display as well. Automated Simulations displayed some of their newer programs for the Apple and Atari computers. While they are known primarily for their adventure games, Automated Simulations has developed a line of educational games (such as Jabbertalky) which are terrific.

The fact that a few dedicated people are willing to make the effort to generate educational games which teach as well as entertain should not go unrewarded. These people are doing an excellent job and are to be congratulated.

Both Activision and IMAGIC showed cartridges for the Atari video game which demonstrated exceptionally high quality graphics. Not surprisingly, these booths were well attended as well.

So much to see, and too little time! I was so busy getting all this information for you dear readers that I barely got to examine the solar-rechargeable flashlights, underwear with built-in loudspeakers, and ball point pens with built-in clocks and music synthesizers.

But that is why there are two shows a year. So until the next CES this June in Chicago, I'll just have to be content with what I saw. ©

Basically Useful BASIC

Tabulation

Paul Lilly
Pelham, AL

If you have ever written a program that output a wide range of numbers, you probably noticed the numbers were left justified (first digit of number printed at next print location regardless of the size of the number). Additionally, if you wanted to print a number such as $9.20, it would be printed as 9.2, omitting the trailing zero.

So what do you do if you want to write a nice clean program that would balance your checkbook, list expenses, or otherwise output a series of numbers, right justified, in a nice straight column? Answer: use a short subroutine to measure, add trailing zeros if needed, and right justify the number when it is printed.

The Subroutine

The key to the subroutine is to convert the number to a string using the STR$ command. After the number is a string, we can add or subtract to its length, adjust it to our desired format, and control where it is printed by using the TAB and LEN commands together.

Program 1 is an example of such a subroutine. The instructions are written starting on line 5000 so it could be inserted in most programs without interfering with the existing instructions. We also have created two variables and one string that should be reserved from the rest of the program.

RJ – the print position for the rightmost digit in the number we want to print.

N – the number we want to format and print.

N$ – the ASCII string of N.

The subroutine can be entered at four different locations, depending on our needs.

1. If we want our number formatted with two digits to the right of the decimal point (as when dealing with dollars and cents figures), we can call the subroutine at 5010. Prior to calling the subroutine we must set N to the number we want to print, and RJ to the print location where we want the rightmost located.

2. If we call this subroutine several times throughout our program, and our print locations vary, we can call subroutine 5000. Line 5000 will set RJ to seven positions past our last print location. This will reduce the programming steps needed in our main program. With seven positions (past last print position) we can print numbers up to 9999.99. For larger numbers, line 5000 can be changed accordingly.

3. If we do not want any decimals in our number, we can call the subroutine at 5090. Here again, N must be set to the number we want printed, and RJ to the print location of our rightmost digit, prior to calling the subroutine.

4. And, finally, if we want to print a non-decimal number, and would like to save programming steps, we can enter the subroutine at 5085 and let the subroutine figure RJ for us.

The Method

Suppose we enter at line 5000 with N set to the number we would like printed. Line 5000 will set RJ to six print positions past our current (last printed) print position. Line 5010 will strip away any more than two digits right of the decimal point (it is still possible at this point to have only one or no digits right of the decimal). Let's look at lines 5040 and 5050. These statements look for the decimal (".") to determine if N$ has a decimal point, and how many digits (one or two) are to the right of it. Line 5040 finds the decimal point one position back from the rightmost character, meaning there is only one digit to the right of the decimal point. Therefore, the statement adds the necessary trailing zero, then jumps ahead to 5070.

Line 5050 finds the decimal point two positions back from the rightmost character, meaning there are already two digits to the right of the decimal point. In this case, no modification is needed to N$, so the statement simply jumps ahead to 5070. Now the only possible configuration left for N$ is for it to have no decimal in it. In this case line 5060 tacks on the required ".00", to maintain the standard format. Now let's go back to 5030. If N equaled 0-9, the LENth of N$ would be only two and would bomb the program at 5050. So, since any string with less than a lenth of three can't have a decimal point in it anyway, 5030 will catch it and route it around 5040 and 5050. (By the way, the STR$ command (at least in my machine) puts a leading space in front of the ASCII representation of the number, thus the number 1 has a string LENth of two.)

That brings us to line 5070, which strips the leading space so it doesn't interfere with our print position calculations. Finally on line 5075 we TAB the difference between RJ and LEN(N$), and print our number. Line 5085 is where we would enter if we wanted to print integers. Line 5090 would strip the number to an integer, then change it to a string. Line 5095 would jump back to where we remove

LET THE MANAGER DO IT!

Inventory Control
Mailing Lists
Accounting systems
Personnel
Costing
Gathering test data
Budgeting
Scheduling

Organize and File Your Records Without a Programmer

The easy to use Database System suited for both businessman and programmer.

Today's Filing Cabinet for CBM

- Never again misplace an invoice, important document or key customer address. The Manager is your filing cabinet.
- Talk to the Manager and it responds.
- Take command of your filing, reporting and analysis requirements.
- Create, utilize and store information in familiar and compatible forms.
- The Manager automatically performs all predefined calculations as required in your business.
- Utilize it for one major function or include many important records for recall, update and print-out for analysis.
- Keep inventory levels and controls manageable in today's economic environment - the Manager shows you how.
- The Manager saves money, no in-house programmer required; you and/or your secretary can operate the system effectively.
- The Manager saves time; fast search capabilities allows speedy recall and easy merging.

Can You Afford Not To Have The (office) Manager Working For You?

For futher information and the name of your nearest dealer, call CMD - the company with world renowed expertise on Commodore and distributors of Mupet and Mupet II, The Manager, DTL Basic Compiler, and Supergraphics.

THE MANAGER

$250.00

CMD

CANADIAN MICRO DISTRIBUTORS LTD.
365 Main St., Milton, Ontario L9T 1P7
416-878-7277

Announcing the Printing Breakthrough of the Century: Smith-Corona® TP-1™ Text Printer

- **Low Cost Daisy Wheel Printer**

$845.00

- Microprocessor Electronics
- Serial or Parallel Interface
- Simple, Reliable Mechanism

ACT NOW: Limited Supply, Low, Low Cost

Smith Corona, one of the largest manufacturers of small printers in the world, gives a whole new perspective to printing with their electronic text printer—**TP-1.** The **TP-1** is a microprocessor controlled, high quality **daisy wheel printer.** It produces perfectly formed, executive quality printouts at the speed of 120 words per minute. Typewriter quality printing at dot matrix prices.

Simple, durable and dependable, **TP-1** may be used with word processing systems, microcomputers and most small business systems. Compact and attractively styled, the **TP-1** blends well with any setting.

Additional daisy print wheels $4.95

Additional ribbons $2.95

Now, all your letters, documents forms and reports can have the crisp, professional look you demand—for business or personal use—at an affordable price. **TP-1,** the electronic text printer.

Don't delay. Order your **TP-1** TODAY at the low price of **$845.**

Micro Printer Marketing offers same day shipping, nationwide service and invites dealer inquiries. Catalogues available. No shipping charges on pre-paid orders.

Micro™ Printer Marketing

Call Micro-Printer Marketing

CALL TOLL FREE **1-800-523-9859**

In PA Call Collect **215 / 433-3366**

MasterCard and Visa Accepted

the leading space and print the number.

Considerations

Line 5075 uses the semicolon after printing N$ to supress the LF/CR. This is because you may want to print several numbers on one line. Once you have printed the last number on a line, have the main program execute a PRINT command so you will get your carriage return. If you only want to print one number per line and no further printing will be done after the number, you may remove the semicolon from line 5075.

Also, be careful not to set RJ to a number smaller than N$ would be after the modifications were made to it, otherwise you will not space properly with the TAB command on line 5075 and, possibly, generate an error that will bomb your program.

Program 2 will show a comparison between two different interest rates, of 15 different loan amounts. The user is asked to enter two rates, principal for first comparison, increment of principal for succeeding comparisons, and the term of the loan in months. The program will output the results neatly.

Figure 1.

Sample Run #1

```
INITIAL PRINCIPAL
? 50000

INCREMENT PRINCIPAL BY
? 2000

INTEREST RATE 1
? 10

INTEREST RATE 2
? 14

TERM OF LOAN IN MONTHS
? 360




     FOR 360 MONTHS

LOAN        10 %      14 %

 50000    438.79    592.44
 52000    456.34    616.13
 54000    473.89    639.83
 56000    491.44    663.53
 58000    508.99    687.23
 60000    526.54    710.92
 62000    544.09    734.62
 64000    561.65    758.32
 66000    579.20    782.02
 68000    596.75    805.71
 70000    614.30    829.41
 72000    631.85    853.11
 74000    649.40    876.80
 76000    666.95    900.50
 78000    684.51    924.20
```

Program 1. Microsoft Version (Apple, PET, etc.)

```
5000 RJ=POS(RJ)+7
5010 N=INT(N*100+.5)/100
5020 N$=STR$(N)
5030 IFLEN(N$)<3GOTO5060
5040 IFMID$(N$,LEN(N$)-1,1)="."THENN$=N$+"0":
     GOTO5070
5050 IFMID$(N$,LEN(N$)-2,1)="."GOTO5070
5060 N$=N$+".00"
5070 N$=RIGHT$(N$,LEN(N$)-1)
5075 PRINTTAB(RJ-LEN(N$));N$;
5080 RETURN
5085 RJ=POS(RJ)+6
5090 N$=STR$(INT(N))
5095 GOTO5070
```

Program 2. Microsoft Version

```
100 PRINT:PRINT"INITIAL PRINCIPAL":INPUTLA
110 PRINT:PRINT"INCREMENT PRINCIPAL BY":INPUTLI
120 PRINT:PRINT"INTEREST RATE 1":INPUTI1
125 IFI1<1ORI1>100GOTO120
130 PRINT:PRINT"INTEREST RATE 2":INPUTI2
135 IFI2<1ORI2>100GOTO130
140 PRINT:PRINT"TERM OF LOAN IN MONTHS":INPUTNM
150 PRINT:PRINT:PRINT:PRINT:PRINT
160 PRINT"     FOR";NM;"MONTHS"
170 PRINT:PRINT"LOAN      ";I1;"%    ";I2;"%":
    PRINT
180 FORJ=LATOLA+(14*LI)STEPLI
190 I3=I1/1200:I4=I2/1200
200 P1=J*(I3/(1-(1/(1+I3)^NM)))
210 P1=INT(P1*100+.5)/100
220 P2=J*(I4/(1-(1/(1+I4)^NM)))
230 P2=INT(P2*100+.5)/100
240 N=J:GOSUB5085
250 PRINTTAB(8);:N=P1:GOSUB5000
260 PRINTTAB(16);:N=P2:GOSUB5000
270 PRINT:NEXTJ
280 END
```

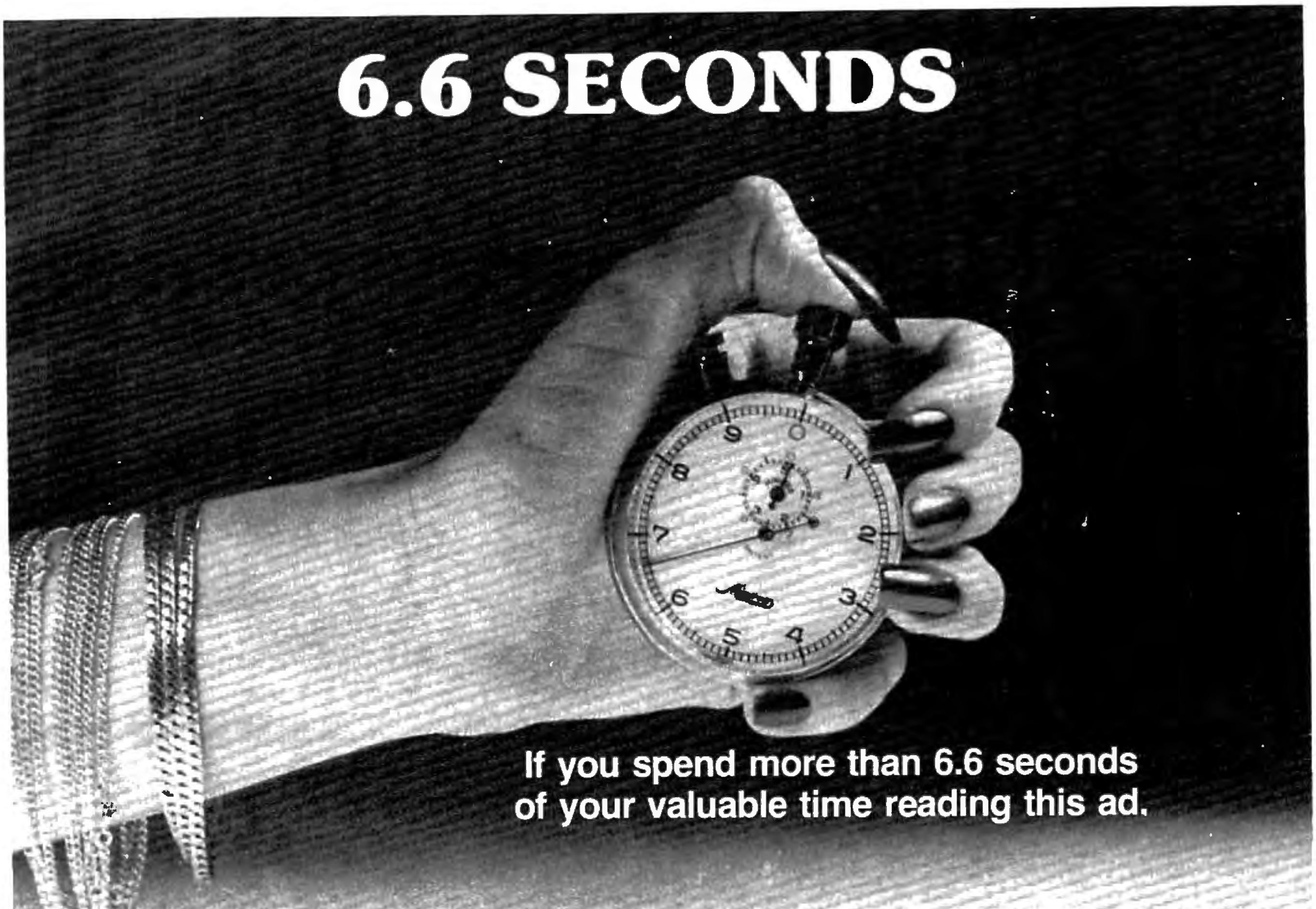
Program 3. Atari Version

```
5000 RJ=PEEK(85)+7
5010 N=INT(N*100+0.5)/100
5020 GOSUB 5100:N$=STR$(N)
5030 IF LEN(N$)<3 THEN 5060
5040 IF N$(LEN(N$)-1,LEN(N$)-1)="." THEN
 N$(LEN(N$)+1)="0":GOTO 5070
5050 IF N$(LEN(N$)-2,LEN(N$)-2)="." THEN
 5070
5060 N$(LEN(N$)+1)=".00"
5070 POKE 85,RJ-LEN(N$):? N$;
5080 RETURN
5085 GOSUB 5100:RJ=PEEK(85)+6
5090 N$=STR$(INT(N))
5095 GOTO 5070
5100 TRAP 5110:DIM N$(15):TRAP 40000
5110 RETURN
```

Program 4. Atari Version

```
100 PRINT :PRINT "INITIAL PRINCIPAL":INP
UT LA
110 PRINT :PRINT "INCREMENT PRINCIPAL BY
":INPUT LI
120 PRINT :PRINT "INTEREST RATE 1":INPUT
 I1
125 IF I1<1 OR I1>100 THEN 120
130 PRINT :PRINT "INTEREST RATE 2":INPUT
 I2
135 IF I2<1 OR I2>100 THEN 130
140 PRINT :PRINT "TERM OF LOAN IN MONTHS
":INPUT NM
150 PRINT :PRINT :PRINT :PRINT :PRINT
160 PRINT "    FOR ";NM;" MONTHS"
170 PRINT :PRINT "LOAN     ";I1;"%    ";
I2;"%":PRINT
180 FOR J=LA TO LA+(14*LI) STEP LI
190 I3=I1/1200:I4=I2/1200
200 P1=J*(I3/(1-(1/(1+I3)^NM)))
210 P1=INT(P1*100+0.5)/100
220 P2=J*(I4/(1-(1/(1+I4)^NM)))
230 P2=INT(P2*100+0.5)/100
240 N=J:GOSUB 5085
250 POKE 85,8:N=P1:GOSUB 5000
260 POKE 85,16:N=P2:GOSUB 5000
270 PRINT :NEXT J
280 END
```

©



YOU NEED

SpeedRead+

- Optimized Systems Software, the company that brought you BASIC A+ and OS/A+, proudly presents another "*PLUS*".
- ***SpeedRead+*** is the *world's first* speed reading tutor designed for use on your personal computer.
- ***SpeedRead+*** *begins* with training your eyes and mind to function as the incredible precision machine they were meant to be.
- ***SpeedRead+*** goes beyond mere words — it trains you to recognize phrases and columns instantly — it exercises your peripheral vision — it increases your comprehension.
- ***SpeedRead+*** matches *your* pace — now and in the future.

AVAILABLE NOW!

For 16K ATARI® computers with disk, Introductory Price **$59.95**
Coming soon for APPLE II®. Inquire about availability of TRS-80® and cassette versions.

SEE YOUR DEALER TODAY!

Call or write for a descriptive brochure of ***SpeedRead+***™ and other fine OSS products, such as OS/A+, BASIC A+, and TINY C®.

ATARI, APPLE II, TRS-80, and TINY C are trademarks of Atari, Inc., Apple Computer, Inc., Radio Shack, and Tiny C Associates, respectively. ***SpeedRead+*** is a trademark of Eagle Software and Optimized Systems Software.

Optimized Systems Software, Inc., 10379 Lansdale Ave., Cupertino, CA 95014, (408) 446-3099

NEECO

WHY BUY FROM THE BEST?
Service... Support...
Software...

commodore

16K B (16K RAM-40 Column) - Lim. Qty	$ 995
32K B (32K RAM-40 Clm.) - Lim. Qty	$1295
4016 (16K RAM 4.0 Basic-40 Clm.)	$ 995
4032 (32K RAM 4.0 Basic-40 Clm.)	$1295
8032 (32K RAM 4.0 Basic-80 Clm.)	$1495
8050 Dual Disk (1 Meg Storage)	$1795
4040 Dual Disk (343K Storage)	$1295
8010 IEEE Modem	$ 280
C2N Cassette Drive	$ 75
CBM - IEEE Interface Cable	$ 40
IEEE - IEEE Interface Cable	$ 50
VIC 20 Home/Personal Computer	$ 295

CALL NEECO FOR ANY OF YOUR COMMODORE COMPUTER NEEDS

NEC SPINWRITER PRINTERS

5530 (Parallel)	$3055
5510 (Serial)	$3055
5520 (KSR-Serial)	$3415
Tractor Option	$ 225

EPSON PRINTERS

MX-80 PRINTER	$ 645
MX-80 FT	$ 745
MX-100	$ 945
MX-70	$ 459
INTERFACE CARDS	
8141 (RS-232)	$ 75
8150 (2K Buffered RS-232)	$ 150
8161 (IEEE 488)	$ 55
8131 (Apple Card)	$ 85
8230 (Apple Card)	$ 25
8220 (TRS-80 Cable)	$ 35

APPLE

16K APPLE II+	$1330
32K APPLE II+	$1430
48K APPLE II+	$1530
APPLE DISK w/3.3 DOS	$ 650
APPLE DRIVE Only	$ 490
APPLE III 128K - In Stock! w/Monitor + Info Analystpak	$4740

DIABLO 630 PRINTER

DIABLO 630 - Serial - RS-232	$2710
Tractor Option	$ 250

AMDEK MONITORS

Video 100 12" B+W	$ 179
Video 300 12" Green	$ 249
Color I 13" Low Res	$ 449
Color II 13" High Res	$ 999

INTERTEC COMPUTERS

64K Superbrain (360 Disk Storage), CP/M™	$3495
64K QD Superbrain (700K Disk Storage), CP/M™	$3995

*CP/M is a registered trademark of Digital Research.

ATARI COMPUTERS

Atari 400 (16K RAM)	$ 399
Atari 800 (32K RAM) - good thru 8/31	$1080
Atari 410 RECORDER	$ 89.95
Atari 810 DISK DRIVE	$ 599.95

NEECO carries all available ATARI Software and Peripherals.

PROFESSIONAL SOFTWARE

WordPro 1 8K	$ 29.95
WordPro 3 (40 Clm.)16K	$ 199.95
WordPro 3+	$ 295
WordPro 4 (80 Clm.) 32K	$ 375
WordPro 4+	$ 450

JUST A SAMPLE OF THE MANY PRODUCTS WE CARRY, CALL US FOR OUR NEW 60-PAGE CATALOG.
WE WILL MATCH SOME ADVERTISED PRICES ON CERTAIN PRODUCTS LISTED UNDER SIMILAR "IN STOCK" CONDITIONS.

NEECO
679 HIGHLAND AVE.
NEEDHAM, MA 02194

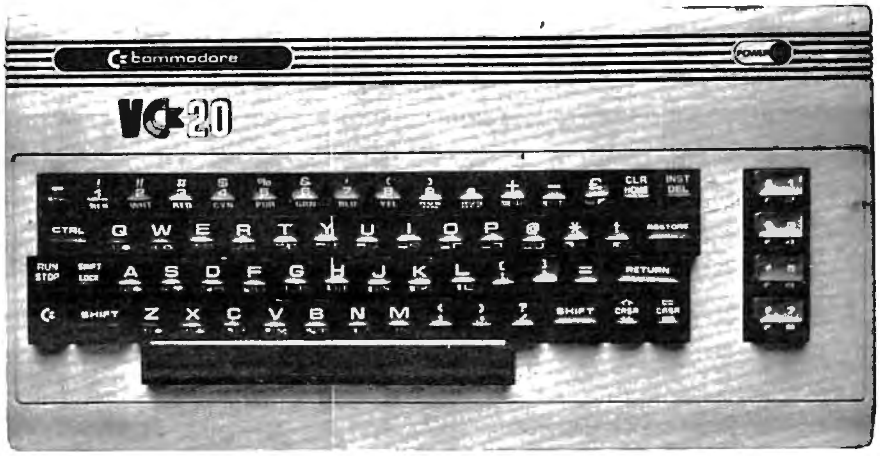
(617) 449-1760
Telex: 951021

MON-FRI 9:00 - 5:00
MasterCharge and VISA Accepted

NEECO INTRODUCES THE CBM VIC-20 COMPUTER!

Commodore breaks the computer price barrier —

$299.95

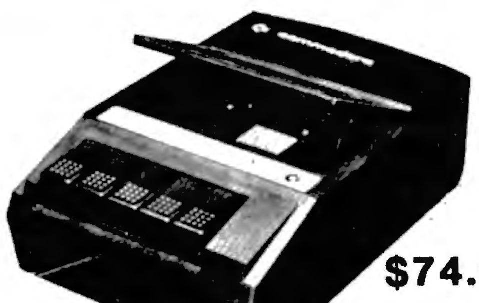


CBM VIC-20 PERSONAL COMPUTER

VIC-20 SPECIFICATIONS

- 8 colors - built in
- sound generation - built in
- programmable function keys
- 5K memory expandable to 32K
- standard PETBASIC in ROM
- full-size typewriter keyboard
- graphics character set
- plug-in program/memory cartridges
- low-priced peripherals
- joystick/paddles/lightpen
- self-teaching materials

* WORKS WITH ANY HOME TELEVISION



$74.95

C2N TAPE CASSETTE DRIVE

CALL NEECO TODAY FOR ADDITIONAL VIC-20 INFORMATION . . .
As the CBM VIC-20 is a "new" product, prices and specifications are subject to change w/o notice.

NEECO

NEECO WILL MATCH ANY ADVERTISED PRICE ON CBM EQUIPMENT FROM ANY OTHER COMPANY WITH PRODUCT IN STOCK.

(617) 449-1760
Telex: 951021

679 HIGHLAND AVE.
NEEDHAM, MA 02194

MON-FRI 9:00 - 5:00
MasterCard and VISA Accepted

Twenty Questions Revisited

Elizabeth Deal
Malvern, PA

Editor's Note: In the September, 1981 Computers And Society column, David Thornburg presented a fascinating program called Twenty Questions. Both versions (Microsoft BASIC and Atari BASIC) are repeated at the end of this article. Here, Liz Deal, a long time contributor to **COMPUTE!**, *presents an unedited transcript of her young son and a friend exploring the intrigues of Twenty Questions. — RCL*

In **COMPUTE!** #16, pgs. 12 and 16, there appeared a program called "Twenty Questions." Robert Lock asked for readers' response to the program. I think the reactions of two ten-year old boys might amuse you.

They liked the game, went through many rounds, obviously had some goal in mind, kept changing the rules and conclusions, and gave up feeling somewhat cheated at the end.

They religiously took turns between the typing and thinking tasks every few minutes, supplementing their typing with some loud thinking. I began recording several minutes into the game and noted words present on the screen. The kids' conversation is preceded by P and R to identify who is saying what. What they typed on the screen and what PET answered is in capital letters (shown on one line). Draw your own conclusions from a sample of two people.

.
.
.

RIGHT? NO
WRONG? NO
R: It's supposed to be an object
P: Ok
BIKE? YES
R: Is it an animal?
ANIMAL? NO
VEHICLE? YES
P: Is it a vehicle?
R: Yeah, try it, that's what it was before
IS IT A PERSON? NO
P: Is it an object? o-b-j-e...
R: I know how to spell it
IS IT AN OBJECT? NO
P: If it's not an object it's not anything...
R: I know
IS IT A NAME?
P: Name not an object, erase it
IS IT A VEHICLE? YES
R: That's what we should have tried before... again?
P: Try vehicle again, it's a lucky question
R: What if it's not?
IS IT A VEHICLE? YES
R: I bet it's always vehicle, let's try this
VEHICLE? YES
R: It's going to be always vehicle
P: Let's try vegetable
VEGETABLE? YES
R: I think if it begins with V is right. Try viking v-i-k-i-n-g
IS IT VIKING? NO
P: Let's see what happens if we go for all twenty...try one word and question mark on it
PAPER? NO
INVENTION? NO
IS IT AMPLIFIER? NO
GLUESTICK? NO
BIG BRICK? NO
HORSE? YES
P: aha...
R: I like horses, they are my favorite animal
CAT? NO
DOG? NO
ANIMAL? NO
P: Put shark
R: No can't be anything that's animal. I'll try tape, we haven't tried that yet
TAPE? YES
P: Try table
TABLE? YES
R: First guess again. This is good!
P: No it is not...it's like trying to eat a live pig without fork and a knife
DOG? NO
P: Do animal instead of certain word
ANIMAL? NO
TAPE? YES
P: Tape again, don't do it again, try t words
TREE? YES
P: Let's try c's
R: No cat didn't work... try h
HOUSE? YES
SNAK? NO
HOUSE? YES
R: I told you to put house before...
P: Well, I can do my own words...from my mind...


2 FROM HORIZON
Fathoms 40™
Time: June, 1943.
Place: North Atlantic.
German Wolf Packs have been driven from their hunting grounds by Allied escort vessels and massive air attacks. But unknown to the Allies is a Nazi super sub completed well ahead of schedule: The Mark XXI. Der Führer has demanded total destruction of enemy merchant shipping — a vital link for continued success of the Allied war effort. Will you, as one of the Kriegsmarine's remaining ace skippers, alter history and save the Third Reich from defeat?
Der Führer will not be disappointed!
FATHOMS 40: $39.95 ppd. Foreign, FPO and APO add 10% for shipping. Please specify Apple or Atari when ordering.
In the 1930's dramas were slowly unfolding in capital cities throughout the world. From dimly lit hotel rooms to highly sophisticated Intelligence Headquarters, men and women were playing a game of cat and mouse for scraps of information essential to the security of their country. CLOAK & DAGGER pits you against computer, or other players, as head of an Intelligence Agency trying desperately to stay one step ahead of its rivals. Moving agents from capital to capital, ferreting information, and offering bribes are fundamental parts of each player's turn . . . and your survival!
CLOAK & DAGGER™
CLOAK & DAGGER: $39.95 ppd. Foreign, FPO and APO add 10% for shipping. Please specify Apple or Atari when ordering.
For Apple II*, DOS 3.3, or Atari 800*, 48K with disk drive. Joysticks required. Games on Dysan** diskettes.
Visa & MasterCard accepted.
Dealer Inquiries Welcome, contact your local dealer for full details.
HORIZON SIMULATIONS
Horizon Simulations Inc
7561 Crater Lake Highway / White City / Oregon 97503
(503) 826-4640

R: Try gas
P: Gas isn't an object...it's a thing...well, it's an object in a way
R: We have to spell 'em right
ROCKET? NO
R: Try shuttle
P: No, it's a name
PIG? NO
R: Can't be an animal?!
P: I have to devise something in my head... (total silence several seconds, followed by STOP key somebody forgot to turn off)
LIST
P: OK, looking for the words...
R: Don't they have...
P: ...don't worry, we'll find the words...(reached end of listing)...all they have is that?!! Not fair! No words. Only goes to 410. Where did it get the words? Has to have memory where to get the words...
R: Let's do something else.

My PET will *think* when this conversation occurs:

ME: IS IT AN ANIMAL?
PET: NOPE
ME: IS IT A HORSE?
PET: I TOLD YOU IN PREVIOUS QUESTION IT'S NOT AN ANIMAL, STUPID

Program 1: Microsoft Version

```
100 REM *TWENTY QUESTIONS
110 PRINT CHR$(147);"WELCOME TO THE GAME
115 PRINT"OF TWENTY QUESTIONS.  BY
120 PRINT "ASKING QUESTIONS WHICH HAVE
130 PRINT "YES OR NO ANSWERS, TRY  TO
140 PRINT "GUESS THE OBJECT WHICH HAS
150 PRINT "BEEN SELECTED."
155 PRINT
160 PRINT "BE SURE TO END EACH QUESTION WITH A
170 PRINT "QUESTION MARK."
180 PRINT
190 PRINT
195 B$="AEIOUY"
200 C=0
210 REM *ROUND
220 C=C+1
230 REM *QUESTION
240 PRINT "ENTER QUESTIONS #";C
250 INPUT A$
260 IF RIGHT$(A$,1)="?" THEN 290
270 PRINT "THAT ISN'T A QUESTION.
275 PRINT "PLEASE ASK A QUESTION."
280 GOTO 230
290 YES=0:NO=1
300 FOR I=1 TO 6
310 IF MID$(A$,LEN(A$)-1,1)=MID$(B$,I,1) THEN YES=1:NO=0
320 NEXT I
330 FOR PAUSE=1 TO 50*RND(1):NEXT PAUSE
340 IF YES THEN PRINT "YES"
350 IF NO THEN PRINT "NO"
360 PRINT
370 IF C<20 THEN 210
380 PRINT "END OF TWENTY QUESTIONS."
390 PRINT "PRESS RETURN TO START AGAIN."
400 GET A$:IF A$="" THEN 400
410 RUN
420 END
READY.
```

Program 2: Atari Version

```
100 REM *20 QUESTIONS
105 DIM A$(40),B$(6)
110 PRINT "}WELCOME TO THE GAME OF TWENT
Y"
120 PRINT "QUESTIONS.  BY ASKING QUESTIO
NS WHICH"
130 PRINT "HAVE YES OR NO ANSWERS, TRY T
O GUESS"
140 PRINT "THE OBJECT WHICH HAS BEEN SEL
ECTED."
150 PRINT
160 PRINT "BE SURE TO END EACH QUESTION
WITH A"
170 PRINT "QUESTION MARK."
180 PRINT
190 PRINT
195 B$="AEIOUY"
200 C=0
210 REM *ROUND
220 C=C+1
230 REM *QUESTION
240 PRINT "ENTER QUESTION #";C
250 INPUT A$
260 IF A$(LEN(A$))="?" THEN 290
270 PRINT "THAT ISN'T A QUESTION.  PLEAS
E ASK A  QUESTION."
280 GOTO 230
290 YES=0:NO=1
300 FOR I=1 TO 6
310 IF A$(LEN(A$)-1,LEN(A$)-1)=B$(I,I) T
HEN YES=1:NO=0
320 NEXT I
330 FOR PAUSE=1 TO 50*RND(0):NEXT PAUSE
340 IF YES THEN PRINT "YES"
350 IF NO THEN PRINT "NO"
360 PRINT
370 IF C<20 THEN 210
380 PRINT "END OF TWENTY QUESTIONS."
390 PRINT "PRESS RETURN TO START AGAIN."

400 INPUT A$
410 RUN
420 END
```

©

NEW PRINTERS. NEW PERIPHERALS. SAME OLD RELIABLE QUALITY AND VALUE.

1982 will find more OEM's, businesses, dealers and personal computer users turning to MICROTEK than ever before.

TekWriter-1

80 Column Dot Matrix Printer
(Formerly BYTEWRITER-1)

The Tekwriter-1 printer is, dollar for dollar, the finest value in the industry. And we've proved it by comparing the Tekwriter-1 to the Epson MX-80. Our print speed is 14 lines per minute faster, our life expectancy is longer, the character sets are the same, and the interface, warranty and printhead replacement cost are all identical.* But the biggest difference is the price. The Tekwriter-1 is about $300 less.

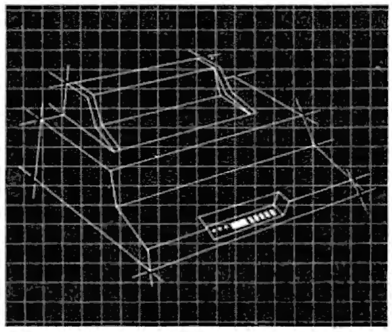
Our extensive testing has proved that the Tekwriter-1 interfaces problem-free to most parallel Centronics and serial (RS-232) computers.

The Tekwriter-1 is tough to beat for performance and quality.

*Data Source: Epson MX-80 Operation Manual

Parallel $349
Serial $389

TekWriter-2



NEW! 80/132 Column Dot Matrix Printer

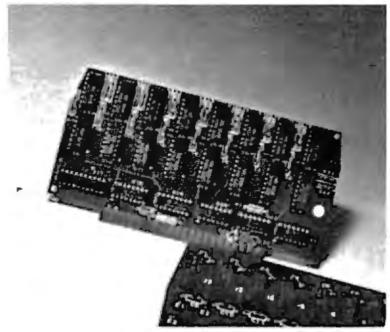
The Tekwriter-2 is perfectly suited to personal, business or OEM applications. Tekwriter-2 is designed to accept single sheet, roll or pin feed paper. It has a 9-wire dot matrix impact print head which produces crisp characters and has underlining capability. The printer is manufactured to run extremely quietly even while operating at peak output levels.

Tekwriter-2 is especially well suited to handle an abundance of text entry because of its data buffer expansion capability to 25K. This ability makes it an efficient graphics generator.

Parallel interface (Centronics type). Interfaces all models of TRS-80, Apple, and Atari 400/800, and most computers with Centronics printer interface.

$695

Peripherals



16K Apple Memory Board
Expands Apple II to 64K RAM Memory. Works with MICROSOFT Z-80 Softcard, Apple PASCAL and Visicalc software.

16K-32K Atari Memory Board
4116 RAM (200NS) Compatible with Atari 400/800

Parallel/Seriel Data Buffer Converter
Interfaces with most computers and printers on the market today.
Switch selectable parallel or serial input/parallel or serial output.
Data buffer 2K standard — expandable to 62K.
Serial I/O BAUD rates switch independently selectable.

Atari Parallel or Serial Printer Cables
Pre-tested. Centronics or (RS-232) compatible.

APPLE PARALLEL INTERFACE CARD

Quantity and OEM discounts available.

Continuing our quest for excellence.

MICROTEK inc.

MICROTEK
9514 Chesapeake Drive
San Diego, CA 92123
(714) 278-0633
Outside CA call
Toll Free (800) 854-1081
TWX. 910-335-1269

COMPUTE!'s Listing Conventions

Many of the programs which are listed in **COMPUTE!** use special keys (cursor control keys, color keys, etc.) To make it easy to tell *exactly* what should be typed in when copying a program into the computer, we have established the following listing conventions.

For The Atari

All the editing and cursor control characters are spelled out and surrounded by brackets in the program listings: {CLEAR} for "clear screen." Other characters, such as CTRL-T (the "ball" character) will be listed as the "normal" character, but it will be within brackets: {T}. A series of identical control characters will be indicated by a number within the brackets: {3DOWN} means type ESC CURSOR-DOWN three times; {12 R} would mean type CTRL-R twelve times. Remember to press the ESC (escape) key before each cursor control key. If you should see {ESC} itself in a program listing, you would press ESC *twice*.

Two of the control characters, {=} and {-}, should be shifted. Any reverse field text will be enclosed within vertical lines. (In other words, any time you see a vertical line within a program listing in **COMPUTE!**, press the Atari logo key {⅄}.)

Atari Conventions

```
{CLEAR}= SHIFT-< (Clear Screen)
{UP}= CTRL-minus (Cursor Up)
{DOWN}= CTRL-equals (Cursor Down)
{LEFT}= CTRL-plus (Cursor left)
{RIGHT}= CTRL-asterisk (Cursor right)
{BACK S}= BACK S (Back space)
{DELETE}= CTRL-DELETE (Delete character)

{DEL LINE}= SHIFT-DELETE (Delete Line)
{INSERT}= CTRL-INSERT (Insert character)

{INS LINE}= SHIFT-INSERT (Insert line)
{ESC}= ESC (ESCape key pressed twice)
{TAB}= TAB (Tab key)
{CLR TAB}= CTRL-TAB (Clear tab setting)
{SET TAB}= SHIFT-TAB (Set tab stop)
{BELL}= CTRL-2 (Ring buzzer)
```

For PET/CBM/VIC

Generally, any PET/CBM/VIC program listings will contain bracketed words which spell out any special characters: {DOWN} would mean to press the cursor-down key; {3DOWN} would mean to press the cursor-down key three times.

To indicate that a key should be *shifted* (hold down the SHIFT key while pressing the other key), the key would be underlined in our listing. For example, S would mean to type the S key while holding the shift key. This would result in the "heart" graphics symbol appearing on your screen.

Sometimes in a program listing, especially within quoted text when a line runs over into the next line, it is difficult to tell where the first line ends. How many times should you type the SPACE bar? In our convention, when a line breaks in this way, the ~ symbol shows exactly where it broke. For example:

```
100 PRINT "TO START THE GAME  ~
    YOU MAY HIT ANY OF THE KEYS
    ON YOUR KEYBOARD."
```

shows that the program's author intended for you to type two spaces after the word *GAME*.

For The Apple

Programs listed as "Microsoft" are written for the PET/CBM, Apple, OSI, etc. Although the programs are general in nature, you may need to make a few changes for them to run correctly on your Apple. Microsoft BASIC programs written for the PET/CBM sometimes contain special cursor control characters. The following table shows equivalent Apple words. Notice that these Apple commands are *outside* quotations (and even separate from a PRINT statement). PRINT"[RVS]YOU WON" becomes INVERSE: PRINT"YOU WON":NORMAL

[CLEAR[(Clear Screen) HOME

[HOME] (Home cursor) VTAB 0:HTAB 0

[DOWN] (Cursor down)
POKE 37,PEEK(37)+(PEEK(37)<23)

[UP] (Cursor up)
POKE 37,PEEK(37)-(PEEK(37)>0))

[LEFT] (Cursor left) PRINT CHR$(8);

[RIGHT] (Cursor right)
POKE 36,PEEK(36)+(PEEK(36)>(PEEK(32)
+PEEK(33)))

[RVS] (Inverse video on. Turns off automatically after a carriage return. To be safe, turn off inverse video after the print statement with NORMAL unless the PRINT statement ends with a semicolon.)
INVERSE

[OFF] (Inverse video off) NORMAL

Shifted characters can represent either graphics characters or uppercase letters. If within text, just use the non-shifted character, otherwise substitute a space. Some "generalized" programs contain a POKE such as POKE 59468,14. Omit these from the program when typing it in. One final note: you will probably want to insert a question mark or colon within an INPUT prompt. PET/CBM and many other BASICs automatically print a question mark:

INPUT "WHAT IS YOUR NAME";N$
becomes
INPUT "WHAT IS YOUR NAME?";N$

All Commodore Machines

Clear Screen	{CLEAR}	**Cursor Left**	{LEFT}
Home Cursor	{HOME}	**Insert Character**	{INST}
Cursor Up	{UP}	**Delete Character**	{DEL}
Cursor Down	{DOWN}	**Reverse Field On**	{RVS}
Cursor Right	{RIGHT}	**Reverse Field Off**	{OFF}

VIC Conventions

Set Color To Black	{BLK}	**Function Two**	{F2}
Set Color To White	{WHT}	**Function Three**	{F3}
Set Color To Red	{RED}	**Function Four**	{F4}
Set Color To Cyan	{CYN}	**Function Five**	{F5}
Set Color To Purple	{PUR}	**Function Six**	{F6}
Set Color To Green	{GRN}	**Function Seven**	{F7}
Set Color To Blue	{BLU}	**Function Eight**	{F8}
Set Color To Yellow	{YEL}	**Any Non-implemented Function**	{NIM}
Function One	{F1}		

8032/Fat 40 Conventions

Set Window Top	{SET TOP}	**Erase To Beginning**	{ERASE BEG}
Set Window Bottom	{SET BOT}	**Erase To End**	{ERASE END}
Scroll Up	{SCR UP}	**Toggle Tab**	{TGL TAB}
Scroll Down	{SCR DOWN}	**Tab**	{TAB}
Insert Line	{INST LINE}	**Escape Key**	{ESC}
Delete Line	{DEL LINE}		

Energy Workbook

David E. Pitts
Houston, TX

If your heating or cooling costs have increased by 30% or more last year like mine, then you are probably considering installing some energy saving measures such as: storm windows, a clock thermostat, more insulation, caulking, or weatherstripping. Since everyone's home is different and there is a wide diversity of climatic conditions in the United States, it is difficult to determine which of the many choices is the best investment. The program described here utilizes the characteristics of the house together with the estimated climate to determine a projected savings for the homeowner for a wide variety of energy improvements at locations within the contiguous 48 states. The homeowner may utilize this savings, together with the projected cost and the economic outlook, to determine if the payout period meets his criteria for a successful investment.

The energy workbook program allows for a wide variety of fuels for both heating and cooling: oil, natural gas, electricity, wood, liquid petroleum, gas, and coal. The savings due to installing storm windows, changing thermostat settings, caulking and weatherstripping, or adding ceiling or floor insulation are calculated for the homeowner. The required inputs are shown in Table 1. Repetitive calculations involving future energy cost can easily be made using the program, thus improving the homeowner's estimate of the accrued energy savings.

The program was written in Microsoft BASIC on an OSI 4PMF using simple I/O so that the program could be easily converted to other systems such as PET, Apple, and TRS-80. Atari owners will have to modify the program by adding PRINTs to the prompted INPUT statements, adding dimension statements for each string variable and changing the string concatenation and splitting as per page 39 of their user's guide. *[Atari owners: make the changes in the lines indicated in Program 2. — Ed.]* The program is based on an algorithm from the Federal Energy Administration (Reference 1) which divides the 48 contiguous states into climatic regions for cooling and heating for average housing, fuel, and climatic conditions. If the user's situation is unusual in terms of home construction, altitude, etc. additional advice from government offices or utility companies may be advisable.

String variables are used to read the table of states, cities, and heating (H) and cooling (C) factors. Commas are used for delimiters separating the states from the cities and their factors. Because of this, cities comprised of two or more words have had the interior blanks removed. Statements 20-45 decode the city and factors from the string B$(I). The heating zone and the cooling zone each range from zero to five, with five being the most severe winter climate and zero being the most severe summer climate. The heating and cooling zones are used to calculate a heating factor and a cooling factor. The fuel factors FH(I) and FC(I) are read from the data statements for the fuel chosen by the user and a heating index (HI) or cooling index (CI) is calculated by the product of the heating (or cooling) factor times the fuel factor times the price per fuel unit.

The fuel index (FI) is calculated by the sum of the heating index and cooling index. The annual heating fuel cost is taken from the total energy cost for the heating season times .85 to account for use of other uses of fuel (e.g. hot water heating). The annual cooling cost is calculated from total cooling season fuel cost times .6 to account for other use such as lighting. These ratios can be checked by determining average off season to average in season usage. The appropriate ratios should be used in statements 175 and 200. The ratios in my home were .56 and .88 for cooling and heating respectively, quite close to the Energy Administration's estimate.

Annual heating saving due to changing the thermostat setting is calculated from the product of the number of degrees turned down times the annual heating cost (HS) times a savings factor (Y) calculated in lines 240-247. Additional savings due to setting back the nightime temperature are calculated using a similar procedure, but with an added factor .3 (due to the reduced time the set back temperature is in effect). Cooling seasons savings are calculated from the annual cooling cost times .02 times the number of degrees the thermostat is turned up. The annual savings from caulking and weatherstripping is calculated from a draft factor times the total floor area times the fuel index. The draft factor is the sum of the factors for windows,

DYNACOMP

Quality software for:*

ATARI
PET/CBM
APPLE II Plus
OSBORNE-1
ALTAIR****
TRS-80 (Level II)**
NORTH STAR***
CP/M Disks/Diskettes
(see Availability box)

DYNACOMP OFFERS THE FOLLOWING

- Widest variety
- Guaranteed quality
- Fastest delivery
- Friendly customer service
- Free catalog
- 24 hour order phone

CARD GAMES

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

NEW **GIN RUMMY (Apple only)** Price: $18.95 Cassette/$22.95 Diskette
This is the best micro computer implementation of GIN RUMMY existing. The computer plays exceptionally well, and the HIRES graphics are superb. What else can be said?

POKER PARTY (Available for all computers) Price: $17.95 Cassette/$21.95 Diskette
POKER PARTY is a draw poker simulation based on the book, POKER, by Oswald Jacoby. This is the most comprehensive version available for microcomputers. The party consists of yourself and six other (computer) players. Each of these players (you will get to know them) has a different personality in the form of a varying propensity to bluff or fold under pressure. Practice with POKER PARTY before going to that expensive game tonight! Apple cassette and diskette versions require a 32 K (or larger) Apple II.

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

THOUGHT PROVOKERS

MANAGEMENT SIMULATOR (Atari, North Star, OSBORNE and CP/M only) Price: $19.95 Cassette $23.95 Diskette
This program is both an excellent teaching tool as well as a stimulating intellectual game. Based upon similar games played at graduate business schools, each player or team controls a company which manufacturers three products. Each player attempts to outperform his competitors by setting selling prices, production volumes, marketing and design expenditures etc. The most successful firm is the one with the highest stock price when the simulation ends.

FLIGHT SIMULATOR (Available for all computers) Price: $17.95 Cassette/$21.95 Diskette
A realistic and extensive mathematical simulation of take-off, flight and landing. The program utilizes aerodynamic equations and the characteristics of a real airfoil. You can practice instrument approaches and navigation using radials and compass headings. The more advanced flyer can also perform loops, half-rolls and similar aerobatic maneuvers. Although this program does not employ graphics, it is exciting and very addictive. See the software review in COMPUTRONICS. Runs in 16K Atari.

VALDEZ (Available for all computers) Price: $15.95 Cassette/$19.95 Diskette
VALDEZ is a computer simulation of supertanker navigation in the Prince William Sound/Valdez Narrows region of Alaska. Included in this simulation is a realistic and extensive 256 × 256 element map, portions of which may be viewed using the ship's alphanumeric radar display. The motion of the ship itself is accurately modelled mathematically. The simulation also contains a model for the tidal patterns in the region, as well as other traffic (outgoing tankers and drifting icebergs). Chart your course from the Gulf of Alaska to Valdez Harbor! See the software reviews in 80 Software Critique and Personal Computing.

BACKGAMMON 2.0 (Atari, North Star, OSBORNE and CP/M only) Price: $14.95 Cassette/$18.95 Diskette
This program tests your backgammon skills and will also improve your game. A human can compete against a computer or against another human. The computer can even play against itself. Either the human or the computer can double or generate dice rolls. Board positions can be created or saved for replay. BACKGAMMON 2.0 plays in accordance with the official rules of backgammon and is sure to provide many fascinating sessions of backgammon play.

CHESS MASTER (North Star and TRS-80 only) Price: $19.95 Cassette/$23.95 Diskette
This complete and very powerful program provides five levels of play. It includes castling, en passant captures and the promotion of pawns. Additionally, the board may be preset before the start of play, permitting the examination of "book" plays. To maximize execution speed, the program is written in assembly language (by SOFTWARE SPECIALISTS of California). Full graphics are employed in the TRS-80 version, and two widths of alphanumeric display are provided to accommodate North Star users. See review in onComputing.

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

FOREST FIRE! (Atari only) Price: $15.95 Cassette/$19.95 Diskette
Using excellent graphics and sound effects, this simulation puts you in the middle of a forest fire. Your job is to direct operations to put out the fire while compensating for changes in wind, weather and terrain. Not protecting valuable structures can result in startling penalties. Life-like variables are provided to make FOREST FIRE! very suspenseful and challenging. No two games have the same setting and there are 3 levels of difficulty.

BLACK HOLE (Apple only) Price: $14.95 Cassette/$18.95 Diskette
This is an exciting graphical simulation of the problems involved in closely observing a black hole with a space probe. The object is to enter and maintain, for a prescribed time, an orbit close to a small black hole. This is to be achieved without coming so near the anomaly that the tidal stress destroys the probe. Control of the craft is realistically simulated using side jets for rotation and main thrusters for acceleration. This program employs Hi-Res graphics and is educational as well as challenging.

NEW **SPACE EVACUATION! (Apple, Atari and TRS-80 only)** Price: $15.95 Cassette/$19.95 Diskette
Can you colonize the galaxy and evacuate the Earth before the sun explodes? Your computer becomes the ship's computer as you explore the universe to relocate millions of people. This simulation is particularly interesting as it combines many of the exciting elements of classic space games with the mystery challenge of ADVENTURE.

MONARCH (Atari only) Price: $11.95 Cassette/$15.95 Diskette
MONARCH is a fascinating economic simulation requiring you to survive an 8-year term as your nation's leader. You determine the amount of acreage devoted to industrial and agricultural use, how much food to distribute to the populace and how much should be spent on pollution control. You will find that all decisions involve a compromise and that it is not easy to make everyone happy. Runs in 16K Atari.

CHOMPELO (Atari only) Price: $11.95 Cassette/$15.95 Diskette
CHOMPELO is really two challenging games in one. One is similar to NIM; you must bite off part of a cookie, but avoid taking the poisoned portion. The other game is the popular board game REVERSI. It fully uses the Atari's graphics capability, and is hard to beat. This package will run on a 16K system.

SPACE LANES (Available for all computers) Price: $10.95 Cassette/$14.95 Diskette
SPACE LANES is a simple but exciting space transportation game which involves up to four players (including the computer). The object is to form and expand space transportation companies in a competitive environment. The goal is to amass more net worth than your opponent. The economics include stock purchases and company mergers. Watch your wealth grow!

AVAILABILITY

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

**ATARI, PET CBM, NORTH STAR, CP M, IBM, OSBORNE and XEROX are registered trademarks and/or trade marks.*

***Except where noted, all TRS-80 Model I software is available on cassette (only) for the TRS-80 Model III. Exceptions VALDEZ, CRIBBAGE, GRAFIX, CHESSMASTER. TRS-80 diskettes are not supplied with either DOS or BASIC.*

****For most North Star disk-based systems. DYNACOMP presently does not support the new North Star Advantage.*

*****For Altair systems having Microsoft BASIC.*

AND MORE...

STARTREK 3.2 (Available for all computers) Price: $11.95 Cassette/$15.95 Diskette
This is the classic Startrek simulation, but with several new features. For example, the Klingons now shoot at the Enterprise without warning while also attacking starbases in other quadrants. The Klingons also attack with both light and heavy cruisers and move when shot at! The situation is hectic when the Enterprise is besieged by three heavy cruisers and a starbase S.O.S. is received! The Klingons get even! See the software reviews in A.N.A.L.O.G., 80 Software Critique and Game Merchandising.

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

SPACE TILT (Apple and Atari only) Price: $10.95 Cassette/$14.95 Diskette
Use the game paddles to tilt the plane of the TV screen to "roll" a ball into a hole in the screen. Sound simple? Not when the hole gets smaller and smaller! A built-in timer allows you to measure your skill against others in this habit-forming action game.

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

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

THE RINGS OF THE EMPIRE (Atari only) Price: $15.95 Cassette/$19.95 Diskette
The empire has developed a new battle station protected by rotating rings of energy. Each time you blast through the rings and destroy the station, the empire develops a new station with more protective rings. This exciting game runs on 16K systems, employs extensive graphics and sound and can be played by one or two players.

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

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

TRIPLE BLOCKADE (Atari only) Price: $14.95 Cassette/$18.95 Diskette
TRIPLE BLOCKADE is a two-to-three player graphics and sound action game. It is based on the classic video arcade game which millions have enjoyed. Using the Atari joysticks, the object is to direct your blockading line around the screen without running into your opponent(s). Although the concept is simple, the combined graphics and sound effect lead to "high anxiety".

GAMES PACK I (Available for all computers) Price: $10.95 Cassette/$14.95 Diskette
GAMES PACK I contains the classic computer games of BLACKJACK, LUNAR LANDER, CRAPS, HORSERACE, SWITCH and more. These games have been combined into one large program for ease in loading. They are individually accessed by a convenient menu. This collection is worth the price just for the DYNACOMP version of BLACKJACK.

GAMES PACK II (Available for all computers) Price: $10.95 Cassette/$14.95 Diskette
GAMES PACK II includes the games CRAZY EIGHTS, JOTTO, ACEY-DUCEY, LIFE, WUMPUS and others. As with GAMES PACK I, all the games are loaded as one program and are called from a menu. You will particularly enjoy DYNACOMP's version of CRAZY EIGHTS.

Why pay $7.95 or more per program when you can buy a DYNACOMP collection for just $10.95?

MOON PROBE (Atari and North Star only) Price: $11.95 Cassette/$15.95 Diskette
This is an extremely challenging "lunar lander" program. The user must drop from orbit to land at a predetermined target on the moon's surface. You control the thrust and orientation of your craft plus direct the rate of descent and approach angle. Runs in 16K Atari.

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

NEW **CHIRP INVADERS (PET/CBM only)** Price: $14.95 Cassette/$18.95 Diskette
CHIRP INVADERS is an addictive game using action graphics. A Federation space station must be reached before the Chirps conquer the Earth. Stationary obstacles, moving meteors, and the attacking Chirps must all be avoided for a successful journey. Good luck.

ADVENTURE

CRANSTON MANOR ADVENTURE (North Star and CP/M only) Price: $19.95 Diskette
At last! A comprehensive Adventure game for North Star and CP/M systems. CRANSTON MANOR ADVENTURE takes you into mysterious CRANSTON MANOR where you attempt to gather fabulous treasures. Lurking in the manor are wild animals and robots who will not give up the treasures without a fight. The number of rooms is greater and the associated descriptions are much more elaborate than the current popular series of Adventure programs, making this game the top in its class. Play can be stopped at any time and the status stored on diskette. Not available in 5¼" CP/M format.

GUMBALL RALLY ADVENTURE (North Star only, 48K) Price: $21.95 Diskette
Take part in this outlaw race from the east coast to the west coast. The goal is to find your way to the finish line while maintaining the highest possible speed. You may choose one of five cars available at the garage. The choice will affect your speed and range. Remember to take spare parts and don't get caught speeding!

NEW **UNCLE HARRY'S WILL (North Star only, 40K)** Price: $24.95 Diskette
Uncle Harry has died and has left you everything. However, he has neglected to mention where everything is! Instead, his will consists of a poem which contains clues. You will have to travel all over the United States both by car and on foot to solve the puzzle, and there are over 300 locations to probe. Be careful and watch out for red herrings!

SPEECH SYNTHESIS

DYNACOMP is now distributing the new and revolutionary TYPE-'N-TALK™ (TNT) speech synthesizer from Votrax. Simply connect TNT to your computer's serial interface, enter text from the keyboard and hear the words spoken. TNT is the easiest-to-program speech synthesizer on the market. It uses the least amount of memory and provides the most flexible vocabulary available anywhere!

List price $375. DYNACOMP'S price $319.95 plus $5.00 for shipping and handling

TALK TO ME (T'N'T Atari only, 24K) Price: $14.95 Cassette/$18.95 Diskette
This program presents a superb tutorial on speech synthesis using the Atari 800 and TYPE 'N TALK™. TALK TO ME will illustrate normal word generation as well as phoneme generation. The documentation includes many helpful programming tips. TALK TO ME has been demonstrated on network (CBS) TV!

MISCELLANEOUS

CRYSTALS (Atari only) Price: $ 9.95 Cassette/$13.95 Diskette
A unique algorithm randomly produces fascinating graphics displays accompanied with tones which vary as the patterns are built. No two patterns are the same, and the combined effect of the sound and graphics are mesmerizing. CRYSTALS has been used in local stores to demonstrate the sound and color features of the Atari. Runs in 16K Atari.

NORTH STAR SOFTWARE EXCHANGE (NSSE) LIBRARY
DYNACOMP now distributes the 23 volume NSSE library. These diskettes each contain many programs and offer an outstanding value for the purchase price. They should be part of every North Star user's collection. Call or write DYNACOMP for details regarding the contents of the NSSE collection.

Price: $9.95 each/$7.95 each (4 or more)
The complete collection may be purchased for $149.95

BUSINESS and UTILITIES

NEW

MAILMASTER (Atari diskette only) — Price: $39.95 Diskette

MAILMASTER is a very versatile software package for managing and manipulating mail lists and mini data bases. Each disk can hold over 700 customer entries containing name, address, three 3-letter key words and a phone number. The display is marked so that entries may be made and edited with ease. The status (e.g., disk space left, options, etc.) is shown at all times. Labels may be printed 1,2 or 3 up, and all sorting (zip code and alphabetic) is performed by a fast machine language program.

PERSONAL FINANCE SYSTEM (Atari and North Star only) — Price: $39.95 Diskette

PFS is a single diskette, menu-oriented system composed of ten different programs. Besides recording your expenses and tax deductible items, PFS will sort and summarize expenses by payee, and display information on expenditures by any of 26 user defined codes by month or by payee. PFS will even produce monthly bar graphs of your expenses by category! This powerful package requires only one disk drive, minimal memory (24K Atari, 32K North Star) and will store up to 600 records per disk (and over 1000 records per disk by making a few simple changes to the programs). You can record checks plus cash expenses so that you can finally see where your money goes and eliminate guesswork and tedious hand calculations. Contains high speed machine language sort. PFS has been demonstrated on network (CBS) TV!

FAMILY BUDGET (Apple and Atari only) — Price: $34.95 Diskette

FAMILY BUDGET is a very convenient financial record-keeping program. You will be able to keep track of cash and credit expenditures as well as income on a daily basis. You can record tax deductible items and charitable donations. FAMILY BUDGET also provides a continuous record of all credit transactions. You can make daily cash and charge entries to any of 21 different expense accounts as well as to 5 payroll and tax accounts. Data are easily retrieved giving the user complete control over an otherwise complicated (and unorganized!) subject.

INTELINK (Atari only) — Price: $49.95 Diskette

This software package contains a menu-driven collection of programs for facilitating efficient two-way communications through a full duplex modem (required for use). In one mode of operation you may connect to a data service (e.g., The SOURCE or MicroNet) and quickly load data such as stock quotations onto your diskette for later viewing. This greatly reduces "connect time" and thus the service charge. You may also record the complete contents of a communications session. Additionally, programs written in BASIC, FORTRAN, etc. may be built off-line using the support text editor and later "uploaded" to another computer, making the Atari a very smart terminal. Even Atari BASIC programs may be uploaded. Further, a command file may be built off-line and used later as controlling input for a time-share system. That is, you can set up your sequence of time-share commands and programs, and the Atari will transmit them as needed; batch processing. All this adds up to saving both connect time and your time.

TEXT EDITOR II (CP/M) — Price: $29.95 Diskette/$33.45 Disk

This is the second release version of DYNACOMP's popular TEXT EDITOR I and contains many new features. With TEXT EDITOR II you may build text files in chunks and assemble them for later display. Blocks of text may be appended, inserted or deleted. Files may be saved on disk/diskette in right justified/centered format to be later printed by either TEXT EDITOR II or the CP/M ED facility. Futher, ASCII CP/M files (including BASIC and assembly language programs) may be read by the editor and processed. In fact, text files can be built using ED and later formatted using TEXT EDITOR II. All in all, TEXT EDITOR II is an inexpensive, easy to use, but very flexible editing system.

NEW

PAYFIVE (Apple II plus diskette, two drives required) — Price: $149.95

This is an enormously flexible employee payroll system with extraordinarily good human engineering features. PAYFIVE prints checks and compiles the required federal, state and local forms for up to 148 employees. The pay methods may be hourly, salary, commission or any combination. There are multiple options for pay periods, and they also can be used in any combination. PAYFIVE includes *many* other features and comes extremely well documented with a 200 page manual. The manual may be purchased separately for $30, and that payment later applied to the software purchase.

SHOPPING LIST (Atari only) — Price: $12.95 Cassette/$16.95 Diskette

SHOPPING LIST stores information on items you purchase at the supermarket. Before going shopping, it will remind you of all the things you might need, and then display (or optionally print) your shopping list and the total cost. Adding, deleting, changing and storing data is very easy. Runs with 16K.

TAX OPTIMIZER (North Star only) — Price: $39.95 Diskette

The TAX OPTIMIZER is an easy-to-use, menu oriented software package which provides a convenient means for analyzing various income tax strategies. The program is designed to provide a quick and easy data entry. Income tax is computed by all tax methods (regular, income averaging, maximum and alternate minimum tax). The user may immediately observe the tax effect of critical financial decisions. TAX OPTIMIZER has been thoroughly field tested in CPA offices and comes complete with the current tax tables in its data files. TAX OPTIMIZER is tax deductible!

NEW

UTIL (Apple only, 48K) — Price: $19.95 Diskette

UTIL is a disk-oriented utility system which permits examining and changing of the contents of DOS 3.2 and 3.3 diskettes at the bit (nibble or byte) level. With UTIL you can easily examine the contents of a diskette sector by sector, restructure the sector pointers, reallocate sectors (e.g. bad sectors may be "hidden"), and perform many other sophisticated operations. For the experienced programmer.

NEW

TURNKEY AND MENU (Atari only) — Price: $17.95 Diskette

TURNKEY is a utility program which allows you to create autoboot/autorun diskettes easily. Simply load and run TURNKEY, load the program diskette to be modified, and answer the questions! The TURNKEY diskette also comes with DOS 2.0 and includes another program, MENU. MENU lists the contents of your diskette alphabetically, and permits the running of any BASIC program on the diskette by typing a single key. TURNKEY and MENU provide you with the ability to run any program on your diskette by simply turning on the computer and pressing a single key.

NEW

STOCKAID (Atari only) — Price: $29.95 Diskette

STOCKAID provides a powerful set of tools for stock market analysis. With STOCKAID you can display point and figure charts, as well as bar charts with oscillators. You can also examine long term moving averages and on-balance volume features. STOCKAID allows you to input daily data with a single diskette storage capability of 239 days × 16 stocks. Included are stock dividend and split adjustment capabilities. A very professional package!

NEW

SHAPE MAGICIAN (Apple II, 48K, diskette only) — Price: $29.95

At last! An utility for painlessly creating graphics shapes for the Apple. Create, edit and save up to 30 shapes which can then be used to develop arcade games or to simply enhance your programs. Add that professional touch!

EDUCATION

HODGE PODGE (Apple only, 48K Applesoft or Integer BASIC) — Price: $14.95 Cassette/$18.95 Diskette

Let HODGE PODGE be your child's teacher. Pressing any key on your Apple will result in a different and intriguing "happening" related to the letter or number of the chosen key. The program's graphics, color and sound are a delight for children from ages 1½ to 7. HODGE PODGE is a non-intimidating teaching device which brings a new dimension to the use of computers in education. See the excellent reviews of this very popular program in INFOWORLD and SOFTALK.

NEW

TEACHER'S AIDE (Atari only) — Price: $13.95 Cassette/$17.95 Diskette

TEACHER'S AIDE consists of three basic modules contained in one program. The first module provides addition and subtraction exercises of varying levels of difficulty. The second module consists of multiplication problems in which the student may be tested both on the final answer and/or on the subtotal answers in the long hand procedure. Several levels of complexity are provided here as well. The third module consists of division problems; one particularly nice feature of the division module is that the long hand division steps can be displayed along with the remainder in order to clearly demonstrate the procedure by which the remainder is derived. Using TEACHER'S AIDE is not merely a drill, but rather a learning experience.

NEW

PHARMACOLOGY UPDATE (PET only) — Price: $149.95 Cassette/$153.95 Diskette

This is DYNACOMP's first educational software entry for the medical profession (more are coming!) PHARMACOLOGY UPDATE was written by a R.N. as a masters project, with the aid of a practicing pharmacologist and an electronics instructor. This package comes in two parts. The first part is a 200 page manual which is divided into 10 sections. Each of these sections provides both concise information and probing questions. The second part consists of 10 programs that are keyed to the text and which test the degree of your understanding of the text material. This package has great educational value for the beginning student as well as the professional interested in an efficient way to review and update his or her knowledge.

ORDERING INFORMATION

All orders are processed and shipped within 48 hours. Please enclose payment with order and include the appropriate computer information. If paying by VISA or Master Card, include all numbers on card. Purchase orders accepted.

Shipping and Handling Charges
Within North America: Add $2.00
Outside North America: Add 15% (Air Mail)

Delivery
All orders (excluding books) are sent First Class

Quantity Discounts
Deduct 10% when ordering 3 or more programs. Dealer discount schedules are available upon request

8" CP/M Disks
Add $2.50 to the listed diskette price for each 8" floppy disk (IBM soft sectored CP/M format). Programs run under Microsoft MBASIC or BASIC-80

5¼" CP/M Disks
All software available on 8" CP/M disks is also available on 5¼" disks, North Star and Osborne format.

Ask for DYNACOMP programs at your local software dealer. Write for detailed descriptions of these and other programs from DYNACOMP.

DYNACOMP, Inc. (Dept. E)
1427 Monroe Avenue
Rochester, New York 14618
24 hour order phone: (716)442-8731 recording
Office phone (9AM-5PM EST): (716)442-8960

VISA

MasterCard

New York State residents please add 7% NYS sales tax.

STATISTICS and ENGINEERING

DIGITAL FILTER (Available for all computers) — Price: $39.95 Cassette/$43.95 Diskette

DIGITAL FILTER is a comprehensive data processing program which permits the user to design his own filter function or choose from a menu of filter forms. The filter forms are subsequently converted into non-recursive convolution coefficients which permit rapid data processing. In the explicit design mode the shape of the frequency transfer function is specified by directly entering points along the desired filter curve. In the menu mode, ideal low pass, high pass and bandpass filters may be approximated to varying degrees according to the number of points used in the calculation. These filters may optionally also be smoothed with a Hamming function. In addition, multi-stage Butterworth filters may be selected. Features of DIGITAL FILTER include plotting of the data before and after filtering, as well as display of the chosen filter functions. Also included are convenient data storage, retrieval and editing procedures.

DATA SMOOTHER (Not available for Atari) — Price: $19.95 Cassette/$23.95 Diskette

This special data smoothing program may be used to rapidly derive useful information from noisy business and engineering data which are equally spaced. The software features choice in degree and range of fit, as well as smoothed first and second derivative calculation. Also included is automatic plotting of the input data and smoothed results.

FOURIER ANALYZER (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

Use this program to examine the frequency spectra of limited duration signals. The program features automatic scaling and plotting of the input data and results. Practical applications include the analysis of complicated patterns in such fields as electronics, communications and business.

TFA (Transfer Function Analyzer) — Price: $19.95 Cassette/$23.95 Diskette

This is a special software package which may be used to evaluate the transfer functions of systems such as hi-fi amplifiers and filters by examining their response to pulsed inputs. TFA is a major modification of FOURIER ANALYZER and contains an engineering-oriented decibel versus log-frequency plot as well as data editing features. Whereas FOURIER ANALYZER is designed for educational and scientific use, TFA is an engineering tool. Available for all computers.

HARMONIC ANALYZER (Available for all computers) — Price: $24.95 Cassette/$28.95 Diskette

HARMONIC ANALYZER was designed for the spectrum analysis of repetitive waveforms. Features include data file generation, editing and storage/retrieval as well as data and spectrum plotting. One particularly unique facility is that the input data need not be equally spaced or in order. The original data is sorted and a cubic spline interpolation is used to create the data file required by the FFT algorithm.

FOURIER ANALYZER, TFA and HARMONIC ANALYZER may be purchased together for a combined price of $49.95 (three cassettes) and $59.95 (three diskettes).

REGRESSION I (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

REGRESSION I is a unique and exceptionally versatile one-dimensional least squares "polynomial" curve fitting program. Features include very high accuracy; an automatic degree determination option; an extensive internal library of fitting functions; data editing; automatic data, curve and residual plotting; a statistical analysis (eg: standard deviation, correlation coefficient, etc.) and much more. In addition, new fits may be tried without reentering the data. REGRESSION I is certainly the cornerstone program in any data analysis software library.

REGRESSION II (PARAFIT) (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

PARAFIT is designed to handle those cases in which the parameters are imbedded (possibly nonlinearly) in the fitting function. The user simply inserts the functional form, including the parameters (A(1), A(2), etc.) as one or more BASIC statement lines. Data, results and residuals may be manipulated and plotted as with REGRESSION I. Use REGRESSION I for polynomial fitting, and PARAFIT for those complicated functions.

MULTILINEAR REGRESSION (MLR) (Available for all computers) — Price: $24.95 Cassette/$28.95 Diskette

MLR is a professional software package for analyzing data sets containing two or more linearly independent variables. Besides performing the basic regression calculation, this program also provides easy to use data entry, storage, retrieval and editing functions. In addition, the user may interrogate the solution by supplying values for the independent variables. The number of variables and data size is limited only by the available memory.

REGRESSION I, II and MULTILINEAR REGRESSION may be purchased together for $51.95 (three cassettes) or $63.95 (three diskettes).

ANOVA (Not available on Atari cassette or for PET/CBM) — Price: $39.95 Cassette/$43.95 Diskette

In the past the ANOVA (analysis of variance) procedure has been limited to the large mainframe computers. Now DYNACOMP has brought the power of this method to small systems. For those conversant with ANOVA, the DYNACOMP software package includes the 1-way, 2-way and N-way procedures. Also provided are the Yates 2^{K-P} factorial designs. For those unfamiliar with ANOVA, do not worry. The accompanying documentation was written in a tutorial fashion (by a professor in the subject) and serves as an excellent introduction to the subject. Accompanying ANOVA is a support program for building the data base. Included are several convenient features including data editing, deleting and appending.

BASIC SCIENTIFIC SUBROUTINES, Volumes 1 and 2 (Not available for Atari)

DYNACOMP is the exclusive distributor for the software keyed to the popular texts *BASIC SCIENTIFIC SUBROUTINES, Volumes 1 and 2* by F. Ruckdeschel (see advertisements in BYTE magazine). These subroutines have been assembled according to chapter. Included with each collection is a menu program which selects and demonstrates each subroutine.

Volume 1
Collection #1 Chapters 2 and 3 - Data and function plotting; complex variables and functions.
Collection #2 Chapter 4 - Extended matrix and vector operations.
Collection #3 Chapters 5 and 6 - Random number generators (Poisson, Gaussian, etc.); series approximations
Price per collection: $14.95 Cassette/$18.95 Diskette

All three collections are available for $39.95 (three cassettes) and $49.95 (three diskettes)

Volume 2
Collection #1 Chapter 1 - Linear, polynomial, multidimensional, parametric least squares
Collection #2 Chapter 2 - Series approximation techniques (economization, inversion, reversion, shifting, etc.)
Collection #3 Chapter 3 - Functional approximations by iteration and recursion.
Collection #4 Chapter 4 - CORDIC approximations to trigonometric, hyperbolic, exponential and logarithmic functions.
Collection #5 Chapter 5 - Table interpolation, differentiation and integration (Newton, LaGrange, splines)
Collection #6 Chapter 6 - Methods for finding the real roots of functions
Collection #7 Chapter 7 - Methods for finding the complex roots of functions.
Collection #8 Chapter 8 - Optimization by steepest descent.
Price per collection: $14.95 Cassette/$18.95 Diskette

All eight collections are available for $99.95 (eight cassettes) and $129.95 (eight diskettes)

Because the texts are a vital part of the documentation, *BASIC SCIENTIFIC SUBROUTINES, Volumes 1 and 2* are available from DYNACOMP

BASIC SCIENTIFIC SUBROUTINES, Vol 1 (319 pages): $19.95 + 75¢ postage
BASIC SCIENTIFIC SUBROUTINES, Vol 2 (790 pages): $21.95 + $1.50 postage

See reviews in KILOBAUD and Dr. Dobbs.

NEW

SOFTNET (Apple II 48K, diskette only) — Price $129.95

SOFTNET may be used to create models of liquid pipeline systems to evaluate their flow performance. Up to 150 nodes with up to 150 connecting elements may be simulated, and models may be combined to form yet larger models. If you are involved in water distribution systems, chemical fluid flow problems, building plumbing, or similar situations, this is an ideal analysis tool.

ACTIVE CIRCUIT ANALYSIS (ACAP) (48K Apple only) — Price: $25.95 Cassette/$29.95 Diskette

ACAP is the analog circuit designer's answer to LOGIC SIMULATOR. With ACAP you may analyze the response of an active or passive component circuit (e.g., a transistor amplifier, band pass filter, etc.). The circuit may be probed at equal steps in frequency, and the resulting complex (i.e., real and imaginary) voltages at each component juncture examined. By plotting the magnitude of these voltages, the frequency response of a filter or amplifier may be completely determined with respect to both amplitude and phase. In addition, ACAP prints a statistical analysis of the range of voltage responses which result from tolerance variations in the components. ACAP is easy to learn and use. Simply describe the circuit in terms of the elements and their placement, and execute. Circuit descriptions may be saved onto cassette or diskette to be recalled at a later time for execution or editing. ACAP should be part of every circuit designer's program library.

LOGIC SIMULATOR (Apple only; 48K RAM) — Price: $24.95 Cassette/$28.95 Diskette

With LOGIC SIMULATOR you may easily test your complicated digital logic design with respect to given set of inputs to determine how well the circuit will operate. The elements which may be simulated include multiple input AND, OR, NOR, EXOR, EXNOR and NAND gates, as well as inverters, J-K and D flip-flops, and one-shots. The response of the system is available every clock cycle. Inputs may be clocked in with varying clock cycle lengths/displacements and delays may be introduced to probe for glitches and race conditions. At the user's option, a timing diagram for any given set of nodes may be plotted using HIRES graphics. Save your breadboarding until the circuit is checked by LOGIC SIMULATOR.

NEW

NUMBERKRUNCHER (TRS-80 only) — Price: $69.95 Cassette/$73.95 Diskette

This program is the most complete numerical analysis system available for the TRS-80. It can handle up to 255 data sets, each set having a six character name. It includes complete data editing facilities and convenient data input/output capability. The analyses available are multiple linear regression and correlation determination of residuals, data transformations and extensive graphics generation, including axis naming, and more. The supporting documentation is extremely well written and well organized, and includes appendices which describe the numerical procedures used in the program.

NEW

STATSORT (TRS-80 only) — Price: $39.95 Cassette/$43.95 Diskette

STATSORT consists of several menu selected programs which allow the user to create (build, edit, merge), format and print files, (machine) sort them on any field, and numerically analyze (maximum, minimum, average, variance, standard deviation) tabulated data. STATSORT is well documented and easy to use. The cassette version can also be employed to create a data type which can be read by the Radio Shack Advanced Statistical Package.

NEW

STATTEST (TRS-80 only) — Price: $19.95 Cassette/$23.95 Diskette

This is a statistical inference package which helps you make wise decisions in the face of uncertainty. In an interactive fashion you can build and edit data files and test the differences in means, variances and proportions. STATTEST will also perform data analysis as well as do linear correlation and regression. This menu-directed statistical workhorse is rounded out with a chi-square contingency test and a (uniform and normal) random sample generator. The documentation is written by a college professor who guides you through the various tests.

ABOUT DYNACOMP

DYNACOMP is a leading distributor of small system software with sales spanning the world (currently in excess of 50 countries). During the past three years we have greatly enlarged the DYNACOMP product line, but have maintained and improved our high level of quality and customer support. The achievement in quality is apparent from our many repeat customers and the software reviews in such publications as COMPUTRONICS, 80 Software Critique, A.N.A.L.O.G., Softalk, Creative Computing and Kilobaud. DYNACOMP software has also been chosen for demonstration on network television. Our customer support is as close as your phone. It is always friendly. The staff is highly trained and always willing to discuss products or give advice.

doors, and general house condition each of which ranges from 0 to .02 in steps of .01 ranging from a tight fit (0) to drafty (lines 250-420).

Annual savings from storm windows are calculated from the product of the single glass area, the factor 0.65 and the fuel index. The annual savings from adding ceiling insulation is computed from the product of the ceiling savings index (X-I), the first floor area, and the fuel index (line 470). The ceiling savings index is calculated as the difference between the conduction factor between the recommended ceiling insulation and the existing ceiling insulation. The conduction factor subroutine is located between lines 828 and 920. The recommended ceiling insulation is calculated in lines 450-460 and is only a function of the heating zone, thereby underestimatinng the savings accrued due to reducing air conditioning cost. The annual savings from floor insulation are calculated (lines 625-630) by the product of the floor factor (J), the floor savings index, the floor area, and the fuel index.

References

Anon, 1977: Home Energy Saver's Workbook, FEA/D-77/117, U.S. Government Printing Office, Washington, DC.

Table 1.

Items needed for Energy Workbook are as follows:

1) state
2) city
3) Heating fuel cost (e.g. .37 cents/cu. ft.)
4) Cooling fuel cost (e.g. 5.14 cents/KWH)
5) Square ft. of single glass windows in house
6) Annual heating fuel cost
7) Annual cooling fuel cost
8) Check leakage around windows and doors with candle or cigarette
9) Floor area of house – sq. ft.
10) Ceiling R value, use following table:

R-Values for Various Thickness of Insulation

	BATTS OR BLANKETS		LOOSE FILL (POURED-IN)			
	glass fiber	rock wool	glass fiber	rock wool	cellulosic fiber	
R-11	3½"-4"	3"	5"	4"	3"	R-11
R-13	4"	4½"	6"	4½"	3½"	R-13
R-19	6"-6½"	5¼"	8"-9"	6"-7"	5"	R-19
R-22	6½"	6"	10"	7"-8"	6"	R-22
R-26	8"	8½"	12"	9"	7"-7½"	R-26
R-30	9½"-10½"	9"	13"-14"	10"-11"	8"	R-30
R-33	11"	10"	15"	11"-12"	9"	R-33
R-38	12"-13"	10½"	17"-18"	13"-14"	10"-11"	R-33

11) First floor area – sq. ft.
12) Floor R value if basement is unheated or house on pillars.

