Guest Commentary:
Isaac Asimov
And The Three Laws

The 6502 Resource Magazine
PET • Apple • Atari • OSI • KIM • SYM • AIM

Interview With
Kit Spencer:
Commodore's
New Marketing VP

COMPUTE!

$2.50
November, 1981
Issue 18
Vol. 3, No.
63379

The Journal For Progressive Computing™

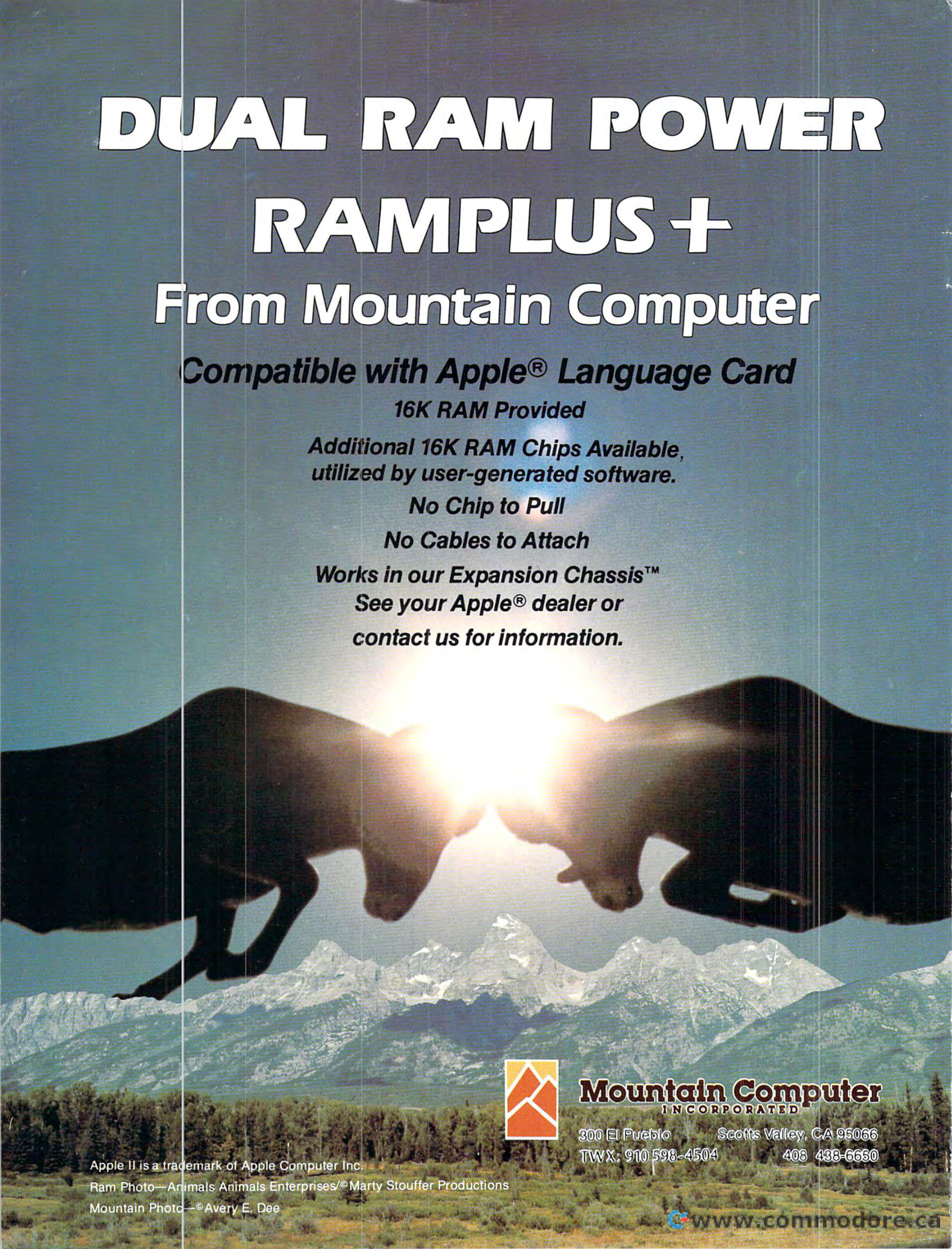
An Atari Data Management Program

An Apple Primer:
Peeks, Pokes, And Calls

Commodore's Super PET Revealed

Insight: Atari
Exclusive I/O Secrets
First of Two Parts

11
0 74470 63379


DUAL RAM POWER
RAMPLUS+
From Mountain Computer
Compatible with Apple® Language Card
16K RAM Provided
Additional 16K RAM Chips Available, utilized by user-generated software.
No Chip to Pull
No Cables to Attach
Works in our Expansion Chassis™
See your Apple® dealer or contact us for information.
Mountain Computer INCORPORATED
300 El Pueblo Scotts Valley, CA 95066
TWX: 910 598-4504 408 438-6650
Apple II is a trademark of Apple Computer Inc.
Ram Photo—Animals Animals Enterprises/© Marty Stouffer Productions
Mountain Photo—© Avery E. Dee
www.commodore.ca

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.

Turn your Apple into the world's most versatile personal computer.

The SoftCard™ Solution. SoftCard turns your Apple into two computers. A Z-80 and a 6502. By adding a Z-80 microprocessor and CP/M to your Apple, SoftCard turns your Apple into a CP/M based machine. That means you can access the single largest body of microcomputer software in existence. Two computers in one. And, the advantages of both.

Plug and go. The SoftCard system starts with a Z-80 based circuit card. Just plug it into any slot (except 0) of your Apple. No modifications required. SoftCard supports most of your Apple peripherals, and, in 6502-mode, your Apple is still your Apple.

CP/M for your Apple. You get CP/M on disk with the SoftCard package. It's a powerful and simple-to-use operating system. It supports more software than any other microcomputer operating system. And that's the key to the versatility of the SoftCard/Apple.

BASIC included. A powerful tool, BASIC-80 is included in the SoftCard package. Running under CP/M, ANSI Standard BASIC-80 is the most powerful microcomputer BASIC available. It includes extensive disk I/O statements, error trapping, integer variables, 16-digit precision, extensive EDIT commands and string functions, high and low-res Apple graphics, PRINT USING, CHAIN and COMMON, plus many additional commands. And, it's a BASIC you can compile with Microsoft's BASIC Compiler.

More languages. With SoftCard and CP/M, you can add Microsoft's ANSI Standard COBOL, and FORTRAN, or Basic Compiler and Assembly Language Development System. All, more powerful tools for your Apple.

Seeing is believing. See the SoftCard in operation at your Microsoft or Apple dealer. We think you'll agree that the SoftCard turns your Apple into the world's most versatile personal computer.

Complete information? It's at your dealer's now. Or, we'll send it to you and include a dealer list. Write us. Call us.

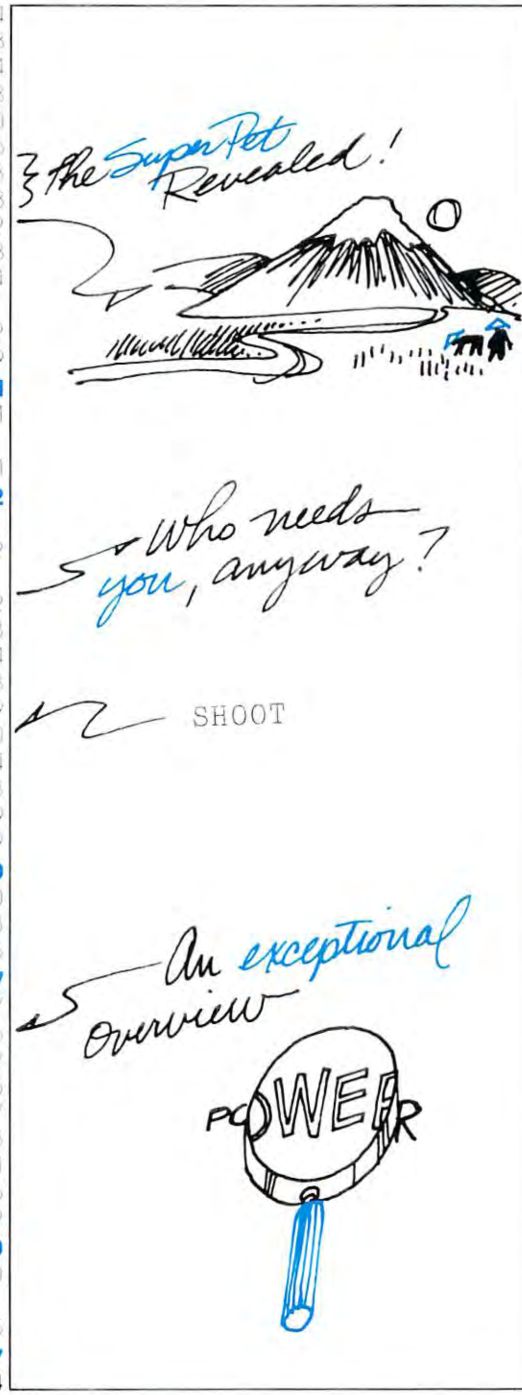
SoftCard is a trademark of Microsoft. Apple II and Apple II Plus are registered trademarks of Apple Computer. Z-80 is a registered trademark of Zilog, Inc. CP/M is a registered trademark of Digital Research, Inc.

MICROSOFT
CONSUMER PRODUCTS

Microsoft Consumer Products, 400 108th Ave. N.E., Bellevue, WA 98004. (206) 454-1315

Table of Contents

November, 1981, Vol. 3, No.



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. Application to mail at controlled circulation rates pending at Hickory, NC 28601.

ISSN 0194-357X.

The Editor's notes...

Robert C. Lock, Publisher/Editor

Atari Educational Sales Revisited

Last issue we mentioned Atari's aggressive pricing moves at the educational "state contract" level. In this context, we mentioned that Atari, Inc. had obtained the new state contract for Minnesota. A recent newsletter from the Minnesota Educational Consortium clarified the current state of the contract. Atari has been added as a vendor as we mentioned last time. Apple, Inc. is still part of the contract as well.

In This Issue

Our extensive articles on Commodore's new "Super Pet" (written by its developers), are fascinating reading. Regardless of your interest, I recommend them as an insight into a new generation of computing.

Bill Wilkinsons' column, *Insight: Atari*, begins a two-part exploration of input/output functions. The information presented is excellent...and unique.

Recreational Computing Magazine Joins COMPUTE! Family

I'm quite pleased to announce that we have merged *Recreational Computing* magazine into our family of publications. Former RC subscribers will now be receiving **COMPUTE!** or *Home And Educational COMPUTING!*. This merger gives additional breadth and depth to our magazines, and an added sense of pride, since RC was the oldest of the personal computing magazines. As always, we welcome the growth!

Home and Educational COMPUTING! Expands

We've made the decision to broaden the base of *Home And Educational COMPUTING!*, in part because of the acquisition of *Recreational Computing*. Beginning with the January/February Issue, *Home and Educational COMPUTING!* will expand editorial coverage to include most of the personal and educational computers selling for $500 or less. We'll provide the same excellent resource and applications information we've established **COMPUTE!** with, to owners and users of the Commodore *VIC-20*, the Radio Shack *Color Computer*, the Texas Instruments *99/4A*, The Atari *400*, and others as well.

Present plans include ongoing columns like *Friends Of The Turtle* by David Thornburg, Ramon Zamora's *Rainbow Machine* column, telecommunications, educational applications and uses, and a great deal more.

Northeast Computer Show

We recently attended the Northeast Computer Show, and were impressed by the growth of the show. Coincidentally, that's the show where **COMPUTE!** was first introduced two years ago. That was the show's first run...we (the collective exhibitors) filled one hall. This year the show filled two downstairs halls, and most of one upstairs hall.