Program 1: Microsoft Version (Apple, OSI, PET)

```
1 REM ENERGY WORKBOOK IS BASED ON FEA/D-77/117, APRIL 1977
2 L=96:REM DAVID PITTS 16011 STONEHAVEN DR. HOUSTON TX 77059
3 FORI=1TO25:PRINT:NEXT:::PRINTTAB(25);"ENERGY WORKBOOK":PRINT:PRINT:PRINT:PRIN
4 PRINT"ITEMS NEEDED FOR ENERGY WORKBOOK":PRINT:PRINT"1) STATE":PRINT"2) CITY"
5 PRINT"3) HEATING FUEL COST (E.G. .37 CENTS/CU FT)"
6 PRINT"4) COOLING FUEL COST (E.G. 5.14 CENTS/KWH)"
7 PRINT"5) SQUARE FT OF SINGLE GLASS WINDOWS IN HOUSE"
8 PRINT"7) ANNUAL HEATING AND COOLING FUEL COST"
9 PRINT"8) CHECK FOR LEAKAGE AROUND WINDOWS AND DOORS WITH CANDLE"
10 PRINT"9) FLOOR AREA OF HOUSE - SQ FT"
11 PRINT"10) CEILING R VALUE - USE TABLE PROVIDED WITH INSTRUCTIONS"
12 PRINT"11) FIRST FLOOR AREA - SQ FT"
13 PRINT"12) FLOOR R VALUE IF BASEMENT IS UNHEATED OR HOUSE IS ON PILLARS"
14 PRINT:PRINT:PRINT:INPUT"STATE(DON'T ABBREVIATE)";B$:FORI=1TOL:READC$
15 IFLEFT$(B$,7)=LEFT$(C$,7)THENB$=C$:READD$
20 NEXT:FORI=1TO4:B$(I)="":NEXT:I=1:Y=LEN(D$):J=1
25 X=ASC(MID$(D$,I,1)):IFX=32THEN45
30 B$(J)=B$(J)+CHR$(X):GOTO55
45 I=I+1:X(J)=VAL(MID$(D$,I,1)):I=I+2:Y(J)=VAL(MID$(D$,I,1))
47 I=I+1:J=J+1
55 IFI<YTHENI=I+1:GOTO25
58 J=J-1:PRINT:PRINT:PRINT:FORI=1TOJ
60 PRINTTAB(15);I;TAB(20);B$(I);TAB(35);B$:NEXT
65 PRINT:PRINT:INPUT"CHOOSE # FOR NEAREST CITY";I:H=X(I):C=Y(I):REM ZONES
80 X=1:FORI=1TO5:IFC=ITHEN90
85 X=X-.25:NEXT
90 HC=X:X=0:FORI=0TO5:IFH=ITHEN100:HF & CF ARE HEAT & COOL FACTORS
95 X=X+.5:NEXT
100 HF=X:PRINT:PRINT
110 FORI=1TO8:READB$,FH(I),FC(I):PRINTSPC(15);I;B$:NEXT
115 PRINT:PRINT:INPUT"CHOOSE # FOR HEATING FUEL";J
```

```
120 INPUT"COST PER UNIT FOR HEATING FUEL(CENTS)";S:S=S/100
125 HI=S*FH(J)*HF:REM HEAT INDEX
126 PRINT:PRINT:INPUT"CHOOSE # FOR COOLING FUEL";J
127 INPUT"COST PER UNIT FOR COOLING FUEL(CENTS)";S:S=S/100
130 CI=S*FC(J)*HC:FI=HI+CI:REM COOL AND FUEL INDEX
135 PRINT:PRINT:PRINT:PRINT
145 PRINT"INPUT # OF SQUARE FT OF SINGLE GLASS WINDOWS, DO NOT"
150 INPUT"COUNT STORM WINDOWS OR SLIDING GLASS DOORS";X
170 X=INT(X*100*FI*.65)/100
175 PRINT"ANNUAL SAVINGS DUE TO STORM WINDOWS= $";X:X=.85:GOSUB800
180 PRINT:PRINT"IS HEATING FUEL USED FOR OTHER PURPOSES, E.G. COOKING ";
190 INPUTB$:IFASC(B$)=78THENX=1
200 INPUT"ANNUAL HEATING FUEL COST (DOLLARS)";HS:HS=HS*X:PRINT:PRINT:X=.6
210 PRINT"IS COOLING FUEL USED FOR OTHER PURPOSES, E.G. LIGHTING";
220 INPUTB$:IFASC(B$)=78THENX=1
230 INPUT"ANNUAL COOLING FUEL COST (DOLLARS)";CS:CS=CS*X
240 PRINT:PRINT:Y=.05:FORI=1TO3:IFI=HTHEN247
245 Y=Y-.01:NEXT:IFH=4THENY=.025
246 IFH=5THENY=.02
247 GOSUB800
250 PRINT"THE FOLLOWING SECTION EVALUATES THE SAVINGS OBTAINED BY TURNING"
255 PRINT"THE THERMOSTAT DOWN IN WINTER OR UP IN SUMMER FROM THE SETTING"
256 PRINT"YOU HAVE BEEN USING.":PRINT:PRINT:PRINT"HEATING":PRINT
260 INPUT"DEGREES TURNED DOWN DURING DAY";X:S=INT(100*Y*HS*X)/100
265 PRINT"SAVINGS =$";S:PRINT"ADDITIONAL DEGREES TURNED DOWN DURING NIGHT";
270 INPUTX:I=INT(100*Y*HS*X*.3)/100:PRINT"SAVINGS=$";I
280 S=S+I:PRINT"ANNUAL TOTAL HEATING SAVINGS =$";S:PRINT:PRINT"COOLING":PRINT
285 INPUT"DEGREES THERMOSTAT TURNED UP DURING COOLING";X
290 I=INT(100*CS*X*.02)/100:PRINT"SAVINGS =$";I
300 PRINT:PRINT"TOTAL ANNUAL SAVINGS =$";S+I:GOSUB800
310 PRINT:PRINT:PRINT"ANNUAL SAVINGS FROM CAULKING AND WEATHERSTRIPPING"
315 PRINT"CHECK DRAFTS HOLDING CANDLE NEAR CRACK ON WINDY DAY"
320 PRINT"CHOOSE ONE OF FOLLOWING":PRINT"     1) WINDOWS WITH GOOD FIT"
340 PRINT"     2) SOME LEAKAGE":PRINT"     3) RATHER DRAFTY"
350 INPUTY:PRINT:PRINT"CHOOSE ONE OF FOLLOWING":PRINT"     1) DOORS FIT GOOD"
360 PRINT"     2) SOME LEAKAGE":PRINT"     3) DRAFTY":INPUTI
370 PRINT:PRINT:PRINT"CHOOSE ONE OF FOLLOWING":PRINT
380 PRINT"     1) CAULKING AND WEATHERSTRIPPING GOOD":PRINT"     2) NEEDREPAIR"
390 PRINT"     3) NO CAULKING OR WEATHERSTRIPPING":INPUTS
400 INPUT"FLOOR AREA OF HOUSE - SQ FT";X
410 X=X*(Y+I+S-3)/100*FI:X=INT(X*100)/100:PRINT
420 PRINT"ANNUAL SAVINGS FOR CAULKING AND WEATHERSTRIPPING= $";X:GOSUB800
440 PRINT:PRINT:PRINT"ANNUAL SAVINGS FROM CEILING INSULATION":PRINT:PRINT
450 Y=38:INPUT"CEILING R VALUE";X:IFH<3THENY=26
455 IFH=3THENY=30
460 IFH=4THENY=33
465 INPUT"FIRST FLOOR AREA OF HOUSE (SQ FT)";F
470 R=Y:GOSUB900:I=R:R=X:GOSUB900:X=R:X=INT(100*(X-I)*F*FI)/100
475 IFX<0THENX=0
480 PRINT"ANNUAL SAVINGS BY BRINGING CEILING R UP TO";Y;" = $";X:GOSUB800
550 INPUT"IS THE HOUSE ON PILLARS OR HAVE AN UNHEATED BASEMENT";B$
555 IFASC(B$)=78THEN799
560 PRINT"CHOOSE FOUNDATION FACTOR FROM LIST BELOW":PRINT
565 PRINT"   FACTOR      FOUNDATION CHARACTERISTICS":PRINT
570 PRINT"     0.5 BUILDING WITH TIGHT CRAWL SPACE"
580 PRINT"     0.5 BUILDING WITH TIGHT BASEMENT (UNHEATED)"
590 PRINT"     0.8 STONE WALL BASEMENT (UNHEATED)"
600 PRINT"     0.8 2 FT OR MORE OF BASEMENT WALL EXPOSED (UNHEATED)"
610 PRINT"     0.8 CRAWL SPACE SKIRTED"
620 PRINT"     1.0 BUILDING ON PILLARS WITH NO SKIRTS"
625 PRINT:INPUT"FLOOR FACTOR FROM ABOVE TABLE";J
627 Y=11:IFH>1THENY=13:IFH>2THENY=19:IFH>3THENY=22
628 R=Y:GOSUB900:Q=R:INPUT"CURRENT R FACTOR FOR FLOOR";R
```

LETTER PERFECT T.M. LJK
WORD PROCESSING

ATARI 400/800 — APPLE II & II+

EASY TO USE — Letter Perfect is a single load easy to use program. It is a menu driven, character orientated processor with the user in mind. FAST machine language operation, ability to send control codes within the body of the program, mnemonics that make sense, and a full printed page of buffer space for text editing are but a few features. Screen Format allows you to preview printed text. Indented margins are allowed. Data Base Merge with **DATA PERFECT** by LJK, form letters, accounting files and mailing labels only with MAIL MERGE/UTILITY by LJK. FEATURES — Proportional/ Incremental spacing * Right Justification * File Merging * Block movement * Headers * Footers * Print Multiple Copies * Auto Page Numbering * Scroll forward/backward * Search and Replaces * Full cursor control * Underlining * Boldface * Superscripts * Subscripts * Auto page numbering * Insert character/line * Delete character/line * Centering * Horizontal tabs/changeable * Multifunction format line (line spacing — left margin — page width — lines/page — change fonts — top/ bot margin adjust) **MUCH MORE! $149.95**

ATARI VERSION 2.0 #2001

Uses proportional font, right justified with Atari 825/Centronics 737, 739 printers. Uses EPSON MX* Series + Graftrax/ italicized font. Can mix type fonts on same page; mix boldface and enhanced font in same line with justification. Can be used with 16K Atari/400.*

"Compared to the price of many other word processors, this package is a steal. It does everything the advertisement claims and more. On top of this the software is very easy to use." **A.N.A.L.O.G. MAGAZINE**

APPLE VERSION 5.0 #1001

DOS 3.3 compatible — Use 40 or 80 column interchangeably (Smarterm — ALS; Videoterm-Videx; Full View 80 — Bit 3 Inc.; Vision 80 — Vista; Sup-R-Term — M&R Ent.) Reconfigurable at any time for different video, printer, or interface. **USE HAYES MICROMODEM II*** LCA necessary if no 80 column board, need at least 24 K of memory. Files saved as either Text or Binary. Shift key modification allowed. Data Base Merge compatible with **DATA PERFECT*** by LJK.

"For $150, Letter Perfect offers the type of software that can provide quality word processing on inexpensive micro-computer systems at a competitive price." **INFOWORLD**

DATA PERFECT T.M. LJK

APPLE & ATARI
DATA BASE MANAGEMENT

INTRODUCTORY PRICE $99.95

Complete Data Base System. User orientated for easy and fast operation. 100% Assembly language. Easy to use. You may create your own screen mask for your needs. Searches and Sorts allowed, Configurable to use with any of the 80 column boards of Letter Perfect word processing, or use 40 column Apple video. Lower case supported in 40 column video. Utility enables user to convert standard files to Data Perfect format. Complete report generation capability. Much More!

EDIT 6502 T.M. LJK

This is a coresident — two pass **ASSEMBLER, DIS-ASSEMBLER, TEXT EDITOR,** and **MACHINE LANGUAGE MONITOR**. Editing is both character and line oriented. Disassemblies create editable source files with ability to use predefined labels. Complete control with 41 commands, 5 disassembly modes, 24 monitor commands including step, trace, and read/write disk. Twenty pseudo opcodes, allows linked assemblies, software stacking (single and multiple page) plus complete printer control, i.e. paganation, titles and tab setting. User can move source, object and symbol table anywhere in memory. Feel as if you never left the environment of **BASIC**. Use any of the 80 column boards as supported by **LETTER PERFECT**, Lower Case optional with **LCG**.

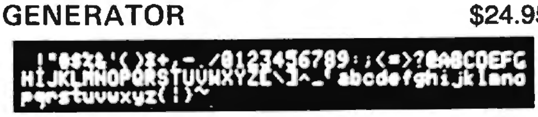
LJK DISK UTILITY — APPLE $29.95

This menu driven program allows the user to manipulate a variety of different file types. Binary, Text, and Source files may be easily converted into each other. The program may be used with **APPLESOFT***, **VISCALC***, and other programs. These program files may be readily adapted for multiple use including editing with **LETTER PERFECT** word processings.

MAIL MERGE/UTILITY — ATARI $29.95

This menu driven program combined with **LETTER PERFECT** allows user to generate form letters and print mailing labels. With the Atari, you may **CONVERT ATARI DOS FILES**, or Visicalc files compatible for editing with **LETTER PERFECT.** Utility creates Data Base files for Letter Perfect.

LOWER CASE CHARACTER GENERATOR — $24.95



Lower Case Character Generator for the Rev. 7, Apple II or II+ computers. When installed, this Eprom will generate lower case characters to the video screen. Lower case characters set has two dot true descenders. Installation instruction included. Manual includes listing of software for full support and complete instructions for shift key modification. Compatible with **LETTER PERFECT.**

LJK™ COMPUTER BASED SOFTWARE — ENTERPRISES INC

LJK ENTERPRISES INC.
P.O. Box 10827
St. Louis, MO 63129
(314) 846-6124

DEALER INQUIRES INVITED

```
630 GOSUB900:X=J*(R-Q)*F*FI:PRINT:PRINT:X=INT(X*100)/100:IFX<0THENX=0
640 PRINT"ANNUAL SAVINGS BY INCREASING FLOOR R VALUE TO ";Y;" =$";X
645 GOSUB800
799 RESTORE:PRINT:PRINT:GOSUB800:PRINT:PRINT:L=96:GOTO14
800 PRINT"==============================================================="
801 RETURN
828 R=Y:GOSUB900:I=R
899 REM CONDUCTION FACTOR SUBROUTINE
900 IFR<11THEN920
901 IFR<12THENR=.077:RETURN
902 IFR<15THENR=.066:RETURN
903 IFR<20THENR=.048:RETURN
904 IFR<24THENR=.042:RETURN
905 IFR<28THENR=.036:RETURN
906 IFR<34THENR=.031:RETURN
910 R=.025:RETURN
920 R=.5-.0385*R:RETURN
1000 DATACALIFORNIA,LOSANGELES 1 4 SANFRANCISCO 3 4 SACRAMENTO 1 3
1010 DATA COLORADO,DENVER 3 4 DURANGO 4 4 ASPEN 4 5
1020 DATAALABAMA,MONTGOMERY 1 2 BIRMINGHAM 1 3 HUNTSVILLE 2 3
1030 DATAARIZONA,PHOENIX 1 3 FLAGSTAFF 3 3,CONNECTICUTT,HARTFORD 3 5
1040 DATANEW MEXICO,ROSWELL 2 3 ALBUQUERQUE 3 3 SANTAFE 3 4
1050 DATAUTAH,SALTLAKECITY 3 4 MOAB 4 4,IDAHO,BOISE 3 5 POCATELLO 4 4
1070 DATAMONTANA,BILLINGS 4 5,OREGON,PORTLAND 2 5 BAKER 3 5
1090 DATAWASHINGTON,SEATTLE 3 5,NEVADA,RENO 3 3 LASVEGAS 1 3
1110 DATANORTH DAKOTA,GRANDFORKS 5 5,OKLAHOMA,OKLAHOMACITY 2 3
1120 DATASOUTH DAKOTA,SIOUXFALLS 4 4 PIERRE 4 5
1130 DATANEBRASKA,OMAHA 3 4,KANSAS,WICHITA 2 4 TOPEKA 3 4
1160 DATATEXAS,DALLAS 1 3 HOUSTON 1 2 BROWNSVILLE 0 1 AMARILLO 2 3
1170 DATALOUISIANA,NEWORLEANS 1 2 SHREVEPORT 1 3
1180 DATAARKANSAS,LITTLEROCK 1 3 FAYETTEVILLE 2 3
1190 DATAMISSOURI,SPRINGFIELD 2 3 STLOUIS 2 4 KANSASCITY 3 4
1200 DATAIOWA,DESMOINES 3 4 SIOUXCITY 4 4,VERMONT,MONTPELIER 4 5
1210 DATAMINNESOTA,MINNEAPOLIS 4 5 DULUTH 5 5,NEW HAMPSHIRE,CONCORD 4 5
1220 DATAWISCONSIN,MADISON 4 4 EAUCLAIRE 4 5,RHODE ISLAND,PROVIDENCE 3 5
1230 DATAILLINOIS,CHICAGO 3 4 SPRINGFIELD 2 4,VIRGINIA,RICHMOND 2 4
1240 DATAMICHIGAN,DETROIT 3 4 GRANDRAPIDS 4 5 SAULTST.MARIE 5 5
1270 DATAINDIANA,INDIANAPOLIS 3 4 EVANSVILLE 2 4,WYOMING,CASPER 4 5
1290 DATATENNESSEE,MEMPHIS 2 3 KNOXVILLE 2 4 CHATTANOOGA 2 3
1300 DATAMISSISSIPPI,JACKSON 1 2 TUPELO 1 3,KENTUCKY,LOUISVILLE 2 4
1320 DATAWEST VIRGINIA,CHARLESTON 3 4,MASSACHUSETTS,BOSTON 3 5
1330 DATAFLORIDA,MIAMI 0 1 JACKSONVILLE 1 2,OHIO,COLUMBUS 3 4
1340 DATAGEORGIA,SAVANNAH 1 3 ATLANTA 2 3,MAINE,PORTLAND 4 5
1350 DATASOUTH CAROLINA,CHARLSTON 1 3,NEW JERSEY,NEWARK 2 4
1360 DATANORTH CAROLINA,RALEIGH 2 3 ASHEVILLE 2 4 WILMINGTON 1 3
1390 DATAPENNSYLVANIA,PITTSBURGH 3 4,MARYLAND,BALTIMORE 2 4
1410 DATANEW YORK,NEWYORK 3 4 ALBANY 4 5,DELAWARE,WILMINGTON 2 4
1480 DATAERROR
1500 DATAOIL/GALLON,1,0,NATGAS/CUFT,120,150,ELECTRICITY/KWH,30,15
1510 DATAWOOD/CORD,.01,0,LPG/CUFT,50,60,LPG/LBS,6,7,LPG/GALLON,1.3,1.5
1520 DATACOAL/TON,.006,0
2000 END
```

Program 2.
Change these lines for Atari:

```
0 DIM B$(20),C$(20),D$(20),BB$(4*10),BBL
(4),X(10),Y(10),FH(10),FC(10)
14 ? :? :? :? "STATE (DON'T ABBREVIATE)"
;:INPUT B$:FOR I=1 TO L:READ C$
15 IF B$(1,7)=C$(1,7) THEN B$=C$:READ D$
20 NEXT I:FOR I=1 TO 4:BBL(J)=0:NEXT I:I
=1:Y=LEN(D$):J=1
25 X=ASC(D$(I,I)):IF X=32 THEN 45
30 BBL(J)=BBL(J)+1:BB$((J-1)*10+1,(J-1)*
10+BBL(J))=CHR$(X):GOTO 55
```

```
45 I=I+1:X(J)=VAL(D$(I,I)):I=I+2:Y(J)=VA
L(D$(I,I))
60 POKE 85,15:? I;:POKE 85,20:? BB$((I-1
)*10+1,(I-1)*10+BBL(I));:POKE 85,35:? B$
:NEXT I
65 ? :? :? "CHOOSE # FOR NEAREST CITY";:
INPUT I:H=X(I):C=Y(I):REM ZONES
85 NEXT I
90 HC=X:X=0:FOR I=0 TO 5:IF H=I THEN 100
:REM HF & CF ARE HEAT & COOL FACTORS
95 X=X+0.5:NEXT I
110 FOR I=1 TO 8:READ B$,A,B:FH(I)=A:FC(
I)=B:PRINT "                    ";I;B$:NEXT I

115 ? :? :? "CHOOSE # FOR HEATING FUEL(C
ENTS)";:INPUT S:S=S/100
126 ? :? :? "CHOOSE # FOR COOLING FUEL";
:INPUT J
127 ? "COST PER UNIT FOR COOLING FUEL(CE
NTS)";:INPUT S:S=S/100
150 ? "COUNT STORM WINDOWS OR SLIDING GL
ASS  DOORS";:INPUT X
200 ? "ANNUAL HEATING FUEL COST (DOLLARS
)";:INPUT HS:HS=HS*X:? :? :X=0.6
230 ? "ANNUAL COOLING FUEL COST (DOLLARS
)";:INPUT CS:CS=CS*X
245 Y=Y-0.01:NEXT I:IF H=4 THEN Y=0.025
260 ? "DEGREES TURNED DOWN DURING DAY";:
INPUT X:S=INT(100*Y*HS*X)/100
285 ? "DEGREES THERMOSTAT TURNED UP DURI
NG   COOLING?";:INPUT X
400 ? "FLOOR AREA OF HOUSE-SQ FT";:INPUT
 X
450 Y=38:? "CEILING R VALUE";:INPUT X:IF
 H<3 THEN Y=26
465 ? "FIRST FLOOR AREA OF HOUSE (SQ FT)
";:INPUT F
550 ? "IS THE HOUSE ON PILLARS OR HAVE A
N    UNHEATED BASEMENT";:INPUT B$
625 ? :? "FLOOR FACTOR FROM ABOVE TABLE"
;:INPUT J
628 R=Y:GOSUB 900:Q=R:? "CURRENT R FACTO
R FOR FLOOR";:INPUT R
```

Figure 1. Sample Run

```
ITEMS NEEDED FOR ENERGY WORKBOOK

1) STATE
2) CITY
3) HEATING FUEL COST (E.G. .37 CENTS/CU FT)
4) COOLING FUEL COST (E.G. 5.14 CENTS/KWH)
5) SQUARE FT OF SINGLE GLASS WINDOWS IN HOUSE
7) ANNUAL HEATING AND COOLING FUEL COST
8) CHECK FOR LEAKAGE AROUND WINDOWS AND DOORS WITH CANDLE
9) FLOOR AREA OF HOUSE - SQ FT
10) CEILING R VALUE - USE TABLE PROVIDED WITH INSTRUCTIONS
11) FIRST FLOOR AREA - SQ FT
12) FLOOR R VALUE IF BASEMENT IS UNHEATED OR HOUSE IS ON PILLARS


STATE(DON'T ABBREVIATE)? TEXAS


                1   DALLAS        TEXAS
                2   HOUSTON       TEXAS
                3   BROWNSVILLE   TEXAS
                4   AMARILLO      TEXAS

CHOOSE # FOR NEAREST CITY? 2

                1 OIL/GALLON
                2 NATGAS/CUFT
```

```
                 3 ELECTRICITY/KWH
                 4 WOOD/CORD
                 5 LPG/CUFT
                 6 LPG/LBS
                 7 LPG/GALLON
                 8 COAL/TON


CHOOSE # FOR HEATING FUEL? 2
COST PER UNIT FOR HEATING FUEL(CENTS)? .45


CHOOSE # FOR COOLING FUEL? 3
COST PER UNIT FOR COOLING FUEL(CENTS)? 6.




INPUT # OF SQUARE FT OF SINGLE GLASS WINDOWS, DO NOT
COUNT STORM WINDOWS OR SLIDING GLASS DOORS? 190
ANNUAL SAVINGS DUE TO STORM WINDOWS= $ 116.7
================================================================

IS HEATING FUEL USED FOR OTHER PURPOSES, E.G. COOKING ? Y
ANNUAL HEATING FUEL COST (DOLLARS)? 175.


IS COOLING FUEL USED FOR OTHER PURPOSES, E.G. LIGHTING? Y
ANNUAL COOLING FUEL COST (DOLLARS)? 800.


================================================================
THE FOLLOWING SECTION EVALUATES THE SAVINGS OBTAINED BY TURNING
THE THERMOSTAT DOWN IN WINTER OR UP IN SUMMER FROM THE SETTING
YOU HAVE BEEN USING.


HEATING

DEGREES TURNED DOWN DURING DAY? 5
SAVINGS =$ 37.18
ADDITIONAL DEGREES TURNED DOWN DURING NIGHT? 5
SAVINGS=$ 11.15
ANNUAL TOTAL HEATING SAVINGS =$ 48.33

COOLING

DEGREES THERMOSTAT TURNED UP DURING COOLING? 5
SAVINGS =$ 48

TOTAL ANNUAL SAVINGS =$ 96.33
================================================================


ANNUAL SAVINGS FROM CAULKING AND WEATHERSTRIPPING
CHECK DRAFTS HOLDING CANDLE NEAR CRACK ON WINDY DAY
CHOOSE ONE OF FOLLOWING
```

```
      1) WINDOWS WITH GOOD FIT
      2) SOME LEAKAGE
      3) RATHER DRAFTY
? 2

CHOOSE ONE OF FOLLOWING
      1) DOORS FIT GOOD
      2) SOME LEAKAGE
      3) DRAFTY
? 2


CHOOSE ONE OF FOLLOWING

      1) CAULKING AND WEATHERSTRIPPING GOOD
      2) NEEDREPAIR
      3) NO CAULKING OR WEATHERSTRIPPING
? 2
FLOOR AREA OF HOUSE - SQ FT? 2000

ANNUAL SAVINGS FOR CAULKING AND WEATHERSTRIPPING= $ 56.7
================================================================
ANNUAL SAVINGS FROM CEILING INSULATION


CEILING R VALUE? 19
FIRST FLOOR AREA OF HOUSE (SQ FT)? 2000
ANNUAL SAVINGS BY BRINGING CEILING R UP TO 26  = $ 22.68
================================================================
IS THE HOUSE ON PILLARS OR HAVE AN UNHEATED BASEMENT? Y
CHOOSE FOUNDATION FACTOR FROM LIST BELOW

   FACTOR      FOUNDATION CHARACTERISTICS

      0.5 BUILDING WITH TIGHT CRAWL SPACE
      0.5 BUILDING WITH TIGHT BASEMENT (UNHEATED)
      0.8 STONE WALL BASEMENT (UNHEATED)
      0.8 2 FT OR MORE OF BASEMENT WALL EXPOSED (UNHEATED)
      0.8 CRAWL SPACE SKIRTED
      1.0 BUILDING ON PILLARS WITH NO SKIRTS

FLOOR FACTOR FROM ABOVE TABLE? 1.
CURRENT R FACTOR FOR FLOOR? 7


ANNUAL SAVINGS BY INCREASING FLOOR R VALUE TO  11  =$ 290.11
================================================================
```

©

Two Short Programs Of CAI For Teaching BASIC

R. Hiatt
Dept. of Chemistry
Brock University
St. Catharines, Ontario

While much is written about, and many programs are produced for, CAI in diverse subjects and disciplines, there seems to be very little about CAI applied to learning about computation itself.

To be sure, the computer itself is the best CAI for individual self-paced effort. But for the classroom, demonstration programs are useful. By CAI, however, I don't mean simple routines that are merely examples, but rather, programs that make their point by manipulation and/or simulation of a process. I venture to present two examples.

1. PEEKPROG – A BASIC program that PEEKs itself.

At some point in teaching a course in BASIC, it seems useful to PEEK out to the screen a page or so of a program. This can be in response to a question as to why conserving line numbers conserves memory space, or a similar query suggesting that a brief digression into actual program storage would be illuminating. (In my experience, it is wise to postpone any mention of bits, bytes, hex or ASCII until this kind of curiousity evinces itself.)

While it's easy enough to PEEK out a page from immediate mode, (for I = 2049 to 2304: Print PEEK (I) " "; NEXT), and fill the screen with numbers, this is more astounding than useful. It takes a trained eye to discern any pattern at all.

Program 1 organizes the PRINT PEEK – so that each line number starts off a new line on the screen. By keeping the program short and the line numbers below 256 it is possible to pseudo list the whole program and have the line numbers easily recognizable (Figure 1). The line terminators can be pointed out, as can the linking bytes and the program terminator.

BASIC tokens can be mentioned, and the students can practice their ASCII by reading the REM statement, etc.

The program itself illustrates READ...DATA – partly because the questions seem to arise about the time we're dealing with that subject, and partly because it seemed an easy way to control the position of carriage returns in the loop.

Actually, this turned up something about the Apple that I hadn't realized; that is, each time a DATA statement is edited, an additional space, actually another byte with value 32, is inserted between this DATA token and the first byte of the data. This, of course, changes the length of the program and the byte at which a new print line is wanted. In other words, the datum just corrected has been made erroneous by the process of correction. The only way out seems to be to retype the line.

2. READ/RESTORE – (Program 2) is a routine that simulates a small portion of program which reads data into two arrays, one numeric and the other string. The displayed portion of the program, (lower screen, Figure 2), consists of two FOR I = ... READ A (I) ..., separated by a RESTORE, the whole followed by a FOR I = ...READ A$(I).... . Two psuedo cursors are employed, one flashing the current value of I at the end of the FOR I ... NEXT I loop, the other acting as a data pointer.

Action is controlled by the instructor, via the programmed "invisible GET"; i.e. IF PEEK - 16384 < 128 THEN .... The first (any) key depression causes a datum to be "READ"; its index and value appear in the table. This second key depression moves the data pointer and movements "I".

Progression of the program being entirely in the hands of the instructor, there is unlimited time to point out the salient features, to comment on indices, the differences between numeric and string data, etc., and to run the program two or three times for emphasis.

For Atari Users...

Program 3 will provide a dump of an Atari BASIC program, similar to Program 1. To merge it with another program, LIST it to tape or disk (with LIST"C:" or LIST"D:BDUMP"), load the program to be viewed, and then use ENTER"C:" or ENTER "D:BDUMP". Start the routine with GOTO 20000.

Program 4 is called "BASIC in Action." It is a display of a BASIC program (the "Atarized" version of Figure 1) as it RUNs. Just type it in and RUN it. A program will be listed at the bottom of the screen with an arrow pointing to the first line. When you press a key, the line will be executed. The action of the program, including the DATA statement pointer,

CALL TOLL FREE
800 424-2738

Programs for ATARI 400 800 Unless Otherwise Noted

the PROGRAM STORE

Save $$ on Popular Programs for Atari

LIMITED TIME OFFER

Now through March 15, take advantage of our price reduction on these popular programs

MATCH RACER

From Gebelli Software
A one or two player racing game that matches your skills against a diabolical road with randomly-occuring obstacles. Barracades, water, rocks and oil slicks all hamper your progress, and in the two-player version, your troubles are doubled! Realistic graphics and sound effects help make this game a delight.

16K disk...~~$29.95~~ Save 20%!

THRU APRIL 15 ONLY: Pay Just $23.95!

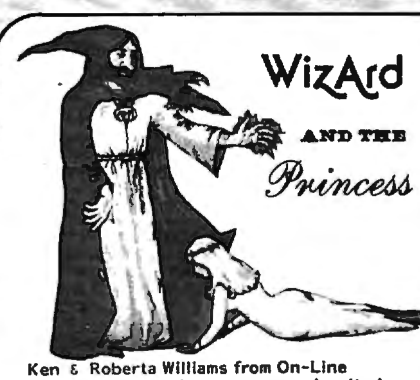
RACE IN SPACE

By Charles Bachand from Analog
Two-player game with 128 variations! Your objective is to travel through space further and faster than your opponent. All the while, you must dodge various densities of tiny asteroids and comets. To further impede your progress, you must avoid your opponent's (and your own) boomerang missiles! Selectable playing time from 3 to 5 minutes.

16K tape...~~$18.95~~
16K disk...~~$24.95~~ Save 20%!

THRU APRIL 15 ONLY:
Pay Just $15.15 (T), $19.95 (D)!

WizArd AND THE Princess

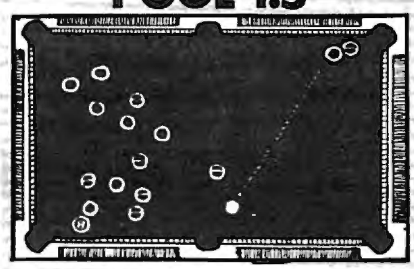


Ken & Roberta Williams from On-Line
HI-RES adventuring comes to Atari! Armed only with bread, water, a knife and a blanket, you set off in search of Princess Priscilla. She has been abducted from the kingdom by the dread Wizard, Harlin. If you can return her safely, half the kingdom is yours! Each scene is depicted in color and you can even get close-up views of important objects. Not for beginners.

40K disk...~~$32.95~~ Save 20%!

THRU APRIL 15 ONLY: Pay Just $26.35!

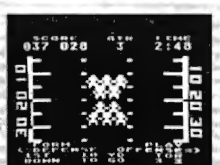
POOL 1.5



By Hoffman, St Germain & Morock from IDS
The pressure is on: if you can just get enough english on the ball to bank it into the corner pocket . . . In POOL 1.5, you can! A remarkable action-simulation of the real thing, this program allows full control of your "cue-stick" for aim and control. Play four different types of pool at your choice of table speed, with "instant replay" of any shot! HI-RES color graphics are used throughout this real-time game.

48K Disk...$34.95

CYPHER BOWL



From ArtSci
The most impressive 2-player football game we've seen for the Atari. Using five players per side instead of 11 keeps the game manageable while providing all the action you can handle. Each player uses a joystick to call formations and plays -- 256 possibilities -- then the players are maneuvered after the ball is hiked. Skill improvements come gradually and you can plan on many hours of excitement & enjoyment with CYPHER BOWL.

16K tape or disk...$49.95

MISSION: ASTEROID
HI-RES ADVENTURE #0

From On-Line
An introduction to HI-RES adventures, it is designed to acquaint beginning adventurers with this exciting genre of games. An asteroid is about to hit earth. Your job as a novice astronaut is to blow it up before it destroys earth. On your way you must cut through red tape impeding your plan, learn to fly the rocket, and follow a flight plan that will put you in striking distance.

40K disk...$24.95

RICOCHET

DeKoven & Connelley from Mind Toys
A unique new approach to computer gaming. RICOCHET combines diabolical strategy with fast action to yield a game that satisfies your intellect as well as your senses. You have 6 playing pieces to move around, plus 2 ball launchers. Each turn consists of either repositioning the pieces or launching a ball. Points are scored each time the ball ricochets. Simple? Deceptively so. Easy to master? Definitely not! 1 or 2 players.

32K disk...$19.95

HOCKEY

From Gamma Software
A high-speed video action game for 2,3, or 4 players. You use joysticks to control the players. The offensive player with the puck and the nearest defensive player are automatically assigned to a joystick; Players not so controlled become "smart" players who play automatically. The action is fast and furious, the color graphics and sound are realistic. Perfect for family entertainment.

16K disk or tape...$29.95

LUNAR LANDER

By Wall, Moncrief & Jessee from A.I.
Grab your joystick and sweat out the touchdown of your LEM. Choose from four lunar landscapes, with many choices for landing sites. Select a more difficult site and get more points -- if you can land successfully. You have complete control of your LEM via main engines and small side thrusters, and five successful landings are heralded with a patriotic event. Great fun!

24K tape (disk compatible)...$14.95

REAR GUARD

By Neil Larimer from Adventure Int.
A space battle between your cruiser and wave-after-wave of enemy ships. You can destroy them with your energy darts or you can ram them -- as long as your shields are intact. The action takes place on a continuous horizontal landscape. You control the altitude and speed of your craft as you pursue the enemy. Skill level 1 is exciting, and there are four more levels available.

24K disk...$24.95

DISK MANAGER

From Synapse Software
An excellent way to keep track of that growing pile of Atari disks. This easy-to-use system lets you categorize them by disk, subject or filename. Supports searches by name or wild-card entry. If you have a printer, DISK MANAGER will print listings or labels, too. Entry and updates are so simple that you'll have no excuse not to use it!

32K disk...$29.95

Visit our other stores: Seven Corners Center · Falls Church, VA & W. Bell Plaza · 6600 Security Blvd. · Baltimore, MD

TO ORDER CALL TOLL FREE 800 424-2738 For information Call (202) 363-9797

MAIL ORDERS: Send check or M.O. for total purchase price, plus $1.00 postage & handling. D.C. residents, add 6% tax. Charge card customers: include all embossed information on card.

THE PROGRAM STORE
4200 Wisconsin Avenue NW, Dept. U03 Box 9609
Washington, D.C. 20016

variables, strings, DIMensioning, and FOR/NEXT loops are visually and audibly demonstrated.

Figure 1.

```
20   8
10   0   178   32   80   69   69   75   32   80   82   79   71   8
 2  65   77   0   27   8
20   0   135   65   0   41   8
30   0   129   73   208   50   48   52   57   193   65   0   53
 8
40   0   173   73   209   66   196   54   48   0   62   8
50   0   135   66   58   186   0   76   8
60   0   186   226   40   73   41   34   32   34   59   0   83   8
70   0   130   73   0   89   8
80   0   128   0   115   8
100   0   131   32   50   50   51   50   44   50   48   53   49
44   50   48   55   48   44   50   48   55   55   0   141   8
110   0   131   32   50   48   57   49   44   50   49   48   51
44   50   49   49   50   44   50   49   50   54   0   167   8
120   0   131   32   50   49   51   51   44   50   49   51   57
44   50   49   54   53   44   50   49   57   49   0   183   8
200   0   131   32   50   50   50   49   55   44   50   51   48   48
0   0   0
```

Figure 2.

40 for I = 1 to 3: Read A(I): Next I
50 Restore
60 For I = 4 to 6: Read A(I): Next I
70 For I = 1 to 5: Read A$(I): Next I
80 Data 76, 40, 67
90 Data "SUE", "ANN", "JOE", "KIM", "JIM"

Program 1.

```
10  REM  PEEK PROGRAM
20  READ A
30  FOR I = 2049 TO A
40  IF I < B THEN 60
50  READ B: PRINT
60  PRINT  PEEK (I)" ";
70  NEXT I
80  END
100  DATA  2232,2051,2070,2077
110  DATA  2091,2103,2112,2126
120  DATA  2133,2139,2165,2191
200  DATA  2217,2300
```

Program 2.

```
10  REM  DATA READ-RESTORE DISPLAY
20  GOTO 50
30  IF  PEEK ( - 16384) < 128 THEN 30
40  POKE  - 16368,0: RETURN
50  DIM A(10),A$(10)
60 B$(0) = "INDEX":B$(1) = "DATA":C$ =  CHR$ (34):B$ = C$ + "," + C$
70  HOME
80  PRINT  TAB( 6)"A(I)" TAB( 25)"A$(I)"
90  PRINT B$(0) TAB( 10)B$(1) TAB( 20)B$(0) TAB( 30)B$(1)
100  VTAB 15
110  PRINT "40 FOR I = 1 TO 3 : READ A(I) : NEXT I"
120  PRINT "50 RESTORE"
130  PRINT "60 FOR I = 4 TO 6 : READ A(I) : NEXT I"
140  PRINT "70 FOR I = 1 TO 5 : READ A$(I) : NEXT I"
150  PRINT : PRINT "80 DATA 76,40,67"
160  PRINT : PRINT "90 DATA "C$"SUE"B$"ANN"B$"JOE"B$"KIM"B$"JIM"C$
199  REM  START THE DISPLAY
200 V1 = 1871:V2 = 1621:VT = 3:A1 = 1:B = 3: GOTO 220
210 V1 = 1143:V2 = 1621:VT = 6:A1 = 4:B = 6
220  FOR I = A1 TO B: POKE V1,I + 112
230 VT = VT + 1:V2 = V2 + 3: POKE V2,96
240  GOSUB 30: READ A(I)
250  VTAB VT: PRINT "  "I; SPC( 7)A(I): GOSUB 30
260  POKE V2,160: NEXT I: POKE V1,160: IF I > 4 THEN 300
270  POKE 1971,96: FOR I = V2 TO V2 - 5 STEP  - 1
280  POKE I,160: POKE I - 1,96: FOR J = 1 TO 300: NEXT : NEXT : GOSUB 30:
290  POKE 1971,160: RESTORE : GOTO 210
300 V2 = 1875:V1 = 1271:VT = 3
310  FOR I = 1 TO 5:V2 = V2 + 6:VT = VT + 1
```

```
320   POKE V1,I + 112: POKE V2,96: GOSUB 30: READ A$(I)
330   VTAB VT: HTAB 22: PRINT I SPC( 8)A$(I): GOSUB 30
340   POKE V2,160: NEXT
350   POKE V1,160
400   END
500   DATA  76,40,67
510   DATA  "SUE","ANN","JOE","KIM","JIM"
```

Program 3.

```
20000 REM ATARI BASIC LINE DUMP
20010 REM TYPE GOTO 20000
20020 CLR
20030 BEGIN=PEEK(136)+256*PEEK(137)
20040 LINE=PEEK(BEGIN)+256*PEEK(BEGIN+1)

20050 IF LINE=20000 THEN PRINT "LAST LIN
E":END
20060 PRINT LINE,
20070 LENGTH=PEEK(BEGIN+2)
20080 FOR I=1 TO LENGTH
20090 PRINT PEEK(BEGIN+2+I);" ";
20100 NEXT I
20110 PRINT :PRINT "-----"
20120 BEGIN=BEGIN+LENGTH
20130 GOTO 20040
```

Program 4.

```
10 GRAPHICS 0:POSITION 12,0:? "| BASIC I
n Action |"
20 POKE 752,1
30 OPEN #1,4,0,"K:"
40 DIM L$(40),T$(20):GOTO 100
50 REM CLICK/PAUSE SUBROUTINE
60 POKE 53279,0:FOR W=1 TO 10:NEXT W:RET
URN
100 POSITION 2,14:? "{37 R}":?
110 ? " 10 DIM A(6),A$(3*5),T$(3)"
120 ? " 20 RESTORE"
130 ? " 30 FOR I=1 TO 3:READ T:A(I)=T:NE
XT I"
140 ? " 40 FOR I=4 TO 6:READ T:A(I)=T:NE
XT I"
150 ? " 50 FOR I=1 TO 5:READ T$:
        A$(I*3-2,I*3)=T$:NEXT I"
160 ? " 60 DATA 76,40,67,29,14,33"
170 ? " 70 DATA SUE,ANN,JOE,KIM,JIM"
180 ? " 80 END";
190 REM START THE DISPLAY
200 FOR I=1 TO 8
210 POSITION 2,14+I+(I>5)
220 PRINT "{ESC}{RIGHT}";
240 GET #1,A
250 ON I GOTO 260,290,300,400,410,600,60
0,600
260 POSITION 2,1:FOR J=1 TO 6:? "A(";J;"
)=?":SOUND 0,PEEK(53770),10,8
265 FOR W=1 TO 50:NEXT W:NEXT J:SOUND 0,
0,0,0
270 ? "A$=";CHR$(34);CHR$(34);"{LEFT}";:
FOR J=1 TO 15:? "{INSERT}";:SOUND 0,J,8,
8:FOR W=1 TO 30:NEXT W:NEXT J
275 ? :? "T$=";CHR$(34);CHR$(34);"{LEFT}
";:FOR J=1 TO 3:? "{INSERT}";:SOUND 0,J,
8,8:FOR W=1 TO 30:NEXT W:NEXT J
280 SOUND 0,0,0,0:GOTO 600
290 FOR J=1 TO 10:POSITION 6,21:? "|DATA
|{4 LEFT}";:SOUND 0,50,12,8:FOR W=1 TO 2
0:NEXT W:SOUND 0,0,0,0:? "DATA":NEXT J
295 GOTO 600
300 OFF=0
310 FOR J=1 TO 3:READ T$
320 POSITION 2,9:? "I=";J+3*OFF:POSITION
 19,17+OFF:? "|READ T|"
330 POSITION 8+3*(J+OFF*3),21:? T$:READ
T$
340 POSITION 2,10:? "T=";T$
345 GET #1,A:POSITION 8+3*(J+OFF*3),21:?
 T$
350 POSITION 19,17+OFF:? "READ T":POSITI
ON 26,17+OFF:? "|A(I)=T|"
360 FOR K=1 TO 15:POSITION 9+K,10:? " ";
T$;:GOSUB 50:NEXT K:? "{2 LEFT}  "
370 FOR K=1 TO J+OFF*3:POSITION 24,10-K:
? T$;:GOSUB 50:? "{2 LEFT}  ";:NEXT K
380 POSITION 24,J+OFF*3:? T$;:POSITION 7
,J+OFF*3:FOR K=1 TO 17:? "{DELETE}";:POK
E 53279,0:NEXT K
```

```
390 GET #1,A:POSITION 26,17+OFF:? "A(I)=
T"
395 NEXT J:POSITION 2,9:? "I=";J+OFF*3:G
OTO 600
400 OFF=1:GOTO 310
410 FOR J=1 TO 5:POSITION 2,9:? "I=";J
420 POSITION 19,19:? "|READ T$|":READ T$
:POSITION 7+J*4,22:? T$
430 READ T$:POSITION 6,8:? T$:GET #1,A
440 POSITION 19,19:? "READ T$":POSITION
6,20:? "|A$(I*3-2,I*3)=T$|":POSITION 7+J
*4,22:? T$
450 FOR V=15 TO 0 STEP -1:SOUND 0,10+20*
RND(1),10,V:NEXT V
470 POSITION 3+J*3,7:? T$
480 GET #1,A:POSITION 6,20:? "A$(I*3-2,I
*3)=T$":NEXT J:POSITION 2,9:? "I=";J:GOT
O 600
600 POSITION 2,14+I+(I>5):? " ";
610 NEXT I
620 POSITION 2,10:POKE 752,0:END
1000 DATA |76|,76,|40|,40,|67|,67,|29|,2
9,|14|,14,|33|,33
1010 DATA |SUE|,SUE,|ANN|,ANN,|JOE|,JOE,
|KIM|,KIM,|JIM|,JIM
```

©

Infinite Precision Multiply

G. H. Watson
Newark, DE

123x456 = 56088. Right? Sure! Just ask a young student and he/she will quickly work out something similar to:

```
  123
x 456
  738
 615
492
56088
```

If you are near a microcomputer, you may verify this result by entering ?123*456 <CR>.

How about 123123 x 456456? At this point your young student may balk and suggest "Let's let the computer do it!" Unfortunately the computer may also balk at such a request. Entering ?123123* 456456 <CR> into a PET yields the following result: 5.62002321E + 10. With some further encouragement the student may work out 56200232088 as the correct product. Why the difference?

In PET BASIC most numbers are represented in floating-point format, which allows convenient handling of real numbers which may be positive or negative, integer, fractional, or mixed, and of greatly varying magnitude. The format in the PET is such that numbers have nearly nine digits of precision. This means that 123456789 may be represented exactly in this format, but something will have to be done for 1234567891. That something is called scientific notation — 1234567891 will be handled as 1.23456789E + 09, where E + 09 means that the decimal would be moved 9 spaces to the right. On the PET enter X = 1234567891 <CR> and then enter ?X <CR>. Notice that the trailing digit "1" has been dropped; it would have been the tenth digit and only nine digits of precision are possible.

Scientific notation has great utility and nine digits of precision will be plenty for many cases of numerical calculation. Occasionally, though, you may run up against a calculation which requires infinite precision — you need the answer exactly (no lost digits). Recently I needed to know the exact product of several large numbers. Working the problem by hand and finding several errors, I decided to write a short infinite precision multiply routine to check my answer. INFINI-MULT is the resulting BASIC program.

Using strings to represent the numbers is the trick for extending the precision. By operating on each string with the string function MID$(,,), each digit of the number may be isolated and an arithmetic operation performed. Only single digit operations are performed in INFINI-MUILT. The microcomputer is doing only second grade arithmetic — addition and multiplication of two numbers between zero and nine.

The result of the operation on two single-digit numbers may be a double-digit number. Here enters the carry digit. Most of the errors I make when doing arithmetic by hand involve the carry digit. Care must also be taken when programming for the carry digit. The subroutine at line 500 separates the double-digit number P into the carry digit C and adds the remaining digit onto the string D$ from the left. For example, if P = 25 and D$ = "456" then line 500 will return with C = 2 and D$ = "5456".

In order to simplify handling the carry digits and make the program as straightforward as possible, the strings involved in the addition routine are made the same length by padding from the left with zeros (line 380). For instance, if Z$ = "123" and A$ = "45600" then we will pad Z$ so that Z$ = "00123". We also pad strings with zeros from the right in the multiplication routine so that the proper power of ten is obtained (line 330).

INFINI-MULT handles 123 x 456 as follows:

```
lines 250-330,   370-440
       123         000
     x   6      +  738
       738         738

       123        0738
     x   5       +6150
      6150        6888

       123       06888
     x   4      +49200
     49200       56088
```

If the two numbers to be multiplied differ in the number of digits, the multiplication routine will be faster with the smaller number as the multiplier (456 in example) and the larger number as the multiplicand (123 in example) — just as you learned in grade school. The digits "0" and "1" are treated preferentially in lines 270 and 280 so time will be saved if the number with many ones and zeros is used as the multiplier. It will be faster to multiply 456 by 123 in INFINI-MULT than 123 by 456.

I have made no provision for handling decimal points: this would be an interesting modification for you to make. To sidestep this limitation use the same trick that you learned in school: count the number chips . . . chips . . . chips . . . chips . . . chips . . . chips . . .

Limited Time: FREE Disk-O-Pro.

Special: You can get this Disk-O-Pro **absolutely free**... when you order the new Commodore 2031 single disk drive from Skyles. (See page 8.)

Command-O or Command-O-Pro?

It's called the Command-O-Pro in Europe, Command-O in the U.S.

But whatever you call it, this 4K byte ROM will provide your CBM BASIC 4.0 (4016, 4032) and 8032 computers with 20 additional commands including 10 Toolkit program editing and debugging commands and 10 additional commands for screening, formatting and disk file manipulating. (*And our technical writer dug up 39 additional commands in the course of doing a 76-page manual!!*)

Extends Commodore's 8032 advanced screen editing features to the ultimate. **SCROLL** up and down, insert or delete entire lines, delete the characters to the left or right of the cursor, select **TEXT** or **GRAPHICS** mode or ring the PET's bell. Redefine the window to adjust it by size and position on your screen. And define any key to equal a sequence of up to 90 key strokes.

Resides in hexadecimal address **$9000**, the rightmost empty socket in 4016 and 4032 or the rearmost in 8032. If space conflicts, Socket-2-ME available at a very special price.

Command-O from Skyles Electric Works $75.00
Complete with **Socket-2-ME** 95.00

"The PET with the new PicChip... is like home movies."

The new ROM that took Europe by storm is available only from Skyles Electric Works in the U.S. and Canada.

PicChip, a ROM extension of the BASIC Version III (BASIC 2), BASIC 4.0 or BASIC 8032 interpreter. Over 40 commands that allow you to create programs with dynamic graphics displays: plots, bar graphs, pictures; and rolling, scrolling, shifting and inverting. All instantly and easily added to your BASIC program.

The address for the 2000/3000 (which would require PicChip module PC2), for the 4000 (PC4) and for the 8000 (PC8) is **$A000**... unless you have a Mikro, Word Pro 3 or 4, or Jinsam, which occupy that same address. In those cases, you will need the PicChip on an interface board... for the 2000/3000 series, PCB2, for the 4000, PCA4, and for the 8000, PCA8. In all cases, the Mikro or WordPro would be switchable manually using the Skyles Socket-2-ME... available at a very special price.

PicChip from Skyles Electric Works $60.00
(Please indicate PC2, PC4, PC8)
Complete with Socket-2-ME 80.00
(Please indicate PCB2, PCA4, PCA8)

Disk-O-Pro ROM:

25 new commands at $75.00, that's only $3.00 a command!

The powerful chip for all "classic" PETs. In any PET with Version III (BASIC 2.0) ROMs (### **COMMODORE BASIC** ###) Disk-O-Pro will give 19 software compatible disk instructions*; 15 identical with the new BASIC 4.0 (or with 8032 ROMs) compatible with both old and new DOS. Plus 4 additional disk commands... including program appending (**MERGE**), program overlaying (**MERGE #**) and **PRINT USING**, allowing formatting output of strings and numbers on the PET screen or on any printer.

**NOTE: Old DOS doesn't recognize 3 commands.*

Plus softtouch key (**SET**) which allows you to define a key to equal a sequence of up to 80 keystrokes; **SCROLL** whereby all keys repeat as well as slow scrolling and extra editing features; **BEEP** which allows you to play music on your PET.

Completely compatible with the BASIC Programmer's Toolkit. (*The chip resides in the socket at hexadecimal address* **$9000**, *the rightmost empty socket in most PETs.*) And for the owners of "classic" (or old) PETs, we do have interface boards.

(For those owning a BASIC 4.0 or 8032, even though the Disk-O-Pro may not be suitable, the Command-O is. We have never abandoned a PET owner.)

Complete with 84-page manual written by Greg Yob.

Disk-O-Pro from Skyles Electric Works $75.00
Complete with Interface Board (for "Classic" PETs) 95.00

The New Mikro:

Now you can program in machine language as easily as writing BASIC.

4K machine language assembler ROM plugs into your main board. Does all the machine language work for you; all you have to do is start laying down the code.

Retains all the great screen editing features of the PET... even all the Command-O or Toolkit commands. Write your own machine language subroutine. The program you write is the source code you can save. Machine language monitor saves the object code. Not as professional as the Skyles MacroTeA... not as expensive, either.

A great learning experience for those new to machine language programming but who want to master it easily. Superb 50-page manual—including brief introduction to machine language programming—included. But we also recommend several of the books in the Skyles Book Barn.

Skyles Mikro Machine Language Assembler
MK 2/3 for 2000 or 3000 Series PETs $80.00
MK 4/8 for 4000 or 8000 Series PETs and CBMs 95.00
Complete with **Socket-2-ME**

Skyles Catalogue Page 1

For PET and CBM owners only:

This is just 1 of 20 pages of the newest and biggest Skyles catalog, hot off the press.

We know you'll want this page, in its full 8½ x 10 splendor, and another 19 pages of peripherals, software and books that will make your PET or CBM computer even nicer to live with. So, if we missed sending you your very own copy within the last few weeks, call us at **(800) 227-9998** (unless you live in California, in which case call **(415) 965-1735.**

From Skyles Electric Works, the oldest and largest professional specialists in the business.

See us at the West Coast Computer Faire-Booth 1619-21 March 19-21.

Skyles Electric Works
231-E S. Whisman Road
Mountain View, California 94041

of places to the right which the decimal point must be moved to reach the last digit in the multiplicand and multiplier, add, and move that many places to the left in the product. For example, consider 1.23 x .456 :

1.23	2 right ->	123
x .456	+3 right ->	x 456
.56088	<-5 left	56088

INFINI-MULT follows the simple rules of arithmetic which you learned early in school. There may be faster or more sophisticated ways of getting the extra precsion you desire. The PET/CBM PERSONAL COMPUTER GUIDE contains some interesting double-precision routines for addition, subtraction, and multiplication. Play around with INFINI-MULT and some evening when you feel like writing a quick program take a shot at INFINI-DIVI.

Program 1: Microsoft Version

```
100 REM  INFINITE PRECISION MULTIPLY
110 REM      G.H.WATSON  6/25/81
120
130 REM  INITIALIZATION
140
150 DIM P,C,D$,J,Z,A,X,Y,X$,Z$,A$
160 DIM I,AL,ZL,XL,YL,Y$
170 INPUT "X = ";X$:INPUT "Y = ";Y$
180 XL=LEN(X$):YL=LEN(Y$)
190 I=XL+YL:DIM N$(9),Q$(I),C$(8)
200 FOR J=1TOI:Q$(J)="0"+Q$(J-1):NEXT
210 FOR J=0TO9:N$(J)=CHR$(J+48):NEXT
215 FOR J=1TO8:C$(J)=N$(J):NEXT
220
230 REM  MULTIPLICATION ROUTINE
240
250 FOR I=YLTO1 STEP-1:C=0:D$=""
260   Y=VAL(MID$(Y$,I,1))
270   IF Y=0 THEN 450
280   IF Y=1 THEN D$=X$:GOTO 330
290   FOR J=XLTO1 STEP-1
300     X=VAL(MID$(X$,J,1))
310     P=X*Y+C:GOSUB 500
320   NEXT
330   A$=C$(C)+D$+Q$(YL-I)
340
350   REM  ADDITION ROUTINE
360
370   ZL=LEN(Z$):AL=LEN(A$)
380   Z$=Q$(AL-ZL)+Z$:C=0:D$=""
390   FOR J=ALTO1 STEP-1
400     Z=VAL(MID$(Z$,J,1))
410     A=VAL(MID$(A$,J,1))
420     P=Z+A+C:GOSUB 500
430   NEXT
440   Z$=C$(C)+D$
450 NEXT
460
470 PRINT "Z = ";Z$
480 END
490
500 C=INT(P/10):D$=N$(P-C*10)+D$:RETURN
READY.
```

Program 2: Atari Version

```
130 REM :  INITIALIZATION
140 REM
150 DIM D$(99),X$(99),Z$(99),A$(99)
160 DIM Y$(99),T$(99)
170 PRINT "X=";:INPUT X$
175 PRINT "Y=";:INPUT Y$
180 XL=LEN(X$):YL=LEN(Y$)
190 I=XL+YL:DIM N$(10),Q$(I),C$(10)
200 FOR J=1 TO I:Q$(J,J)="0":NEXT J
210 N$="0123456789":C$=" 12345678 "
220 REM
230 REM :  MULTIPLICATION ROUTINE
240 REM
250 FOR I=YL TO 1 STEP -1:C=0:D$=""
260 Y=VAL(Y$(I,I))
270 IF Y=0 THEN 450
280 IF Y=1 THEN D$=X$:GOTO 330
290 FOR J=XL TO 1 STEP -1
300 X=VAL(X$(J,J))
310 P=X*Y+C:GOSUB 500
320 NEXT J
330 A$="":IF C THEN A$=C$(C+1,C+1)
335 A$(LEN(A$)+1)=D$:IF YL-I THEN A$(LEN
(A$)+1)=Q$(1,YL-I)
340 REM
350 REM :  ADDITION ROUTINE
360 REM
370 ZL=LEN(Z$):AL=LEN(A$)
380 T$=Z$:Z$="":IF AL-ZL THEN Z$=Q$(1,AL
-ZL)
385 Z$(LEN(Z$)+1)=T$:C=0:D$=""
390 FOR J=AL TO 1 STEP -1
400 Z=VAL(Z$(J,J))
410 A=VAL(A$(J,J))
420 P=Z+A+C:GOSUB 500
430 NEXT J
440 Z$="":IF C THEN Z$=C$(C+1,C+1)
445 Z$(LEN(Z$)+1)=D$
450 NEXT I
460 REM
470 PRINT "Z=";Z$
480 END
490 REM
500 C=INT(P/10):T$=D$:D$=N$(P-C*10+1,P-C
*10+1):D$(LEN(D$)+1)=T$:RETURN
```

```
VARIABLE TABLE FOR INFINI-MULT

X$      multiplicand
Y$      multiplier
Z$      product
A$      intermediate addend
D$      working string for creating
         A$ and Z$ from digits

X,Y     value of selected digit of X$,Y$
Z,A     .........................·Z$,A$
XL,YL   length of X$,Y$
ZL,AL   ......... Z$,A$
C       carry digit
P       result formed by numerical
         operations on digits
J,I     indices for selecting digits
         from strings

Q$()    zeroes for padding strings
N$()    decimal numerals
C$()    carry numerals, same as N$()
         except that C$(0)="" and N$(0)="0"
         and C$(9) is unneeded.
```


More About Numbers

Edmund N. Ricchezza

Certain numbers when acted upon in a certain manner will converge to a definite number. There are many of these, though the best known are Ulam's Conjecture, The 6174 Problem, and The Golden Mean.

Ulam's Conjecture states that any positive integer will always converge to "1" if acted upon as follows:

1) If it is even, divide by 2.
2) If it is odd, multiply by 3 and add 1.

For example, consider 15:

Since it is odd we multiply by 3 and add 1 obtaining	46
46 is even so we divide by 2 and obtain	23
Multiply 23 by 3 and add 1	70
Divide 70 by 2	35
Multiply 35 by 3 and add 1	106
Divide 106 by 2	53
Multiply 53 by 3 and add 1	160
Divide 160 by 2	80
Divide 80 by 2	40
Divide 40 by 2	20
Divide 20 by 2	10
Divide 10 by 2	5
Multiply 5 by 3 and add 1	16
Divide 16 by 2	8
Divide 8 by 2	4
Divide 4 by 2	2
Divide 2 by 2 QED	1

Programming this is elementary but it will reveal many curious situations. Consider this: 50,000,000 requires 106 operations to converge to 1 while the much smaller 63 requires 107 operations to converge to 1.

The "6174 Problem" states that any positive 4-digit integer except those with all the same digits will converge to "6174" when acted upon as follows:

1. Arrange the number in descending order.
2. Arrange the number in ascending order.
3. Subtract. Take this subtracted number as the new 4-digit number and repeat the process.

After not more than five iterations the difference will always be 6174. For example consider 4389:

9843	6543	8730	8532
-3489	-3456	- 378	-2358
6354	3087	8352	6174

This makes for an interesting and not too difficult program exercise.

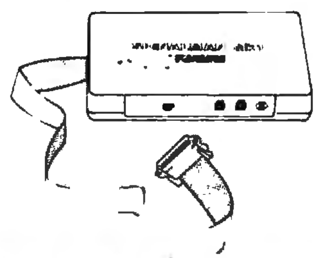
Quotients of successive terms of the Fibonacci Sequence will converge to .61803398. The ancient Greeks called this the "Golden Mean" because it expressed the ideal ratio of width to length that gave the most aesthetically appealing building or room.

I found this relationship fascinating considering that Fibonacci was born c. 1170, centuries after the Greeks discovered this number.

SPECIALS on INTEGRATED CIRCUITS

6502	7.45	10/6.95	50/6.55	100/6.15
6502A/6512A	8.40	10/7.95	50/7.35	100/6.90
6520 PIA	5.15	10/4.90	50/4.45	100/4.15
6522 VIA	6.45	10/6.10	50/5.75	100/5.45
6532	7.90	10/7.40	50/7.00	100/6.60
2114-L200		3.75	25/3.50	100/3.25
2114-L300		3.15	25/2.90	100/2.65
2716 EPROM		7.00	5/6.45	10/5.90
2532 EPROM				14.50
6116 Hitachi 2K x 8 CMOS RAM				**14.50**
4116				**8 for 17**
Zero Insertion Force 24 pin Socket				2.00
6550 RAM (PET 8K)				12.70
S-100 Wire Wrap Socket				2.40

A P Products 15% OFF
A P Hobby-Blox 15% OFF



MODEM SPECIAL $99

SIGNALMAN Mk1 from Anchor Automation

DIRECT CONNECT Modem with RS232 Cable and Connector included. Fully compatible with all Bell 103 modems. 0 to 300 bps, full duplex, frequency shift keyed modulation, automatic ANSW/ORIG selection, direct connect, audible tone carrier detect indicator, self-contained battery powered.

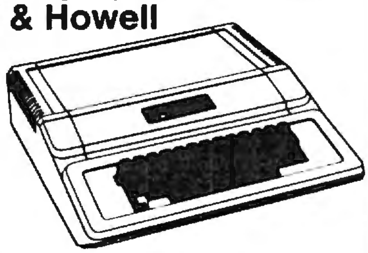
PET/CBM Version (Mk1P) $169

For Commodore Computers, the Signalman Mk1P includes connector, cable, and machine language software (parallel port).

STAR MODEM Prentice/Livermore Data Systems

RS232 MODEM	**SALE**	**$128**
IEEE 488 MODEM	**SALE**	**$199**
RS232 CCITT		**$170**
IEEE 488 CCITT		**$270**

We carry Apple II+ from Bell & Howell

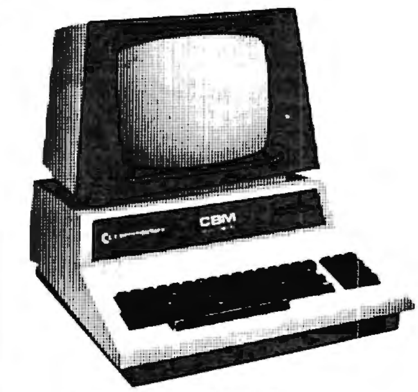


fullFORTH+ for Apple II **$75**

A full-featured FORTH with enhancements. Conforms to FORTH Interest Group standards.

REVERSAL (Spracklen) 32K Apple	28.00
Data Manager (Lutus) 24K Apple	40.00
Energy Miser - PET, Apple, Zenith	24.50
Histo-Graph (Boyd) 48K Apple	24.50
Data-Graph (Boyd) 48K Apple	40.00
Apple II User's Guide (Osborne)	12.00
Introduction to Pascal (Sybex)	10.30
Pascal Handbook (Sybex)	12.00
Musical Applications of Micros (Chamberlin)	20.00
Basic FORTRAN (Coan)	7.25
CP/M Handbook (with MP/M) Zaks	11.85
Programming the 6502 - Zaks	10.00

commodore



CBM-PET SPECIALS

	list	SALE
8023 Printer - 136 col, 150 cps bi-directional	(995)	775
8300 Daisy Wheel - 40 cps bi-directional	(2250)	1750
8032 80 x 25 CRT, business keyboard	(1495)	1100
Super Pet	(1995)	1650
8096 Board (extra 64K RAM for 8032)	(500)	400
8050 Dual Disk Drive - 1 megabyte	(1795)	1345
8250 Dual Disk Drive - 2 megabyte	(2195)	1760
CBM IEEE Modem	(395)	199
4016 full size graphics keyboard	(995)	795
4032 full size graphics keyboard	(1295)	999
4040 Dual Disk Drive - 330,000 bytes	(1295)	999
2031 Single Disk Drive - 165,000 bytes	(695)	550
4022 Tractor Feed Printer	(795)	630
C2N External Cassette Deck	(75)	65
VIC 20 Color Computer	(299)	259
VIC 1515 Graphic Printer	(395)	335
Used CBM/PET Computers		CALL
8024-7 High Speed Printer	(1995)	1345

WE WILL MATCH ANY ADVERTISED PRICE

*** EDUCATIONAL DISCOUNTS ***

Buy 2 PET/CBM Computers, receive 1 FREE

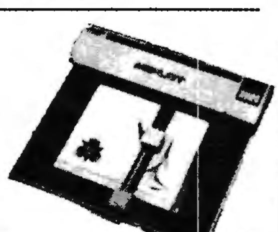
WordPro 3 Plus - 32K CBM, disk, printer	215
WordPro 4 Plus - 8032, disk, printer	325
OZZ Data Base System for CBM 8032	335
VISICALC for PET, ATARI, or APPLE	155
SM-KIT - Super PET ROM Utilities	**40**
Programmers Toolkit - PET ROM Utilities	35
PET Spacemaker II ROM Switch	36
2 Meter PET to IEEE or IEEE to IEEE Cable	40
Dust Cover for PET	7
IEEE-Parallel Printer Interface for PET	110
IEEE-RS232 Printer Interface for PET	120
The PET Revealed	17
Library of PET Subroutines	17

4 PART HARMONY MUSIC SYSTEM for PET

The Visible Music Monitor, by Frank Levinson, allows you to easily enter, display, edit, and play 4 part harmony music. Includes whole notes thru 64ths (with dotted and triplets), tempo change, key signature, transpose, etc. The KL-4M unit includes D to A converter and amplifier ready to hook to your speaker.

KL-4M Music Board with VMM Program **$60**

Watanabe Intelligent Plotter



WATANABE WX4671 Plotter	**1195**
WATANABE WX4675 6-pen Plotter	**1445**

DISK SPECIALS

SCOTCH (3M) 5"	10/2.75	50/2.65	100/2.60
SCOTCH (3M) 8"	10/2.80	50/2.70	100/2.65
Verbatim 8" Double Dens.	10/3.45	50/3.35	100/3.20
Verbatim 5" Datalife	10/2.45	50/2.40	100/2.35
(add 1.00 for Verbatim 5" plastic storage box)			
BASF 5" soft	10/2.40	20/2.35	100/2.30
Wabash 5" in Plastic Box	10/2.70	50/2.60	100/2.50
Wabash 8" in Plastic Box	10/2.75	50/2.65	100/2.55

WE STOCK MAXELL DISKS

Diskette Storage Pages		10 for 3.95
Disk Library Cases	8" - 2.85	5" - 2.15
Disk Hub Rings	8" - 50 @ 7.50	5" - 50 @ 6.00

CASSETTES - AGFA PE-611 PREMIUM

High output, low noise, 5 screw housings.

C-10	10/.56	50/.50	100/.48
C-30	10/.73	50/.68	100/.66

All other lengths available. Write for price list.

SPECIALS

EPSON MX-80 Printer	
EPSON MX-80 F/T Printer	
EPSON MX-70 Printer	
EPSON MX-100 Printer	
Centronics 739 Printer with dot graphics	675
STARWRITER Daisy Wheel Printer	**1445**
Zenith ZVM-121 Green Phosphor Monitor	115
Amdek Color Monitor	355

ALL BOOK and SOFTWARE PRICES DISCOUNTED

OSBORNE/McGraw-Hill, HAYDEN, SYBEX,etc.

Synertek Systems

SYM-1 Microcomputer SALE	**199**
SYM BAS-1 BASIC or RAE 1/2 Assembler	85
KTM-2/80 Synertek Video and Keyboard	349
KTM-3/80 Synertek Tubeless Terminal	385

ZENITH data systems

Z90-80 64 K	2170
Z90-82 64K, 1 double dens. drive	2395
Z89-0 48K	1950
Z89-1 48K, 1 drive	2150
Z67 10 Megabyte + Floppy Drive	4495
Z37 1.3 Megabyte Dual Floppy	1495
Z25 High Speed Printer	1195
Z19 Video Terminal (VT-52 compatible)	670
ZVM-121 Green Phosphor Monitor	119
All Zenith Software discounted	

ATARI SPECIALS

800 Computer	695	410 Recorder	69
400 - 16K	339	Pilot	68
810 Disk Drive	449	Microsoft BASIC	68
825 Printer	629	Educ. Series	20% off
850 Interface	175	**MISSILE COMMAND**	**32**
822 Printer	359	**ASTEROIDS**	**32**
Paddle Pair	17	**STAR RAIDERS**	**32**
Joystick Pair	17	Space Invaders	32
16K RAM	85	Music Composer	45
Assembler/Editor	46	Chess	30
TeleLink	20	Super Breakout	30

Write for prices on other Atari items.

252 Bethlehem Pike
Colmar, PA 18915

215-822-7727

A B Computers

WRITE FOR CATALOG

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

KMMM Pascal for PET/CBM — $85

A subset of standard Pascal with extensions.
- Machine language Pascal Source Editor with cursor oriented window mode
- Machine Language P-Code Compiler
- P-Code to machine language translator for optimized object code
- Run-time package
- Floating point capability
- User manual and sample programs

Requires 32K Please specify configuration.

EARL for PET (disk file based) — $65

Editor, Assembler, Relocater, Linker

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

RAM/ROM for PET/CBM

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

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

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

RAM/ROM - - 4K	$85
RAM/ROM - - 8K	120
Battery Backup Option	30

SUPERSORT by James Strasma — $35

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

SuperGraphics

by John Fluharty — **$30**

SuperGraphics provides machine language extensions to Commodore BASIC to allow fast and easy plotting and manipulation of graphics on the PET/CBM video display, as well as SOUND commands

Animations that previously were too slow or impossible without machine language subroutines now can be programmed directly in BASIC. Move blocks (or rocketships, etc.), or entire areas of the screen with a single, easy to use BASIC command. Scroll any portion of the screen up, down, left, or right. Turn on or off any of the 4000 (8000 on 8032) screen pixels with a single BASIC command. In high resolution mode, draw vertical, horizontal, and diagonal lines Draw a box, fill a box, and move it around on the screen with easy to use BASIC commands.

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

SuperGraphics commands include GRAPHIC, TEXT, RVS, SET, DRAW, FILL, PLOT, MOVE, PRINT, CSET, CMOVE, DISPLAY, PUT, SWAP, PAUSE, and SOUND.

Please specify machine type and ROM version, disk or tape.

FLEX-FILE for PET/CBM Computers

Self Calculating
DATA BASE
REPORT WRITER
MAILING LIST

FLEX-FILE is a set of flexible, friendly programs to allow you to set up and maintain a data base. Print files with a versatile Report Writer or a Mail Label routine. Programmers will find it easy to add subroutines to their own programs to make use of Data Base files.

RANDOM ACCESS DATA BASE

Record size limit is 250 characters. The number of records per disk is limited only by the size of each record and the number of records per disk is limited only by the size of each record and the amount of free space on the disk. File maintenance lets you step forward or backward through a file, add, delete, or change a record, go to a numbered record, or find a record by specified field. The Find command locates any record when you enter all (or a portion of) the desired key. Field lengths may vary from record to record to allow maximum packing of information. Files may be sorted by any field, and any field may be specified as a key. Sequential files from other programs may be converted to Flex-File format, and Flex-File records may be converted to sequential (WordPro, PaperMate, other word processors may also use Flex-File data). Maximum record size, fields per record, and order of fields may be changed at any time.

MAILING LABELS

With typical record size of 127 characters, each disk can handle over 1000 records (about 2800 with 8050 drive). Labels may be printed any number wide, and may begin in any column position. There is no limit on the number or order of fields on a label, and two or three fields may be joined together on one line (like first name, last name, and title). A "type of customer" field allows selective printing.

REPORT WRITER

Print any field in any column. For numeric fields, use decimal point justification (and round to any accuracy). Define any column as a series of mathematical functions performed on other columns. These functions include arithmetic operations and various log and trig functions. Pass results of operations such as running total from row to row. At the end of the report, print total and/or average for any column. Complete record selection, including field within range, pattern match, and logical functions can be specified individually or in combination with other parameters.

FLEX-FILE by Michael Riley — $60

Please specify equipment configuration when ordering.

Low Cost Disk Drive for PET/CBM

PEDISK II from cgrs Microtech is a new disk system ready to plug into your large keyboard PET/CBM.

PEDISK II offers speed, reliability, IBM compatibility.

Complete system prices with DOS and cable.

5" 40 track, 1 drive, 143K	$525
5" 40 track, 1 drive, 286K	690
8" IBM 3740 format, 77 track, 250K	995

PROGRAM YOUR OWN EPROMS

Branding Iron for PET/CBM — $79

EPROM Programmer with software for all ROM versions. Includes all hardware and software to program or copy 2716 and 2532 EPROMs.

CBM Software

Legal Time Accounting Package	445
Medical Accounting Package	
Complete CBM Business Software Package	
Can be tailored to meet most business requirements.	
Technician's Investment Analysis Package	500
Dow Jones Portfolio Management	129
Personal Tax Calculator	65
Tax Preparation System	445
Wordcraft 80 Wordprocessor Package	325
Pascal Development Package	235
Assembler Development Package	99
Intelligent Terminal Emulator	30
Softpac-1 (Competitive Software)	29

FORTH for PET

BY L. C. Cargile and Michael Riley — **$50**

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

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

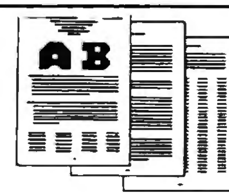
Available soon:

Metacompiler for FORTH — $30

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

PaperMate 60 COMMAND WORD PROCESSOR

by Michael Riley



Paper-Mate is a full-featured word processor for CBM/PET. Paper-Mate incorporates 60 commands to give you full screen editing with graphics for all 16K or 32K machines (including 8032), all printers, and disk or tape drives.

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

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

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

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

Paper-Mate functions with 16/32K CBM/PET machines, with any printer, and with either cassette or disk.

To order Paper-Mate, please specify configuration.

Paper-Mate on disk or tape — 40.00

BASIC INTERPRETER — $200

Designed to support the CBM 8096 (8032 with add-on 64K board). A full interpreter implementation to automatically take advantage of the extra memory available to the 8032.

BPI General Ledger — 8032/8050	300
BPI Accounts Receivable — 8032/8050	300
Hayden Software	
Complex Mathematics	12.70
Engineering Mathematics	12.70
General Mathematics	12.70
MCAP:Circuit Analysis Program	21.00
Energy Miser	24.50

CASH MANAGEMENT SYSTEM — $45

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

MICRO-REVERSI for PET by Michael Riley — **10**
super machine language version of Othello

Tunnel Vision / Kat & Mouse by Michael Riley — **10**
two excellent machine language maze programs

252 Bethlehem Pike
Colmar, PA 18915
215-822-7727

A B Computers

WRITE FOR CATALOG

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

Word Hunt

Robert W. Baker
Atco, NJ

This game is designed to test your ability to find specific words or letter sequences hidden in a 10 by 10 letter matrix. Scoring is based on the time it takes to enter your correct answer within a given time period determined by the skill level selected. The program uses very little memory and will easily run in 8K.

To play the game, first select the skill level you want to play at, between 1 and 5. One is the easiest, allowing the maximum time of approximately 1.5 minutes to find each word. Skill level 5, however, will only allow about 20 seconds to find each word.

Next enter a list of ten words, each being three to eight characters long. Only the letters *A* to *Z* may be entered, but they really do not have to be words. You can even use the game to make learning foreign languages more fun. If two people are playing, let one player enter the words for the other to find. Try to mix the word lengths, entering both long and short words for best results. If too many long words are entered it may take a while for the puzzle to be generated. If any word will not fit into the matrix, enter a new list of words when asked.

When the puzzle is ready, hit any key to start the game. Timing will start when the first word is shown.

Scoring for a correct answer is based on the amount of time it takes to respond, with 100 points maximum for each of ten words. If a correct answer is given in five seconds, you score 100 points. After that, your possible score decreases with time to a minimum of ten points for a correct answer. A wrong answer does not score any points and you only get *one* try for each word.

To enter your answer, you give a row and column number of the first character of the word followed by the direction code (see the diagram in the game!). Any invalid entries are discarded and you only type numbers, you do not type a comma or RETURN.

Before looking at how the program actually works, let's take a look at the major variables used in the program:

S – defines the size of the letter matrix to be created.

W – defines the number of words to be entered and used in the matrix.

M(S,S) – is the actual letter matrix, note that a floating point numeric matrix is used instead of a string matrix. More about this later.

W$(W) – contains the word list.

L(W,3) – remembers the starting location and direction of each word after it has been placed in the letter matrix. Each entry directly corresponds to the entry in the same position in the word matrix.

P(S,S) and F(8) – are working matricies used to create the actual letter matrix used in the game.

Now let's take a look at how the program works. First the program gets the desired skill level (SL) as a number between 1 and 5. The program sets a default value of 3 on the input line that the user can change before hitting the RETURN key. Lines 130-290 then get the list of words and check each is a valid character string (A-Z). The words are put into the word list in alphabetical order as each word is entered by the user. This avoids the time consuming process of sorting the entire word list at the end. In this way, there is a short delay as each word is entered. This short delay is not even noticeable by the user!

Line 340 initializes the latter matrix to all *'s (decimal value 42). Now each word in the word list is inserted randomly in the letter matrix in the following fashion:

1) The point matrix is cleared (line 360) so we can remember what points in the matrix have been tried for a particular word in the word list.

2) Lines 400-440 check that there is still at least one point in the letter matrix that has not been tried (entry in P is still 0). If all points have been tried, the user is asked to enter a new list of words since this list will not fit properly in the letter matrix.

3) A random starting point (that has not been tried) is chosen in line 450.

4) The starting point is flagged as having been tried (P value now 1) and then a check is made to see if the matrix position is open (still *) or matches the first letter of the word (lines 460-470).

5) Now the direction matrix (F) is cleared to remember what directions have been tried from this starting point (line 490).

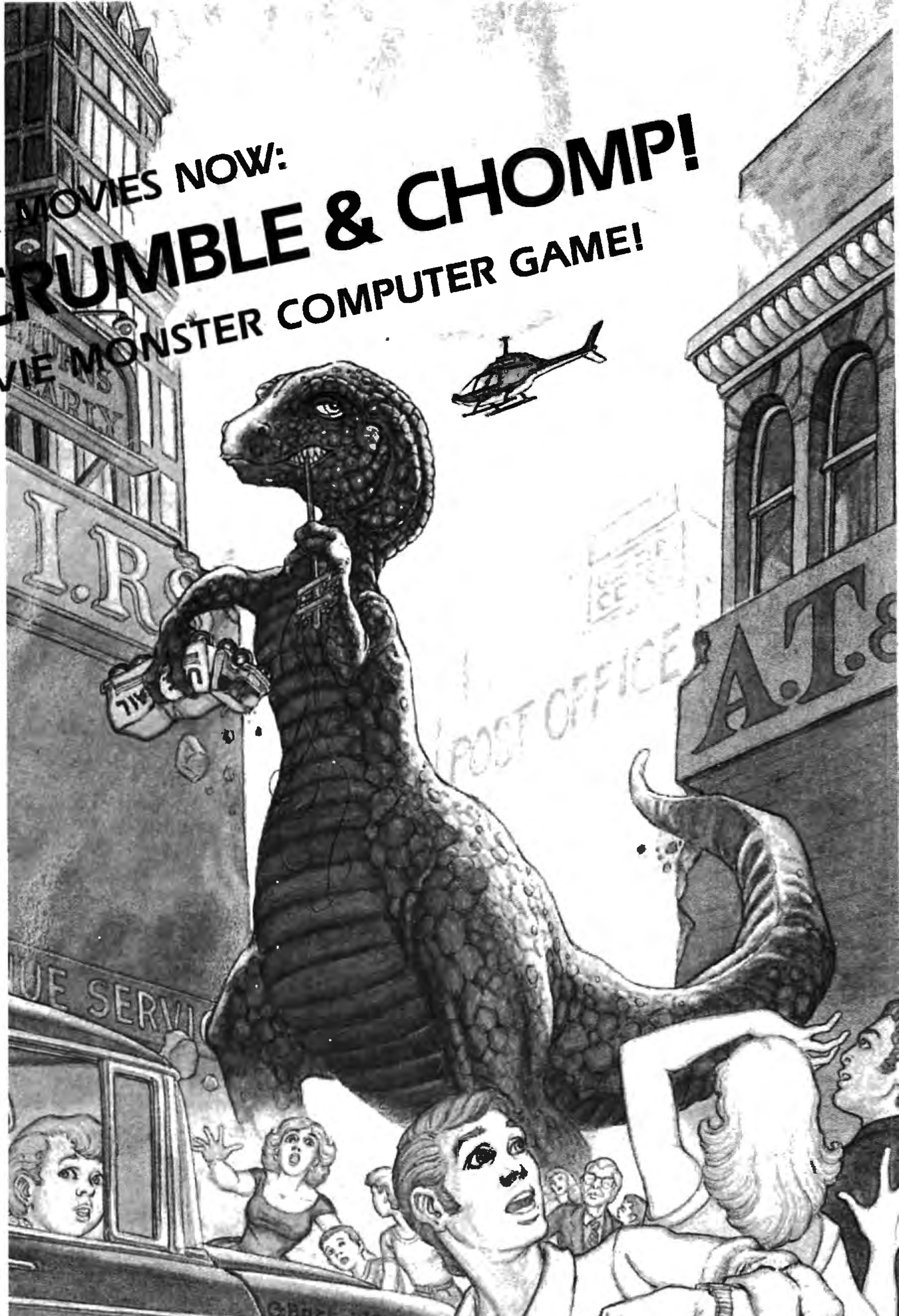
6) A check is made that at least one direction still hasn't been tried from this point (lines 500-510).

7) A random direction (that has not been tried) is chosen in line 520.

FIRST: MONSTER MOVIES NOW:

CRUSH, CRUMBLE & CHOMP!

THE GREAT MOVIE MONSTER COMPUTER GAME!



And guess who stars as the iovie monster. You! As any of six ifferent monsters. More if you ave the disk version.

You can terrorize and destroy our of the world's largest and iost densely populated cities in ver 100 possible scenarios. rom Tokyo to the Golden Gate, ou are the deadliest creature in ie air, on the land, or in the sea.

You can be the deadly amhibian who simultaneously nashes street cars, lunches on elpless humans and radiates a ay of death.

If you were a giant winged reature, think of the aerial ttacks you could make on the errified but tasty tidbits beneath ou.

But as in all the best monster iovies, you're up against everyning the human race can throw t you—even nuclear warheads nd a strange concoction develbed by a team of mad scientists.

For only $29.95 you get 6 stuendous monsters, each with its wn monstrous summary card, teeming metropoli displayed in raphic detail on your computer splay and mapped in the ccompanying 48-page illustrat-d book, the awesome sounds of onsterly mayhem, and spineigling, real-time, edge-of-yourat excitement.

GET CRUSH, CRUMBLE & CHOMP

iow at your local dealer for your APPLE, ATARI, or TRS-80 . . **before it's too late.**

EPYX
COMPUTER GAMES THINKERS PLAY

8) Then the word is checked to see if it can physically fit in the matrix in the selected direction from the current starting point (lines 530-650). This insures the word will not exceed the boundaries of the letter matrix from this point.

9) If the word can fit, then each character position in the selected direction is checked against the corresponding character of the word (lines 760-690). Each character in the matrix must match the corresponding character in the word or must be unused (still *).

10) If the word can be entered at this starting point and in this direction, each letter is inserted in the latter matrix (lines 710-720). Then the starting location and direction are saved for later use (line 740).

11) If the word will not fit, then the next direction is tried until all directions are exhausted from this point.

When all words have been put into the matrix, the remaining unused positions (still *) are filled in with random letters (lines 760-770).

Everything is now set to play the game, as soon as the player hits a key (lines 780-800). The letter matrix is displayed along with a direction code diagram and a score box (lines 820-960). A word is given to the player for him to find in the matrix and the timer is restarted (lines 970-1000). Then the program prompts the player for the starting location and direction code (lines 1020-1170). The values entered are then checked to see if correct, first against the values saved when the word was put into the matrix (lines 1190-1210). If the value does not match, then the program checks to see if a "double" was created when the unused positions were filled with random letters. Thus the program checks the player's answer again to insure it is right or wrong (lines 1230-1280). If a bad answer is entered, it is indicated and the correct answer is displayed with no score added (lines 1360-1430). A good answer is indicated and the appropriate score displayed and added to the player's total. The score is based on the selected skill level and the time it takes to enter the answer.

That's all there is to it! I should explain that a numeric vector was used for the actual letter matrix since it was easier and faster to use. Most people who have tried this game have found it to be very interesting and fun to play. At times it can even be educational. I only hope it's as much fun for you!

For those who might not want to type in the program, I'll supply copies on cassette tape for $2. Be sure to send requests to me and not through the magazine.

Program 1: Atari Version

```
10 REM **** WORD HUNT ****
20 REM
30 REM BY:ROBERT W. BAKER
40 REM ATCO, NJ
50 REM ******************
60 REM
70 OPEN #1,4,0,"K:"
80 S=10:W=10:DIM M(S,S),W$(W*10),LN(W),P
(S,S),L(W,3),F(8),R$(10),T$(10)
85 T$="          ":FOR I=0 TO 9:W$(I*10+
1,I*10+10)=T$:NEXT I
90 POKE 752,0:PRINT "{CLEAR}{DOWN}WHAT S
KILL LEVEL"
100 ? :? "1 (EASY) - TO - 5 - (HARD)?3{2
 LEFT}";
110 INPUT X:IF X<1 OR X>5 THEN 100
120 SL=6-X
130 ? "{2 DOWN}ENTER ";W;" WORDS,"
140 ? "Each 3 to 7 characters long{2 DOW
N}"
150 REM *** GET WORDS & PUT IN ORDER
160 REM *** LONGEST TO SHORTEST
170 FOR X=1 TO W:L(X,1)=0:L(X,2)=0:L(X,3
)=0
180 PRINT "WORD ";X;:INPUT R$
190 Q=LEN(R$)
200 IF Q<3 THEN ? "* TOO SHORT *":GOTO 1
80
210 IF Q>7 THEN ? "* TOO LONG  *":GOTO 1
80
220 X9=0:T$="*":T$(2)=R$:T$(LEN(T$)+1)="
*":FOR Y=1 TO Q:A=ASC(T$(Y+1,Y+1))
230 IF A<65 OR A>90 THEN X9=1:Y=Q
240 NEXT Y:IF X9=1 THEN PRINT "* BAD WOR
D *":GOTO 180
250 IF X=1 THEN T$=R$:T$(Q+1)="*":W$(X*1
0-9,X*10)=T$:LN(X)=Q+1:GOTO 290
260 X9=0:FOR Y=1 TO X-1:IF Q<=LN(Y)-1 TH
EN 280
270 FOR B=X TO Y+1 STEP -1:T$=W$((B-1)*1
0-9,(B-1)*10):W$(B*10-9,B*10)=T$:LN(B)=L
N(B-1):NEXT B
275 T$=R$:T$(Q+1)="*":W$(Y*10-9,Y*10)=T$
:LN(Y)=LEN(T$):Y=X-1
280 NEXT Y:IF X9=0 THEN T$=R$:T$(Q+1)="*
":W$(X*10-9,X*10)=T$:LN(X)=LEN(T$)
290 NEXT X
300 POKE 752,1:? "{CLEAR}{7 DOWN}That's
enough words!"
310 PRINT "{6 DOWN}Please be patient....
."
320 ? "{3 DOWN}              I'm now makin
g the puzzle!"
330 REM *** INITIALIZE LETTER MATRIX ***
```

TYPE RIGHT on your own MICROCOMPUTER

TYPE RIGHT is a complete course of 22 typing programs.

TYPE RIGHT has easy directions. The beginner will find all necessary instructions displayed on the screen, but there is extensive documentation for the programmer who wishes to make adaptations.

TYPE RIGHT is varied to keep your interest. You will be typing hundreds of real words, not just random letter drills. Your spelling and writing skills will improve. There are even five games!

TYPE RIGHT emphasizes accuracy first, then speed.

TYPE RIGHT computes words per minute the official way. (Errors are subtracted.)

TYPE RIGHT is a fantastic educational value at less than $2 per program.

BARRON ENTERPRISES
714 Willow Glen Road
Santa Barbara, CA 93105
(805) 687-5873

$29.95 PET (8K)
+ $1.50 shipping &
6% CA tax

☐ cassette ☐ diskette

SWIFTY SOFTWARE
TOP RATED PRODUCTS FOR ATARI

DEDICATED TO ATARI

NEW! SAVE $$

HARDWARE
DISK SENTRY TM
An intelligent digital accessory for your ATARI 810 Disk Drive, lets you selectively write data to both sides of single sided and write protected disks. DISK SENTRY cannot harm your drive or disks. Installs and removes easily; no soldering required. DISK SENTRY's LED signals system status, preventing accidental erasure of data. DISK SENTRY is a convenient push button write-protect override which can pay for itself with your first box of disks. $39.95 + $2.50 Shipping and Handling.

ARCADE GAMES
24K Disk; 16K Cassette; Joystick required
Add these popular HIGH RESOLUTION, REAL-TIME, ANIMATED games to your software arsenal. Get FAST ACTION and FULL SOUND GRAPHICS that take advantage of the unique features of your ATARI. Enjoy challenge that requires strategy and skill.

SPACE CHASE TM
Fly against intelligent invader clones. Arm yourself with Nuclear Defense Charges and play with or without Defense Shields. Enjoy this action-packed multicolor space odyssey. Displays top score, number of planets saved and number of galaxies conquered. $14.95 cassette; $19.95 disk

TIMEBOMB TM
Meet the challenge of this fast moving animated race against time, enemy aircraft and enemy bombs as you attempt to disarm timebombs set to explode ammunition depots. Avoid aircraft of varying sizes and speeds — and their bombs. Choose one of ten Day or Night Missions. Use from one to four Joysticks. Any number can play; top players listed on scoreboard. $14.95 cassette; $19.95 disk

NEW!

AND MORE GAMES.....
TRIVIA TREK TM
Unlimited fun and lots of laughs for one or two players. Five hundred questions and two thousand multiple choice answers are supplied on the master diskette. A powerful datafile handling program allows creation of your own trivia questions and answers. Features include: Player Missile Graphics, user or random selection of subjects and numerous comical answer choices. This DISK ONLY package comes complete with user instructions. An incredible value for only $29.95. Requires 32K and disk drive.

NEW!

FUN "n" GAMES #1 TM
WORDGAMES, POSSIBLE and LEAPFROG giving you hours of fun, challenge and entertainment. WORDGAMES, two games in one, contains GUESSIT - a deductive alphabetic reasoning game for one or two players and WORDJUMBLE - a multiple word descrambling puzzle with play-on-word hints and mystery answers. Instructions show how you can substitute your own words. Use POSSIBLE to help descramble word jumble puzzles or to create your own. All letter/number combinations or permutations of input are printed to screen or optional printer. LEAPFROG is a Chinese-Checker type jumping game in which you try to position two sets of animated jumping frogs in a minimum number of moves. 16K Cassette $19.95; 24K Disk $24.95. Disk version of GUESSIT works with VOTRAX Type "n" TALK. A real crowd pleaser.

COMING SOON! Space Shuttle Adventure Series TM
Real-time Space Flight Simulations

★

PERSONAL DATA MANAGEMENT
FILE-IT 2 TM
Contains all the programs in FILE-IT plus five additional file handling and financial programs. Financial entry and report generator programs create a powerful personal accounting system while two additional utility programs provide random access updating and user controlled record selection. Subfiles may be created, merged and sorted by any field. A monthly Bar Graph program generates a visual picture of financial data on the screen and/or printer. Supports up to four disk drives as well as the AXLON RAMDISK. Minimum requirements are 24K, 1 disk drive and an 80 column printer. Extensive documentation, supplied in a ring binder, provides clear instruction along with a tutorial on computer filing. $49.95 + $3.25 Shipping and Handling. AXLON RAMDISK not required.

FILE-IT TM
Use this start up database system to file and manage personal information and data. Create, sort, store and manipulate information such as appointment calendars, address, or telephone data, credit or charge records, stock investments, medical or prescription information, hobby, coupon or other types of collection information....and more. With printer you get 1 or 2 across mailing labels, disk jacket inventory covers and neatly written copy of all your data files. Comes with well documented instruction manual explaining basics of computer filing. Fast and easy to use. Holds over 300 records in 40K. Requires minimum of 24K and 1 disk drive. Printer optional. $34.95 (Disk Only)

COMING SOON! The Family Financier TM
AN easy to use financial package.

UTILITIES
DISKETTE INVENTORY SYSTEM TM
Use this system to gain control of your expanding disk/program inventory. Quickly get locations of single or multiple copies of your programs and all your valuable files. An invaluable tool, this system is easy and convenient to use and to update. 24K disk system required. $24.95 Printer suggested.

SWIFTY UTILITIES
A valuable collection of programming utilities for the ATARI programmer. This DISK ONLY package includes all of Programming Aids I and additional programs designed to make programming time more efficient. Special MENU program runs both saved and listed programs. REM REMOVER eliminates REM statements so programs take less core and run faster. PRINT 825 and PRINTEPS custom print programs prepare condensed, indented and paginated program listings on your ATARI 825 or EPSON MX-80 printer. Listings identify machine code, graphics and inverse video characters. VARIABLE LIST and VARIABLE PRINT programs help you prepare alphabetized annotated list of your program variables. A delete lines utility provides convenience of line deletion while a DOS CALLER gives you convenient access to many DOS utilities while your program is in core. Disklist prepares disk jacket labels. Many of these programs work coresident with each other and with your program. Disk Drive and minimum of 24K required. $29.95

PROGRAMMING AIDS PACKAGE I TM
Four utility programs to help increase programming efficiency and learn more about your computer. RENUMBER handles references and even variables. Generates Diagnostic Tables for programming error detection. PROGRAM DECODER, DECIMAL to BCD and BCD to DECIMAL programs give you a practical way of studying internal program representation and ATARI number-conversion procedures. Comes with comprehensive user's manual. 16K cassette $14.95; 24K disk $19.95

SWIFTY DATALINK TM
High Quality Smart Terminal Communications program. Easy to use Multi-Option, Menu Driven. Full performance uploading/downloading. Works in Duplex or Simplex modes supporting ASCII and ATASCII transmission. Printer Dump, Screen Dump and Disk Search options. Use as remote terminal. Send/receive and store programs and data files. Saves connect time charges with commercial services. Requires 24K RAM, 810 Disk Drive, 850 Interface or equivalent, 830 or other 300 Baud modem. (Printer optional) $39.95

NEW!

SWIFTY TACH MASTER TM
An accurate disk speed diagnostic utility program designed specifically for ATARI 810 Disk Drives. Provides easy-to-read visual indication of the speed of any drive connected to your system. Using the accuracy of machine language, DISK DOCTOR displays five RPM readings per second with a working tachometer accurate to ¼ RPM. Allows you to adjust your drive(s) to factory specs easily and at any time in the convenience of your own home. Comes complete with easy to follow user's manual. $29.95

NEW!

ACCESSORIES
VINYL DUST COVERS
New, glove soft, vinyl dust covers for the ATARI 800 Computer, the 400 Computer and the 825 Printer. Custom made from heavy duty upholstery grade vinyl, these covers completely cover the top and sides of your valuable equipment. Do not confuse them with cheap, flimsy plastic covers available elsewhere. Accessory ports and other input/output plugs are exposed for convenience of use. Available in either black or "ATARI" beige. ATARI 400: $9.95; ATARI 800: $10.95; ATARI 825: $10.95. Specify model and color. Any two covers for $18.95. Please include $2.50 for Shipping and Handling.

send check or money order to:

SWIFTY SOFTWARE, INC.
64 BROAD HOLLOW ROAD
MELVILLE, N.Y. 11747
(516) 549-9141
N.Y. Residents add 7¼% sales tax

send for free catalogue dealer orders and c.o.d.'s accepted

```
340 FOR X=1 TO S:FOR Y=1 TO S:M(Y,X)=42:
NEXT Y:NEXT X:Q=0
350 REM *** INIT POINT MATRIX & GET NEXT
 WORD
360 FOR X=1 TO S:FOR Y=1 TO S:P(Y,X)=0:N
EXT Y
370 NEXT X:Q=Q+1:IF Q>W THEN 760
380 G=LN(Q)-2
390 REM *** TRY ALL POINTS FOR EACH WORD
400 X9=0:FOR X=1 TO S:FOR Y=1 TO S:IF P(
Y,X)=0 THEN X9=1:X=S:Y=S
410 NEXT Y:NEXT X:IF X9=1 THEN 450
420 REM *** WORD WILL NOT FIT, TRY AGAIN
430 ? "{CLEAR}This list of words will no
t all fit."
440 ? :? "Please enter another list of w
ords!":GOTO 130
450 A=INT(S*RND(1)+1):B=INT(S*RND(1)+1):
IF P(B,A)<>0 THEN 450
460 P(B,A)=1:IF M(B,A)=42 THEN 490
470 IF M(B,A)<>ASC(W$(Q*10-9)) THEN 400
480 REM *** TRY ALL DIRECTIONS FROM THIS
 POINT
490 FOR X=1 TO 8:F(X)=0:NEXT X
500 X9=0:FOR X=1 TO 8:IF F(X)=0 THEN X9=
1:X=8
510 NEXT X:IF X9=0 THEN 400
520 D=INT(8*RND(1)+1):IF F(D)=1 THEN 520
530 F(D)=1:ON D GOTO 550,590,580,620,610
,650,640,560
540 REM *** CHECK WORD WILL FIT
550 IF (A+G)>S THEN 500
560 IF (B-G)<1 THEN 500
570 GOTO 670
580 IF (B+G)>S THEN 500
590 IF (A+G)>S THEN 500
600 GOTO 670
610 IF (A-G)<1 THEN 500
620 IF (B+G)>S THEN 500
630 GOTO 670
640 IF (B-G)<1 THEN 500
650 IF (A-G)<1 THEN 500
660 REM *** CHECK WORD MATCHES INTO MATR
IX
670 X=A:Y=B:X9=0:FOR N=2 TO G+1:GOSUB 15
50:IF M(Y,X)=42 THEN 690
680 IF M(Y,X)<>ASC(W$((Q-1)*10+N)) THEN
X9=1:N=G+1
690 NEXT N:X=A:Y=B:IF X9=1 THEN 500
700 REM *** ENTER WORD
710 FOR N=1 TO G+1:IF M(Y,X)=42 THEN M(Y
,X)=ASC(W$((Q-1)*10+N))
720 GOSUB 1550:NEXT N
730 REM *** SAVE START & DIRECTION INFO
740 L(Q,1)=A-1:L(Q,2)=B-1:L(Q,3)=D:IF D<
W THEN 360
750 REM *** FILL IN SPACES
760 FOR Y=1 TO S:FOR X=1 TO S:IF M(Y,X)=
42 THEN M(Y,X)=INT(25*RND(1)+65)
770 NEXT X:NEXT Y:WP=0:TS=0
780 ? "{CLEAR}{10 DOWN}                IR
EADY!"
790 ? "{6 DOWN}Depress any key when read
y to play!"
800 IF PEEK(764)=255 THEN 800
805 POKE 764,255
810 REM *** SET UP DISPLAY
820 ? "{CLEAR}{DOWN}     |COLUMN|";:POKE
85,26:? "|W O R D|"
830 REM *** PRINT 'ROW' DOWN LEFT COLUMN
840 REM *** START OUT DOWN 4
850 REM *** LATER DO 5 UP & 3 RIGHT
860 ? "{4 DOWN}|R|{DOWN}{LEFT}|O|{DOWN}{
LEFT}|W|{5 UP}{3 RIGHT}";
870 FOR X=0 TO S-1:? X;:NEXT X:? :Y=1:GO
SUB 1650
880 FOR Y=1 TO S:? "{2 RIGHT}";Y-1;"{=}"
;
890 FOR X=1 TO S:? CHR$(M(Y,X));:NEXT X
900 ? "{=}":NEXT Y:Y=0:GOSUB 1650
910 ? :? "|DIRECTIONS:|":? "{DOWN}  7 8
1"
920 ? "    {G}{=}{F}":? "  6{R}{T}{R}2":?
 "    {F}{=}{G}":? "  5 4 3"
930 G=16:GOSUB 1700:? "       |  SCORE  |"
:POKE 85,25:? "{V}          {B}"
940 POKE 85,25:? "{V}   0     {B}"
950 POKE 85,25:? "{V}          {B}"
955 POKE 85,25:? "{9 M}"
960 G=0:GOSUB 1700:? "
 ":REM <-- 19 SPACES
970 WP=WP+1:IF WP>W THEN 1450
980 Q=LN(WP)-1
990 REM *** NEXT WORD
1000 GOSUB 1700:POKE 85,29-(Q/2):? W$((W
P-1)*10+1,(WP-1)*10+Q)
1005 POKE 20,0:POKE 19,0:REM KILL RTCLK
1010 REM *** GET START LOC
1020 G=3:GOSUB 1700:? "STARTING LOCATION
":POKE 85,20:? "(ROW,COLUMN):"
1030 FOR G=6 TO 14:GOSUB 1700
1040 ? "                   ":NEXT G:G=6:
GOSUB 1700:REM <-- 19 SPACES
1050 GET #1,B:IF B=155 THEN 1050
1070 PRINT CHR$(B);",";:IF B=48 THEN B=0
:GOTO 1090
1080 B=B-48:IF B<1 OR B>9 THEN PRINT "{2
 BACK S}";:GOTO 1050
1090 GET #1,A
1100 IF A=155 THEN 1090
1110 PRINT CHR$(A);:IF A=48 THEN A=0:GOT
```

```
0 1140
1120 A=A-48:IF A<1 OR A>9 THEN 1030
1140 G=8:GOSUB 1700:PRINT "DIRECTION:":?
 :POKE 85,20:? " {LEFT}";
1150 GET #1,D
1160 IF D=155 THEN 1150
1170 PRINT CHR$(D);:D=D-48:IF D<1 OR D>8
 THEN 1140
1180 REM *** CHK IF GOOD INFO INPUT
1190 WT=PEEK(20)+256*PEEK(19):IF B<>L(WP
,2) THEN 1230
1210 IF D=L(WP,3) THEN 1360
1220 REM *** CHK IF A DOUBLE MAY EXIST
1230 X=A+1:Y=B+1:G=LN(WP)-1:IF M(Y,X)<>A
SC(W$(WP*10-9)) THEN 1300
1240 X9=0:FOR N=2 TO G:GOSUB 1550:IF X<1
 OR X>10 THEN 1270
1250 IF Y<1 OR Y>10 THEN 1270
1260 IF M(Y,X)=ASC(W$((WP-1)*10+N)) THEN
 1280
1270 X9=1:N=G
1280 NEXT N:IF X9=0 THEN 1360
1290 REM *** BAD START/DIR - NO SCORE
1300 G=6:GOSUB 1700:PRINT "
 ";:B=L(WP,2):A=L(WP,1):REM 14 SPACES
1310 ? B;",";A
1320 G=10:GOSUB 1700:? "              ";L
(WP,3):REM 13 SPACES
(WP,3):REM 13 SPACES
1330 G=12:GOSUB 1700:? "{ESC}{UP}
      {ESC}{UP}":REM 13 SPACES
1340 G=13:GOSUB 1700:? "{Z} |NO|, CORREC
T {C}":GOTO 1420
1350 REM *** GOOD ANSWER - GET SCORE
1360 IF WT<(SL*60) THEN WS=100:GOTO 1390
:REM <-- MAX SCORE
1370 IF WT>(SL*1200) THEN WS=10:GOTO 139
0:REM <-- MIN SCORE
1380 WS=5+INT(((SL*1200)-WT)/60)
1390 G=12:GOSUB 1700:? "{ESC}{UP}"
1400 G=13:GOSUB 1700:? "{Z} |YES|, ";WS;
" POINTS":TS=TS+WS
1410 REM *** UPDATE TOTAL SCORE
1420 G=18:GOSUB 1700:? "{8 RIGHT}";TS
1430 FOR X=1 TO 500:NEXT X:GOTO 960
1440 REM *** END GAME ***
1450 POSITION 2,15
1460 FOR X=1 TO 8:? "            ":NEXT
X:REM <-- 12 SPACES
1470 FOR G=-2 TO 14:GOSUB 1700
1480 PRINT "                 ":NEXT G:RE
M 17 SPACES
1490 POSITION 2,15:? "PLAY AGAIN (Y OR N
) ?"
1500 GET #1,R
1510 IF R=ASC("Y") THEN 90
1520 IF R<>ASC("N") THEN 1500
1530 END
1540 REM *** SUBR TO INC COORDINATES IN
DIR
1550 ON D GOTO 1560,1570,1580,1590,1600,
1610,1620,1630
1560 Y=Y-1
1570 X=X+1:RETURN
1580 X=X+1
1590 Y=Y+1:RETURN
1600 Y=Y+1
1610 X=X 1:RETURN
1620 X=X-1
1630 Y=Y-1:RETURN
1640 REM *** SUBR FOR BOT TOP/BOTTOM
1650 PRINT "{3 RIGHT}";:IF Y=1 THEN ? "{
Q}";:GOTO 1670
1660 PRINT "{Z}";
1670 FOR X=0 TO S-1:? "{R}";:NEXT X:IF Y
=1 THEN PRINT "{E}":RETURN
1680 ? "{C}":RETURN
1690 REM *** SUBR TO POSITION
1700 POSITION 20,G+2:RETURN
```

Program 2: Microsoft Version

```
10 REM ***** W O R D   H U N T ***
   **
20 REM
30 REM BY: ROBERT W. BAKER, ATCO, ~
   NJ
40 REM
50 REM ****************************
   **
60 :
70 FOR X=1 TO VAL(RIGHT$(TI$,2)) :
   R=RND(1) :NEXT :POKE 59468
   ,12
80 S=10 :W=10 :DIM M(S,S),W$(W),P(
   S,S),L(W,3),F(8)
90 PRINT"{CLEAR}{DOWN}WHAT SKILL L
   EVEL"
100 PRINT :PRINT"1 (EASY) - TO - 5 ~
    (HARD)    3{03 LEFT}";
110 INPUT R$ :X=VAL(R$) :IF X<1 OR ~
    X>5 THEN 90
120 SL=6-X
130 PRINT"{02 DOWN}ENTER"W"WORDS,"
140 PRINT :PRINT"EACH 3 TO 8 CHARAC
    TERS LONG{02 DOWN}
150 REM **** GET WORDS & PUT IN ORD
    ER
160 REM ****   LONGEST TO SHORTEST
170 FOR X=1 TO W :L(X,1)=0 :L(X,2)=
    0 :L(X,3)=0
180 PRINT"WORD";X;TAB(8);"{02 RIGHT
```

```
     RIGHT}?{03 LEFT}";
190  INPUT R$ :Q=LEN(R$)
200  IF Q<3 THEN PRINT TAB(26);"{UP}
     * TOO SHORT *" :GOTO 180
210  IF Q>8 THEN PRINT TAB(26);"{UP}
     * TOO LONG  *" :GOTO 180
220  X9=0 :FOR Y=1 TO Q :A=ASC(MID$(
     "*"+R$+"*",Y+1,1))
230  IF A<65 OR A>90 THEN X9=1 :Y=Q
240  NEXT Y :IF X9=1 THEN PRINT TAB(
     26);"{UP}* BAD WORD  *" :G
     OTO 180
250  IF X=1 THEN W$(X)=R$+"*" :GOTO ~
     290
260  X9=0 :FOR Y=1 TO X-1 :IF Q<=LEN
     (W$(Y))-1 THEN 280
270  FOR B=X TO Y+1 STEP -1 :W$(B)=W
     $(B-1) :NEXT B :W$(Y)=R$+"
     *" :X9=1 :Y=X-1
280  NEXT Y :IF X9=0 THEN W$(X)=R$+"
     *"
290  NEXT X
300  PRINT"{CLEAR}{07 DOWN}";SPC(8);
     "THAT'S ENOUGH WORDS!" :RE
     M <-- 7 DOWN
310  PRINT"{06 DOWN}PLEASE BE PATIEN
     T ....." :REM <-- 6 DOWN
320  PRINT"{03 DOWN}";SPC(12);"I'M N
     OW MAKING THE PUZZLE!
330  REM **** INITIALIZE LETTER MATR
     IX
340  FOR X=1 TO S :FOR Y=1 TO S :M(Y
     ,X)=42 :NEXT Y :NEXT X :Q=
     0
350  REM **** INIT POINT MATRIX & GE
     T NEXT WORD
360  FOR X=1 TO S :FOR Y=1 TO S :P(Y
     ,X)=0 :NEXT Y
370  NEXT X :Q=Q+1 :IF Q>W THEN 760
380  G=LEN(W$(Q))-2
390  REM **** TRY ALL POINTS FOR EAC
     H WORD
400  X9=0 :FOR X=1 TO S :FOR Y=1 TO ~
     S: IF P(Y,X)=0 THEN X9=1 :
     X=S :Y=S
410  NEXT Y :NEXT X: IF X9=1 THEN 45
     0
420  REM **** WORD WILL NOT FIT, TRY
      AGAIN!
430  PRINT"{CLEAR}THIS LIST OF WORDS
      WILL NOT ALL FIT
440  PRINT :PRINT"PLEASE ENTER ANOTH
     ER LIST OF WORDS !" :GOTO ~
     130
450  A=INT(S*RND(1)+1) :B=INT(S*RND(
     1)+1) :IF P(B,A)<>0 THEN 4
     50
460  P(B,A)=1 :IF M(B,A)=42 THEN 490
470  IF M(B,A) <> ASC(LEFT$(W$(Q),1)
     ) GOTO 400
480  REM **** TRY ALL DIRECTIONS FRO
     M THIS POINT
490  FOR X=1 TO 8 :F(X)=0 :NEXT X
500  X9=0 :FOR X=1 TO 8 :IF F(X)=0 T
     HEN X9=1 :X=8
510  NEXT X :IF X9=0 THEN 400
520  D=INT(8*RND(1)+1) :IF F(D)=1 GO
     TO 520
530  F(D)=1 :ON D GOTO 550,590,580,6
     20,610,650,640,560
540  REM **** CHECK WORD WILL FIT
550  IF (A+G)>S THEN 500
560  IF (B-G)<1 THEN 500
570  GOTO 670
580  IF (B+G)>S THEN 500
590  IF (A+G)>S THEN 500
600  GOTO 670
610  IF (A-G)<1 THEN 500
620  IF (B+G)>S THEN 500
630  GOTO 670
640  IF (B-G)<1 THEN 500
650  IF (A-G)<1 THEN 500
660  REM **** CHECK WORD MATCHES INT
     O MATRIX
670  X=A :Y=B :X9=0 :FOR N=2 TO G+1 ~
     :GOSUB 1550 :IF M(Y,X)=42 ~
     GOTO 690
680  IF M(Y,X)<>ASC(MID$(W$(Q),N,1))
      THEN X9=1 :N=G+1
690  NEXT N :X=A :Y=B :IF X9=1 THEN ~
     500
700  REM **** ENTER WORD
710  FOR N=1 TO G+1 :IF M(Y,X)=42 TH
     EN M(Y,X)=ASC(MID$(W$(Q),N
     ,1))
720  GOSUB 1550 :NEXT N
730  REM **** SAVE START & DIRECTION
      INFO
740  L(Q,1)=A-1 :L(Q,2)=B-1 :L(Q,3)=
     D :IF Q<W THEN 360
750  REM **** FILL IN SPACES
760  FOR Y=1 TO S :FOR X=1 TO S :IF ~
     M(Y,X)=42 THEN M(Y,X)=INT(
     25*RND(1)+65)
770  NEXT X :NEXT Y :WP=0 :TS=0
780  PRINT"{CLEAR}{10 DOWN}";TAB(15)
     ;"{REV}READY" :REM <-- 10 ~
     DOWN
790  PRINT"{06 DOWN}  DEPRESS ANY KE
     Y WHEN READY TO PLAY !" :R
     EM <-- 6 DOWN
```

```
800 R$="" :GET R$ :IF R$="" THEN 80
    0
810 REM **** SET UP DISPLAY
820 PRINT"{CLEAR}{DOWN}      {REV}COL
    UMN";TAB(25);"{REV}W O R D
    "
830 REM *** PRINT 'ROW' DOWN LEFT C
    OLUMN
840 REM *** START OUT DOWN 4
850 REM *** LATER DO 5 UP & 3 OVER
860 PRINT"{04 DOWN}{REV}R{DOWN}{LEF
    LEFT}O{DOWN}{LEFT}W{05 UP}
    {03 RIGHT}{OFF}";
870 FOR X=0 TO S-1 :PRINT RIGHT$(ST
    R$(X),1); :NEXT X :PRINT :
    Y=1 :GOSUB 1650
880 FOR Y=1 TO S :PRINT"{02 RIGHT}"
    ;RIGHT$(STR$(Y-1),1);"]";
890 FOR X=1 TO S :PRINT CHR$(M(Y,X)
    ); :NEXT X
900 PRINT"]" :NEXT Y :Y=0 :GOSUB 16
    50
910 PRINT :PRINT"{REV}DIRECTIONS:" ~
    :PRINT"{DOWN}   7 8 1"
920 PRINT"   M]N" :PRINT"   6@Q@2" :
    PRINT"N]M":PRINT"   5 4 3
930 G=16 :GOSUB 1700 :PRINT"      {R
    REV}  SCORE  " :PRINT TAB(
    25);"5       6
940 PRINT TAB(25);"5   0     6"
950 PRINT TAB(25);"5         6" :PRIN
    T TAB(25);"888888888
960 G=0 :GOSUB 1700 :PRINT"          ~
                   " :REM <-- 19 S
    PACES
970 WP=WP+1 :IF WP>W THEN 1450
980 Q=LEN(W$(WP))-1
990 REM **** NEXT WORD
1000 GOSUB 1700 :PRINT TAB(29-(Q/2))
     ;LEFT$(W$(WP),Q) :TI$="000
     000"
1010 REM **** GET START LOC
1020 G=3 :GOSUB 1700 :PRINT"STARTING
     LOCATION" :PRINT TAB(20);
     "(ROW,COLUMN):
1030 FOR G=6 TO 14 :GOSUB 1700
1040 PRINT"                    " :NE
     XT G :G=6 :GOSUB 1700 :REM
     <-- 20 SPACES
1050 B$="" :GET B$ :IF B$="" THEN 10
     50
1060 IF ASC(B$)=13 THEN 1050
1070 PRINT B$;","; :IF B$="0" THEN B
     =0 :GOTO 1090
1080 B=VAL(B$) :IF B<1 OR B>9 THEN]P
     RINT"{02 LEFT}  {02 LEFT}"
     ; :GOTO 1050
1090 A$="" :GET A$ :IF A$="" THEN 10
     90
1100 IF ASC(A$)=13 THEN 1090
1110 PRINT A$ :IF A$="0" THEN A=0 :G
     OTO 1140
1120 A=VAL(A$) :IF A<1 OR A>9 THEN 1
     030
1130 REM **** GET DIRECTION
1140 G=8 :GOSUB 1700 :PRINT"DIRECTIO
     N:" :PRINT :PRINT TAB(20);
     " {LEFT}";
1150 GET D$ :IF D$="" THEN 1150
1160 IF ASC(D$)=13 THEN 1150
1170 PRINT D$ :D=VAL(D$) :IF D<1 OR ~
     D>8 THEN 1140
1180 REM **** CHK IF GOOD INFO INPUT

1190 WT=TI :IF B<>L(WP,2) THEN 1230
1200 IF A<>L(WP,1) THEN 1230
1210 IF D=L(WP,3) THEN 1360
1220 REM **** CHK IF A DOUBLE MAY EX
     IST
1230 X=A+1 :Y=B+1 :G=LEN(W$(WP))-1 :
     IF M(Y,X)<>ASC(LEFT$(W$(WP
     ),1)) THEN 1300
1240 X9=0 :FOR N=2 TO G :GOSUB 1550 ~
     :IF X<1 OR X>10 THEN 1270
1250 IF Y<1 OR Y>10 THEN 1270
1260 IF M(Y,X)=ASC(MID$(W$(WP),N,1))
      THEN 1280
1270 X9=1 :N=G
1280 NEXT N :IF X9=0 THEN 1360
1290 REM **** BAD START/DIR - NO SCO
     RE
1300 G=6 :GOSUB 1700 :PRINT SPC(14);
     :B$=STR$(L(WP,2)) :A$=STR
     $(L(WP,1))
1310 PRINT RIGHT$(B$,LEN(B$)-1);",";
     RIGHT$(A$,LEN(A$)-1)
1320 G=10 :GOSUB 1700 :PRINT SPC(13)
     ;L(WP,3)
1330 G=12 :GOSUB 1700 :PRINT"^";SPC(
     13);"^"
1340 G=13 :GOSUB 1700 :PRINT"J {REV}
     NO{OFF}, CORRECT K" :GOTO ~
     1420
1350 REM **** GOOD ANSWER - GET SCOR
     E
1360 IF WT<(SL*60) THEN WS=100 :GOTO
      1390 :REM <-- MAX SCORE
1370 IF WT>(SL*1200) THEN WS=10 :GOT
     O 1390 :REM <-- MIN SCORE
1380 WS=5+INT(((SL*1200)-WT)/60)
1390 G=12 :GOSUB 1700 :PRINT"^"
1400 G=13 :GOSUB 1700 :PRINT"J {REV}
```

```
    YES{OFF},";WS;"POINTS" :TS
    =TS+WS
1410 REM **** UPDATE TOTAL SCORE
1420 G=18 :GOSUB 1700 :PRINT"{07 RI
    G RIGHT}";TS
1430 FOR X=1 TO 1000 :NEXT X :GOTO
    960
1440 REM **** END GAME ****
1450 PRINT"{HOME}{15 DOWN}" :REM <-
    - DOWN 15
1460 FOR X=1 TO 6 :PRINT"
    ~   " :NEXT X :REM <-- 12 SPA
    CES
1470 FOR G=-2 TO 13 :GOSUB 1700
1480 PRINT"                   " :NEXT
    ~G :REM 17 SPACES
1490 PRINT"{HOME}{20 DOWN}PLAY AGAI
    N (Y OR N) ?" :REM <-- DOWN
     20
1500 R$="" :GET R$ :IF R$="" THEN 1
    500
1510 IF R$="Y" THEN 90
1520 IF R$<>"N" THEN 1500
1530 END
1540 REM **** SUBR TO INCR COORDINA
    TES IN DIR
1550 ON D GOTO 1560,1570,1580,1590,
    1600,1610,1620,1630
1560 Y=Y-1
1570 X=X+1 :RETURN
1580 X=X+1
1590 Y=Y+1 :RETURN
1600 Y=Y+1
1610 X=X-1 :RETURN
1620 X=X-1
1630 Y=Y-1 :RETURN
1640 REM **** SUBR FOR BOX TOP/BOTT
    OM
1650 PRINT"{03 RIGHT}"; :IF Y=1 THE
    N PRINT"0"; :GOTO 1670
1660 PRINT"-";
1670 FOR X=0 TO S-1 :PRINT"@"; :NEX
    T X :IF Y=1 THEN PRINT"." :
    RETURN
1680 PRINT"=" :RETURN
1690 REM **** SUBR TO POSITION
1700 PRINT"{HOME}"TAB(20); :FOR X9=
    1 TO G+3 :PRINT"{DOWN}"; :N
    EXT X9 :RETURN
```

©

Count The Hearts

Christopher J. Flynn
Herndon, VA

"Count the Hearts" is a VIC program which will help you to develop your child's counting skills. VIC will display a certain number of hearts on your television screen. Ask your child to count them. If your child can correctly count the hearts, he or she will be rewarded by a duet of chirping birdies. Otherwise, ... To challenge older children, you can place a limit on the time VIC allows for a response.

Once it is set up, no reading is required to play "Count the Hearts." However, preschoolers will probably need you to help them with the keyboard.

Setting Up

When you first start "Count the Hearts," it will ask you for a range of numbers and a time limit.

You can tailor the game to your child's counting skills by trying different number ranges. For example, you may want to start with numbers between one and five. Gradually, a child will work up to counting up to ten. If you notice difficulties with some numbers, you might want to work within that range (say from six to nine).

Here is how VIC will ask you to set the number range:

1. VIC will display:
ENTER NUMBER RANGE
LOW NUMBER (1) ?

2. You should type in the low number in the range (don't forget to hit RETURN). If you just hit RETURN, VIC will use one as the low number.

3. Next VIC will ask:
HIGH NUMBER (9) ?

4. Now type in the high number. Again, if you just hit RETURN, VIC will use nine as the high number.

VIC will make sure that your low number is really lower than your high number. It will also make sure that neither number is greater than 484. Why 484? Well, that's how many spaces are left on the screen for displaying the hearts.

The time limit gives you a way to speed up "Count the Hearts." If you don't take a guess at how many hearts there are within the time limit, then VIC will let you know that time's up. VIC will then just start another game.

VIC will ask you for the time limit:

5. VIC will display:
TIME LIMIT PER SET
SECONDS (120) ?

6. Type in the number of seconds you want to use for the time limit. If you just type RETURN, VIC will set the time limit to 120 seconds or two minutes.

By the way, if, in any of the above steps, VIC didn't understand your response, it will either ask the question again or ask you to repeat your response.

Counting Hearts

O.K. The television screen goes blank for just an instant. In that brief instant VIC is deciding how many hearts it will ask you to count. Then, one by one, VIC displays the hearts at random locations on the television screen. As it shows each new heart, VIC says in a deep voice, "BEEP!". Notice how VIC paints the hearts in different colors.

Now VIC will ask:

HOW MANY HEARTS ?

Ask your child to count them. Type in the number (don't forget RETURN!) and see what happens. What happens if your child gives the right answer? How about a wrong answer? What is your child's reaction?

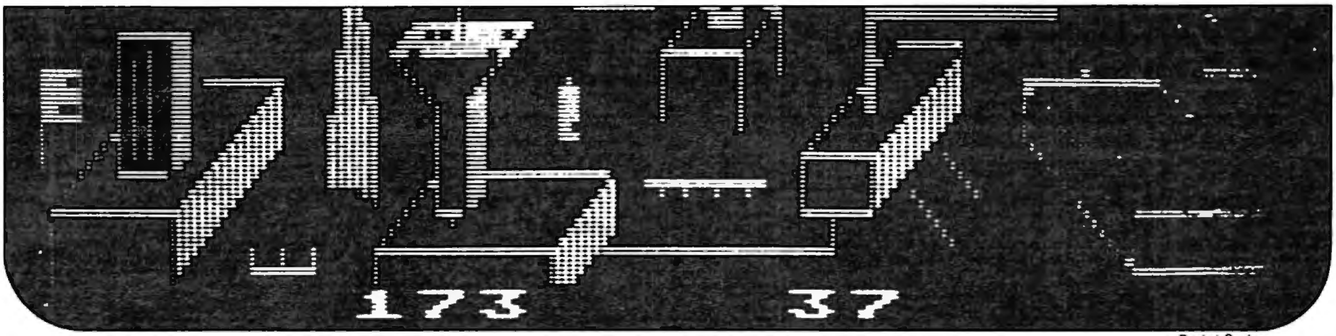
VIC will start a new game when the right answer is typed in or when time runs out and nothing has been heard from the keyboard. VIC is very patient with small folks learning to count. When a wrong answer is given, VIC just resets its timer and gives you another try.

Scoring

When you are finished playing "Count the Hearts", just hit the F1 key in reply to the "HOW MANY HEARTS ?" question. VIC will promptly clear the screen and tell you:

– how many games were played
– how many correct answers there were
– how many wrong answers there were
– how many times the player ran out of time

By keeping track of the number range (VIC shows you the range you used) and the scores, you can note your child's progress. For example, do you notice a little slowness in your child's learning to count past ten? We did. That seems to be the upper limit for our our three year old for a while.



Rocket Raiders

Our High Quality Software Is More Than A Stroke Of Genius... It's A Work Of Art.

□ **PM EDITOR:** by Dennis Zander (Atari, 16K)
Create your own fast action graphics game for the Atari 400 or 800 using its player missile graphics features. By using player data stored as strings, players can be moved or changed (for animation) at machine language speed. All this is done with string variables (POS(Y)=SHIP4). This program is designed to permit creation of up to 4 players on the screen, store them as string data and then immediately try them out in the demo game included in the program. Instructions for use in your own game are included. PM EDITOR was used to create the animated characters in **ARTWORX** RINGS OF THE EMPIRE and ENCOUNTER AT QUESTAR IV.
PRICE . . . . **$29.95** cassette **$33.95** diskette

□ **ROCKET RAIDERS** by Richard Petersen (Atari 24K)
Defend your asteroid base against pulsar bombs, rockets, lasers, and the dreaded "stealth saucer" as aliens attempt to penetrate your protective force field. Precise target sighting allows you to fire at the enemy using magnetic impulse missiles to help protect your colony and its vital structures.
PRICE **$19.95** cassette **$23.95** diskette

□ **INTRUDER ALERT!** by Dennis Zander (Atari, 16K)
This is a fast paced action game in which you must escape from the "Dreadstar" with the secret plans. The droids are after you and you must find and enter your ship in order to escape. If you fail, the rebellion is doomed. **PRICE** . . . **$16.95** cassette **$20.95** diskette

□ **THE RINGS OF THE EMPIRE:** by Dennis Zander (Atari 16K)
The Empire has developed a series of battle stations protected by one or more rings of energy. You must destroy these weapons by attacking them in your Y-wing fighter armed with Zydon torpedoes. Each time you blast through the rings and destroy the station, the Empire develops a new station with more protective rings.
PRICE . . . . **$16.95** cassette **$20.95** diskette

□ **FOREST FIRE!:** by Richard Petersen (Atari, 24K)
Using excellent color graphics, your Atari is turned into a fire scanner to help you direct operations to contain a forest fire. You must compensate for changes in wind, weather and terrain. Not protecting valuable property can result in startling penalties. Life-like variables make FOREST FIRE a very suspenseful and challenging simulation. **PRICE** . **$16.95** cassette **$20.95** diskette

□ **PILOT:** by Michael Piro (Atari, 16K)
Pilot your small airplane to a successful landing using both joysticks to control throttle and attack angle. PILOT produces a true perspective rendition of the runway, which is constantly changing. Select from two levels of pilot proficiency.
PRICE . . . . **$16.95** cassette **$20.95** diskette

□ **ALPHA FIGHTER:** by Douglas McFarland (Atari, 16K)
Consisting of two different programs, ALPHA FIGHTER requires you to destroy the alien starships. As you become more successful, the games get harder and and harder. **PRICE** . **$14.95** cassette **$18.95** diskette

□ **GIANT SLALOM:** by Dennis Zander (Atari, 16K)
Bring the Winter Olympics to your computer anytime of the year! Use the joystick to guide your skier's path down a giant slalom course consisting of open and closed gates. Choose from three levels of difficulty. Take practice runs or compete against from two to eight additional skiers.
PRICE . . . . **$15.95** cassette **$19.95** diskette

□ **HODGE PODGE:** by Marsha Meredith (Apple 48K, Applesoft or Integer BASIC)
This captivating program is a marvelous learning device for children from 18 months to 6 years. HODGE PODGE consists of many cartoons, animations and songs which appear when any key on the computer is depressed. A must for any family containing young children and an Apple.
PRICE . . . . **$19.95** diskette

□ **STUD POKER:** by Jerry White (Atari, 16K)
This is the classic gambler's card game. You will find the computer to be a worthy opponent who occasionally bluffs but never cheats! STUD POKER employs all of the Atari's sound, color and graphics capabilities.
PRICE . . . . **$14.95** Cassette **$18.95** diskette

TYPE-'N-TALK*
ARTWORX is offering the fantastic TYPE-'N-TALK™ from Vortrax*. This easy-to-use unit connects to your computer's serial port. Text is automatically translated into electronic speech enabling the TYPE-'N-TALK™ hobbyist to use and enjoy it immediately.
PRICE **$329.00**

The following **ARTWORX** programs are available for TYPE-'N-TALK

STUD POKER (Atari, 24K)	**$16.95** cassette **$20.95** diskette
TEACHER'S PET (Atari, 24K; North Star)	**$16.95/ $20.95**
BRIDGE 2.0 (Atari, 24K; North Star)	**$19.95 / $23.95**
NOMINOES JIGSAW (Atari, 24K)	**$17.95 / $21.95**

Please specify "TNT" version when ordering programs.

□ **CRANSTON MANOR ADVENTURE:** by Larry Ledden (Atari, North Star and CP/M)
You must enter mysterious Cranston Manor and attempt to collect its many treasures. This extemely challenging program will provide you with many hours (days?) of adventure. The program may be interrupted at will and your status saved onto the diskette.
PRICE . **$21.95** diskette

□ **BLOCKADE:** by Edward Schneider (Atari, 16K)
Every games library needs Blockade program, and this is one of the best. Choose from three levels of difficulty and play against another person or by yourself against the clock.
PRICE **$14.95** cassette **$18.95** diskette

□ **TEACHER'S PET:** by Arthur Walsh (Atari, Apple, TRS-80, PET, North Star and CP/M (MBASIC) systems)
This is an introduction to computers as well as a learning tool for the young computerist (ages 3 7). The program provides counting practice, letter-word recognition and three levels of math skills.
PRICE **$14.95** cassette **$18.95** diskette

□ **FORM LETTER SYSTEM:** (Atari, North Star and Apple)
This is the ideal program for creating personalized form letters! FLS employs a simple-to-use text editor for producing fully justified letters. Addresses are stored in a separate file and are automatically inserted into your form letter along with a personalized salutation. Both letter files and address files are compatible with **ARTWORX** MAIL LIST 3.0 and TEXT EDITOR programs.
PRICE . . . **$39.95** diskette

□ **TEXT EDITOR:** (Atari and North Star)
This program is very "user friendly" yet employs all essential features needed for serious text editing with minimal memory requirements. Features include common sense operation, two different justification techniques, automatic line centering and straightforward text merging and manipulation. TEXT EDITOR files are compatible with **ARTWORX** FORM LETTER SYSTEM.
PRICE . . . . **$39.95** diskette

□ **MAIL LIST 3.0:** (Atari, Apple and North Star)
The very popular MAIL LIST 2.2 has now been upgraded. Version 3.0 offers enhanced editing capabilities to complement the many other features which have made this program so popular. MAIL LIST is unique in its ability to store a maximum number of addresses on one diskette (typically between 1200 and 2500 names!). Entries can be retrieved by name, keyword(s) or by zip codes. They can be written to a printer or to another file for complete file management. The program produces 1, 2 or 3-up address labels and will sort by zip code (5 or 9 digits) or alphabetically (by last name). Files are easily merged and MAIL LIST will even find and delete duplicate entries! The address files created with MAIL LIST are completely compatible with **ARTWORX** FORM LETTER SYSTEM.
PRICE . . . . **$49.95** diskette

□ **THE VAULTS OF ZURICH:** by Felix and Ted Herlihy (Atari, 24K, PET)
Zurich is the banking capital of the world. The rich and powerful deposit their wealth in its famed impregnable vaults. But you, as a master thief, have dared to undertake the boldest heist of the century. You will journey down a maze of corridors and vaults, eluding the most sophisticated security system in the world. Your goal is to reach the Chairman's Chamber to steal the most treasured possession of all: THE OPEC OIL DEEDS!
PRICE . . . . **$21.95** cassette **$25.95** diskette

□ **BRIDGE 2.0** by Arthur Walsh (Atari (24K), Apple TRS-80, PET, North Star and CP/M (MBASIC) systems)
Rated #1 by Creative Computing, BRIDGE 2.0 is the only program that allows you to both bid for the contract and play out the hand (on defense or offense!). Interesting hands may be replayed using the "duplicate" bridge feature. This is certainly an ideal way to finally learn to play bridge or to get into a game when no other (human) players are available.
PRICE . . . . **$17.95** cassette **$21.95** diskette

□ **ENCOUNTER AT QUESTAR IV:** by Douglas McFarland (Atari, 24K)
As helmsman of Rikar starship, you must defend Questar Sector IV from the dreaded Zentarians. Using your plasma beam, hyperspace engines and wits to avoid Zentarian mines and death phasers, you struggle to stay alive. This BASIC/Assembly level program has super sound, full player missile graphics and real time action.
PRICE . **$23.95** cassette **$27.95** diskette

□ **THE NOMINOES JIGSAW PUZZLE:**
by C. Minns/B. Brownlee (Atari, 24K, TRS-80, and Apple)
We quote . . ."A brainteaser supreme. . . the concept of NOMINOES JIGSAW is brillant. . . this video jigsaw game is so clever and completely original that only the most hardhearted puzzle hater could fail to be charmed."—ELECTRONIC GAMES MAGAZINE
PRICE **$17.95** cassette (also available for TRS-80 color computer) **$21.95** diskette

Artworx — **Highest Quality Software*, Guaranteed.**

ARTWORX SOFTWARE COMPANY
150 North Main Street Fairport, NY 14450 (716) 425-2833

Call ARTWORX toll-free number to order direct:
800-828-6573 In New York, Alaska, Hawaii call: (716)-425-2833
All orders are processed and shipped within 48 hours.

Shipping and handling charges:
Within North America: Add $2.00
Outside North America: Add 10% (Air Mail)
New York State residents add 7% sales tax.
Quantity Discounts:
Deduct 10% when ordering 3 or more programs

Write for FREE Catalogue listing more information about these and other quality ARTWORX programs.

Ask for ARTWORX at your local computer store.

*ATARI, APPLE, TRS-80, PET, NORTH STAR, CP/M, and TYPE'N TALK are registed tradenames and/or trademarks.

Hints

You probably don't need to be reminded that the attention span of preschoolers is not long. Try to move on to another activity before your child gets bored and begins to act silly. You want your child to remember counting as something that is fun to do.

One way for you to help beginners is for you to point to the hearts very slowly one by one. Let your child count them as you point to them. Gradually your child will take over the pointing. And, before you know it, your child will be typing in the numbers on the keyboard! Experiment. Try out different arrangements. What works best for you?

The program will run on a standard VIC without memory expansion. If you need to, you should be able to modify Count the Hearts without too much trouble.

Now you're ready to play "Count the Hearts"! But remember, to stop the game and see your score, all you need to do is press the F1 key. Have fun.

Program 1: Microsoft Version

```
100 REM VIC-20
110 REM COUNT THE HEARTS
120 REM V1.0 7/81
130 REM
200 REM
210 REM HEARTS.BEGIN
220 GOSUB 30000
230 REM PLAY GAMES
240 GOSUB 1000
250 IF Q=0 THEN 240
260 REM HEARTS.END
270 GOSUB 31000
280 END
1000 REM PLAY GAMES
1010 PRINT CHR$(147)
1020 REM DISPLAY HEARTS
1030 N=LO+INT((HI-LO+1)*RND(1))
1040 FOR I=1 TO N
1050 P=INT(484*RND(1))
1060 CL=INT(8*RND(1)):IF CL=1 THEN 1
     060
1070 IF PEEK(VA+P)=83 THEN 1050
1080 POKE VA+P,83
1090 POKE CA+P,CL
1100 POKE VL,15
1110 POKE S2,200
1120 FOR Z=1 TO 400:NEXT
1130 POKE S2,0:POKE VL,0
1140 NEXT I
1150 G=G+1:REM GAMES
1160 PRINT CHR$(19);
1170 FOR I=1TO21:PRINT" ";:NEXT
1180 PRINTCHR$(19);"HOW MANY HEARTS ~
     ? ";
1190 REM GET RESPONSE
1200 GOSUB 3000
1210 IF R$="QUIT" THEN Q=1:RETURN
1220 IF R$="TIME" THEN GOSUB 9000:RE
    TURN
1230 REM O.K.?
1240 R=VAL(R$)
1250 IF R<>N THEN GOSUB 5000:GOTO 11
    60
1260 IF R=N THEN GOSUB 7000
1270 RETURN
3000 REM TIMED RESPONSE
3010 T1=TI+SC*60
3020 R$=""
3030 REM TRY A KEY
3040 GET A$
3050 IF TI>T1 THEN R$="TIME":RETURN
3060 IF A$="" THEN 3040
3070 IF ASC(A$)=133 THEN R$="QUIT":R
    ETURN
3080 IF ASC(A$)=13 THEN RETURN
3090 IF ASC(A$)=20 AND LEN(R$)>0 THE
    N GOSUB 3300:R$=LEFT$(R$,(
    LEN(R$)-1)):GOTO3040
3095 IF ASC(A$)=20 THEN 3040
3100 PRINT A$;
3110 IF A$<"0" OR A$>"9" THEN GOSUB ~
    3300:GOTO 3040
3120 R$=R$+A$
3130 GOTO 3040
3300 REM BACKSPACE
3310 PRINT CHR$(157);
3320 PRINT " ";
3330 PRINT CHR$(157);
3340 RETURN
5000 REM WRONG
5010 WR=WR+1
5030 REM UFO-VARIATION
5040 POKE VL,15
5050 FOR L=1 TO 15
5060 POKE SB,42
5070 FOR M=200 TO 220+L*2
5080 POKE S3,M
5090 NEXT M
5100 POKE SB,25
5110 FOR Z=1TO 25:NEXT Z
5120 NEXT L
5130 POKE VL,0:POKE S3,0
5140 POKE SB,27
5150 RETURN
7000 REM RIGHT
7010 RI=RI+1
7020 REM BIRDS VARIATION
7025 PRINT CHR$(19);:FOR Z=1 TO 21:P
     RINT " ";:NEXT Z
7030 POKE VL,15
7040 FOR L=1 TO 20
7050 PRINT CHR$(19);SPC(5);CHR$(106)
     ;CHR$(113);CHR$(107);
```

```
7055 PRINT SPC(5);CHR$(117);CHR$(113
     );CHR$(105);
7060 FOR M=254 TO 240+RND(1)*10 STEP
     -1
7070 POKE S3,M
7080 NEXT M
7090 POKE S3,0
7100 FOR M=1 TO 100:NEXT M
7110 PRINT CHR$(19);SPC(5);CHR$(117)
     ;CHR$(113);CHR$(105);
7115 PRINT SPC(5);CHR$(106);CHR$(113
     );CHR$(107);
7120 FOR M=1 TO 120*RND(1):NEXT M
7130 NEXT L
7140 POKE S3,0:POKE VL,0
7150 RETURN
9000 REM TIME
9010 TM=TM+1
9020 VM=VA+253:CM=CA+253
9030 REM TONE
9040 POKE S3,240:POKE VL,15
9050 J=0
9060 FOR L=15 TO 0 STEP -2
9070 POKE VM+J,81:POKE CM+J,2
9080 POKE VM-J,81:POKE CM-J,2
9090 POKE VM+22*J,81:POKE CM+22*J,2
9100 POKE VM-22*J,81:POKE CM-22*J,2
9110 FOR Z=1 TO 50:NEXT Z
9120 POKE VM+J,32
9130 POKE VM-J,32
9140 POKE VM+22*J,32
9150 POKE VM-22*J,32
9160 FOR Z=1TO50:NEXT Z
9170 POKE VL,L
9180 J=J+1
9190 NEXT L
9200 POKE VM-2,20:POKE CM-2,4
9210 POKE VM-1,9:POKE CM-1,4
9220 POKE VM,13:POKE CM,4
9230 POKE VM+1,5:POKE CM+1,4
9240 POKE S3,0:POKE VL,0
9250 FOR Z=1 TO 2000:NEXT Z
9260 RETURN
30000 REM HEARTS.BEGIN
30010 REM CONSTANTS/VARS
30020 VA=7702
30030 CA=38422
30040 SB=36879
30050 VL=36878
30060 S2=36875
30070 S3=36876
30080 S4=36877
30090 LO=1
30100 HI=9
30110 SC=120
30120 G=0
30130 TM=0
30140 RI=0
30150 WR=0
30160 Z=RND(-TI)
30170 PRINT CHR$(147);
30180 PRINT SPC(8);"VIC-20"
30190 PRINT
30200 PRINT"   COUNT THE HEARTS"
30210 PRINT:PRINT
30215 PRINT CHR$(158);
30220 PRINT"    COPYRIGHT 1981"
30230 PRINT"  HOMESPUN  SOFTWARE"
30235 PRINT CHR$(31);
30240 PRINT:PRINT
30250 PRINT"ENTER NUMBER RANGE"
30260 INPUT"LOW NUMBER (1)";LO
30270 LO=ABS(INT(LO))
30275 IF LO<1 OR LO>484 THEN PRINT "S
      ORRY":LO=1:GOTO 30260
30280 INPUT"HIGH NUMBER (9)";HI
30290 HI=ABS(INT(HI))
30300 IF HI<=LO OR HI>484 THEN PRINT"
      SORRY":HI=9:GOTO 30260
30310 PRINT
30320 PRINT"TIME LIMIT PER SET:"
30330 INPUT"SECONDS (120)";SC
30340 SC=ABS(INT(SC))
30345 IF SC<1 THEN PRINT "SORRY":SC=1
      20:GOTO 30330
30350 PRINT:PRINT
30360 PRINT"KEY F1 TO STOP"
30370 PRINT:PRINT
30380 PRINT"THANK YOU. HAVE FUN!"
30390 FOR Z=1TO2000:NEXT
30400 RETURN
31000 REM HEARTS.END
31010 PRINT CHR$(147)
31020 PRINT"COUNT THE HEARTS"
31030 PRINT:PRINT
31032 PRINT"LOW #",LO
31034 PRINT"HIGH #",HI
31036 PRINT:PRINT
31040 PRINT "# GAMES",G
31050 PRINT "# RIGHT",RI
31060 PRINT "# WRONG",WR
31070 PRINT "# TIME OUTS";TM
31080 RETURN
```

Program 2: Atari Version

```
100 REM ATARI 400/800
110 REM COUNT THE HEARTS
120 REM V1.1 7/81
130 REM
200 REM
210 REM HEARTS.BEGIN
220 GOSUB 30000
230 REM PLAY GAMES
240 GOSUB 1000
250 IF Q=0 THEN 240
```

```
260 REM HEARTS.END
270 GOSUB 31000
280 END
1000 REM PLAY GAMES
1010 GRAPHICS 1+16:SETCOLOR 0,0,0:POKE 7
56,226:DL=PEEK(560)+256*PEEK(561)+4:POKE
 DL-1,66
1015 VA=PEEK(88)+256*PEEK(89)+40
1020 REM DISPLAY HEARTS
1030 N=LO+INT((HI-LO+1)*RND(0))
1040 FOR I=1 TO N
1050 P=INT(440*RND(1))
1060 CL=INT(3*RND(1)+1)
1070 T=PEEK(VA+P):IF T=64 OR T=128 OR T=
192 THEN 1050
1080 POKE VA+P,64*CL
1090 REM
1100 SOUND 0,100,10,8
1110 REM
1120 FOR Z=1 TO 100:NEXT Z
1130 SOUND 0,0,0,0
1140 NEXT I
1150 G=G+1:REM GAMES
1160 POKE 87,0:POSITION 1,0:? "
          ";:REM 19 SPACES
1180 POSITION 1,0:? "How many hearts?";
1190 REM GET RESPONSE
1200 GOSUB 3000
1210 IF R$="QUIT" THEN Q=1:RETURN
1220 IF R$="TIME" THEN GOSUB 9000:RETURN

1230 REM O.K.?
1240 P=10^(LEN(R$)-2):R=0:FOR I=2 TO LEN
(R$):R=R+(ASC(R$(I,I))-48)*P:R=INT(R+0.0
5):P=P/10:NEXT I
1250 IF R<>N THEN GOSUB 5000:GOTO 1160
1260 IF R=N THEN POKE DL-1,70:GOSUB 7000

1270 RETURN
2000 STOP
3000 REM TIMED RESPONSE
3010 T1=PEEK(20)+256*PEEK(19)+SC*60
3020 R$=" "
3030 REM TRY A KEY
3040 IF PEEK(20)+256*PEEK(19)>T1 THEN R$
="TIME":RETURN
3050 IF PEEK(53279)<7 THEN R$="QUIT":RET
URN
3060 IF PEEK(764)=255 THEN 3040
3070 GET #1,A
3080 IF A=155 THEN RETURN
3090 IF A=126 AND LEN(R$)>1 THEN GOSUB 3
300:R$=R$(1,LEN(R$)-1):GOTO 3040
3095 IF A=126 AND LEN(R$)=1 THEN 3020
3097 IF A=126 THEN 3040
3100 IF A<48 OR A>57 THEN 3040
3110 PRINT CHR$(A);
3120 R$(LEN(R$)+1)=CHR$(A)
3130 GOTO 3040
3300 REM BACKSPACE
3310 PRINT CHR$(30);" ";CHR$(30);:RETURN

5000 REM WRONG
5010 WR=WR+1
5030 REM UFO-VARIATION
5040 FOR L=1 TO 15
5050 FOR M=100 TO 140+L*2 STEP 2
5060 SOUND 0,M,10,8:POKE 712,PEEK(53770)

5070 NEXT M
5080 FOR Z=1 TO 10:NEXT Z
5090 NEXT L
5100 SOUND 0,0,0,0:POKE 712,0
5110 RETURN
7000 REM RIGHT
7010 RI=RI+1:POKE 87,1
7020 REM BIRDS VARIATION
7025 COLOR 32:PLOT 0,0:DRAWTO 19,0
7040 FOR L=1 TO 5
7050 POSITION 5,0:? #6;CHR$(17);CHR$(20)
;CHR$(5);
7055 POSITION 10,0:? #6;CHR$(26);CHR$(20
);CHR$(3);
7060 FOR M=50 TO 10+10*RND(1) STEP -1
7070 SOUND 0,M,10,8
7080 NEXT M
7090 SOUND 0,0,0,0
7100 FOR M=1 TO 50:NEXT M
7110 POSITION 5,0:? #6;CHR$(26);CHR$(20)
;CHR$(3);
7120 POSITION 10,0:? #6;CHR$(17);CHR$(20
);CHR$(5);
7130 FOR M=1 TO 60*RND(1):NEXT M
7140 NEXT L:POKE 87,0
7150 RETURN
9000 REM TIME
9005 POSITION 1,0:? "                TIME
 OUT          ";
9010 TM=TM+1
9020 UM=VA+229
9030 REM TONE
9040 SOUND 0,100,12,8
9050 J=0
9060 FOR L=15 TO 0 STEP -1
9070 POKE UM+J,148
9080 POKE UM-J,148
9090 POKE UM+20*J,148
9100 POKE UM-20*J,148
9110 FOR Z=1 TO 50:NEXT Z
9120 POKE UM+J,0
9130 POKE UM-J,0
9140 POKE UM+20*J,0
```

```
9150 POKE VM-20*J,0
9160 FOR Z=1 TO 50:NEXT Z
9170 SOUND 0,L*5,12,L
9180 J=J+(L/2<>INT(L/2))
9190 NEXT L
9200 RETURN
30000 REM HEARTS.BEGIN
30010 REM CONSTANTS/VARS
30090 LO=1:HI=9:SC=120:G=0:TM=0:RI=0:WR=
0
30100 DIM R$(20):OPEN #1,4,0,"K:"
30170 GRAPHICS 0
30180 ? CHR$(125);:POSITION 12,0:? " ATA
RI 400/800"
30190 POSITION 11,2:? "Count the Hearts"

30210 ? :?
30220 ? "          Copyright 1981"
30230 ? "        HOMESPUN SOFTWARE"
30240 ? :?
30250 ? "Enter number range:"
30260 TRAP 30250:? "Low number (1)";:INP
UT LO:TRAP 40000
30270 LO=ABS(INT(LO))
30275 IF LO<1 OR LO>440 THEN ? "SORRY":L
O=1:GOSUB 30260
30280 ? "High number (9)";:TRAP 30280:IN
PUT HI:TRAP 40000
30290 HI=ABS(INT(HI))
30300 IF HI<=LO OR HI>440 THEN ? "SORRY"
:HI=9:GOTO 30260
30310 ?
30320 ? "TIME LIMIT PER SET:"
30330 ? "Seconds (120)";:TRAP 30330:INPU
T SC:TRAP 40000
30340 SC=ABS(INT(SC))
30350 ? :?
30360 ? "PRESS [OPTION] TO STOP"
30370 ? :?
30380 ? "THANK YOU. Have fun!"
30390 FOR Z=1 TO 100:NEXT Z
30400 RETURN
31000 REM HEARTS.END
31010 GRAPHICS 0
31020 ? "COUNT THE HEARTS"
31030 ? :?
31032 ? "LOW #",LO
31034 ? "HIGH #",HI
31036 ? :?
31040 ? "GAMES",G
31050 ? "RIGHT",RI
31060 ? "WRONG",WR
31070 ? "TIME OUTS ";TM
31080 RETURN
```

©

Family: A Simulation In Genetics

Andy Gamble
Columbia College
Vancouver, Canada

Biology teachers know only too well the practical problems of illustrating the effects of gene selection. Mendel struck it lucky with his pea plants, but even they are a little too complicated for an introductory class. Besides, any meaningful experiment would take months, even years, to conduct. Barring a few thousand fruit-flies, what can one do?

Once again the mild-mannered computer steps into the nearest telephone booth and saves the day. This program lightheartedly illustrates the genealogy of a Martian couple.

Mars Genetics

A subject was clearly needed which could be easily displayed on the PET screen, with control over a few features. In this case, the Martians have either white or gray hair (green or light green on some PETs) and round or slanted eyes. The gene for white hair is dominant, as is the gene for round eyes. Male and female Martians are represented by square and round jawlines respectively.

The usual method of denoting dominant and recessive genes by upper and lower case letters is not used here, as the PET cannot display both with the graphics characters at the same time.

The program can be used in two ways. First, merely as a demonstration of the effects of gene selection. Genes are chosen at random from the parents, and control the facial features of their 24 children. The sexes are also randomly assigned. The genes for the parents can be picked at random by the program, or chosen by the user.

Alternatively the user may choose to have the faces of the parents and children displayed, but not their genes. The parents' genes can therefore be determined from the features, giving practice of a more experimental kind.

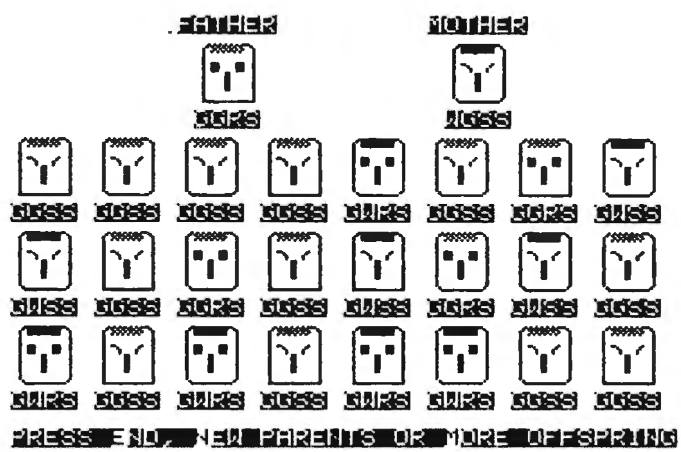
The instructions and the RUN of the program should be self-explanatory. There are several techniques used in the program which I think are quite interesting. I believe very strongly in making programs as user-friendly as possible, and this is particularly important when dealing with INPUT statements. When a yes/no answer is needed, the easiest method is to use a trick INPUT statement (see lines 280-410).

Parts of this program need input which is less obvious to the user: namely, the genes R,S,W and G. It's perfectly possible to remind the user of this when necessary, but here a different method is used. When that input is called for, the genes are displayed on the screen, as for example:

WG
↑

The arrow is moved left or right by the < and > keys; the genes are picked by pressing return. This is foolproof as far as I can tell, and there is the added advantage of it being obvious which genes are to be chosen (see lines 510-1080).

The program uses over 8K as given so, to run it on 8K machines some editing is necessary. The instructions can be removed (lines 280-310, 1520-1870) and printed separately for student use. Removing all the REM statements also (none are referenced) brings the memory needed down to just over 6K. Family will run on all 40-column PETs.




```
100 REM FAMILY
110 REM ANDY GAMBLE JUNE 81
120 REM COLUMBIA COLLEGE, 1619 W10
    AVE
130 REM VANCOUVER BC V6J 2A2
140 GOTO1430
150 REM HP<=35,VP<=19
160 VT$="{HOME}{25 DOWN}"
170 X=RND(-RND(0))
180 DEFFNR(X)=INT(2*RND(1)+1)
190 NO$="]'%]"
200 CH$(1)="J@@K":REM FEM
210 CH$(2)="-@@=":REM MALE
220 EY$(1)="];,]"
230 EY$(2)="]IU]"
240 HA$(1)="U""I"
250 HA$(2)="U((I"
```

```
26Ø REM W OVER G, R OVER S
27Ø GE$="N"
28Ø INPUT"{CLEAR}{Ø4 DOWN}DO YOU NE
    ED INSTRUCTIONS  _"{Ø3 LEFT
    LEFT}";ZZ$
29Ø IFZZ$=""THEN28Ø
3ØØ IFLEFT$(ZZ$,1)="Y"THEN152Ø
31Ø IFLEFT$(ZZ$,1)<>"N"THEN28Ø
32Ø GE$="N":PRINT"{CLEAR}{Ø3 DOWN}D
    O YOU WANT TO GUESS THE GE
    NOTYPES";
33Ø INPUT"(Y/N)  _"{Ø3 LEFT}";GE$
34Ø IFGE$=""THEN32Ø
35Ø GE$=LEFT$(GE$,1):IFGE$="N"THEN3
    7Ø
36Ø GOTO43Ø
37Ø PRINT"{CLEAR}{Ø3 DOWN}ARE THE G
    ENES FOR THE PARENTS TO BE
         CHOSEN RANDOMLY ";
38Ø INPUT"(Y/N)  _"{Ø3 LEFT}";ZZ$
39Ø IFZZ$=""THEN37Ø
4ØØ IFLEFT$(ZZ$,1)="N"THEN5ØØ
41Ø IFLEFT$(ZZ$,1)<>"Y"THEN32Ø
42Ø REM RANDOM
43Ø FORI=1TO2:FE$(I)="R":IFRND(1)>.
    5THENFE$(I)="S"
44Ø FH$(I)="W":IFRND(1)>.5THENFH$(I
    )="G"
45Ø NEXT
46Ø FORI=1TO2:ME$(I)="R":IFRND(1)>.
    5THENME$(I)="S"
47Ø MH$(I)="W":IFRND(1)>.5THENMH$(I
    )="G"
48Ø NEXT
49Ø GOTO1Ø9Ø
5ØØ PA=33Ø67
51Ø PRINT"{CLEAR}{Ø2 DOWN}CHOOSE GE
    NES FOR FATHER'S HAIR:"
52Ø PRINT"{Ø3 DOWN}"TAB(19)"{REV}WG
    "
53Ø E$(1)="":E$(2)=""
54Ø GOSUB2Ø2Ø
55Ø POKEPA,3Ø
56Ø FORI=1TO2
57Ø GOSUB196Ø
58Ø FH$(I)="G":IFPA=33Ø67THENFH$(I)
    ="W"
59Ø H$(I)=FH$(I)
6ØØ NEXT
61Ø VP=1Ø:HP=18:SEX=2
62Ø GOSUB192Ø
63Ø GOSUB136Ø
64Ø GOSUB2Ø4Ø
65Ø PRINT"{Ø2 DOWN}":GOSUB189Ø
66Ø PRINT"{CLEAR}{Ø2 DOWN}CHOOSE GE
    NES FOR FATHER'S EYES:"
67Ø PRINT"{Ø3 DOWN}"TAB(19)"{REV}RS
    "
68Ø GOSUB2Ø2Ø
69Ø POKEPA,3Ø
7ØØ FORI=1TO2
71Ø GOSUB196Ø
72Ø FE$(I)="S":IFPA=33Ø67THENFE$(I)
    ="R"
73Ø E$(I)=FE$(I)
74Ø NEXT
75Ø VP=1Ø:HP=18:SEX=2
76Ø GOSUB192Ø
77Ø GOSUB136Ø
78Ø GOSUB2Ø4Ø
79Ø PRINT"{Ø2 DOWN}":GOSUB189Ø
8ØØ PRINT"{CLEAR}{Ø2 DOWN}CHOOSE GE
    NES FOR MOTHER'S HAIR:"
81Ø PRINT"{Ø3 DOWN}"TAB(19)"{REV}WG
    "
82Ø GOSUB2Ø2Ø
83Ø E$(1)="":E$(2)=""
84Ø POKEPA,3Ø
85Ø FORI=1TO2
86Ø GOSUB196Ø
87Ø MH$(I)="G":IFPA=33Ø67THENMH$(I)
    ="W"
88Ø H$(I)=MH$(I)
89Ø NEXT
9ØØ VP=1Ø:HP=18:SEX=1
91Ø GOSUB192Ø
92Ø GOSUB136Ø
93Ø GOSUB2Ø4Ø
94Ø PRINT"{Ø2 DOWN}":GOSUB189Ø
95Ø PRINT"{CLEAR}{Ø2 DOWN}CHOOSE GE
    NES FOR MOTHER'S EYES:"
96Ø PRINT"{Ø3 DOWN}"TAB(19)"{REV}RS
    "
97Ø GOSUB2Ø2Ø
98Ø POKEPA,3Ø
99Ø FORI=1TO2
1ØØØ GOSUB196Ø
1Ø1Ø ME$(I)="S":IFPA=33Ø67THENME$(I)
     ="R"
1Ø2Ø E$(I)=ME$(I)
1Ø3Ø NEXT
1Ø4Ø VP=1Ø:HP=18:SEX=1
1Ø5Ø GOSUB192Ø
1Ø6Ø GOSUB136Ø
1Ø7Ø GOSUB2Ø4Ø
1Ø8Ø PRINT"{Ø2 DOWN}":GOSUB189Ø
1Ø9Ø PRINTCHR$(147)TAB(1Ø)"{REV}FATH
     ER{Ø9 RIGHT}MOTHER"
11ØØ VP=1:HP=11:SEX=2
111Ø FORI=1TO2:H$(I)=FH$(I):E$(I)=FE
     $(I):NEXT
112Ø GOSUB192Ø
113Ø GOSUB136Ø
114Ø VP=1:HP=26:SEX=1
115Ø FORI=1TO2:H$(I)=MH$(I):E$(I)=ME
     $(I):NEXT
116Ø GOSUB192Ø
117Ø GOSUB136Ø
```


commodore
Check Our New Lowest Prices!
EQUIPMENT:
VIC-20 (Includes RF Modulator) 265
CBM 4016 CPU (40 Col. Screen, 16K RAM) 790
CBM 4032 CPU (40 Col. Screen, 32K RAM) 990
CBM 8032 CPU (80 Col. Screen, 32K RAM) 1090
CBM 8096 CPU (80 Col. Screen, 96K RAM) 1590
CBM Micro Mainframe (Super PET) 1690
VIC Single Disk Drive (170K per 5¼ Diskette) 470
NEW CBM 2031 Single Disk Drive (170K per 5¼ Diskette) 570
CBM 4040 Dual Disk Drive (170K per 5¼ Diskette) 990
CBM 8050 Dual Disk Drive (½ Meg per 5¼ Diskette) 1340
VIC Tractor Feed Printer (80 Col., 30 CPS) 325
CBM 4022 Tractor Feed Printer 625
8023P Dot Matrix Printer (136 Col. 150 CPS) 790
Tally 8024-7 (7x7 Matrix) Printer 1275
NEW Tally 8024-9 (7x9 Matrix) Printer 1425
8300P Letter Quality Printer (40 CPS) 1790
NEW CBM 8010 Phone Modem 225
CBM C2N Cassette Deck (New Style) 65
CBM CPU/IEEE Cable 35
CBM IEEE/IEEE Cable 40
SOFTWARE:
Wordcraft 80 Wordprocessor 295
Wordpro 4+ Wordprocessor 325
OZZ Data Base System 295
Visicalc 175
Tax Preparation System 395
IRMA Information Retrieval 395
Dow Jones Portfolio 115
PUBLICATIONS:
CBM User Guide 7.95
CBM Basic 4.0 Reference Manual 9.95
CBM Disk Manual 7.95
CBM Printer Manual 7.95
MOS Hardware Manual 6.95
MOS Programming Manual 6.95
The PET Revealed 19.95
Library of PET Subroutines 19.95
Commodore Software Encyclopedia 4.95
VIC SOFTWARE:
VT 106A Recreation Six Pack 43.95
Includes Car Chase, Blue Meanies, Space Math,
Slither/Super Slither, Biorhythm Capability
VT 107A Home Utility Six Pack 43.95
Includes Personal Finance I, Personal Finance II,
VIC Typewriter, Expense Calendar, Loan & Mortgage
Calculator, Home Inventory
VIC 1901 VIC Super Alien 29.95
VIC 1904 Super Slot 29.95
VIC 1907 Super Lander 29.95
VIC 1908 Draw Poker 29.95
"CALL ABOUT SUPPLIES"
COMING SOON:
More VIC Peripherals and Software
CBM 8250 Dual Disk Drive
Data Acquisition and Control Devices
All items Insured
COD – UPS
Prepaid Orders Shipped Free
In stock items shipped within 48 hours
MASTERCHARGE OR VISA ADD 3%
GA. RESIDENTS ADD 4% SALES TAX
MART
3300 Buckeye Rd., N.E.
Suite 641
Atlanta, Ga. 30341
404-458-0729
Call 9 AM-5 PM EST M-F
WORD PROCESSING SPECIAL
CBM 8032
CBM 4040
CBM 8300 (Letter Quality, DIABLO 630, 40 CPS, Bi-directional)
WORDCRAFT 80
• Includes all cables
$4200
CALL ABOUT EQUIPMENT AND/OR SOFTWARE SUBSTITUTIONS

```
1180 REM OFFSPRING
1190 FORI=1TO3:FORJ=0TO7
1200 VP=1+5*I:HP=5*J
1210 H$(1)=FH$(FNR(1)):H$(2)=MH$(FNR
     (1))
1220 E$(1)=FE$(FNR(1)):E$(2)=ME$(FNR
     (1))
1230 HC=1:IF(H$(1)="G")AND(H$(2)="G"
     )THENHC=2
1240 EC=1:IF(E$(1)="S")AND(E$(2)="S"
     )THENEC=2
1250 SEX=FNR(1)
1260 GOSUB1360
1270 NEXTJ,I
1280 IFGE$="Y"THENGOSUB2070
1290 PRINT"{DOWN}{REV}PRESS {OFF}E{R
     REV}ND, {OFF}N{REV}EW PARE
     NTS OR {OFF}M{REV}ORE OFFS
     PRING"
1300 GOSUB1900
1310 IFZZ$="N"THEN320
1320 IFZZ$="M"THEN1190
1330 IFZZ$<>"E"THEN1300
1340 PRINT"{HOME}";:END
1350 REM DRAW FACE
1360 PRINTLEFT$(VT$,VP+1)TAB(HP);
1370 PRINTHA$(HC)"{04 LEFT}{DOWN}";
1380 PRINTEY$(EC)"{04 LEFT}{DOWN}";
1390 PRINTNO$"{04 LEFT}{DOWN}"CH$(SE
     X)"{04 LEFT}{DOWN}";
1400 IFGE$="N"THENPRINT"{REV}"H$(1)H
     $(2)E$(1)E$(2)"{OFF}":RETU
     RN
1410 IFGE$="Y"THENPRINT"     "
1420 RETURN
1430 PRINT"{CLEAR}":FORI=32768TO3280
     7:POKEI,224:POKEI+960,224:
     NEXT
1440 FORI=32808TO33688STEP40:POKEI,2
     24:POKEI+39,224:NEXT
1450 PRINT"{HOME}{04 DOWN}"
1460 PRINTTAB(9)"  O# $'      L
1470 PRINTTAB(9)"  L$:::::MNLLLLNM
1480 PRINT"{HOME}{10 DOWN}"TAB(16)"{
     REV}FAMILY"
1490 PRINT"{HOME}{12 DOWN}"TAB(25)"A
     NDY GAMBLE"
1500 PRINT"{07 DOWN}";:GOSUB1890
1510 GOTO160
1520 PRINT"{CLEAR}{02 DOWN}"TAB(15)"
     {REV}FAMILY{OFF}{02 DOWN}"

1530 PRINT"THIS PROGRAM SHOWS YOU A
     TYPICAL FAMILY OF MARTIANS
     : TWO PARENTS";
1540 PRINT" AND ";
1550 PRINT"THEIR 24   CHILDREN. YOU
     CAN TELL THE DIFFERENCE"
1560 PRINT"BETWEEN MALE AND ";
1570 PRINT"FEMALE MARTIANS VERY   EA
     SILY. MALES HAVE SQUARE JA
     WS AND"
1580 PRINT"FEMALES HAVE ";
1590 PRINT"ROUND ONES. OTHER THAN TH
     ATTHEY HAVE WHITE OR GRAY
     HAIR, AND ROUND
1600 PRINT"OR   SLANTED EYES. THESE
     TRAITS ARE";
1610 PRINT"      CONTROLLED BY GENES
     W AND G FOR ";
1620 PRINT"THE HAIRAND R AND S FOR T
     HE EYES. W IS DOMINANT OVE
     R G AND R IS ";
1630 PRINT"DOMINANT OVER S. HERE   A
     RE TWO TYPICAL MARTIANS:"
1640 VP=17:HP=15:SEX=1:H$(1)="G":H$(
     2)="W":E$(1)="R":E$(2)="S"
     :EC=1:HC=1
1650 GOSUB1360
1660 VP=17:HP=22:SEX=2:H$(1)="G":H$(
     2)="G":E$(1)="S":E$(2)="S"
     :EC=2:HC=2
1670 GOSUB1360:PRINT"{DOWN}";
1680 GOSUB1890
1690 PRINT"{CLEAR}{02 DOWN}YOU MAY C
     HOOSE THE GENES FOR THE HA
     IR"
1700 PRINT"AND EYES OF BOTH THE MOTH
     ER AND THE"
1710 PRINT"FATHER WHEN THE PROGRAM R
     EQUESTS IT,"
1720 PRINT"BY CHOOSING  TWO GENES SU
     CH AS 'GG'"
1730 PRINT"OR 'RR' - WHATEVER YOU LI
     KE. YOU CAN"
1740 PRINT"ALSO LET THE PROGRAM CHOO
     SE THE"
1750 PRINT"PARENTS' GENES RANDOMLY."

1760 PRINT"{DOWN}AFTER YOU ARE SHOWN
      THE OFFSPRING, YOU"
1770 PRINT"WILL HAVE A CHOICE AS TO
     MORE OFFSPRING"
1780 PRINT"BY THE SAME PARENTS, DIFF
     ERENT PARENTS,"
1790 PRINT"OR ENDING THE PROGRAM.{02
      DOWN}"
1800 GOSUB1890:PRINT"{CLEAR}{02 DOWN
     DOWN}ALTERNATIVELY YOU CAN
      CHOOSE TO BE SHOWN"
1810 PRINT"{UP}THE FACES, AND TRY TO
      GUESS THE"
1820 PRINT"PARENTS' GENES (THEIR GEN
     OTYPES)."
1830 PRINT"{DOWN}NOTE THAT THE COMPU
     TER HAS IN MIND ONE"
1840 PRINT"PARTICULAR SET OF GENES:
     OTHERS MAY BE"
```

```
1850 PRINT"POSSIBLE BUT WILL GIVE A ~
    WRONG ANSWER."
1860 PRINT"{03 DOWN}";:GOSUB1890
1870 GOTO320
1880 REM GET-CONT
1890 PRINTTAB(8)"{REV}PRESS ANY KEY ~
    TO CONTINUE"
1900 GETZZ$:IFZZ$=""THEN1900
1910 RETURN
1920 HC=1:IF(H$(1)="G")AND(H$(2)="G"
    )THENHC=2
1930 EC=1:IF(E$(1)="S")AND(E$(2)="S"
    )THENEC=2
1940 RETURN
1950 REM CHOOSE GENES
1960 GOSUB1900
1970 IFZZ$="<"ANDPA=33068THENPOKEPA,
    32:PA=33067:POKEPA,30
1980 IFZZ$=">"ANDPA=33067THENPOKEPA,
    32:PA=33068:POKEPA,30
1990 IFZZ$<>CHR$(13)THEN1960
2000 RETURN
2010 REM MESSAGE
2020 PRINTLEFT$(VT$,19);"{REV}PRESS ~
    < TO MOVE LEFT, > TO MOVE ~
    RIGHT "
2030 PRINTLEFT$(VT$,20);"{REV}    PRE
    SS RETURN WHEN GENE IS CHO
    SEN    ":RETURN
2040 PRINTLEFT$(VT$,19);"             ~
                                    ~

      "
2050 PRINTLEFT$(VT$,20);"             ~
                                    ~

      ":RETURN
2060 REM HAIR GENE GUESS
2070 PRINTLEFT$(VT$,22);"{REV}INPUT ~
    GENES FOR HAIR OF ONE PARE
    NT (W/G)"
2080 FG=0
2090 GOSUB1900
2100 IFZZ$="W"ORZZ$="G"THEN2130
2110 IFZZ$="E"THEN1340
2120 GOSUB1900
2130 HG$(1)=ZZ$
2140 GOSUB1900:IFZZ$="W"ORZZ$="G"THE
    N2160
2150 GOSUB1900
2160 HG$(2)=ZZ$
2170 PRINTLEFT$(VT$,22);"             ~
                                    ~

      "
2180 IF FH$(1)=HG$(1) ANDFH$(2)=HG$(
    2)THEN2220
2190 IF FH$(1)=HG$(2) ANDFH$(2)=HG$(
    1)THEN2220
2200 GOTO2250
2210 REM FATHER GUESS CORRECT
2220 PRINTLEFT$(VT$,6);TAB(11)"{REV}
```

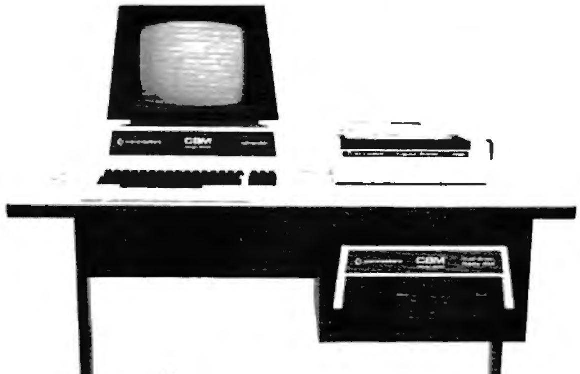
```
    "FH$(1)FH$(2)
2230 FG=1
2240 GOTO2340
2250 IF MH$(1)=HG$(1) ANDMH$(2)=HG$(
    2)THEN2290
2260 IF MH$(1)=HG$(2) ANDMH$(2)=HG$(
    1)THEN2290
2270 GOTO2320
2280 REM MOTHER GUESS CORRECT
2290 PRINTLEFT$(VT$,6);TAB(26)"{REV}
    "MH$(1)MH$(2)
2300 GOTO2340
2310 REM BOTH WRONG
2320 PRINTLEFT$(VT$,22);"{REV}    AN
    SWER IS WRONG: TRY AGAIN O
    R {OFF}E{REV}ND    "
2330 GOTO2090
2340 PRINTLEFT$(VT$,22);"{REV}  INPU
    T GENES FOR HAIR OF OTHER ~
    (W/G)    "
2350 GOSUB1900
2360 IFZZ$="W"ORZZ$="G"THEN2390
2370 IFZZ$="E"THEN1340
2380 GOSUB1900
2390 HG$(1)=ZZ$
2400 GOSUB1900:IFZZ$="W"ORZZ$="G"THE
    N2420
2410 GOSUB1900
2420 HG$(2)=ZZ$
2430 PRINTLEFT$(VT$,22);"             ~
                                    ~

      "
2440 IFFG=1THEN2520
2450 IF FH$(1)=HG$(1) ANDFH$(2)=HG$(
    2)THEN2500
2460 IF FH$(1)=HG$(2) ANDFH$(2)=HG$(
    1)THEN2500
2470 IFFG=0THEN2590
2480 GOTO2520
2490 REM FATHER GUESS CORRECT
2500 PRINTLEFT$(VT$,6);TAB(11)"{REV}
    "FH$(1)FH$(2)
2510 GOTO2620
2520 IF MH$(1)=HG$(1) ANDMH$(2)=HG$(
    2)THEN2560
2530 IF MH$(1)=HG$(2) ANDMH$(2)=HG$(
    1)THEN2560
2540 GOTO2590
2550 REM MOTHER GUESS CORRECT
2560 PRINTLEFT$(VT$,6);TAB(26)"{REV}
    "MH$(1)MH$(2)
2570 GOTO2620
2580 REM BOTH WRONG
2590 PRINTLEFT$(VT$,22);"{REV}    AN
    SWER IS WRONG: TRY AGAIN O
    R {OFF}E{REV}ND    "
2600 GOTO2350
2610 REM EYE GENE GUESS
2620 PRINTLEFT$(VT$,22);"{REV}INPUT ~
```

```
    GENES FOR EYES OF ONE PARE
    NT (R/S)"
2630 FG=0
2640 GOSUB1900
2650 IFZZ$="R"ORZZ$="S"THEN2680
2660 IFZZ$="E"THEN1340
2670 GOSUB1900
2680 EG$(1)=ZZ$
2690 GOSUB1900:IFZZ$="R"ORZZ$="S"THE
    N2710
2700 GOSUB1900
2710 EG$(2)=ZZ$
2720 PRINTLEFT$(VT$,22);"


    "
2730 IF FE$(1)=EG$(1) ANDFE$(2)=EG$(
    2)THEN2770
2740 IF FE$(1)=EG$(2) ANDFE$(2)=EG$(
    1)THEN2770
2750 GOTO2800
2760 REM FATHER GUESS CORRECT
2770 PRINTLEFT$(VT$,6);TAB(13)"{REV}
    "FE$(1)FE$(2)
2780 FG=1
2790 GOTO2890
2800 IFME$(1)=EG$(1)ANDME$(2)=EG$(2)
    THEN2840
2810 IFME$(1)=EG$(2)ANDME$(2)=EG$(1)
    THEN2840
2820 GOTO2870
2830 REM MOTHER GUESS CORRECT
2840 PRINTLEFT$(VT$,6);TAB(28)"{REV}
    "ME$(1)ME$(2)
2850 GOTO2890
2860 REM BOTH WRONG
2870 PRINTLEFT$(VT$,22);"{REV}     AN
    SWER IS WRONG: TRY AGAIN O
    R {OFF}E{REV}ND    "
2880 GOTO2640
2890 PRINTLEFT$(VT$,22);"{REV}    INP
    UT GENES FOR EYES OF OTHER
     (R/S)   "
2900 GOSUB1900
2910 IFZZ$="R"ORZZ$="S"THEN2940
2920 IFZZ$="E"THEN1340
2930 GOSUB1900
2940 EG$(1)=ZZ$
2950 GOSUB1900:IFZZ$="R"ORZZ$="S"THE
    N2970
2960 GOSUB1900
2970 EG$(2)=ZZ$
2980 PRINTLEFT$(VT$,22);"


    "
2990 IFFG=1THEN3070
3000 IF FE$(1)=EG$(1)ANDFE$(2)=EG$(2
    )THEN3050
3010 IFFE$(1)=EG$(2)ANDFE$(2)=EG$(1)
    THEN3050
3020 IFFG=1THEN3140
3030 GOTO3070
3040 REM FATHER GUESS CORRECT
3050 PRINTLEFT$(VT$,6);TAB(13)"{REV}
    "FE$(1)FE$(2)
3060 GOTO3170
3070 IFME$(1)=EG$(1)ANDME$(2)=EG$(2)
    THEN3110
3080 IFME$(1)=EG$(2)ANDME$(2)=EG$(1)
    THEN3110
3090 GOTO3140
3100 REM MOTHER GUESS CORRECT
3110 PRINTLEFT$(VT$,6);TAB(28)"{REV}
    "ME$(1)ME$(2)
3120 GOTO3170
3130 REM BOTH WRONG
3140 PRINTLEFT$(VT$,22);"{REV}     AN
    SWER IS WRONG: TRY AGAIN O
    R {OFF}E{REV}ND    "
3150 GOTO2900
3160 REM ALL CORRECT
3170 PRINTLEFT$(VT$,22);"{REV}
      CORRECT! TRY AGAIN? (Y/N
    )      "
3180 GOSUB1900
3190 IFZZ$="Y"THEN320
3200 IFZZ$<>"N"THEN3180
3210 GOTO1340
```



A Wooden Computer?