The atmosphere was festive, with Commodore giving away (by drawing) a VIC-20 every day, talking cars, numerous "robots," and an Atari booth that covered an entire stage at one end of a downstairs hall. An interesting change reflecting our mutual growth is that we're seeing less and less of the national firms at these shows, and more and more regional distributors and marketing organizations. Atari and Commodore do continue to bring in corporate level support. Apple, on the other hand, was in evidence through local dealers.

Atari User-Group Drive

I had the pleasure of spending some time with Earl Rice, Atari's new User Group Support Manager. He was showing off an excellent video teaching-tape produced by him and Chris Crawford of Atari. The tape is the first of a planned series which will eventually be made available to user groups and others on a loaner basis. The sketches we saw were not only quite humorous, but excellent and informative. We'll try to keep you posted on availability.

By the way, if you're involved with an Atari User Group, or interested in starting one up, contact Earl at Atari, Inc. He's working hard to set up two-way communications and support. His address is:

Earl Rice, Manager
User Group Support Program
Computer Division
Atari, Inc.
P.O. Box 427
Sunnyvale, CA 94086

"Teaching" Software

Several vendors were displaying well-conceived, well-structured software designed for youngsters. It's really a pleasure to see vendors utilizing the features of the computers they support.

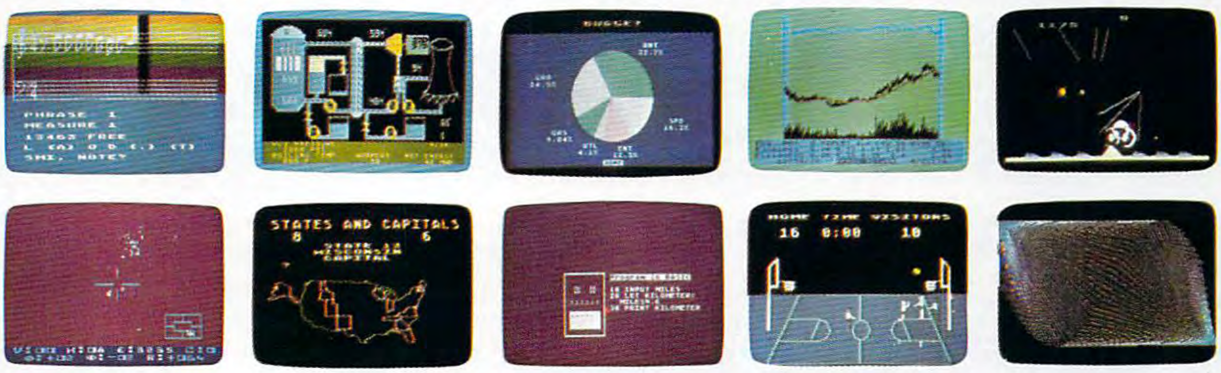
For example, "Sammy The Sea Serpent" from Program Design, Inc.: with a joystick, an eight year old, and good graphics, first glance seemed to indicate another good "game" of simple quality. By using the voice-over capability of the Atari, however, this program is transformed into a talking/learning story.

We watched an absolute novice eight year old work through a set of exercises of ever-increasing complexity, cleverly couched in a storybook setting. We were all entranced by the narrative (and rather pleased when Sammy escaped to the sea!).

John Victor of PDI is one of the few vendors treating the voice capability of the Atari as an extra dimension, and the merits of that treatment were obvious in the resulting software. ©

THE GRAPHIC DIFFERENCE

BETWEEN ATARI® COMPUTERS AND ALL OTHERS.



3.7 million reasons why the ATARI Personal 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 Personal 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... *because 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 800™ computer only.

ATARI®

Computers for people.™

Robert C. Lock, Publisher/Editor
Kathleen Martinek, Managing Editor
Richard Mansfield, Assistant Editor
Alice S. Wolfe, Publisher's Assistant
Charles Brannon, Editorial Assistant

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, MD 65726
Eric Rehnke, 1067 Jadestone Lane, Corona, CA 91720
David Thornburg, 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

Georgia Papadopoulos, Art Director/ Production Manager
Terry Cash, Typesetting/Production Assistant
Dia Rees, Production Assistant
Harry Blair, Director, Advertising and Promotion
Joretta Klepfer, Manager, North American Retail Sales

Bonnie Valentino, Circulation Assistant
Sonja Whitesell, Office Assistant

Subscription Information

(12 Issue Year):
COMPUTE! Circulation Dept.
P.O. Box 5406
Greensboro, NC 27403 USA
U.S. $20.00
Canada $25.00 (U.S. funds)
Europe: Surface Subscription, $25.00 (U.S. funds)

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

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

Advertising Sales

If you're in **Oklahoma, Texas or the Western States, we're now represented by Phoebe Thompson and Associates.** Give them a call for space reservations, contract/insertion information or questions. You can reach them through the following office:

Phoebe Thompson and Associates
101 Church Street
Suite 13
Los Gatos, CA 95030
408-354-5553
PHOEBE THOMPSON

If you're in the East, we're now represented by **The Gittleman Company**. You can reach them through the following offices:

New England, New York State:
The Gittleman Company
Statler Office Building
Suite 582
20 Providence Street
Boston, MA 02110
617-451-0822
JOAN DONAHUE

The Gittleman Company
Summit Office Centre
7622 Summit Avenue
Fort Washington, PA 19034
215-646-5700
DOUG JOHNSON

New York City Metro Area, Mid-Atlantic and Southeastern States:
Local Numbers:
New York 212-567-6717
Atlanta 404-523-1252

If you're in the **Midwest, we're now represented by GB & Associates**. You can reach them through the office of:

GB & Associates
P.O. Box 335
Libertyville, IL 60048
312-362-1821
GORDON BENSON

COMPUTE! Home Office Contacts

Advertising Sales Manager, Harry Blair
Advertising/Production Coordinator, Alice S. Wolfe

Address all advertising materials to:

625 Fulton Street
Greensboro, NC 27403 USA

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

Telephone: 919-275-9809

Small System Services, Inc. publishes:

COMPUTE! The Journal For Progressive Computing

Home and Educational COMPUTING!

COMPUTE! Books

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

Corporate offices are located at 625 Fulton Street, Greensboro, NC 27403 USA

COMPUTE! Subscription Rates

US	$20 (one yr.)
	$36 (two yrs.)
	$54 (three yrs.)
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

Coming In The December Issue of COMPUTE!:

Adding High Speed Vertical Positioning To Atari Player Missile Graphics

File Recovery On The Commodore 8050 And 2040 Disk Drives

Animating Applesoft BASIC Graphics

Speeding Up Your Atari Repeat Key

A Commodore Super PET Update

Computers And Society

David D. Thornburg
Innovision
Los Altos, CA

Interfaces And Languages For The Mass-Market Micro

Last month we explored some ideas about what it will take to get personal computers into the true mass market. Clearly, extreme simplicity of operation will be a major factor for the acceptance of this product by its millions of potential users. Properly designed software can allow the computer to be used for many things without requiring that the user be a proficient touch typist.

Any true "mass-market" computer which, for example, lets you connect to a remote data will *not* require the extensive time consuming log-on procedures in use today. It is easy to see why this must be so. Consider the effort it takes for a typical data-base user to connect to an information utility. First, a telephone number must be dialed (seven keystrokes, minimum). Second, a log-on procedure must be carried out (typically 24 keystrokes). Third, the data base access commands must be entered (anywhere from 10 to 50 characters or more). To read one's mail on Source Telecomputing requires 41 keystrokes, minimum.

While you or I might be willing to go to all this effort in exchange for the tremendous power these data bases provide, I find it hard to believe that such lengthy procedures will be acceptable to the non-technical user who presently gets access to the evening news by simply pressing a button on the TV remote control while seated comfortably in the living room armchair.

The home terminal enviroment of the future will most likely offer menu-driven access to data bases (mail, stocks, news, sports, theater tickets, etc.). A simple joystick or light pen will be used to move the cursor to the desired selection. Once the choice has been made, the *computer* will then carry out the lengthy procedure of dialing the remote host and logging on to the system. The use of non-volatile memory (such as battery powered RAM, or bubble memory) will allow the user to establish the log-on procedure once. After that, this data will remain available in the system until altered by the user.

If a full keyboard isn't required for menu selection tasks, it certainly will be needed by users who want to generate electronic messages for others, or who want to generate their own programs. It is my guess that the home computer of the future will have a full keyboard as an option. Users who are primarily information *receivers* will be able to use simple pointing devices and menu-driven software. People who are also information *providers* will want the flexibility inherent in a full alphanumeric keyboard.

If we believe this scenario, we can then speculate on the shape that such a device might take. The mass-market computer may very likely resemble today's programmable video games more than it resembles today's computers. Imagine a video game with a disk drive and a telephone link and you might not be far from the mark. Many of the popular video games contain complete microcomputers inside them (the Atari VCS uses the 6502, and the Mattel Intellevision uses a 16-bit processor and support chips from General Instruments). The 8-bits of parallel interface needed to support two joysticks can also support a keyboard quite nicely. I have even heard of someone who is selling a plug-in cartridge for the Atari video game which lets you write your own assembly language programs for it. That's something to think about, isn't it!

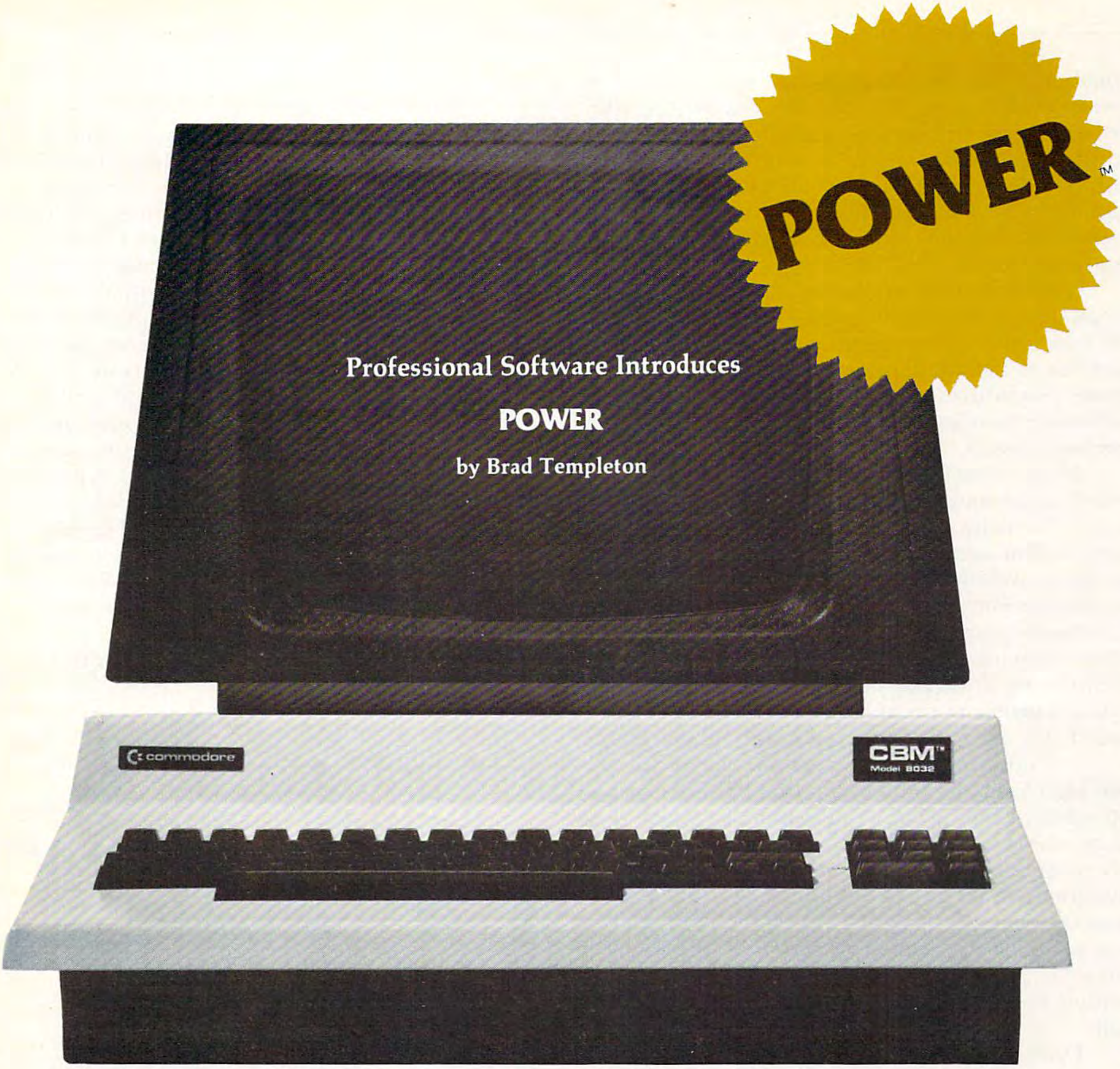
Our scenario of the true mass-market computer is not complete, of course. The "video game" model presented above carries with it the idea that none of the software will be user generated. Imagine, the next time you walk in a record store, what that store will look like with racks of software. Instead of Country, Jazz, and Classical labels, you might see Business, Games, and Home Management labels instead. There will probably even be "cut-out" bins for software that has peaked in popularity!