Not from Commodore!

So why should the desk look like wood? A pleasant cream and charcoal trimmed desk looks so much better with Commodore systems. One look and you'll see. Interlink desks are right. By design.

The specifications only confirm the obvious:

- Cream and charcoal color beautifully matches the Commodore hardware and blends with your decor.
- An ideal 710 mm (28") keyboard height yet no bumping knees because a clever cutout recesses the computer into the desktop.
- High pressure laminate on both sides of a solid core for lasting beauty and strength.
- Electrostatically applied baked enamel finish on welded steel legs—no cheap lacquer job here.
- T-molding and rounded corners make a handsome finish on a durable edge that won't chip.
- Knocked down for safe, inexpensive shipment.
- Patented slip joints for quick easy assembly.
- Leveling glides for uneven floors.
- Room enough for a Commodore printer on the desk, yet fits into nearly any den or office niche—H: 660 mm (26") W: 1170 mm (46") D: 660 mm (26").
- Matching printer stand available with slot for bottom feeding.

Price: $299

In short, as Commodore dealers, we won't settle for anything that looks good only in the catalog! Our customers won't let us. They don't buy pictures. And neither should you. This is why **we will let you use one of our desks for a week** and then decide. If for any reason you don't like it, just return it in good condition for a cheerful refund.

If your Commodore dealer doesn't carry our desks yet, send a check for $299 and we will ship your desk freight paid!

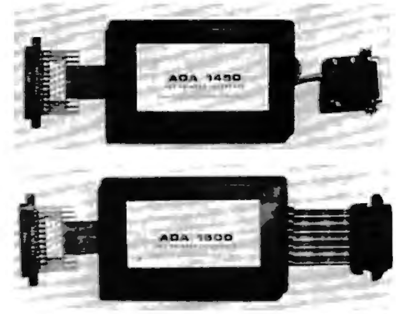
Name____________________________

Address__________________________

City______________ **St**_____ **Zip**______

Interlink, Inc., Box 134, Berrien Springs, MI 49103
Master Charge and Visa welcome. Call our order line:
616-473-3103

CBM/PET INTERFACES



RS-232 SERIAL PRINTER INTERFACE – addressable – baud rates to 9600 – switch selectable upper/lower, lower/upper case – works with WORDPRO, BASIC and other software – includes case and power supply.
MODEL – ADA1450 149.00

CENTRONICS/NEC PARALLEL INTERFACE - addressable – high speed – switch selectable upper/lower, lower/upper case – works with WORDPRO, BASIC and other software – has Centronics 36 pin ribbon connector at end of cable.
MODEL – ADA1600 129.00

CENTRONICS 730/737 PARALLEL INTERFACE – as above but with Centronics card edge connector at end of cable.
MODEL – ADA730 129.00

COMMUNICATIONS INTERFACE WITH SERIAL AND PARALLEL PORTS – addressable – software driven – true ASCII conversion – selectable reversal of upper-lower case – baud rates to 9600 – half or full duplex – X-ON, X-OFF – selectable carriage return delay – 32 character buffer – centronics compatible — much more.
MODEL – SADI 295.00

ANALOG TO DIGITAL CONVERTER – 16 channels – 0 to 5.12 volt input voltage range – resolution is 20 millivolts per count – conversion time is less than 100 microseconds per channel.
MODEL – PETSET1 295.00

REMOTE CONTROLLER WITH CLOCK/CALENDAR – controls up to 256 devices using the BSR X10 remote control receivers – 8 digital inputs, TTL levels or switch closure — 8 digital outputs, TTL levels.
MODEL – PETSET2 295.00

All prices are in US dollars for 120VAC.
Prices on 220 VAC slightly higher.
Allow $5.00 shipping & handling, foreign orders add 10% for AIR postage.
Connecticut residents add 7½% sales tax.
All prices and specifications subject to change without notice.
Our 30 day money back trial period applies.
MASTER CHARGE/VISA accepted.
MENTION THIS MAGAZINE WITH YOUR ORDER AND DEDUCT 5% FROM TOTAL.

IN CANADA order from: Batteries Included, Ltd., 71 McCaul Street, F6 Toronto, Canada M5T2X1, (416)596-1405.
IN THE USA order from your local dealer or direct: Connecticut microComputer, Inc., 34 Del Mar Drive, Brookfield, CT 06804, (203)775-4595.

Dealer inquiries invited.

CmC

Connecticut microComputer, Inc.
34 Del Mar Drive, Brookfield, CT 06804
203 775-4595 TWX: 710 456-0052

Learning With Computers

Word Processing In The Classroom

Glenn Kleiman and Mary Humphrey
Teaching Tools: Microcomputer Services
P.O. Box 50065
Palo Alto, CA 94303

"All right, class. Load the word processing program and put your name at the top of a new file. We're all going to write thank you letters to the PTA for buying the computers for our class."

In recent months we have heard from a number of teachers, students and researchers who have been using computerized word processing in classrooms. All have found it to be a successful and valuable experience, even with children as young as second grade. In this month's column we focus on word processing – what it is, some ways it is being used in classrooms, its effects on children's writing, and how to get started with it.

What Is Word Processing?

Word processing is the use of computer-controlled writing systems. The computer system replaces not only the typewriter, pen or pencil, but also the scratch paper, eraser, scissors and tape, and, in some advanced systems, the dictionary.

Programs are available to turn almost every personal computer into a word processing system. For most computers there is a choice ranging from very simple word processing programs to programs so sophisticated they match the capabilities of the word processors found in many offices. While there are important differences, even the simpler programs provide valuable writing aids.

Word processing programs make it easy to create and modify essays, notes, letters, outlines – any form of written material. The text is typed on the computer keyboard and appears on the screen. All word processors provide ways to correct typing errors, insert or delete words, save your writing for later work, and print it when you are finished. More advanced word processors can search for a given sequence of letters in the text and replace it with another sequence (great for correcting habitual spelling errors, or replacing all instances of "utilize" with "use"). They make it possible to move sections of text, such as when you decide a paragraph you put in the introduction would be better in the conclusion. They also let you format the print-out: setting margins and spacing between lines, centering headings, numbering pages and so on. Very advanced word processors add a dictionary so spelling can be checked automatically. An on-line thesaurus and systems that do some checking of sentence syntax are being developed. Perhaps some day we will have a computerized Strunk and White's *Elements of Style* program to point out the needless words we should omit.

How Can Word Processing Be Used In Education?

Teachers often report that children are unwilling to write and even more unwilling to edit and revise what they have written. Writing requires both the mental processes of composing text and the physical processes of producing it. Computerized word processing makes the physical aspects of producing and editing text much easier, so more attention can be devoted to the mental aspects. Word processors can also be used to create situations which stimulate children's creativity with language and motivate them to write.

A fourth grade class in Oceanside, California, has produced a school newsletter with their word processing system. The well written, eleven page newsletter contains news stories, book reviews, jokes, original stories, and letters to the editor. The newsletter reflects careful use of the formatting capabilities of the word processing system. It has centered headlines and is neatly divided into pages with two columns of print on each page.

The children in this class did most of their writing in pairs, helping each other both in using the computer and in creating and editing text. The children could access each other's working drafts and offer comments on them. They used a word processing program developed by researchers at the University of California at San Diego. It contains some special features, such as a command that automatically arranges the text into a paragraph

format or a sentence format. The ease of making changes made editing and revising fun rather than a chore. In a letter to the editor of the newsletter, two of the children reported that writing with the computer was "...funner and easier than writing with pencil and paper. Also it does not hurt your hand."

The potential of word processors to facilitate children's writing has been further demonstrated at the Trillium School, a special school for learning disabled children in Ontario, Canada. The students range from 7 to 16 years old, but their reading levels range from first grade to fifth grade. Many of these children had previously refused to do any form of writing. The school recently began using computers in the classrooms and found the children were eager to use them. When word processing was introduced, the teachers were delighted to find the students' enthusiasm for computers carried through to using them for writing. These students now write school reports, take essay exams and do creative writing assignments with the word processor.

In the East York schools in Ontario, Canada, children from second to eighth grade are being given the opportunity to use word processing programs. Teachers of all grades indicate that children are eager to write with the computer. After some initial typing practice, the children find it much easier than using a pencil, pen or typewriter. They write more, edit more, and produce better compositions.

The teachers and researchers who have observed children using word processors report very consistent patterns of change in the children's writing. The most immediate result is that students want to write more often and produce longer compositions. Teachers of young children have reported that the length of the average essay doubles. The next change occurs when the children become familiar with the editing capabilities of the word processor. First they start being more careful to correct typing, spelling and punctuation errors. Then they begin to change words and sentences. Finally, they learn to reorganize the material, moving, adding and deleting large sections of text. They no longer just edit for details but also pay more attention to the meaning of ideas and the order of presentation.

Overall, using word processors has been very beneficial in the classes we have seen. The children enjoy writing more, they are more willing to revise their work, and they produce better essays. They take pride in the quality of their writing, the final neat print-outs, and the fact that they know how to use a computer.

The potential of word processors as a creative teaching resource is just beginning to be realized. We have heard of a number of interesting ideas that are now being developed or tested. The Department of Education has contracted with Bolt Beranek and Newman, Inc., a company in Cambridge, Massachusetts, to develop a writing curriculum using computer technology. The aims of the project are to develop tools to facilitate writing, and environments to encourage and motivate writing. One plan is to set up a within-school computer based message system. Students can use this to exchange information, take surveys of students' and teachers' opinions, ask for information, and other such uses. Using the message system will require learning to use a computer and text editor.

Researchers from the University of California at San Diego are developing a new use of computers in schools. They will have children in California exchange written messages with children in Alaska. All of the writing will be done on computers and the messages will be sent via an electronic communication system. This will allow immediate responses and on-line interactions.

Another possibility for using word processing is interactive stories in which the child helps create a story as he or she reads it. This can be done in various ways. In one use of interactive stories, the children are given incomplete stories. They then use the word processor to fill in the missing parts and perhaps change parts of the original story. Another possibility is to give children paragraphs describing various events. The children choose which events to put in their stories, and how these events should be sequenced and interrelated.

Getting Started With Word Processing

Once you have a computer, you need to add a word processing program and a printer to use it for producing and editing text. There are many word processing programs available for each of the widely sold computers. In a future column we may review some of them with an eye towards classroom rather than office use.

Two general points merit mentioning here. One is that children seem best able to use programs with what are known as *screen editors*. With a screen editor, what you see on the screen is what you get on the printer, and making a change on the screen automatically makes the same change in the computer's memory. The other point is to remember that for most classroom purposes you do not need as sophisticated and expensive a word processing program as you would want in an office.

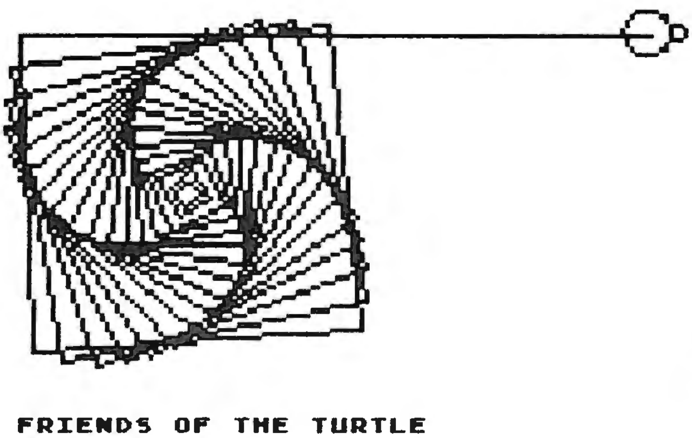
There are three classes of printers. Thermal or electrostatic printers are the least expensive to buy. They print quickly and are relatively quiet. Their disadvantage is that they require special paper and, if they are to be used a great deal, this can become very costly. The next type is the dot matrix printer. These are more expensive to buy than thermal printers, but since they use regular paper they may be more economical in the long run. Dot matrix printers are widely used and are the best choice for most classroom word processing applications. Their main disadvantage is they tend to be noisy. The third type is typewriter-quality printers. These produce the nicest print, but are much more expensive. When checking into printers, be sure to check the cost of the interface you will need to attach the printer to your computer.

One worry is that children have to learn to type in order to use word processing programs. We have found that with just a little practice most children prefer typing to writing with a pen or pencil. Also, several programs, such as *Typing Tutor* by Microsoft, are available to help master typing.

We do not have the space here to mention all the relevant projects, ideas and products. (Fortunately, we do all our writing on a word processor, so that when we realized we had written too much, it was easy to edit and reorganize this article to fit our space.) We have covered just a few of the many possible uses of word processing programs in education. We hope to hear from you about other innovative projects and ideas. ©

Friends Of The Turtle

David D. Thornburg
Los Altos, CA



Procedures And Pathways

All turtle languages incorporate at least two basic commands – one to move the turtle forward and another to make it turn. In Atari PILOT, for example, one can have the turtle draw a 40 unit square by entering the commands:

```
GR: DRAW 50
GR: TURN 90
GR: DRAW 40
GR: TURN 90
GR: DRAW 50
GR: TURN 90
GR: DRAW 40
GR: TURN 90
```

Figure 1.

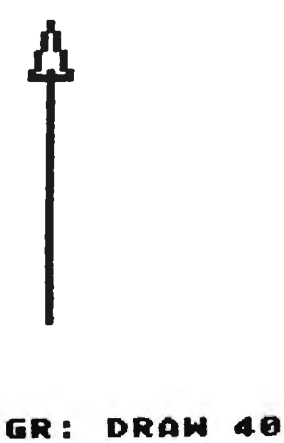


Figure 2.

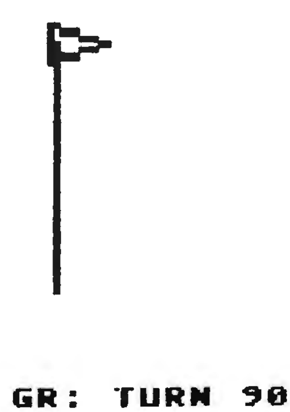


Figure 3.

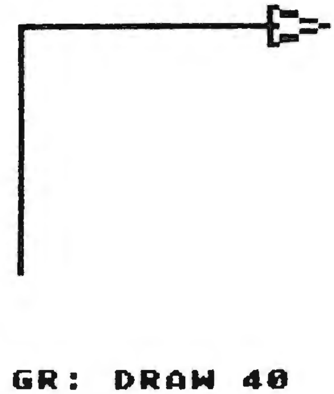


Figure 4.

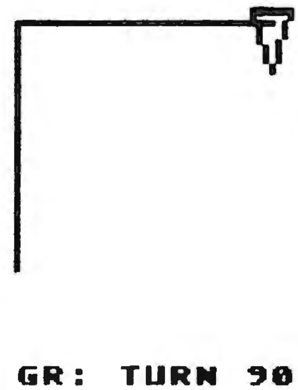


Figure 5.

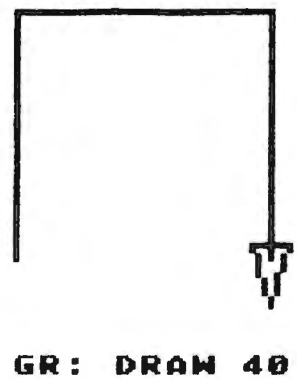


Figure 6.

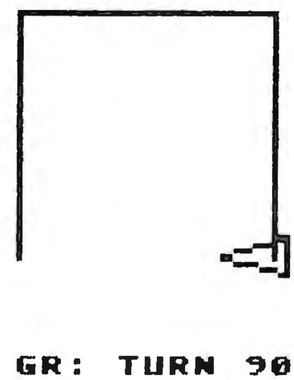


Figure 7.

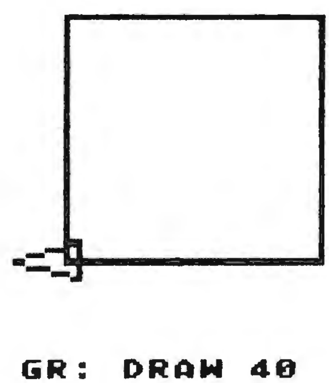
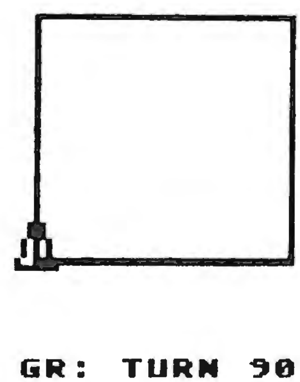


Figure 8.



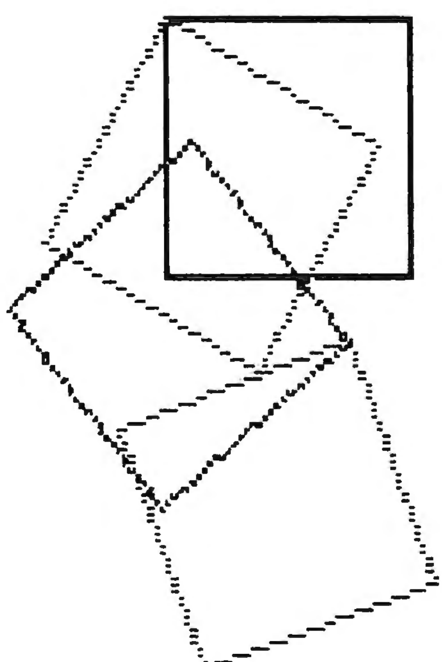
If you want lots of these squares, most turtle environments will let you create a *procedure* which can be used anytime you want to draw this figure. In our case (using Atari PILOT), the procedure starts with a name (for example, *SQUARE). Next, the commands shown above are entered, and finally the *end* command is entered. In PILOT this last command is simply E:.

Once a procedure is defined, it can be used to create copies of squares at any screen location,

orientation, or color you may desire. In our case, one simply uses the procedure with the *use* command; e.g., U: *SQUARE. In this manner, procedures let you extend the number of things the turtle can "understand". To see how handy this is, look at the following program which draws several squares:

```
GR: PEN YELLOW
GR: GOTO -30,0
U: *SQUARE
GR: PEN BLUE
GR: TURN 30
U: *SQUARE
GR: GOTO 20,30
GR: TURN 40
U: *SQUARE
GR: PEN RED
GR: TURN 70
U: *SQUARE
```

Figure 9.



While this isn't a particularly pretty picture, it does illustrate how to use procedures to save a lot of typing! Procedures also make programs easier to read.

An even greater value of procedures is the freedom they give you while you are writing a program. As you think about what you want your program to do, you can write the program in outline form, with procedure names being used for those activities you haven't fully defined. Next, you can create each procedure and test it out independently of the others to make sure it works. In this way you can make steady progress from the outline to the final program without having to deal with massive numbers of statements at a time. I tend to keep procedures short and sweet – and to use lots of them.

The next topic for this month is the idea of a closed *pathway*. Closed turtle paths have some interesting properties. If you look at the figures shown above for the square, you might think that we were done when we drew the fourth side (Figure 7). If you think about it some more, you will see that the turtle is back at the place where it started, but that it hasn't returned to its original orientation. Closed turtle pathways have the property that the turtle returns to its original location and orientation at the end of the trip. This is a very important point to remember.

Now that we have defined a pathway, let's look at a simple way to create some special closed paths in Atari PILOT. One type of closed path creates geometric shapes called regular polygons. A regular polygon is a closed figure which is made from equal length sides and equal turning angles. While we could repeat our DRAW and TURN commands for each side and angle, this would make our procedures very long and tedious to type out. Fortunately, Atari PILOT allows some shorthand to make this task easier. For example, the command:

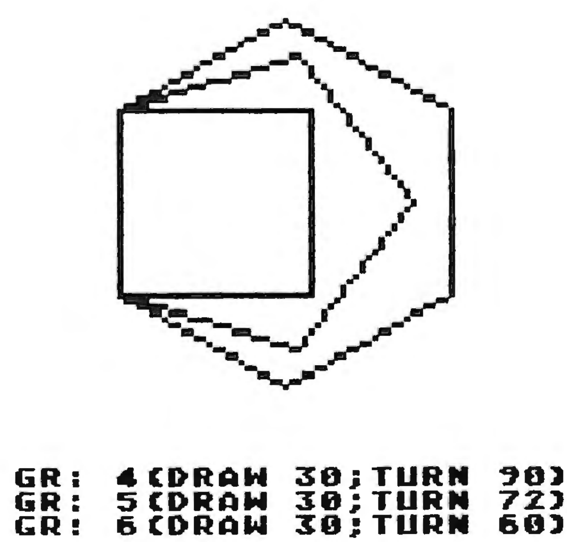
```
GR: 4(DRAW 30; TURN 90)
```

will draw a square on the display screen. The command says, in effect, "Repeat, four times, the commands DRAW 30 and TURN 90".

Using this shorthand, we can create several polygons to study.

```
GR: 4(DRAW 30; TURN 90)
GR: 5(DRAW 30; TURN 72)
GR: 6(DRAW 30; TURN 60)
```

Figure 10.



We have created three closed paths – a square, a pentagon, and a hexagon. If you look at the commands which created these figures, you will notice that the only thing that changed was the number of sides and angles, and the amount that was turned each time. If you are really on your toes, you might have noticed that the total amount turned for each figure was the same: 4x90 = 360, 5x72 = 360, and 6x60 = 360. The total amount of turning for simple closed paths is 360 degrees, regardless of the number of sides on the polygon. This is called the Turtle Total Trip Theorem, and it is a beautiful unifying concept that makes turtle geometry quite valuable.

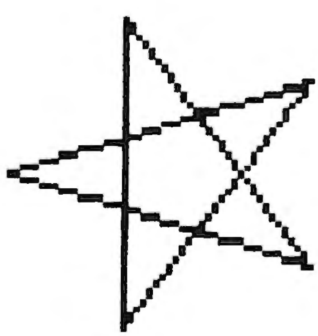
If you would like some challenges until next time, think about these two problems.

1. Can you use the Turtle Total Trip Theorem to help you make a figure which looks like a circle?

2. Look at the picture which results from this command:

GR 5(DRAW 50; TURN 144)

Figure 11.



How much total turning did this figure require? Why?

Until next time, keep those turtles moving, and send me ideas, pictures, programs, and anything else you want to share with your fellow members. Friends of the Turtle chapters should be started in your home town. Let me know what you are doing.

Resource List

Turtle graphics is increasing in popularity both as an educational and as an artistic tool. From time to time, we will publish updates of books, languages, and organizations which incorporate and/or describe turtle geometry. As you look at this list, you might find that I have left some important references out – please let me know what is missing! In the meantime, here is a beginning list to get us started.

Books:

Mindstorms: Children, Computers, and Powerful Ideas by Seymour Papert (Basic Books, 1980).

Turtle Geometry: The Computer as a Medium for Exploring Mathematics, by Harold Abelson and Andrea diSessa (MIT Press, 1981).

Computer Languages and Products:

Big Trak (programmable robot vehicle from Milton Bradley)

Atari PILOT (language cartridge for Atari 400 and 800 from Atari)

TI LOGO (language cartridge for the TI 99/4 and 99/4A from Texas Instruments)

WSFN (language disk or tape for the Atari 400 and 800 from Atari Program Exchange)

WSFN (language tape for the Commodore PET from Peninsula School Computer Project, Peninsula Way, Menlo Park, CA 94025)

Organizations:

Young People's LOGO Association
1208 Hillsdale Drive
Richardson, TX 74081

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

Large Alphabet For The VIC

Doug Ferguson
Elida, OH

There are many exciting applications for the 64 programmable characters on the VIC-20. David Malmberg's article in the first issue of *Home and Educational COMPUTING!* explains fully how the VIC can generate programmable characters merely by changing the contents of memory location 36869, and by redefining the 64 eight-pixel tall characters beginning at 7168.

Another interesting memory location in the VIC is nearby: 36867. Changing its contents creates double-sized characters. By POKEing a 47 into 36867, the bottom border of the screen drops out of sight and vertically-paired characters occupy "stretched" screen locations. After clearing the screen, type an *A* and get $\overset{B}{C}$. Actually, the VIC's first character is the "@" (screen POKE 0) which yields $\overset{@}{A}$. Continue to type the alphabet and see how the stacked letters follow a pattern. To return to normal, POKE 36867,46 or hit the RESTORE and RUN/STOP keys simultaneously.

I set about to combine these two ideas so that I could get a large alphabet. I painstakingly reprogrammed the B to look like the top of a stretched "A" and the C to look like its matching bottom half. Continuing on for nearly two hours, I made it to the "O" and gave up for the night.

Somehow, the clear light of day the next morning directed me toward a much simpler approach: if the characters already reside in ROM, just read each eighth of a character *twice* into the RAM space for programmable characters to program two letters at a time!

Clearly, only 32 such stretched characters can be made since only 64 unstretched characters can be readily programmed. The space key and all the numerals fall in the wrong half of the 64, but all 26 letters of the alphabet can be stretched with the following, surprisingly short, program:

```
10 POKE 56,28: REM RELOCATE END-OF-MEMORY
   POINTER
20 CH=32776: REM LOCATION OF ALPHABET
   IN ROM
30 FOR X=7184 TO 7600 STEP 2: REM ALPHABET
   IN RAM
40 POKE X, PEEK(CH): POKE X+1, PEEK(CH):
   REM STRETCH
50 CH=CH+1: NEXT X: REM LOOP
60 POKE 36879,25: REM NO MORE BORDER
70 POKE 36869,255: REM PROGRAMMABLE
   CHARACTERS
80 POKE 36867,47: REM STRETCHED CHARACTERS
90 PRINT "(clear)ABCDEFGHIJKLMNOPQRSTUV
   WXYZ": END
```

Lines 20 through 50 read the normal alphabet (8x8 pixels) out of ROM and into RAM. Since RAM is also where a longer program will do its work, line 10 tells the computer not to go beyond 7134 (28 times 256). Line 60 is for the purist who notices the lack of a bottom border with the "normal" screen.

Simple? Certainly. The biggest drawback is the lack of numerals and spaces. In string variables with spaces, e.g., A$="HELLO THERE", the space can be replaced by the symbol for cursor-right.

The applications of this large alphabet program are left to the reader. Although it is obvious that any characters can be programmed for stretching, only the alphabet (and a few insignificant symbols) can be programmed in a way that an exact keyboard-to-character correspondence can be realized.

I would appreciate hearing from anyone who can expand on this or who has a clever application. ©

SOFTWARE • HARDWARE

FOR THE **VIC 20®**

COMMUNICATIONS AND COMPUTER TOOLS

- **UMI RS232 COMMUNICATOR INTERFACE** — **$49.95**
  Our RS-232 Communicator Interface enables the VIC-20 to talk to most RS232 devices.
- **VICTERM A** — **$19.95**
  With the VICTERM A program, begin telephone communications.
- **UMI 3K RAM CARTRIDGE** — **$39.95**
  The easiest way to have 6655 bytes of program memory in your VIC.
- **UMI 3K RAM EXPANDER** — **$79.95**
  For programmers, ROM developers and advanced users, our 3K RAM EXPANDER adds two ROM slots and 3K of RAM memory to the Commodore VIC-20 to give a total of 6655 bytes of available user memory. An 8-position dipswitch provides switch - selectable base address and single-socket enable/disable to two ROM sockets.
- **UMI 8K RAM CARTRIDGE** — **$89.95**
  Our 8K RAM CARTRIDGE adds 8K of user memory to the VIC-20.
- **UMI SOCKETED 8K RAM/ROM BOARD** — **$29.95**
  Our 8K board has four sockets for mixing and matching RAMS, ROMS, or EPROMS.
- **BUTI** (BASIC PROGRAMMER'S UTILITY ROM) — **$34.95**
  A beauty of a ROM that plugs into CVH-0002, our 3K RAM EXPANDER, or CVU-0006, our RAM/ROM/EPROM Board.
- **UMI VIC-20 REFERENCE CARD** — **$4.95**
  Our 8-1/2" by 11" laminated reference card is packed with commonly needed Commodore VIC20 information.

PROGRAMS FOR THE COMMODORE PET®

- **SATELLITES AND METEORITES** — **$49.95**
  Satellites and Meterorites is a greater challenge than most space pilots have met.
- **SKYMATH** — **$12.95**
  Decimal addition and subtraction with super PET graphics!
- **MATH SERIES DISKETTE I** — **$29.95**
  Pedagogically correct programs SKYMATH, SPACEDIV, and LONGDIV, on a 5-1/4" Diskette for Commodore Disk Drives.
- **SPACEDIV** — **$12.95**
  Division practice with graphics that hold the child's interest.
- **LONGDIV** — **$14.95**
  20 problems of step-by-step long division with super graphics and 1-inch high digits.
- **SUPERHANGMAN** — **$16.95**
  Superior graphic version of the popular spelling game.

GAMES AND ENTERTAINMENT

- **KIDDIE PAK I** — **NOW $39.95**
  The KIDDIE PAK I includes the following games
  KIDDIE CHECKERS (SEPARATE PRICE $7.95)
  Checkers for very small children.
  SHAPE MATCHER (SEPARATE PRICE $11.95)
  Fun for very small children, matching hidden patterns
  DOGGIE MAZE (SEPARATE PRICE $11.95)
  A dog's race against time to reach the hydrant
  FUN TENSES (SEPARATE PRICE $11.95)
  Nonsense sentences which aid children
- **SUPER FOUR I** — **NOW $49.95**
  SUPER FOUR I includes the following games.
  3-D MAZE (SEPARATE PRICE $14.95)
  Choose a 4 to 9 level maze, then find your way down the hallways
  BREAKOUT (SEPARATE PRICE $14.95)
  No program library is complete without *BREAKOUT!*
  CAROM (SEPARATE PRICE $14.95)
  Planning and skillful play with deflector zaps the target.
  RACEWAY (SEPARATE PRICE $14.95)
  Beat the competition with a high-speed race car.
- **FABULOUS FOUR I** — **NOW $59.95**
  Fabulous Four I contains the following games
  STAR WARS (SEPARATE PRICE $16.95)
  Blast the tie fighter!
  DRAGON MAZE (SEPARATE PRICE $16.95)
  Lurking man eating dragon.
  LASER WAR (SEPARATE PRICE $18.95)
  Blast away space debris and aliens.
  INVADER FALL (SEPARATE PRICE $21.95)
  Alien paratroopers descend from space (Requires 3K Expn.)
- **SUPER STARS I** — **NOW $69.95**
  SUPER STARS I includes the following games:
  THE ALIEN (SEPARATE PRICE $24.95)
  Fast work with inflatable alien traps. (Requires 3K Expn.)
  AMOK (SEPARATE PRICE $18.95)
  Killer robots with one instruction - get the intruder.
  ALIEN BLITZ (SEPARATE PRICE $24.95)
  Split second reactions at the 9th level of ALIEN BLITZ will find how good you are
  GLOBBLER (SEPARATE PRICE $24.95)
  The Conqueror Worm gobbles globs under your control.

ORDER NOW WHILE SUPPLIES LAST!

EDUCATION

- **SUPER ADDITION, SUBTRACTION, DON'T FALL** — **$14.95**
  One cassette contains all three educational games.
- **SKYMATH** — **$14.95**
  With excellent graphics, SKYMATH shoots 5-digit, 3-place addition and subtraction problems into screen's "sky" in 1-inch high digits.
- **SPACEDIV** — **$14.95**
  Division practice with graphics that hold the child's interest.

GAME PROGRAM CARTRIDGES NOW AVAILABLE—ORDER NOW!

- **DEALER INQUIRIES INVITED**
- **ATTENTION SOFTWARE DEVELOPERS**
  Please contact UMI for distribution and top royalties. **Blank Cartridge kits available in quantity.**
- **MASTERCARD/VISA Accepted**

united microware industries, inc.

• 3431 H POMONA BLVD. • POMONA, CA 91768 • PHONE (714) 594-1351 •

Clip Coupon for FREE CATALOG

united microware industries inc.
3431 H Pomona Blvd
Pomona, CA 91768

Please send me my FREE CATALOG describing your Hardware and Software Products

NAME ____________________

ADDRESS ____________________

CITY ____________ STATE ________

ZIP ________ COMPANY ____________

Concentration

Charles Brannon
Editorial Assistant

One application of a user-definable character set is high-resolution, five-color games in GRAPHICS modes one and two. For example, the invaders in Atari's Space Invaders game are GRAPHICS 1 characters. The illusion of smooth motion is performed with the aid of a special feature of the Atari, horizontal fine scrolling. Although my game is less ambitious, it shows what you can do with minimum effort – I spent no more than three hours programming – from the design to the finished game.

The game is based on the card game "Concentration." Two decks of cards are thoroughly shuffled together, then laid out in a matrix of 8 by 13 cards. Each player takes his turn by turning over two cards. If they match, they are removed from the set and this "point" is credited to the player. If not, they are flipped back over. The game continues until all the cards have been matched and removed.

The Atari Version Is Slightly Different

The Atari version of the game is rather different, but the idea is similar. Nineteen different graphics symbols (people, sailboats, "happy faces," cars, etc.) are randomly hidden in a 16 by 20 array. When the game is run, the computer draws the "board," a solid green rectangle. It then flashes the prompt "START/SELECT" at the bottom of the screen. Press [SELECT] to change the number of players, and [START] to begin play. A solid red cursor is placed at the top left corner of the board. Move the cursor with joystick #1 (everyone uses the same joystick). When you wish to "flip" a card, press the red button. Then try to match the revealed symbol by selecting another. If successful, your score is increased by one. The play then passes to the next player. Since the array is 16 by 20 elements, (a total of 320) there could be as many as 160 matches. Unlike the card game version, there are multiple pairs of each symbol. This could make for a very long game, so, instead, the first player to get ten matches wins. SuperFont (**COMPUTE!** #20) could be used to design other gaming characters.

```
100 REM |   Concentration   |
110 REM
120 REM (C) 1981 Small Systems
            Services, Inc.
130 REM Charles Brannon 12/03/81
140 REM
150 GOSUB 740
160 GRAPHICS 1+16:POKE 756,BASE
170 POKE (PEEK(560)+256*PEEK(561)+3),7+6
4
180 SETCOLOR 2,0,10:SETCOLOR 4,6,0:SETCO
LOR 1,12,6
190 IF T=0 THEN DIM A(16,20),CH$(20),SC(
4),PROMPT$(24)
200 FOR I=1 TO 4:SC(I)=0:NEXT I
210 CH$=")*+,-./:;<=>?@[\]^_"
220 COLOR 1
230 PROMPT$="|START |SELECTSTART |SELECT
|"
240 FOR Y=1 TO 20:FOR X=1 TO 16:A(X,Y)=I
NT(19*RND(0)+1):PLOT X+1,Y+2:NEXT X:NEXT
 Y
250 POSITION 3,0:? #6;"|concentration|"
260 NP=1:POSITION 2,2:? #6;"ABCDEFGHIJKL
MNOP":FOR I=1 TO 20:COLOR 224+I:PLOT 1,I
+2:NEXT I
270 POSITION 5,1:? #6;"|PLAYERS| ";NP:PO
KE 53279,8:POKE 20,26:K=0
280 IF PEEK(20)>25 THEN POSITION 4,23:?
#6;PROMPT$(1+K*12,12+K*12):POKE 20,0:K=1
-K
290 IF T THEN 310
300 T=PEEK(53279):IF T=7 THEN T=0:GOTO 2
80
310 IF PEEK(53279)=T THEN 310
320 IF T=5 THEN NP=NP*(NP<4)+1:T=0:P=T:G
OTO 270
330 IF T<>6 THEN 300
340 POSITION 4,23:? #6;"            "
350 REM MAIN LOOP
360 P=P*(P<NP)+1:POSITION 2,1:? #6;"|PLA
YER| ";P;" score ";SC(P)
370 GOSUB 610:X1=X:Y1=Y:V1=V
380 GOSUB 610:IF V=V1 THEN 450
390 SOUND 0,20,2,8:SOUND 1,100,12,8:FOR
W=1 TO 50:NEXT W:SOUND 0,0,0,0:SOUND 1,0
,0,0:POSITION 5,23:? #6:"|PRESS FIRE|"
400 IF STRIG(0)=1 THEN 400
410 IF STRIG(0)=0 THEN 410
420 POSITION 5,23:? #6;"            "
430 COLOR 1:PLOT X+1,Y+2:PLOT X1+1,Y1+2:
SOUND 0,12,12,8:FOR W=1 TO 20:NEXT W
440 SOUND 0,0,0,0:GOTO 360
450 FOR I=1 TO 15 STEP 0.4:SOUND 0,I*17,
12,I:SOUND 1,I*17,12,I:NEXT I:SOUND 0,0,
0,0:SOUND 1,0,0,0
460 SC(P)=SC(P)+1:POSITION 17,1:? #6;SC(
P):FOR I=1 TO 10:POKE 709,PEEK(53770):PO
KE 53279,0
470 FOR W=1 TO 10:NEXT W:NEXT I:POKE 709
,198:IF SC(P)=10 THEN 520
```

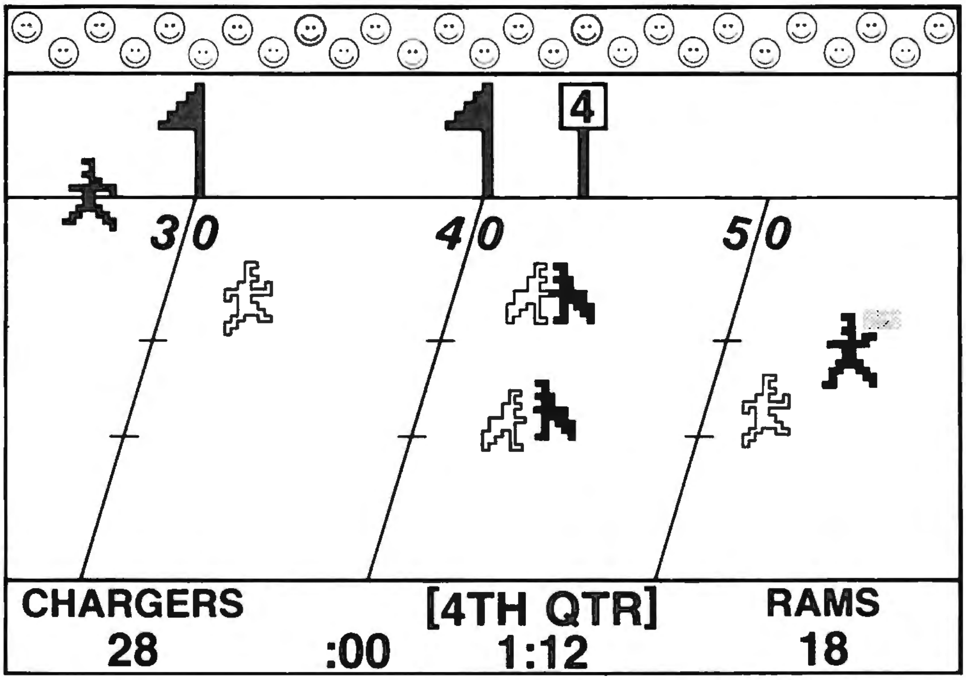
,198:IF SC(P)=10 THEN 520

```
480 POSITION 5,23:? #6;"IPRESS FIRE!"
490 IF STRIG(0)=1 THEN 490
500 IF STRIG(0)=0 THEN 500
510 POSITION 5,23:? #6;"             ":GOTO
 360
520 POSITION 0,2:? #6;"player number ";P
;" |wins!":POKE 53279,8
530 FOR I=0 TO 15 STEP 0 4:SETCOLOR 4,I,
6+INT(4*RND(0)):SOUND 0 10+5*RND(0).10,4
:SOUND 1,50+10*I,12,8:NEXT I
540 SOUND 0.0.0,0:SOUND 1,0,0,0:SETCOLOR
 4,6,0
550 T=PEEK(53279):IF T=7 THEN 550
560 IF T<>3 THEN 160
570 FOR X=1 TO 16:FOR Y=1 TO 20
580 LOCATE X+1,Y+2,Z:IF Z<>1 THEN COLOR
Z-128:PLOT X+1,Y+2:GOTO 600
590 COLOR ASC(CH$(A(X,Y)))+128:PLOT X+1,
Y+2
600 NEXT Y:NEXT X:GOTO 550
610 X=1:Y=1
620 LOCATE X+1,Y+2,Z:COLOR Z+32-160*(Z>1
):PLOT X+1,Y+2:TX=X:TY=Y
630 ST=STICK(0):TR=STRIG(0):IF TR=0 AND
Z=1 THEN 720
640 IF PEEK(53279)<7 THEN COLOR Z:PLOT X
+1,Y+2:GOTO 550
650 IF ST=15 THEN 630
660 T=INT(100*RND(0)+50):SOUND 0,T,10,8:
SOUND 1,T+20,10,8
670 IF ST=14 OR ST=10 OR ST=6 THEN Y=Y-1
:IF Y<1 THEN Y=20
680 IF ST=9 OR ST=5 OR ST=13 THEN Y=Y+1:
IF Y>20 THEN Y=1
690 IF ST>8 AND ST<12 THEN X=X-1:IF X<1
THEN X=16
700 IF ST>4 AND ST<8 THEN X=X+1:IF X>16
THEN X=1
710 COLOR Z:PLOT TX+1,TY+2:SOUND 0,0,0,0
:SOUND 1,0,0,0:GOTO 629
720 FOR I=1 TO 7:COLOR 1+I:PLOT X+1,Y+2:
SOUND 0,100+I*10,12,8:FOR W=1 TO 20:NEXT
 W:NEXT I:SOUND 0,0,0,0
730 V=A(X,Y):COLOR ASC(CH$(V,V))+128:PLO
T X+1,Y+2:RETURN
740 REM INITIALIZE CHARACTER SET
750 BASE=PEEK(106)-8:CHSET=BASE*256
760 GRAPHICS 2+16:POSITION 3,4:? #6;"|C|
On|cE|NtirA|Ti|oN|"
770 POSITION 2,6:? #6;"patience please"
780 FOR I=CHSET TO CHSET+127:READ A:POKE
 I,A:POKE 712,A:SOUND 0,A,10,8:NEXT I
790 FOR I=CHSET+26*8 TO CHSET+32*8+7:REA
D A:POKE I,A:POKE 712,A:SOUND 0,A,10,8:N
EXT I
800 FOR I=CHSET+59*8 TO CHSET+63*8+7:REA
D A:POKE I,A:POKE 712,A:SOUND 0,A,10,8:N
EXT I
810 FOR I=128 TO 207:A=PEEK(57344+I):POK
E CHSET+I,A:POKE 712,A:SOUND 0,A,10,8:NE
XT I
820 FOR I=264 TO 471:A=PEEK(57344+I):POK
E CHSET+I,A:POKE 712,A:SOUND 0,A,10,8:NE
XT I
830 SOUND 0,0,0,0:RETURN
840 DATA 0,0,0.0,0,0,0,0
850 DATA 255,255,255,255.255,255,255,255

860 DATA 0,255,255,255,255,255,255,255
870 DATA 0,0.255.255,255,255,255,255
880 DATA 0,0,0,255,255,255,255,255
890 DATA 0,0,0,0,255,255,255,255
900 DATA 0,0,0,0,0.255,255,255
910 DATA 0,0,0,0,0.0.255,255
920 DATA 0,0,0.0,0,0,0,255
930 DATA 24,24,19,124,88,24,20,54
940 DATA 0,0,0,28,20,127,34,0
950 DATA 129,66,60,36,36,60,66,129
960 DATA 0,0,0,96,95,101,5,0
970 DATA 0.16,40,68,254,124,0,0
980 DATA 0,102,102,0,129,66,60,0
990 DATA 0,56,124,84,124,56,40,68
1000 DATA 0,0,68,34,63,34,68,0
1010 DATA 0,195,102,60,126,36,0,0
1020 DATA 66,255,102,90,90,102,255,66
1030 DATA 16,16,16,56,56,56,124,254
1040 DATA 0,0,56,68,84,68,56,0
1050 DATA 0,16,40,68,254,68,40,16
1060 DATA 0,170,108,198,16,198,108,170
1070 DATA 170,85,170,85,170,85,170,85
1080 DATA 16,56,124,254,84,16,16,124
1090 DATA 14,8,12,8,8,56,120,48
1100 DATA 0,255,0,255,0,255,0,255
1110 DATA 24,56,120,8,8,136,127,62
```

©

ARCADE PRO FOOTBALL™



An Armchair Quarterback's Dream Come True!

It's 4th down and 14 on your opponent's 44 yard line. You're out of time outs with just over a minute left in the game. You call it...sweep right, screen pass left or go for the bomb!

Arcade Pro Football™ puts you in the middle of 60 minutes of exciting gridiron play. You control the action *and* call all the shots. It's the most realistic computer sports game available and it's designed exclusively for the Atari 400/800™ personal computers by Arcade PLUS.

With features like these, Arcade Pro Football™ is in a league of its own!

- Play offense *and* defense head-to-head against the computer or against another player!
- Control two full-color, animated teams in 3-D perspective on a 100 yard, scrolling field!
- 4 game variations, including player handicapping!
- Over 25 offensive and defensive play possibilities...passing, *and* catching, running and kicking!
- All the fun and excitement of real Pro Football...penalties, interceptions, fumbles, bad snaps, 30 second clock—even 4-channel sound and a crowd to cheer you on!

Arcade Pro Football™ is available on cassette or disk for Atari® 400/800™ computers with 16K minimum memory from your local Atari computer dealer. Or send $29.95 (cassette)/$34.95 (disk) plus $2.50 postage and handling (California residents please add 6% sales tax).

arcade PLUS

5276 Hollister Avenue Suite 208 Santa Barbara, CA 93111 (805) 683-2305

Comment Your Catalog

Richard Cornelius
Department of Chemistry
Wichita State University
Wichita, KS

Since the first day that I had my Apple II, I have been frustrated by the inability to fully identify stored programs and files except by using long names. Wouldn't it be nice, for example, to have the date of the latest revision of a program stored along with its name? Of course, a person can always make the date part of the name, but I thought that there ought to be a better way. There is a better way. I have written a program to make writing comments in the catalog easy.

Control Characters

You may have already discovered that some control characters can be part of program and file names in the catalog. For example, a CTRL-J at the end of a program name is helpful in formatting the catalog. The CTRL-J is a linefeed which, when entered as the last character in a program name, has the effect of leaving an empty line between that program name and the next one when the catalog is listed. Another control character which can be inserted into a program name is CTRL-G which will make the Apple beep when the name of the program is listed in the catalog.

Most of the other control characters can be entered into program names, but generally they are not particularly useful. One application they do have is based on the fact that control characters in a name do not actually appear on the screen in the catalog, but they must be used in order to access the program on the disk. Their invisibility can provide a measure of security by preventing someone else from readily loading programs off of your disk. (See your Apple DOS manual for a program to detect most of these control characters.)

The control character that I have found useful in creating comments for the catalog is CTRL-H, the backspace character. This character cannot easily be entered directly into a program name. Typing CTRL-H is the same as pressing the left arrow; you can backspace over characters, but the character that you backspace over is deleted from the name as you backspace. The solution to this difficulty is to put CHR$(8) into a string variable that you use as the program name. In *immediate mode,* [not in a program – just type it on the screen directly] try going through the routine below using an initialized disk with only the HELLO program on it:

```
]CATALOG
DISK VOLUME 254
 A 002 HELLO
]D$=CHR$(4)
]NAME$="ABC"+CHR$(8)+CHR$(8)+"DEF"
]?D$"SAVE";NAME$
]CATALOG
DISK VOLUME 254
 A 002 HELLO
 A 004 ADEF
]LOAD ADEF
FILE NOT FOUND
]
```

The lines that start with a "]" prompt are the ones that I typed into the Apple. The others are those that the computer wrote. When I try to load ADEF the computer tells me FILE NOT FOUND because the name is not ADEF, but "ABC"+CHR$(8)+CHR$(8)+"DEF". Although the program name in the catalog appears to be four characters long, if you were to ask ?LEN(NAME$) you would find that it is actually eight characters long.

This information about CHR$(8) is really all that you need in order to be able to write comments into your catalog. You simply create a string variable that contains enough backspace characters to backspace over the letter that identifies the file type and the number that gives how many sectors are occupied on the disk by the file. Once all of that information is backspaced over, the desired comment is entered into the string. The string variable is then used as shown above to SAVE a program – any program. The "comment" is actually the name of a program – whatever program you had in memory when you do the SAVEing – but it doesn't look like a program name because the file type and sector-count information is missing.

Some Limitations

This commenting technique does have its limitations. Names of programs are limited to 30 characters by DOS. Since the first character of a name cannot be a control character, seven backspaces are needed to erase the information that is normally printed. The first character, plus these backspaces, consume eight of the available 30 characters, so only 22 characters can go into a comment. In addition, you have only limited control over where in the catalog the comment appears. This kind of comment is best used for disks on which people are not going to be making many changes. As long as you start with a fresh disk and put the files, programs, and comments onto the disk in the order you wish them to appear, the catalog will come out fine. If you modify programs in such a way as to change their length, then the order of items in the catalog may be changed and the comments will no longer be adjacent to the program name. One more limitation is that hard copies of the catalog are harder to make appear as nice as the screen listing of the commented catalog. If you try to print the catalog directly, the printer will backspace and overstrike the original characters.

This difficulty can be overcome by listing the catalog on the screen and then, using a program such as that by Jeff Schmoyer (**COMPUTE!** #6) to route the screen image to the printer. In spite of these limitations, I have prepared commented catalogs such as the one in Figure 1. Each line of letters is actually a program name, but the only programs of interest are the ones that have the file type and sector count next to them. The other program names serve only as comments, and the actual programs could be anything (or nothing).

Clearly typing all of these names with the CHR$(8) feature inserted could be quite a chore at the keyboard, so I wrote a program to enter the comments into the catalog. The program is called simply "Catalog Commenter" and is a short BASIC (Applesoft) program. The program shows just how long the name can be and lets you either erase or write names. It then gets a catalog so that you can see what you have done. Hitting any key clears the screen and takes you back to the beginning of the program. This program is the one that was used to prepare the catalog Figure 1. After the backspace characters, two spaces are inserted into the initial part of the string variable used for the name. This spacing makes the comments appear lined up with the sector count of the "real" program names in the catalog, but further limits the length of the comments to 20 characters.

Figure 1.

```
DISK VOLUME 254

 A 025 PH PLOT-BUFFER CAPACITY
   (MAIN PROGRAM WHICH
   LOADS OTHER FILES)

*B 002 OR LOADER & LINE ERASE
   (OVERLAYS HIRES PAGE
   2 ONTO PAGE 1 AND
   ERASES HIRES TEXT
   LINES. A$300; A$325)

*B 027 MZCHAR3
   (SPECIAL WHITE CHAR-
   ACTER SET. A$6000)

*B 006 INSTRUCTIONS
   (BINARY TEXT FILE OF
   INSTRUCTIONS.A$8000)

*B 034 COVER PAGE
   (BINARY HIRES FILE.
```

```
100 REM  ** CATALOG COMMENTER**
110 REM BY RICHARD CORNELIUS
120 REM CHEMISTRY DEPARTMENT
130 REM WICHITA STATE UNIV.
140 REM WICHITA, KS  67208
150 REM (316) 689-3120
160 REM  **INITIALIZATION**
170 D$= CHR$(13) + CHR$(4)
180 REM D$ SIGNALS DOS COMMAND
190 N$= CHR$(8) + CHR$(8) + CHR$(8)

200 REM CHR$(8) IS BACKSPACE
210 N$="A"+N$+N$+CHR$(8)+"  "
220 HOME: VTAB 5
230 REM  **GET COMMENT**
240 PRINT "TYPE IN COMMENT"
250 PRINT"---UP TO THIS LONG--"
260 INPUT"";C$
270 PRINT
280 PRINT"WRITE(W), ERASE(E), OR QU
    IT(Q)?";
290 GET G$
300 IF G$= "Q" THEN 410
310 IF G$ <> "E" AND G$ <>"W" THEN
    GOTO 220
320 REM  **CREATE PROGRAM NAME**
330 N$= N$ + C$
340 REM  **WRITE TO DISK**
350 IF G$= "E" THEN 370
360 PRINT D$"SAVE";N$:GOTO 380
370 PRINT D$"DELETE";N$
380 PRINT D$"CATALOG"
390 GET G$
400 IF G$ <> "Q" THEN 220
410 PRINT:PRINT"THE END"
```

©

STARFIGHT3

David R Mizner
Houston, TX

STARFIGHT3 is a program that will let you fight off Klingons to save the Federation. Before you start typing away, a little word of warning is needed. This program *loves* memory. In fact, STARFIGHT3 will use it all up; so be careful entering the program. An extra space added now may cause a "no memory" message later.

Have fun!!!

Program Description

A new Galaxy is generated each time the program is RUN. A random number of stars (maximum of 25) and Klingons (maximum of 3) are generated and, along with the Enterprise, are randomly placed in a 10x10 Galaxy.

The Enterprise is equipped with three photon torpedos for every Klingon, and three shield units. Three hits on the Enterprise from Klingon attacks will deplete its shield, a fourth hit will destroy the enterprise. There will be self-destruction if the Enterprise runs into a star or Klingon while traveling around the Galaxy.

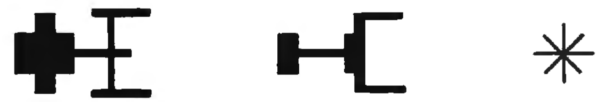
Klingons (all that have not been destroyed) will fire at the Enterprise if your response time for a command is too slow or if your torp misses. Only one hit on the Enterprise is allowed per attack. Take note that the Klingons fire their torps in eight directions while the good guys can only fire in one direction at a time. However, neither side can fire through a star.

The stars and Klingons remain stationary throughout the game.

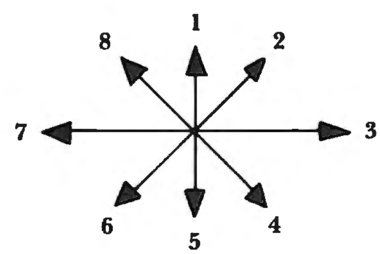
Program Directions

1. Observe operating procedures for VIC20.
2. Commands
   a. Move: VIC will request direction and distance. Direction is a number from 1 through 8, while distance is the number of spaces you want to move.
   b. Torp: VIC will request a direction. Torp does not have a distance since a photon torpedo will travel until it hits a star, Klingon, or Galaxy boundary.
   c. End: This command ends the game. "You surrendered" is the real meaning of "end."
3. Scan
   a. A scan is generated before each command request.
   b. The Galaxy is displayed so you can see the actual location of stars, Klingons, and the Enterprise. At the same time, the direction code is printed out.
   c. Scan code.

Enterprise **Klingon** **Star**



4. Direction
   The direction for moving the Enterprise or firing a photon torpedo is given by entering a number from 1 through 8. These numbers will let you move or fire a torp every forty-five degrees.



5. Changing the game's difficulty
   a. You can change the number of torps allowed by modifying line 120.
   b. Another way is to change the time you are allowed before the Klingons fire. The value of TIS is changed by modifying lines 450,545, and/or 1530.

```
10 PRINT"{CLEAR}   ** STARFIGHT3 **"
20 PRINT:PRINT"DAVID R MIZNER,SEP81"
30 X=PEEK(56)-2:POKE52,X:POKE56,X:POKE51
   ,PEEK(55):CLR
40 CS=256*PEEK(52)+PEEK(51)
50 FORI=CSTOCS+511:POKEI,PEEK(I+32768-CS
   ):NEXT
60 FORI=7168TO7175:READJ:POKEI,J:NEXT
70 DATA15,68,228,254,228,68,15,0
80 FORI=7448TO7455:READJ:POKEI,J:NEXT
90 DATA7,12,204,252,204,12,7,0
100 POKE36869,255
110 DIMA%(10,10),KL(6)
120 FORI=1TO10
130 FORJ=1TO10
140 A%(I,J)=0
150 NEXTJ
160 NEXTI
170 K=INT(RND(1)*3+1):S=INT(RND(1)*25+1)
180 KC=K:T=3*K:H=3
190 FORI=1TOS
200 GOSUB840
210 IFA%(C1,C2)<>0THEN200
```

```
220 A%(C1,C2)=1
230 NEXTI
240 FORI=1TOK
250 GOSUB840
260 IFA%(C1,C2)<>0THEN250
270 A%(C1,C2)=2:KL(I)=C1:KL(I+3)=C2
280 NEXTI
290 GOSUB840
300 IFA%(C1,C2)<>0THEN290
310 A%(C1,C2)=3:E1=C1:E2=C2
320 PRINT:PRINT:PRINT"KLINGONS",K
330 PRINT:PRINT"TORPS",T
340 PRINT:PRINT"STARS",S
350 FORI=1TO3000:NEXT
360 GOSUB860
370 PRINT:PRINT"ENTER YOUR COMMAND"
380 PRINT"1=MOVE 2=TORP 3=END"
390 TI$="000000"
400 INPUTC
410 IFTI$<"000015"THEN440
420 GOSUB1130
430 GOTO360
440 ONCGOTO470,580
450 PRINT">YOU SURRENDERED"
460 GOTO1420
470 PRINT:PRINT"ENTER DIRECTION,DISTANCE"

480 C1=E1:C2=E2:TI$="000000"
490 INPUTC,D
500 IFTI$<"000015"THEN530
510 GOSUB1130
520 GOTO350
530 IFC>8ORD>14THEN490
540 A%(E1,E2)=0:GOSUB670
550 E1=T1:E2=T2
560 IFA%(E1,E2)=1ORA%(E1,E2)=2THENPRINT">
    HIT A STAR OR KLINGON":GOTO1420
570 A%(E1,E2)=3:GOTO360
580 IFT>0THENGOSUB1270
590 IFT>0ANDKC>K0THEN360
600 PRINT">NO MORE TORPS"
610 IFKC>1THEN640
620 PRINT">RAM LAST KLINGON"
630 GOTO470
640 PRINT">YOU'RE OUTNUMBERED"
650 PRINT">FEDERATION IS LOST"
660 GOTO1420
670 ONCGOTO690,700,710,720,730,740,750
680 U=-1:V=-1:GOTO760
690 U=-1:V=0:GOTO760
700 U=-1:V=1:GOTO760
710 U=0:V=1:GOTO760
720 U=1:V=1:GOTO760
730 U=1:V=0:GOTO760
740 U=1:V=-1:GOTO760
750 U=0:V=-1
760 FORI=1TOD
770 T1=C1+I*U:T2=C2+I*V
780 IFT1<1ORT1>10ORT2<1ORT2>10THEN820
790 IFA%(T1,T2)>0THEN830
800 NEXTI
810 GOTO830
820 T1=C1+(I-1)*U:T2=C2+(I-1)*V
830 RETURN
840 C1=INT(RND(1)*10+1):C2=INT(RND(1)*10+
    1)
```

```
850 RETURN
860 PRINT:PRINT" *** SCAN ***"
870 PRINT:PRINT" ++++++++++++"
880 FORI=1TO10
890 PRINT" +";
900 FORJ=1TO10
910 ONA%(I,J)+1GOTO940,960,980
920 PRINT"@";
930 GOTO990
940 PRINT" ";
950 GOTO990
960 PRINT"*";
970 GOTO990
980 PRINT"#";
990 NEXTJ
1000 ONIGOTO1020,1030,1040,1050,1060,1070,
    1080
1010 GOTO1090
1020 PRINT"+  COURSE":GOTO1100
1030 PRINT"+":GOTO1100
1040 PRINT"+     1":GOTO1100
1050 PRINT"+   8 2":GOTO1100
1060 PRINT"+  7   3":GOTO1100
1070 PRINT"+   6 4":GOTO1100
1080 PRINT"+     5":GOTO1100
1090 PRINT"+"
1100 NEXTI
1110 PRINT" ++++++++++++"
1120 RETURN
1130 FORM=1TOK
1140 C1=KL(M):C2=KL(M+3):D=14
1150 IFA%(C1,C2)=0THEN1210
1160 PRINT">KLINGON SHOOTING"
1170 FORC=1TO8
1180 GOSUB670
1190 IFA%(T1,T2)=3THEN1230
1200 NEXTC
1210 NEXTM
1220 GOTO1260
1230 H=H-1:IFH<0THEN650
1240 PRINT:PRINT">ENTERPRISE IS HIT"
1250 PRINTH"SHIELD UNITS LEFT"
1260 RETURN
1270 PRINT:PRINT"PHOTON TORP DIRECTION"
1280 TI$="000000"
1290 INPUTC
1300 IFTI$<"000015"THEN1330
1310 GOSUB1130
1320 GOTO1410
1330 C1=E1:C2=E2:T=T-1:D=14
1340 IFC>8THEN1270
1350 GOSUB670
1360 IFA%(T1,T2)<>2THEN1400
1370 A%(T1,T2)=0:KC=KC-1
1380 IFKC=0THENPRINT"> FEDERATION SAVED <"
    :GOTO1420
1390 GOTO1410
1400 GOSUB1130
1410 RETURN
1420 PRINT:PRINT
1430 INPUT"ANOTHER GAME 1=YES";Z
1440 IFZ=1THEN120
1450 END
```

©

Swirl And Scribble

Matt Giwer
Annandale, VA

Swirl produces extremely complex designs in Graphics 8 which have to be seen to be appreciated. These are not simple sinusoidal or trigonometric plots, but rather are of some artistic merit and may be suitable for logos, letterheads and the like.

The basis for these plots is the set of equations in lines 230 and 235. They arise from the study of modern control theory and are of interest in that a very small change in the two input constants, A and C, can produce a very large change in the shape and character of the plots. The program is easily adaptable to computers other than the Atari by simply plotting the values of X and Y as in line 250. The values of R and T merely center the plot on the screen.

On your first few plots you will notice that, for the first minute or so, the points will all be in a small area in the center. REM lines 2249 and 2250 show how to change line 250 to show this region. To get you started the 501 through 660 REM lines show pairs of values for A and C respectively.

A note of caution: since this uses Graphics 8+16, if the program should end, the display will go back to Graphics 0 and tell you that it is ready. This is why the I loop in line 215 is set to 3000. Although a few hundred would be more than enough to fill enough to fill the screen the extra hundreds hurt nothing and permit unexpected phone calls and the like.

Scribble

A computer program should be scaled to its users. Scribble is a simple program thrown together at the insistance of my six year old who remembered that our last computer had a built in game called scribbling. The Atari would never be up to his standards until there was a way to scribble on the screen. So in order to keep down the heated discussions as to which computer to hook up to the TV I threw this short program together. To my surprise this little program is held higher in his estimation than Star Raiders and is second only to his favorite sea serpent. I offer it here for your child's enjoyment.

To use, a joystick is inserted into position number one and this draws a line on the screen. Pushing the trigger erases the screen. No other provision for operator interaction is made. Keeping it simple kept it popular.

Scribble

```
1 REM NAME SCRIBBLE
1100 GRAPHICS 5+16
1102 COLOR 1
1210 A=STICK(0)
1220 IF A=7 THEN X=X+1
1230 IF A=11 THEN X=X-1
1240 IF A=14 THEN Y=Y-1
1250 IF A=13 THEN Y=Y+1
1260 IF A=6 THEN X=X+1:Y=Y-1
1270 IF A=5 THEN X=X+1:Y=Y+1
1280 IF A=9 THEN X=X-1:Y=Y+1
1290 IF A=10 THEN X=X-1:Y=Y-1
1400 IF X<0 THEN X=0
1410 IF X>79 THEN X=79
1420 IF Y<0 THEN Y=0
1430 IF Y>47 THEN Y=47
1500 PLOT X,Y
1510 IF STRIG(0)=0 THEN GRAPHICS 5+16
1550 GOTO 1210
```

Swirl

```
50 GRAPHICS 0
80 ? :? :? :?
90 ? "INPUT A AND C :";
100 INPUT A,C
110 ? "A=";A;"  C=";C
151 GRAPHICS 8+16:COLOR 1
152 R=150
153 T=85
154 SETCOLOR 2,1,0
155 SETCOLOR 1,4,13
170 X=1
180 Y=1
215 FOR I=1 TO 3000
220 S=X
230 X=A*Y+C*X+S*X*X*(1-C)/(1+X*X)
235 Y=-S+C*X+2*X*X*(1-C)/(1+X*X)
250 TRAP 315:PLOT X+R,Y+T:TRAP 40000
315 NEXT I
```

```
320 GOTO 220
330 END
501 REM 1.01,-1
502 REM 1.01,-.95
503 REM 1.01,-.92
504 REM VERY GOOD, BLACK HOLE 1.01,+0.8
505 REM 1.01,-.1
600 REM 1.0001,-2
601 REM A RANGE .999 AND .992; C RANGE -
2.0055 AND -1.9
650 REM 1.01,0
651 REM 1.008,+.001<>-.001
660 REM 1.008,+.05<>-.05
2249 REM TO SEE THE CENTER OF THESE PLOT
SCHANGE LINE 250 TO
2250 REM LINE 250 TRAP 315:PLOT X*10+R,Y
*10+T:TRAP 40000
4900 END
5000 GRAPHICS 0:LIST 1,330
```

©

WEBS

Loran Gruman
Burnsville, MN

Here is a one-player game for a 40-column PET [or an 80 column machine with the program in **COMPUTE!** #12, pg. 130 loaded — Ed.]. If your machine has sound, turn it on.

```
100 REM WEB WRITTEN 1980 BY LORAN G
    RUMAN 2300 SO SKYLINE DR. ~
    BURNSVILLE
110 REM MINNESOTA, 55337
120 POKE59467,0
130 PRINT"{CLEAR}{02 DOWN}{08 RIGHT
    RIGHT}{REV}WEB INSTRUCTION
    S:{OFF}"
140 PRINT"{02 DOWN}YOU ARE THE NUMB
    ER."
150 PRINT"KEEP THE MOVING NUMBER FR
    OM TOUCHING ANYWEB ON THE ~
    SCREEN."
160 PRINT"{DOWN}THE NUMBER IS CONTR
    OLLED BY PUSHING:
170 PRINT"{DOWN}{03 RIGHT}8=UP      ~
             8"
180 PRINT"{03 RIGHT}4=LEFT          ~
        B"
190 PRINT"{03 RIGHT}6=RIGHT         ~
      4C5C6"
200 PRINT"{03 RIGHT}2=DOWN          ~
        B"
210 PRINT"{03 RIGHT}5=STOP          ~
        2"
220 PRINT"{DOWN}          TEN HITS A
    ND YOUR OUT.{DOWN}"
230 PRINT"{02 DOWN}{04 RIGHT}PUSH A
    NY KEY WHEN READY TO START
    "
240 GETK$:IFK$=""THEN240
250 PRINT"{CLEAR}":A=32768:F=49
260 R=INT(RND(1)*500)+1:Q=A+R
270 GETB$
280 IFB$="4"THENC=-1:S=1
290 IFB$="6"THENC=1:S=1
300 IFB$="8"THENC=-40:S=1
310 IFB$="2"THENC=40:S=1
320 IFB$="5"THENC=0:S=0
330 IFC=40ORC=-40THEN360
340 IFP+C>39ORP+C<0THENC=0:S=0:GOTO
    360
350 P=P+C
360 IFAA=ETHENE=INT(RND(1)*25)+1:I=
    TT:TT=INT(RND(1)*4)+1:AA=0
370 IFTT=1THENQ=Q+1
380 IFTT=2THENQ=Q-1
390 IFTT=3THENQ=Q-40
400 IFTT=4THENQ=Q+40
410 IFTT=4ANDI=3THENQ=Q+1
420 IFTT=3ANDI=4THENQ=Q-1
430 IFQ>33768THENTT=3:GOTO360
440 IFQ<32768THENTT=4:GOTO360
450 LETAA=AA+1
460 POKEQ,81
470 IFA+C>33767ORA+C<32768THENS=0:G
    OTO270
480 T=T+S:IFS<>0THEN GOSUB680
490 A=A+C
500 V=PEEK(A)
510 IFV<>32ANDDV<>FTHENN=1
520 IFV=FTHENN=1
530 IFC=0THENN=0
540 IFN=1THENGOSUB650
550 F=F+N:IFF=58THEN570
560 N=0:POKEA,F:GOTO270
570 PRINT"YOU SCORED A TOTAL OF";T"
    {LEFT} ":PRINT:GOSUB690
580 PRINTTAB(30);"          ";
590 PRINT"                              ~
                         {02 LEFT}{0
    2 UP}"
600 PRINT"DO YOU WISH TO PLAY AGAIN
     (Y/N)";
610 GETPG$:IFPG$=""THEN610
620 IFPG$="Y"THENCLR:GOTO250
630 IFPG$="N"THENPRINT"{CLEAR}THANK
    S FOR PLAYING ":END
640 IFPG$<>"Y"ORPG$<>"N"THEN610
650 POKE59466,0:POKE59467,16:POKE59
    466,15
660 FORNN=30TO90STEP6:POKE59464,NN:
    NEXT
670 POKE59467,0:RETURN
680 POKE59464,150:POKE59467,16:POKE
    59466,15:FORZ=1TO10:NEXT:P
    OKE59467,0:RETURN
690 POKE59466,0:POKE59467,16:POKE59
    466,51
700 FORNN=225TO120STEP-2:POKE59464,
    NN:NEXT:FORNN=120TO255STEP
    2
710 POKE59464,NN:NEXT:POKE59467,0:R
    ETURN
```

Review:

Votrax Type 'n Talk: TNT

Charles Brannon
Editorial Assistant

The concept of the Votrax Type 'n Talk speech synthesizer is simple: you send the device a word, and it pronounces it. For example, the command PRINT#1, "HELLO" would cause the Votrax to say "hello." This makes programming it simple and fun. Other synthesizers can require you to construct words from one or two letter *phonemes*, the simplest units of speech. For example, the word "hello" might be coded as: "H EH3 L O" or "[@X&." Yet another kind of synthesizer lets you send English words, but has a memorized vocabulary which is limited by memory size. What makes Votrax unique is the combination of ease-of-use and flexibility.

The voice is distinct and understandable, but it is obviously artifical. It sounds robotic, similar to the voice synthesizers found in many arcade and electronic pinball games. Both volume and frequency (pitch) can be adjusted with knobs. The voice sounds most natural at its lower frequency.

Built into the unit is a "text-to-speech" algorithm that converts English words into phonemes that can be pronounced by the device – no easy task. Considering the complexity of the English language, it is a remarkably good algorithm, permitting you to generate speech with straightforward PRINT statements. Its arbitrary methods can cause some problems. "COMPUTE!" sounds like "comput." "HELLO" sounds a bit slurred, "HUH LO" sounds better. It is sometimes necessary to intentionally misspell. "COMPUTE!" sounds excellent when spelled "COM PEWT." The space breaks longer words into distinct syllables. Some few words are tougher to generate; for example, MOUSE becomes "mus" (the *ous* is treated like the *ous* in danger*ous*). Spelling it MOWSE doesn't help; it comes out "mose" as in *most*. To solve any such problems you can also program speech directly with phonemes.

Is Votrax for you? It depends on the application. Votrax can be the basis for some fascinating dialogue games, such as ELIZA and Adventure. It can liven up arcade games with threats, taunts, and warnings (We Are The MURLOD Invaders).

Voice synthesis is an alternate (superior?) man-machine interface; it can streamline business (can you imagine your computer saying "Please insert the Word Processing Disk?"). It would be of tremendous aid to the blind, where every character typed could be spoken and, when SPACE was pressed, the preceeding word spoken.

Votrax can be attached to almost any computer, via an RS-232 interface. It can even be attached between the computer and another device, permitting data to be spoken automatically (CompuServe becomes TalkuServe?). Although a one-watt amplifier is built in, you must provide a speaker (eight-ohm).

The significance of the Votrax Type 'n Talk is its text-to-speech routine. It permits beginners to use it immediately, and relieves professionals of the tedium of phoneme construction. The Votrax deserves its acronym – any device that can pack so much power into such a small box is truly TNT!

Votrax Division of Federal Screw Works
500 Stephenson Highway
Troy, Michigan 48084
$375

Review:

Olympia's ES 100 KRO Typewriter/ Printer

Richard Mansfield
Assistant Editor

The ES100, one of a line of Olympia printer-typewriter combinations, can serve as an advanced, stand-alone typewriter with correction facilities or as a computer printer. It contains a built-in RS 232-C serial interface and will work with most personal computers. As one of the new "intelligent" typewriters, it operates somewhat differently from the venerable machines so common only a few years ago.

The first thing you notice is that very little is *mechanical* – you don't move margin stops, you simply set them with left and right margin keys. All keys are repeating, when used with the "repeat" key. Reverse vertical half-line spacing (for super-scripts), choice of two pitches, reverse tabulation, CR without LF, and several line spacings are all key-selected. Unlike the older generation of electrics, most of the formatting and spacing is done from the keyboard. As when using a word processor, you can move around the page without taking your fingers from the keys.

Another feature of this latest generation of typewriters is their feel. They resemble a computer keyboard in layout, versatility, and touch. Instead of a direct mechanical relationship between a pressed key and struck paper, the keys simply click to let you know that they've been acknowledged by the system. The 96-character typewheel responds at 16 characters per second (if you could type that fast). This separation of the mechanical from the keyboard activities makes sense when the printing mechanism does not care whether it gets information from the keys or from a computer.

A green LED shows, on a numbered scale, the precise typing position. The value of some of these features might not be immediately obvious, but, in use, their utility becomes clear. The carriage return without line feed, for example, makes underlining easier. Reverse tabulation means that you don't need to return to the left hand margin to access the tab stops – you can move left through the stops as well as tabbing right, the traditional direction.

Specifications

The typewriter stores functions in an accumulator with the margin release and tab settings "remembered" for 70 to 90 hours. A "correction memory" allows the revision of up to eight characters if the mistake is noticed at once. Depending on the platen size (13/15 inches) the printer supports a maximum paper width of 12.9/15.3 inches and a line length of 11.6/15.5 inches. The unit weighs 30.3/36.3 pounds.

There are 92 characters on the keyboard and line spacing can be either 1, 1½, or double. Horizontal spacing (keyboard selected) is between ten and twelve characters per inch. A variety of type-styles are available on the printwheels and there are five types of ribbon cartridges (black, black/red, carbon, correctable film, or multi-strike).

Using a standard Type D 25 connector, the interface permits odd, even, or no parity bits and the data rate is jumper selectable between 110, 134.5, 150, or 300 baud.

There are a variety of "daisywheel" printers on the market. These printers feature excellent, crisp lettering and typefaces which are easily and quickly changed. The Olympia ES 100 KRO deserves to be considered even if the intent is simply to upgrade an older electric typewriter to the new generation of intelligent electrics. If you ever want computerized, full word processing – the purchase of one of the state-of-the-art electrics would make the transition painless.

Olympia USA, Inc.
Box 22
Somerville, NJ 08896
$1680

©

RPL: A FORTH Sequel?

Jim Butterfield
Toronto, Canada

RPL is a FORTH-related language produced by Samurai Software. There are versions for all PET/CBM machines, and it will fit in systems as small as 8K. It is similar to FORTH in many ways ... but there are fundamental differences.

RPL stands for Reverse Polish Language. This is the backwards-type of coding which calls for you to write X+Y as X Y + and PRINT M as M PRINT. Owners of Hewlitt-Packard calculators will be used to this kind of thing by now and, in fact, it makes good coding sense to do it this way.

Proprietary

Since RPL is a proprietary system, the language must be considered in a different category from FORTH, FORTRAN, or BASIC. It seems unlikely that competing RPL's would be generated by various sources, and RPL literature will be confined to a relatively small community of purchasers.

Timothy Stryker, the author of the language and compiler, has taken many of the characteristics of FORTH, rebuilding and reconceptualizing as he saw desirable. The result will not please FORTH traditionalists – it has a different style from FORTH – but it does form an interesting new language.

Faster? Simpler?

One-to-one comparisons of FORTH versus RPL programs shows that RPL fits in slightly less space and runs slightly faster. This is surprising, since FORTH is known for its compactness and high speed.

Savings in time and memory are achieved, at least in part, by reducing the generality of the language. FORTH works interactively with a user; each program module can be checked out the moment it is typed in, and the user can try things out as he builds his program. RPL is less interactive: the user writes code and then gives the command COMPILE to generate a runnable program. This allows RPL to be more efficient, but reduces user interaction; however, RPL has features to offset this problem during debugging.

Another reason for RPL's speed and compactness is in the internal representation of the program. FORTH uses threaded code, where each "action" of a command is represented by a subroutine address; RPL uses p-code, with each action represented by a token value.

RPL has a streamlined vocabulary of operators; slightly over forty commands are implemented, and all are useful. This compares well with FORTH, which seems to the beginner to be cluttered up with hundreds of commands, many of which are seldom needed by the programmer. The commands are nicely chosen for newcomers; many closely parallel BASIC keywords.

PET/CBM owners will be pleased to see that their machine's characteristics are well supported by RPL. BASIC can co-exist with RPL, and file input/output capabilities are preserved. There's a danger, of course: Programs using "custom" features won't transport well to other computers.

SIM, a symbolic debugger, is sold as a separate package. It allows users to try out sequences of commands before writing them into a program. It has a nice way of presenting the stack visually which may help give users an intuitive feel for how RPL works.

Considerable documentation comes with RPL (60 pages) and SIM (12 pages). The material is nicely written and is quite well done; the approach is tutorial in nature and uses examples liberally.

We've been comparing RPL to FORTH because of the similarities in the languages. RPL deserves to be rated on its own merits.

It's not as easy as BASIC or as pretty as APL. But RPL is fast, compact, and relatively straightforward to program. Users will have to learn to cope with stacks and the backwards-like Reverse Polish Notation. It may take a particular mentality to get hot in an RPL-like language; but the payoff in efficiency can be very good.

Samurai Software, P.O. Box 2902, Pompano Beach, FL 33062. RPL Compiler, $49.95 on disk, $44.95 on cassette; Debugger Compiler and Debugger are ordered together. Specify computer ROM system and disk type.

Review: Ricochet

Richard Mansfield
Assistant Editor

An intriguing new game from Automated Simulations, Ricochet (for the Apple, Atari, or TRS-80) demonstrates why there is so much new interest in games. With the advent of the computer, suddenly there are entirely new categories of games: simulations, interactive adventure stories, exciting hybrids which combine the preplanning involved in traditional strategy games like chess with the visual, physical action of games like pinball.

Ricochet falls into the hybrid category; it has to be seen to be understood, but it's something of a combination of pool and checkers. Each player (you vs. the computer or you vs. a friend) has nine "bars" which initially appear in front of a set of "bumpers." The bars start out in a 4-3-2 pattern, guarding your bumpers, since your opponent can score points by landing in your bumpers.

There are two possible ways to react during your turn. You can change the arrangement of your bars or you can *launch* which sends a ball out from one of your corners ricocheting off walls, bars, and bumpers, and gaining points for each one hit. The ball continues to ricochet until it goes past a bumper into space or hits a corner launcher. Hitting a corner, aside from ringing up points, can render that particular launcher useless for the remainder of a game. You make your moves and launches either from the keyboard or with joysticks.

Broadly defined, the idea is to arrange your bars (which toggle between vertical and horizontal orientation, when hit) so that you best protect your bumpers and launchers. Likewise, you attempt to launch in such a way as to maximize the damage to your opponent.

A Smart Clock

Ricochet takes full advantage of the computer's ability to handle many variations of play. If you play against the computer, it can take on four distinct "personalities" each of which use different strategies. Beyond this, there are five variations of the game itself. In variant two, you can win extra launches, and variant three adds two extra bumpers to each side. Variant five removes all the position markers from the playfield and it becomes more difficult to predict the ricochet effects of a launch.

If a player takes too much time planning or arranging his bars, a *smart clock* starts giving points to the opponent. It is smart because it determines how much is "too much time" by averaging the opponent's decision-making time. In effect, if you make your moves quickly, you force your opponent to move quickly too.

The game is "intelligent" in several senses. If you lose a game, the next game adds point value to your opponent's bumpers while your bumpers retain their original value. This evens things up since you will score more points when you hit the opponent's bumpers.

In the past few years, with computers becoming widely available in homes and game arcades, a variety of new types of games have appeared. Ricochet is an excellent example of this emerging art form.

Automated Simulations, Inc.
P.O. Box 4247
Mountain View, CA 94040
Apple, Atari, TRS-80. $19.95

Review:

Atari Microsoft BASIC (Part I)

Jerry White
Levittown, NY

Editor's Note: This review is in three parts. The second and third parts will appear in **COMPUTE!** *April and May. — RTM*

Not long ago, the Atari Personal Computer owner had two programming alternatives: 8K Atari BASIC and Assembler Language. Now there are three versions of BASIC from which to choose, plus PILOT and PASCAL.

The most recent Basic on the market is called Atari Microsoft BASIC (AMSB). Those of you familiar with other versions of Microsoft will feel right at home with the Atari version. It is said to be the most powerful Microsoft of them all and will certainly make program conversion much easier. The manual provides all the information needed for converting from many other versions of BASIC including PET BASIC, Apple and Applesoft BASIC, Radio Shack Level II BASIC, and Atari 8K BASIC.

This series of articles is being written to help you decide if AMSB is for you. If the Atari is the only computer you've ever had, and 8K Atari BASIC is the only version you've ever used, you will need some specific comparisons to understand the advantages and disadvantages of using AMSB.

Disadvantages??? Yes, although AMSB provides dozens of advantages over 8K Atari BASIC, there are always two sides to every story. So let's get the bad news out of the way first.

The most obvious of the bad news is cost, about $80.00. You'll also need at least 32K RAM and a disk drive since, as of this writing, AMSB is available only on diskette and requires 11,252 bytes more than 8K Atari BASIC. Since the language must load from disk, there's 40 seconds of boot and load time.

Some Tradeoffs

If you can live with the previously mentioned disadvantages, you'll surely find the power and flexibility of AMSB worth looking into. There are, however, a few other sacrifices that must be made by the 8K BASIC user. AMSB has no immediate syntax error checking and permits only two abbreviations, ? = PRINT and ! = REM. Oh how I miss typing GR.0. You also must give up that unlimited length string in trade for string arrays. The 8K STICK, STRIG, PADDLE, and PTRIG commands are not included, but they are easily replaced with PEEK and POKE.

Now for the good news! Here are a few of the most significant advantages AMSB has to offer:

COMMAND	PROVIDES...
AUTO	Automatic line numbering.
COMMON	Variable values are passed from one program run to another.
DEF	Define integer, single, and double precision.
DEL	Delete range of lines from program.
DIM	Three Dimensional Alpha/Numeric Arrays
ELSE	IF THEN ELSE decisions.
INSTR	Search for a small string within a larger string.
MOVE	MOVE a number of bytes from one area of memory to another.
OPTION	Reserve RAM for Assembler Routines, Player Missile Graphics, Redefined Character Sets.
PRINT	AT specified coordinates.
PRINT	(TAB) and (SPC) positioning.
PRINT	USING for formatting output such as right justified currency amounts.
RENUM	Renumber lines and references.
TIME	In 60ths of a second.
TIME$	Current time in HH:MM:SS format.
TRON	Current line number trace display on.
	Turn off trace function.
VERIFY	Verify Program in memory with program on tape or disk.
WAIT	Loop until specified conditions exist.

Many commands are identical in both Atari BASICs. Some commands perform identical functions but are formatted differently. For example, 8K BASIC uses the XIO command for many useful functions. AMSB makes things easier to remember with commands like FILL, KILL, LOCK, MERGE, NAME, and UNLOCK. ASMB uses PLOT TO instead of DRAWTO, CLS instead of ?CHR$(125), and SCRN$ instead of LOCATE.

Some of the other commands available in AMSB include AFTER, CLEAR STACK, EOF, ERL, ERR, ERROR, INKEY$, LEFT$, LINE INPUT, MID$, ON ERROR, RANDOMIZE, SAVE with LOCK, STACK, and STRING(n,X$).

One beautiful feature was added to the SOUND command. An optional fifth variable for duration has been added. The duration is a value of up to 255 JIFFIES (60ths of a second). Up to 25 SOUND commands may be stored on the STACK, eliminating the need for many time delay loops. AMSB can go on to calculations or display work while SOUND commands execute at previously specified intervals.

The ability to define integers allows floating point routines to be bypassed. This can account for significantly faster execution. How much faster, you ask? I'll get into speed comparisons and routine examples in part two of this series.

I Use All Three

Before closing this segment, I'd like to voice some of my own personal opinions. AMSB will certainly find its place in the rapidly growing Atari software market. Both the beginner and experienced programmer can benefit from the wide range of commands offered. The buyer should also be aware of another alternative called BASIC A+.

Anything you can do using AMSB can be done in 8K BASIC with occasional help from an Assembler subroutine. AMSB offers a great deal to the BASIC only programmer, but cannot be used by those with less than 32K RAM or without a disk drive. Personally, I've grown to really appreciate the amazing number of features Atari BASIC has squeezed into an 8K ROM cartridge. I've also learned to appreciate fast binary I/O and the DIR (Disk Directory) feature available in BASIC A+, as well as the speed made possible by the AMSB integer feature. They all have their advantages and disadvantages. Which one do I recommend you ask? I use all three. ©

TELECOMMUNICATIONS

Modem Applications

Michael E. Day
Chief Engineer
Edge Technology

In **COMPUTE!**, *September, 1981, #16, Mr. Day discussed technical specifications for MODEM's. Here he explores several uses for MODEM's in everyday computing. — The Editors*

One of the questions I am often asked is: "Why do I need a modem?" It is interesting that this question would be asked rather than just "Do I need a modem?" since this indicates several things. The need for the modem is already felt.

The feeling of the need for the modem comes about because of the large amount of information presented to the person about telecommunications both in magazines such as this one and in talking to other computer users. This tends to lead to the belief that if you do not have a modem you are not using your computer to its fullest potential. Unfortunately, the reasoning for this belief is not readily apparent. Analysis of the information generally presented on telecommunications shows why this is so. The most common type of information that is presented is of a technical nature. This assumes that you already know why you do or do not need a modem, and are simply after "how does it work" information. The other type of information that is presented is applications information. Again this assumes that you already know why the modem is needed, and that you are simply after the information on how to use it for a particular type of application.

The question *why* is one of the hardest of this type to answer. It cannot be answered directly. When you ask *why*, what you are really saying is give me more information so that I can decide if I really need it. The information that is normally provided is reference information with which you are familiar. In answer to "Why do I need a car?", one might answer "In order to get to and from work." This provides a base point that you can expand upon to gain the information needed to determine how the car would fit into your lifestyle. A response could be "But I can take the bus." with a return of "But what if you work odd hours when the bus doesn't run?" This generates the pros and cons necessary to make a final decision.

The problem that we have with the modem is the same problem that the computer has experienced — a lack of readily discernible common reference points. In answer to why do I need a computer, the easily determined reference points tended to be rather weak, such as to balance your checkbook, or keep records of your gas mileage. Since these could be done far more cheaply with existing alternative methods, they hardly generate a decision in favor of the computer. The computer is slowly overcoming this problem by creating its own reference points. The computer is doing things that were not possible before (controlling heating and lighting to minimize utility bills, or writing letters (or magazine articles) with greater ease than ever before, even playing exciting new games and, as a side benefit, you can balance your checkbook too.

The modem is going through the same stage of development of use. It is a device that has entirely new uses and concepts that are not currently realized, and it must "create" these in our awareness so that they can be realized of their own accord.

Computerized Bulletin Boards

Originally the question was easy to answer, the modem was for the purpose of operating a computer from a remote location. If you had to do this, you had to have the modem. If you did not have to, then you did not need a modem.

Now, however, that use of the modem has been radically altered. With the advent of the personal computer we can put the computer at the remote location along with the user.

If you are only going to use the computer to play games or balance your checkbook, you probably don't need a modem. If you want to communicate with other computer users, however, there is a very good probability that, at some point, you will need a modem.

One of the new uses is the Computerized Bulletin Board Systems that appeared. These are public access message systems which can be used by anyone to post messages or read those left by others. These tend to be messages that don't fit into normal modes of communication and include calls for help, general notices of information, advertisements, classifieds, and personal messages. There is no charge for the use of these systems, they are

HUNTINGTON COMPUTING

Now Selling Atari®, PET®, TRS-80® Software

Atari®

Compu-Math Decimals (disk)	$39.95	now	$35.14
Comp Magic Kayos (disk)	$34.95	now	$30.74
Ali Baba & 40 Thieves	$32.95	now	$28.94
Arcade - Ghost Hunter (cass.)	$29.95	now	$26.34
Arcade - Ghost Hunter (disk)	$34.95	now	$30.74
Comp Magic Kayos (cass.)	$34.95	now	$30.74
Epyx Crush Crumble Chomp (cassette or disk)	$29.95	now	$26.34
Creative ATC (cass.)	$11.95	now	$10.44
Mouskattack	$39.95	now	$35.14
The Next Step	$39.95	now	$35.14
Poker Solitaire (cass.)	$14.95	now	$12.74
Gomoku (cass.)	$19.95	now	$16.94
Reversi (cass.)	$19.95	now	$16.94
Cypher Bowl (cass.)	$29.95	now	$25.44
Rescue at Rigel (cass.)	$29.95	now	$25.44
Star Warrior (cass.)	$39.95	now	$33.94
Invasion Orion (cass.)	$24.95	now	$21.24
Datestones of Ryn (cass.)	$19.95	now	$16.94
Conflict 2500 (cass.)	$15.00	now	$12.74
Empire of the Overmind (disk)	$35.00	now	$29.44
Tanktics (cass.)	$24.00	now	$20.44
Atari* Mailing List (disk)	$24.95	now	$21.24
Atari* Character Generator (disk)	$19.95	now	$16.94
Text Wizard (disk)	$99.95	now	$84.44
Atari* Character Gen. (cass.)	$15.95	now	$13.54
Le Stick	$39.95	now	$33.94
Checker King (cass.)	$19.95	now	$16.94
MicroChess (cass.)	$19.95	now	$16.94
Survival/Adventure (disk)	$24.95	now	$21.24
3-D Supergraphics (disk)	$39.95	now	$33.94
3-D Supergraphics (cass.)	$39.95	now	$33.94
Mind-Bogglers I (disk)	$19.95	now	$16.94
Mind-Bogglers I (cass.)	$15.95	now	$13.54
VersaWriter Graphics Tablet	$300.00	now	$254.44
Hidden Words	$17.50	now	$15.44
Spatial Relations	$17.50	now	$15.44
Word-Scramble	$15.00	now	$13.54
Preschool Fun	$15.00	now	$13.54
Fastgammon (cass.)	$19.95	now	$16.94
Assembler (cass.)	$24.95	now	$21.14
6502 Disassembler (cass.)	$11.95	now	$10.14
6502 Disassembler (disk)	$14.95	now	$12.64
Tank Trap (cass.)	$11.95	now	$10.14
Tank Trap (disk)	$14.95	now	$12.64
Tari Trek (cass.)	$11.95	now	$10.14
QS Forth (disk)	$79.95	now	$67.94
Starbase Hyperion (disk)	$22.95	now	$19.44
Name That Song (cass.)	$14.95	now	$12.64
Jaw Breaker (disk)	$29.95	now	$25.44
Pornopoly (disk)	$29.95	now	$25.44
The Broker (CCI) (disk)	$99.95	now	$84.94
Super Modem Pak (CCI) (disk)	$49.95	now	$42.44
Atari* Snuff (CCI) (disk)	$29.95	now	$25.44
Utility Man (CCI) (disk)	$99.95	now	$84.94
Tanktics (cass)	$24.00	now	$20.44
Fantasyland (disk)	$59.95	now	$50.94
Empire of the Overmind (cass.)	$30.00	now	$25.44
Bridge 2.0 (cass)	$17.95	now	$15.24
Nominoes Jigsaw (cass)	$17.95	now	$15.74
Intruder Alert (cass)	$16.95	now	$14.84
Alpha Fighter (disk)	$18.95	now	$16.44
Compu-read (disk)	$29.95	now	$25.44
Letter Perfect (disk)	$150.00	now	$127.44
Sammy Sea Serpent (cass.) PDI	$16.95	now	$14.34
Cribbage (Thesis) (cass.)	$15.00	now	$13.54
Visicalc	$200.00	now	$149.00
Kross 'N Quotes PDI (cass.)	$16.95	now	$14.34
Star Raiders (cart.)	$39.95	now	$33.94
Stock Charting	$24.95	now	$21.14
Adventure International	10% - 15% off sug. retail		
Atari* Safari (CDS) (disk)			$33.94
Atari* Safari (CDS) (cass.)			$25.44
Computation (Thesis) (cass.)	$15.00	now	$13.94

Order by Phone

800-344-5109

800-692-4146

In California

Pet®

AVALON HILL GAME COMPANY

B-1 Nuclear Bomber (cass.)	$15.00	now	$12.77
Midway Campaign (cass.)	$15.00	now	$12.77
No. Atlantic Convoy Raider (cass.)	$15	now	$12.77
Nukewar (cass.)	$15.00	now	$12.77
Conflict 2500 (cass.)	$15.00	now	$12.77
Planet Miners (cass.)	$15.00	now	$12.77
Computer Acquire (cass.)	$20.00	now	$16.97
Lords of Karma (cass.)	$20.00	now	$16.97

VIC SOFTWARE

UMI Kiddie Checkers	$7.95	now	$6.87
UMI Star Wars	$16.95	now	$14.87
UMI AMOK	$18.95	now	$16.67
UMI Globber	$29.95	now	$21.87

ARTWORX

Teacher's Pet (disk)	$18.95	now	$16.67
Teacher's Pet (cass.)	$14.95	now	$13.07
Vaults of Zurich (disk)	$25.95	now	$22.77

AUTOMATED SIMULATIONS

Introductory 3-Pack (disk) (Rescue, Morloc's, and Datestones)	$49.95	now	$39.97
Rescue at Rigel (cass.)	$29.95	now	$23.47
Temple of Apshai (cass.)	$39.95	now	$33.97
Hellfire Warrior (cass.)	$39.95	now	$33.97
Starfleet Orion (cass.)	$24.95	now	$21.27
Invasion Orion (cass.)	$24.95	now	$21.27
Morloc's Tower (cass.)	$19.95	now	$16.97
Datestones of Ryn (cass.)	$19.95	now	$16.97

PERSONAL SOFTWARE

Checker King (cass.)	$19.95	now	$16.97
Gammon Gambler (cass.)	$19.95	now	$16.97
MicroChess (cass.)	$19.95	now	$16.97
Bridge Partner (cass.)	$19.95	now	$16.97
Time Trek (cass.)	$19.95	now	$16.97

UNITED SOFTWARE OF AMERICA

KRAM (disk)	$99.95	now	$84.97
Super KRAM (disk)	$175.00	now	$153.97
Request (disk)	$225.00	now	$191.27
Thinker (disk)	$495.00	now	$420.77
Space Intruders (cass.)	$19.95	now	$16.97
All MICRO-ED			10% Off List
All Microcomputer Workshops			15% Off List

VIC SOFTWARE

Addcom Missile Commander	$16.66
Channel Vic Data Logger	13.16
MMA Star Command	$8.96
TIS Basic Programming I	$17.46

Apple®

See full page of Apple* products elsewhere in this magazine.

Gorgon	$39.95	now	$33.99
Word Star	$375.00	now	$289.00
Mail Merge	$125.00	now	$106.19
Super Sort	$200.00	now	$169.99
VisiCalc 3.3	$200.00	now	$149.00
Wurst of Huntington Computing			$19.99
Nibble Express	$12.95	now	$11.99
Soft Porn Adventure	$29.95	now	$25.39
Time Lord	$29.95	now	$25.39
French Hangman	$29.95	now	$25.39
Alicia-Sp. bilingual reader	$29.95	now	$25.39
H&H Stock Trader	$190.00	now	$161.49
Grow (CIA)	$35.00	now	$31.49
All Edu-Ware			10% to 15% Off List
VersaCalc	$100.00	now	$84.99
Hebrew	$60.00	now	$50.99
All Serendipity			10% to 15% Off List
All Sirius			15% Off List
All Sybex Courses			10% to 15% Off List
Win at the Races	$39.95	now	$33.99
Disk Prep	$25.00	now	$21.19
PLE Chip	$60.00	now	$50.99

We maintain a huge inventory of software for Apple* and hardware. Call us toll free for the latest programs. We also stock a large supply of computer books. Visit us in person at our new 3300-square foot store at 1945 South Dairy in Corcoran, Calif.

Great Grandma Huntington

Great Grandma Huntington always said to try harder – and we do. We will soon have tee shirts for sale with Great Grandma Huntington's picture on them. Watch for Granny!

TRS-80®

BIG FIVE SOFTWARE

Super Nova (cass.)	$15.95	now	$13.58
Galaxy Invasion (cass.)	$15.95	now	$13.58
Attack Force (cass.)	$15.95	now	$13.58
Cosmic Fighter (cass.)	$15.95	now	$13.58
Meteor Mission II (cass.)	$15.95	now	$13.58

ADVENTURE INTERNATIONAL

Adv. 4-5-6 (disk)	$39.95	now	$35.08
Adv. 1-2-3 (disk)	$39.95	now	$35.08
Adv. 7-8-9 (disk)	$39.95	now	$35.08

ARTWORX

Teacher's Pet (disk)	18.95	now	$16.58
Nominoes Jigsaw (disk) color	$21.95	now	$19.28
Nominoes Jigsaw (cass.) color	$17.95	now	$15.78
Bridge 2.0 (cass.)	$17.95	now	$15.78
Hearts (disk)	$19.95	now	$17.48

BRODERBUND SOFTWARE

Galactic Trilogy (disk)	$39.95	now	$33.98
Galactic Empire (cass.)	$14.95	now	$12.68
Galactic Trader (cass.)	$14.95	now	$12.68
Galactic Revolution (cass.)	$14.95	now	$12.68
Tawala's Last Redoubt (cass.)	$19.95	now	$16.98

DATASOFT

Iago (disk)	$24.95	now	$21.18
Football Classics (disk)	$24.95	now	$21.18
Arcade-80 (disk)	$24.95	now	$21.18
Iago (cass.)	$19.95	now	$16.98
Football Classics (cass.)	$19.95	now	$16.98
Arcade-80 (cass.)	$19.95	now	$16.98
Sigmon (COLOR) (cass.)	$29.95	now	$25.38
SECS (COLOR) (cass.)	$29.95	now	$25.38

ACORN SOFTWARE

Invaders From Space (disk)	$20.95	now	$17.78
Duel-N-Droids (disk)	$20.95	now	$17.78
Pinball (disk)	$20.95	now	$17.78
Pigskin (disk)	$20.95	now	$17.78
Quad (disk)	$20.95	now	$17.78
Basketball (disk)	$20.95	now	$17.78
Gammon Challenger (disk)	$20.95	now	$17.78
Everest Explorer (disk)	$20.95	now	$17.78
SuperScript (disk)	$29.95	now	$25.38
System Savers (cass.)	$14.95	now	$12.68
Invaders From Space (cass.)	$14.95	now	$12.68
Duel-N-Droids (cass.)	$14.95	now	$12.68
Pinball (cass.)	$14.95	now	$12.68
Pigskin (cass.)	$14.95	now	$12.68
Quad (cass.)	$14.95	now	$12.68
Basketball (cass.)	$14.95	now	$12.68
Gammon Challenger (cass.)	$14.95	now	$12.68
Everest Explorer (cass.)	$14.95	now	$12.68
All Automated Simulations			10% to 15% Off List
All Avalon Hill			10% to 15% Off List
All Hayden			10% to 15% Off List
All Microsoft			10% to 15% Off List

FOREIGN ORDERS

We regret that we can no longer accept checks (bank charges were sometimes greater than the amount of the check). We will gladly accept U.S. currency, VISA, MasterCard or American Express at no extra charge, or you may make direct wire transfers to our bank, Security Pacific, Corcoran, CA 93212, for a $6.00 charge. All overseas orders are shipped by air.

VISCALC

Special for Pet®, Atari® & Apple®

Regular $200.00 List

Now $149.00

HUNTINGTON COMPUTING
Post Office Box 1235
Corcoran, California 93212

Order by Phone 800-344-5109
In California 800-692-4146

Apple* is a registered trademark of Apple Computer, Inc.
Pet* is a registered trademark of Commodore.
TRS-80* is a registered trademark of Tandy Corp.
Atari* is a registered trademark of Atari, Inc.

(209) 992-5411

VISA

master charge

We take MasterCard, American Express or VISA (Include card # and expiration date). California residents add 6% tax. Include $2.00 for postage. Foreign and hardware extra. Send for free catalog. Prices subject to change.

wholly supported through donations and out-of-pocket expenses by the owners.

As an outgrowth of the BBS's are the remote computer systems and database systems. Although many of them are open to the general public, they are not readily usable due to the technical knowledge needed. Additionally, these systems tend to be very specific in the application to which they are oriented and are generally of little or no use to the general public.

Because the bulletin boards are privately supported, they are limited in the scope of services they can provide. For those who are willing to pay, there are more elaborate systems available. The most widely-known are Compuserve, The Source, and Micronet. These systems provide a wider range of services including message transfer, information retrieval (stock reports, news, etc.), conferencing, program storage and retrieval, and running programs.

Often there is a need to find information of a more extensive or technical nature than can be provided by the general services systems. This need is provided for by the technical information database systems. These systems are usually oriented around a particular subject area or group of areas. The technical data systems, by being very specific can carry a much wider range of information on a subject than is possible on a general information system. Because this information is also the most expensive to obtain, these systems are the most expensive to use. They can cost over $100 an hour.

Multiuser Systems

Finally we come to the original multiuser computer systems, time-share computer systems. These systems are rented on a usage basis to anyone who needs a computer, but, for some reason, does not have a computer of his or her own available. These are generally used for overflow work, temporary, or occasional applications where it is not possible or practical to use one's own computer. The cost of using these systems can vary widely depending on how the usage is determined.

It is interesting that now that the personal computer has come into being, another application appears to be evolving. This can best be understood by describing the need that has been generated.

If you wish to say something to George who lives down the street, you could go to his house and speak to him directly, or you could call him up on the telephone and talk to him. In the first case there was no *equipment* involved in talking to George, you went to his house. This is *direct* communications. In the second instance you used the telephone to talk to him. Rather than expend the energy to go to George, you used a device which allowed you to talk to George without actually going to his house and thus you were *communicating at a distance.*

If you and George both have a computer and you wish to share programs you have written, there are many ways this could be done. You could put a copy of the program on a cassette or floppy disk and give it to George to read into his computer. This works great if George has a similar computer and can read the tape or disk.

If the two systems are not compatible, another way will have to be found. One way that has been used a lot is for you to simply provide George with a written copy of the program and let him type it into his system. This isn't too bad if the program isn't very long and is in human-readable form. This is the way most magazines provide programs as it is the surest way to cover a wide range of computers. But, as mentioned, if the program is not in a human-readable form, or is excessively long, this method does not work very well.

Computers Talking To Computers

A method of communication that computer hobbists have often used is to directly tie their computers *back to back.* This is a form of *direct communication.* This allows the computers to talk to each other, but has the disadvantage of requiring that both computers be next to each other. To date, it has also meant that the computer operator be fully knowledgable of the way the computer internals work as well as the programming needed to allow the two computers to talk to each other. This can be a bit much for the general user and, in fact, has baffled quite a few experienced computer technicians.

The modem provides a common link that both computers can communicate through. By defining a standard of how the interconnection between the computers is to be accomplished, the problem of how to hook the two computers together is eliminated. What is occuring now is a definition of the method of communication between the computers. Although there are some communications programs in use already, they are currently machine-type dependent. An Apple can talk to another Apple, but it can't talk to an Altair. Most of the programs that are used to allow one computer to communicate to another are in the early stages of development: they allow the communication to occur, but there is little or no provision for options or alternatives. They tend to be very restrictive in their use.

As the need to communicate between different types of computers grows, the communications programs will become simultaneously more comprehensive and easier to use. ©

Machine Language: Loops And Quality

Jim Butterfield
Toronto, Canada

Program loops seem to be a byproduct of laziness. When a programmer tires of writing a series of instructions, he produces a loop to save coding time and processor memory. Yet something more profound happens at the same time: the program usually becomes more generalized.

Suppose I wanted to place the value hexadecimal 20 into locations $8000 to $8027. My first instinct is to code: LDA #$20 : STA $8000 : STA $8001 : STA $8002 ... and so on. Around the time I reach $800B, it will probably occur to me that I'm writing a lot of essentially similar code. Creative sloth comes into play. I observe that the repeated instruction is STA $something. Racking my brains, I decide that if I could vary the "something" part, I could then do most of the job with a variable instruction.

"Indexing!", I cry, and proceed to tear up the old sheets and code LDX #$00: LDA #$20: (loop) STA $8000,X: INX: CPX #$28: BNE (loop). This drops coding to six instructions instead of forty-one and memory usage to twelve bytes instead of one hundred and twenty-two; but the running time increases from 162 to 443 microseconds. There's no use crying over spilt microseconds: the time difference is less than a three-thousandth of a second, and I'll usually happily take it rather than a case of writer's cramp.

But something more important has happened than just mechanics. If I want to convert my first ("hard way") program so that it stored into 64 locations, or stored to address $0400 and up, I have no choice but to rewrite. On the second program which uses loops, it's a snap. A mere stroke of the coding pen, a one or two byte change, and the job's done. We've somehow created a program that's more general and more applicable to a range of tasks.

As we consolidate our program, we have to generalize. And as we generalize, we not only shorten the code: we create sturdier and more broadly applicable code.

A word to those picky bit-and-microsecond counters who will point out that we could save two bytes and a few dozen microseconds by starting our index X at 39 and counting it down to zero. Sure you can. But that kind of picking is not what makes sounder code. We want to look for methods that generalize; they are the ones that will produce sturdy and reliable code ... and perhaps save us a few coding lines and bytes.

A Larger Scale

The same ideas apply to coding that repeats several lines. When you find yourself writing the same code, look for a generalization. Take these two sets of coding:

```
ONE   LDX  #$09          ZRO   LDX  #$06
      PHA                      PHA
ONE1  BIT  CLKRDI        ZRO1  BIT  CLKRDI
      BPL  ONE1                BPL  ZRO1
      LDA  #126                LDA  #195
      STA  CLK1T               STA  CLK1T
      LDA  #$A7                LDA  #$A7
      STA  SBD                 STA  SBD
      ...                      ...
```

The above subroutines are from the tape write program of the KIM. ONE writes a logic 1 to tape; ZRO writes a logic 0 to tape. They are very similar. The only differences are: nine versus six on the first line, and 126 versus 195 on the fifth line. How might we consolidate these two pieces of program?

At the moment, the Y register doesn't seem to be used. We could ask the calling routine to set Y to zero or one, depending on whether we wanted to call ZRO or ONE activities; and then write a common routine:

```
ZONE   LDX  TABLE,Y
       PHA
ZONE1  BIT  CLRKRDI
       BPL  ZONE1
       LDA  TIMING,Y
       STA  CLK1T
            ...etc.
```

We have now consolidated the two routines. The values 6 and 9 which count the number of cycles in each signal are now stored in a table TABLE. The values 126 and 195 which set the timing of each cycle are in a second table TIMING.

Have we accomplished anything other than saving a few bytes of code? Yes, almost accidentally. Now that the number of cycles are stored neatly in a table, we can easily adjust them to change the type of signal we write. In fact, this particular coding was part of the sequence that lead to the introduction of the high speed tape format known as Hypertape.

Deeper...

The programmer doesn't always have free registers, of course; but the methodology of saving registers isn't hard to do.

Where addresses within a program change from routine to routine, the best way to handle this is via indirect addresses. If program 1 searches table 1, and program 2 searches table 2 and so forth, indirect address.

Consider: if you have written a game with planes and tanks moving around the screen, you may find that, with a little work, a single subroutine can move both craft around. Once you have generalized, all sorts of bonuses arrive: the bombs and shells can likely be folded into the same subroutine. Collisions and other effects can now be handled in their generalized form rather than as special coding (did a bomb hit a shell? did a plane hit a bomb? did a shell go off screen? etc...)

What seems to start out as laziness or convenience develops into something more important. In reaching for the general solution, we write much better code.

Many programmers often find themselves very pleased with a program they have written; it seems "good" to them, although they don't know exactly why. It's usually because they have solved more than the specific problem – they have solved a whole class of problems. ©

INSIGHT: ATARI

Bill Wilkinson
Optimized Systems Software

Good news! I have finally found out how and where you will be able to obtain copies of *De Re Atari* ... and it won't even cost you your left thumb. The Atari Program Exchange now has it available for $19.95 plus shipping. The part number for it is APX-90008, and you can order it through 800-538-1862 (800-672-1850 in California). There are several changes and improvements from earlier versions, including a section on the GTIA. One disappointment is that an appendix on random access files has been deleted. Oh well, leaves room for me to do a future article.

The How and Why articles on Atari BASIC that appeared in the last two issues were the result of requests for ways of "hooking into" BASIC, in order to add commands, etc. I am trying to gently break the news that you *can't* add commands to a RUNning program (though direct, keyboard commands can be done by intercepting keyboard input, as I presume the Eastern House "Monkey Wrench" does.). But I have been trying to lead up to *why* you can't add commands, so that people won't waste time on false leads in trying to prove me wrong.

However, I am suspending the How and Why series this month in order to take a look at the USR function. It is my belief that the USR function will give most of you access to all the added comands you could write, which lessens somewhat the impact of not being able to integrate your own commands. In addition to some suggestions on usage, this month we implement a really powerful USR function: one which will play a song (or most any kind of sound) in the background while your BASIC program continues to chug away (zapping Klingons, etc.). Naturally, there will also be the usual mix of tricks, etc.

In order to deliver on my promise to the BASIC users regarding the song-playing USR function, I must first lead the assembly language fanatics through a short intro to the Atari's interrupt system. As far as I know, the Atari is the only low-end personal computer that gives you such complete access to a fully-integrated, usable interrupt system. The Atari OS is structured to take advantage of several of these interrupts; and, more importantly, the user is invited to gain full or partial control of most interrupt routines. This despite the fact that Atari's interrupt service routines are in ROM.

The 6502 microprocessor supports two types of interrupts: NMI (Non-Maskable Interrupt) and IRQ (Interrupt ReQuest). A bit in the CPU status byte controls whether IRQ's will generate interrupts, but if an NMI signal is presented to it the 6502 will always call in interrupt service routine. Atari, however, allows the user to prevent NMI's from reaching the CPU (except for the RESET button), thus giving even greater control. Once again, I must refer you to the Atari Technical Manual for full details, but herewith is a summary of the available interrupts.

Table 1. Available Interrupts

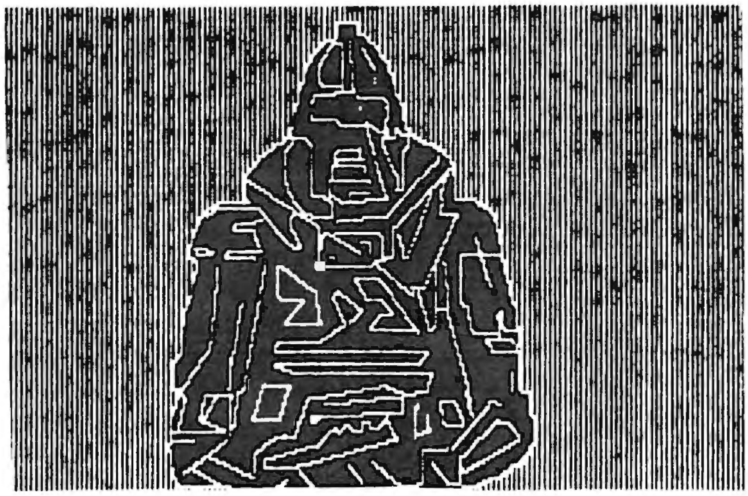
Type	Description
NMI	Reset Button (the only uncontrollable interrupt)
NMI	Display List Interrupt
NMI	Vertical Blank Interrupt (60 times per second)
IRQ	BREAK key
IRQ	any other key
IRQ	Serial Input (for SIO communication with disk, etc.)
IRQ	Serial Output (ditto)
IRQ	Serial Transmission Completed (ditto)
IRQ	Timer #4
IRQ	Timer #2
IRQ	Timer #1
IRQ	6520 parallel port "A"
IRQ	6520 parallel port "B"
IRQ	BRK instruction encountered (internal to 6502)

Each of the available interrupts, except the Reset Button and the BREAK key (and Timer #4 on all except newest machines), has a vector (two byte pointer) through RAM. To take control of an interrupt, simply put the address of your routine in the vector, and OS will call you instead of the default routine. The only exception is the Vertical Blank Interrupt, which is handled slightly differently and is the real subject of this article.

The Vertical Blank Interrupt (VBI) is really the key to many of Atari's unique features. It occurs 60 times per second, at the bottom of each scan of the TV screen, and is used by the OS ROMs to do all sorts of things. First, and perhaps most obvious, it drives the three-byte clock at locations $12,$13,$14 (18,19,20 decimal) as well as several other usable event timers (e.g., serial bus timeout), most of which are accessible to the user. Second, and most useful, it allows changes to the graphics-related hardware at a time when nothing is being displayed on the screen: it moves all the "shadow" locations (see the technical manual) to their corresponding hardware ports.

Of necessity, then, the user would not normally want to interfere with the operations of the VBI routines. But, once again, the Atari software design team thought ahead: they provided not one, but two, VBI vectors. Thus; upon receipt of a VBI request, the ROM code first calls the routine pointed

FOR ATARI



FOR ATARI

The above is a graphics 8 screen printout on the EPSON, with our new AESD II (tm)

★ ★

001 ATARI EPSON SCREEN DUMP
This is a screen dump program that transfers a screen image to the MX-80. . . . . (D) $19.95

002 ATARI EPSON SCREEN DUMP II
This is a screen dump program which allows you to copy anything from the screen. It also supports all graphics modes and text modes. It supports all the features of the EPSON(tm) MX-80 and MX-100. The program is in machine language and is relocatable. . . . . (C) $26.95
(D) $29.95

003 ATAR-RENUM
This is a utility that will renumber any tokenized BASIC program that is co-resident in RAM. . . . . (C) $19.95
(D) $21.95

004 INFO-FILE
This program is designed to act as an electronic filing cabinet. This is a FAST database program that utilizes a dynamic keyboard to move the user quickly through this menu driven program. Use it to create, add, delete, edit, print, selectively search, and store your custom files. (D) $21.95

005 BINARY LOAD CASSETTE TO DISK
This utility will take binary load cassette files like SPACE INVADERS (tm) and allow their transfer to disk. No more waiting for loading! The duplicate is AUTO-BOOTING and uncopyable. . . . . (D) $21.95

006 DEVIL DWELL DUNGEON
This disk based adventure has excellent graphics. Prepare yourself for hours of unpredictable entertainment as you venture into the depths of the Devil Dwell Dungeon in search of the Golden Septor and the rating of Superlord of Superlords. This is an AUTO-BOOTING program. (D) $21.95

007 DOWNLOADER
This is a true SMART TERMINAL EMULATOR PROGRAM which allows you to upload and download files between computers and save to DISK, CASSETTE, or a PRINTER. ALSO WORKS WITH THE D.C. HAY'S SMARTMODEM.
(C) $26.95
(D) $29.95

009 ELECTRONIC CALCULATOR
This program is a tool for the electronics hobbyist. It makes the necessary resistive and capacitive calculations for both series and parallel circuits. It shows formula, decodes resistors, plus power calculations for both AC and DC circuits. . . . . (C) $19.95
(D) $21.95

011 ELBBARCS
This is a word game program which is in high resolution graphics. . . . . (C) $19.95
(D) $21.95

012 UTILITY PAK 1
These four utility's are for the serious programmer. XREF is a variable cross reference utility which tells you where and when a variable is used in a program. VARIABLE-CHANGER is a program that allows you to easily change the name of any or all of the variables in your program. Lister and Denumber are also included. . . . . (C) $19.95
(D) $21.95

013 PIE BAR UTILITY
This utility is designed to provide a screen dump capability for the ATARI® GRAPH IT(tm) using EPSON® MX-80 printer. Features STORAGE and RECALL of both Pie and Bar Charts. Runs in 32K of RAM Screen Dump feature can be used separately in 24K of RAM. . . . . (C) $19.95
(D) $21.95

014 BACKUP MASTER
A machine language program that allows you to make backup copies of boot load diskettes. Also displays any sectors that the disk drive had trouble reading and skips over them. . . . . (D) $39.95

COMPUTER AGE SOFTWARE

9433 GEORGIA AVE.
SILVER SPRINGS, MD 20910
(301) 588-6565

CASH, CERTIFIED CHECK,
MASTERCARD & VISA ACCEPTED
FOR IMMEDIATE SHIPMENT.

PERSONAL CHECKS
ALLOW 10 DAYS TO CLEAR.

to by vector VVBLKI (at $0222) and then calls via the vector VVBLKD (at $0224). The 'I' and 'D' stand for "Immediate" and "Deferred," respectively.

Normally, the user routine would not replace the vector at VVBLKI. Thus the Atari ROM code can update its clocks and move its "shadow" registers in confidence that it will finish its job before the screen starts displaying the next TV frame. The user may replace VVBLKD to cause his routine to execute directly after the Atari system code.

Some cautions are in order: (1) Disaster will strike if your VBI routine is not done before the next VBI occurs. If you simply need to synchronize your routine to a vertical blank, just wait for the system clock to tick before starting (see the label WAITVB in this month's example program). (2) As with most Atari vectors, the safest way to use these is to move them somewhere in your own data area, replace them with your pointer, and have your code finish up by jumping back via the original Atari routine. This is particularly important to do with interrupt handlers, else the interrupt system may not be properly reset.

Finally, let me note that you may, if you really have to, steal the entire VBI processing for yourself. This is not necessarily bad (especially if you are writing a dedicated game, etc.), but be forewarned that you will have to worry about shadow registers, etc., yourself. There is a lot more to this subject, including what Atari refers to as time-critical I/O, but for most purposes you should be able to work within the rules I have outlined.

A Real, Live Example

The example program this month is designed to be used via USR from BASIC, but there is a simplified entry point from assembly language. You could lift this program as is and plunk it into any assembled game, etc. The idea behind the program is simple: a routine is passed a sequence of bytes which are interpreted to be commands to the sound generators of the Atari hardware. The routine examines the bytes and performs the requests. One of the available requests is to "play" sound(s) for a specified length of time; upon encountering this request, the routine waits the appropriate time before processing the next byte. Simple.

Except that this routine will operate (invisible to a running BASIC program) merrily playing

Main Assembly Listing

```
0000              .PAGE "    equates, origins, etc."
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ;
            ; PLAYIT  -- a demonstration of performing
            ;            clocked, interrupt-driven
            ;            tasks under Atari OS.
            ;
            ; Written by Bill Wilkinson
            ;   for March, 1982, COMPUTE!
            ;
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ;
0600        ORIGIN =     $0600
0000              *=     ORIGIN
            ;
00FF        LOW    =     $FF
0100        HIGH   =     $100
            ;
D200        AUDF1  =     $D200      ; Frequency, audio channel 1 (sound
            0)
D201        AUDC1  =     $D201      ; Channel 1 control & volume
            ;
0224        VVBLKD =     $0224      ; Delayed Vertical Blank routine
            ;
0014        CLOCKLSB =   $14        ; the system clock, LSB of 3
            ;
00CE        PLAYADDR =   $00CE      ; 2 byte pointer in safe place
            ;
            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            ;
            ; Equates for our private sound commands
            ;
00FF        CMDR   =     255        ; Repeat
00FE        CMDS   =     254        ; Stop sound (keep routine going)
00FD        CMDN   =     253        ; Number of voices
00FC        CMDTV  =     252        ; set Tone and Volume
0000        CMDE   =     0          ; End (but sound not turned off)
            ;

0600              .PAGE "    install our PLAYIT routine "
            ;
            ; INSTALL is the entry point called from BASIC
            ;
            ; The BASIC program calls us via
            ;      USR( INSTALL, ADR(playit-command-string) )
            ;
            ; The routine may be called from
            ;      assembly language at INSTALL1
            ;      by placing the address of the
            ;      command string in A,Y (LSB,MSB)
            ;
            INSTALL
0600 68           PLA               ; BASIC tells us how many parameters
0601 C901         CMP   #1          ; better just have one!
0603 D0FE   GOOF  BNE   GOOF        ; else only RESET will get him out!
0605 68           PLA
0606 A8           TAY               ; MSB to Y register
0607 68           PLA               ; LSB to A register
            ;
0608        INSTALL1 =  *           ; assembly language entry point
            ;
            ; first, we wait for a vertical blank
            ; ...to ensure we don't get a VBLANK
            ;    interrupt while we are working!
            ;
0608 A614         LDX   CLOCKLSB
            WAITVB
060A E414         CPX   CLOCKLSB    ; has clock ticked?
060C F0FC         BEQ   WAITVB      ; no...keep waiting
            ;
            ; OKAY TO PROCEED
            ;
060E 85CE         STA   PLAYADDR    ; we preempted a zero page spot
```

along while BASIC continues what it is doing. To accomplish this, we have hooked into VVBLKD (as described above). The user specifies the note duration as a number of "jiffies" (60ths of a second), and we let the VBI count down the duration for us.

The commands are imbedded in a string of bytes passed to the routine. Playit recognizes six command types, as shown in Table 2. Playit is not particularly sophisticated. For example, all voices must play sounds for the same duration and, when chang-

Table 2. Playit Command Codes

Byte value	Name	Description
255 ($FF)	CMDR	Repeat the entire sound command string
254 ($FE)	CMDS	Stop all sounds (do *not* end command string)
253 ($FD)	CMDN	Number of voices is specified in next byte (0-4)
252 ($FC)	CMDTV	Specify Tone and Volume (as in SOUND 0,freq, TONE,VOLUME). Must be followed by 0-4 bytes (one per each voice as specified by CMDN), each of which specifies a Tone/ Volume for one channel.
0 ($00)	CMDE	End command, unhook from VVBLKD. Does not turn off sound, so is usually preceded by CMDS.
any other	---	Any other value is assumed to be a duration, given in 'jiffies' (60ths of a second). Must be followed by 0-4 bytes (one per voice as specified by CMDN), each of which specifies the frequency of the sound for one channel (as in SOUND 0,FREQ, tone, volume).

ing volume or tone quality, all voices must be respecified. A more sophisticated sound interpreter would presumably mean smaller command strings but a bigger interpreter. If you go to the trouble to type in both Playit and Playit From BASIC, you will see that some more than acceptable sounds can be accomodated, so I am reasonably happy with the results.

Some interesting projects remain: Why not convert Atari's Music Composer disk files to Playit-compatible strings? Or how about a real Music Compiler written in BASIC? How about making Playit relocatable, a la last month's article? Please write and tell of your successes (or failures?).

Last but not least, another caution: since I/O to anything but the screen or keyboard uses the SIO serial bus driver, and since the serial bus uses the sound generators to get its baud rates, etc., you MUST turn off sound generation (commands CMDS, CMDE) before doing such I/O.

Atari BASIC: On Sounds, Hex Numbers, And The USR Function

The featured idea and program in this issue is the Playit From BASIC listing which follows. The program itself is not very sophisticated: it simply allows the one-character command codes (R,S, N,T,E) and hex data bytes to be translated into characters in a string. It then passes the address of the string to Playit (the assembly language program) and comes back to the user, ready to compile the next string of commands. If you intend to emulate this scheme, rather than use the program as is, you might be advised to put the sound command string into memory you have reserved (e.g., via the "Simplest Method" given in previous articles in this series). Putting the command in a string is inviting trouble: if your program stops, if you ENTER new

```
0610 8DC206 1710         STA  REPEAT     ; just in case of a repeat cmd
0613 84CF   1720         STY  PLAYADDR+1
0615 8CC306 1730         STY  REPEAT+1   ; similarly for MSB
            1740 ;
0618 AD2402 1750         LDA  VVBLKD     ; prepare to save the ptr
061B AC2502 1760         LDY  VVBLKD+1
061E C93C   1770         CMP  #PLAYIT&LOW ; already saved?
0620 D004   1780         BNE  NOWINSTALL ; no
0622 C006   1790         CPY  #PLAYIT/HIGH
0624 F010   1800         BEQ  INSTALLED ; yes
            1810 ;
            1820 NOWINSTALL
0626 8DC406 1830         STA  SAVEVBLK
0629 8CC506 1840         STY  SAVEVBLK+1 ; save system vector
            1850 ;
062C A93C   1860         LDA  #PLAYIT&LOW
062E 8D2402 1870         STA  VVBLKD     ; and install our own

0631 A906   1880         LDA  #PLAYIT/HIGH
0633 8D2502 1890         STA  VVBLKD+1
            1900 INSTALLED
0636 A901   1910         LDA  #1         ; A single clock tick
0638 8DC706 1920         STA  DURATION   ; until we start playing
063B 60     1930         RTS             ; done with install!
            1940 ;

063C        1950         .PAGE "     The actual PLAYIT routine"
            1960 ;
            1970 ; PLAYIT is the entry point for our Delayed
            1980 ;   Vertical Blank routine
            1990 ;
            2000 ; PLAYIT 'reads' the sound command string
            2010 ;    and plays our 'song'
            2020 ;
            2030 ; SAM is simply the looping point for cmds
            2040 ;
            2050 PLAYIT
063C CEC706 2060         DEC  DURATION   ; keep on playing?
063F D029   2070         BNE  EXIT       ; yep...no changes
            2080 ;
            2090 SAM
0641 20B706 2100         JSR  GETCMD     ; get a byte from command string
0644 C900   2110         CMP  #CMDE      ; End it now?
0646 F053   2120         BEQ  DOEND      ; yes
0648 C9FF   2130         CMP  #CMDR      ; D.C. al Fine?
064A F05E   2140         BEQ  DORPT      ; yes
064C C9FE   2150         CMP  #CMDS      ; Stop all sound ?
064E F03F   2160         BEQ  DOSTOP     ; yep
0650 C9FC   2170         CMP  #CMDTV     ; Tone and Volume on TV?
0652 F02A   2180         BEQ  DOTV       ; yeah
0654 C9FD   2190         CMP  #CMDN      ; Number of voices change?
0656 F015   2200         BEQ  DONUM      ; uh-huh
            2210 ;
            2220 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            2230 ;
            2240 ; if none of the above, must be duration
            2250 ;
            2260 DODURATION
0658 8DC706 2270         STA  DURATION   ; we assume so
065B AEC606 2280         LDX  NUMVCS
065E 300A   2290         BMI  EXIT       ; no voices, just duration
            2300 FREQLP
0660 20B706 2310         JSR  GETCMD     ; yes...get next byte
0663 9D00D2 2320         STA  AUDF1,X    ; and set the frequency
0666 CA     2330         DEX
0667 CA     2340         DEX             ; see if more voices
0668 10F6   2350         BPL  FREQLP     ; yes...keep trying
            2360 ; no...fall through to EXIT
            2370 ;
            2380 EXIT
066A 6CC406 2390         JMP  (SAVEVBLK) ; let OS clean things up
            2400 ;
            2410 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            2420 ;
            2430 ; set number of voices
            2440 ;
            2450 DONUM

066D 20B706 2460         JSR  GETCMD     ; next byte...
0670 AA     2470         TAX
0671 CA     2480         DEX
0672 8A     2490         TXA             ; less one
0673 3003   2500         BMI  NUMOK      ; if < zero, leave it alone
0675 2903   2510         AND  #$03       ; Ensure 1-4 voices
0677 0A     2520         ASL  A          ; doubled, for ease of use
            2530 NUMOK
0678 8DC606 2540         STA  NUMVCS     ; as number of voices
```

lines, if you DIMension more variables, etc., the string may move and Playit would start playing random sounds.

The commands have simply been entered into the program via DATA statements starting at line 9000. Those of you who go to the trouble to enter all this will, I hope, be pleasantly surprised by the sounds generated by lines 9400-9418. You will probably be dismayed, however, at the idea of putting in such a complex sound yourself. That is why I encourage someone to come up with a better "Music Compiler" along these same lines.

In any case, I invite you to compose your own music or sounds to be put into this system. Generally, I wrote a sound in BASIC to test it before committing it to DATA statements. For example, the "CHOO-CHOO" sound evolved from this BASIC line:

```
FOR V=15 TO 0 STEP -1 : SOUND
O,V,O,V : NEXT V
```

The above sounds like an explosion, but if you slow it down a little and repeat it regularly you can train it as you wish. On to the short subjects.

HexDec

If you have already peeked at the listing of Playit From BASIC, you may have noted an unusual looking hexadecimal to decimal conversion routine. In fact, I herewith present you with a "one-liner" HexDec program:

```
1 DIMH$(23),N$(9):H$=",ABCDEF
GHI!!!!!!! JKLMNO":IN.N$:F.I=
1TOLEN(N$):N=N*16+ASC(H$(ASC
(N$(I))-47))
:N.I:?N:RUN
```

The underlined characters are control characters (control-comma is the heart, etc.). The abbreviations are necessary to get it to fit on one line. To see how it works, figure out what happens when you input "9A". Recall that ASC("9") is 57 and ASC("A") is

65. 57-47 is 10 and 65-47 is 18. Look at the 10th and 18th characters in H$. What is ASC("control-I")? ASC("control-J")?

You can avoid the control characters by adding the -64 shown in Playit From BASIC. Simple.

DecHex

This isn't really pertinent, but while we are on the subject of one-liners:

1DIMH$(16):H$="0123456789AB CDEF":IN.N:M=4096:F.I=1TO4:J= INT(N/M):?H$(J+1);:N=N-M*J:M= M/16:N.N:?:RUN

The USR And ADR Functions

Even though the methods of using the USR function are fairly thoroughly covered in the *Atari BASIC Reference Manual,* I find that many users are not fully aware of the real power of this function. Recall that the general syntax of this function is:

USR(addr [,expr [,expr ...]])

In other words, in addition to giving BASIC an address to call, you may pass *any number* of expressions to the assembly language routine. BASIC converts each expression to a 16-bit integer, pushes the result on the CPU stack, and cleans up by pushing on a single byte which tells the number of such expressions it pushed. (The address, which may itself be an expression, is *not* pushed and is not counted by that single byte.)

So what can we pass to assembly language? Obviously, numbers in the range of 0 to 65535. But what about characters? Conceive of

USR(addr, ASC("T"), expr),

where the "T" might be used as a mnemonic command to tell the routine which of several functions is desired. How about strings of characters? Recall that the three essential ingredients defining a

```
067B 4C4106          JMP  SAM
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ; set tone and volume
                ;
                DOTV
067E AEC606          LDX  NUMVCS
0681 30BE            BMI  SAM         ; no voices to set
                TVLP
0683 208706          JSR  GETCMD      ; get next byte
0686 9D01D2          STA  AUDC1,X     ; treat as t&v command
0689 CA              DEX
068A CA              DEX              ; more voices?
068B 10F6            BPL  TVLP        ; yes
068D 30B2            BMI  SAM         ; no
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ;  STOP the sound (by clring all sound regs)
                ;
                DOSTOP
068F A207            LDX  #7
0691 A900            LDA  #0
                STOPLP
0693 9D00D2          STA  AUDF1,X    ;freq and vol to zero
0696 CA              DEX
0697 10FA            BPL  STOPLP
0699 30A6            BMI  SAM         ; sound stops, pgm keeps going
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;
                ; END the processing (but doesn't stop sound)
                ;
                DOEND
069B ADC406          LDA  SAVEVBLK
069E 8D2402          STA  VVBLKD     ; restore system ptr
06A1 ADC506          LDA  SAVEVBLK+1
06A4 8D2502          STA  VVBLKD+1   ; and, to OS, we aren't here
06A7 6CC406          JMP  (SAVEVBLK) ; one last time
                ;
                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;

                ; repeat the same stuff again
                ;
                DORPT
06AA ADC206          LDA  REPEAT
06AD 85CE            STA  PLAYADDR
06AF ADC306          LDA  REPEAT+1
06B2 85CF            STA  PLAYADDR+1 ; just reset the address
06B4 4C4106          JMP  SAM        ; and try it again

06B7                 .PAGE "     the GETCMD subroutine"
                ;
                ; simply gets next byte from
                ; command string
                ;
                GETCMD
06B7 A000            LDY  #0
06B9 B1CE            LDA  (PLAYADDR),Y ; get the byte
06BB E6CE            INC  PLAYADDR   ; bump LSB of pointer
06BD D002            BNE  GCEXIT     ; done
06BF E6CF            INC  PLAYADDR+1 ; and the MSB
                GCEXIT
06C1 60              RTS
                ;

06C2                 .PAGE "     ram usage"
                ;
06C2 0000       REPEAT .WORD 0        ; in case we hear it again
06C4 0000       SAVEVBLK .WORD 0      ; so we can jmp indirect
06C6 00         NUMVCS .BYTE 0        ; controls TVLP and FREQLP
06C7 00         DURATION .BYTE 0      ; how long we hold a sound
                ;
                ;
06C8                 .END

=0600 ORIGIN       =00FF LOW          =0100 HIGH         =D200 AUDF1
=D201 AUDC1        =0224 VVBLKD       =0014 CLOCKLSB     =00CE PLAYADDR
=00FF CMDR         =00FE CMDS         =00FD CMDN         =00FC CMDTV
=0000 CMDE          0600 INSTALL       0603 GOOF         =0608 INSTALL1
 060A WAITVB        06C2 REPEAT        063C PLAYIT        0626 NOWINSTALL
 0636 INSTALLED     06C4 SAVEVBLK      06C7 DURATION      066A EXIT
 0641 SAM           06B7 GETCMD        069B DOEND         06AA DORPT
 068F DOSTOP        067E DOTV          066D DONUM         0658 DODURATION
 06C6 NUMVCS        0660 FREQLP        0678 NUMOK         0683 TVLP
 0693 STOPLP        06C1 GCEXIT
```

string in Atari BASIC are its DIMension, LENgth, and address. Since your program presumably DIMensioned the string, you know that value and may pass it as an expression. And the address and length are available from the ADR and LEN functions!

Would you like your assembly language routine to modify your string, affecting its length? Try something like this:

```
DIM XX$( XXDIM )
XX$( USR( addr, ADR(XX$), XXDIM ) + 1 ) = ""
```

Recall that the USR function may return any 16-bit value to the BASIC program, which is automatically converted to floating point as needed. Assume that this USR routine puts something in the XX$ string and returns the number of characters it put in. The above will then set the LENgth of XX$ properly for use by other BASIC statements and functions.

Finally, there is floating point. How about writing a matrix inversion program? If we are limited to passing 16-bit integers, how do we pass a floating point number via USR? Simple: we pass the address of the number, just as we do with a string. And how do we get the address of a number, when the ADR function only works with strings? Like this:

```
DIM FF$(1),FF( dim1, dim2 )
JUNK = USR( addr, ADR(FF$) + 1, dim1, dim2 )
```

A little published fact about Atari BASIC is that DIMensioning of both strings and arrays proceeds in an orderly fashion according to the DIM statements encountered. And you are guaranteed that the order you DIM strings and arrays is the order they will occur in memory! So, by DIMensioning that one-byte string, FF$, directly before the DIMension of the array, FF( ), we *know* that the address of the array is one greater than the address of the string. Thus we can pass all the pertinent information about the array (its address and dimensions) to our assembly language routine. Incidentally, if you don't want to waste a one-byte string for this purpose, there is no reason FF$ can't be any DIMension you need: just adjust the '+ 1' to reflect the actual DIM you use.

One last note on this subject: the fact that you can predict the memory order of strings and arrays has fascinating possibilities in regards to record structures, etc. But (and how many times have you read this from me) that's a topic for another article.

Program 1.

```
10 AUDCTL=53768:DBL=120
20 AUDF1=53760:AUDC1=53761
30 SOUND 1,10,10,15:SOUND 3,10,10,15
40 POKE AUDC1,0:POKE AUDC1+4,0
50 POKE AUDCTL,DBL
60 FOR J=10 TO 15:POKE AUDF1+2,J:POKE AUDF1
   +6,20-J
```

```
70 FOR I=0 TO 255:POKE AUDF1,I:POKE AUDF1+4
   ,255-I:NEXT I
80 NEXT J
```

...VERY SMOOTH GLIDES...

Program 2.

```
10 AUDCTL=53768:DBL=120
12 OSC=1789790/2
20 AUDF1=53760:AUDC1=53761
30 SOUND 1,10,10,0
40 POKE AUDC1,0:POKE AUDC1+4,0
50 POKE AUDCTL,DBL
60 P2=2^(1/12)
70 NTE=16:REM C IN THE REAL BASS
80 FOR I=1 TO 109
90 FREQ=INT(OSC/NTE-7+0.5):F0=INT(FREQ/256)
92 F1=FREQ-256*F0
100 POKE AUDF1,F1:POKE AUDF1+2,F0
102 POKE AUDC1+2,175
103 PRINT "NOW PLAYING ";INT(NTE+0.5);" HZ"
105 FOR J=1 TO 100:NEXT J
110 NTE=NTE*P2
120 NEXT I
130 GOTO 70
```

...9 OCTAVE CHROMATIC SCALE...

Playit From BASIC

```
1000 REM ******************************
1020 REM *
1040 REM * PLAYIT FROM BASIC, SAM
1060 REM *
1080 REM * This routine is a simple
1100 REM * sound "compiler", which
1120 REM * takes DATA statements and
1140 REM * converts them into command
1160 REM * strings suitable for use by
1180 REM * the interrupt-driven PLAYIT
1200 REM * routine.
1220 REM *
1240 REM *
1260 REM * Written by Bill Wilkinson
1280 REM *
1300 REM *   for March, 1982, COMPUTE!
1320 REM *
1340 REM ******************************
1360 REM
1380 REM First, constants, routine addresses, etc.
1400 REM
1420 DIM HX$(2),CMD$(11),PLAY$(1000),HEX$(23),TYPE$(1),
     PLAYIT$(1000)
1440 HEX$="@ABCDEFGHI!!!!!!!JKLMNO"
1460 DOCMD=2300:LOOP=1800:HEXDEC=2600
1480 AGAIN=1700:EXITLOOP=2100
1500 PLAYIT=6*256:REM or wherever you put the routine
1520 REM
1530 SOUND 0,0,0,0:REM needed to initialize properly
1540 REM The command equates...
1560 REM notice that these match the
1580 REM assembly language routine
1600 CMDR=255:CMDS=254:CMDN=253:CMDTV=232:CMDE=0
1620 REM
1640 REM ******************************
1660 REM
1680 REM This is the AGAIN of
1700 REM PLAY IT AGAIN, ATARI
1720 REM
1730 PRINT "    <processing...please wait>"
1740 PLAY$="":PLAY=0
1760 REM
1780 REM This is LOOP
1800 PLAY=PLAY+1:REM to next cmd byte
1820 READ CMD$:REM a bunch of commands
1840 REM
1860 TYPE$=CMD$:REM use the command character
1880 IF TYPE$="R" THEN PLAY$(PLAY)=CHR$(CMDR):GOTO EXIT
     LOOP
1900 IF TYPE$="S" THEN PLAY$(PLAY)=CHR$(CMDS):GOTO LOOP
1920 IF TYPE$="N" THEN NUMVCS=1:CMD=CMDN:GOSUB DOCMD:NU
     MVCS=DEC:GOTO LOOP
1940 IF TYPE$="T" THEN CMD=CMDTV:GOSUB DOCMD:GOTO LOOP
1960 IF TYPE$="E" THEN PLAY$(PLAY)=CHR$(CMDE):GOTO EXIT
     LOOP
1980 REM *** IF TO HERE, ASSUME DURATION & FREQ ***
2000 HX$=CMD$:GOSUB HEXDEC:CMD=DEC:REM command is
     duration
2020 CMD$=CMD$(2):REM to fool DOCMD
2040 GOSUB DOCMD:GOTO LOOP
2060 REM
2080 REM exitloop
2100 REM
2120 REM do the sound playing
2140 REM
2150 PLAYIT$=PLAY$:REM else we alter what we are playing
2160 JUNK=USR(PLAYIT,ADR(PLAYIT$))
2180 REM
2200 PRINT "HIT RETURN FOR NEXT SOUND ";:INPUT TYPE$
2220 GOTO AGAIN
2240 REM
2260 REM
2280 REM ***************************
2300 REM THE SUBROUTINES
2320 REM
2340 REM first, DOCMD
2360 REM
2380 PLAY$(PLAY)=CHR$(CMD):REM The command byte
2400 IF NUMVCS=0 THEN RETURN
2420 REM we process NUMVCS bytes
2440 FOR I=2 TO NUMVCS+NUMVCS STEP 2
2460 HX$=CMD$(I):GOSUB HEXDEC:REM convert the byte
2480 PLAY=PLAY+1:PLAY$(PLAY)=CHR$(DEC):REM and stuff it
     away
2500 NEXT I
2520 RETURN
2540 REM
2560 REM ............
2580 REM ***************************
2600 REM and now HEXDEC
2620 REM
2640 DEC=0:REM our accumulator
2660 FOR L=1 TO LEN(HX$)
2680 DEC=DEC*16+ASC(HEX$(ASC(HX$(L))-47))-64
2700 NEXT L
2720 RETURN
8999 REM ...a siren-like sound...
9000 DATA N01,TCF,1408,1412,R
9099 REM ...a fanfare of sorts...
9100 DATA S,N01,TA2,30F3
9102 DATA N02,TA3A3,30F3C1
9104 DATA N03,TA4A4A4,30F3C1A1
9106 DATA N04,TA5A5A5A5,60F3C1A17A
9108 DATA T00000000
9110 DATA N00,C0,R
9199 REM ...beeping off the seconds...
9200 DATA S,N01
9202 DATA TAE,0130
9204 DATA TAC,0130
9206 DATA TAA,0130
9208 DATA TA8,0130
9210 DATA TA6,0130
9212 DATA TA4,0130
9214 DATA TA2,0130
9216 DATA T00,3500
9218 DATA R
9299 REM ...choo-choo ??? ...
9300 DATA S,N01
9302 DATA T0E,010E
9304 DATA T0C,010C
9306 DATA T0A,010A
9308 DATA T08,0108
9310 DATA T06,0106
9312 DATA T04,0104
9314 DATA T02,0102
9316 DATA T00,0300
9318 DATA R
9400 DATA S,N01,TAC
9402 DATA 3051,305B,3044,183C,182D,3035
9404 DATA 183C,182D,3035,3044,303C,3051,305B
9406 DATA N04,TACA4A4A8
9408 DATA 30516C89A2,305B7990B6,30446C89A2
9410 DATA 183C4879B6,182D4879B6,3035485BD7
9412 DATA 183C4879B6,182D5BB6B6,303544SB89
9414 DATA 3044516CA2,38325179F3
9416 DATA 423C485BB6,50445B6C89
9418 DATA S,N00,F0,R
9898 REM ...stop and end...to quit...
9999 DATA S,E
```

©

PART I

Disk Checkout For 2040, 4040, And 8050 Disks

Jim Butterfield
Toronto, Canada

Editor's Note: In Part I of this article Jim explains disk manipulations via machine language. Next month, in Part II, he concludes with a machine language disk routine and a program that can analyze the condition of files and blocks on the disk. — RTM

The disk doesn't know or care who's giving it instructions: BASIC or Machine Language. All that's needed is to send or receive the same information as BASIC uses.

For all input and output, I recommend opening the necessary channels from BASIC. It's easier and works the same in all systems. Machine language may then take over and use the previously opened files as it wishes, connecting and disconnecting at will.

You'll often want to check the status byte ST. It's located at hexadecimal 96 in PET's memory. It's especially important for checking end-of-file on sequential records and end-of-record on relative records. You can also detect IEEE problems here, especially timeouts.

Let's take a simple example. We might want to do a Block Read of a given track and sector from disk and then dump part of the contents to the screen. To make our example easy, we'll display only bytes one through eight. Byte zero is sometimes hard to get on early disk systems due to a bug in the Buffer-Pointer routine; we'll sidestep that question.

The BASIC Program

We're planning to read bytes one through eight of track 18, sector 0. That might be the BAM (Block Availability Map) block, but perhaps not: these programs will also work on 8050 disks.

We must: Open the Command channel, secondary address 15; Initialize the disk, in case it's a 2040; Open a direct access channel; Cause the block read; Set the Buffer pointer; and, finally, read the channel. At the finish we should close our channels. Our BASIC program would read:

```
100 OPEN 6,8,15                    (Command Channel)
110 PRINT#6,"I0"                   (Initialize)
120 OPEN 2,8,3,"#"                 (Direct Access
                                    channel)
130 PRINT#6,"U1:";3;0;18;0         (Read Block)
140 PRINT#6,"B-P:";3;1             (Set Buffer Pointer)
150 GET#2,X$                       (Get a byte)
160 PRINT ASC(X$+CHR$(0) );(Print it)
170 C=C+1                          (Count them)
180 IF C<8 GOTO 150                (Do more?)
190 CLOSE 2:CLOSE 6                (Quit)
```

You might like to try this to see it work. If you like, change the buffer pointer (line 140), the number of values displayed (line 180) or the track and sector (line 130). Now let's try the same thing in machine language.

The BASIC Driver

It's convenient to OPEN from BASIC, so we type NEW and enter the following BASIC program which will set things up for Machine Language:

```
100 OPEN 6,8,15
110 PRINT#6,"I0"
120 OPEN 2,8,3,"#"
125 SYS 1200
190 CLOSE 2:CLOSE 6
```

Don't run this yet, since the Machine Language is not in place.

Planning The Machine Language Program

We want to send exactly the same stuff as was sent by BASIC, to the same logical channels. We know that the ML equivalent of PRINT#6... is LDX #$06, JSR $FFC9 ... JSR $FFCC. Note that we use the logical file number, 6. Similarly, we know the equivalent of GET#2 is: LDX #$02, JSR $FFC6, JSR $FFE4,... JSR $FFCC. So we can code:

```
LDX   #$06
JSR   $FFC9    (Open channel 6)
LDA   #$55     (Letter U)
JSR   $FFD2    (.. print it)
LDA   #$31     (Digit 1)
JSR   $FFD2    (.. print it)
LDA   #$3A     (Colon)
JSR   $FFD2
LDA   #$20     (Space)
JSR   $FFD2
LDA   #$33     (Digit 3)
JSR   $FFD2
LDA   #$20     (Space)
JSR   $FFD2
LDA   #$30     (Digit 0)
JSR   $FFD2
LDA   #$20     (Space)
```

```
JSR     $FFD2
LDA     #$31      (Digit 1)
JSR     $FFD2
LDA     #$38      (Digit 8)
JSR     $FFD2
LDA     #$20      (Space)
JSR     $FFD2
LDA     #$30      (Digit 0)
JSR     $FFD2
LDA     #$0D      (Return)
JSR     $FFD2
JSR     $FFCC     (End transmission)
```

Note that we are sending exactly what BASIC sent from line 130. Most programmers would quickly realize that a program loop would save a good deal of memory here. In Part II of this article, we'll rewrite the code and complete it.

Organizing Data Storage

John Hudson
Los Angeles, CA

There are many storage media available to minicomputer users. Minicomputer users with a disk unit know that the disk unit enhances the storage and retrieval powers of their minicomputer. One type of file that can be created for the purposes of storage and retrieval is a text file (for storage of such things as mailing addresses, telephone numbers, receipts, etc.).

For small text files, the time involved in disk retrieval and storage is not a problem. However, when a text file becomes larger than 2,000 records, the retrieval and storage of information can become time consuming.

Large text files can be organized in one of two ways: sequentially, and randomly. In sequentially organized text files, fields are stored back to back, where the beginning character of a new field immediately follows the return character ending the previous field. Information is retrieved in a linear fashion, i.e., from the beginning to the end of the file.

Disk Can Also Be Slow

When a text file does not require much updating or ongoing revision, sequential organization of the text file is indicated. However, if a large text file is ordered sequentially, and there is need for frequent updating or revision of the file, or frequent retrieving of information from end of text file, a disk unit is not much better than a cassette unit. This accessing of information at end of file may take a couple of minutes, due to the reading and verification of each record, each time.

In this type of situation, the random method of text file organization is more effective. A random-access text file is like a collection of equally-sized records; the records may be full, or they may be empty, but the length of each record in a random text file is fixed. Thus, a record at the end of the file can be accessed at approximately the same speed as records in any other location in the file.

However, the controlling program needs to know where in the file a specific record is located. Most random files are organized by 'keying' a field within the record. For example, a mailing address text file can be organized by last names. The problem when using a random text file keyed to a specific field in the record is *collision*. Collision is when two or more records address the same location within the text file, as, for example, when two people have the same last name (B. JONES and J. JONES).

A method of reducing collision is called *hashing* the key field. The basic idea of hashing, or hash addressing, is that each stored record occurrence is placed in the text file at a location whose address may be computed as some function (the hash function) of a value which appears in the occurrence – usually the primary key value.

One of the disadvantages of hash-addressing is that the sequence of stored record occurrences within the text file will almost certainly not be the keyed field sequence. In addition, there may be gaps of arbitrary size between consecutive occurrences of records.

In fact, a text file in a hash-addressing organization is usually, though not invariably, considered to have no particular sequence.

Using Mod To Hash

The following is an example of a hash function: given that the number of unique records is 1,000; the "mod" arithmetic function can be used to assign unique address locations. The mod function divides one number by another and returns the remainder. The mod parameter used in this function should be the prime number closest to the number of the records in the text file (see Table 1 for prime numbers). For this example, the closest prime number is 997. (Note: if the key field is alphabetic, it should be converted to numeric.) The function will be (key field) MOD 997. The hash function thus minimizes collision.

There are text files, such as a monthly inventory file, that require multiple entries of the same record over a period of time. Inventory may be taken at the end of each week, and the quantity stored into a text file. This presents a different type of collision problem – same record hash to same location in text file.

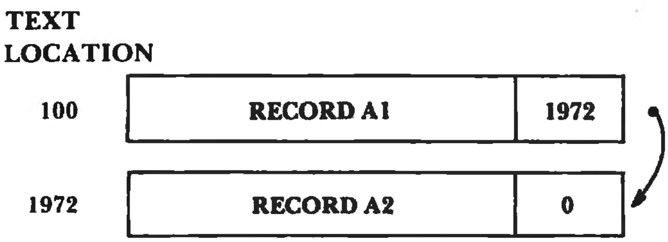
In the case where hashing records into a text file still causes collision, the controlling program needs to be able to insert the colliding record into another location and, when it goes to retrieve this record, it needs to know where it is located. A solution to this problem is to link the records in the text file. From the previous example, you have 1,000 unique records; in addition, each record is entered more than once.

A link field (LF) can be added to the end of each record to allow the linking of records. For example:

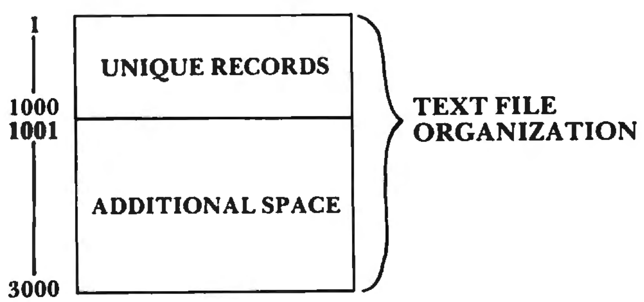
RECORD	LINK FIELD

This LF is used to point to successive entries of the same type of record, and contains the address locations of the successive record entries. The first record, A1, hashed into the text file at location 100 has '0' in the link field.

When the controlling program tries to hash another record, A2, into record location 100, it notes that there already exists a record at that location, and inserts the new record, A2, at another text address. It changes the LF of the record A1 from 0 to the next text address of record A2 (in this case, 1972), inserts 0 into the LF of record A2, and the results are as follows:



Thus, in this example, record A1 points to record A2, However, a problem arises with this type of organization: how to set up the text file? The text file can be organized with 1,000 unique hashing locations, occupying text address locations 1-1000. Any additions to a unique record can be located at text address locations 1001-3000.