While many of the future users of computers will be more than content to purchase their software off the shelf, there will be quite a few users who will want to generate their own programs. Just as the home computer of the future may be different from the machines we use today, the languages used by these people most certainly will be.

Most readers of **COMPUTE!** probably know how to write programs in one of several dialects of BASIC. I would guess that, after BASIC, the popular languages would be Assembler, Pascal, and Forth. Given these choices, the casual home user would gladly embrace BASIC.

And yet BASIC is far from being a user friendly language. For example, the fact that program branching commands go to line numbers rather than labels can make it hard to trace program flow in this language. Nonetheless, BASIC has proven quite useful, and has allowed many millions of people to create programs which might not otherwise have been created.

The problem with introducing BASIC into the true home market is that it is perhaps too



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

"mathematical," and therefore too intimidating for someone who wants to gently ease into writing his or her own programs. There are other languages which are more user friendly — notably PILOT and LOGO. It is not mere coincidence that Atari's and Texas Instruments' push into the mass market coincided with their respective introductions of Atari PILOT and TI LOGO.

One of the most appealing aspects of these languages (from the viewpoint of the casual user) is the ease with which small procedures can be created and tested, and then used as building blocks in larger procedures and programs. It is as if the user were able to create extensions to the language, thus personalizing it.

I have written a real-time game in Atari PILOT which is (for me) quite large — over 20 thousand bytes. The main program is less than 30 statements long, and most of the procedures used by this program and by other procedures are 20 to 40 statements long. By building the program out of a great many small modules, many of which use each other from time to time, I was able to create a very complex program and debug it quite rapidly. When a problem was uncovered in an output routine, I was able to fix it and test it without tampering with any other part of the program. One result of this high level of modularity is that PILOT is kept quite busy keeping track of the pointers which show where the modules should return when they are completed. Some portions of the program involve up to six nested modules at any given time. The fact that this program was easy to read after it was written is its greatest asset to me — especially since the program's speed of execution was not unduly compromised by these nested procedure calls.

Even though I write in BASIC almost every week, I would never have tackled this project in that language. I have heard that people who write in LOGO feel the same way.

Are user friendly languages enough to entice the mass market into programming? They will probably capture the interest of many people, but my guess is that the typical home user wants something even simpler.

Recall that it was Visicalc which was responsible for the tremendous acceptance of personal computers by business users. Visicalc sits on the fine line between a language and a program. The user creates a "mask" which contains all the personalized information associated with the spread sheet. Once this is created, the data is then entered much as it would be in a fully "canned" program. The two tasks of the user (creating the mask, and entering the data) are separable. Since the user both *creates* and *uses* the spread sheet, he or she plays the role of both programmer and user, without having to learn about data types, loop structures, recursion, and the like.

Perhaps there will be a new class of languages for the home market which are generalized "task translators." By responding to displayed prompts, the user conveys information to the computer which it can then use in creating a program which is tailored exactly to the user's needs. I have experimented with a few such programs and find their promise to be quite exciting. There is, of course, a great deal of difference between a spread sheet program like Visicalc and a Generalized program writer which could handle anything from video games to personal finance, but I would not be surprised to see our concept of computer languages undergo a radical change in the next five years.

The power contained in the smallest personal computer today can be put to tremendous benefit to all who would use this technology. The key to the acceptance of this technology in the consumer marketplace will be the software tools which allow the user to unleash and mold this power to fill personal needs.

Once this happens, we will have entered a new era. ©

NEECO

WHY BUY FROM THE BEST?

Service... Support...
Software...

MULTI-CLUSTER

For Commodore Systems, allows 3 CPU's (Expandable to 8) to access a single Commodore Disk.

MULTI-CLUSTER (3 CPU's)	$ 995
Each Additional CPU (up to 8)	$ 250

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

ALTOS

ACS 8000-2 64K 1M	$ 4500
ACS 8000-15 64K 1M	$ 5990
ACS 8000-6 208K 14.5M	$10490
ACS 8000-7 208K 29.0M	$11690
ACS 8000-10 208K 10M	$ 8500
ACS 8000-10/MTU	$10990

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

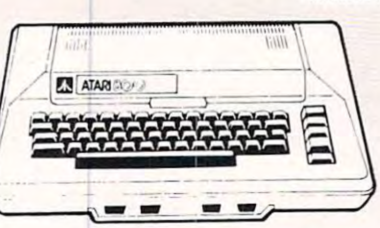
AMDEK MONITORS

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 ANY ADVERTISED PRICE ON 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.

679 HIGHLAND AVE.
NEEDHAM, MA 02194

(617) 449-1760

Telex: 951021

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

Ask The Readers

Robert Lock, Richard Mansfield
And Readers

Please address questions or answers to: Ask The Readers, **COMPUTE!** Magazine, P.O. Box 5406, Greensboro, NC 27403. Special thanks this month to Joseph Wrobel who sent in several extensive answers.

Answers

*"In reply to Rita Norton (**COMPUTE!** #15): filmstrips in Computer Science are available from Educational Activities, Inc., Freeport, NY 11520."*
Arnold Friedman

"In your August issue, you published a letter in the "ASK THE READERS" column about Edward Sweeney's problems with interfacing the Vortrax TYPE 'N TALK speech synthesizer with his Atari 800. The best way to do this is with the Atari 850 interface (list price is about $219.95). Since the TYPE 'N TALK uses an RS-232 interface to communicate with the computer, an RS-232 interface (such as the Atari 850) should be used to hook it up to any personal computer.

I have seen (and heard) the TYPE 'N TALK at a computer store, and I think that it is fantastic*! It may not sound as human as the TI line of synthesizers (Speak 'N Spell, etc.), but it* does *have an* unlimited vocabulary*! (The TI ones don't).*

For those of you who have been to the arcade recently, you may have noticed the video games called "Gorf" and "Wizard Of Wor." The Vortrax TYPE 'N TALK sounds almost exactly like the voice synthesizers used in these games (but with much more clarity and understandability). For fun (after you get the connections up and the software running) try making it say "bite the dust space cuddet" (exactly as spelled). Programming instructions are included with the Atari 850 in a large manual. You program using such common things a XIO XX, #X,X,X,"X:", OPEN #X,X,X,"X:", etc. (Note: The TYPE 'N TALK may also require an interface cable from Atari. Contact Atari for details on the cables.) The last thing that the TYPE 'N TALK needs is an 8-ohm speaker (available at Radio Shack, etc.).

One last thing (now changing the subject). In the same issue as Mr. Sweeney's letter, there was another one by Jerry Stern, asking about using the keypads for data entry. The answer to his question is "Yes!". For a program, Dr. Stern should get an Atari Basic Reference Manual *and look at appendix H, page H-14. I hope I was of help!" Greg Marquez*

"I'm writing with regards to the questions raised in your [Clyde Spender] **COMPUTE!** *#14 article entitled "Atari Graphics: 16 Colors!" concerning graphics modes 9 through 11. Actually, there are two questions I'll attempt to answer. The first asks how graphics modes 9 through 11 are supposed to operate. The second asks how they are currently implemented.*

The answer to the first question can be found on pages 172-174 of the Atari Personal Computer System Operating System User's Manual *(inside California call ATARI at 1-800-672-1430 for purchase info). These pages comprise the manual's Appendix H, entitled "Screen Mode Characteristics," which describes the characteristics of graphics modes 0 through 11. What it tells us about modes 9 through 11 is that they are all 80 pixel by 192 line modes, and all use four bits per pixel. Thus, they occupy the same amount of display memory as graphics mode 8, namely 192 line of 40 bytes per line. The difference between the three modes is how the four bits per pixel are interpreted.*

In mode 9 these bits are interpreted as luminance data; since the low-order luminance bit is ignored, this mode supports an eight-level gray scale display. The color of the display is determined by color register 4, the one which also sets the background color. Mode 11 is the inverse of mode 9; the pixel data select one of ATARI's 16 standard colors while the overall luminance is determined by color register 4. Graphics mode 10 is somewhat less straightforward. Here the different data values reference one of the 9 color registers (5 normal + 4 player missile). Because there are 16 possible data values and only 9 color registers, there is some duplication. The bottom line is that 9 different color/luminance combinations of your choosing are supported in graphics mode 10.

I believe the colors that you observe are false, due to the alaising caused by the high frequency transitions in the 320 dot per line display. If you're skeptical, run the accompanying program. If your ATARI behaves like mine, you should see no change in the graphics display except for the border. This border change is due to the fact that in graphics mode 11, and only in graphics mode 11, color register 4 is initialized to six rather than its normal initial value of zero.

Note that the program accesses the graphics modes using a BASIC command of the general form:

```
OPEN #6,X,G,"S:"
```

where G is the graphics mode and X is usually 8, i.e. OPEN for writing. This command can be used for all graphics modes (0 through 11). Add 32 to X to inhibit screen clear; add 16 to X for split screen display."
Joseph Wrobel

```
10 OPEN #6,8,10,"S:"
20 FOR I=0 TO 4
30 FOR J=0 TO 15
```

SOFT ROM

- 4096 BYTES OF SOFT ROM
- STORE MACHINE CODE SOFTWARE BEYOND THE BOUNDARIES OF BASIC
- WRITE PROTECT RAM WITH A FLIP OF A SWITCH
- STORE VARIABLES OR INDICES OUTSIDE OF BASIC
- SOLVE THE CONFLICTING ROM PROBLEM BY SOFT-LOADING THE APPROPRIATE ROM IMAGES

The SOFT ROM is compatible with any large keyboard PET/CBM or similar 2532 EPROM systems. It may be placed in any ROM socket to give the user room for machine code. If the SOFT ROM is placed in an occupied ROM socket, the user can transfer the PET/CBM ROM into the on-board ROM socket and select between ROM and RAM to manipulate the Commodore operating system.

Since the SOFT ROM places write protectable RAM into any of the computer's ROM sockets, it is ideally suited to use as a development tool to test ROM or EPROM based software systems before they are burned in.

Examples of software presently available for the SOFT ROM includes BASIC AID, UNIVERSAL WEDGE, SUPERMON, EXTRAMON, USER PORT PRINTER (Centronics parallel) and a buffered BACKGROUND PRINTER routine.

Installation is a simple plug-in into any available ROM socket.