This type of text file organization needs to be initialized, since the Apple system does not allow reading of a text file that does not contain records, and will produce an "END OF DATA" error message. An example of an initialization routine follows:

```
5  D$=" "
10 DLOC=66:DDTE=9999:DBS=1:DSN=2:DLP
   =333:DTRK=444:DCAST=555:DLINK=8888
11 PRINT D$; "OPEN RECORD,L29"
20 I=2001
30 PRINT D$; "WRITE RECORD,R0"
40 PRINT I: PRINT DDTE: PRINT DBS: PRINT
   DSN: PRINT DLP: PRINT DTRK: PRINT
       DCAST: PRINT DLOC:
1001 FOR J=1 TO 4200
1006 PRINT D$; "WRITE RECORD,R";J
1007 PRINT DLOC: PRINT DDTE: PRINT DBS:
     PRINT DSN: PRINT DLP: PRINT DTRK:
     PRINT DCAST: PRINT DLINK:
1009 NEXT J
1010 PRINT D$;"CLOSE RECORD"
1013 END
```

This routine initializes enough space for 4,200 records of length 29. It writes into every record a set of dummy values.

When you wish to insert a record into the main text area, the controlling program will read the text address and check a specific field for 9999, (DDTE). If it finds 9999, the controlling program can insert the record into the read text location. If it does not, then it will insert the record into the additional text area. After inserting the record, the LF of the main record is updated to point to the location of the additional record(s).

A method of keepinng track of available space in the additional text area is to store this address location and length of records into address location 0 of the text file. After each "additional text area" insertion, the available address is incremented. At the start, the controlling program will read this information, update it as needed, and, upon completion of the program, will rewrite the record 0 with the new address location.

The following is an example of a program using the link organization of a text field:

Line 70 reads text location 0 to determine the next available additional space, which is indicated by the variable "FREESPACE."

Lines 120 through 140 determine the location where the new record will be inserted. Note that this is not a hashing function.

Line 190 checks to see if the text field location DDTE has the dummy value of 9999, or if it is filled.

Lines 191 through 200 insert the new record into the unique text space.

Lines 212 through 214 traverse the link lists to get to the last record in the link.

Lines 220 through 260 update the last record in the link, and insert the new record into the additional text space area.

Lines 280 through 290 update record 0 when the program is completed.

2	3			
5	7	11	13	17
19	23	29	31	37
41	43	47	53	59
61	67	71	73	79
83	89	97	101	103
107	109	113	127	131

137	139	149	151	157
163	167	173	179	181
191	193	197	199	211
223	227	229	233	239
241	251	257	263	269
271	277	281	283	293
307	311	313	317	331
337	347	349	353	359
367	373	379	383	389
397	401	409	419	421
431	433	439	443	449
457	461	463	467	479
487	491	499	503	509
521	523	541	547	557
563	569	571	577	587
593	599	601	607	613
617	619	631	641	643
647	653	659	661	673
677	683	691	701	709
719	727	733	739	743
751	757	761	769	773
787	797	809	811	821
823	827	829	839	853
857	859	863	877	881
883	887	907	911	919
929	937	941	947	953
967	971	977	983	991
997	1009	1013	1019	1021

```
10 INPUT "PLEASE ENTER STORE
   NUMBER ", SN
11 CALL - 936: FOR X = 1 TO 9
   : CALL - 922: NEXT X
15 PRINT:PRINT"      I N S E R
    T     D I S K    ";SN
16 FOR X=1 TO 3000: NEXT X
17 CALL -936
20 INPUT "PLEASE ENTER DATE .
   . MMDD .. ",DTE
30 INPUT "PLEASE ENTER PURCHA
   SE OR SELL 1 = PURCHASE ..
   2 = ~ SELLS ",BS
40 D$=""
50 PRINT D$;"OPEN RECORD,L29"
60 PRINT D$;"READ RECORD,R0"
70 INPUT FREESPACE: INPUT DDT
   E: IN PUT DBS: INPUT DSN:
   INPUT ~ DLP: INPUT DTRK
71 INPUT DCAST: INPUT DLINK
80 IF FREESPACE>=5000 THEN GO
   TO 320
90 PRINT D$;"CLOSE RECORD"
100 INPUT "PLEASE ENTER RECOR
    D CODE,LPS,TRK8S,CASETTES
    ",LOC, LP,TRK,CAST
110 IF LOC=9999 THEN GOTO 270
120 LOCA=LOC/100
130 LOCA=LOC-LOCA*100
140 LOC=LOC/100:LINK=0
150 PRINT D$;"OPEN RECORD,L29"
160 PRINT D$;"READ RECORD,R";
    LOC
170 INPUT DLOCA: INPUT DDTE:
    INPUT ~ DBS: INPUT DSN:
    INPUT DLP: INPUT DTRK
171 INPUT DCAST: INPUT DLINK
190 IF DDTE#9999 THEN GOTO 212
191 PRINT D$;"WRITE RECORD,R";LOC
200 PRINT LOCA: PRINT DTE: PRINT BS
    : PRINT SN: PRINT LP: PRIN
    T TRK: PRINT CAST: PRINT L
    INK
210 GOTO 90
212 IF DLINK=0 THEN GOTO 220:LOC=DL
    INK
213 PRINT D$;"READ RECORD,R";DLINK
214 GOTO 170
220 PRINT D$;"WRITE RECORD,R";LOC
225 DLINK=FREESPACE
230 PRINT DLOCA: PRINT DDTE: PRINT ~
    DBS: PRINT DSN: PRINT DLP:
     PRINT DTRK: PRINT DCAST: ~
    PRINT DLINK
240 FREESPACE=FREESPACE+1
250 PRINT D$;"WRITE RECORD,R";DLINK

260 GOTO 200
270 PRINT D$;"WRITE RECORD,R0"
280 PRINT FREESPACE: PRINT DDTE: PR
    INT DBS: PRINT DSN: PRINT ~
    DLP: PRINT DTRK: PRINT DCA
    ST: PRINT DLOCA
290 PRINT D$;"CLOSE RECORD"
300 INPUT  "DO YOU WISH TO CONTINUE
     .. Y/N ",K$
310 IF K$="Y" THEN GOTO 10
320 END
```

©

Machine Language Sort Utility

Ronald and Lynn Marcuse
Freehold, NJ

There have been occasional articles in the various personal computer magazines concerning the sorting of data files. Some of these have presented sort routines coded in BASIC that can be utilized by existing programs. The complex string handling required by the sort logic is not really suitable for BASIC's rather slow execution speed. Clearly, any type of repetitive string manipulations (as performed by sorting or searching functions) would benefit from machine language code. If you continue reading you will find out how much faster it really is.

Before we get into the programs themselves, it would probably be beneficial to include some background information. The verb *sort* is defined: "to put in a certain place or rank according to kind, class or nature; to arrange according to characteristics." This comes pretty close to what we sometimes want to do with the data we store in our computers and files; put it in some kind of order. Once we have arranged it we can search it quicker (imagine a disorganized phone book), list it in a more readable format, or even match it to other files that have been sorted the same way.

The Main Questions

First we must decide where will we do the actual sorting. All of us have arranged things on a desk or table. Our sort area is, therefore, the desk or table that we used. In a computer system we have a choice of using the memory within the machine (internal) or our disk drive (external). There are problems with both of these. Computer memory is limited in size and this, in turn, will limit the number of records that can be read in. The disk drive may be able to hold more data, but the speed of the device is snail-like when compared to memory. We could use both: divide the file up into smaller chunks which can be sorted in memory, store these on disk as temporary files, and then merge all of them together. This process is usually referred to as "sub-listing" or "sort-merge."

The next question involves the type of sort logic (there are many ways of putting things in order). The algorithm used here is called a *bubble* sort. The file or list is examined two records at a time. If the second has a lower sort key than the first, the two will exchange places within the file. Why then, you ask, is it called a bubble sort. Because records appear to "bubble" upward in memory (I didn't coin the phrase so don't blame me). Although this is not a very exotic methodology, it does offer several advantages. It requires no other memory allocations for sorting and is fast if the file is not too disorganized. It will also not disturb the relative positioning of records that have equal sort keys.

There are numerous other types of sort algorithms. A *selection* sort would go through a list of (n) items (n-1) times, pulling out the next lowest record and adding it to the current end of a new list. This would need double the memory, though. A *selection and exchange* would perform a similar function within the main sort area, selecting the lowest element during each pass, moving it upward in the list to be exchanged with the element occupying its new position . This method tends to upset the existing relative positioning. Other types involve binary tree searches and more complex algorithms.

Why Machine Language

The choice of language is, as stated above, rather clear. Unless you have a lot of time to kill, your sort must be in executable object code (machine language). When you're doing several hundred thousand (or million ?) character comparisons and swaps, you don't have time to pull out a "BASIC dictionary" for each line in the program (this, in essence, is what the BASIC does).

Here are some representative execution times, based on some testing we did last winter. The speeds are approximate and do not include disk input/output time. The test file consisted of 200 records, each 75 characters in length. The sort key occupied ten positions:

BASIC selection/exchange sort (in memory) – 8 minutes

BASIC bubble sort (in memory) – 12 minutes

BASIC selection sort (on disk) – 2 HOURS plus (hit BREAK key)

Machine Language bubble (memory) – 3 seconds

The sort program was developed with flexibility in mind. It will sort fixed length records up 150 bytes in size. The sort key itself may be located anywhere in the record and can be any length (up to the size of the record). It will sort in either ascending or descending order. The records themselves must be comprised of ASCII (ATASCII) characters. While in memory, they need not be terminated by end-of-line ($9B) characters.

The nominal limit of 150 characters is imposed by a possible bug in ATARI's DOS II. The second half of page five (memory addresses 0580-05FF Hex, 1408-1535 Decimal) appears to be utilized as an internal I/O buffer. When more than 128 bytes are input, the excess winds up on page six. The sort program also resides in the safe (?) user area of page six (beginning at $0620 or 1568). There is a physical law that states: two things cannot occupy the same place at the same time. This also holds true in computer memory. The program has been pushed as far into page six as it can go (there is data stored behind it).

Using The Sort

In order to use the sort, you must feed it certain parameters. The record length must be POKEd into location 205 ($00CD). The sort type (0-Ascending, 1-Descending) would be POKEd into 206 ($00CE). The starting and ending positions of the sort key will also have to be POKEd into locations 203 ($00CB) and 204 ($00CC). The program is expecting to see the offset of the sort key. The offset is the number of positions in front of that byte. For example; the first position of a record has a 0 offset, the second has an offset of 1, and the 100th has an offset of 99. The USeR function that calls the sort will also pass the address of the string containing the file and the record count. For those who are a little unsure of what this is all about, there are a few examples coming up.

Now that you have a routine that will sort your data faster than you can say Rumplestilskin, how do you use it? Here are several suggestions. The best method is to link through our sort/file loader in Program 3. Your existing program that is processing the data file is probably much, much longer than the short loader. The main advantage of using a small program is that you wind up with more free memory. And, since memory is our sort area, the more that is free, the larger the file. If you don't type the REMark statements, you'll have even a larger sort area. The disk file must be fixed length records terminated by end-of-line characters. Your existing processing program must contain the POKEs mentioned above. It may look something like this:

```
POKE 203,SKEYA-1:POKE 204,SKEYB-1:POKE 205,
RECLEN:POKE 206,0 (for Ascending).
```

The call to the loader would be a RUN "D: SORTLOAD" (give the loader this file name when you save it). The sort/file loader must have your file name in the variable F$ and your program name in P$. If your processing program handles several files, you can also pass the file name by using the following statements. First, your program:

```
FOR I=0 TO 14:POKE 1776+I,32:NEXT I
FOR I=0 TO LEN(F$):POKE 1776+I,ASC(F$(I,I)):
  NEXT I
```

Note: F$ is your file's name.

The sort/file loader will require the following lines to be added:

```
70 FOR I=0 TO 14:F$(I,I)=CHR$(PEEK(1776+I)):
  NEXT I
80 IF F$(1,2)<>"D:" THEN ? "ERROR":END
```

If your processing program or file is small, you may do all of the above from within your program. Besides the same POKEs as above (you wouldn't need the file name, of course), you will need the following line added to your program:

```
IF RC>1 THEN A=USR(1568,ADR(X$),RC)
```

(RC is the number of records stored in the string X$.) Substitute your names where applicable.

Program 4 is a sort/merge utility that uses the same sort routine. This will give you the ability to handle much larger files. With a 40 or 48K machine you will be able to sort files that are 60,000 bytes long (If the record length is 60 characters, that will translate to 1,000 records). This particular version divides the file into two manageable sub-files, sorts each, and then merges them. Be careful with your disk space; the temporary file will need room also. If you have more than one drive, you can modify the program to split it three or more ways and sort even more records. For example, put the temporaries on drive 2 and the new file on drive 3. Who said micros can't handle larger files?

Your Options

The sort/merge program is a stand-alone. By swapping the front end with the sort loader (Program 3), you can do a sort/merge from a call (RUN "D: SORTMERG") in your existing software.

Now that you know how to feed the sort its required parameters and call it, you must still get it into memory. Once again, you have several options. If you have the Assembler/Editor cartridge (or a similar assembler), the source appears in Program 1. Please feel free to modify it if you so desire. If you're limited to BASIC, Program 2 will load the machine language code when it is run. After doing either of these, you should go directly to DOS (DOS II only) and do a binary save (option K) with the following parameters:

```
D1:AUTORUN.SYS,0620,069D
```

Saving the code as AUTORUN.SYS will enable the program to auto-boot when you power up with the disk (You *must* power up with that disk). Do *not* append an INIT or RUN address to the file unless you want the machine to lock up every time you turn it on.

Program 1.

```
;  RON MARCUSE, FREEHOLD NJ   11/29/81
;
;      CALLED FROM BASIC WITH:
;
;      A=USR(1568,ADR(X$),RC)
;
;  NOTE: X$ IS THE STRING THAT CONTAINS
THE FILE
;         RC IS THE NUMBER OF RECORDS
;
;  THE FOLLOWING ARE POKED BY BASIC PROG
RAM:
;
;       SS - BEGINNING OF SORT KEY (DECIM
AL- 203)
;       SE - END OF SORT KEY (DECIMAL - 2
04)
;       RL - RECORD LENGTH (DECIMAL - 205
)
;       TYPE - ASCENDING (0)  OR DESCENDI
NG (1)
;              (DECIMAL - 206)
;
; THE ROUTINE WILL LOOP THROUGH "FILE" S
WAPPING UNSORTED
; ADJOINING MEMBERS UNTIL THE "SWAP FLAG
" HAS NOT BEEN SET
; IN A GIVEN PASS. THE ZERO PAGE ADDRESS
ES "FST" AND "SEC"
; POINT AT THE INDIVIDUAL MEMBERS BEING
COMPARED. THE Y
; REGISTER IS USED AS AN INDEX POINTER F
OR TESTING OR
; MOVING BYTES WITHIN THE TWO RECORDS.
;
;
      *=   $0620   START AT PAGE 6
;     MEMBER n ADDRESS (LSB,MSB)
FST   =    $D4
;     MEMBER (n+1) ADDRESS (LSB,MSB)
SEC   =    $D6
;     BASE ADDRESS OF LIST (LSB,MSB)
BASE  =    $D8
;     FIRST POSITION OF SORT KEY
SS    =    $CB
;     LAST POSITION OF SORT KEY
SE    =    $CC
RL    =    $CD      ELEMENT LENGTH
SWAP  =    $DA      SWAP SWITCH
;     NUMBER OF ELEMENTS (LSB,MSB)
RC    =    $D0
;     RECORD COUNTER (MSB, X REG IS LSB)
CNTH  =    $CF
;     SORT TYPE, 0-ASC  1-DES
TYPE  =    $CE
;
;
;     POP # OF ARGUMENTS FROM STACK
      PLA
      PLA
      STA   BASE+1   SET BASE ADDRESS
      PLA
      STA   BASE
      PLA
      STA   RC+1     SET ELEMENT COUNT
      PLA
      STA   RC
;
;
;     START EACH PASS THROUGH FILE
BEGIN LDA   #$00
      STA   SWAP     SET SWAP TO 0
      STA   CNTH     SET HIGH COUNT TO 0
;     SET X REGISTER TO 1 (LOW COUNT)
      LDX   #$01
;     SET POINTER (n) TO BASE
      LDA   BASE
      STA   SEC
      LDA   BASE+1
      STA   SEC+1
;
CONT  CLC
      LDA   SEC      RESET POINTERS-
      STA   FST       (n) to (n+1)
      ADC   RL
      STA   SEC       (n+1) to (n+2)
      LDA   SEC+1
      STA   FST+1
      ADC   #$00
      STA   SEC+1
;     ASCII STRING COMPARISON
      LDY   SS
;
;     ASCENDING OR DESCENDING?
COMP  LDA   TYPE
      BEQ   ASC      SORT IS ASCENDING
      LDA   (SEC),Y TYPE = DESCENDING
;     COMPARE ADJOINING MEMBERS
      CMP   (FST),Y
      BCC   BACK     (n)>(n+1)
      BEQ   INCR     (n)=(n+1) TRY AGAIN
      BCS   FLIP     (n)<(n+1)
;
ASC   LDA   (SEC),Y TYPE = ASCENDING
;     COMPARE ADJOINING MEMBERS
      CMP   (FST),Y
      BCC   FLIP     (n)>(n+1)
      BEQ   INCR     (n)=(n+1) TRY AGAIN
      BCS   BACK     (n)<(n+1)
;
INCR  INY            ADD 1 TO POINTER
      CPY   SE       END OF SORT KEY?
      BEQ   COMP     NO
      BCS   BACK     YES, NEXT ELEMENT
```

```
       BCC  COMP    NO
;
;      SWAP ELEMENTS (n),(n+1)
FLIP   LDA  #$01
       STA  SWAP    SET SWAP SWITCH ON
       LDY  RL      LOAD LENGTH
;
MOVE   DEY          SET  DISPLACEMENT
       LDA  (SEC),Y EXCHANGE BYTES
       PHA
       LDA  (FST),Y
       STA  (SEC),Y
       PLA
       STA  (FST),Y
       CPY  #$00    MORE BYTES TO SWAP?
       BNE  MOVE    YES
;
;      INCREMENT RECORD COUNTER
BACK   INX
       CPX  #$00    CHECK FOR >255
       BNE  TEST
       INC  CNTH    ADD 1 TO HIGH COUNT
;
TEST   CPX  RC      END OF FILE?
       BNE  CONT    NO
       LDA  RC+1    CHECK HIGH EOF
       CMP  CNTH
       BNE  CONT    NOT END OF FILE
       LDA  SWAP    TEST FOR END OF SORT
       CMP  #$00    ANY SWAPS?
       BNE  BEGIN   YES, START OVER
      NO, RETURN TO CALLING PROGRAM
       RTS
       .END
```

Program 2.

```
100 FOR I=1568 TO 1693:READ A:POKE I,A:N
EXT I
1568 DATA 104,104,133,217,104,133
1574 DATA 216,104,133,209,104,133
1580 DATA 208,169,0,133,218,133
1586 DATA 207,162,1,165,216,133
1592 DATA 214,165,217,133,215,24
1598 DATA 165,214,133,212,101,205
1604 DATA 133,214,165,215,133,213
1610 DATA 105,0,133,215,164,203
1616 DATA 165,206,240,10,177,214
1622 DATA 209,212,144,44,240,12
1628 DATA 176,19,177,214,209,212
1634 DATA 144,13,240,2,176,30
1640 DATA 200,196,204,240,227,176
1646 DATA 23,144,223,169,1,133
1652 DATA 218,164,205,136,177,214
1658 DATA 72,177,212,145,214,104
1664 DATA 145,212,192,0,208,241
1670 DATA 232,224,0,208,2,230
1676 DATA 207,228,208,208,172,165
1682 DATA 209,197,207,208,166,165
1688 DATA 218,201,0,208,144,96
```

Program 3.

```
10 REM SORT LOAD PROGRAM  LYNN MARCUSE 1
1/27/81
11 REM
12 REM CALLING PROGRAM MUST:
13 REM
14 REM *  POKE RECORD LENGTH INTO LOCATI
ON 205
15 REM *  POKE BEGINNING OF SORT KEY INT
O LOC 203
16 REM *  POKE END OF SORT KEY INTO LOCA
TION 204
17 REM *  POKE TYPE (ASCENDING - 0 OR DE
SCENDING - 1) INTO LOC 206
18 REM
19 REM THIS PROGRAM WILL LOAD FILE INTO
MEMORY AND CALL MACHINE
20 REM LANGUAGE ROUTINE. WHEN COMPLETED,
 YOUR PROGRAM MAY BE
21 REM RE-CALLED BY EQUATING P$ TO YOUR
PROGRAM NAME.
22 REM
50 DIM X$(FRE(0)-600),R$(130),F$(15),P$(
15),I$(1)
58 REM
59 REM REPLACE X'S WITH YOUR FILE & PROG
RAM NAMES
60 P$="XXXXXX":F$="XXXXXX"
99 REM GET RECORD LENGTH
100 RET=100:R=PEEK(205)
109 REM OPEN FILE AND INPUT RECORDS
110 ? " LOADING ";F$:TRAP 600:OPEN #2,4,
0,F$:L=1
120 TRAP 140:INPUT #2,R$:TRAP 40000
130 X$(L,L+R-1)=R$:L=L+R:GOTO 120
140 CLOSE #2:L=L-1:N=L/R:? " RECORDS LOA
DED= ";N
149 REM CALL MACHINE LANGUAGE SORT ROUTI
NE
150 IF N>1 THEN ? " BEGIN SORT":A=USR(15
68,ADR(X$),N)
160 RET=170:? " COMPLETED  SAVING ";F$
169 REM ERASE OLD FILE AND SAVE NEW ONE
170 TRAP 600:XIO 36,#2,0,0,F$:OPEN #2,8,
0,F$
180 FOR I=1 TO L STEP R:R$=X$(I,I+R-1):?
 #2;R$:NEXT I
190 CLOSE #2:XIO 35,#2,0,0,F$
199 REM RETURN TO YOUR PROGRAM ?
200 RET=200:TRAP 600:IF P$(3,4)<>"XX" TH
EN ? " LOADING ";P$:RUN P$
210 END
600 ? " ERROR - ";PEEK(195):CLOSE #2
610 ? " PRESS RETURN TO CONTINUE";:INPUT
```

```
 I$:GOTO RET
```

Program 4.

```
10 REM SORT MERGE PROGRAM  RON MARCUSE 1
2/01/81
11 REM
12 REM THIS PROGRAM WILL LOAD FILE INTO
MEMORY AND CALL MACHINE
13 REM LANGUAGE ROUTINE. IF FILE IS TOO
LARGE, THE SORTED DATA
14 REM WILL BE SAVED AS "D:TEMP" AND BAL
ANCE OF FILE WILL BE
15 REM READ AND SORTED. WHEN THIS STEP I
S FINISHED, THE TEMPORARY
16 REM FILE  WILL BE MERGED WITH THE SOR
TED DATA IN MEMORY.
17 REM
20 GRAPHICS 0:DIM F$(15):? :? ,"SORT/MER
GE UTILITY":POKE 82,1
30 ? :? "ENTER:":? :? "FILENAME (D:name.
ext) ";:INPUT F$
40 ? "RECORD LENGTH ";:TRAP 40:INPUT R:T
RAP Q3:IF R<2 OR R>150 THEN 40
50 ? "SORT KEY (1st,2nd) ";:TRAP 50:INPU
T SS,SE:TRAP Q3
55 IF SS>=SE OR SS<0 OR SE>R THEN 50
60 ? "ASCENDING - 0  OR DESCENDING - 1 "
;:TRAP 60:INPUT T:TRAP Q3
65 IF T<0 OR T>1 THEN 60
70 POKE 205,R:POKE 203,SS-1:POKE 204,SE-
1:POKE 206,T
80 XL=FRE(0)-600:DIM X$(XL),R$(R),T$(R),
D$(6)
90 Q1=210:Q2=600:Q3=40000:D$="D:TEMP"
100 ? "LOADING ";F$:TRAP Q2:OPEN #2,4,0,
F$:M=0
120 L=1:? "PASS 1 - ";:GOSUB 500:IF M=0
THEN 160
140 ? "WRITING ";D$:OPEN #3,8,0,D$:GOSUB
 560
150 ? "PASS 2 - ";:L=1:GOSUB 500
160 CLOSE #2:? "DELETING ";F$
170 TRAP Q2:XIO 36,#3,0,0,F$:OPEN #3,8,0
,F$
180 ? "WRITING ";F$:IF M=0 THEN GOSUB 56
0:GOTO 400
200 TRAP Q2:OPEN #2,4,0,D$:J=1:A=1:B=1:A
E=1:BE=1
210 IF A=1 THEN TRAP 330:INPUT #2,R$:TRA
P Q3
220 IF B=1 THEN TRAP 340:T$=X$(J,J+R-1):
J=J+R:TRAP Q3
230 IF AE=0 AND BE=0 THEN 390
240 IF AE=1 AND BE=0 THEN 300
```

```
245 IF AE=0 AND BE=1 THEN 310
250 IF T=1 THEN 280
260 IF R$(SS,SE)>T$(SS,SE) THEN 310
270 GOTO 300
280 IF R$(SS,SE)<T$(SS,SE) THEN 310
300 ? #3;R$:A=1:B=0:IF AE=0 THEN A=0:B=B
E
302 GOTO Q1
310 ? #3;T$:A=0:B=1:IF BE=0 THEN B=0:A=A
E
312 GOTO Q1
330 AE=0:GOTO 220
340 BE=0:GOTO 230
390 CLOSE #2:? "DELETING ";D$:XIO 33,#2,
0,0,D$
400 CLOSE #3:XIO 36,#3,0,0,F$
410 END
500 TRAP 530:INPUT #2,R$:TRAP Q3
510 X$(L)=R$:L=L+R:IF (L+R)<XL THEN 500
520 M=1
530 L=L-1:N=L/R:? "RECORDS LOADED = ";N
540 IF N>1 THEN ? "BEGIN SORT  ";:A=USR(
1568,ADR(X$),N)
550 ? "END SORT":RETURN
560 FOR I=1 TO L STEP R:R$=X$(I,I+R-1):?
 #3;R$:NEXT I:CLOSE #3:RETURN
600 ? "ERROR - ";PEEK(195):END
```

©

Dynamic Renumber

R. D. Young
Ottawa, Ontario

Program line renumbering is often more than just cosmetic. Afterthoughts, frequently called *bugs*, invariably use up all those spaces left between original program lines. There are a number of line renumbering programs/utilities available for PET (and other computer) owners. Unfortunately, those that I have seen, including Toolkit, renumber the entire program, once invoked. It is therefore impossible to retain blocks of subroutines, as might be initially intended.

Blocks of subroutines, 1000-1999 or 2000-2999 for example, are particularly helpful during program development. It is easier to remember a thousand-line block while debugging (and leaving lots of space between blocks) than, for example, something like 760-790. At the same time, the mainline program or a subroutine block of lines may require renumbering during the debugging stage. A segment of the program can now be renumbered with Dynamic Renumber.

This program is a modified version of Resequencer by Joe Trimble from PET User Notes, Issue 5, July-August 1978, which was modified by Jim Russo and Henry Chow in PET User Notes, Issue 7, November-December 1978.

Dynamic Renumber will renumber the selected range of lines beginning with the desired new line number and using the desired increments. It will abort if the highest renumbered line overlaps a line not selected for renumbering, but it will give erroneous line numbers if the overlap occurs at the beginning of the renumbered segment. The program will then locate all GOTO's, GOSUB's, THEN's, ON...'s, and RUN's, and insert the new target line number if required. If, however, the new target line number is longer than the old line number, only part of the new line number will be inserted. When such an event occurs, the line number of the line in which the shortened insertion is being made and the proper target line will be printed side-by-side on the screen. An asterisk is printed as each program line is being analyzed for required changes.

This program will function quite nicely as a utility stored in and run from a 4K memory partition. The program to be renumbered must, of course, reside in the normal low end of memory. Alternatively, this program can be readily appended to a program already in memory.

Dynamic Renumber can be easily converted to other than PET BASIC, provided that line numbers are stored in the same manner (see also "Program Compactor," **COMPUTE!** #11). The first four bytes of each line are defined as follows:

Pointer to next line – low byte
Pointer to next line – high byte
Line number – low byte
Line number – high byte

Changes to Dynamic Renumber, required before implementation with other BASIC's, are the start-of-BASIC pointer and the GOTO, GOSUB, etc. token values. The start-of-BASIC in the PET is 1025 decimal; this is the number that must be changed in lines 63895, 63933, and 63937. The applicable statement tokens are in line 63940 (assigned to variable P).

As one last precaution, you may wish to retain the space between the variable LE and the statement THEN in the associated IF...THEN statements, thus avoiding BASIC confusion with the LET statement.

```
63776 REM END RENUMBER
63887 REM LINE RENUMBER - RUN63888
63888 PRINT"RENUMBER":INPUT"START AT LINE #";LS
63889 INPUT"END AT LINE #";LE:IFLE>=63776THENLE=63775
63890 IF LS>= LE THEN63888
63891 INPUT"FIRST NEW LINE #";Z
63892 INPUT"INCREMENT NEW LINES BY";K
63895 DIML(500):L=1025:DEFFNR(X)=PEEK(X)+256*PEEK(X+1):REM*OLD ROM DIM L
      (255)*
63900 DEFFNM(X)=INT((K*X-K+Z)/256)
63902 N=FNR(L):X=FNR(L+2):IFX<LSTHENL=N:GOTO63902
63904 L1=L
63910 N=FNR(L):X=FNR(L+2):IFX<= LE THENA=A+1:L(A)=X:L=N:IFN=0THEN63920
63912 IFX<=LE THEN63910
63915 Y=INT(K*A-K+Z):IFX<=YTHENPRINT"MAX. LINE OVERLAP - CK. PGM":END
```

```
63920 L=L1:FORB=1TOA:N=FNR(L):POKE(
      L+3),FNM(B)
63930 POKE(L+2),K*B-K+Z-256*FNM(B):
      L=N:NEXT
63933 L=1025
63935 N=FNR(L):X=FNR(L+2):IFX<63776
      THENAA=AA+1:L=N:IFN<>0THEN
      63935
63937 L=1025:FORB=1TOAA:N=FNR(L):X=
      FNR(L+2)
63940 F=0:FORC=L+4TON-1:P=PEEK(C):
      IFP=137ORP=141ORP=167ORP=138
      THENF=1:GOTO63999
63950 IFF>0THENF=0:IFP<58THENF=1:G=
      G+1:IFP>47THEND=10*D+P-48:
      GOTO63999
63960 IFD=0GOTO63999
63970 FORE=1TOA:IFD=L(E)GOTO63990
63980 NEXTE:D=0:G=0:GOTO63999
63990 D=0:E$="      "+STR$(E*K-K+Z):
      H=LEN(E$):C=C-G:IFP<48THENG=
      G-1:C=C+1
63995 IFH-6>GTHENPRINTX;E*K-K+Z;
63997 FORI=1TOG:POKEC,ASC(MID$(E$,I
      +H-G,1)):C=C+1:NEXTI:G=0
63999 NEXTC:L=N:PRINT"*";:NEXTB:END
READY.
```

©

ASERT yourself . . . with CFI's new Database Retrieval System

WHO CAN USE ASERT?

libraries
personnel departments
dating services

OFFICIALLY APPROVED BY COMMODORE

schools
employment agencies
accountants

ANY BUSINESS THAT KEEPS RECORDS CAN USE ASERT TO:

- Create up to 21 fields per record
- Restructure fields at any time
- Sort on any field at any time
- Use FREE-TEXT area for comments
- Create up to 90 searchwords
- Search & retrieve on any combination of 90 searchwords
- Search with MUST HAVE, MAY NOT HAVE and OPTIONAL operators
- Print out hardcopy including labels
- Output to any word processor
- Compile summary statistics
- Maintain 1900 records per disk with "virtual" 5K record length

ASERT — Aid for Search & Retrieval of Text — $495 complete

For the 8032 CBM and 8050 disk drive — Commodore Approved Software

OTHER CFI SOFTWARE

Federal Income Tax Preparation System*
Personal Tax Calculator*
Emergency Control Program*
VIC Animation Tutorial

*Distributed for CFI under the Commodore label

ALL CFI SOFTWARE AVAILABLE

from your local Commodore dealer
or direct from CFI

CFI . . . *Computer Solutions,* 201 West 92 St., New York, NY 10025

NOW COMMODORE TALKS. AND LISTENS.

COGNIVOX VIO-1002 is a speech recognition and voice output peripheral for Commodore computers. It offers state-of-the-art capabilities with pleasant sounding natural voice output and recognition performance equal to that of units costing many times more. Its capabilities, the very affordable price and its variety of uses makes it a "must have" peripheral.

Many uses

With COGNIVOX your imagination is not the limit, as the saying goes. It is the starting point. Use it for data entry when the hands and/or eyes are busy. As an educational tool. As an aid to the handicapped. Or as a foreign language translator, a sound effects generator, a telephone dialing device, an answering machine, a talking calculator or clock. Use it in conjunction with the IEEE 488 port to control by voice instruments, plotters, test systems. And all these devices can talk back to you telling you their readings, alarm conditions, even their names. Or use it with a BSR controller interface (see Compute, Oct. 81) to control by voice lights and appliances in the house.

Some specifications.

COGNIVOX can be trained to recognize words or short phrases drawn from a vocabulary of up to 32 entries chosen by the user. To train COGNIVOX to your vocabulary, all you have to do is repeat the entries three times at the prompting of the computer. The voice output vocabulary can also have up to 32 words or phrases of your choice. Data rate is approximately 700 bytes per word. Vocabularies can be stored and recalled from disk, giving in effect unlimited selection of entries.

COGNIVOX VIO-1002 will work with all Commodore computers (old, new and newer ROMs) with at least 16K of RAM. It comes complete with assembled and tested hardware in a quality instrument case, speaker/amplifier, power supply, microphone, cassette with software and detailed user manual.

Easy to use

All you need to get COGNIVOX up and running is to plug it in the user port and load one of the programs supplied. Load the demon program and start talking to your computer right away. Or load one of the games and discover the magic of voice control.

It is easy to write your own talking and listening programs too. A single statement in BASIC is all that you need to say or to recognize a word. Full instructions on how to do it are given in the manual.

Order your COGNIVOX now.

To order by mail send us a check or money order for $249 plus $5 shipping and handling (CA residents add 6% tax). You may also order by phone and charge it to your VISA or MASTERCARD. Call us at (805) 685-1854, 9AM to 5PM, PST, M-F. Foreign orders are welcome, please add 10% for air mail shipping and handling.

VOICETEK

P.O. Box 388, Goleta, CA 93116

Disk Data Structures:

An Interactive Tutorial

David Young
Richardson, TX

The floppy disk is a marvelous and yet mysterious medium for mass storage of data. Indeed, understanding exactly how a bit of data is stored and retrieved from the surface of the disk requires a good knowledge of physics. However, to learn about the data structures found on a disk requires mathematics no more complex than hexadecimal arithmetic. The manual supplied with the computer usually does an adequate job of supplying all the technical details, but wouldn't it sink in better if the actual data on the media could be viewed while it is being described?

The program that is presented here, Diskpeek, was created just for that purpose. Though this program was written for the Atari Personal Computer (DOS 2.0S), the interactive tutorial which follows contains information which should apply, in one form or another, to most other disk based computer systems. Those with a disk based Atari computer should type in Diskpeek before proceeding. This program is used to demonstrate the disk data structures as they are being described. The instructions integrated into the program should make its use self-explanatory.

The Disk Medium

The first disk structure to be aware of is the sector which, on any computer system, consists of a group of contiguous bits recorded at a specific location on the disk. The disk drive hardware always operates on whole sectors, that is to say, it is not possible to read or write partial sectors. Groups of sectors are organized into tracks forming concentric rings about the center of the disk.

The Atari system divides the disk into 40 tracks with 18 sectors per track for a total of 720 sectors. This is best visualized by taking the lid off of the disk drive and watching the read/write head move as certain sectors are addressed. On the Atari 810 disk drive this is accomplished by removing the four phillips head screws hidden under gummed tabs at each corner of the lid. While inside the case, a bit of lubrication on the 2 cylindrical guide rails supporting the head will make the drive less noisy.

If sectors 1 through 18 are read with Diskpeek, the head remains fixed on the outermost track. When sector 720 is read, the head moves in to the innermost track. When a disk is formatted, the head can be seen to bump sequentially through all 40 tracks. It is laying down the patterns on the oxide surface which will be recognized by the drive hardware as the sectors. The sectors are all initially empty (128 bytes of 0), but at the end of the formatting routine, as described in the next section, the Atari DOS records special data into certain sectors. The top of the drive can now be resecured. No more information about the hardware is needed to underatand the higher level disk data structures of the software.

Boot Sector

At the end of the formatting process, DOS reserves and initializes certain sectors for special tasks. Into sectors 1 through 3 is stored the bootstrap for DOS. On power-up the Atari operating system reads sector 1 to determine how many sectors to read and where into memory to load them. After it has loaded in the specified number of sectors, DOS starts executing the new code at the load address + 6. Put Diskpeek into the hex mode and read sector 1 of any DOS disk. Byte 0 says that 3 sectors are read (sequentially) and bytes 1 and 2 specify a load address of $700. (A 2 byte number is always specified with the least significant byte first.) Byte 6 is the first instruction to be executed (a $4C1407 is a JMP $714). In this case the code which follows sets up to load the File Management System of DOS into memory. This is called the second stage of the boot. Look at the first sector of any other boot disk available (any game or program which loads in from disk on power-up). It might be seen that the program loads in entirely during the first stage of the boot, i.e. byte 1 of sector 1 has a sector count which represents the entire program. For more details on the disk boot process, see the Atari *Operating System User's Manual.*

Volume Table Of Contents

Besides the first three boot sectors, DOS sets up sectors 360 to 368 as the directory of the disk. DOS uses the directory to keep track of where files are stored on disk and how much disk space remains. Read sector 360 of a DOS disk with Diskpeek in the hex mode and view a part of the directory called the Volume Table of Contents (VTOC). Information pertaining to the availability of every sector on the disk is stored in this sector. Bytes 1 and 2 specify the maximum number of user data sectors on the disk ($2C3 = 707) and bytes 3 and 4 specify the number of free sectors remaining on

Announcing

ATARI

software ™

from the
authors of
An Invitation to Programming

exciting games
and educational programs
for kids,
teenagers
and
adults
featuring sound
and color graphics.

available on
guaranteed-to-load
cassettes
at fine
computer dealers in your
area or,
write us directly for
descriptive materials

PDI

Program Design, Inc.
Department CA
11 Idar Court
Greenwich, CT 06830

203-661-8799

See us at the West Coast Computer Faire — Booth 1348.

MICROWORLD, an adventure within your computer, is available on the ATARI and the TRS-80. You are transformed into an electroid, and must explore the circuits of your computer. Over 80 locations and many original problems exist within the maze of transformers and transistors. We dare you to explore the maze of bit cells! Each version of Microworld explores the workings of its respective computer, Atari or TRS-80. Microworld comes with a booklet defining terms and describing the function of the mystifying inner workings of home computers. Come face to face with a staticon! Explore the Microworld!

SATISFACTION GUARANTEED!
If for any reason you are not satisfied with our products, return your order within 14 days for a prompt and cheerful refund.

ORDERING INFORMATION
Orders are processed within five working days. Shipping and handling charge of $1.00 will be added to all orders within the U.S. and Canada. Overseas orders please add $3.00 for air post.

Atari Microworld Atari 400 and 800
32K Cassette $19.95
Atari 400 and 800
32K Diskette $22.95
TRS-80 Microwrold TRS-80 Model I and Model III
Level II 16K Cassette $19.95
TRS-80 Model I and Model III
Level II 32K Diskette $22.95

■ MICROWORLD $
SHIPPING 1.00
TOTAL $

■ TRS-80 ■ ATARI
■ CASSETTE ■ DISKETTE
NAME
STREET
CITY STATE ZIP
■ CHECK ■ VISA ■ MASTERCARD
EXP. DATE

MED SYSTEMS SOFTWARE
P.O. Box 2674, Chapel Hill, NC 27514
1-800-334-5470

the disk (707 for an empty disk, 0 for a full one). Starting in bit 6 (the second to highest order bit) of byte $0A, each bit up through byte $63 corresponds to a sector. A 1 corresponds to a free sector while a 0 means the sector is being used.

When a file is stored on the disk, the bits corresponding to the sectors used are set to 0. When the file is erased, the bits are set back to 1. That is why DOS, when it deletes a file, can be heard reading the entire file. It is determining which sectors were being used by the file so that it can free them back up. Notice that even on a newly formatted disk, sector bits 1, 2 and 3 (bits 6, 5 and 4 of byte $0A) are set to 0. These correspond to the 3 boot sectors. Likewise, the nine bits starting in byte $37 are 0 because they correspond to the sectors of the directory. These 12 sectors are thus kept from being overlaid by user files.

If the VTOC is viewed on an older disk which has had many file additions and deletions, it may be noted that the VTOC has become quite fragmented. Any file added to the disk may get stored into sectors scattered about the disk. How DOS keeps track of files spread over multiple sectors will be discussed shortly. By the way, even though the operating system recognizes sector 720 (try reading it; should be all zeroes), DOS never makes use of it. True to Murphy's Law, it adopted the number scheme of 0 to 719 instead of 1 to 720. No need to bother trying to read sector 0!

The Directory

Of all the disk data structures, probably the most important one to be acquainted with is the directory. The eight sectors following the VTOC (361-368) contain a list of all the files on the disk along with their size, starting sector, and status. Put Diskpeek into character mode and read sector 361 of the DOS disk that has several files on it. It can be seen that the name of the first file starts in byte $05 and the extension (if any) starts in byte $0D. If any of the 11 character positions of the filespec are unused, it contains a blank. Notice that the filenames start every 16 bytes, allowing eight directory entries per 128 byte sector. Thus, the maximum number of entries for the eight sectors of the directory is 64.

Now put Diskpeek in hex mode and read sector 361. The first byte of each 16 byte entry contains the status of the file. For a normal file that byte is $42, unless it is locked, in which case it has a status of $62. A deleted file has a status of $80. An anomaly occurs whenever a file is opened for output (from BASIC, perhaps) but is not closed before the computer is powered down or glitched. Since the status of an open file is $43, DOS will neither recognize the entry as "in use" nor "deleted." Even the sectors which may have been written out will not really exist on disk because the VTOC is not updated until the file is closed. The only harm done is that this bogus entry will take up space in the directory until the disk is reformatted. The second and third bytes of each entry contain the size in sectors of the file (low order byte first) while the fourth and fifth bytes specify the first sector of the file. DOS only needs to know the first sector of a file because each sector points to the next sector of the file in a process called "linking."

Linking

At this point it would be best to explain how DOS forms a data file on disk. First, the user must open an I/O channel for output to the disk, perhaps with the BASIC "OPEN" command. DOS responds by creating an entry in the directory with the specified filename and a status of $43. DOS reads the VTOC into memory and searches the disk map for the first free sector. If a free sector is found, its number is used as the starting sector in the directory entry. Now, when the user begins to output data via this I/O channel, perhaps with the BASIC "PUT" command, DOS waits until it has collected 125 bytes of user data in a buffer. Then DOS adds three special bytes of its own and outputs the sector to the disk. I call these three bytes the "sector link."

The sector link, bytes 125 to 127 of the sector, contains three pieces of information. The high order six bits of byte 125 contain a number which represents the position of the file's entry within the directory (0 to 63). DOS uses this number to check the integrity of the file. If ever this number should fail to match the position of the file's directory entry, DOS generates an error. The low order two bits of byte 125 and all of byte 126 form a pointer to the next sector of the file. A pointer is the address of a record in the computer's memory or, in this case, the address of a record on disk, the sector number.

The next sector of the file is determined by scanning the bit map of the VTOC for the next free sector, which may or may not be the next sequential sector of the disk. Thanks to the link pointers, all sectors of a file need not be contiguous sectors on the disk. The last byte of the sector link (byte 127 of the sector) contains the number of bytes used within the sector. This byte will always be $7D (125) except for the last sector of a file, which will probably be only partially filled. DOS writes out this partial sector only when the user closes the file, perhaps with the BASIC "CLOSE" command.

When an output disk file is closed, DOS writes the newly updated VTOC back out to sector 360. It then updates the file's directory entry by changing the status to $42 and filling in the file size (bytes 1 and 2) with the number of sectors used by the file. This completes the process of creating a file on disk. Now, when DOS is requested to read a file from disk, it finds the directory entry of the specified file to determine the start sector. Then, following the link pointers, it reads the file, sector by sector, until EOF (end of file) is reached, indicated by a link pointer of 0.

Equipped with a basic understanding of how a file is stored on disk, try looking at a file with Diskpeek. In character mode, first locate the name of the desired file in the directory (sectors 361-368). Then put Diskpeek in hex mode and look at the fourth and fifth byte of the entry to determine the start sector. For example, if these two bytes were "01 02" then type "$201" to read the first sector.

Observe the last three bytes of the sector and verify that the high order six bits of byte 125 correspond to the directory entry position and that byte 127 is the number of bytes used (probably $7D). Then determine the next sector of the file from the low order two bits of byte 125 and byte 126. For example, if bytes 125 and 126 are "06 02" then the next sector of the file is $202 and the file is the second entry of the directory (the first entry being entry zero). If the file is not too long, it would be instructive to follow the sector links to EOF. Once the ability of finding a file on disk and following the sector links is mastered, all that remains is to become familiar with the three types of files used by DOS.

File Types

The first type of file is not a true file, per se, because there is no entry in the directory for it. This file type includes the boot record and the directory itself. And, since the sectors which make up these files are not linked, but, instead, are related to each other sequentially, I call these records "sequentially linked files." When examining a sector of the boot record or directory, merely increase the sector number by one to get to the next sector of the record.

An example of the second type of file is that which is created with the BASIC LIST or SAVE command. This file consists of ASCII characters which either represent straight text, as in a LISTed file, or a sort of condensed text, as in a tokenized or SAVEd file. Except when viewing the sector links, the character mode of Diskpeek is best suited for examining this type of file. At this point it would be instructive to locate (in the directory of a DOS disk) a file created with the BASIC LIST command.

Upon determining the start sector, observe the file in the character mode. The BASIC program can be easily recognized. It may be noted that the carriage return-line feed character (CRLF) is displayed in its ATASCII representation (an inverse escape character) instead of being executed. Now observe a file that consists of a program that was SAVEd from BASIC. Since the text has been tokenized, the program is harder to recognize. However, certain parts of the program are not altered during the tokenization process, notably text following REM and PRINT statements. Now, having investigated ASCII files, it is time to discuss the last file type, the *binary load* file.

The binary load file is primarily used to load 6502 machine code into memory for execution. However, its format is so general that it can be used just as easily to load any type of data, including ASCII text. Locate a game or other program which is run with the BINARY LOAD option of DOS. Alternatively, create a binary load file by saving any part of memory (except ROM) with the BINARY SAVE option. Now observe the first sector of the file with Diskpeek in the hex mode.

First, notice that all binary load files start with two bytes of $FF. The next four bytes are the start and end addresses, respectively, where the data to follow will be loaded into memory. If these four bytes were "00 A0 FF BF" then the data would be loaded between the addresses of $A000 and $BFFF. I call these four bytes a *load vector*. After DOS has loaded in enough bytes to satisfy the load vector, it assumes (unless EOF is reached) that the next four bytes specify another load vector. DOS will continue inputting the file at this new address.

Upon completion of a BINARY LOAD, control will normally be passed back to the DOS menu. However, DOS can be forced to pass control to any address in memory by storing that two byte address at location $2E0. To store the two bytes, it is necessary to specify another load vector as part of the file. If, for example, it were desired to execute the program loaded in at $A000, the following load vector would be part of the file: E0 02 E1 02 00 A0. I call this specialized load vector an *autorun vector*. It achieves the same result as the RUN AT ADDRESS option of DOS. Try to find the autorun vector in the file being viewed. Although it could be at the beginning, it is most likely located at the very end of the file.

```
10 REM DISKPEEK: David Young 11/10/81
20 SETCOLOR 1,0,4:SETCOLOR 2,10,10
30 DIM HEXCHAR$(16),HEXBYTE$(2)
40 DIM HEXNUM$(113),SECTRW$(68)
50 DIM TEMP$(3),DFORM$(1)
60 ? CHR$(125):? "WAIT A FEW SECONDS..."

70 GOSUB 1130:GOSUB 970
80 GOSUB 660:RESTORE 90
```

```
90 DATA 0123456789ABCDEF
100 READ HEXCHAR$:OPEN #1,4,0,"K"
110 DFORM$="H"
120 ? CHR$(125):? "        DISKPEEK by Da
vid Young":?
130 ? "This is a disk utility for viewin
g"
140 ? "individual sectors of a disk. It"

150 ? "reads the sector specified by the
"
160 ? "user and then displays it's conte
nts"
170 ? "as a matrix of hex bytes or ATASC
II"
180 ? "characters.":?
190 ? "The sector number can be specifie
d in"
200 ? "decimal ('361') or hex ('$169').
Type"
210 ? "RETURN to toggle from one display
"
220 ? "format to the other."
230 POSITION 2,20:? CHR$(156);:? "Sector
 #";
240 INPUT HEXNUM$:IF LEN(HEXNUM$)<>0 THE
N 280
250 IF DFORM$="C" THEN DFORM$="H":GOTO 2
70
260 DFORM$="C"
270 GOSUB 770:GOTO 230
280 GOSUB 500:IF BYTE<0 OR BYTE>720 THEN
 GOSUB 350:GOTO 230
290 SECNUM=BYTE
300 GOSUB 880:IF X=1 THEN GOSUB 770
310 GOTO 230
320 REM
330 REM *** PRINT ERROR MESSAGE ***
340 REM
350 POSITION 2,19:? CHR$(156);CHR$(156);
CHR$(156);"NOT LEGAL NUMBER!":RETURN
360 REM
370 REM **** PRINT HEX BYTE ******
380 REM
390 GOSUB 430:PRINT HEXBYTE$;:RETURN
400 REM
410 REM *** HEX CONVERSION ***
420 REM
430 TEMPB=BYTE:BYTE=INT(BYTE/16)+1
440 HEXBYTE$(1,1)=HEXCHAR$(BYTE,BYTE)
450 BYTE=(TEMPB-(BYTE-1)*16)+1
460 HEXBYTE$(2,2)=HEXCHAR$(BYTE,BYTE)
470 BYTE=TEMPB:RETURN
480 REM
490 REM *** NUMBER CONVERSION ***
500 REM
510 TRAP 630:IF HEXNUM$(1,1)<>"$" THEN G
OTO 620
520 HEXNUM$=HEXNUM$(2)
530 IF LEN(HEXNUM$)=3 THEN HEXNUM$(4)=HE
XNUM$(3):HEXNUM$(3,3)=HEXNUM$(2,2):HEXNU
M$(2,2)=HEXNUM$(1,1):HEXNUM$(1,1)="0"
540 IF LEN(HEXNUM$)=2 THEN HEXNUM$(4)=HE
XNUM$(2):HEXNUM$(3,3)=HEXNUM$(1,1):HEXNU
M$(1,2)="00"
550 IF LEN(HEXNUM$)=1 THEN HEXNUM$(4)=HE
XNUM$(1):HEXNUM$(1,3)="000"
560 IF ASC(HEXNUM$(1,1))>64 THEN HEXNUM$
(1,1)=CHR$(ASC(HEXNUM$(1,1))-7)
570 IF ASC(HEXNUM$(2,2))>64 THEN HEXNUM$
(2,2)=CHR$(ASC(HEXNUM$(2,2))-7)
580 IF ASC(HEXNUM$(3,3))>64 THEN HEXNUM$
(3,3)=CHR$(ASC(HEXNUM$(3,3))-7)
590 IF ASC(HEXNUM$(4,4))>64 THEN HEXNUM$
(4,4)=CHR$(ASC(HEXNUM$(4,4))-7)
600 BYTE=(ASC(HEXNUM$(4,4))-48)+16*(ASC(
HEXNUM$(3,3))-48)+256*(ASC(HEXNUM$(2,2))
-48)+4096*(ASC(HEXNUM$(1,1))-48)
610 TRAP 40000:RETURN
620 TRAP 630:BYTE=VAL(HEXNUM$):GOTO 610
630 GOSUB 350:BYTE=-1:GOTO 610
640 REM
650 REM *** DISK READ/WRITE ***
660 REM
670 RESTORE 680:FOR K=1 TO 68:READ Q:SEC
TRW$(K,K)=CHR$(Q):NEXT K:RETURN
680 DATA 104,104,104,201,83,169,82,144
690 DATA 2,169,87,72,169,0,72,169
700 DATA 1,72,169,0,72,169,128,72
710 DATA 169,6,72,72,104,104,141,5
720 DATA 3,104,141,4,3,104,104,141
730 DATA 1,3,104,104,141,2,3,104
740 DATA 141,11,3,104,141,10,3,32
750 DATA 83,228,173,3,3,133,212,169
760 DATA 0,133,213,96
770 REM
780 REM *** DISPLAY SECTOR ***
790 REM
800 BYTE=INT(SECNUM/256):? CHR$(125)
810 ? "SECTOR # = ";SECNUM;
820 ? " ($";:GOSUB 370
830 BYTE=SECNUM-256*INT(SECNUM/256)
840 GOSUB 370:? ")"
850 IF DFORM$="H" THEN GOTO 870
860 X=USR(ADR(MEMCHAR$),1536+128):RETURN

870 X=USR(ADR(MEMHEX$),1536+128):RETURN


880 REM
890 REM *** READ SECTOR ***
900 REM
910 X=USR(ADR(SECTRW$),82,SECNUM)
```

```
920 IF X=1 THEN 950
930 POSITION 2,19
940 ? "CAN'T READ SECTOR ";SECNUM;"!"
950 RETURN
960 REM
970 REM *** DISPLAY MEM IN HEX ***
980 REM
990 DIM MEMHEX$(122)
1000 RESTORE 1010:FOR K=1 TO 122:READ Q:
MEMHEX$(K,K)=CHR$(Q):NEXT K:RETURN
1010 DATA 104,104,133,229,104,133,228,16
9
1020 DATA 0,72,104,72,16,7,169,155
1030 DATA 32,164,246,104,96,169,155,32
1040 DATA 164,246,104,72,74,74,74,74
1050 DATA 201,10,48,2,105,6,105,48
1060 DATA 32,164,246,104,72,41,15,201
1070 DATA 10,48,2,105,6,105,48,32
1080 DATA 164,246,169,32,32,164,246,169
1090 DATA 32,32,164,246,104,72,168,177
1100 DATA 228,74,74,74,74,201,10,48
1110 DATA 2,105,6,105,48,32,164,246
1120 DATA 104,72,168,177,228,41,15,201
1130 DATA 10,48,2,105,6,105,48,32
1140 DATA 164,246,169,32,32,164,246,104
1150 DATA 24,105,1,72,41,7,208,204
1160 DATA 240,144
1170 REM
1180 REM *** DISPLAY MEM IN CHAR FORMAT
***
1190 REM
1200 DIM MEMCHAR$(122)
1210 RESTORE 1220:FOR K=1 TO 122:READ Q:
MEMCHAR$(K,K)=CHR$(Q):NEXT K:RETURN
1220 DATA 104,104,133,229,104,133,228,16
9
1230 DATA 0,72,104,72,16,7,169,155
1240 DATA 32,164,246,104,96,169,155,32
1250 DATA 164,246,104,72,74,74,74,74
1260 DATA 201,10,48,2,105,6,105,48
1270 DATA 32,164,246,104,72,41,15,201
1280 DATA 10,48,2,105,6,105,48,32
1290 DATA 164,246,169,32,32,164,246,169
1300 DATA 32,32,164,246,169,1,141,254
1310 DATA 2,104,72,168,177,228,201,155
1320 DATA 208,11,169,0,141,254,2,169
1330 DATA 219,133,93,169,31,32,164,246
1340 DATA 169,32,32,164,246,169,32,32
1350 DATA 164,246,169,0,141,254,2,104
1360 DATA 24,105,1,72,41,7,208,204
1370 DATA 240,144
```

©

Apple Addresses

Bill Grimm
Mountain View, CA

The Apple II uses three types of addressing depending upon the language being used. Apple's machine language uses hexadecimal addresses in the range from $0000 to $FFFF. Its Floating Point BASIC language uses decimal addresses in the range from 0 to 65535. Its Integer BASIC uses decimal addresses in the range from 0 to 32767 to -32767 to -1. This means that, if you want to address a particular memory location, you must choose the correct address for the language you are using. Since I program in all three languages and my references are a mixture from all three, I needed an address cross-reference program. So I wrote "Apple Addresses."

"Apple Addresses" can be used "as is" to convert one language's address to another's, and to give the high and low byte values which need to be poked into a BASIC program to store that address. Alternatively, you could extract the subroutines in Apple Addresses which convert between hex and decimal numbers and insert them in your own program. See the last paragraph of this article for more details.

The program begins by asking the user which of the six possible conversions he would like to make. This is followed by a request to select the way the results of the conversions are to be displayed. There are four possible displays:

1. single conversions displayed on the monitor one at a time.
2. Single conversions printed out on a Silentype printer* one at a time.
3. a range of conversions displayed on the monitor.
4. a range of conversions printed out on a Silentype printer*.

*With slight program modifications other printers could be used.

Subroutines

"Apple Addresses" makes extensive use of subroutines. This helps in organizing the program as well as making it shorter and easier to debug. The controlling or EXECutive routine is called Apple Addresses – Exec. It starts on line 100 and goes to line 310. Since a picture is worth a thousand words, I made what I call a *balloon diagram* (Figure 1) to show how data flows through the program. These are the conventions I used to make the diagram;

1. Each balloon represents a subroutine. The name of the subroutine and the line numbers where it is located are placed in the balloon.
2. Data flows through a subroutine in the direction of the arrows on the outside of the balloon.
3. Data flows between subroutines in the direction of the arrows on the *strings*.
4. If conditions are placed on what data flows through a subroutine, these conditions are written in along the *strings*.

As an additional aid for understanding how the program works I have included the following variable descriptions list:

A( ) — each A(I) holds the decimal equivalent value of the Ith hexadecimal numeral in the hex number being created from a decimal number — appropriate numbers are then added to convert these to ASCII codes.

A$( ) — holds the characters represented by the ASCII codes in A( ).

CHOICE — holds the number of the conversion chosen — see lines 120 to 178.

DVL — holds the decimal value of the number being converted — may be either FP or INT decimal.

DVL$ — is the string equivalent of DVL and is used in the output routines.

FLAG — if flag = 1 then an invalid number was entered and the program returns to get a new number.

FRST — holds the FP Basic address equivalent of the lowest address in the selected range.

FRST$ — holds the smallest address chosen — this address is then processed and stored in FRST.

HVL$ — holds the hex number selected or the hex number resulting from the conversion — if no hex numbers are involved then it holds the converted decimal number.

LST — holds the FP Basic address equivalent of the largest address in the selected range.

LST$ — holds the largest address chosen — this address is then processed and stored in LST.

N — holds the decimal equivalent of each hex numeral in a hex number being converted to a decimal number.

PHI% — holds the number that would be poked into the high byte when placing the address into memory.

PLO% — holds the number that would be poked into the low byte when placing the address into memory.

POK — holds the address from which PLO% and PHI% are derived.

SELECT — holds the type of output selected — see lines 462 to 470.

STP — holds the positive decimal stepping interval chosen.

STP$ — holds the stepping interval chosen which is later changed and stored in STP.

TB — the horizontal tab value desired.

TN — holds the intermediate numbers of the decimal address that is being converted into a hex address.

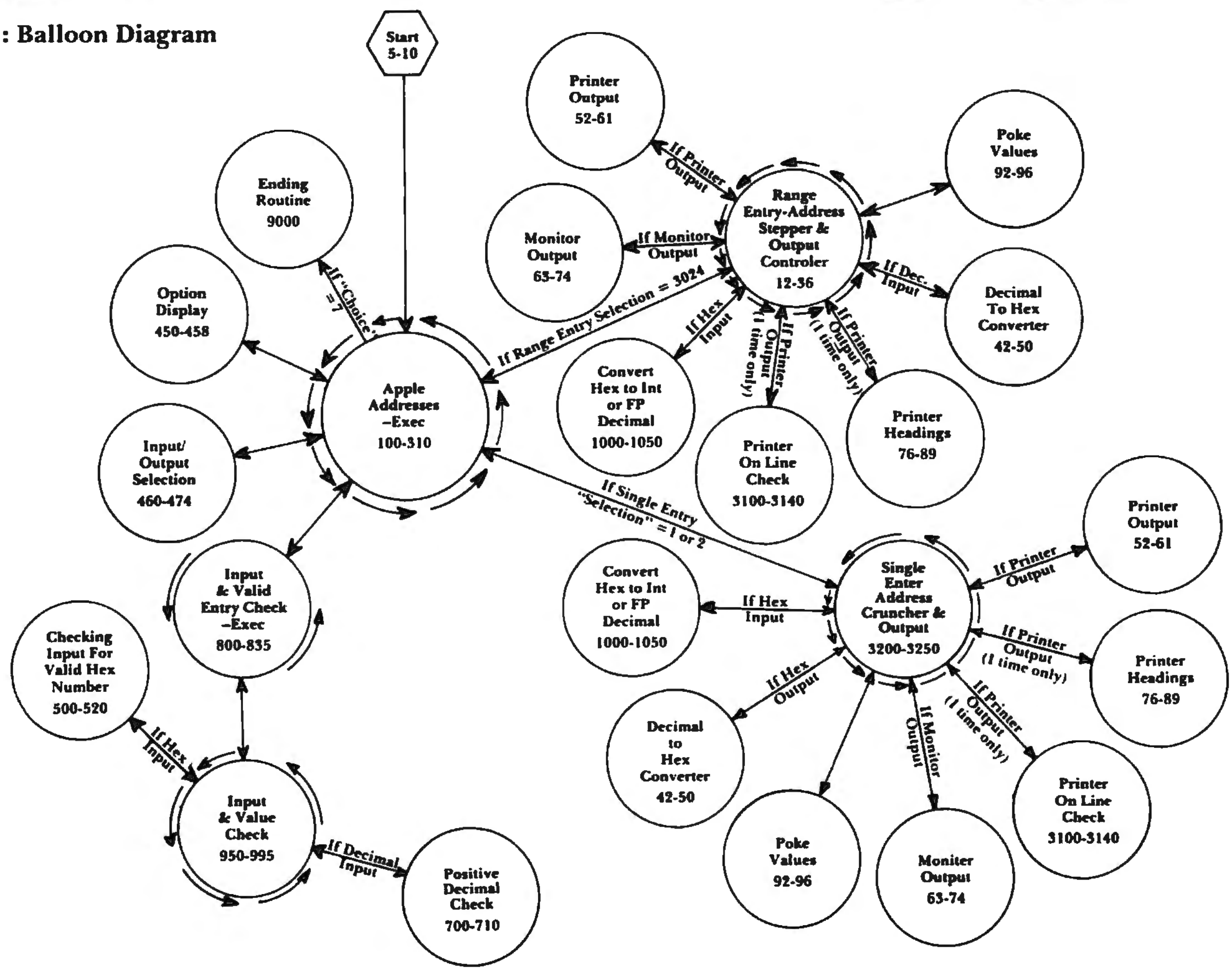
VTB — used to control the vertical tabbing of the monitor output.

Some Suggestions

I have found that the easiest way to debug a program while I am entering it is to first type in the EXEC program. Then, if I place return statements at all the branching locations, I can check the EXEC for bugs. Once the EXEC is free of bugs, I add one subroutine at a time in the order that the EXEC uses them, checking for bugs as I go.

If you have a need for subroutines which convert numbers from hex to decimal or from decimal to hex, two subroutines in this program may be of help. The first is called "decimal to hex converter" (lines 42 to 50). The input to this routine is TN which must hold a positive decimal number <65536. The output is HVL$ which holds the hex equivalent to the number in TN. The second is called "convert hex to INT or FP decimal" (lines 1000 to 1050). The input to this routine is HVL$ which must hold a hex number <=$FFFF and choice. If choice = 1 then you get the positive decimal equivalent. Otherwise you get Int BASIC's equivalent. The output is a decimal number in DVL.

Figure 1: Balloon Diagram




```
10  GOTO 100
12  IF CHOICE < 3 THEN IN$ = STP$: GOSUB 1000:STP = DVL:IN$ = LST$: GOSUB
     1000:LST = DVL:IN$ = FRST$: GOSUB 1000:FRST = DVL: GOTO 16
```

```
14 STP =  VAL (STP$):LST =  VAL (LST$):FRST =  VAL (FRST$)
16 VTB = 7:TB = 1: IF SELECT = 4 THEN  GOSUB 3100: POKE  - 12526,83: PR#
    1: PRINT : PRINT "CONVERTING FROM ";: ON CHOICE GOSUB 76,78,80,82,84
    ,86: POKE  - 12526,80
18  IF LST < 0 THEN LST = LST + 65536: IF FRST < 0 THEN FRST = FRST + 655
    36
19  FOR DVL = FRST TO LST STEP STP: IF CHOICE <  > 4 OR CHOICE <  > 6 THEN
    TN = DVL: GOSUB 42
20  IF CHOICE = 3 AND DVL > 32767 OR CHOICE = 4 AND DVL > 32767 OR CHOICE
     = 2 AND DVL > 32767 OR CHOICE = 6 AND DVL > 32767 THEN DVL = DVL -
    65536
22  IF CHOICE = 4 THEN HVL$ =  STR$ (DVL): IF DVL < 0 THEN HVL$ =  STR$ (
    DVL + 65536)
24  IF CHOICE = 6 THEN HVL$ =  STR$ (DVL): IF DVL < 0 THEN DVL = DVL + 65
    536
26  GOSUB 92
28  IF SELECT = 4 THEN  GOSUB 52: GOTO 32
30  GOSUB 62
32  IF DVL < 0 THEN DVL = DVL + 65536
34  NEXT DVL: IF SELECT = 4 THEN  PRINT : PR# 0
36  RETURN
42 HVL$ = "": FOR I = 4 TO 1 STEP  - 1:A(5 - I) =  INT (TN / (16 ^ (I - 1
    ))):TN = TN - (A(5 - I) * (16 ^ (I - 1))): NEXT I
44  FOR I = 1 TO 4: IF A(I) < 10 THEN A(I) = A(I) + 48: GOTO 48
46 A(I) = A(I) + 55
48 A$(I) =  CHR$ (A(I)):HVL$ = HVL$ + A$(I): NEXT I
50  RETURN
52 DVL$ =  STR$ (DVL): IF CHOICE < 3 THEN 58
54  PRINT  SPC( 6 -  LEN (DVL$));DVL$;: IF CHOICE = 5 OR CHOICE = 3 THEN
     PRINT ">$";HVL$; SPC( 1);: GOTO 59
56  PRINT ">"; SPC( 6 -  LEN (HVL$));HVL$;: GOTO 59
58  PRINT " $"; SPC( 4 -  LEN (HVL$));HVL$;">"; SPC( 6 -  LEN (DVL$));DVL
    $;
59  PRINT  SPC( 9 -  LEN (PLO$));PLO$; SPC( 14 -  LEN (PHI$));PHI$;:TB =
    TB + 39: IF TB > 42 OR SELECT = 2 THEN TB = 1: PRINT
60  HTAB TB: IF TB = 40 THEN  PRINT  SPC( 3);
61  RETURN
62  REM
63 DVL$ =  STR$ (DVL): VTAB VTB: HTAB TB: IF CHOICE < 3 THEN 68
64  PRINT  SPC( 6 -  LEN (DVL$));DVL$;: IF CHOICE = 5 OR CHOICE = 3 THEN
     PRINT ">$";HVL$; SPC( 2);: GOTO 70
66  PRINT ">"; SPC( 6 -  LEN (HVL$));HVL$; SPC( 1);: GOTO 70
68  PRINT "$0000>";: HTAB TB + 5 -  LEN (HVL$): PRINT HVL$;: HTAB TB + 12
     -  LEN (DVL$): PRINT DVL$; SPC( 2);
70  PRINT  SPC( 8 -  LEN (PLO$));PLO$; SPC( 14 -  LEN (PHI$));PHI$:VTB =
    VTB + 1: IF VTB > 23 THEN  HTAB 3: INPUT "PRESS <RETURN> TO CLEAR SC
    REEN";IN$: HOME :VTB = 6:TB = 1: GOTO 72
71  GOTO 74
72  IF IN$ = "Q" THEN  POP : GOTO 100
73  IF SELECT = 3 THEN VTB = 7
74  RETURN
76  PRINT "HEX TO FP DECIMAL": GOSUB 88: RETURN
78  PRINT "HEX TO INT DECIMAL": GOSUB 88: RETURN
80  PRINT "INT DECIMAL TO HEX": GOSUB 88: RETURN
82  PRINT "INT DECIMAL TO FP DECIMAL": GOSUB 88: RETURN
84  PRINT "FP DECIMAL TO HEX": GOSUB 88: RETURN
86  PRINT "FP DECIMAL TO INT DECIMAL": GOSUB 88: RETURN
88  IF SELECT = 2 THEN  PRINT : PRINT " CONVERSION   POKE LO BYTE   POKE H
    I BYTE": RETURN
```

```
89 PRINT : PRINT " CONVERSION   POKE LO BYTE  POKE HI BYTE   CONVERSION
     POKE LO BYTE  POKE HI BYTE": RETURN
92 POK = DVL: IF POK < 0 THEN POK = POK + 65536
94 PHI% = POK / 256:PLO% = POK - PHI% * 256
96 PHI$ =  STR$ (PHI%):PLO$ =  STR$ (PLO%): RETURN
100  POKE  - 16298,0: TEXT : HOME :FLAG = 0
110  VTAB 7
120  PRINT " 1. CONVERT HEX ADDRESSES TO FP BASIC": PRINT
130  PRINT " 2. CONVERT HEX ADDRESSES TO INT BASIC": PRINT
135  PRINT " 3. CONVERT INT BASIC ADDRESSES TO HEX": PRINT
140  PRINT " 4. CONVERT INT BASIC ADDRESSES TO FP": PRINT
150  PRINT " 5. CONVERT FP BASIC ADDRESSES TO HEX": PRINT
160  PRINT " 6. CONVERT FP BASIC ADDRESSES TO INT": PRINT
162  PRINT " 7. QUIT": PRINT
165  PRINT : PRINT "NOTE: ENTERING A 'Q' AT ANY POINT            RETURNS
     YOU TO THIS MENU."
170  VTAB 4: INPUT "CHOOSE ONE:";IN$
175  IF IN$ = "7" THEN 9000
178 CHOICE =  VAL (IN$): IF CHOICE < 1 OR CHOICE > 6 THEN 100
180  GOSUB 450: GOSUB 460: HOME : VTAB 1: HTAB 13: ON SELECT GOTO 190,195
     ,200,210
190  PRINT ": SINGLE ENTRY : MONITOR": GOTO 220
195  PRINT ": SINGLE ENTRY : PRINTER": GOTO 220
200  PRINT ": RANGE ENTRY : MONITOR": GOTO 220
210  PRINT ": RANGE ENTRY : PRINTER"
220  HOME : IF SELECT < 3 THEN  PRINT "ENTER NUMBER": GOTO 250
230  PRINT "FIRST NUMBER";: HTAB 22: PRINT "LAST NUMBER"
240  PRINT "STEPPING INTERVAL"
250  FOR I = 0 TO 39: PRINT  CHR$ (45);: NEXT I: PRINT " CONVERSION   POK
     E LO BYTE  POKE HI BYTE": POKE 34,6: IF SELECT < 3 THEN  POKE 34,5
260  HOME
280 CNT = 0:TB = 1:VTB = 7: IF SELECT < 3 THEN VTB = 6
290  GOSUB 800
300  ON SELECT GOSUB 3200,3200,12,12: IF SELECT < 3 THEN 290
310  VTAB 24: HTAB 5: CALL  - 868: INPUT "PRESS <RETURN> TO CONTINUE.";IN
     $: GOTO 100
450  HOME : HTAB 4: ON CHOICE GOSUB 452,456,458,455,454,457: FOR I = 0 TO
     39: PRINT  CHR$ (45);: NEXT I: POKE 34,2: RETURN
452  PRINT "HEX->FP": RETURN
454  PRINT "FP->HEX": RETURN
455  PRINT "INT->FP": RETURN
456  PRINT "HEX->INT ": RETURN
457  PRINT "FP->INT": RETURN
458  PRINT "INT->HEX": RETURN
460  HOME : VTAB 8
462  PRINT "  1. SINGLE ENTRY - MONITOR OUTPUT": PRINT
463  PRINT "  2. SINGLE ENTRY - PRINTER OUTPUT": PRINT
464  PRINT "  3. RANGE  ENTRY - MONITOR OUTPUT": PRINT
466  PRINT "  4. RANGE  ENTRY - PRINTER OUTPUT": PRINT
468  VTAB 6: INPUT "CHOOSE ONE:";IN$: IF IN$ = "Q" THEN  POP : GOTO 100
470 SELECT =  VAL (IN$)
472  IF SELECT < 1 OR SELECT > 4 THEN 460
474  RETURN
500  FOR I = 1 TO  LEN (IN$): IF  ASC ( MID$ (IN$,I,1)) > 70 OR  ASC ( MID$
     (IN$,I,1)) < 48 THEN 520
510  IF  ASC ( MID$ (IN$,I,1)) > 57 AND  ASC ( MID$ (IN$,I,1)) < 65 THEN  520
512  NEXT I: RETURN
520 FLAG = 1: RETURN
700  FOR I = 1 TO  LEN (IN$)
```

```
705  IF  ASC ( MID$ (IN$,I)) > 57 OR ASC ( MID$ (IN$,I)) < 48 THEN 710
709  NEXT I: RETURN
710 FLAG = 1: RETURN
800  IF SELECT > 2 THEN 815
805  VTAB 3: HTAB 13: CALL  - 868: GOSUB 950: IF FLAG = 1 THEN FLAG = 0: GOTO
     805
810  GOTO 835
815  VTAB 3: HTAB 13: POKE 33,21: CALL  - 868: GOSUB 950:FRST$ = IN$: POKE
     33,40: IF FLAG = 1 THEN FLAG = 0: GOTO 815
820  VTAB 3: HTAB 33: CALL  - 868: GOSUB 950:LST$ = IN$: IF FLAG = 1 THEN
     FLAG = 0: GOTO 820
825  VTAB 4: HTAB 18: CALL  - 868: GOSUB 950:STP$ = IN$: IF DVL < 0 THEN
     FLAG = 1
830  IF FLAG = 1 THEN FLAG = 0: GOTO 825
835  RETURN
950  IF CHOICE > 2 THEN 970
955  INPUT "=$";IN$: IF IN$ = "Q" THEN  POP : POP : GOTO 100
957  IF IN$ = "" THEN FLAG = 1: GOTO 995
960  IF  LEN (IN$) > 4 THEN FLAG = 1: GOTO 995
965  GOSUB 500: GOTO 995
970  INPUT "=";IN$: IF IN$ = "Q" THEN  POP : POP : GOTO 100
972  IF IN$ = "" THEN FLAG = 1: GOTO 995
975  IF CHOICE < 5 AND  VAL (IN$) <  - 32767 THEN FLAG = 1: GOTO 995
977  IF CHOICE < 5 AND  VAL (IN$) > 32767 THEN FLAG = 1: GOTO 995
980  IF CHOICE > 4 AND  VAL (IN$) < 0 THEN FLAG = 1: GOTO 995
983  IF CHOICE > 4 AND  VAL (IN$) > 65535 THEN FLAG = 1: GOTO 995
985 DVL =  VAL (IN$): IF DVL < 0 THEN IN$ =  MID$ (IN$,2): GOSUB 700:IN$ =
      STR$ (DVL + 65536): GOTO 995
990  GOSUB 700
995  RETURN
1000 HVL$ = IN$
1010 DVL = 0: FOR I = 1 TO  LEN (IN$): IF  ASC ( MID$ (IN$,I,1)) > 64 THEN
     N =  ASC ( MID$ (IN$,I,1)) - 55
1018  IF  ASC ( MID$ (IN$,I,1)) < 64 THEN N =  ASC ( MID$ (IN$,I,1)) - 48

1020 DVL = DVL + N * 16 ^ ( LEN (IN$) - I): NEXT I
1030  IF CHOICE = 1 THEN 1050
1040  IF DVL > 32767 THEN DVL = DVL - 65536
1050  RETURN
3100  FOR I = 1 TO 7
3110 J =  - 16384 + 256 * I
3120  IF  PEEK (J + 23) = 201 AND  PEEK (J + 55) = 207 AND  PEEK (J + 76)
      = 234 THEN  RETURN
3130  NEXT I
3140  HOME : VTAB 10: PRINT "NO SILENTYPE PRINTER INSTALLED.": PRINT "SEL
     ECTION ABORTED!": FOR K = 1 TO 3000: NEXT K: POP : RETURN
3200  IF CHOICE < 3 THEN  GOSUB 1000: GOSUB 92: GOSUB 62: GOTO 3230
3210  IF CHOICE = 3 OR CHOICE = 5 THEN TN =  VAL (IN$): GOSUB 42: GOSUB 9
     2: GOSUB 62: GOTO 3230
3220 HVL$ = IN$: IF CHOICE = 6 AND  VAL (IN$) > 32767 THEN HVL$ =  STR$ (
     DVL - 65536)
3225  GOSUB 92: GOSUB 62
3230  IF SELECT = 2 AND CNT = 0 THEN  GOSUB 3100: POKE  - 12526,83: PR# 1
     : PRINT : PRINT "CONVERTING FROM ";: ON CHOICE GOSUB 76,78,80,82,84,
     86:CNT = CNT + 1
3240  IF SELECT = 2 THEN  PR# 1: GOSUB 52: PR# 0
3250  RETURN
9000  POKE  - 16300,0: POKE  - 16298,0: TEXT : CALL  - 936: POKE  - 16368
     ,0: END
```

©

More VIC Maps

Jim Butterfield
Toronto, Canada

Editor's Note: For more, see Jim's VIC maps in last month's issue, **COMPUTE!** *#20. — RTM*

It's interesting to look at the innards of the VIC. In some ways, it's much like the PET/CBM and many things are quite recognizable. But new things have crept in, too: some are associated with new features such as color, others are there to implement advanced ideas such as an improved INPUT statement. Inner-space explorers will recognize many familiar landmarks.

The most noticeable new feature is the massive tables of vectors and links that have been implemented in page three. In hopes of explaining things better, I am using the terms rather carefully. Both vectors and links are addresses in RAM. An advanced application program can use these addresses, or even change them; and this gives the VIC remarkable programming flexibility. The term "Link" is used when the address is normally used to connect adjacent code; in this case, it doesn't affect the program flow until the link is broken with a new address. A vector, on the other hand, is used as a jump point, and the normal program jumps somewhere else through the vector. In other words, a ROM program hits a link point and normally keeps going; it hits a vector point and branches.

I wish Commodore had chosen to keep VIC addresses compatible with those in the PET/CBM. If they had done so, many programs would have been portable between machines with no coding changes at all. But that's wishful thinking and, since many things are still the same style, it's not a serious hardship to trim up the PEEK and POKE addresses for transfer to the VIC.

I have inserted the "normal" address contents of many of the links/vectors in the brackets behind the description; they may not be valid for current machines, but a serious user can easily PEEK them himself.

The input and output ports are somewhat congested. There are almost as many I/O bits available as on the PET/CBM, but extra features such as joysticks and RS232 have caused a bit of a crunch.

The Video Interface Chip (VIC) itself is a remarkable piece of electronics. I hope my chart helps; but a full description can only be obtained in Commodore's technical reference.

I haven't noted the standard Jump Table in this map. Near the top of both the PET and the VIC are a series of standard locations to allow inputting, outputting, checking the stop key, and other jobs. Users familiar with their use in the PET/CBM will be pleased to know that the Jump Table is exactly the same in the VIC. All of the old favorites, such as FFD2 for PRINT and FFE4 for GET are still there.

Beginners shouldn't be scared by the mass of technical detail given here. The VIC can be used effectively without any of this information. But for those who love to tinker with the innards of the machine, there's a lifetime of experimental PEEKing and POKEing to be done; this map will help direct your efforts.