$129.00

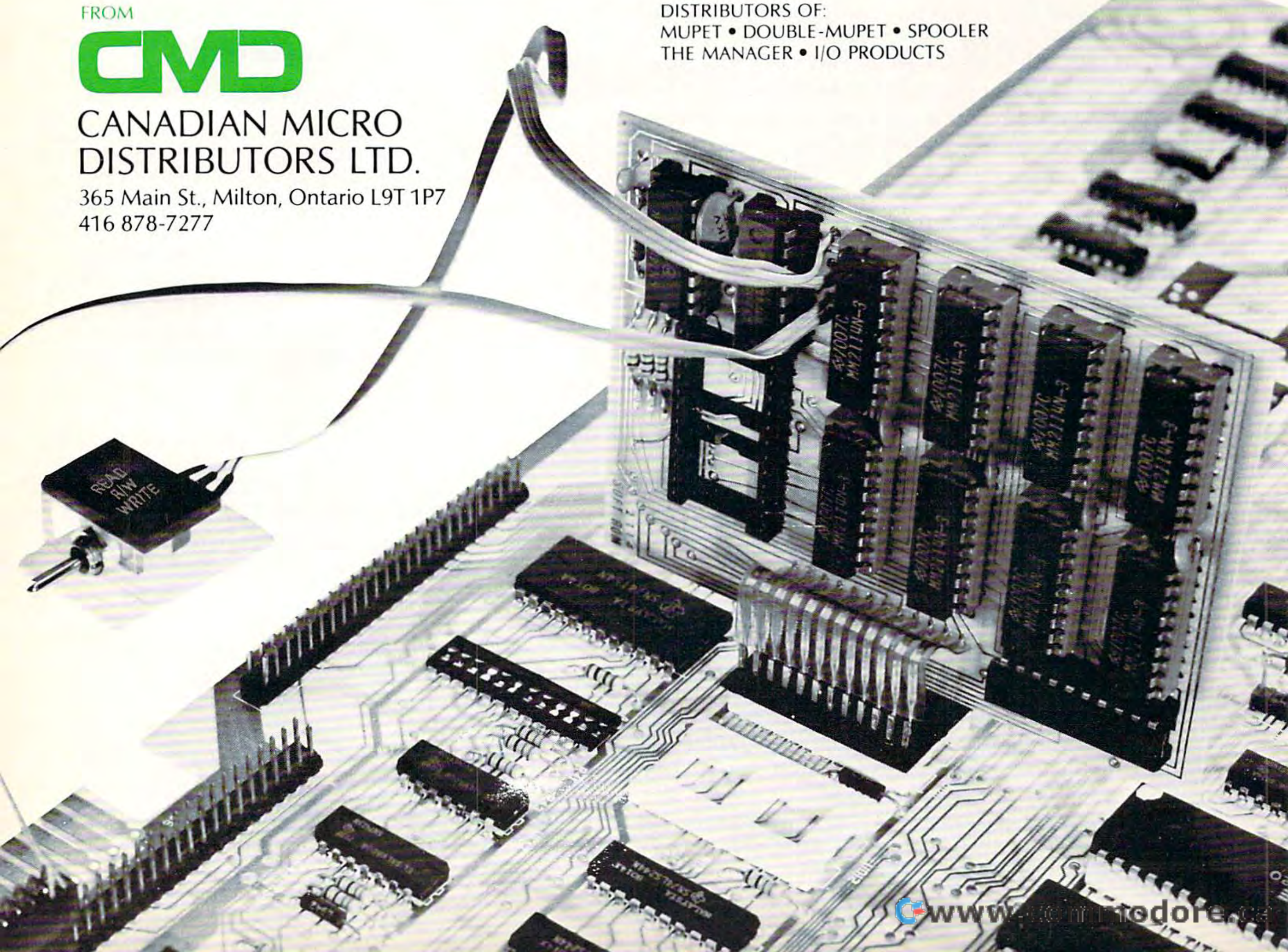
FROM

CMD

CANADIAN MICRO DISTRIBUTORS LTD.

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

DISTRIBUTORS OF:
MUPET • DOUBLE-MUPET • SPOOLER
THE MANAGER • I/O PRODUCTS



```
 40 COLOR J
 50 PLOT 15*I+J,0:DRAWTO 15*I+J,191
 60 NEXT J
 70 NEXT I
 80 CLOSE #6
 90 FOR G=8 TO 11
100 OPEN #6,40,G,"S:"
110 FOR I=1 TO 250:NEXT I
120 CLOSE #6
130 NEXT G
140 GOTO 90
150 END
```

Questions

"What can we poor $395.00 Commodore 4010 Voice Synthesizer victims do to get any kind of programming help? Can you advise me of any users that can give me some kind of back-up in this matter?" L. W. Goesch

---

"I have written a program to cover my personal financial accounts which I run through monthly and add on the month's expenditures and income under various headings. The program then does various things with the data, such as forecasting for the whole and future years, highlighting items of over- and under-expenditure, etc.. The problem is that each month I have to list the DATA statements in order to amend them by adding on the current month's figures.

"Is there any way I can get the program to automatically update the date information so that I do not have to do this by hand? Any help you can offer would be appreciated...I have an 8K Original ROM PET...." Peter Shafe

You want the computer to imitate what you are now doing by hand: print a changed DATA statement on the screen and then press RETURN to place it into the program. This is a "self-modifying program." It can be done by telling the computer to print the new DATA statement on the screen and then POKEing carriage returns into the keyboard input buffer. Its starting address is 527 (your Original ROM PET), 623 (Upgrade and 4.0). The number of automatic RETURNs you need must be POKEd into address 525. For example, if you POKE 527,13: POKE 528,13 (putting two carriage returns into the buffer), you must then POKE 525,2. (For Upgrade and 4.0: POKE 158,2.) Finally, this line of POKEs must end with END.

This program would replace the DATA statement in line "L" with an updated set of DATA. (Notice that the new data is in variables Y and Z which, when printed on the screen, will be numbers. Also, the CLEAR SCREEN and HOME are used to correctly position the cursor so that the RETURNS will be made over the new DATA line):

```
100 PRINT"{CLEAR}{03 DOWN}"L"DATA"Y","Z"{
    DOWN}L="L+2":GOTO500{04 UP}":POK
    E525,2:POKE527,13:POKE528,:END
500 IFL>50THEN540:REM YOU PASS BACK THE V
    ALUE OF L TO THE PROGRAM
510 REM TO ALLOW THE PROGRAM TO KNOW WHIC
    H DATA LINE IT LAST CHANGED--
520 REM A DIRECT-MODE RETURN WILL NOT LEA
    VE THE VARIABLE VALUES INTACT.
530 GOTO 1000: REM THEN YOU CONTINUE TO C
    HANGE OTHER DATA UNTIL DONE.
540 PRINT"JOB ACCOMPLISHED":END
```

A similar technique is used to automatically delete lines in a program in **COMPUTE!** #12, pg. 116. This same result can be achieved on the Atari *as well: see* **COMPUTE!** #15, pg. 80. Also see last month's **COMPUTE!** page 22.

"First allow me to compliment you for creating what has been needed in a computer magazine for a long time. Your "Ask the Readers" column really puts you head and shoulders above the competition! Here is a problem that has had me frustrated for a year now:

I own an OSI C3 and the OS65D 3.0 operating system, with which I have been perfectly happy with except for one thing: I cannot successfully use sequential access data files. For example, let us say I have created the file, "DATA1". I would use the following program:

```
10 A$="DATA1"
20 DISK OPEN,6,A$
30 PRINT#6,A:PRINT#6,B:PRINT#6,C
40 DISK CLOSE,6,A$
50 PRINT "DATA NOW ON DISK"
60 GOTO 100
70 ..... (etc.)
```

Although I would have a line number 100 in the program, I would always get a US (undefined statement — no such line number) error in line 60. When I try to LIST the program, I just get garbage on the screen. I have followed the manual letter for letter and asked others about it, but all for naught; I still can't get it to work. I know that there must be some way — I have software in BASIC that does it successfully. Any suggestions? Please help!"
John Fry

"I wish to operate a Commodore 3016 computer and 3040 floppy disk drive from a marine 12v battery supply via wiring modifications and outboard circuitry if necessary. The result should allow conversion back to the usual AC power, but it need not be switchable between the two (although this would be an advantage). Can you supply any of the following?

1. *Details of required DC power supply voltages, current capacity, and regulation specifications for the two units.*

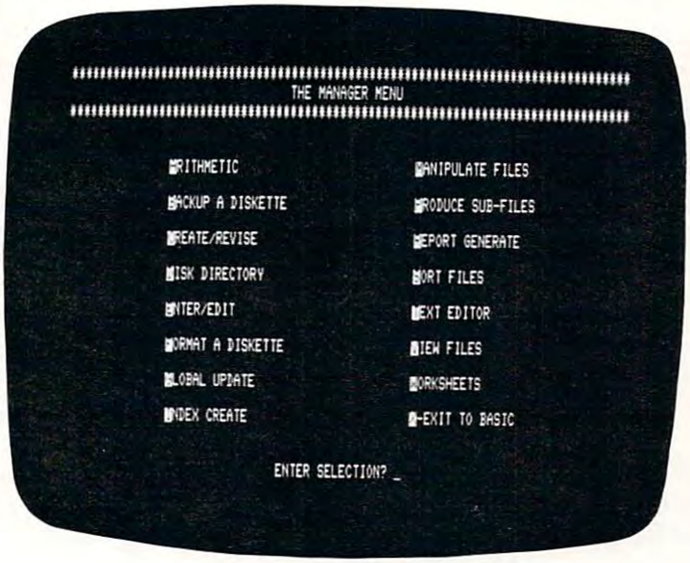
2. *A tested circuit by someone who has done it*

I realize that a 12v-to-240v inverter of 200 watts capacity is an alternate, but the direct approach is preferred." Frank Chambers ©

THE MANAGER

The first truly user-friendly Database Management System available at reasonable cost.

This suite of programs is ideally suited for both the businessman and programmer, for use with the CBM 8032.



For the Business User

- Uses Menu Options – no programming experience needed.
- Lets you enter data in the form you wish, then lets you recall it using any search criteria.
- Performs predefined calculations on the record in realtime as record is displayed on the screen.
- Reports can be produced using any search criteria and/or arithmetic functions.
- Useful applications can be developed quickly.

Typical Applications include –

- Inventory Control
- Mailing Lists
- Accounting systems
- Personnel
- Costing
- Gathering test data
- Budgeting
- Scheduling
- Examples of use included on disk supplied.

As Programmers Tool

- Uses standard PET ASCII files.
- Software interface is in Basic and available to the programmer.
- No special disk formatting so that word processing or other programs can be stored on the same disk.
- No ROM Based Security thus no need to open CPU.
- Fast 'n' key Sort/Merge included.
- Full realtime intra & inter record arithmetic performed on the screen as record is displayed.
- Professional software support including unique security available.

TRY IT!
IF YOU ARE
NOT SATISFIED WITHIN 30 DAYS
WE WILL RETURN YOUR MONEY

$250.00

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

DISTRIBUTORS OF:
MUPET • DOUBLE-MUPET • SPOOLER
THE MANAGER • I/O PRODUCTS

Guest Commentary:

The Three Laws

Isaac Asimov
New York, NY

Now that computerized robots are not only possible, but actual; now that they are rapidly invading industry, and becoming a factor that may produce extraordinary economic and social changes over the next generation; I can't help but think back forty years to the time when I invented the Three Laws of Robotics, in 1942.

These are:

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.

2. A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.

3. A robot must protect its own existence, as long as such protection does not conflict with the First or Second Law.

Since these Laws are often quoted, quite seriously, in articles and books on robotics (a word I was, apparently, the first ever to use, back in 1942), I am sometimes tempted to wonder at the prescience of my 21-year-old self, and to suspect that perhaps the high opinion some people have of me may possibly be deserved.

But then rationality intervenes, and I know that this is nonsense. The Three Laws are obvious from the start, and everyone is aware of them subliminally. The Laws just never happened to be put into brief sentences until I managed to do the job.

The Laws apply, as a matter of course, to every tool that human beings use.

Consider a knife, for instance. The first law of knifedom is that it be used safely. No one would use a knife if it meant cutting one's fingers off in the process. Therefore, to begin with, a knife is equipped with a handle. To generalize, any cutting instrument must offer a way of being safely held while it is being used to cut.

The second law of knifedom is that it be used effectively. Therefore, a knife must be given a sharp edge (provided that is safe), for no one is interested in hacking away uselessly with a dull blade.

The third law of knifedom is that it maintain its integrity during cutting. Of what use would a knife be if it broke or dulled while cutting? A knife is therefore made of some tough material that holds an edge and that doesn't snap (provided such toughness doesn't interfere with either its safety or its effectiveness.)

You can apply this sort of reasoning, not only to material tools, but, also, without too much difficulty, to a social institution such as the Constitution of the United States.

The delegates to the Constitutional Convention of 1787 endeavored to work out a document that (first) would be safe to use, and would not subject Americans to a tyranny; and that (second) would be flexible enough to be responsive to the needs of the people, provided that did not compromise its

safety; and that (third) would be sufficiently durable to serve new times and new conditions, by means of amendments if necessary, provided that did not compromise either its safety or its effectiveness.

You can even apply this sort of reasoning to your own behavior: to your attitude toward your diet, or toward exercise, or toward your job. That behavior must insure first safety — then effectiveness — then durability.

Consequently, I have my answer ready whenever someone asks me if I think that my Three Laws of Robotics will actually be used to govern the behavior of robots, once they become versatile and flexible enough to be able to choose among different courses of behavior.

My answer is, "Yes, the Three Laws are the only way in which rational human beings can deal with robots — or with anything else."

— But when I say that, I always remember (sadly) that human beings are not always rational. ©

WHY THE MICROSOFT RAMCARD™ MAKES OUR SOFTCARD™ AN EVEN BETTER IDEA.

Memory — you never seem to have quite enough of it.

But if you're one of the thousands of Apple owners using the SoftCard, there's an economical new way to expand your memory dramatically.

16K ON A PLUG-IN CARD.

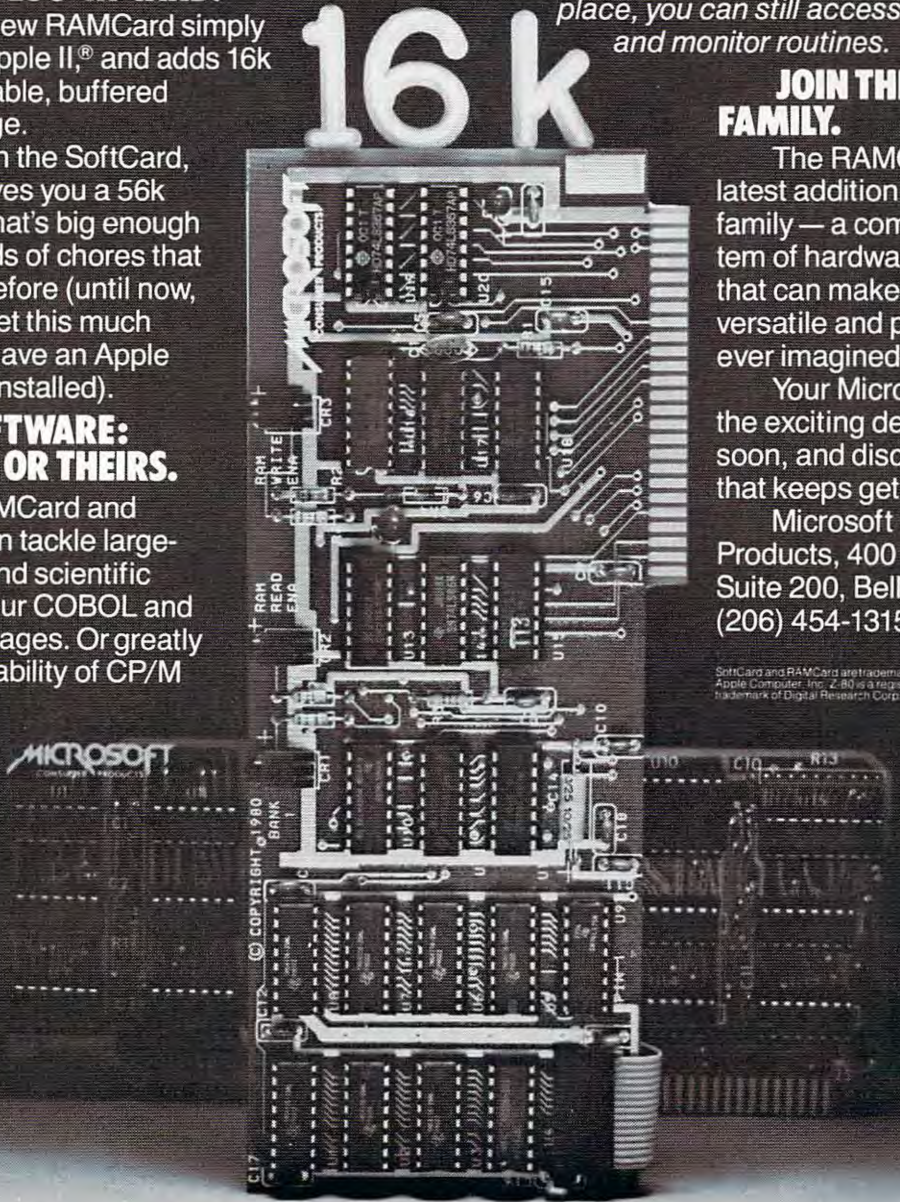
Microsoft's new RAMCard simply plugs into your Apple II,® and adds 16k bytes of dependable, buffered read/write storage.

Together with the SoftCard, the RAMCard gives you a 56k CP/M® system that's big enough to take on all kinds of chores that would never fit before (until now, the only way to get this much memory was to have an Apple Language Card installed).

GREAT SOFTWARE: YOURS, OURS, OR THEIRS.

With the RAMCard and SoftCard, you can tackle large-scale business and scientific computing with our COBOL and FORTRAN languages. Or greatly increase the capability of CP/M applications like the Peachtree Software accounting systems. VisiCalc™ and other Apple software packages can take advantage of RAMCard too.

And RAMCard gives you the extra capacity to develop advanced programs of your own, using the SoftCard and CP/M. *Even with the RAMCard in place, you can still access your ROM BASIC and monitor routines.*



JOIN THE SOFTCARD FAMILY.

The RAMCard is just the latest addition to the SoftCard family — a comprehensive system of hardware and software that can make your Apple more versatile and powerful than you ever imagined.

Your Microsoft dealer has all the exciting details. Visit him soon, and discover a great idea that keeps getting better.

Microsoft Consumer Products, 400 108th Ave. N.E., Suite 200, Bellevue, WA 98004. (206) 454-1315.

MICROSOFT

The Beginner's Page

Richard Mansfield
Assistant Editor

Searching Files

Here is a program to maintain a master index of all **COMPUTE!** articles. It will demonstrate a way that data can be *managed* by a computer to make entering and retrieving information fast, easy, and accurate. In specific, we will look at the problem of searching for data within a data file.

Central to most data management tasks is the job of searching through a list of records (a file) for a particular record or class of records. You might want to see all **COMPUTE!** articles on the topic of computer shows. The string-manipulating BASIC commands (LEFT$, RIGHT$, MID$, and LEN) are both fast and flexible when used as searching tools. This program will illustrate some of the major considerations when setting up a database management program:

Program 1. Microsoft Version

```
4 REM ******************************
5 REM *       INITIALIZATION       *
6 REM ******************************
1Ø T = 1ØØ
2Ø DIMA$(T)
3Ø FOR I = 1 TO T:READ A$(I)
4Ø IF A$(I)="END"THEN T=I-1:GOTO1ØØ
5Ø NEXT I
99 REM ******************************
1ØØ REM*         MAIN LOOP          *
1Ø1 REM******************************
11Ø PRINT: PRINT "PLEASE CHOOSE:"
12Ø PRINT "1.AUTHOR
13Ø PRINT "2.SUBJECT
14Ø INPUT K$
16Ø ON VAL(K$)GOTO 2ØØ,3ØØ
199 REM******************************
2ØØ REM*        SUBROUTINES         *
2Ø1 REM******************************
21Ø PRINT "TYPE AUTHOR'S LAST NAME
22Ø INPUT NAME$
23Ø FOR I = 1 TO T
24Ø L = LEN(A$(I))
25Ø FOR B = 1 TO L
26Ø IF MID$(A$(I),B,1)  =  "*"  THEN ~
  ~ GOTO 28Ø
27Ø NEXT B:PRINT "MISSING *'S IN RECO~
  ~RD #"I:END
28Ø B = B+1:IF MID$(A$(I),B,LEN(NAME$~
  ~)) = NAME$ THEN PRINT A$(I)
29Ø NEXT I:GOTO 1ØØ
3ØØ PRINT "PLEASE TYPE THE TARGET SUB~
  ~JECT"
31Ø INPUT SUBJECT$
32Ø FOR I = 1 TO T
33Ø L = LEN(SUBJECT$)
34Ø IF LEFT$(A$(I),L) = SUBJECT$ THEN~
  ~ PRINT A$(I):Q = 1
35Ø NEXT I:IF Q = Ø THEN PRINT "NO MA~
  ~TCHES FOUND"
36Ø Q = Ø:GOTO 1ØØ
499 REM******************************
5ØØ REM*           DATA             *
5Ø1 REM******************************
51Ø DATA SHOW--TRENTON COMPUTER FESTI~
  ~VAL*BUTTERFIELD*15
52Ø DATA PREVIEW--CBM FAT 4Ø*BUTTERFI~
  ~ELD*15
2ØØØØ DATA END
READY.
```

Program 2. Atari Version

```
2 REM ATARI VERSION
3 REM
4 REM **********************
5 REM *   INITIALIZATION   *
6 REM **********************
1Ø T=1ØØ
2Ø DIM A$(T*8Ø),T$(8Ø),L(T),NAME$(4Ø),SU
BJECT$(4Ø)
3Ø FOR I=1 TO T:READ T$
4Ø IF T$="END" THEN T=I-1:GOTO 1ØØ
5Ø A$((I-1)*8Ø+1,I*8Ø)=T$
6Ø L(I)=LEN(T$)
7Ø NEXT I
1ØØ REM **********************
1Ø1 REM *      MAIN LOOP     *
1Ø2 REM **********************
11Ø PRINT :PRINT "Please choose:"
12Ø PRINT "1. Author"
13Ø PRINT "2. Subject"
14Ø INPUT K
15Ø ON K GOTO 2ØØ,3ØØ
199 REM **********************
2ØØ REM *    SUBROUTINES     *
2Ø1 REM **********************
21Ø PRINT "TYPE AUTHOR'S LAST NAME"
22Ø INPUT NAME$
23Ø FOR I=1 TO T
24Ø L=L(I):T$=A$((I-1)*8Ø+1,(I-1)*8Ø+L)
25Ø FOR B=1 TO L
```

ATTENTION COMPARISON SHOPPERS

HOW DOES A $299 BYTEWRITER-1 STACK UP AGAINST A $650 EPSON MX-80? YOU DECIDE!

The Only 80 Column Dot Matrix Printer Under $300.

Why do we dare to compare the Bytewriter-1 to the Epson MX-80, the industry leader? Because we feel strongly that dollar for dollar, the Bytewriter-1 is tough to beat for performance and quality.

Our extensive testing has proved that the Bytewriter-1 interfaces problem-free to the TRS-80, the Apple II and the Atari 400 and 800.

We are not going to tell you that the Bytewriter-1 is better than the MX-80, but by comparison, and for half the cost, you get more than a reliable printer — you get a great value.

Call or write for more information today.

Comparable features. Uncomparable price.