VIC Zero Page Memory Map

Hex	Decimal	Description
0000-0002	0-2	USR jump
0003-0004	3-4	Float-Fixed vector
0005-0006	5-6	Fixed-Float vector
0007	7	Search character
0008	8	Scan-quotes flag
0009	9	TAB column save
000A	10	0=LOAD, 1=VERIFY
000B	11	Input buffer pointer/# subscrpt
000C	12	Default DIM flag
000D	13	Type: FF=string, 00=numeric
000E	14	Type: 80=integer, 00=floating point
000F	15	DATA scan/LIST quote/memry flag
0010	16	Subscript/FNx flag
0011	17	0=INPUT;$40=GET;$98=READ
0012	18	ATN sign/Comparison eval flag
0013	19	Current I/O prompt flag
0014-0015	20-21	Integer value

0016	22	Pointer: temporary strg stack
0017-0018	23-24	Last temp string vector
0019-0021	25-33	Stack for temporary strings
0022-0025	34-37	Utility pointer area
0026-002A	38-42	Product area for multiplication
002B-002C	43-44	Pointer: Start-of-Basic
002D-002E	45-46	Pointer: Start-of-Variables
002F-0030	47-48	Pointer: Start-of-Arrays
0031-0032	49-50	Pointer: End-of-Arrays
0033-0034	51-52	Pointer: String-storage(moving down)
0035-0036	53-54	Utility string pointer
0037-0038	55-56	Pointer: Limit-of-memory
0039-003A	57-58	Current Basic line number
003B-003C	59-60	Previous Basic line number
003D-003E	61-62	Pointer: Basic statement for CONT
003F-0040	63-64	Current DATA line number
0041-0042	65-66	Current DATA address
0043-0044	67-68	Input vector
0045-0046	69-70	Current variable name
0047-0048	71-72	Current variable address
0049-004A	73-74	Variable pointer for FOR/NEXT
004B-004C	75-76	Y-save; op-save; Basic pointer save
004D	77	Comparison symbol accumulator
004E-0053	78-83	Misc work area, pointers, etc
0054-0056	84-86	Jump vector for functions
0057-0060	87-96	Misc numeric work area
0061	97	Accum#1: Exponent
0062-0065	98-101	Accum#1: Mantissa
0066	102	Accum#1: Sign
0067	103	Series evaluation constant pointer
0068	104	Accum#1 hi-order (overflow)
0069-006E	105-110	Accum#2: Exponent, etc.
006F	111	Sign comparison, Acc#1 vs #2
0070	112	Accum#1 lo-order (rounding)
0071-0072	113-114	Cassette buff len/Series pointer
0073-008A	115-138	CHRGET subroutine; get Basic char
007A-007B	122-123	Basic pointer (within subrtn)
008B-008F	139-143	RND seed value
0090	144	Status word ST
0091	145	Keyswitch PIA: STOP and RVS flags
0092	146	Timing constant for tape
0093	147	Load=0, Verify=1
0094	148	Serial output: deferred char flag
0095	149	Serial deferred character
0096	150	Tape EOT received
0097	151	Register save
0098	152	How many open files
0099	153	Input device, normally 0
009A	154	Output CMD device, normally 3
009B	155	Tape character parity
009C	156	Byte-received flag
009D	157	Direct=$80/RUN=0 output control
009E	158	Tp Pass 1 error log/char buffer
009F	159	Tp Pass 2 err log corrected
00A0-00A2	160-162	Jiffy Clock HML
00A3	163	Serial bit count/EOI flag

00A4	164	Cycle count
00A5	165	Countdown,tape write/bit count
00A6	166	Tape buffer pointer
00A7	167	Tp Wrt ldr count/Rd pass/inbit
00A8	168	Tp Wrt new byte/Rd error/inbit cnt
00A9	169	Wrt start bit/Rd bit err/stbit
00AA	170	Tp Scan;Cnt;Ld;End/byte assy
00AB	171	Wr lead length/Rd checksum/parity
00AC-00AD	172-173	Pointer: tape bufr, scrolling
00AE-00AF	174-175	Tape end adds/End of program
00B0-00B1	176-177	Tape timing constants
00B2-00B3	178-179	Pntr: start of tape buffer
00B4	180	1=Tp timer enabled; bit cnt
00B5	181	Tp EOT/RS232 next bit to send
00B6	182	Read character error/outbyte buf
00B7	183	# characters in file name
00B8	184	Current logical file
00B9	185	Current secndy address
00BA	186	Current device
00BB-00BC	187-188	Pointer to file name
00BD	189	Wr shift word/Rd input char
00BE	190	# blocks remaining to Wr/Rd
00BF	191	Serial word buffer
00C0	192	Tape motor interlock
00C1-00C2	193-194	I/O start adds
00C3-00C4	195-196	Kernel setup pointer
00C5	197	Last key pressed
00C6	198	# chars in keybd buffer
00C7	199	Screen reverse flag
00C8	200	End-of-line for input pointer
00C9-00CA	201-202	Input cursor log (row, column)
00CB	203	Which key: 64 if no key
00CC	204	0=flash cursor
00CD	205	Cursor timing countdown
00CE	206	Character under cursor
00CF	207	Cursor in blink phase
00D0	208	Input from screen/from keyboard
00D1-00D2	209-210	Pointer to screen line
00D3	211	Position of cursor on above line
00D4	212	0=direct cursor, else programmed
00D5	213	Current screen line length
00D6	214	Row where curosr lives
00D7	215	Last inkey/checksum/buffer
00D8	216	# of INSERTs outstanding
00D9-00F0	217-240	Screen line link table
00F1	241	Dummy screen link
00F2	242	Screen row marker
00F3-00F4	243-244	Screen color pointer
00F5-00F6	245-246	Keyboard pointer
00F7-00F8	247-248	RS-232 Rcv pntr
00F9-00FA	249-250	RS-232 Tx pntr
00FF-010A	256-266	Floating to ASCII work area

[Additional VIC Maps appeared in **COMPUTE!**, *January, 1982, #20, pgs. 181-3. — Ed]*

```
FF8A-FFF5 65418-65525 Jump Table, Including:
  FFC6 - Set Input channel
  FFC9 - Set Output channel
  FFCC - Restore default I/O channels
  FFCF - INPUT
  FFD2 - PRINT
  FFE1 - Test Stop key
  FFE4 - GET
```

```
c000    ROM control vectors
c00c    Keyword action vectors
c052    Function vectors
c080    Operator vectors
c09e    Keywords
c19e    Error messages
c328    Error message vectors
c365    Miscellaneous messages
c38a    Scan stack for FOR/GOSUB
c3b8    Move memory
c3fb    Check stack depth
c408    Check memory space
c435    'OUT OF MEMORY'
c437    Error routine
c469    Break entry
c474    'READY.'
c480    Ready for Basic
c49c    Handle new line
c533    Re-chain lines
c560    Receive input line
c579    Crunch tokens
c613    Find Basic line
c642    Perform [NEW]
c65e    Perform [CLR]
c68e    Back up text pointer
c69c    Perform [LIST]
c742    Perform [FOR]
c7ed    Execute statement
c81d    Perform [RESTORE]
c82c    Break
c82f    Perform [STOP]
c831    Perform [END]
c857    Perform [CONT]
c871    Perform [RUN]
c883    Perform [GOSUB]
c8a0    Perform [GOTO]
c8d2    Perform [RETURN]
c8f8    Perform [DATA]
c906    Scan for next statement
c928    Perform [IF]
c93b    Perform [REM]
c94b    Perform [ON]
c96b    Get fixed point number
c9a5    Perform [LET]
ca80    Perform [PRINT#]
ca86    Perform [CMD]
caa0    Perform [PRINT]
cb1e    Print message from (y,a)
cb3b    Print format character
cb4d    Bad-input routines
cb7b    Perform [GET]
cba5    Perform [INPUT#]
cbbf    Perform [INPUT]
cbf9    Prompt & input
cc06    Perform [READ]
ccfc    Input error messages
cd1e    Perform [NEXT]
cd78    Type-match check
cd9e    Evaluate expression
cea8    Constant - PI
cef1    Evaluate within brackets
cef7    Check for ')'
ceff    Check for comma
cf08    Syntax error
cf14    Check range
cf28    Search for variable
cfa7    Set up FN reference
cfe6    Perform [OR]
cfe9    Perform [AND]
d016    Compare
d081    Perform [DIM]
d08b    Locate variable
d113    Check alphabetic
d11d    Create variable
d194    Array pointer subroutine
d1a5    Value 32768
d1b2    Float-fixed conversion
d1d1    Set up array
d245    'BAD SUBSCRIPT'
d248    'ILLEGAL QUANTITY'
d34c    Compute array size
d37d    Perform [FRE]
d391    Fixed-float conversion
d39e    Perform [POS]
d3a6    Check direct
d3b3    Perform [DEF]
d3e1    Check FN syntax
d3f4    Perform [FN]
d465    Perform [STR$]
d475    Calculate string vector
d487    Set up string
d4f4    Make room for string
d526    Garbage collection
d5bd    Check salvageability
```

d606	Collect string
d63d	Concatenate
d67a	Build string to memory
d6a3	Discard unwanted string
d6db	Clean descriptor stack
d6ec	Perform [CHR$]
d700	Perform [LEFT$]
d72c	Perform [RIGHT$]
d737	Perform [MID$]
d761	Pull string parameters
d77c	Perform [LEN]
d782	Exit string-mode
d78b	Perform [ASC]
d79b	Input byte parameter
d7ad	Perform [VAL]
d7eb	Get params for poke/wait
d7f7	Float-fixed
d80d	Perform [PEEK]
d824	Perform [POKE]
d82d	Perform [WAIT]
d849	Add 0.5
d850	Subtract-from
d853	Perform [SUBTRACT]
d86a	Perform [ADD]
d947	Complement fac#1
d97e	'OVERFLOW'
d983	Multiply by zero byte
d9ea	Perform [LOG]
da2b	Perform [MULTIPLY]
da59	Multiply-a-bit
da8c	Memory to FAC#2
dab7	Adjust FAC#1/#2
dad4	Underflow/overflow
dae2	Multiply by 10
daf9	+10 in floating pt
dafe	Divide by 10
db12	Perform [DIVIDE]
dba2	Memory to fac#1
dbc7	FAC#1 to memory
dbfc	FAC#2 to fac#1
dc0c	FAC#1 to FAC#2
dc1b	Round FAC#1
dc2b	Get sign
dc39	Perform [SGN]
dc58	Perform [ABS]
dc5b	Compare FAC#1 to mem
dc9b	Float-fixed
dccc	Perform [INT]
dcf3	String to fac
dd7e	Get ascii digit
dddd	Float to ascii
df16	Decimal constants
df3a	TI constants
df71	Perform [SQR]
df7b	Perform [POWER]
dfb4	Perform [NEGATIVE]

dfed	Perform [EXP]
e040	Series evaluate 1
e056	Series evaluate 2
e094	Perform [RND]
e0f6	?? Breakpoints ??
e127	Perform [SYS]
e153	Perform [SAVE]
e162	Perform [VERIFY]
e165	Perform [LOAD]
e1bb	Perform [OPEN]
e1c4	Perform [CLOSE]
e1d1	Parameters for load/save
e203	Check default parameters
e20b	Check for comma
e216	Parameters for open/close
e261	Perform [COS]
e268	Perform [SIN]
e2b1	Perform [TAN]
e30b	Perform [ATN]
e378	Initialize
e387	CHRGET for zero page
e3a4	Initialize Basic
e429	Power-up message
e44f	Vectors for $300
e45b	Initialize vectors
e467	Warm restart
e476	Program patch area
e4a0	Serial output '1'
e4a9	Serial output '0'
e4b2	Get serial input & clock
e4bc	Program patch area
e500	Set 6522 addrs
e505	Set screen limits
e50a	Track cursor location
e518	Initalize I/O
e54c	Normalize screen
e55f	Clear screen
e581	Home cursor
e587	Set screen pointers
e5bb	Set I/o defaults
e5c3	Set vic chip defaults
e5cf	Input from keyboard
e64f	Input from screen
e6b8	Quote mark test
e6c5	Set up screen print
e6ea	Advance cursor
e715	Retreat cursor
e72d	Back into previous line
e742	Output to screen
e8c3	Go to next line
e8d8	Do 'RETURN'
e8e8	Check line decrement
e8fa	Check line increment
e912	Set colour code
e921	Colour code table
e929	Code conversion

e975	Scroll screen
e9ee	Open space on screen
ea56	Move screen line
ea6e	Synch colour transfer
ea7e	Set start-of-line
ea8d	Clear screen line
eaa1	Print to screen
eaaa	Store on screen
eab2	Synch colour to char
eabf	Interrupt (IRQ)
eb1e	Check keyboard
ec00	Set text mode
ec46	Keyboard vectors
ec5e	Keyboard maps
ed21	Graphics/text control
ed30	Set graphics mode
ed5b	Wrap up screen line
ed6a	Shifted key matrix
eda3	Control key matrix
ede4	Vic chip defaults
edfd	Screen line adds low
ee14	Send 'talk'
ee17	Send 'listen'
ee1c	Send control char
ee49	Send to serial bus
eeb7	Timeout on serial
eec0	Send listen SA
eec5	Clear ATN
eece	Send talk SA
eee4	Send serial deferred
eef6	Send 'untalk'
ef04	Send 'unlisten'
ef19	Receive from serial bus
ef84	Clock line on
ef8d	Clock line off
ef96	Delay 1 ms
efa3	RS232 send (NMI)
efee	New RS232 byte send
f016	Error or quit
f027	Compute bit count
f036	RS232 receive (NMI)
f05b	Setup to receive
f09d	Receive parity error
f0a2	Receive overrun error
f0a5	Receive break error
f0a8	Receive frame error
f0b9	Bad device
f0bc	File to RS232
f0ed	Send to RS232 buffer
f116	Input from RS232 buffer
f14f	Get from RS232 buffer
f160	Check serial bus idle
f174	Messages
f1e2	Print if direct
f1f5	Get..
f205	..from RS232
f20e	Input
f250	Get.. tape/serial/RS232
f27a	Output..
f290	..to tape
f2c7	Set input device
f309	Set output device
f34a	Close
f3cf	Find file
f3df	Set file values
f3ef	Abort all files
f3f3	Restore default I/O
f40a	Do file opening
f495	Send SA
f4c7	Open RS232
f542	Load program
f647	'SEARCHING'
f659	Print file name
f66a	'LOADING/VERIFYING'
f675	Save program
f728	'SAVING'
f734	Bump clock
f760	Get time
f767	Set time
f770	Action stop key
f77e	File Error Messages
f7af	Find any tape header
f7e7	Write tape header
f84d	Get buffer address
f854	Set buffer start, end pointers
f867	Find specific header
f88a	Bump tape pointer
f894	'PRESS PLAY .. '
f8ab	Check cassette status
f8b7	'PRESS RECORD ..'
f8c0	Initiate tape read
f8e3	Initiate tape write
f8f4	Common tape read/write
f94b	Check tape stop
f95d	Set timing
f98e	Read bits (IRQ)
faad	Store characters
fbd2	Reset pointer
fbdb	New tape character setup
fbea	Toggle tape
fc06	Data write
fc0b	Tape write (IRQ)
fc95	Leader write (IRQ)
fccf	Restore vectors
fcf6	Set vector
fd08	Kill motor
fd11	Check read/write pointer
fd1b	Bump read/write pointer
fd22	Powerup entry
fd3f	Check A-rom
fd52	Set kernal2

fd8d	Initialize system constants
fdf1	IRQ vectors
fdf9	Initialize I/O regs
fe49	Save data name
fe50	Save file details
fe57	Get status
fe66	Flag ST
fe6f	Set timeout
fe73	Read/set top of memory
fe82	Read/set bottom of memory
fe91	Test memory location
fea9	NMI interrupt entry
fed2	RESET/STOP warm start
fede	NMI RS232 sequences
ff56	Restore & exit
ff5c	RS232 timing table
ff72	Main IRQ entry
ff8a	Jumbo jump table
fffa	Hardware vectors

©

EPROM Reliability

Michael E. Day
West Linn, OR

Although EPROMs are in widespread use, there are continuing problems with the use of the device affecting their overall reliability.

The following report describes how to obtain the maximum performance and reliability from the 2708 EPROM. The concepts involved, however, may be applied to most of the ultra-violet erasable PROMs on the market to date.

The EPROM 'cell' consists, basically, of a capacitor which either has a charge on it or does not. The charge is created by applying a high voltage pulse to the device, and is removed by exposing the device to high intensity ultra-violet light.

The cell is programmed by injection of high energy electrons through the oxide onto the floating gate. Once there, the charge is trapped, as there are no electrical connections to this floating gate. This action is similar to the action of a zener diode in that, as the voltage increases, it finally passes a point where it can overcome the barrier presented by the silicon oxide surrounding the gate and allows the electrons to flow to the gate and collect there. As the voltage is removed it finally drops to a point where it can no longer maintain the bridge through the oxide, and it again becomes isolated. However, the gate now has a charge of electrons on it.

The charge is removed from the cell by exposure with ultra-violet light of the correct wave length (2537A) and energy (10 watt seconds/cm^2) which will impart sufficient photon energy to the trapped electrons to allow the floating gate to be fully discharged.

The presence of charge on the floating gate causes a shift of the cell threshold. In the discharged state (no charge on the floating gate) the cell has a low threshold, and selection of the cell turns on the transistor. Storing a charge on the gate shifts the threshold of the cell above the select voltage so that the transistor will not turn on when it is selected. The amount of charge on the gate determines the level of select voltage at which the transistor will change from a non-conducting to a conducting state. The cell is designed so that the discharged threshold and charged threshold are equally above and below the select voltage. This provides for maximum immunity against marginal cells.

Data retention can be measured by baking the device at an elevated temperature (250°C). 168 hours at this temperature is equivalent to 10 years at 70°C. Test samplings have shown that the time to 5% batch failure is 100 years.

Experiments have been made to determine the effects of prolonged exposure to UV light. Through the first 20 hours the threshold voltage increased slightly after which it stabilized out to 30 days at which time the test was terminated. Although no study has been made to determine what is causing the initial change, it is thought to be caused by some radiation damage caused by the UV.

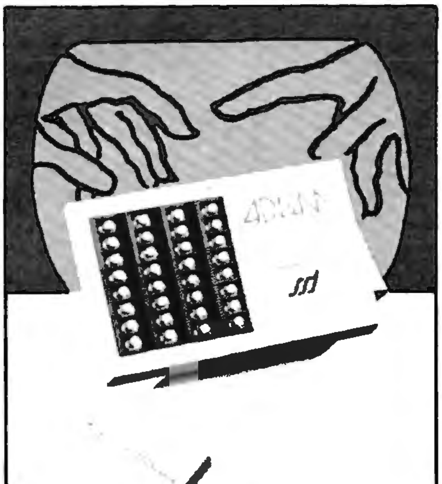
It is believed that UV lamps with short wavelengths (less than 1800A) and high intensity can ionize oxide with long exposure. The theory is that this will shift the threshold until the part will not function properly. This is not a permanent shift and a bake at 150° for 24 hours should correct the problem.

Some EPROMs exhibit a sensitivity to ambient light. This does not erase them, but they may not function properly. This is a common phenomenon with most semiconductors. Covering the lid with some sort of opaque material will prevent this.

For a given device, given that the programming equipment is operating at factory specifications, the failure to take a charge is device-related, and attempts to bring the charge level higher by reprogramming will seldom be successful. Failure to erase is the most common problem. There are many factors which can cause inadequate erasure; among them are weak UV lamp due to age, dirt on the IC (both internal and external), dirt on the UV lamp, erase requirements outside of normal specifications, or a defective component.

The EPROM is read by determining if the charge on the capacitor of the cell is above or below the threshold of the sensing transistor (the threshold being that level of applied voltage which causes the transistor to change from a non-conducting state to a conducting state). This threshold can be affected by shifts in the -12 volt and -5 volt supplies at the device and temperature. Due to this, if the charge on the cell is near the threshold of the sensing transistor, a shift in the supply voltage or temperature can cause the cell to appear to change state, have an excessive access time, or be intermittent. A cell which is sufficiently near the threshold of the sensing transistor so that it can be affected by temperature or voltage shifts is called "marginally programmed" or "marginal."

One failure of the EPROM is a "leaky cell" (a cell that loses its charge after a short period of time). A leaky cell can be found several ways. One way is to bake the device at 250° after programming it, and then test for lost data.



monitor analog data quickly and inexpensively

ADAM

provides analog data monitoring for your micro-computer. With proper transducers it allows you to sense temperature, wind direction, electrical current, or joysticks and convert the sensors analog voltages into digital readings your micro-computer can process.

SPECS

8 bit resolution A/D
0-5 volt input range
16 Analog Input channels

MODELS

ADAM with TRS-80 interface $190.00
ADAM with R5-232/RS-423 interface $250.00
ADAM with RS-422 interface $275.00

Dealer inquiries welcomed.

small system design

P. O. BOX 4546 • MANCHESTER, NH 03108

Representatives Pacific Northwest Data Systems Marketing (503) 297-8444
New England American Data Systems Marketing (617) 341-0171

Oryx software

What does the Oryx, an African antelope, have in common with reliable service? They are both almost extinct.
Oryx Software wants to bring trustworthy service back to life. We're here to stay!

GUARANTEED LOWEST PRICES!
To celebrate our opening, we are matching any advertised price. Just show us the ad.

ORDER TOLL FREE - Outside WI
1-800-826-1589

SOFTWARE

Graphpower	299.00
Tax Preparer '82	127.00
Real Estate Analyzer	127.00
Creative Financing	127.00
Word Processor	425.00
Financial Projection	550.00
Mail Management	250.00
Micropro Wordstar	248.00
Mailmerge	90.00
Spellstar	169.00
Supersort II	159.00
Context Connector	180.00
Easywriter (Pro)	195.00
Easymover	45.00
Tellstar II	70.00
Easymailer	160.00
Datadex	249.00
Microplan Basic	419.00
Executive Secretary	212.00
Nevada CoBol	129.00

ACCESSORIES

Apple II Dual Thermometer w/software	260.00
Mountain Expansion Chassis	625.00
CPS Multifunction	199.00
Mountain Music System	395.00
100,000 Day Clock	375.00
Mountain Clock	252.00
Supertalker SD 2000	199.00
A/D + D/A	299.00
CCS 12K ROM/PROM Board	99.00
CCS A/D Converter Board	99.00
CCS Serial Asynch. Board	139.00
CCS Calendar/Clock Board	99.00
VOTRAX Type-n-Talk	375.00
ADALAB Package	495.00
DOS Dater	49.00
ABT Numeric Keypad	125.00
ABT Softkey	150.00
ABT Barwand	195.00
ABT Cash Box	295.00

. . . AND MANY MORE!

Call for price information:
Printers - Epson, Okidata, Microline, NEC, C. ITOH, Qume, Diablo, Anadex, IDS, MPI, Microtek, Printek.
Monitors - Amdek, Sanyo, Hitachi, NEC, Kaga/TSK.
Modems - Novation, Hayes.
Disk Drives - Percom, Corvus, Morrow, CCS, etc.
Plotters - Watanabe.

Quoted prices valid for stock on hand and subject to change without prior notice.
Ask for our FREE catalog.

Please:
- Wisconsin residents - add 4% sales tax
- Foreign orders - please add 15%
- Add $2.50 for shipping per software and small item. Call regarding others.

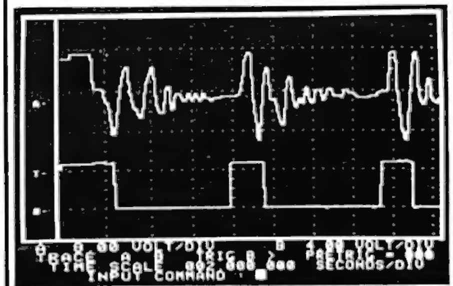
We Welcome:
- Visa, MasterCharge - (Add 4%)
- Checks (Allow 1-2 weeks for clearing)
- COD (Add 4%)

TOLL FREE - Outside Wisconsin:
1-800-826-1589
For Technical Info & in Wisconsin:
715-848-2322
Oryx Software
P.O. Box 1961 • Wausau, WI 54401

APPLESCOPE

DIGITAL STORAGE OSCILLOSCOPE

Interface for the Apple II Computer



The APPLESCOPE system combines two high speed analog to digital converters and a digital control board with the high resolution graphics capabilities of the Apple II computer to create a digital storage oscillosocpe. Signal trace parameters are entered through the keyboard to operational software provided in PROM on the DI control board.

- DC to 3.5 Mhz sample rate with 1024 byte buffer memory
- Pretrigger Viewing
- Programmable Scale Select
- Continuous and Single Sweep Modes
- Single or Dual Channel Trace

Price for the two board Applescope system is **$595**
EXTERNAL TRIGGER ADDAPTER **$29**

APPLESCOPE ACCESSORIES

APPLESCOPE-HR12 High resolution 12 bit analog to digital converter with sample rates to 100 Khz. Requires 48K Apple II with disk drive. Software provide on floppy disk includes basic SCOPE DRIVER package. Price per channel **$695**

APPLESCOPE-HRHS High Resolution AND High Speed. Circuit combines two 6 bit flash analog to digital converters to give a 10 bit dynamic range. The 10 bit converter resolution is maintained at sampling rates up to the 7 Mhz. maximum for signal slew rates less than .5 volts per microsecond. Larger inputs slew rates will reduce the converter resolution to 6 bits until the signal stabilizes within the .5 Volt per microsecond limit. Requires 48K Apple II with disk drive. Software provided on disk includes the basic SCOPE DRIVER package Price per channel **$695**

APPLESCOPE-EXT External trigger adapter has a switch selectable external trigger input to a BNC connector mounted in a rear slot of the Apple II computer. Price **$29.00**

APPLESCOPE-BNC BNC adapter connects the Berg stick connectors on the A1 circuit card to male BNC plugs mounted in a rear slot of the Apple II compter. Price **$14.95**

BUS EXTENDERS Allow easy access to Apple II peripheral circuit cards. Price **$19.95**

SCOPE PROBES Oscilloscope probes for use with the APPLESCOPE - BNC adapter Price**$29.95**

SCOPE DRIVER Advanced software for the Applescope system provided on 5¼" floppy disk. Available options include:

- Signal Averaging - Acquires 1 to 255 signal sweeps and displays the averaged result.
- Digital Volt Meter - Allows use as a real time DVM or use to measure points on an acquired sweep.
- Hard Copy - Uses graphics printer to produce hard-copy output of displayed traces.
- Disk Storage - Allows automatic storage and recover of acquired data on floppy disks
- Spectrum Analyzer - Calculates and displays frequency spectrum of acquired data

BUS RIDER
LOGIC ANALYZER for the APPLE II

The BUS RIDER circuit card silently rides the Apple II peripheral bus and allows real time tracking of program flow. Software provided on EPROM allows set up of trace parameters from the keyboard and read back of disassembled code after a program has been tracked

- 32 bit by 1024 sample memory buffer
- Monitors Data and Address bus plus 8 external inputs
- Trigger on any 32 bit word or external trigger
- Pretrigger viewing

The BUS RIDER is an invaluable development tool for anyone working with Apple II or Apple II+ computers. Price **$295**

RC ELECTRONICS INC.
7265 Tuolumne St., Goleta, CA 93117
Visa **(805) 968-6614** M/C

Another method of testing for leaky cells is to make an erase profile for the suspect EPROM. This is done by programming the device, and then erasing it in one to two minute increments, measuring the number of erased bits after each increment. Making a graph with this information will give you a profile of the erasure characteristics of the EPROM. Any cell that erases twice as fast as the overall average should be cosidered suspect.

Another failure mode of the EPROM is the "sticky cell" (a cell which is difficult to program or erase). Although a sticky cell can be overcome by a longer program or erase time, in a production environment it is not acceptable to adjust these times for each device. Therefore, any device which requires more than three times the normal time to program or erase should be considered defective.

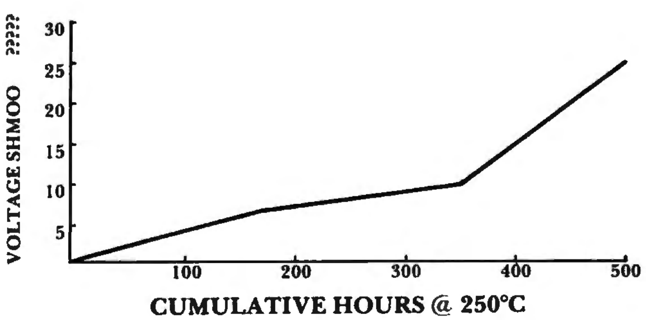
The major source of problems with the 2708 EPROM is inadequate erasure. In testing the EPROM to determine if it has been adequately programmed or erased, it is not acceptable to simply read the PROM and compare the information read against the true data, since marginal cells may not be found with this method. A more reliable method of verifying if an EPROM has been properly programmed or erased is to measure the depth of the charge at each cell. This can be done by shifting the threshold level of the sensing transistor above and below the normal level and by doing a normal read and compare.

In this way, a map of the charge level of the cells in the EPROM can be generated by observing the level at which the output changes state.

The threshold level of the 2708 EPROM can be shifted by adjusting the -5 volt supply (VBB). Causing the -5 volts to go more negative will determine how deep the cell has been charged; bringing it more positive will determine how much it has been erased.

The charge limits will vary greatly not only from manufacturer to manufacturer, but from device to device. Therefore, an acceptable limit must be determined at which the device may be considered good or bad. For the 2708 this is greater than twice the tolerance for the -5 volt supply. This can be simply generated by using the forward voltage drop across the diode (.7 volts) above and below the -5 volt level. In more critical applications a two-diode level drop (1.4 volts) might be considered.

More is not always better. Just because the charge on one device is deeper than on another does not mean that it will retain the charge longer. Data retention is related to cell isolation and not necessarily to the level of the charge.



TEMPERATURE	FAILURE RATE 60% CONFIDENCE (% / 1000 hours)	FAILURE RATE 90% CONFIDENCE (% / 1000 hours)
70°C	0.013	0.027
55°C	0.006	0.013

Operating Life Test Results

TEMPERATURE	SAMPLE SIZE	HOURS	EQUIVALENT DEVICE HOURS @ 70°C	FAILURES	FAILURE MODE
160°C	64	2243	39.9×10^6	1	Charge Loss
160°C	49	2028	27.6×10^6	0	
160°C	51	2028	28.7×10^6	1	Charge Loss
160°C	40	2830	31.4×10^6	2	Charge Loss
160°C	80	1176	26.1×10^6	1	Charge Loss
160°C	77	1176	25.1×10^6	4	Charge Loss
160°C	79	984	21.6×10^6	1	Charge Loss

DOS FOR AIM-65®: $499.00 COMPLETE!

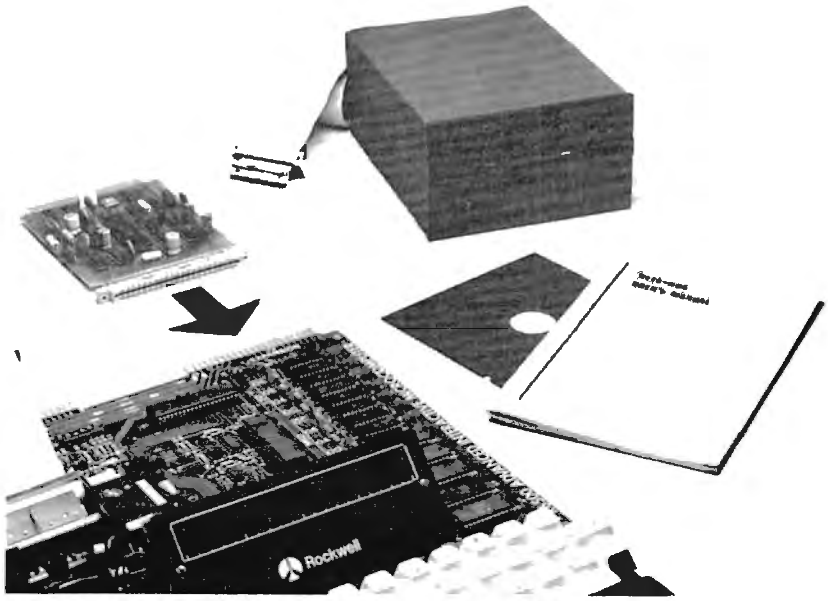
INSTANT EASY ACCESS TO YOUR AIM-65 PROGRAMS!
MANIPULATE OBJECT FILES AT THE TOUCH OF A KEY.
NEW BYTE-DOS DISC OPERATING SYSTEM.
153K BYTES ON-LINE. INEXPENSIVELY!

BYTE–DOS COMMANDS

- **FORMAT** - Prepares New Disc
- **BAD SECTOR SCAN** - QC's Disc
- **DUMP** - RAM File to Disc
- **LOAD** - Disc File to Ram
- **LIST** - Prints Disc File Names
- **ERASE** - Eliminate a File
- **UNERASE** - File Recovery
- **RENAME** - Rename a File
- **COPY** - Copy 1 File
- **TRANSFER** - Copy Complete Disc
- **KOMPRESS** - Optimizes Disc Space

AND MORE!

IT'S EASY AND FUN TO USE!



master charge
VISA

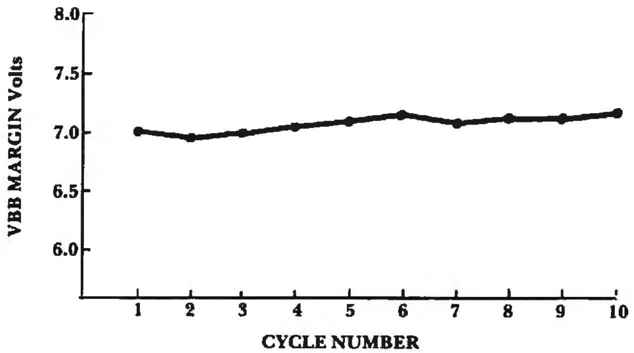
BYTE MICROSYSTEMS
CORPORATION
1477 ELKA AVENUE, SAN JOSE, CA 95129 408/446/0559

- *BYTE-DOS OPERATES WITH THE AIM-65 MONITOR, BASIC, ASSEMBLER, EDITOR, PL/65 AND FORTH. ACCESS IS THROUGH PROGRAM I/O AND F1-F3 KEYS. BYTE-DOS SUPPORTS ONE OR TWO 5¼" OR 8" FLOPPYS!*
- *BYTE-DOS $499 SYSTEM INCLUDES DISC CONTROLLER CARD, TEAC FD-50A DISC DRIVE (153K FORMATTED BYTES), BYTE-DOS SOFTWARE SYSTEM, CABLES, CONNECTORS, MANUALS. COMPLETE! JUST PLUG IT TOGETHER AND USE!*

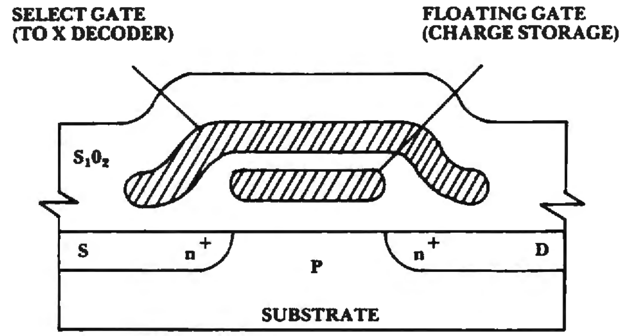
SEND ME THE FULL STORY!

Name ______________________
Company ______________________
Address ______________________
City __________ State _____ Zip _____
Telephone __________ ☐ home ☐ work

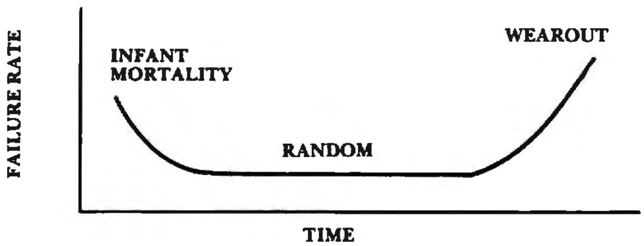
☐ I own an AIM-65.
☐ I'm thinking of buying an AIM-65.
☐ Send full BYTE-DOS Data.



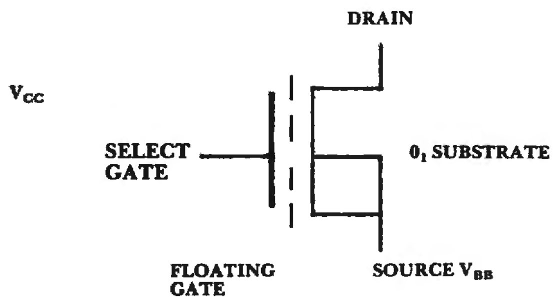
Multiple Program Erase Experiment



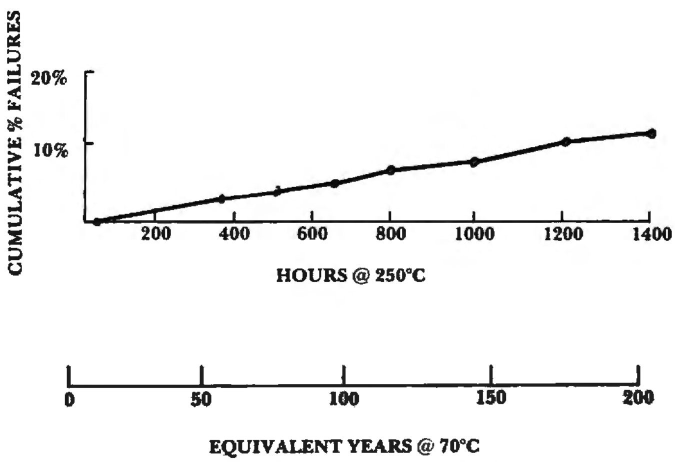
(a) CROSS SECTION



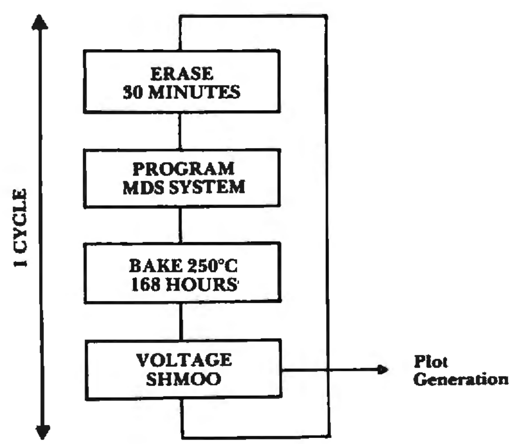
Reliability Life Curve



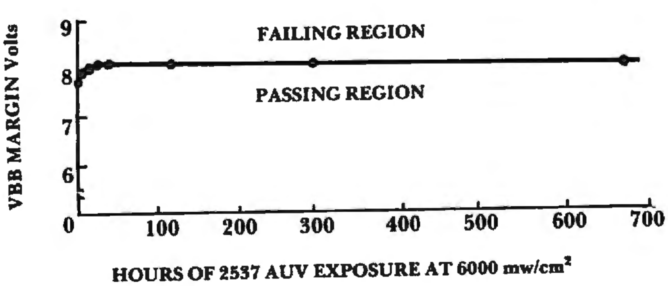
(b) SCHEMATIC SYMBOL



Intel® 250°C Bake Failure Rate



Program Erase Cycles



Random Music Composition On The PET

Alfred J. Bruey
Jackson, MI

This program, MUSICOMP, lets the PET computer compose and play music. MUSICOMP was written to provide the user with an introduction to computer generated music. The music is output using the CB2 method of music generation which is described many places in PET literature. Attachments A and B give descriptions of the hardware that you can use if you don't already have CB2 sound. Figure 1 shows the connections necessary to output sounds from the PET to an audio amplifier. Figure 2 shows a simple audio amplifier that you can make if you don't have one.

Program Description

MUSICOMP generates three kinds of music: white music, brown music, and 1/f music. For a complete description of these three types of music, see Martin Gardener's Mathematical Games column in the April, 1978, issue of *Scientific American* magazine.

A. White music: White music is a sequence of completely random sounds. In this program, you have your choice of two different types of white music:

1. Option 1 on the menu allows any of 256 different frequencies to be generated. The notes are not correlated with each other in any way. It is unlikely that you will want to go away humming the tunes you generate using this option.

2. Option 2 also generates random sounds, but these sounds are restricted to: the 25 piano notes (well-tempered scale) beginning with the B below middle C.

B. Brown Music: The second type of music is called brown music (Option 3). It is similar to the Brownian motion of particles. In brown music, each note can vary by only one tone (half-step) from the preceding note. The only randomness is in choosing the starting note and in determining whether each note is one tone higher or lower than its predecessor. You will probably find this music boring. It sounds something like a finger exercise for a violinist.

To get brown music, enter a 1 when you are asked for the maximum variation. Entering some other number, a 3 for example, will allow each note to vary three tones from its predecessor. True brown music allows only a one tone variation from note to note. The option of choosing a maximum variation is given so you can experiment with sounds.

C. 1/f Music: The final type of random music in this program is 1/f music. This music is somewhere between the randomness of white music and the boring regularity of brown music. 1/f music was discovered by an investigator who was trying to find music in nature. The algorithm used in this program is the same as the one described in the previously mentioned article except five different colored dice were used instead of three so that tunes 32 notes long could be created. Most listeners agree that 1/f music is much more musical than either white or brown music.

Extensions

I assume that anyone who knows BASIC and a minimum of music will want to change this program. That's why an annotated listing of the program is provided.

You might want to add options which impose different rules on the composition. You might also want to add the coding to save the composition on tape or disk. The place where you might do this is marked in the listing.

Using The Program

Load the program in the usual way. The main menu will be displayed on the screen as follows. Press the proper key from 1 to 5 to make your selection, but do not press RETURN. (If you press RETURN accidently and get the READY signal, type CONT and press RETURN and you'll be right back where you left off.)

```
COMPOSITION SELECTION
1 RANDOM TUNE
2 RANDOM TUNE, WELL-TEMPERED
3 RANDOM TUNE, WELL-TEMPERED
  WITH STEP SIZE LIMIT
4 1/F MUSIC
5 END PROGRAM
```

A brief description of each of the options follows:

Option 1: Random notes – This option will compose and play tunes based on 256 different tones, ranging from a tone slightly below the B below middle C to a tone that's probably even too high for your dog

SM-KIT

for Commodore Computers

A Programming Productivity Tool

The SM-KIT is a collection of machine language firmware programming and test aids for BASIC programmers. SM-KIT is a 4K ROM (twice the normal capacity) which you simply insert in a single ROM socket on any BASIC 4 CBM/PET–either 80 column or 40 column. Includes both programming aids and disk handling commands.

ERROR DETECTION: the SM-KIT automatically indicates the erroneous line and statement for any BASIC program error.
LINE NUMBERING. the SM-KIT automatically numbers BASIC statements until you turn the function off.
SCREEN OUTPUT. the commands FIND, DUMP, TRACE and DIRECTORY display on the CRT while you hold the RETURN key (display pauses when the key is released). Continuous output is selected with shift-lock.
OUTPUT CONTROL to DISK or PRINTER: in addition to displaying on the CRT, you can direct output to either disk or printer.
HARDCOPY allows screen displays to be either printed or stored on disk.
FIND. searches all or any part of a program for text or command strings or variable names. Either exact search or wild card search supported.
RENUMBER: the SM-KIT can renumber all or any part of a program. The selective renumbering allows you to move blocks of code within your program.
VARIABLE DUMP displays the contents of floating point, integer, and string variables (both simple and array). Can display all variables or any selected variables.
TRACE: SM-KIT can trace program execution either continuously or step by step starting with any line number. Selected program variables can be displayed while tracing.
DISK COMMANDS. as in DOS Support (Universal Wedge), the "shorthand" versions of disk commands may be used for displaying disk directory, initializing, copying, scratching files, load and run, etc.
LOAD. SM-KIT can load all or part of BASIC or machine language programs. It can append to a program in memory, overwrite any part of a program, load starting with any absolute memory location, and load without changing variable pointers.
MERGE. allows merging all or any part of a program on disk with a program in memory.
SAVE and VERIFY: SM-KIT provides one step program save and verification. It also allows you to save any part of a program, or any address range.

ONLY $40

A 4K ROM with both programming and disk handling aids.

Developed by (and available in Europe from) SM Softwareverbund-Microcomputer GmbH, Scherbaumstrasse 29, 8000 Munchen 83, Germany

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

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

---

NEW PRODUCT

SWARM-100

For 24 Pin ROM Machines Only.

JUST PLUG IT IN

- No soldering • No messy wires

SOFTWARE SELECTABLE

1. Software select one of two operating systems. (BASIC 2.0 / BASIC 4.0)
2. Software select utility ROMs at conflicting addresses.

$150.00

Master Charge and VISA accepted.

BATTERIES INCLUDED

Village by the Grange
71 McCaul Street
Toronto, Ontario
Canada M5T 2X1
(416) 596-1405

to hear. When you press the 1 key, a series of questions will be displayed (Press RETURN after each answer):

HOW LONG IS THE TUNE
(Answer with a number from 1 to 150)

DIFFERENT LENGTH NOTES (Y OR N)
(If you enter a Y, each note length will be one second long. If you answer M it will be ½ second long. If you answer F it will be ¼ second long. All other note lengths will be scaled accordingly.)

REPEAT NOTES (Y OR N)
(If you reply N, the tune will play one time and then the main menu will reappear. If you reply Y, the tune will repeat. In either case, you can stop the tune while it is playing by holding down the X key. You will return to the main menu.)

After you have answered these four questions, there will be a short pause while time values are being calculated for all the notes. Then the tune will begin to play.

Option 2: Random notes, well-tempered. This is the same as Option 1 except that all notes are chosen randomly from one of 25 tones. These tones are the 25 piano notes beginning with the B below middle C.

Option 3: Random notes, well-tempered, with step-size limit. You will be asked the same questions as in Options 1 and 2. After you answer them, you will receive an additional question:

MAX. VAR. FROM LAST NOTE

This question is asking you for the maximum variation in tone (half-steps) that are permissible from one note to the next. If you reply 1, you will get brown music. You may enter any other value just to see what kind of tune the PET will compose.

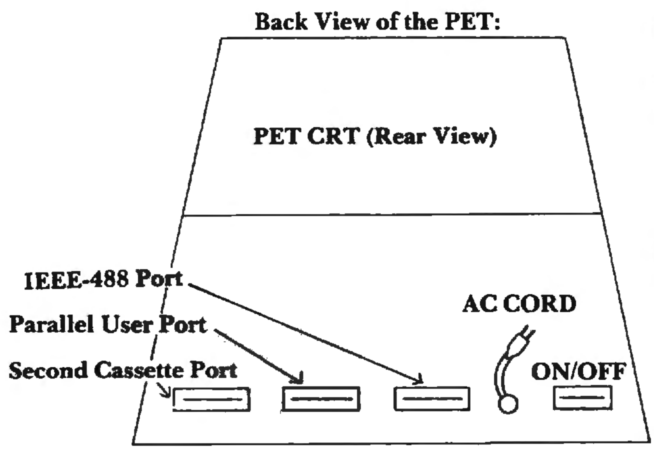
Option 4: 1/f Music. Pressing the 4 key will generate 1/f music. The 1/f tunes will all be 32 notes long, so you will not be asked for the length of the tune. Otherwise, you will be asked the same questions as in Options 1 and 2.

Option 5: End Program. Select Option 5 when you are ready to quit.

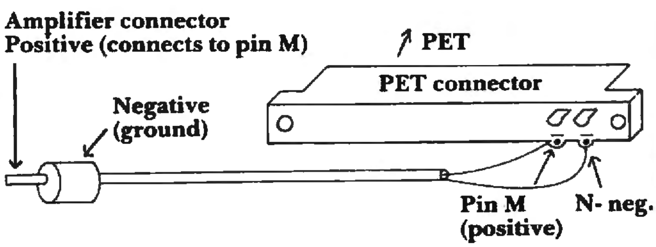
A Circuit For A PET Amplifier

Below is a circuit for a PET amplifier for making music or adding sound effects to your games. Use an RCA phono jack as the input and you'll be able to use the same connector cable as described previously.

Figure 1.

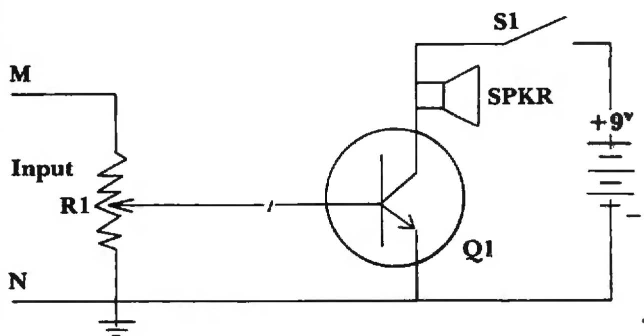


The edge connector that you need plugs into the Parallel user port of the PET. Do not attach it to the IEEE-488 port. (It's not a bad idea to put a strip of masking tape across the IEEE port so you don't accidently plug into it.) Here's what the completed cable should look like. The amplifier end might look different if your system doesn't use the RCA type jack.



You should use shielded cable for the line between the PET and the amplifier. *Be sure you don't put the PET connector on upside down!*

Figure 2.



R1 – 100K potentiometer with switch
S1 – Part of R1
Q1 – RS2031 (Radio Shack 276-2031)
SPKR – 8 ohm SPEAKER
SMALL PLASTIC BOX, 9 VOLT BATTERY, BATTERYHOLDER

Substitutions may be made for Q1, but a high-gain transistor should be used to be sure of sufficient volume. The transistor can be mounted directly on the leads of the potentiometer.

Program 1.

```
150 DIM SN(150),ST(150),PI(25),PN(1
    50)
160 FORI=1TO25:READPI(I):NEXTI
170 DATA251,237,223,211,199,188,177
    ,167,157,148,140,132,125,1
    17,111,104
180 DATA98,93,87,82,78,73,69,65,61
190 REM ****************************
    ****
200 REM VARIABLE LIST:    *
210 REM T=TIME OF NOTE IN 60THS OF
    SECOND
220 REM P=POKE NUMBER FOR NOTE
230 REM TY$=TYPE OF SONG
240 REM     1=RANDOM
250 REM     2=RANDOM, WELL-TEMPERED
260 REM     3=RANDOM, WELL-TEMPERED,
     LIMIT ON STEP SIZE
270 REM     4=1/F MUSIC
280 REM     5=STOP
290 REM L%=LENGTH OF SONG
300 REM L$="Y" NOTES DIFFERENT LENG
    TH
310 REM     "N" NOTES SAME LENGTH
320 REM S$="S" SLOW SONG,S=1
330 REM     "M" MEDIUM SPEED SONG,S=
    2
340 REM     "F" FAST SONG,S=4
350 REM ***************************
    ***
360 PRINT"{CLEAR}{03 RIGHT}{03 DOWN
    DOWN}{REV}COMPOSITION SELE
    CTION"
370 PRINT"{DOWN}{04 RIGHT}{REV}1{OF
    OFF} RANDOM TUNE
380 PRINT"{DOWN}{04 RIGHT}{REV}2{OF
    OFF} RANDOM TUNE, WELL-TEM
    PERED"
390 PRINT"{DOWN}{04 RIGHT}{REV}3{OF
    OFF} RANDOM TUNE, WELL-TEM
    PERED
400 PRINT"{DOWN}{04 RIGHT}      WITH
    STEP SIZE LIMIT"
410 PRINT"{DOWN}{04 RIGHT}{REV}4{OF
    OFF} 1/F MUSIC
420 PRINT"{DOWN}{04 RIGHT}{REV}5{OF
    OFF} END PROGRAM
430 GET TY$:IFTY$=""THEN430
440 ONVAL(TY$)GOTO500,590,690,980,4
    60
450 GOTO430
460 REM ****************************
    ***
470 REM EXIT ROUTINE **************
    ***
480 REM ****************************
    ***
490 PRINT"{CLEAR}{03 RIGHT}{04 DOWN
    DOWN}{REV}ROUTINE ENDED":E
    ND
500 REM ********************
510 REM PLAY RANDOM ********
520 REM ********************
530 GOSUB 1190 :REM GET SONG DATA
540 FORI=1TOL%
550 SN(I)=INT(RND(3)*255+1)
560 NEXTI
570 GOSUB1410:REM GENERATE NOTES AN
    D PLAY
580 GOTO360
590 REM*****************************
    ***
600 REM RANDOM, WELL-TEMPERED *****
    ***
610 REM ****************************
    ***
620 GOSUB 1190 :REM GET SONG DATA
630 FORI=1TOL%
640 SN(I)=INT(RND(5)*25+1)
650 SN(I)=PI(SN(I))
660 NEXTI
670 GOSUB 1410
680 GOTO360
690 REM ****************************
    ***
700 REM RANDOM,WELL-TEMP,STEP-SIZE
    ****
710 REM ****************************
    ***
720 GOSUB 1190 :REM GET SONG DATA
730 SN(1)=INT(RND(6)*25+1):PN(1)=PI
    (SN(1))
740 IFMV>1THEN850
750 REM BROWNIAN MOVEMENT
760 FORI=2TOL%
770 IFSN(I-1)=1THENSN(I)=2:PN(I)=PI
    (2):GOTO830
780 IFSN(I-1)=25THENSN(I)=24:PN(I)=
    PI(24):GOTO830
790 KR=RND(7)
800 IFKR< .5THENSN(I)=SN(I-1)+1
810 IFKR>=.5THENSN(I)=SN(I-1)-1
```

HOME FINANCES
Commodore PET™

These programs were developed on a Commodore 2001 series, thirty-two kilobyte (32K required) personal electronic transactor with a single tape cassette.

Will categorize and review entries, give totals & tallies for all past-entered data (not a ledger; no dates or specific item names).

Great for use every 2-6 months in conjunction with check book records, stubs, bills, or your cancelled checks.

(a) Personal cash flow & tax deduction-related; 24 specific to general categories: "tax accounts" $30
or "TAX ACCOUNTS" (No shift)

(b) Forty categories mostly specific for common home expenditures: "house accounts" $50
or "HOUSE ACCOUNTS" (No shift)

(c) Both above programs; very slight overlap: $70

Choice desired . . . ________

Mail, with check or other payment and your address, to:
PractEd Tapes, Inc.
12162 S.E. 14th St.
Bellevue, Washington 98005
(206)747-8485

Programs not guaranteed against loss misuse or system difficulties

---

CBM/PET? SEE SKYLES ... CBM/PET?

"Should we call it Command-O or Command-O-Pro?"

That's a problem because this popular ROM is called the Command-O-Pro in Europe. (Maybe Command-O smacks too much of the military.)

But whatever you call it, this 4K byte ROM will provide your CBM BASIC 4.0 (4016, 4032) and 8032 computers with 20 additional commands including 10 Toolkit program editing and debugging commands and 10 additional commands for screening, formatting and disc file manipulating. (And our manual writer dug up 39 additional commands in the course of doing a 78-page manual!)

The Command-O extends Commodore's 8032 advanced screen editing features to the ultimate. You can now SCROLL up and down, insert or delete entire lines, delete the characters to the left or right of the cursor, select TEXT or GRAPHICS modes or ring the 8032 bell. You can even redefine the window to adjust it by size and position on your screen. And you can define any key to equal a sequence of up to 90 key strokes.

The Command-O chip resides in hexadecimal address $9000, the rightmost empty socket in 4016 and 4032 or the rearmost in 8032. If there is a space conflict, we do have Socket-2-ME available at a very special price.

Skyles guarantees your satisfaction: if you are not absolutely happy with your new Command-O, return it to us within ten days for an immediate, full refund.

Command-O from Skyles Electric Works..$75.00
Complete with **Socket-2-Me**.. 95.00
Shipping and Handling........................*(USA/Canada)* $2.50 *(Europe/Asia)* $10.00
California residents must add 6%/6½% sales tax, as required.

Skyles Electric Works
231E South Whisman Road
Mountain View, California 94041
(415) 965-1735

Visa/Mastercard orders: call tollfree (800) 227-9998 (except California). California orders: please call (415) 965-1735.

IBM COMPATIBLE 8" FLOPPY DISK for CBM/PET,* Aim, Sym

- HIGH SPEED — at least FOUR times faster than the 488 bus disks.
- HIGH CAPACITY — up to 850K bytes of storage in a three drive system.
- RELIABILITY — wide timing margins insure long trouble-free operation.
- SOFTWARE — the PDOS II disk software provides a full set of BASIC commands with standard syntax.

5¼" MINI FLOPPY DISK SYSTEMS:

Model 540-1 Single Drive, Double Density (143K)	$595.00
Model 540-2 Dual Drive, Double Density (286K)	$895.00
Model 580-1 Single Drive, Quad Density (286K)	$795.00
Model 580-2 Dual Drive, Quad Density (572K)	$1,195.00

8" MINI FLOPPY DISK SYSTEMS:

Model 877-1 Single Drive, IBM standard (295K)	$1,095.00
Model 877-2 Dual Drive, IBM standard (590K)	$1,695.00

PEDISK II

---

fullFORTH +

FULL FEATURE "FORTH" FOR 6502 SYSTEMS
Interpreter — Cross-Compiler — Cond. Assembler — Screen Editor — String Handling — Floating Point
SPECIFY PEDISK II, PET 2040 or 4040 DISK, OR APPLE $75.00

FOR INFORMATION, SEE YOUR DEALER OR:

cgrs MICROTECH

DEALER INQUIRIES INVITED

ADD-ON DRIVES FOR TRS-80*

Model I, 5¼", 40 Track (102K)	$329.00
Model I, 5¼", 80 Track (204K)	$429.00

P.O. BOX 102 • LANGHORNE, PA 19047 • (215) 757-0284

*PET IS A TRADEMARK OF COMMODORE

```
820 PN(I)=PI(SN(I))
830 NEXTI
840 GOTO950
850 FORI=2TOL%
860 MX=SN(I-1)+MV
870 IFMX>25THENMX=25
880 MN=SN(I-1)-MV
890 IFMN<1THENMN=1
900 NO=MX-MN+1
910 CG=INT(RND(6)*NO)
920 SN(I)=MN+CG
930 PN(I)=PI(SN(I))
940 NEXTI
950 FORI=1TOL%:SN(I)=PN(I):NEXTI
960 GOSUB 1410:REM SET TIMES AND PL
    AY NOTES
970 GOTO360
980 REM ****************************
    ***
990 REM 1/F MUSIC *****************
    ***
1000 REM ****************************
    ***
1010 GOSUB 1190 :REM GET SONG DATA
1020 L%=32
1030 FORI=1TO5:D(I)=INT(RND(8)*6+1):
    NEXTI
1040 SN(1)=D(1)+D(2)+D(3)+D(4)+D(5)-
    5
1050 IFSN(1)<1THENSN(1)=1
1060 SN(1)=PI(SN(1))
1070 FORI=2TOL%
1080 IFI=17THEND(1)=INT(RND(8)*6+1)
1090 IFINT((I-1)/8)=(I-1)/8THEND(2)=
    INT(RND(8)*6+1)
1100 IFINT((I-1)/4)=(I-1)/4THEND(3)=
    INT(RND(8)*6+1)
1110 IFINT(I/2)<>I/2THEND(4)=INT(RND
    (8)*6+1)
1120 D(5)=INT(RND(8)*6+1)
1130 SN(I)=D(1)+D(2)+D(3)+D(4)+D(5)-
    5
1140 IFSN(I)<1THENSN(I)=1
1150 SN(I)=PI(SN(I))
1160 NEXTI
1170 GOSUB1410
1180 GOTO360
1190 REM **************************
1200 REM ASK FOR SONG DATA
1210 REM **************************
1220 PRINT"{CLEAR}{03 RIGHT}{03 DOWN
    DOWN}{REV}COMPOSITION DATA
    "
1230 IFTY$="4"THEN1270
1240 INPUT"{02 RIGHT}{DOWN}ENTER LEN
    GTH, IN NOTES";L%
1250 IFL%<=0THENPRINT"{DOWN}TOO SHOR
    T":GOTO1240
1260 IFL%>150THENPRINT"{DOWN}MAXIMUM
     LENGTH 150":GOTO1240
1270 INPUT"{02 RIGHT}{DOWN}DIFFERENT
     LENGTH NOTES (Y OR N) ";L
    $
1280 IFRIGHT$(L$,1)<>"Y"ANDRIGHT$(L$
    ,1)<>"N"THENPRINT"{DOWN}EN
    TER Y OR N":GOTO1270
1290 INPUT"{DOWN}{02 RIGHT}SLOW, MED
    IUM, FAST (S,M,F)";S$
1300 IFS$<>"S"AND S$<>"M"ANDS$<>"F"T
    HENPRINT"{DOWN}S,M, OR F":
    GOTO1290
1310 IFS$="S"THENS=1
1320 IFS$="M"THENS=2
1330 IFS$="F"THENS=4
1340 INPUT"{DOWN}{02 RIGHT}REPEAT NO
    TES (Y OR N) ";RP$
1350 IFRP$<>"Y"ANDRP$<>"N"THENPRINT"
    {DOWN}ENTER Y OR N":GOTO13
    40
1360 IFTY$<>"3"THEN1400
1370 INPUT"{02 RIGHT}{DOWN}MAX. VAR.
     FROM LAST NOTE ";MV
1380 MV=INT(MV)
1390 IFMV<=0THENPRINT"{DOWN}INVALID
    VALUE ":GOTO1370
1400 RETURN
1410 REM ************************
1420 REM GENERATE TIMES AND PLAY NOT
    ES
1430 REM ****************************
    ***
1440 IFL$="Y"THEN1490
1450 FORI=1TOL%
1460 ST(I)=16/S
1470 NEXTI
1480 GOTO1540
1490 W=64/S
1500 FORI=1TOL%
1510 R=INT(RND(4)*5+1)
1520 ST(I)=W/R
1530 NEXTI
1540 POKE59467,16:POKE59466,15
1550 FORI=1TOL%
1560 POKE59464,SN(I)
1570 T=TI
1580 IFTI-T<ST(I)THEN1580
1590 POKE59464,0
1600 GETA$:IFA$="X"THEN1630
1610 NEXTI
1620 IFRP$="Y"THEN1550
1630 POKE59467,0:POKE59466,0
1640 RETURN
```

©

Ghost Programming

Aric Wilmunder
Los Angeles, CA

I will show how it is possible for 16K Atari users to write and run BASIC programs normally requiring 24 or even 32K. This method is not at all like the method given to us in the BASIC manual where small programs simply call each other and passing of variables and arrays is difficult. Instead, this method is many times more powerful than chaining. Passing of variables is easy, and chaining is unnecessary.

In this article, I will explain how it is possible to write lines of code, subroutines, even entire programs without using any memory except for the space necessary for variables, arrays, and strings. How it is even possible to call and execute programs without changing or destroying the currently stored program. However, like every silver lining, mine too has a dark cloud – there are a number of restrictions involved. I will try to cover these restrictions thoroughly, but only after explaining the technique.

I should mention that, although all of the programming examples are disk oriented, all of the techniques used can be easily modified for cassette users.

After spending nearly four weeks trying to cram close to 40K worth of program into a 32K machine, I began to re-examine the problem of conserving memory. There are many ways to save memory space on the Atari, from removing I/O buffers on the DOS to complete recoding (of which I have done quite a bit). (A list of memory conservation techniques is included as part of *De Re Atari*, and anyone interested in writing large programs should become familiar with them.)

Instant Exec

What kept nagging me were the fifty or more lines of initialization code that were executed only once during my entire program. After their execution, these lines simply took up precious memory space which could be used for other purposes. Also, many of these lines are simply variable assignment statements like J = 12 or I = 1, or string assignments like A$ = "PHASERS." These statements must be executed at the beginning of each execution, but could be forgotten during execution.

Of the two types of assignments, variable and string, the string assignments concerned me the most. The statements DIM A$(26):A$ = "ABCD...Z" does not use only the 26 bytes for storing the string, but you are also using another 26 or more bytes for the assignment. The result is that your program is using more than twice the memory that is necessary in order to store a string. This may be no problem with smaller strings of up to fifty bytes, but, when using larger strings in a program where memory is already scarce, it can be quite alarming.

The method that has solved most of my problems goes something like this: create a file with all of the assignment statements used in the opening of the program in the same structure as a LIST file but minus the line numbers. For example: rather than having a LIST file that, when dumped, looks exactly like a program listing. You have the same line of code, but with commands only. The line:

```
1000 FOR I=65 TO 90: ?CHR$(I);: NEXT I
```

would read:

```
FOR I=65 TO 90: ?CHR$(I);: NEXT I
```

When entered, this line would act exactly as if it were typed on the keyboard by hand. At the beginning of my main program I use the command 'ENTER"D:<filename>" '. This command causes the system to enter each line of code from my Exec program and execute it using virtually no memory space.

You can create a file with only an initialization routine, or go so far as to write an entire program with this method. To execute any of these programs you simply type 'ENTER' or 'E.', the extension and the file name. BASIC will treat this Exec Program exactly as if you were typing in each statement from the keyboard, thereby using no memory space for lines used only once. The amount of memory that can be saved from this method ranges from 5% to virtually an entire program's space.

One of the restrictions with this technique is that programs must be single step or step by step executable. The program must step one line at a time executing each line separately for the entire length of the program. Another restriction is that you cannot have multi-line FOR/NEXT loops (where both the FOR and the NEXT do not reside on the same line). The difficulty is in that, by the time the NEXT is encountered, BASIC will have discarded the FOR statement, giving the loop nowhere it can return to, and causing an error. The lines:

```
FOR I=65 TO 90
?CHR$(I);
NEXT I
```

would have to be restructured into one single line. A simple test for writing and developing Exec Programs is to try to write the program by typing

each statement directly into the machine without using line numbers and then checking the results.

Another Restriction

Still another restriction is that, because EXEC programs have no line numbers, GOTOs and GOSUBs to points within the EXEC program are not allowed. However, if you currently have a program in memory, you can call outside routines that exist in your main program without affecting program control. Say you have a delay routine at line 100 in your main program; you can have your Exec Program GOSUB that line and then return to the next line of the EXEC program. If you want, you can even have a loop that will repeatedly call that routine. This technique is shown in Program 3.

In order to create EXEC program files like the one I described, I have written a simple demo program which will write them. In this demo, you write your own program starting at line 1000 and continuing anywhere up to line 9999. The program writes itself out to disk in a LIST file containing only the lines between 1000 and 9999. This LIST file is then opened as an input file and each line is read individually, the line numbers are removed, and the line is rewritten to a new file. When the program ends, you can test your file by typing: E."D:NOMEM.EXE.

If your Exec Program was properly written, the file should be executed and your original program will remain unchanged. If you tried the disk directory program, (Program 1), you would now have a program on disk which could be called at any time and would leave no leftover lines to be deleted later.

One feature which I should mention about this demo program is the ability to test your program before making a file. By typing GOTO and the line number of the first line of your program, you can follow the program execution and even make changes where necessary before creating an Exec Program. This is important because, if an error occurs anytime during execution, the EXEC program will stop and control will return to the monitor. For testing, type E."D:<Filename> and check for proper program flow. If problems arise, you can list the line numbers, make changes, and RUN the program again until all bugs are removed.

Transfer Of Control

Two aspects of using this method merit close attention. The first is that if you wish to enter this program from a running program, it is necessary to have a GOTO (next line in Main program) as the last statement. This will turn program control over from your Exec program to your Main program when the Exec is over. If this is forgotten, when the EXEC program is over, execution will stop. Since variables, arrays, and strings are passed on, the Main program can use variables from the Exec and vice versa.

The other interesting aspect is that keyboard input will be changed while the machine is reading from the file. The problem arises from the fact that, while the EXEC program is running, the machine acts as if all commands are being typed in directly on the keyboard. When a regular INPUT command is encountered, rather than inputing from the keyboard, the next piece of information will be read in from the disk. If a string is being input, that string will look like the next series of commands. The way around this is to open the keyboard as an input buffer. (OPEN #1,4,0,"K:") Strings and numeric values would then be entered in a loop using repeated GET commands and ending when a <CR> is encountered. The routine given will automatically terminate after a specified number of characters have been entered. (In the sample program, 20 characters are entered, but this can be changed by replacing both 20's in the routine with whatever you like.) The routine also tests for DELETE characters and modifies the string accordingly. For numeric values, you can simply let A=VAL(A$). This is shown in Program 2.

After you have tried a number of programs, you will notice that the prompt READY will appear after each line is executed. So far, I have no cure for this problem, but if one is found I'll be sure to let you know.

In a short period of time, you can build a substantial library of Exec functions. By changing the name of the output file, you can label the functions any way you find convenient. For example; E."D:DIR would display your current directory, and E."D:HEXDEC would convert hex values to decimal. Except for variable declarations, none of these would affect the current program in memory.

All in all, I have shown only a handful of the potential uses of Exec Programs. Other uses might include complex Batch jobbing and self-deleting line numbers. Any new ideas or feedback about this technique would be greatly appreciated. Like many aspects of the Atari, I feel that we are still only beginning to understand the full potential of this fantastic machine.

Main Program

```
100 DIM A$(500)
110 TRAP 200
120 LIST"D:XYZZY.TMP",1000,9999
130 OPEN#1,4,0,"D:XYZZY.TMP"
140 OPEN#2,8,0,"D:NOMEM.EXE"
150 INPUT#1;A$
160 PRINT#2;A$(6)
170 GOTO 150
200 IF PEEK(195)<>136 THEN ?"ERROR -";
```

```
    PEEK(195)
210 CLOSE#1
220 CLOSE#2
230 END
```

Program 1: Disk Directory

```
1000 GRAPHICS 0:CLOSE#1:OPEN#1,6,0,"D:*.*"
     : FOR I=1 TO 999:GET#1,A:?CHR$(A);:
     IF A<>155 OR B<>83 THEN B=A: NEXT I
```

Program 2: Input A Value

```
1000 CLOSE#1:?"ANSWER?";:OPEN#1,4,0,"K:":
     FOR I = 1 TO 20: GET#1,A: ?CHR$(A);:
     A$(I)= CHR$(A): I=I+20*(A=155)-2*(A=
     126): NEXT I
```

Program 3: Calling Outside Routines

```
500 FOR I=0 TO 127
510 PRINT CHR$(27);CHR$(I);
520 NEXT I
1000 FOR J = 1 TO 5: GOSUB 500: NEXT J
```

Program 4: List Program Variables

```
1000 J=PEEK(130)+256*PEEK(131)
1010 FOR J=J TO PEEK(132)+256*PEEK(133)-1:
     ?CHR$(PEEK(J)-128*(PEEK(J)>127));CHR$
     (27+128*(PEEK(J)>127));:NEXT J
1020 I=0: FOR J=PEEK(130)+256*PEEK(131) TO
     PEEK(132)+256 * PEEK(133)-1: I = I +
     (PEEK(J)<127):NEXT J: ? I;" VARIABLES"
```

BASIC 4.0 To Upgrade Conversion Kit

Elizabeth Deal
Malvern, PA

Q: When is a NEXT not a NEXT?
A: When it's a DCLOSE command from Basic 4, of course.

This article is intended primarily for users of the Upgrade PET/CBM systems. It discusses several BASIC 4 disk commands, as they appear on the Upgrade screen.

BASIC 4 programs can often run in the upgrade system with or without conversion. But, to convert, one must know the author's intent in the program and the Upgrade system obliterates the necessary information. Reflect on a three way analogy you might, some day, see on your screen:

NEXT = RETURN WITHOUT GOSUB = DCLOSE

It looks curious, but it makes sense.

A Bit Of History

Some time ago, I had the pleasure of using a BASIC 4 CBM. I was writing a relative file program. At one point I *had* to renumber the program, CBM couldn't do it for me, and the only sensible solution was to load the program into my trusty old Upgrade PET equipped with Toolkit™. I listed the program to see how the disk commands would behave in a new environment.

Assorted quotes from BASIC 4:

```
300  FOR I = 1TONF:RECORD#(DF),(CR),(FP%(I))
310  PRINT#DF,F$(I):GOSUB230:NEXT:RETURN
READY.
2020  DOPEN#(DF),(FF$),D(DD),L(RS)  :GOSUB230
2030  RECORD#(DF),)NR):GOSUB230:PRINT#(DF),
      CHR$(255):GOSUB230
2040  CLOSEDF:GOSUB230:OPENDF:GOSUB230:FR
      = 1:RETURN
READY.
3090  SCRATCH(KY$)
READY.
4020  DCLOSE
READY.
```

As seen by the Upgrade system:

```
300  FORI = 1TONF:DATA#(DF),(CR),(FP%(I))
310  PRINT#DF,F$(I):GOSUB230:NEXT:RETURN
READY.
2020  FOR#(DF),(FF$),D(DD),L(RS)  :GOSUB230
2030  DATA#(DF),(NR):GOSUB230:PRINT#(DF),
      CHR$(255):GOSUB230
2040  CLOSEDF:GOSUB230:OPENDF:GOSUB230:
      FR = 1:RETURN
READY.
3090 GOSUB(KY$)
READY.
4020  NEXT
READY.
```

The screen showed FOR# where DOPEN# should have been (line 2020), and DATA# where a relative file statement RECORD# should have been (lines 300 and 2020). Worse still, it translated SCRATCH(KY$) into GOSUB(KY$) in line 3090. Finally, a conversion of a simple DCLOSE into NEXT (line 4020) seemed incredible.

Both the Toolkit and the PET left those keyword tokens intact (I did not retype the BASIC 4 keywords, doing that would have destroyed them). The program worked fine after transfer to the BASIC 4 computer. And that was that.

Recently, I had to look at that undocumented mess of code. I remembered some of the nasties, but couldn't recall them all. Several of these commands leaped out in a listing as invalid ones, but I didn't catch NEXT, of course. It seemed to belong. However, Power didn't let this one slip by.

While scrolling through the program, back and forth, looking for additional trouble, I noticed that GOSUB(KY$) translated into STRING TOO LONG(KY$) and there appeared a strange looking 4020 RETURN WITHOUT GOSUB statement. That was my NEXT. (I cannot provide a printout, because to print we use the LIST command, whereas these two long sentences were not done by LIST, they resulted from scrolling.)

I was lucky in that I was looking at a program I had written and had a vague idea of what it did. But imagine, for an instant, that somebody sends you a program containing BASIC 4 disk commands. How can you go about finding out which are used? How can you distinguish the true Upgrade commands, like NEXT from BASIC 4 disk commands?

Solution

It always helps to understand the process. The Power manual was useful in solving this one for me, because it explained where and how Power, and the PET for that matter, pick up the keywords and error messages contained within ROM.

One way to get at the keywords is to look in ROM in both Upgrade and BASIC 4 systems and produce a side-by-side listing of tokens and messages. The search addresses were taken from memory maps.

I used this routine:

```
140 N=0:F=1:M=128:P=127:TP=PEEK(50003)
150 S=49298:E=49812:REM UPGR & ORIG
160 IFTP=160THENS=45234:E=45858:REM 4
    80 COLUMN
170 FORJ=STOE
180 IFFTHENF=0:PRINT:PRINTN;N+M;:N=N+1
190 V=PEEK(J):IFVANDMTHENV=VANDP:F=1
200 PRINTCHR$(V);
210 NEXTJ
READY.
```

The results are shown in Figure 1. A list nearly identical to the BASIC 4 listing was in **COMPUTE!** #15, and the list of the Upgrade tokens was in **COMPUTE!** #1. The list presented here also adds the messages which follow the list of tokens.

Note that tokens on the Upgrade PET range in number from 128 to 203. From 204 down we have the PET-people interface. On the BASIC 4 systems, tokens range from 128 to 218 with tokens 128-203 being common between the two systems. Messages follow the tokens and begin at number 219.

The tokens that give us trouble are the ones in BASIC 4 numbered 204-218. They line up with Upgrade PET's messages or with the beginning of the token list, depending who is doing the lining up, LIST or Power's scroller.

The Logic Of It All

The reason behind it goes like this (I think) : The program that runs the PET, the BASIC interpreter, takes a BASIC 4 token that was loaded in, for instance token 206 (DCLOSE). In order to print it on the screen, it scans the table looking for 206. But the Upgrade PET knows that the highest valid token number in its list is 203. When the list is exhausted, it wraps around and starts at the top of the list, goes down three more items and, consequently, returns an inconspicuous NEXT. Power, on the other hand, doesn't wrap around. When a token, invalid for the system, exists in the program, it goes down the list to number 206 and finds a clearly visible RETURN WITHOUT GOSUB message, equivalent to DCLOSE. All quite logical. And simple.

The conversion kit, therefore, consists of a list of tokens and messages. By some careful work on your part, BASIC 4 programs can be read on an Upgrade screen. If you see a strange looking command, you can find out what it means by aligning the tokens and messages.

Try to guess what BASIC 4 statement is intended when the LIST command says END and Power's scroller says NEXT WITHOUT FOR? How about LIST showing GOTO and the scroller showing REDIM'D ARRAY?

Subsequent to the disk commands having been decoded from their curious appearance, the only remaining job is to rewrite those commands into words Upgrade PET can understand (to achieve reverse compatibility). Relative file commands cannot be converted that easily. For this you might consult reference (4) below. If you see RECORD # scattered in the BASIC 4 program, you'll need to do some work. In any case, make sure that you add a semicolon at the end of all PRINT statements. Other commands can be translated with little difficulty by consulting the disk manual, once you know what they are supposed to be.

Don't Jump To Conclusions

WARNING: Trying to write a BASIC 4 program on an Upgrade PET cannot work easily. Writing FOR#4 will not result in DOPEN#4, unless you scan the program and add 75 to the selected FOR token value leaving intended FORs alone. It makes no sense to try to do it, because you couldn't debug your hybrid creation anyway.

REFERENCES:

1) Butterfield's Memory maps in **COMPUTE!** *issues 2 and 7.*
2) POWER Manual (Professional Software).
3) User's Manual for CBM 5¼-inch Dual Floppy Disk Drives, Commodore Business Machines, part # 320899.
4) Butterfield's Mixing and Matching Commodore disk system.

I am grateful to COMPUTER FORUM of FRAZER, PA for permitting me to use their BASIC 4 equipment.

DR. DALEY Introduces...

THE WIZ

Data Management System

THE WIZ is here!!!

THE WIZ, a system powerful enough to manage most of your data storage and manipulation needs - yet is easy to use. A system we are so sure of that we have an offer you can't resist. First though, let's take a look at a few of the many features of this program.

Feature	Benefit
1. ON-LINE help	At your fingertips is the equivalent of a 60 page manual. At any time the computer is waiting for a response from you, you may press the 'h' key or type 'help'. THE WIZ will then provide you with an explanation of the function you are working with.
2. Plotting capability	This is a feature unique to THE WIZ. It can produce a bar graph with up to 18 bars or a histogram with up to 100 points plotted. Graphically presented data is easy to interpret.
3. Wordpro interface	This option is standard with THE WIZ. With many of the competing data managers, if available, it is an extra cost option.
4. Read a sequential file	You may reorganize your files or even read sequential files generated by other data management systems.
5. Search for keywords	Here you can search for a word in ANY field in your record. It can even ignore differences due to upper case and lower case characters.
6. Constants in data entry	You may store up to three separate sets of contant fields. Each set can have as many fields as you like filled with information. Then two keystrokes will call the appropriate set.

And there is more. There is not room enough to tell you all the features in a one page ad!

SPECIAL OFFER

You will like our system. In fact we are so convinced of this, that we are going to pay you to try it! If you have another commercially available data management program, you can receive TRADE-IN credit for your purchase of THE WIZ. Call us on our toll free number (800) 548-3289 for our offer on your present system. Remember that this offer expires March 15, 1982.

DR. DALEY'S SOFTWARE

Water St. Darby, MT. 59829

(800) 548-3289 (for Orders & Information)

(406) 821-3924 (Montana and technical assistance)

Call between 9 a.m. and 6 p.m. Mountain time Monday through Fridav

UPGRADE

```
0   128 END
1   129 FOR
2   130 NEXT
3   131 DATA
4   132 INPUT#
5   133 INPUT
6   134 DIM
7   135 READ
8   136 LET
9   137 GOTO
10   138 RUN
11   139 IF
12   140 RESTORE
13   141 GOSUB
14   142 RETURN
15   143 REM
16   144 STOP
17   145 ON
18   146 WAIT
19   147 LOAD
20   148 SAVE
21   149 VERIFY
22   150 DEF
23   151 POKE
24   152 PRINT#
25   153 PRINT
26   154 CONT
27   155 LIST
28   156 CLR
29   157 CMD
30   158 SYS
31   159 OPEN
32   160 CLOSE
33   161 GET
34   162 NEW
35   163 TAB(
36   164 TO
37   165 FN
38   166 SPC(
39   167 THEN
40   168 NOT
41   169 STEP
42   170 +
43   171 -
44   172 *
45   173 /
46   174 ↑
47   175 AND
48   176 OR
49   177 >
50   178 =
51   179 <
52   180 SGN
53   181 INT
54   182 ABS
```

BASIC 4

```
0   128 END
1   129 FOR
2   130 NEXT
3   131 DATA
4   132 INPUT#
5   133 INPUT
6   134 DIM
7   135 READ
8   136 LET
9   137 GOTO
10   138 RUN
11   139 IF
12   140 RESTORE
13   141 GOSUB
14   142 RETURN
15   143 REM
16   144 STOP
17   145 ON
18   146 WAIT
19   147 LOAD
20   148 SAVE
21   149 VERIFY
22   150 DEF
23   151 POKE
24   152 PRINT#
25   153 PRINT
26   154 CONT
27   155 LIST
28   156 CLR
29   157 CMD
30   158 SYS
31   159 OPEN
32   160 CLOSE
33   161 GET
34   162 NEW
35   163 TAB(
36   164 TO
37   165 FN
38   166 SPC(
39   167 THEN
40   168 NOT
41   169 STEP
42   170 +
43   171 -
44   172 *
45   173 /
46   174 ↑
47   175 AND
48   176 OR
49   177 >
50   178 =
51   179 <
52   180 SGN
53   181 INT
54   182 ABS
```

UPGRADE

55	183	USR
56	184	FRE
57	185	POS
58	186	SQR
59	187	RND
60	188	LOG
61	189	EXP
62	190	COS
63	191	SIN
64	192	TAN
65	193	ATN
66	194	PEEK
67	195	LEN
68	196	STR$
69	197	VAL
70	198	ASC
71	199	CHR$
72	200	LEFT$
73	201	RIGHT$
74	202	MID$
75	203	GO
76	204	NEXT WITHOUT FOR
77	205	SYNTAX
78	206	RETURN WITHOUT GOSUB
79	207	OUT OF DATA
80	208	ILLEGAL QUANTITY
81	209	OVERFLOW
82	210	OUT OF MEMORY
83	211	UNDEF'D STATEMENT
84	212	BAD SUBSCRIPT
85	213	REDIM'D ARRAY
86	214	DIVISION BY ZERO
87	215	ILLEGAL DIRECT
88	216	TYPE MISMATCH
89	217	STRING TOO LONG
90	218	FILE DATA
91	219	FORMULA TOO COMPLEX
92	220	CAN'T CONTINUE
93	221	UNDEF'D FUNCTION
94	222	ERROR IN

BASIC 4

55	183	USR
56	184	FRE
57	185	POS
58	186	SQR
59	187	RND
60	188	LOG
61	189	EXP
62	190	COS
63	191	SIN
64	192	TAN
65	193	ATN
66	194	PEEK
67	195	LEN
68	196	STR$
69	197	VAL
70	198	ASC
71	199	CHR$
72	200	LEFT$
73	201	RIGHT$
74	202	MID$
75	203	GO
76	204	CONCAT
77	205	DOPEN
78	206	DCLOSE
79	207	RECORD
80	208	HEADER
81	209	COLLECT
82	210	BACKUP
83	211	COPY
84	212	APPEND
85	213	DSAVE
86	214	DLOAD
87	215	CATALOG
88	216	RENAME
89	217	SCRATCH
90	218	DIRECTORY
91	219	NEXT WITHOUT FOR
92	220	SYNTAX
93	221	RETURN WITHOUT GOSUB
94	222	OUT OF DATA
95	223	ILLEGAL QUANTITY
96	224	OVERFLOW
97	225	OUT OF MEMORY
98	226	UNDEF'D STATEMENT
99	227	BAD SUBSCRIPT
100	228	REDIM'D ARRAY
101	229	DIVISION BY ZERO
102	230	ILLEGAL DIRECT
103	231	TYPE MISMATCH
104	232	STRING TOO LONG
105	233	FILE DATA
106	234	FORMULA TOO COMPLEX
107	235	CAN'T CONTINUE
108	236	UNDEF'D FUNCTION
109	237	ERROR IN

©

Screen Save Routine

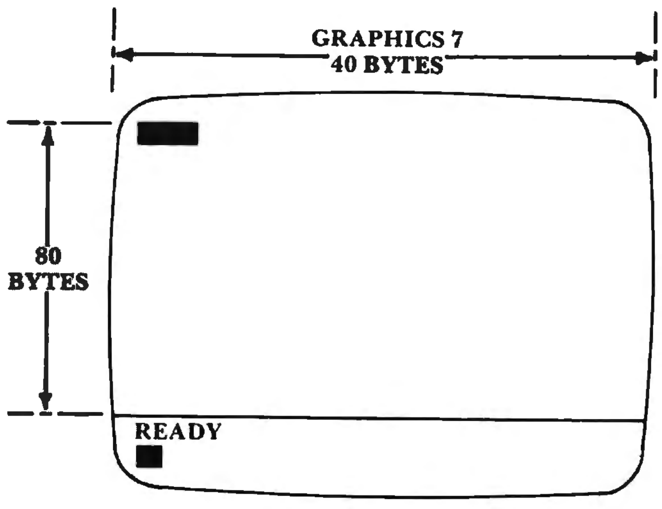
Joseph Trem
Garfield Heights, OH

A while back, **COMPUTE!** published an excellent article for drawing pictures entitled "SUPER CUBE" (**COMPUTE!** #11). Pictures are drawn to look 3-dimensional. After playing with this program, some very interesting pictures developed... all to be lost when the computer was turned off! Gee, it sure would be nice to save those Rembrandts!