FEATURES	BYTEWRITER-1	EPSON MX-80*
Print speed	60 lines per minute	46 lines per minute
Paper feed	Friction feed original plus 3 copies	Pin feed original plus 2 copies
Ribbon	Black, cartridge $9.95	Black, cartridge $14.00
Life expectancy	Printhead — 100 million char. Drive Mech. — 10 million char. Ribbon — 5 million char.	50 — 100 million char. 5 million char. 3 million char.
Dimensions	3.8″ × 15″ × 9″	5.2″ × 14.7″ × 12″
Character set	96 ASCII	96 ASCII
Interface	Parallel	Parallel
Warranty	90 days	90 days
Printhead replacement	$29.95	$30
Cost	**$299**	**$650**

*Data source: Epson MX-80 Operation Manual

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

30 Day Money Back Guarantee

```
260 IF T$(B,B)="*" THEN GOTO 280
270 NEXT B:PRINT "MISSING *'S IN RECORD
#";I:END
280 B=B+1:IF T$(B,B+LEN(NAME$)-1)=NAME$
THEN PRINT T$
290 NEXT I:GOTO 100
300 PRINT "PLEASE TYPE THE TARGET SUBJEC
T"
310 INPUT SUBJECT$
320 FOR I=1 TO T
330 L=LEN(SUBJECT$)
340 T$=A$((I-1)*80+1,(I-1)*80+L(I))
345 IF T$(1,L)=SUBJECT$ THEN PRINT T$:Q=
1
350 NEXT I:IF Q=0 THEN PRINT "NO MATCHES
 FOUND."
360 Q=0:GOTO 100
499 REM ****************
500 REM *     DATA     *
501 REM ****************
510 DATA SHOW--TRENTON COMPUTER FESTIVAL
*BUTTERFIELD*15
520 DATA PREVIEW--CBM FAT 40*BUTTERFIELD
*15
20000 DATA END
```

How It Works

Let's see how this program searches through the file for records which match whatever is required. This kind of file (lines 510-520) is called a *variable-field file* because each of the three fields — the subject, the author, and the issue number — can be of any length. The DATA statements each contain one record. Within each record, the author field could be as short as "Cox" or as long as "Butterfield." The fields can *vary* in length. This saves memory space, but at the expense of speed. So, for large databases, where a search will go through hundreds or thousands of records, *fixed fields* are used because speed becomes an important factor. A name like "Cox" would be padded with blanks to take up the proper amount of fixed space required by its field. But we are setting up a smaller database and will take advantage of the memory efficiency of variable fields in this program.

Because the fields *are* variable, we have to let the computer know where one ends and another begins. Otherwise, how would it know that the author's name in line 510 wasn't Festival Butterfield? We separated the fields by using the "*". This means that we will have to look for "*" within each record when we are searching for the author fields. This is why the variable-field can be slow: each record must be handled individually, watching for *delimiters* (fences between fields). Let's take a look at what the program does:

Line Number

10 Here we tell the computer that there are a maximum of 100 records in our file. This is necessary because (in line 20) the computer must set aside memory space for each record.

20 Set up an *array* which is DIMensioned to 100 string-variable zones. Each string (record) will fill each zone, but Microsoft BASIC dynamically expands the zone sizes so the DIMensioning is merely the *number* of zones, not their size. (In Atari BASIC, the DIM statement sets up one giant string. The DIMensioning in Atari is the *size* of this huge string).

...for large databases, where a search will go through hundreds or thousands of records, *fixed fields* are used because speed becomes an important factor.

30 READ each DATA statement (place it into its proper, numbered zone in memory). READ up to the maximum (T) unless...

40 one of the DATA statements is the word "END" in which case you went *beyond* the last real record — you must subtract one from counter (I) so the program can now know the true total (T) number of records in this database.

50 Keep raising the counter (I) until you reach the limit (100) or get the word "END" (line 40), then...

110-160 A typical *main loop* structure with a menu of choices, an INPUT from the user, and then a *branching* which depends on the INPUT.

230 Establish a loop which will "pull out" each record, from record 1 to the total (T).

240 In each case, find out the length (L) of a record.

250 A second loop, *nested* within the other loop, which will count each *character* within a particular record. It will end, of course, when it reaches the total number of characters in a record (L).

260 When one of the characters is a "*", we know that we are positioned at the "author field" of the record. So we skip the "NEXT B" count up, in line 270.

270 If we have no more "NEXT B," then we went through an entire record without finding the "*" symbol and we print an error message on the screen and END the program so the bad DATA statement can be fixed.

280 First, since B now is the position of the

"*" within the record, we add one to it so that B points to the first character of the author's name. Then we compare the portion of the string (record) which starts at position B and which is as long as the name which was INPUT in line 220. This will give us an exact comparison. It also has the benefit of allowing us to INPUT "BU" and search for all names starting with "BU" instead of writing out the entire name "BUTTERFIELD."

290 If we have finished checking and printing out the matches, we simply return to the main loop menu to see if other searches will be requested.

320 Again we loop through each record. This time we are looking at the first field: the subject field in a record. Therefore, we will not need to look for the "*". The subject field starts at the first position within the record.

330 Here we measure the length of the INPUT request. This allows us to have any level of specification. For example, if "DISK" is INPUT in line 310, all the records which refer to disks will be printed. If "DISK STORAGE" is INPUT, however, only records which match the entire SUBJECT$, "DISK STORAGE," will appear.

340 A simpler version of the compare in line 280. Q is set to equal 1 to show that a match was found so that line 350 will not print its message.

360 Q is reset to zero so it will function correctly as a *flag* of matches. We return to the main loop menu.

If you add more DATA lines, this program will hold and examine as many records as your RAM memory permits. If you have 16K RAM, you can put in more than twice as many DATA statements as would be possible in 8K RAM. (The program itself uses up some of the RAM.) ©

Basically Useful BASIC

Editor's Note: Peter is nine. His mother says that Peter designed the program himself, but she helped with the coding. — RTM

A Flower Sale Program

Peter Deal
Malvern, PA

I am a Cub Scout in Den 3, Willistown Pack 98. Every year we sell pansies. All my neighbors buy them, like almost everybody on my street. The next Saturday, my mom and I deliver the pansies. To know how many boxes are sold, and to know how much money everyone owes me, I use this program and my mom helped me write it on the PET. This year, pansies sold for $2.25 a box.

Helper's notes:

- PET people can type lines 100–390. Users of other computers should type in lines 270–590.
- This little program checks input for valid entries (between 1 and 50). The PET version does so by overwriting the input prompt, the other version by repeating that prompt. Incorrect entries can be further changed after the computer asks "7 boxes Y/N". Lines to 200–210 in the PET Version, lines 540–560 in the other version handle that.
- It is likely that lines 530–540 are system dependent, so they deserve an explanation. Line 530 clears the buffer of all key presses. Line 540 waits until "Y" or "N" is typed in. All other keys are rejected. Line 550 then sends the control back to the input prompt if the answer was "N". Otherwise the program proceeds to calculate.
- Jim Butterfield's routine is used in several places by specifying V, V1 and V2 parameters. This routine has been fully described in **COMPUTE!** #9 (pg. 30).

```
100 REM===============================
110 REM PANSY SALE - PET VERSION
120 REM===============================
130 PR=2.25:MN=0:BX=0:TC=0:TB=0:REM PRICE
    ,MONEY,BOXES,TOTAL MONEY,TOTAL B
    OXES
140 SP$="                   ":SP$=SP$+SP$+SP$
150 PRINTSP$:PRINT"{UP}";:V$=""
160 INPUT"> 'END' OR HOW MANY BOXES    _"{03
     LEFT}";B$ : IFB$="END"GOTO250
170 BX=VAL(B$):IF BX<=0 OR BX>50 THENPRIN
    T"{UP}";:GOTO150
180 V1=3:V2=0:V=BX:GOSUB300:PRINT "     {0
    4 LEFT}" V$ " BOXES - Y/N";
190 FORJ=1TO9:GETQ$:NEXT
200 GETQ$:IFQ$<>"Y"ANDQ$<>"N"GOTO200
210 IFQ$="N"THENPRINT:PRINT"{02 UP}";:GOT
    O150
220 MN=PR*BX : TB=TB+BX : TC =TC+MN
230 V1=4:V2=2:V=MN:GOSUB300:GOSUB370: PRI
    NT"--------------"V$
240 GOTO160
250 PRINT:PRINT:V1=7:V=TC:GOSUB300:GOSUB3
    70:PRINT"I SOLD" TB "BOXES FOR"V
    $:END
260 REM===============================
270 GOTO480
280 REM 'USING' ARRANGE IN COLUMNS, JIM B
    UTTERFIELD ROUTINE TO 350
290 REM V IS VALUE; V1.V2 PRINTS
300 V4=INT(V*10↑V2+.5):REM ROUNDED
310 V$=RIGHT$("               "+STR$(V4),V1+V2
    +1):IFV2<1GOTO340
320 FORV5=V1+2TOV1+V2+1:IFASC(MID$(V$,V5)
    )<48THENNEXTV5
330 V6=V5-V1-1:V$=MID$(V$,V6,V1+1)+LEFT$(
    ".000000000",V6)+MID$(V$,V5)
340 IFASC(V$)>47THENV$=LEFT$("*********",
    V1+V2+2+(V2=0))
350 RETURN:---
360 REM FLOAT $
370 FORV7=1TOLEN(V$):IFMID$(V$,V7,1)=" "T
    HENNEXTV7
380 IFV7>1THENV$=LEFT$(V$,V7-1)+"$"+MID$(
    V$,V7)
390 RETURN:---
400 REM===============================
410 REM SIMPLIFIED VERSION FOR
420 REM OTHER MICROSOFT SYSTEMS
430 REM    U N T E S T E D  !
440 REM===============================
450 REM TYPE CODE FROM LINE 270 DOWN
460 REM TO INCLUDE BUTTERFIELD ROUTINE
470 REM===============================
480 PR=2.25:MN=0:BX=0:TC=0:TB=0:REM PRICE
    ,MONEY,BOXES,TOTAL MONEY,TOTAL B
    OXES
490 PRINT"> 'END' OR HOW MANY BOXES":INPU
    T B$: IF B$="END"GOTO590
500 BX=VAL(B$):IF BX<=0 OR BX>50 GOTO490
510 V1=3:V2=0:V=BX:GOSUB300
520 PRINT V$ " BOXES - Y/N"
530 FORJ=1TO9:GETQ$:NEXT
540 GETQ$:IFQ$<>"Y"ANDQ$<>"N"GOTO540
550 IFQ$="N"GOTO490
560 MN=PR*BX : TB=TB+BX : TC=TC+MN
570 V1=4:V2=2:V=MN:GOSUB300:GOSUB370: PRI
    NT"------------------------"V$
580 GOTO490
590 PRINT:PRINT:V1=7:V=TC:GOSUB300:GOSUB3
    70:PRINT"I SOLD" TB "BOXES FOR"V
    $:END
```

©

Blue Ribbon Computer Books

Small Computers for the Small Businessman

Nicholas Rosa and Sharon Rosa

"Must reading for businessmen who are considering a purchase now or in the future."
— *West Coast Review of Books*

If you've ever considered a computer for your business but didn't know where to turn, this is the book that will arm you with all the information you'll need to make an intelligent, cost-effective decision.

ISBN 0-918398-31-2 $12.95

Instant BASIC

Jerald Brown

"In short, this book is a winner."
— *Kilobaud*

Here is an active participation workbook designed to use on your home computer. It's an easy, painless way to learn BASIC.

ISBN 0-918398-21-5 $10.95

Computers for Everybody

Jerry Willis and Merl Miller

"The wide range of basic material covered, plus a chapter that evaluates current computers on the market, makes this an attractive guide. . ."
— *booklist*

This fun-to-read book covers all the things you should know about computers. If you're anxious to buy one, use one or just want to find out about them, read this book first.

ISBN 0-918398-49-5 $4.95

Peanut Butter and Jelly Guide to Computers

Jerry Willis

Chosen by the Library Journal as the most note-worthy computer book of 1979.

"Should interest a wide range of readers seeking an understandable work on computers." — *Library Journal*

This entertaining book is a simple, easy-to-digest source of information on personal computing. It leads you through all the essential knowledge you need to get started.

ISBN 0-918398-13-4 $9.95

From dilithium Press

Write or call us for a free catalog
dilithium Press, P.O. Box 606, Beaverton, OR 97075

Name____________________
Address____________________
City/State/Zip ____________________

dilithium Press
P.O. Box 606
Beaverton, OR 97075
800-547-1842

Editor's Note: Program in APL, FORTRAN, Assembly, BASIC, or PASCAL. Have 96K user RAM available. Connect directly to mainframes. Modify variables during a program RUN. These are some of the reasons the new PET is called Super.

We'll be testing a SuperPET here at **COMPUTE!** *and next month we'll let you know the results, when the machine will be in the stores, and 8032 upgrade news. — RTM*

SuperPET's Super Software

Terry Wilkinson
Waterloo Computing Systems

As most of our readers know, the University of Waterloo has been involved for many years in the development of computer software for its own applications. In the 1960's, a number of specialized batch systems were created for teaching computing. As well, pioneering work was done in the development of interactive terminal systems to make the large batch oriented computers easier to use. Language processors such as WATFOR, WATFIV and WATBOL were augmented by such systems as WITS (an early interactive terminal system). Through it all, the main emphasis was on enhancing the learning environment at the University of Waterloo by developing software to meet our specific needs.

In the 1970's, these efforts continued, using the concepts of distributed processing and the emerging minicomputer technology. Powerful interactive systems on remote PDP-11's and IBM SERIES/1's allowed the preparation of jobs and the examination and printing of output to take place "offline" from the computer. Waterloo developed packages like WIDJET which allowed more students to use the computer at a lower cost. High-speed BISYNC lines provided communication between the minis and the mainframes allowing access to the language processors available there. In addition, packages such as WATFOR-11 and WATBOL-11 were created to run on stand alone mini systems.

The underlying theme through all these developments was the production of systems that could make allowances for small mistakes and give good error diagnostics, thereby making the program development process a little easier. These systems have always included reasonable enhancement beyond prevailing language definitions so users could become familiar with, and make use of, the latest programming technology as it was being developed in the industry.

It is, therefore, consistent with this longstanding tradition that the University of Waterloo would extend these concepts one step further in the 1980's. That step is in microcomputer technology. It addresses the use of "stand-alone" microcomputer systems as well as their use in distributed processing.

The First Step: Waterloo Microsystems

The first step in exploiting this new technology was to apply the lessons learned over the years concerning software development. A new family of language processors was created which would support APL, BASIC, FORTRAN and PASCAL. Also, a general purpose text EDITOR was developed to allow easy manipulation of program and data files. These packages were written in a system-independent, portable manner to provide a very high degree of flexibility in implementation. The success of this approach is evident in that completely compatible versions of these packages have been installed on IBM's VM/CMS system for large 370-like machines and the new Commodore SuperPET with the Motorola 6809 microprocessor chip. Work is also underway to install them on the DEC PDP-11 system with RSTS/E.

This means that, using this family of language processors and subject to memory constraints, a program written for one of these machines will run on any other of these machines *unchanged*. This provides great flexibility in software development. Applications can be created which run on a variety of computers without modification. Components of a system can be created using one type of computer and then used on another type of computer.

In addition, a 6809 Assembler Development System was created to provide a straightforward, yet powerful, facility to create programs in machine language for use on the SuperPET or other 6809-based systems.

The second step was to provide a simple-to-use interface between the microcomputer and the mainframe computer. Programs and files of data would need to be transferred from one machine to the other. And, data files on the mainframe should be easily accessible by the programs running on the microcomputer.

Since most large-to medium-size computers support ASCII-type terminals, the approach involved an RS232-C serial line from the micro to the mainframe. Also required was an interface program for the mainframe to service the data management requests from the micro in the appropriate way. This interface program is called HOSTCM, standing for "host communication module."

Because of this approach, programs on the microcomputer can access data files on a local disk

JINI MICRO-SYSTEMS, Inc.

DATABASE MANAGEMENT SYSTEM DESIGN

BOX 274 KINGSBRIDGE STN., RIVERDALE, N.Y. 10463 (212) 796-6200 or on a remote host disk *with equal ease.*

The result of these developments is a very powerful collection of hardware and software which can be used in various configurations to service the needs of a wide range of users. For example, a stand-alone SuperPET can have local disk(s) and printer(s) and support all five languages and the editor without any connection to a remote computer. Alternatively, a SuperPET connected to a host computer might have no local devices and keep all its files on the host. Of course, a combination of the above configurations yields a powerful micro configuration with the additional ability to transfer information to and from the host machine.

A SuperPET With: APL, BASIC, FORTRAN, and PASCAL

It is clear that the IBM 370-like machines and the PDP-11's were quite capable of handling their part in a system such as the one described above. It was not as easy, however, to discover a microcomputer which could be used in this design. The best approach seemed to be to modify an existing micro and thereby give it the required facilities. Our previous involvement with the Commodore PET had given us considerable knowledge of its construction and we felt confident that the CBM 8032 could be modified to do the job. It had a MOS6502 micro-processor chip and 32K (kilobytes) of user RAM.

Three fundamental changes were required:

a) conversion to the Motorola 6809 microprocessor chip
b) addition of more RAM
c) addition of an RS-232 serial interface

It also seemed desirable to retain the previous 6502 processor and allow the machine to operate as a normal CBM 8032, if desired. This would preserve its ability to run already existing packages developed for the 6502.

The initial Waterloo designs were taken by BMB CompuScience Ltd. of Milton, Ontario and developed into a working prototype which could be mass produced. This firm used its considerable experience in hardware design to produce the final product which contains two microprocessors, an MC6809 and an MOS6502. An external switch was included to allow the user to select one mode or the other. In 6502 mode, the machine operates as a CBM 8032 using Commodore BASIC in ROM and has a 32K RAM. In 6809 mode, a different ROM is selected. At the same time, Waterloo Computing Systems Ltd. undertook the task of implementing system software to operate using the BMB hardware configuration in 6809 mode. The following list of software was implemented: the Waterloo microSystems Supervisor (resident in the 6809 ROM set); interactive interpreters for APL, BASIC, FORTRAN and PASCAL; and a development system for 6809 machine language programming. Subsequently, Commodore has begun manufacturing this hardware configuration, called the "SuperPET," under license from BMB and is including the entire collection of software with each machine sold.

An additional 64KB of RAM was installed to allow room for the Waterloo microSystems language processors. The user selects which language he wishes to use from a menu which appears on the screen when the unit is turned on. The processor for that language is "soft-loaded" into the additional RAM. This means that the user still has the entire 32K of original RAM available for his use regardless of which language he chooses.

The technique is called *bank-switching* and... 64K of RAM is logically divided into 16 *pages*, each containing 4K.

The usable space in the 16-bit address structure of the 6809 system was almost fully allocated and a special technique was required to allow addition of the 64K of RAM which the processors would need. The technique used is called *bank-switching* and, with it, the 64K of RAM is logically divided into 16 *pages*, each containing 4K. A 4K *window* in the address space can be positioned over any one of these 16 pages by simply setting a byte in the I/O area of memory. In the SuperPET, this window occupies addresses 9000-9FFF (hexadecimal) in the address space (see figure).

The Language Processors

The various high level languages are implemented in the Waterloo microSystem by means of interpretive language processors. This means the APL, BASIC, FORTRAN and PASCAL programs are stored internally in an encoded format. These encoded statements are then interpretively executed by a "run-time" supervisor unique to each language. Such an approach makes it possible to stop program execution, examine and modify program variables and data, and then resume execution from the place where it suspended. In APL and BASIC, it is even possible to interrupt and modify the program itself and then continue execution.

All the languages interface with the *host* computer, the serial line and the various disks and printers on the IEEE-488 bus using a common file system interface supplied in the Waterloo microSystem library. It provides 100 percent compatibility among data files across the various language processors. These file system functions, and many

NEW FROM CYBERIA ★ NEW FROM CYBERIA ★ NEW FROM CYBERIA

Mari

A comprehensive accounts receivable and insurance billing system for modern health care offices and clinics

DESIGNED FOR COMMODORE 8000 SERIES COMPUTERS AND DISK DRIVES

- Help functions are always on-line
- Supports CPT, ICD and RVS medical coding
- Supports open item or balance forward accounting systems
- Interfaces to popular word processing programs
- Multiple terminals may be added with Superbus 4.0
- Multiple disk drives may be used—no limit on number of patients or accounts
- Includes a data base and forms generator to fill out any insurance form
- Includes a computer aided instruction program to train new users

Includes these standard reports:

- A/R aging
- A/R transactions
- General ledger
- Instant cash receipt
- Income analysis by physician
- Doctor referral report
- Patient/account cross reference
- Standard SuperBill insurance form

NEW FROM CYBERIA ★ NEW FROM CYBERIA ★ NEW FROM CYBERIA

SUPERBUS 4.0

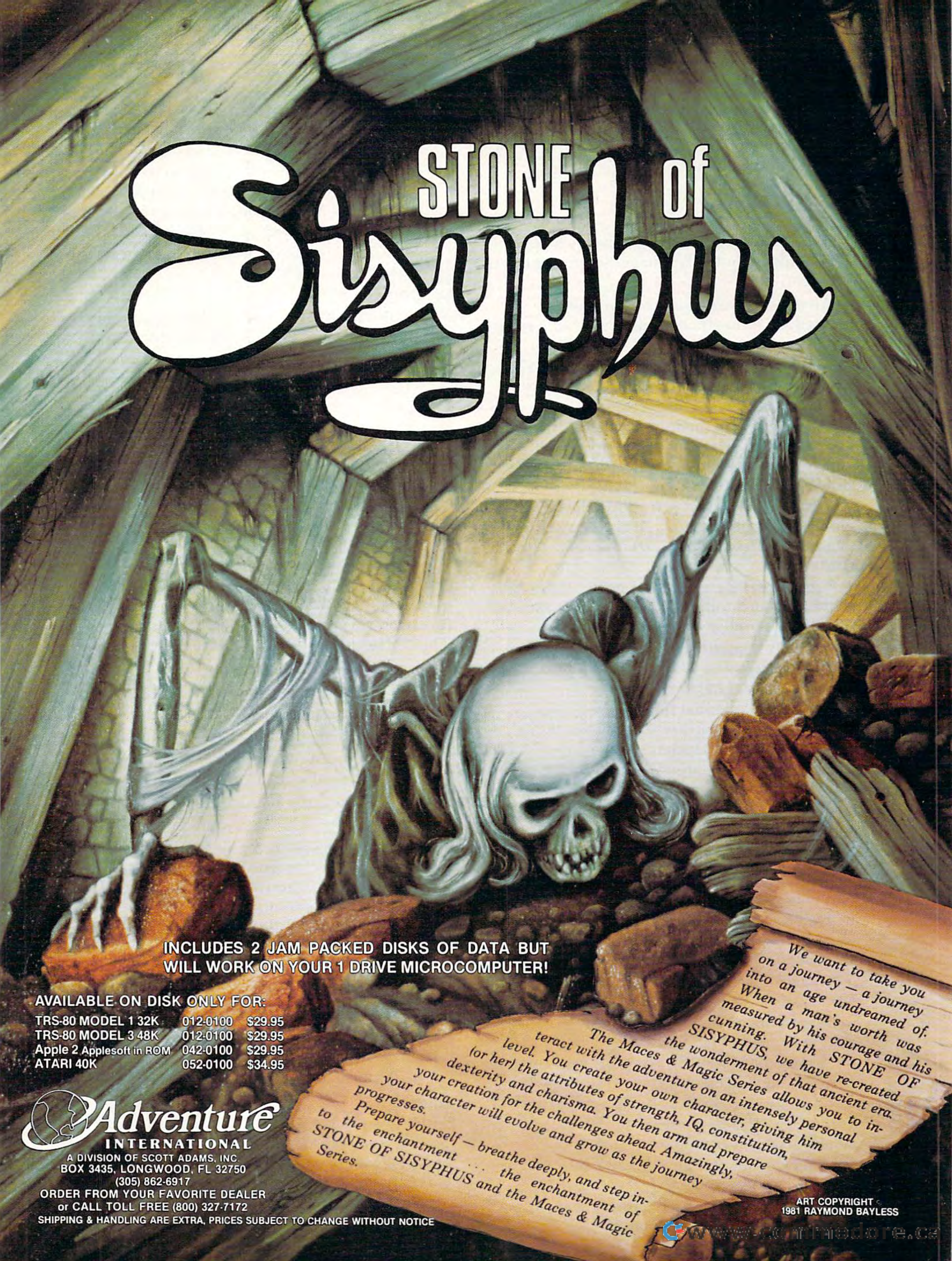
Commodore Computer Networking System

- Up to 18 computers can be interconnected
- Multiple disk drives, printers and other devices may be added to the network
- Sophisticated security system prevents unauthorized use
- WordPro and Wordcraft programs are supported to allow multiple terminal word processing

NEW ★ NEW ★ NEW

*WordPro is a trademark of Professional Software, Inc.