The following utility program can be appended to the end of your favorite drawing programs and will enable you to save those Rembrandts. A sample drawing is included at the beginning of this utility.

The Atari computer is fascinating indeed. The more you delve, the more intriguing it becomes. This program is based on three screen-related memory addresses – 87, 88, & 89. Location 87 contains the graphics mode presently in use. Type "GRAPHICS 7", then type "PRINT PEEK(87)". The computer will respond with "7". Locations 88 and 89 store the starting addresses of screen memory. 88 contains the low byte and 89 contains the high byte. Again, type "GRAPHICS 7", then type "PRINT PEEK(88) + PEEK(89)*256". This will return the memory starting address for Graphics 7. Note that each computer may return a different number depending on the memory size of the machine. Now type "POKE(memory start), 255". This will light up one full byte at the top left corner of the screen (Figure 1).

Figure 1.



Type "POKE(memory start + 40,255", and this will light up the next full byte directly under the first byte. Knowing this, it is possible to keep track of every byte on the screen. There are 40 bytes horizontally and 80 bytes vertically in Graphics 7. In the utility program, line 33240 locates the starting address of your picture. Lines 32125 and 32225 scan and set screen memory locations. You may adapt these lines to any graphics mode using the chart provided. For example, if you happen to be using graphics 5, change the "40*80" in those lines to "20*40"

	HORIZONTAL BYTES	VERTICAL BYTES
GR.8	40	160
GR.7	40	80
GR.6	20	80
GR.5	20	40
GR.4	10	40
GR.3	10	20
GR.2	20	10
GR.1	20	20
GR.0 (full screen)	40	24

After running this program, you may want to append only the utility part to your favorite drawing program. Here's how to do it. First, make sure your drawing program does not exceed line 30999. Now type LIST"C:",31000,32240 or LIST "D:filename", 31000,32240. This will save only lines 31000 through 32240. When completed type NEW and load your drawing program (e.g. SUPER CUBE). Now load your utility program back in. This is done by typing ENTER"C:" or ENTER "D:filename". This will append your utility to the end of any drawing program.

Some programs may have to be modified slightly, but with a little effort you may find it worth it. Run your program. Draw your masterpiece. When you are satisfied with your creation, press the BREAK key and type "GOTO 31000". This will initialize the save and load routine. Then sit back, relax and surprise someone with a genuine work of art worthy only of the great masters.

10	Initializes SCREEN SAVE ROUTINE
40-195	Draws sample picture (e.g. Space game playfield)
200	Reinitializes menu after drawing
30000	Sets GR.2 & title
31000	Opens IOCB for keyboard
32000 to 32060	Prints menu. Gets keyboard input and directs to appropriate line.
32100 to 32103	Prints save menu. Gets keyboard input and directs to appropriate line.

32105 to 32208	Prints disk instructions.
32110 to 32210	Prints cassette instructions.
32200	Prints load menu.
32122	Stores graphics mode and color register data.
32125	Stores screen RAM data.
32222	Reads graphics mode and color register data and pokes it into correct location.
32225	Reads screen RAM data and pokes into correct locations.
32240	Determines start address of screen.

```
10 GOTO 30000
40 GRAPHICS 7:SETCOLOR 0,6,6:REM %SET GR
APHICS 7 MODE
60 INC=49:CO=1:COLOR 1
65 INC=INC+0.05
70 X=SIN(INC)%20:Y=COS(INC)%20
75 PLOT X+80,Y+35:SOUND 1,X+50,10,8
80 CO=CO+1:IF CO<130 THEN 65
100 CO=1
110 COLOR 3
130 INC=INC+0.05
140 X=SIN(INC+1)%30:Y=COS(INC)%30
145 IF CO>49 AND CO<70 THEN 160
150 COLOR 2:PLOT X+80,Y+35:SOUND 1,X+50,
6,8
155 COLOR 3:PLOT X+81,Y+38:COLOR 1:PLOT
X+79,Y+32
160 CO=CO+1:IF CO<130 THEN 110
170 COLOR 1:FOR X=1 TO 159:PLOT X,79:DRA
WTO X,79-RND(0)%5:SOUND 1,X,10,8:NEXT X
180 FOR X=1 TO 20:COLOR RND(0)%1+1:PLOT
40,40:DRAWTO RND(0)%10,RND(0)%10:SOUND 1
,X+20,8,8:NEXT X
183 FOR X=120 TO 159:COLOR RND(0)%2+1:PL
OT X,20:DRAWTO X,RND(0)%20:NEXT X
185 COLOR 2:PLOT 130,30:DRAWTO 130,24:DR
AWTO 134,24:DRAWTO 134,30:PLOT 130,27:DR
AWTO 134,27
186 PLOT 142,25:DRAWTO 142,24:DRAWTO 138
,24:DRAWTO 138,30:DRAWTO 142,30:DRAWTO 1
42,29
187 PLOT 150,24:DRAWTO 146,24:DRAWTO 146
,30:DRAWTO 150,30:PLOT 146,27:DRAWTO 149
,27
190 COLOR 2:PLOT 0,0:DRAWTO 159,0:DRAWTO
 159,79:DRAWTO 0,79:DRAWTO 0,0
195 COLOR 1:PLOT 30,70:PLOT 40,10:PLOT 1
40,30:PLOT 150,70:PLOT 105,35:SOUND 1,0,
0,0
200 GOTO 32000
30000 GRAPHICS 2:? #6;" SCREEN SAVE ROUT
INE"
31000 CLOSE #1:OPEN #1,4,0,"K:"
32000 SCRN=32240:GOSUB SCRN:POKE 752,1:?
 "              1...DRAW PICTURE"
32010 ? "              2...SAVE PICTURE"
32020 ? "              3...LOAD PICTURE"
32050 GET #1,A:IF A<49 OR A>51 THEN 3205
0
32060 ON A-48 GOTO 40,32100,32200
32100 ? :? "  SAVE TO   1...CASSETTE?":?
 "            2...DISK?"
32101 GET #1,A:IF A<49 OR A>50 THEN 3210
1
32103 ON A-48 GOTO 32110,32105
32105 ? :? "   PLEASE INSERT DISKETTE AN
D PRESS        |RETURN|":GET #1,A:OPEN #2,
8,0,"D:PICTURE":GOTO 32120
32110 ? :? "PLEASE PLACE CLEAN TAPE IN R
ECORDER             AND PRESS |RETURN|"
32115 OPEN #2,8,0,"C:":REM %OPEN FILE TO
 SAVE
32120 ? :? "SIT BACK AND RELAX... |SAVIN
G PICTURE|"
32122 MODE=PEEK(87):PUT #2,MODE:FOR I=0
TO 4:COL=PEEK(708+I):PUT #2,COL:NEXT I
32125 FOR I=SCREEN TO SCREEN+(40%80)-1:L
OC=PEEK(I):PUT #2,LOC:NEXT I:CLOSE #2
32130 GOTO 32000
32200 ? :? "  LOAD TO    1...CASSETTE?":
? "              2...DISK?"
32201 GET #1,A:IF A<49 OR A>50 THEN 3220
1
32203 POKE 752,1
32205 ON A-48 GOTO 32210,32208
32208 ? :? "  PLEASE INSERT DISKETTE AND
 PRESS        |RETURN|":GET #1,A:OPEN #2,4
,0,"D:PICTURE":GOTO 32220
32210 ? " PLEASE INSERT TAPE AND PRESS |
RETURN|"
32215 OPEN #2,4,0,"C:":REM %OPEN FILE TO
 LOAD
32220 ? :? :? "RELAX AND ENJOY... |LOADI
NG PICTURE|"
32222 GET #2,MODE:GRAPHICS MODE:GOSUB SC
RN:FOR I=0 TO 4:GET #2,COL:POKE 708+I,CO
L:NEXT I
32225 FOR I=SCREEN TO SCREEN+(40%80)-1:G
ET #2,LOC:POKE I,LOC:NEXT I:CLOSE #2
32230 GOTO 32000
32240 SCREEN=PEEK(88)+PEEK(89)%256:RETUR
N
32500 REM %%WRITTEN BY JOSEPH TREM%%
32501 REM %% SCREEN SAVE ROUTINE  %%
32502 REM %%       7/81           %%
```

©

Part I:

A Superboard II Monitor

Frank Cohen
Pacific Palisades, CA

In the last article I stated my opinion of the OSI Superboard II's video output. The result was an advanced cursor routine to supplement the powerful ROM-based Microsoft BASIC. Super-Cursor adds many functions to the output routine of BASIC. However, Super-Cursor is a very long routine (almost 500 bytes long) and entering it into the Superboard can take forever using the existing monitor program. This gave me the inspiration to write the next couple of programs.

Upon turning on the Superboard and pressing the BREAK key, one is presented with a choice of entering the BASIC, disk or monitor programs. Unless one knows enough about how BASIC works to write a machine code interface, using the monitor program is the only way to program the computer in machine language. It only takes one look at the monitor program to know that one is very limited in the things that can be done.

The monitor will perform only three different things with the Superboard. First, it allows the viewing of one memory location at a time. This gives the capability of seeing one whole byte of memory for every given keyboard entry. Imagine trying to look through a program like Super-Cursor one byte at a time, keeping in mind there are over 500 instructions. Finding one byte incorrectly entered would take hours. Second, the monitor allows the modification of only one location at a time. After a byte has been entered, the next address is displayed. A problem will occur here if you are not sure the byte you just entered was correct. In this case you would have to go back and look at the last memory location. It then becomes necessary to retype the four digits of the previous address. Finally, the monitor allows the entry of a machine language program from a cassette recorder. Of course, it doesn't allow one to store a program on a cassette recorder (which leads one to wonder what one would be loading to begin with).

An Advanced Monitor

It doesn't take long after you start programming in machine language to realize the necessity of a more advanced monitor program. This program (and two others to follow) form an advanced monitor routine. To be quite original I have named the total program Super-Monitor. It is intended for the type of person who knows the basics of machine language programming and wants to expand his knowledge. The three programs are fully documented and in assembly language format. They are written in three separate packages so that you, the user, can modify to your specifications while, at the same time, learn some simple machine language programming techniques.

Before we start, an outline of what a monitor program should do will make the whole thing come together at the end. First, the monitor should be able to display as few or as many memory locations at one time as is desired. Second, it should allow easy entry of many bytes into the Superboard's memory along with the ability to see what was just entered. Third, it should allow the user to fill many memory locations with certain strings or combinations of bytes. Fourth, it should be able to move a whole block of data from one location to another in memory. Fifth, we should have a routine to store memory onto tape. Finally, we should have a routine to read a program from tape to memory with verification. Once all this is accomplished, entering and editing machine language programs will be a snap.

Hexdump

This program allows us to see many locations of memory at once. It will ask you for a beginning memory address and then it will print one line of eight bytes of data from the memory address specified. It will then wait for your command. If you want the next line of eight bytes to be listed, you simply hit the carriage return key. If you want to list another part of memory, you hit the line feed key and HEXDUMP will start again by asking for a new beginning address.

Before we get into seeing just how Hexdump works, here is a small word of warning. If you read the article describing Super-Cursor V1.3 in **COMPUTE!** #18 you should nave noted that my Superboard II has had the video modifications added to give a video display of 26 lines with 48 characters per line, using the Super-Cursor program. The modifications are simple and well described in conversion plans which can be purchased from Elcomp Publishing (Silver Spur Electronics, Chino, California.)

Hexdump does use some of the routines in Super-Cursor so it is necessary that you load Super-Cursor as well. If you don't want to use Super-Cursor, you will have to write your own output routines to allow Hexdump to display information onto the screen. Writing these routines is not very difficult and you probably could copy the individual routines out of Super-Cursor with only a few modifications.

The assembly listing of Hexdump shows that it loads into locations 1D20 through 1E38. This is the area in memory directly below Super-Cursor V1.3. There are seven bytes of memory which separate the two programs. These will be used later for another part of Super-Monitor. Hexdump may be moved to another location in memory by reassembling it; however, if you do not own an assembler, relocating Hexdump may become very difficult as it uses absolute addresses extensively.

Hexdump is an example of structured machine language programming. It uses a main supervisor routine which branches into other routines. The supervisor is labeled Hexdump in the assembly listing, and calls all of the other subroutines into play.

Upon starting Hexdump, the supervisor homes the cursor using the Home routine of Super-Cursor V1.3. This positions the solid block cursor in the upper left position of the screen. The program then goes to the subroutine labeled Padr. This subroutine prints the two byte address held in location 00E7 and 00E8. This address is later used to find the memory location you want to display.

The program continues by jumping to another subroutine. This one called Inadr. This subroutine reads the keyboard four times, allowing the input of a four digit memory address. The resulting two byte address is put into locations 00E7 and 00E8 (ADR). Once the four numbers have been entered, the program jumps back to the supervisor which again homes the cursor. The program then jumps to another subroutine labeled Pline.

Up to this point you probably will not be modifying the program very much as the function of the subroutines are very straightforward. However, Pline will probably need to be modified as it prints the start address of the desired memory locations and then prints the eight bytes of data contained in those locations. The number of data bytes printed is what may need to be changed, if you have not added the video modifications to your Superboard. On a 24 by 24 video display you can only print up to five bytes at a time unless you don't mind the information falling off one line and continuing on the next. The number of bytes printed is controlled by the byte at 1E24. In the assembly listing you can see that the program between 1E22 and 1E30 is concerned only with checking to see if Pline has printed eight characters and, if it has, to return back to the supervisor. To change the number of bytes printed per line, it is necessary to put the desired number at 1E23 and a copy of that number at 1E2B.

Now that Pline is finished and we have returned to the supervisor, you can see that Hexdump checks the keyboard and, if a carriage return is entered (ASCII value of 0D), it will branch back to print the next eight bytes of memory. If a line feed is entered (ASCII value of 0A), it will branch back to the part of the program which homes the cursor and starts it all again.

Next Month

We now have a program which is the first part of a very advanced monitor program. What comes next is two other routines which include the functions listed in our outline. They are smaller routines than Hexdump. The resulting Super-Monitor will allow you to enter large programs in a single bound.

```
                                ;This program uses some subroutines from
                                ;Super-Cursor V1.3  (COMPUTE! Nov. '81)
                                ;
                                ;Zero page usage is limited to only
                                ;two bytes-
                                ;  00E7 - ADR
                                ;  00E8 - ADR+1
                                ;which are the low and high bytes forming an
                                ;address for which HEXDUMP looks into memory.
                                ;
                                ;
*=1D20                          ;Start of program and entry point.
                                ;
1D20 20 80 1E   HEXDUMP JSR HOME        ;Home the Cursor
1D23 20 3C 1D   DIA     JSR PADR        ;Print address
1D26 20 80 1E           JSR HOME        ;Home the cursor again
1D29 20 93 1D           JSR INADR       ;Input address
1D2C 20 00 1E   DAL     JSR PLINE       ;Print one line
1D2F 20 BA FF   BD      JSR KEYIN       ;Reads the keyboard result in A
1D32 C9 0D              CMP #$0D        ;Key pressed = CR?
1D34 F0 F6              BEQ DAL
1D36 C9 0A              CMP #$0A        ;Key pressed = LF?
```

```
1D38 F0 E9               BEQ DIA
1D3A D0 F3               BNE BD
                                              ;
1D3C A5 E8       PADR    LDA ADR+1            ;Print address on screen
1D3E 29 F0               AND #$F0             ;Start with high 4 bits of
1D40 4A                  LSR                  ;ADR +1  (00E8)
1D41 4A                  LSR
1D42 4A                  LSR
1D43 4A                  LSR
1D44 20 72 1D            JSR CVHA             ;Convert Hex to Ascii
1D47 20 40 1E            JSR CURSOR           ;Part of Super-Cursor
1D4A A5 E8               LDA ADR+1            ;Now do the low 4 bits of ADR+1
1D4C 29 0F               AND #$0F
1D4E 20 72 1D            JSR CVHA
1D51 20 40 1E            JSR CURSOR
1D54 A5 E7               LDA ADR              ;Now work on high 4 bits of ADR
1D56 29 F0               AND #$F0
1D58 4A                  LSR
1D59 4A                  LSR
1D5A 4A                  LSR
1D5B 4A                  LSR
1D5C 20 72 1D            JSR CVHA
1D5F 20 40 1E            JSR CURSOR
1D62 A5 E7               LDA ADR              ;work on low 4 bits of ADR
1D64 29 0F               AND #$0F
1D66 20 72 1D            JSR CVHA
1D69 20 40 1E            JSR CURSOR
1D6C A9 2D               LDA #$2D             ;Print '-' on screen after address
1D6E 20 40 1E            JSR CURSOR
1D71 60                  RTS
                                              ;
1D72 A2 00       CVHA    LDX #$00             ;Convert whats in A from Hex to
1D74 8E 78 1D    CVST    STX CON+1            ;ASCII
1D77 C9 00       CON     CMP #$00             ;This value is changed in CVST
1D79 F0 04               BEQ CFIN
1D7B E8                  INX
1D7C 4C 74 1D            JMP CVST
1D7F BD 83 1D    CFIN    LDA CDATA,X          ;Put result in A
1D82 60                  RTS
1D83 30 31 32    CDATA   DATA                 ;Data used in both CVHA and CVAHX
1D86 33 34 35            DATA                 ;to convert Hex to Ascii and back.
1D89 36 37 38            DATA
1D8C 39 41 42            DATA
1D8F 43 44 45            DATA
1D92 46                  DATA
                                              ;
1D93 20 80 1E    INADR   JSR HOME             ;Input 4 digit (2 byte) Address
1D96 20 BA FF            JSR KEYIN            ;Read Keyboard Routine from ROM
1D99 20 40 1E            JSR CURSOR           ;Super-cursor
1D9C 20 F3 1D            JSR CVAHX            ;Convert Ascii to Hex
1D9F 0A                  LSL
1DA0 0A                  LSL
1DA1 0A                  LSL
1DA2 0A                  LSL
1DA3 8D AC 1D            STA INADC+1          ;Pokes A into 1DAC
1DA6 A5 E8               LDA ADR+1
1DA8 29 0F               AND #$0F
1DAA 18                  CLC
1DAB 69 00       INADC   ADC #$ 00
1DAD 85 E8               STA ADR+1
```

```
1DAF 20 BA FF             JSR KEYIN        ;Get second digit
1DB2 20 40 1E             JSR CURSOR
1DB5 20 F3 1D             JSR CVAHX
1DB8 8D C1 1D             STA INBDC+1      ;Pokes A into 1DC1
1DBB A5 E8                LDA ADR+1
1DBC 29 F0                AND #$F0
1DBF 18                   CLC
1DC0 69 00       INBDC    ADC #$00
1DC2 85 E8                STA ADR+1
1DC4 20 BA FF             JSR KEYIN        ;Get third digit
1DC7 20 40 1E             JSR CURSOR
1DCA 20 F3 1D             JSR CVAHX
1DCD 0A                   ASL
1DCE 0A                   ASL
1DCF 0A                   ASL
1DD0 0A                   ASL
1DD1 8D DA 1D             STA INCDC+1      ;Pokes A into 1DDA
1DD4 A5 E7                LDA ADR
1DD6 29 0F                AND #$ 0F
1DD8 18                   CLC
1DD9 69 00       INCDC    ADC #$00
1DDB 85 E7                STA ADR
1DDD 20 BA FF             JSR KEYIN        ;Get the last digit
1DE0 20 40 1E             JSR CURSOR
1DE3 20 F3 1D             JSR CVAHX
1DE6 8D EF 1D             STA INDCD+1      ;Poke A into 1DEF
1DE9 A5 E7                LDA ADR
1DEB 29 F0                AND #$F0
1DED 18                   CLC
1DEE 69 00       INDCD    ADC#$00
1DF0 85 E7                STA ADR
1DF2 60                   RTS
                                           ;
1DF3 A0 00       CVAHX    LDY#$00          ;Convert contents of A from
1DF5 D9 83 1D    CVCON    CMP CDATA,Y      ;Ascii to Hex
1DF8 F0 04                BEQ CVFIN
1DFA C8                   INY
1DFB 4C F5 1D    CVFIN    JMP CVCON
1DFE 98                   TYA              ;Put result in A
1DFF 60                   RTS
                                           ;
1E00 20 3C 1D    PLINE    JSR PADR         ;Print one line of eight bytes
1E03 A0 00                LDY #$ 00
1E05 B1 E7       PBYTE    LDA (ADR),Y      ;Print one byte from ADR
1E07 29 F0                AND #$F0
1E09 4A                   LSR
1E0A 4A                   LSR
1E0B 4A                   LSR
1E0C 4A                   LSR
1E0D 20 72 1D             JSR CVHA         ;Convert A to Ascii
1E10 20 40 1E             JSR CURSOR       ;Super-cursor
1E13 B1 E7                LDA (ADR),Y      ;Print low 4 bits on screen
1E15 29 0F                AND #$0F
1E17 20 72 1D             JSR CVHA
1E1A 20 40 1E             JSR CURSOR
1E1D A9 20                LDA #$20         ;Print a space " " to separate
1E1F 20 40 1E             JSR CURSOR       ;the bytes
1E22 C8                   INY              ;Are we finished?
1E23 C0 08                CPY #$08
1E25 D0 DE                BNE PBYTE        ;If not display another byte
```

```
1E27 A5 E7              LDA ADR        ;If yes add 08 to ADR
1E29 18                 CLC
1E2A 69 08              ADC #$08
1E2C 85 E7              STA ADR
1E2E 90 02              BCC RCOM
1E30 E6 E8              INC ADR+1
1E32 20 95 1E   RCOM    JSR CR         ;Were finished printing one line
1E35 20 AB 1E           JSR LF         ;so carriage return and line feed
1E38 60                 RTS
                                       ;
                                       ;
                                       ;
;Statistics
;
*=      1D20    ;Start
1D3C    PADR    ;Print ADR and ADR+1 on screen
1D72    CVHA    ;Converts Hex to Ascii
1D93    INADR   ;Input a two byte address for ADR and ADR+1
1DF3    CVAHX   ;Converts Ascii to Hex
1E00    PLINE   ;Print a line of 8 bytes from ADR
1E40    CURSOR  ;Prints what ever is in A to where the cursor is
1E80    HOME    ;Home the cursor
1EC2    CLS     ;Clear screen
```

©

New Products

Good News For Alkemstone Seekers

Level-10 has increased the reward for the return of the Alkemstone from $5,000 to $7,500.

Alkemstone is a computer adventure which now offers a $7,500 cash reward to the first person who can find the missing Alkemstone.

The Alkemstone is actually hidden somewhere on the planet

The quest for the Alkemstone will lead the player through underground paths of the lair of the original owner. There are unusual messages, fragments of words, sketches and other clues written on the walls. Some items are distributed randomly, so that one may be visible numerous times while some will only appear once in several trips. Each trip will result in a different combination of possible hints. If all bits of information are pieced together correctly, then the location of the Alkemstone will be obvious.

Alkemstone is more than just a treasure hunt. It also has the challenge and intrigue of a fascinating computer game. It features hi-res color graphics, 3-D animation, sound effects and an illustrated short story. Alkemstone is written in machine language for any Apple Computer with 48K, one 16 sector disk drive and Applesoft or language card.

Level-10
7475 Dakin St.
Suite 401
Denver, CO 80221

VIC-20 Timesharing With Printout

The MDM-1 is a MODEM driver module that plugs directly into the user port of the Commodore VIC microcomputer.

The MDM-1 has two additional serial ports for connecting a MODEM and a serial printer.

The MDM-1 does not require any external power. A terminal program supplied with the unit permits telephone line connections to large timesharing computers, and to various computer networks, such as source, Compuserve, Dow Jones, etc.

There are two indicators on the MDM-1 that illustrate the transmissions to and from the VIC. These transmissions are simultaneously recorded on the VIC TV screen and the serial printer.

The price of the MDM-1 is $59, plus $3 shipping. Delivery is two weeks after receipt of order.

RVR Systems
P.O. Box 265
Dewitt, NY 13214

MatheMagic™ The Ultimate Calculator

International Software Marketing, Ltd. (ISM) in Syracuse, New York has announced a unique software product for microcomputers. The product is called MatheMagic™. Essentially, the product transforms a microcomputer into a sophisticated programmable calculator with all of the power and facilities of the computer system available to it.

"MatheMagic was created by ISM to fill a void which exists in mathematical microcomputer software," says Stephen T. Brightbill, ISM Director. "A wide variety of software is available, primarily for the business community, to manipulate numeric data in a spreadsheet format. MatheMagic is not a "spreadsheet." MatheMagic allows a wide range of users to easily handle an even wider range of mathematical applications. MatheMagic is completely programmable in commonly understood mathem-

NOW TWO LOCATIONS

SAVE TIME • SAVE SHIPPING

ATARI Computers for people.™

800™ $679

410 Recorder	$76.00
810 Disc Drive	$449.00
822 Printer	$229.00
825 Printer	$629.00
830 Modem	$159.00
820 Printer	$269.00
850 Interface	$169.00
New DOS 2 System	$21.00
CX30 Paddle	$18.00
CX40 Joy Stick	$18.00
CX853 16K RAM	$89.00
Microtek 16K RAM	$75.00
Microtek 32K RAM	$159.00
Ramcram (128K)	$539.00
One year extended warranty	**$50.00**

16K.... $329
32K.... $478
48K.... $555

ATARI SOFTWARE

CX404 Word Processor	$119.00
CX405 PILOT (educational)	$105.00
CX413 Microsoft Basic	$68.00
CX4101 Invitation To Programing I	$17.00
CX4102 Kingdom	$13.00
CX4103 Statistics	$17.00
CX4104 Mailing List	$17.00
CX4105 Blackjack	$13.00
CX4106 Invitation to Programing 2	$20.00
CX4107 Biorythm	$13.00
CX4108 Hangman	$13.00
CX4109 Graph It	$17.00
CX4110 Touch Typing	$20.00
CX4112 States & Capitals	$13.00
CX4114 European Countries & Capitals	$13.00
CX4115 Mortgage & Loan Analysis	$13.00
CX4116 Personal Fitness Program	$59.00
CX4117 Invitation To Programing 3	$20.00
CX4118-20 Conversational Languages (ea.)	$45.00
CX4121 Energy Czar	$13.00
CXL4001 Educational Master	$21.00
CX6001-17 Talk & Teach Series (ea.)	$23.00
CX8106 Bond Analysis	$20.00
CX8107 Stock Analysis	$20.00
CX8101 Stock Charting	$20.00
CXL4002 Basic Computing Language	$46.00
CXL4003 Assembler Editor	$46.00
CXL4004 Basketball	$24.00
CXL4005 Video Easel	$24.00
CXL4006 Super Breakout	$30.00
CXL4007 Music Composer	$45.00
CXL4008 SPACE INVADERS	$32.00
CXL4009 Chess	$30.00
CXL4010 3-D Tic-Tac-Toe	$24.00
CLS4011 STAR RAIDERS	$39.00
CXL4012 MISSLE COMMAND	$32.00
CXL4013 ASTEROIDS	$32.00
CXL4015 TeleLink	$20.00
Visicalc	$149.00
Letter Perfect (Word Processor)	$109.00
Source	$89.00
CX481 The Entertainer	$75.00
CX482 The Educator	$130.00
CX483 The Programmer	$54.00
CX 484 The Communicator	$329.00

Texas Instruments

TI-99/4A $359

PHP 1600 Telephone Coupler	$169.00
PHP 1700 RS-232 Accessories Interface	$169.00
PHP 1800 Disk Drive Controller	$239.00
PHP 1850 Disk Memory Drive	$389.00
PHP 2200 Memory Expansion (32K RAM)	$319.00
PHA 2100 R.F. Modulator	$43.00
PHP 1100 Wired Remote Controllers(Pair)	$31.00
32K Expansion	$329.00
PHP Printer Solid State	$319.00

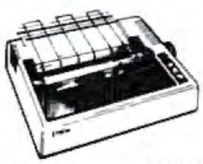
CALL FOR SOFTWARE SELECTION AND PRICES

XEROX 820

Xerox 820	
System I 5¼"	$2450.00
System II 8"	$2950.00
CPM 5¼"	$169.00
Word Processing	$429.00
Super Calc	$269.00

CALL FOR MORE INFORMATION

PRINTERS



Centronics 739-1	$649.00
Diablo 630 Special	$1799.00
Epson	
MX70	$359.00
MX80	$469.00
MX80FT	Call
MX100	Call
NEC	
8023	$639.00
7730	Call
7720	Call
7710	Call

Okidata	
82A	$499.00
83A	$769.00
84	$1129.00
Citoh Starwriter	
25 CPS-P	$1329.00
45 CPS-P	$1699.00
Paper Tiger	
445G	$699.00
460G	$899.00
560G	$1129.00
Talley	
8024-7	$1399.00
8024-L	$1629.00

Terminals

Televideo	
910	$579.00
912C	$699.00
920C	$749.00
950	$939.00
Call for computers	
Zenith Z19	$749.00
Adds	$549.00

Modems

Novation Auto	$239.00
D Cat	$169.00
Cat	$159.00
Hayes	
Smart	$239.00

HOW TO ORDER: Phone orders invited or send check or money order and receive free shipping in the continental United States. PA. and NEV. add sales tax.

computer mail order west

800-648-3351

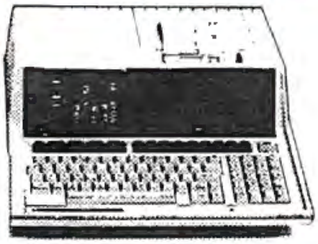
IN NEVADA, CALL (702) 588-5654

P.O. BOX 6689, STATE LINE, NEVADA 89449

TO SAVE YOU MORE!

COSTS • SAVE SALES TAX

HEWLETT PACKARD



HP•85 $1999

80 Column Printer	$799.00
NEW! HP•125	$2999.00
HP•83	$1699.00
HP•85 16K Memory Module	$249.00
5¼" Dual Master Disc Drive	$2129.00
Graphics Plotter (7225B)	$2079.00

NOW IN STOCK!

The new HP41CV Calculator

$259

41C	$189.00
11C	$119.00
12C	$129.00
34C	$117.00
38C	$119.00
HP•41 Printer	$340.00
Card Reader	$164.00
Optical Wand	$99.00

HPIL CALCULATOR PERIPHERALS

IL Modual	$104.00
Digital Cassette	$449.00
Printer/Plotter	$419.00

CALL FOR SOFTWARE INFORMATION

Monitors

Amdex 12" B&W	$149.00
12" Green	$169.00
13" Color	$349.00
Sanyo 12" B&W	$259.00
12" Green	$269.00
13" Color	$449.00
TI 10" Color	$349.00

Electronics

Pioneer Lazer Disk	$599.00
BSR X-10 Systems	
PK 500	$84.00
LM 501	$16.00
AM611	$17.00
AM286	$17.00

commodore

CBM 8032

$1039

4032	$969.00
4016	$769.00
8096	$1569.00
Super Pet	$1599.00
2031	$529.00
8050	$1299.00
4022	$599.00
4040	$949.00
8300 (Letter Quality)	$1799.00
8023	$769.00
Pet to IEEE Cable	$37.00
IEEE to IEEE Cable	$46.00
Tractor Feed for 8300	$240.00
8010 Modem	$229.00

SOFTWARE

WordPro3 Plus	$199.00
WordPro4 Plus	$299.00
Commodore Tax Package	$399.00
Visicalc	$149.00
BPI General Ledger	$329.00
OZZ Information System	$289.00
Dow Jones Portfolio	$129.00
Pascal	$239.00
Legal Time Accounting	$449.00
Word Craft 80	$289.00
Power	$79.00
Socket 2 Me	$20.00
Jinsam	$Call
MAGIS	$ Call
The Manager	$209.00
Softrom	$129.00
Real Estate Package	$799.00
BPI Inventory Control	$319.00
BPI Job Costing	$319.00
BPI Payroll	$319.00

VIC 20

$259

COMPLETE

Vic 6 Pack Program	$44.00
VIC1530 Commodore Datassette	$69.00
VIC1540 Disk Drive	$499.00
VIC1515 VIC Graphic Printer	$339.00
VIC1210 3K Memory Expander	$32.00
VIC1110 8K Memory Expander	$53.00
VIC1011 RS232C Terminal Interface	$43.00
VIC1112 VIC IEEE 488 Interface	$86.00
VIC1211 VIC 20 Super Expander	$53.00
VIC1212 Programmers Aid Cartridge	$45.00
VIC1213 VICMON Machine Language Monitor	$45.00
VIC1901 VIC AVENGERS	$23.00
VIC1904 SUPERSLOT	$23.00
VIC1906 SUPER ALIEN	$19.00
VIC1907 SUPER LANDER	$23.00
VIC1908 DRAW POKER	$23.00
VIC1909 MIDNIGHT DRIVE	$23.00
VT106A Recreation Pack A	$44.00
VT107A Home Calculation Pack A	$44.00
VT164 Programmable Character/Gramegraphics	$12.00
VT232 VICTerm I Terminal Emulator	$9.00

New VIC Software

Household Finance	$27.00
VIC Games	$19.00
VIC Home Inventory	$13.00
VIC Rec/Ed II	$13.00
I SAM	$79.00
General Ledger	$229.00
Accounts Receivable	$229.00
Inventory	$229.00
Terminal	$13.00
Un Word	$13.00
Grafix Menagerie	$11.00
VIC PICS	$15.00
Ticker Tape	$13.00
Banner Headliner	$13.00
RS 232	$39.00

Above are cash prices, add 3% for Master Card and Visa purchases.

computer mail order east

800-233-8950

IN PA. CALL (717) 327-9575

501 E. THIRD ST., WILLIAMSPORT, PA 17701

atical terms. Now, without the need for programming or computer knowledge, the user can access the computer as an extremely powerful calculator. In short, if the question that you have is more complicated than two plus two and less complicated than differential calculus, then MatheMagic is your answer. MatheMagic can provide the answers to mathematical questions in the business world, in engineering and science, in the classroom and in the home."

Some of the features which make this possible are:

- Menu-driven format for rapid learning and consistent ease of use.
- On-line "HELP" for every MatheMagic command and option.
- Free-form entry of expressions of up to 240 characters (including standard math operators, pre-defined functions, user-defined formulas, parentheses, variables, etc.).
- Unlimited use of formulas within formulas.
- Built-in mathematical functions (trig, logs, conversions, etc.).
- Easy to create and edit variable sets and values.
- Disk storage and recall of user-defined formulas and variable sets.
- Automatic retention of variable values for chain calculations.
- Free use of "Ask Variables" to allow "What if" and repeat types of calculations.
- Rapid calculation or stepped calculation for viewing intermediate results.
- Full hard copy support, including a trace function.

MatheMagic has been released on the Apple II and II+ and Z80 based micros running under CP/M 2.2 ISM plans to release versions for the Atari 800, TRS-80 I, II, and III, Commodore PET/CBM and the IBM PC during January and February 1982.

ISM's Home Offices are located at:

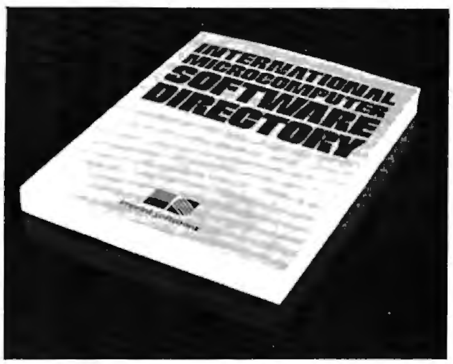
International Software Marketing, Ltd.
Suite 421, University Building
120 E. Washington Street
Syracuse, New York 13202
(315)474-3400

The International Software Directory

The International Software Directory lists tens of thousands of software packages and classifies them by machine, operating system, subject, vendor and price. Review information is also included. It is independent of both machine manufacturers and software houses, and thus allows for comparison and one-stop shopping. It is backed by an excellent search service and regular updates are available from the publisher. In addition, an on-line service is available through Lockheed dialog.

The ISD is available in two volumes:

1) The International Microcomputer Software Directory which contains over 7000 programs and



is available at the extremely competitive price of $34.95 plus $2.95 postage and handling.

2) The International Minicomputer Software Directory which is available for $65.00

Imprint Editions
420 South Howes
Fort Collins, CO 80521
(303) 493-2710

Commodore Canada Announces New Mailing List Software

Toronto – A versatile new electronic mailing list software package called Scratchpad has been announced by Commodore Business Machines Limited.

Intended for business applications, the program is easy to use, flexible and rapid in accessing required data. There are 20 different data items, each with a 30-character length. A great deal of data may be stored on file as a file can consist of any number of disks.

Some of the modes of operation in Scratchpad are Edit, Global Search, Format, Recreate, Print and Backup.

Any data created with the Scratchpad, developed by Richvale Telecommunications of Richmond Hill, Ont., may be interfaced with Wordpro software by the use of a small additional program.

Suggested retail price in Canada is $295. For further information:

Isobel McBurney
Software Manager
Commodore Business Machines Limited
3370 Pharmacy Avenue
Scarborough, Ont. M1W 2K4.
(416)449-4292

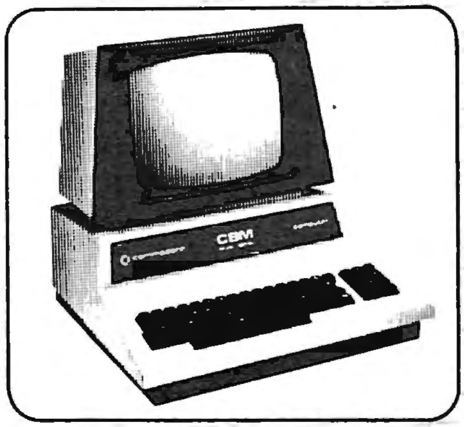
Dragon's Eye Now Available

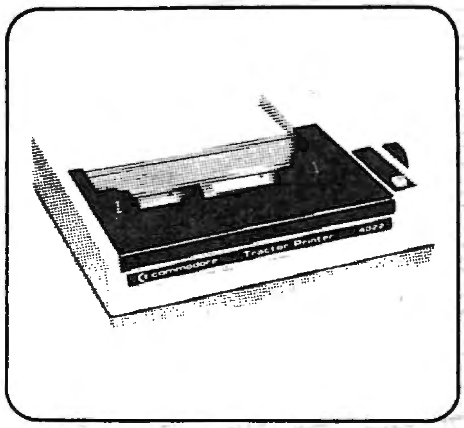
Mountain View, CA – Dragon's Eye, an EPYX game from Automated Simulations, Inc., is now available for the Atari 400 and 800 computers.

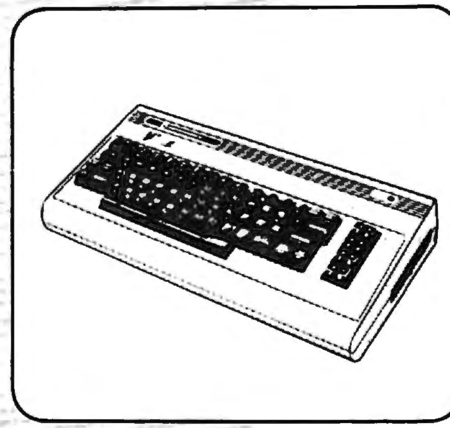
Dragon's Eye, an overland fantasy role playing game, challenges the player to find a magical gem, the Dragon's Eye, in 21 game-days (approximately half-an-hour playing time).

commodore
SHOW SPECTACULAR

8032-32K 80 COL CRT	REG $1495	$1135
64K ADD-ON MEMORY	REG $500	$395
9000 134K SUPER PET	REG $1995	$1795
4032 32K 40 COL CRT	REG $1295	$995
4016 16K 40 COL CRT	REG $995	$795
8050-DUAL DISK 950K	REG $1795	$1395
4040-DUAL DISK 343K	REG $1295	$995
2031-SINGLE DISK 170K	REG $695	$555
C2N-CASSETTE DRIVE	REG $75	$65
4022-80 COL PRINTER	REG $795	$649
8023P-136 COL PRINTER	REG $995	$849
8300P-40CPS LTR QLTY	REG $2250	$1995
8024-MANNESMAN TALLEY	REG $1995	$1595
8024L-LETTER TALLEY	REG $2495	$1995
25CPS-STARWRITER	REG $1895	$1445
CBM-IEEE MODEM	REG $279	$229
VOICE SYNTHESIZER	REG $395	$329







PET TO IEEE CABLE	REG $39.95	$34
IEEE TO IEEE CABLE	REG $49.95	$39
VIC 20	REG $299	$269
VIC 1540 DISK 170K	REG $599	$499
VIC 1515 30CPS PRINTER	REG $395	$349
VIC 1011 RS 232 INTER	REG $49.95	$39
VIC 1112 IEEE INTER	REG $99.95	$79

MATOR SHARK HARD DISK DRIVE

*Winchester Disk
*Commodore DOS 2.5
*24 Megabyte
$6475

Strobe 100 Plotter
*PET Interface
*Software Pack
$975

SCHOOL SPECIALS
3 for 2 Pricing
CALL US!

20% Off on Software

Special pricing on Atari, Apple, Digilog, Epson, Hayes Modem, Printwheels

PHILADELPHIA COMPUTER DISCOUNT©

P.O. Box 170 St. Davids, PA 19087 (215)687-8540

MAIL ORDER PHONE 1-800-345-1289

PREPAID ORDERS SHIPPED FREE
VISA & MASTERCARD ADD 3%
COD · UPS
H1181C PA RESIDENTS ADD 6% SALES TAX

The Eye is hidden somewhere within one of the seven provinces, and the player must find it and return it to Fel City, where his journey began.

The player chooses one of 16 characters and gains a set of magical abilities, such as healing, flying, time travel and teleport. Which spells he gets are different each time he plays. He is equipped with his choice of four swords, a bow and arrows, and magic bolts.

While searching, the player sees a detailed map of the provinces on the screen, along with his location, strength, health, and other information.

The player encounters dragons, bats, vampires, ghosts, golems, serpents, skeletons and other monsters. He can choose between 13 commands, from firing an arrow and fighting with sword to casting magic spells and searching for hidden paths.

When a battle is engaged, fully animated graphics display the action between player and beast.

Dragon's Eye is available on cassette for the ATARI 400/800 (32K) and PET (32K), or on disk for the APPLE (48K with ROM Applesoft), and ATARI 400/800 (32K), from:

Automated Simulations, Inc.
P.O. Box 4247
Mountain View, CA 94040
$24.95

Commodore Introduces Lowest Priced Modem In The Computer Industry

Valley Forge, PA, January 7, 1982 – A low priced modem has been introduced by Commodore Business Machines, Inc.

The new "VICMODEM," which retails for $109.95, is an easy-to-use plug-in cartridge that connects directly to the user port of Commodore's VIC 20™ home computer, and may be used with any modular style telephone.

The VICMODEM, which is planned for retail sale in the Spring of 1982, allows users to communicate and exchange data with other computer owners over the telephone. This latest VIC peripheral also allows users to inexpensively access telecomputing networks such as the Source™ or Compuserve™, which provide services such as stock quotes and company reports, newswire stories, research data, sports scores, airline reservations, shopping services and more.

The VICMODEM is a direct connect, 300 baud modem with originate/answer and half/full duplex capabilities. The combined cost of the VIC 20 and a VICMODEM is less than $410.00.

Microsette Introduces Mini-Diskettes

Microsette Co. is adding 5¼" diskettes to its line of computer and audio cassettes. The diskettes are useable with Apple, Atari, Commodore, Tandy TRS-80, PMC-80 and PMC-81 computers from Personal Micro Computers and many mini-computer and word processing systems.

The diskette surface is certified error free on one side and is for all soft-sectored, single or double density applications. The burnished diskette surface insures longer head life and excellent media durability. Furthermore, the diskettes feature a reinforced hub which gives longer diskette life in this high wear area.

Microsette MD-5 diskettes are attractively packaged in a sturdy green box with each of the 10 diskettes in a long-lasting Tyvek envelope. Each diskette has a corner label and each box of 10 contains 20 silver write

protect tabs and 20 large color coded labels.

Microsette MD-5 diskettes are sold by mail order in units of 10 (1 box) or 50 (5 boxes). Suggested list price is $3.95 per diskette and mail order prices are $2.50 each in the box of 10, or $2.20 each in the box of 50. Prices include UPS shipping.

Further information and dealer pricing may be obtained from Microsette.

Microsette Co.
475 Ellis Street
Mountain View, CA 94043
(415)968-1604

Five New Programs From Atari

Sunnyvale, CA – January 5, 1982 – Three new home computer game programs, a bookkeeping package and a home filing system were introduced by the Home Computer Division of Atari, Inc.

Pac-Man is the Atari Home Computer version of a very popular coin-operated game. In this version, a player's character must negotiate a maze without being eaten by any of four pursuers. By eating an "energy" dot the player's character can gain the ability to attack and gobble up the pursuers. For use on both the Atari 400 and Atari 800 Home Computers, this $44.95 game comes in cartridge form, and will be available in the second quarter of 1982.

Centipede is another Atari Home Computer version of a best-selling coin-operated game. In this version, the player uses a "Bug Blaster" to defeat colorful legions of attacking spiders, fleas, scorpions, the Centipede itself, and poisonous mushrooms. The $44.95 game can be used on both Atari 400 and Atari 800 Home Computers. It comes in cartridge form and will be available in the second quarter of 1982.

Caverns Of Mars is a fascinating new game that takes players beneath the surface of Mars. The player's character must penetrate several layers of defenses to reach the Aliens' stronghold. Floating mines, "enemy" ships on patrol and deadly laser fire menace the player constantly. This game was originally developed for the Atari Program Exchange and won a prize in the APX quarterly software contest for its 17-year-old author. This $39.95 game can be used on both the Atari 400 and Atari 800 Home Computers with an Atari 810™ Disk Drive. It comes in diskette form and will be available in the first quarter of 1982.

The Bookkeeper is an accounting system designed to meet the needs of people who do business from their homes and for those who run their personal finances in a business-like fashion. It is a low-cost, comprehensive accounting system that generates professional calibre reports, such as Profit and Loss Statements, Balance Sheets, Accounts Receivable and Payable. It will handle 1,000 transactions a month and up to 350 General Ledger accounts, vendors and customers.

The Bookkeeper consists of four diskettes, including a sample company data diskette that lets you familiarize yourself with The Bookkeeper before you start entering data of your own. The package requires an Atari 800 Home Computer with Atari BASIC language cartridge and 48K of Random Access Memory, an Atari 810 Disk Drive, an Atari 825™ 80-Column Printer and an Atari 850™ Interface Module.

Price and availability will be announced later.

The Home Filing Manager lets a user create, store, edit, retrieve and print information similar to that found on 3"x5" index cards. It provides a convenient way to catalog books, addresses, clothes, recipes, record collections, Christmas card lists, vocabulary words, term paper notes, and the like.

The user can search in alphabetical or reverse alphabetical order or locate an entry by its title or a phrase it contains.

Price and availability will be announced later.

Scheduled Educational Conferences

National Educational Computing Conference (NECC-82)
Kansas City, MO
June 28-30, 1982

The purpose of the conference is to provide a forum for discussion among those interested in educational computing. Based on previous conferences, approximately 1000 people from institutions at all levels are expected to attend. Between 50 and 75 vendors are expected to exhibit at the conference.

Papers will be presented which describe actual experiences with computer use in the classroom or consequences of such use on the educational process in general. The diversity of disciplines and participation by individuals from elementary, secondary, and post secondary education provide a unique opportunity for cross-pollination of ideas and experiences. Additional conference activities include pre-conference workshops, project presentations, vendor exhibits, special sessions, tutorials, and birds-of-a-feather sessions.

Additional information may be obtained from:

E. Michael Staman
NECC-82 General Chairman
Campus Computing Services
University of Missouri-Columbia
305 Jesse Hall
Columbia, MO 65211

****

Computers in Education Conference
Seattle Pacific University
Seattle, WA
March 12 & 13, 1982

The annual conference will include talks, workshops, and exhibits with emphasis on the use of the microcomputer in K-12 classrooms of various disciplines.

Tony Jongejan
Everett High School
2416 Colby
Everett, WA 98201

****

Microcomputer Conference
The University of Victoria
Victoria, British Columbia
May 6-8, 1982

Moving Microcomputers Into The Mainstream Of Education will be an opportunity for both a formal and an informal sharing of ideas, information and experiences amongst educators. The major focus of the conference will be the integration of the microcomputer into the established curriculum into the classroom, and into the administrative offices of school districts.

The conference is designed to provide educators with information regarding applications that have been tried and tested in the schools to achieve this purpose. A series of keynote addresses, workshops, papers, and short presentations have been planned. A call for presentations can be found in the upcoming issue of Micro-scope.

The conceptual threads of the conference are:

1. Classroom applications.
2. Administration applications.
3. Courseware development and evaluation.
4. Future implications of technology in instruction.

University housing will be available at reasonable rates.

University Extension
Conference Office
721-8475

****

Southern California Computers in Education Conference
University High School
Irvine, CA
May 14-15, 1982

This second annual conference is sponsored by Computer-Using Educators.

Workshops and field trips will be held on Friday, May 14 at schools and industry scattered throughout Southern California. On Saturday all sessions will be held at University High School, Irvine starting at 9:00 a.m.

Classroom applications of computers to all areas of the curriculum will be presented covering all grade levels from kindergarten through two-year college. The major emphasis of the conference will be on getting started with microcomputers.

The Friday night banquet speaker will be Prof. Lud Braun, School of Engineering, University of New York. The keynote address on Saturday morning will be delivered by Dr. William "Sandy" Wagner.

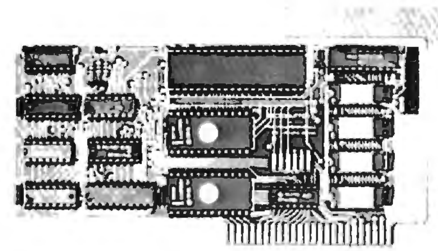
Preregistration is $10 for CUE members and $16 for nonmembers. Preregistration must be received by April 30th, 1982. *No purchase orders accepted.* On-site registration will be available on Saturday for $18. All checks should be made out to CUE (or Computer-Using Educators) and mailed to the address below.

One unique feature of the conference will be several computer workshops held at sea. The unofficial slogan of the conference is "PET your APPLES in Orange County."

Craig Walker
1982 Conference Chairperson
Arrowview Intermediate School
2299 North "G" Street
San Bernardino, CA 92405
(714)886-9118

Buffered Parallel Printer Interface

The Microbuffer II is an intelligent buffered parallel printer interface for the Apple II computer. The Microbuffer II includes 16K of on-board RAM so the computer does not have to wait for the printer to finish before continuing with other



processing. Buffer memory size is user expandable to 32K using industry standard 64K RAM chips. The Microbuffer II is compatible with Applesoft, CP/M, and Pascal and includes advanced high resolution graphics print routines for the Epson MX-80 with Graphtrax, Anadex, IDS Paper Tiger, NEC Prowriter, as well as complete print formatting features. The Microbuffer II is available for $259.

Practical Peripherals, Inc.
31245 La Baya Drive
Westlake Village, CA 91362
(213)991-8200

MIT LOGO For Apple Computers

The Krell Software Corp. Stony Brook, NY is pleased to announce that *MIT's LOGO* for Apple II computers is now available. This is the authorized version of the LOGO language for Apple developed by MIT under the sponsorship of the National Science

Foundation.

This unique and powerful language was written specifically for educational application. It is highly graphic and ideal for introducing young people to the use of microcomputers. *MIT LOGO For Apple* is by no means an abridged version suitable to children only, it is the full programming language whose capabilities and versatility rival and at many places, supersede those of BASIC, FORTRAN, and PASCAL.

Krell's complete package includes the LOGO language system, Krell's own Instant LOGO tutorial program, and its unique introduction to LOGO, *Alice In LOGOLAND.* The entire package is fully documented for teachers and students. Since LOGO requires a disk system with 64K of memory an optional 16K RAM board extension is available at the special price of $109.95. The entire software package is available for $179.95.

Krell Software Corp.
21 Millbrook Drive
Stony Brook, NY 11790
(516)751-5139

Atari Special Additions

Sunnyvale, CA – January 6, 1982 – One hundred and seventeen vendors are represented with more than 400 products in *Atari Special Additions*, the first edition of a new catalog of programs, equipment accessories, furniture and publications designed for users of Atari 400 and Atari 800 Home Computers.

Atari Special Additions is intended to be a useful guide to equipment, and programs produced by other manufacturers," said Roger H. Badertscher, president of the company's Home Computer Division. "It is a measure of the continuing success of our product line that so many vendors are offering so many products to be used with Atari Home Computers."

The catalog, with a cover price of $3.00, is available from Atari Computer retailers, and will be mailed free to owners of Atari Home Computers who have filed warranty cards and to subscribers of *The Atari Connection*, the quarterly magazine the company publishes for owners of its computer.

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.

CAPUTE!:

Corrections And Amplifications

1. "Spacewar Part 2," *Home and Educational COMPUTING!*, Fall 1981, pg. 21: an ?Out Of Memory error can occur after several rounds of the game. The following changes will prevent this as well as the problem with the black hole option:

```
6 SX=50:SY=50:POKEV-9,255:?"[CLEAR]":X=S+A:
  FORI=XTOX+505:POKEI,T:NEXTI
```

And change lines 18 and 45 to contain GOTO 47 (instead of GOSUB47).

Change lines 59 and 70 to contain GOTO17 (instead of RETURN).

Changing line 140 to: 140 POKE52,28: POKE56,28:CLR will allow you to make modifications and debug the program without running out of memory. (Our thanks to Ken Denniston and Ronald Gruenzel for these suggested improvements.)

2. "Window Analysis," **COMPUTE!**, December, 1981, pg. 35, line 720: GOSUB FNTRC should read GOSUB FTRC.

3. "A Simple Printer Interface For Apple II," **COMPUTE!**, December, 1981, pg. 85: 220 OHMS should be 2200 OHMS (and vice versa).

4. "PET To PET Communication Over The User Port," **COMPUTE!**, December, 1981, pg. 142: The second paragraph on the second column of this page should begin, "Now the receiver can leave line 40 and read the data byte (on line 50). The transmitter is now stuck on line 90 . . ."

5. "File Recovery," **COMPUTE!**, December, 1981, pg. 164: to apply Program 1 to the 2040 Disk Drive, the following lines are required.

```
145 PRINT "ENTER TRACK NUMBER OF 1ST
    DATA BLOCK": INPUT TT
414 PRINT#15,"B-P:"3;5+32*R-2
416 PRINT#1,CHR$(TT);
```

6. "The Beginner's Page," **COMPUTE!**, January, 1982, pg. 24: part of the Atari version of the program was missing. Here it is in its entirety:

```
100 DATA SUPER,ACME,AMERICAN,RAINBOW,QUA
LITY,INTERGALACTIC,RELIABLE,FOOLPROOF
110 DATA PROGRAMS,SOFTWARE,COMPUTERWARE,
CODE,LISTINGS,INFORMATION,MAGIC
120 DIM ADJECTIVE$(8*20),NOUN$(7*20),TEM
P$(20),L1(8),L2(7)
130 FOR I=1 TO 8:READ TEMP$:ADJECTIVE$(I
*20-19,I*20)=TEMP$:L1(I)=LEN(TEMP$):NEXT
 I
140 FOR I=1 TO 7:READ TEMP$:NOUN$(I*20-1
9,I*20)=TEMP$:L2(I)=LEN(TEMP$):NEXT I
150 FOR I=1 TO 7
160 FOR J=1 TO 8
170 PRINT ADJECTIVE$((J-1)*20+1,(J-1)*20
+L1(J));" ";NOUN$((I-1)*20+1,(I-1)*20+L2
(I))
180 NEXT J
190 NEXT I
```

7. "Invest," **COMPUTE!**, January, 1982, pg. 39: the author suggests that the following changes be made to his program since equity buildup for an investment should be multiplied by the initial amount financed rather than the yearly payment amount. As the program stands, it makes an investment look a little better than it really is. The following will correct the problem:

```
7490 PN = PR - DP : REM THIS IS THE AMOUNT
     YOU'RE FINANCING
7500 EB = PN * E1 (Y,I) : REM THIS IS THE EQUITY
     BUILDUP FOR 12 MONTHS, 1ST YEAR
7502 EB = EB/12
7503 EB = EB * M : REM THIS IS THE EQUITY
     BUILDUP FOR THE MONTHS YOU'LL OWN
     IT, THE 1ST YEAR
7505 EB = EB/1000
7510 ET = PN * E2 (Y,I) : REM EQUITY BUILDUP
     FOR THE 2ND YEAR
7515 ET = ET / 1000
```

8. "Apple Addresses," **COMPUTE!**, January, 1982, pg. 83: the program was missing from this article. The entire article is being reprinted in this issue, starting on page 163.

9. "Tinymon1," **COMPUTE!**, January, 1982, pg. 176: Jim Butterfield writes, "I bungled it! In my zeal to make the whole thing as painless and foolproof (!) as possible, I added four bytes to the end of the program. This made the checksum straightforward, and made the hex entry a little simpler.

All of this would have been OK had I not added four bytes to the BASIC pointers, too. The 'first change' should have been:

```
.: 0028 01 04 14 08 14 08 14 08
```

If you punched the whole thing up and it doesn't work, all is not lost. Reload the program, make the above change, go back to BASIC and SAVE...and you will find yourself with a working Tinymon.

My apologies for all those nights of lost sleep. I really was trying to make it easier and bug-free...."

MICRO BUSINESS WORLD INC.

Information Line (213) 996-2252
TOLL FREE MAIL ORDER LINES
(800) 423-5886 Outside Calif.

HEWLETT PACKARD HP 85 — Special Of The Month

OUR PRICE $1995.00

The HP85 — MSL 3250.00 — Save 1255.00

We Turn Problems Into Solutions

HP DISC DRIVE — Single Master Model 82902M — Our Price $1099.00 — MSL $1500.00 — Save 401.00

Dual Master Model 82901M — $1849.00 — MSL $2500.00 — Save 651.00

SHARP BUSINESS COMPUTER 64K SYSTEM

- 5.25" diskette
- 284K byte per disk drive
- 2 disk drives per unit
- Expandable up to 4 units
- Including SHARP's FDOS (Floppy
- Disk Operating System)

Complete System Including Printer

MSL 5995.00 — OUR PRICE $3995.00 — Save $2000.00

NEC STUDENT SYSTEM 64K

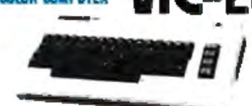
- NEC PC 8001 A Computer
- NEC PC 8012 A Expansion
- NEC PC 8031 A Dual Drive
- 12" Grn. Phs. Video Monitor

MSL 3565.00 — Our Price $2395.00 — Save $1170.00

XEROX 820 WORD PROCESSING SYSTEM (INCLUDES SOFTWARE)

MSL 3495.00 — OUR PRICE $2649.00 — Save 846.00

commodore COLOR COMPUTER VIC-20



Vic TV Modual	$19.00
Vic Cassette	$69.00
Vic Disk Drive	$ Call
Vic 6 Pack program	$44.00

MSL 299.00 — OUR PRICE $255.00 — Save 44.00

HEWLETT PACKARD

	List	Our Price
HP-125 Microcomputer	3750.00	2750.00
HP-85 Microcomputer	3250.00	2475.00
HP-83 Microcomputer	2250.00	1777.00
16K Exp-Memory Module	295.00	255.00
Graphics Plotter 7225	2450.00	2075.00
Personality Mod. For 7225	750.00	665.00
2631B Impact/Printer/ Hvy Dty	3950.00	3250.00
Option 020 For 2631B	150.00	125.00
8 Drives to Choose from 82902S	1300.00	1125.00
9895A 8" Dual Drive	6850.00	5500.00
Graphics Tablet 9111A	2050.00	1678.00
HP-11C Slim-Line Advanced	135.00	119.00
HP-12C Slim-Line Financial	150.00	129.00
HP-41 CV New 2.2 Bytes Mem	325.00	250.00
HP-41 C Calculator	250.00	185.00
Card Reader For 41CV/C	215.00	162.00
Printer For 41CV/C	385.00	289.00
Optical Wand For 41 CV/C	125.00	97.00
Quad Ram Equals 4 Mem. Mods	95.00	81.00
Memory Modules For 41C		25.00
HP-97 Programble Printer	750.00	595.00
HP-67 Programble Calculator	375.00	295.00
HP-34C Programble Scientific	150.00	117.00
HP-38C Programble Bus. R/E	150.00	117.00
HP-32E Adv. Scientific	55.00	48.00
HP-37E Business Mgmt.	75.00	57.00

commodore

	List	Our Price
8032-32K 80 Col Crt	1495.00	1097.00
4032-32K 40 Col Crt	1295.00	989.00
4016-16K 40 Col Crt	995.00	788.00
8050-Dual Disk 950K	1795.00	1344.00
4040-Dual Disk 343K	1295.00	989.00
C2N-Cassette Drive	95.00	77.00
4022-80 Col Printer	795.00	639.00
8024-Mannsman Talley	1995.00	1545.00
25CPS-Starwriter	1895.00	1399.00
CBM IEEE Modem	279.00	225.00
Voice Synthesizer	395.00	325.00
VIC 20	299.00	255.00
PET to IEEE Cable	39.95	34.00
IEEE to IEEE Cable	49.95	39.00

XEROX

	List	Our Pr
Xerox 820-1 5-1/4" Disk	2995.00	2399.00
Xerox 820-2 8" Disk	3795.00	2995.00
CP/M Operating System	200.00	169.00
Word Processing Software	500.00	429.00
Super Calc	295.00	249.00

ATARI PERSONAL COMPUTERS

	List	Our Price
400 16K Bytes of Ram	595.00	337.45
800 16K Bytes of Ram	1080.00	739.00
410 Program Recorder	90.00	77.00
810 Disk Drive	600.00	457.00
825 Printer (80 col Centronic 737)	999.95	769.00
820 Printer (40 col impact)	450.00	353.00
830 Acoustic Modem	200.00	155.00
850 Interface Module	220.00	192.00
Atari Visicalc	200.00	164.00

TEXAS INSTRUMENTS

	List	Our Price
TI 99/4A Console New Improved	950.00	369.00
10" Color Monitor High Resolution	399.95	339.95
32K Memory Module	399.95	312.95
Speech Synthesizer	149.95	127.95
Disk Memory Drive	499.95	390.95
RF Modulator	49.95	42.50
Telephone Coupler (Modem)	224.95	185.95
Printer (Solid State)	399.95	315.95

PRINTERS

	List	Our Price
EPSON		
MX 80 FT	$ 745.00	$ 549.00
MX 80 IMPACT	645.00	450.00
MX 70 IMPACT	500.00	390.00
MX 100	995.00	749.00
ANADEX 9501	1650.00	1299.00
NEC		
5510	3195.00	2445.00
5515	3295.00	2545.00
3510	2495.00	1795.00
3515	2545.00	1849.00
OKIDATA		
MICROLINE 80	545.00	395.00
MICROLINE 82	649.00	549.00
MICROLINE 83	1050.00	769.00
PAPER TIGER		
445G with Graphics	795.00	695.00
460G with Graphics	1,394.00	899.00
560G New full size	1,695.00	1,139.00
DIABLO (LETTER QUALITY)		
630 R102 bi-directional tractors	2,965.00	2,450.00
1640K109 keyboard tractors	4,000.00	2,899.00
630 RO Receive Only	2,710.00	2,250.00
1650K 136 keyboard tractors	4,000.00	3,100.00

MONITORS

	List	Our Price
BMC		
12" Grn. Phs KQ (15 Hz)	219.00	165.00
12" Grn. Phs EQ (18 Hz)	$249.00	185.00
12" Grn Phs (20 Hz)	279.00	209.00
12" Colour Composite Hi Res	439.00	319.00
Grn. Phs 12"	275.00	165.00
NEC Grn. Phs. 12"	225.00	179.00
Zenith 12"	159.00	119.00
LE MONITOR		
* 9" B & W	179.00	159.00
* 9" Green Phs	189.00	165.00
* 12" B & W	209.00	185.00
* 12" Green Phs.	229.00	199.00
** 12" Green Phs.	199.00	159.00
* Anti Reflective Screen		
** T.V. Grade Screen		
SANYO MONITORS		
13" Color (new) high quality	$550.00	399.00
12" Green Phs	360.00	269.00
12" B & W	340.00	239.00
15" B & W	370.00	255.00
9" B & W The Best Seller!	235.00	159.00
9" Green Phs	275.00	199.00

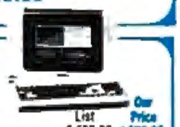
1 Year Extended WARRANTY

$99.00

INQUIRE

$ — SAVINGS, SERVICE, SELECTION, SATISFACTION

TERMINALS — TELEVIDEO



	List	Our Price
910	$ 699.00	$ 589.00
912C	950.00	699.00
920C	995.00	795.00
950C	1,195.00	949.00

DISKETTES SOLD IN BOXES OF 10

(Min. Purchase) $100

	List	Our Price
DYSAN PRICE PER DISKETTE		
104/1 5" SOFT SECTOR	6.00	3.99
104/1D " " DBL. DEN. SOFT SEC.	6.40	4.60
3740/1 8" SOFT SECTOR	7.25	4.75
3740/1D 8" DBL. DEN. SOFT SECTOR	10.75	6.90
MAXELL		
MD-1 5" SOFT SEC-TOR/DBL. DEN.	5.00	3.50
MD-2 " SOFT SECTOR/DBL. SIDE/DBL. DEN	7.00	4.90
FD-1 8" SOFT SEC./DBL. DEN.	6.50	4.50
FD-1 8" SOFT SEC./DBL. SIDE/DBL. DEN	8.50	5.95

DRIVES

	List	Our Price
CORVUS		
5 MBYTES	3,750.00	$3,050.00
10 MBYTES	5,350.00	4,449.00
20 MBYTES	6,450.00	5,325.00

EXPANSION BOARD

Q STAR 16K RAM BOARD

LIST 199.00 — OUR PRICE 129.00

SOFTWARE FOR APPLE II

	List	Our Price
Language System with Apple Pascal	495.00	399.00
BPI General Ledger System	395.00	319.00
Visidex	200.00	159.00
Visicalc	200.00	159.00
Desktop Plan II	200.00	159.00
Microlab Database System	150.00	119.00
Stoneware DB Master	229.00	189.00
Muse Supertext II	150.00	119.00
Softape Magic Window	99.00	72.00

SOFTWARE FOR COMMODORE

	List	Our Price
Ozz The Information Wizard	395.00	$289.00
Wordcraft 80	395.00	289.00
Irma-Info Retrieval & Mgmt. Aid	495.00	389.00
Dow Jones Portfolio Mgmt.	149.00	119.00
Pascal Development Pkg.	295.00	219.00
Ebs-Receivables, Inventory	750.00	569.00
Bpi-General Ledger	395.00	309.00
Word Pro 3-40 Column	250.00	175.00
Word Pro 4-80 Column	375.00	299.00
Word Pro 4 Plus	450.00	319.00

MODEMS

	List	Our Price
NOVATION CAT MODEM	189.95	140.00
NOVATION D-CAT	199.00	150.00
NOVATION APPLE-CAT	389.00	329.00
HAYES MICROMODEM	379.00	295.00
HAYES SMARTMODEM	279.00	239.00

CALCULATORS

	List	Our Price
CASIO		
HR-10 Paper Feed	49.95	39.00
HR-12 Paper Feed	54.95	42.00
FR-100 Paper Feed	79.95	59.00
FR-1210 Paper Feed	129.95	99.00
PQ-20	29.95	23.00
LC-785	12.95	10.00
LC-3165	12.95	10.00
FX-68 Scientific	29.95	23.00
FX-81 Scientific	19.95	17.00
FX-3600P Scientific	39.95	29.95
FX-602P "Computer Talk" 88 Memories Programming Upper & Lower Case Dot Matrix 512 Step	129.95	99.00
FX-702P Solves Problems with Alpha-Numeric Clarity, uses Basic Language	199.95	159.00

TELE. ANSW. DEVICES

	List	Our Pr
PHONE MATE		
900 Economy	119.95	86.00
905 Remote Dual Tape Sys	199.95	144.00
910 60 Calls Per Side	159.95	115.00
920 C Vox	199.95	144.00
925 Remote Call Monitor	239.95	173.00
930 Remote Twin Cass	299.95	216.00
950 Remote	339.95	245.00
960 Remote Built in Tele Microcomputer Controlled	399.95	288.00

MICRO BUSINESS WORLD WAREHOUSE — MAIL ORDER

18720 Oxnard, #108
Tarzana, CA 91356

OUTSIDE CA CALL TOLL FREE 1 (800) 423-5886 IN CA (213) 996-2252

Telex: 182852
Answer: MICKO TZNA

Name (Please print) ______________________
Address ______________________
City ____________ State ________ Zip ________

Qty	Make	Model	Description	Price	Total

*California residents add 6% sales tax

**Add 3% Shipping & Handling — Add 3% surcharge for credit cards. Orders cannot be shipped unless accompanied by payment, including shipping, handling and tax where applicable

TOTAL ORDER $ ________
TAX IF APPLICABLE* ________
SHIPPING & HANDLING** ________
TOTAL ENCLOSED $ ________

☐ Certified Check or M.O. ☐ Allow 2 weeks clearance for personal check
☐ Bank Wire Transfer

VISA — MasterCard

CREDIT CARD # ______________________
Exp. Date ____________ Signature ______________________

Advertisers Index

AB Computers ... 66,67,183
AG Associates ... 222
Aardvark Technical Services, Ltd. ... 205
Abacus Software ... 129,169
Actek ... 123
Alternate Reality Software ... 85
Anderson Peripherals, Inc. ... 121
Applefest ... 49
Arcade Plus ... 109
Artworx Software Company ... 79
Atari, Inc. ... 7
Automated Simulations ... 69
Axlon ... 123
Barron Enterprises ... 71
Batteries Included ... 183
John Bell Engineering ... 162
Beta Computer Devices ... 140
The Bit Bucket ... 216
R. J. Brachman Associates, Inc. ... 181
Byte Microsystems Corporation ... 179
C-Mart ... 91
CE Software ... 116
CFI ... 155
CGRS Microtech ... 187
CMS Software Systems ... 2,3
C&O Micro Systems, Inc. ... 185
Canadian Micro Distributors Ltd. ... 24,25,27,29,31
Cansoft Data, Inc. ... 23
Color Computer Concepts ... 191
Comm*Data Systems, Inc. ... 169
Commodore Computer Systems ... BC
COMPUTE!'s First Book of Atari ... 11
COMPUTE!'s First Book of PET/CBM ... 10
Computer Age Software ... 131
Computer Country ... 150
Computer House ... 145
Computer Mail Order ... 210,211
Computer Shopping Center ... 220,221
The Computerist's Directory ... 61
Computer Mat Software ... 175,185
Computertime, Inc. ... 42
Connecticut microComputer, Inc. ... 95
Consultors International ... 42
Cow Bay Computing ... 61,83
Creative Software ... 34
Cursor, The Code Works ... 116
Cyberia Inc. ... 153
Dr. Daley's Software ... 197
Data Equipment Supply Corp. ... 86,87
Dilithium Press ... 99
Dunham Software & Consulting Co. ... 191
Dynacomp ... 46,47
ECX Computer Company ... 154
Eastern House Software ... 127
Eclectic Systems Corporation ... 217
Elcomp Publishing, Inc. ... 89
Electronic Specialists, Inc. ... 99
Electronic Technology Corporation ... 199
Execom Corp. ... 185
FSS ... 154
GOLF ... 189
Gebelli Software, Inc. ... 13
HW Electronics ... 133
High Country Microsystems ... 151
Horizon Simulations ... 39
Human Engineered Software ... 114
Huntington Computing ... 125
IDSI ... 21
Inhome Software ... 135
Intec Peripherals Corp. ... 147
Interlink, Inc. ... 95
Iridis, The Code Works ... 147
Jini Micro-Systems, Inc. ... 19
Krell Software Corp. ... 101
LJK Enterprises Inc. ... 51
Leading Edge ... IBC
LemData Products ... 189
Lyco Computers ... 215
MDS ... 113
MED Systems Software ... 157
MIS ... 175
Madison Computer ... 129
Magic Carpet ... 189
Micro Business World Inc. ... 223
Micro Printer Marketing ... 32
Micro Spec Ltd. ... 175
Micro Technical Products, Inc. ... 118
Micro-Ed, Inc. ... 97
Micrograms, Incorporated ... 89
The Micromint Inc. ... 140
Microsoft Consumer Products ... 4
Microtek Inc. ... 41
Miles Computing ... 147
Mosaic Electronics ... 107
Mountain Computer ... IFC
Muse Software ... 162
National Computer Shows ... 59
Netronics R&D Ltd. ... 181
New England Electronics Company ... 36,37
Nüfekop ... 169
Olympic Sales Company ... 127
On Line Systems ... 18,137
Optimal Technology Inc. ... 65
Optimized Data Systems ... 185
Optimized Systems Software, Inc. ... 35,145
Orion Software ... 208
Oryx Software ... 177
PSK Evaluation Service ... 189
Pacific Exchanges ... 108,119,169,200
Pendulum Software ... 94
Petted Microsystems ... 196
Philadelphia Computer Discount ... 213
Poquette's ... 193
PractEd Tapes, Inc. ... 187
Professional Software ... 1,9
Program Design, Inc. ... 157
The Program Store ... 57
The Programmer's Institute ... 42
Protronics ... 181
Quality Software ... 83
Quantum Data, Inc. ... 169
Questar International Inc. ... 195
RC Electronics Inc. ... 177
Samurai Software ... 45
Santa Cruz Educational Software ... 81
Sebree's Computing ... 147
Silicon Valley Systems Inc. ... 143
Skyles Electric Works ... 63,187
Small System Design ... 177
The Software Connection ... 73
Software Galore ... 191
Software Street ... 161
Spyder Software ... 216
Strategic Simulations, Inc. ... 15
Street Electronics Corporation ... 162
Sunrise Software ... 151
Supersoft ... 209
Swifty Software, Inc. ... 71
Syncro, Inc. ... 17,151
Synergistic Software ... 77
T.H.E.S.I.S. ... 18
TIS Inc. ... 104
T'Aide Software Company ... 103
United Microware Industries, Inc. ... 105
Unicomm, Inc. ... 219
Vixel, The Code Works ... 85
Voicetek ... 155
Wildfire Publishing ... 193

The 6502 Resource Magazine

COMPUTE! PET•ATARI•APPLE OSI•KIM•SYM•AIM

For Fastest Service, Call Our **Toll-Free** US Order Line **800-345-8112** **In Pennsylvania call 800-662-2444**

My computer is:
☐ **PET** ☐ **APPLE** ☐ **ATARI** ☐ **OSI**
☐ **KIM** ☐ **SYM** ☐ **AIM** ☐ **OTHER** ________
☐ **Don't yet have one...**

☐ Please enter my 1 year (12 issue) subscription to **COMPUTE!**
☐ New subscription ☐ Renewal subscription
☐ $20.00 US subscription
☐ $25.00 Canada and international surface mail Payment in US funds, drawn on US bank or money order
☐ Bill me (US only) $1.00 billing fee

Name, Address: ______________________

Charge my: ☐ Visa ☐ MC
Number______________ Expires____/____

3 4 5 6 7 8 9 10 11 12

The Editor's Feedback:

Computer: ☐ PET ☐ ATARI ☐ APPLE ☐ VIC-20 ☐ OSI ☐ OTHER ________

Are you a **COMPUTE!** Subscriber? ☐ Yes ☐ No I would like to see:

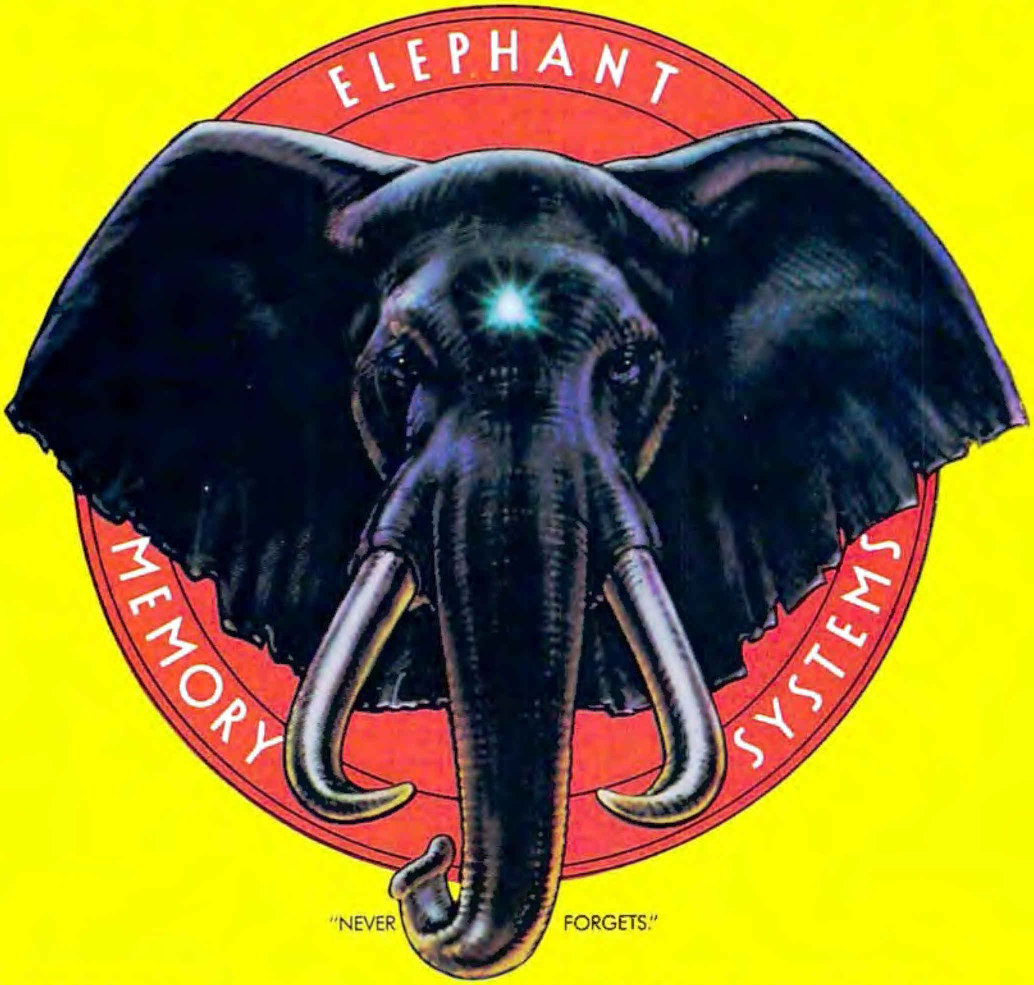
☐ More ☐ Fewer Specific applications programs.
☐ More ☐ Fewer BASIC programs.
☐ More ☐ Fewer Machine language programs.
☐ More ☐ Fewer Tutorials.
☐ More ☐ Fewer Educational articles.
☐ More ☐ Fewer Detailed explanations of programs.
☐ More ☐ Fewer Games.
☐ More ☐ Fewer Reviews of game software.
☐ More ☐ Fewer Reviews of business software.
☐ More ☐ Fewer Reviews of educational software.
☐ More ☐ Fewer Reviews of hardware.

What do you like best about **COMPUTE!**?

What do you like least?

3 4 5 6 7 8 9 10 11 12

REMEMBER.



Elephant™ floppies.
They're guaranteed to meet or beat every industry standard for quality. They come standard with reinforced hub rings at no extra cost. They come in every popular 5¼" model, in both hard and soft sector. And they sell at some of the lowest prices in the business.
Elephant Flexible Disks.
They're heavy duty. They work for peanuts. They never forget. Get yourself a trunkful.

HEAVY DUTY DISKS.

Distributed Exclusively by **Leading Edge Products, Inc.**, 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.