CYBERIA INC. 2330 LINCOLN WAY, AMES, IA 50010 • 515-292-7634



STONE of
Sisyphus
INCLUDES 2 JAM PACKED DISKS OF DATA BUT WILL WORK ON YOUR 1 DRIVE MICROCOMPUTER!
AVAILABLE ON DISK ONLY FOR:
TRS-80 MODEL 1 32K 012-0100 $29.95
TRS-80 MODEL 3 48K 012-0100 $29.95
Apple 2 Applesoft in ROM 042-0100 $29.95
ATARI 40K 052-0100 $34.95
Adventure INTERNATIONAL
A DIVISION OF SCOTT ADAMS, INC.
BOX 3435, LONGWOOD, FL 32750
(305) 862-6917
ORDER FROM YOUR FAVORITE DEALER
or CALL TOLL FREE (800) 327-7172
SHIPPING & HANDLING ARE EXTRA, PRICES SUBJECT TO CHANGE WITHOUT NOTICE
We want to take you on a journey — a journey into an age undreamed of. When a man's worth was measured by his courage and his cunning. With STONE OF SISYPHUS, we have re-created the wonderment of that ancient era.
The Maces & Magic Series allows you to interact with the adventure on an intensely personal level. You create your own character, giving him (or her) the attributes of strength, IQ, constitution, dexterity and charisma. You then arm and prepare your creation for the challenges ahead. Amazingly, your character will evolve and grow as the journey progresses.
Prepare yourself — breathe deeply, and step into the enchantment ... the enchantment of STONE OF SISYPHUS and the Maces & Magic Series.
ART COPYRIGHT © 1981 RAYMOND BAYLESS
www.commodore.ca

others such as trigonometric functions, string manipulation routines and a floating point emulator, are available through documented interfaces to the machine language program as well using the 6809 Assemble Development System.

System Highlights

APL

The APL processor follows the well known IBM/ACM 79 standard for the language. Some highlights of the system:

— All the standard primitives are implemented, including matrix-divide, dyadic-transpose, format, and execute.

— Functions such as quad-CR and quad-FX allow the dynamic creation and modification of functions.

— Direct access to memory and machine language programs is provided with quad-PEEK, quad-POKE, and quad-SYS (including generalized parameter-passing).

— A powerful, full-screen editor allows easy modification of functions. It also accepts indentation and comments to enhance program readability.

— The SuperPET screen supports all the APL characters (including overstrikes) and also provides a number of common graphics.

— An APL-sequential file feature allows storage and retrieval of complete APL data items including rank, shape, and type.

— A BARE-sequential file feature allows transmission of arbitrary strings of bytes in and out of the workspace.

— Relative files are also supported.

BASIC

The BASIC processor includes the ANS Minimal BASIC standard features as well as several noteworthy extensions:

— Variable, Array, Function and Procedure names can each be uniquely defined using up to 31 characters.

— Multi-line functions and procedures can be written and called with parameter passing. This feature can be used to implement recursive algorithms.

— The family of MAT (matrix manipulation) statements are implemented.

— A number of structured control statements provide a facility to enhance program style.

— Support a program text indentation and comments following statements further enhance program readability.

— A powerful generalized string/substring feature has been included.

— Built into the system is a broad set of functions to perform common operations such as SIN, COS, LEN, HEX$, ORD, and VALUE. There are about 35 such functions available.

— Error trapping allows interception of, and recovery from, most run-time errors.

— Commands such as RENUMBER, AUTOLINE, and MERGE allow easy manipulation of the BASIC program source code.

A full-screen editor makes changing existing statements simple.

FORTRAN

The FORTRAN processor implements a powerful subset/extension of the standard language. It includes many of the popular features of the well known WATFIV-S compiler as well as many features described in the FORTRAN-77 standard. This "Waterloo dialect" of FORTRAN includes:

Multi-line functions and procedures can be written and called with parameter passing.

— FORMAT statements

— Subroutines and functions

— Multi-dimensional arrays

— Extended character string manipulation

— Structured Program Control statements

— Sequential and Relative file support

— An interactive debugging facility

PASCAL

The PASCAL processor implements a version of the language which corresponds closely to the draft of the International Standard Organization (ISO) PASCAL committee, a refinement of the original language definition by Jenses and Wirth. Features include:

— Text file support

— Pointer variables

— Multi-dimensional arrays

— An interactive debugging facility including breakpoints, single step etc.

— Extensive data-typing capabilities

Assembly

The EDITOR is a processor which provides a powerful means of creating and maintaining general data and program files. It is a line-oriented, contextual editor with many features:

— GET and PUT commands retrieve entire

At long last. A complete line of Voice I/O peripherals for Commodore computers, starting at $119.

Let's face it. Voice I/O is a fascinating and efficient way to communicate with computers. And now, thanks to VOICETEK, Voice I/O peripherals are easily available, easy to use and very affordable.

If you own a Commodore computer, we give you a choice of three different peripherals that will enable your computer to understand your spoken commands. And two of these peripherals will talk back.

What's in a name.

We call each member of our family of Voice I/O peripherals COGNIVOX, from COGNItive and VOX. It helps you remember that when you talk, COGNIVOX thinks about what you say.

The top-of-the-line COGNIVOX model for the PET/CBM is VIO-1002. It offers natural sounding voice output and excellent performance as a speech recognizer. It costs only $249.

If the quality of voice output is not important for your application, then you can save $100 by ordering VIO-432. Priced at $149, VIO-432 is ideal for hobbyists or persons mainly interested in speech recogniton.

Finally, if you have an 8K PET, there is insufficient memory for voice response, so we offer a recognition-only COGNIVOX, model SR-100P. It costs $119, making it the lowest priced speech recognizer ever offered for sale. Yet its performance rivals that of units selling at much higher prices.

Which brings us to the next point we would like to make, namely, why we offer so much performance for so little money.

It's the technology.

Our Voice I/O peripherals are based on a technological breakthrough that made it possible to compress the required electronics onto a single integrated circuit chip. We are the only company so far that has achieved this remarkable feat. No wonder we offer such reasonably price voice peripherals.

In addition, COGNIVOX uses an exclusive non-linear, learning pattern matching algorithm to do speech recognition. Which means more reliable performance and ease of use.

What makes it talk.

COGNIVOX digitizes and stores in memory (using a data compression algorithm) the voice of the user. This gives three major advantages:

First, there are no restrictions to the words COGNIVOX can say. If you can say it (or sing it, or whistle it for that matter) your computer can do it too. Second, It is very easy to program your favorite words: just say them in the microphone.

Third, you have a choice of voices, male, female, child, accents, etc. this unprecendented flexibility offered by COGNIVOX is a must in the personal computer environment. Voice synthesizers and the "talking chips" do not offer this flexibility and therefore we feel they are not suitable for use with personal computers. In addition, voice output quality can be poor, especially for synthesizers. In that respect, VIO-1002 is clearly superior to anything else on the market and it is a must if voice quality is important (for example, business applications).

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.

Training COGNIVOX to your vocabulary is easy. All you have to do is repeat the words three times at the prompting of the computer.

If you would like to have COGNIVOX respond to more than 32 words, you can have two or more vocabularies of 32 words and switch back and forth between them using a word.

The Voice output vocabulary can have up to 32 words phrases. Data rate is approximately 700 byte per word.

Ready to listen.

All COGNIVOX units are complete Voice I/O peripherals ready to plug in and use. They come assembled and tested and they include microphone, cassette with software and manuals. VIO units include built-in speaker and amplifier (yes, CB2 is also connected for music and sound effects).

They all plug into the user port and they receive their power from the cassette port except VIO-1002 which uses a wall transformer supplied with the unit.

Easy to use.

All you need to get COGNIVOX up and running is to plug it in and load one of the programs supplied. Load the demo 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 a word or to recognize a word. Full instructions on how to do it are given in the manual.

Works with all versions.

COGNIVOX will work with all versions of the PET/CBM line. Old, new and newer ROMs. At least 16K of RAM is required (SR-100P will work with 8K of RAM).

If you have a disk system, you can use it to save vocabularies. Instructions are given in the manual.

Many uses.

With COGNIVOX your imagination is not the limit as the saying goes. It is the starting point. Cognivox is a super toy, an educational tool, an aid to handicapped, a data entry device while hands and eyes are busy, a foreign language translator, a sound effects generator, a telephone dialing device, an answering machine, a talking calculator. Using the IEEE 488 port you can control by voice instruments, plotters, test systems. And all these devices can talk back to you, telling you their readings, alarm conditions, even their name.

Order your COGNIVOX now.

To order by mail, give us the model number of the unit you wish to order, the make and model of your computer and your name and address. Enclose a check or money order and make sure to include $5 for shipping and handling. CA residents please add 6% tax. You may also order by phone and charge it to your Master Charge or VISA. Our phone number is (805) 685-1854 9AM to 5PM PST, Monday through Friday. Foreign orders are welcome, please add 10% for air mail shipping and handling. Payments must be in US funds. COGNIVOX is backed by an 120-day limited warranty against manufacturing defects.

VOICETEK

P.O. Box 388, Goleta, CA 93116

files of data into the machine for editing and then save them back on external devices.

— SEARCH and CHANGE facilities, including global change, provide for easy modification of text.

— A full-screen capability allows individual changes to be made by simply moving the cursor around the screen and entering the changes.

— A set of function keys provides extensive cursor movement and scrolling facilities.

The 6809 Development System is comprised of an Editor, and Assembler and a Linker. Code entered via the Editor can be assembled into relocatable modules by the Assembler. Then these modules are combined and relocated by the Linker to produce an "executable load module." This load module can be loaded into the machine, executed, and debugged using the Monitor built into the Waterloo microSystems Supervisor. Some features of the Assembler are:

— Motorola 6809 Assembler language
— Macro capability
— Pseudo-opcodes for structured programming
— Long label names allowed
— Produces relocatable object code

Some features of the Linker are:

— The ability to combine many relocatable object files into a single load module.

— Relocates code to any arbitrary machine location

— Supports the "bank-switched" RAM feature of the SuperPET making it almost transparent to the user.

— When used with the supplied file "WATLIB.EXP," provides automatic symbolic linkage to the functions available in the Waterloo microSystems Library.

The Operating System

The Waterloo microSystems Supervisor resides in the ROM of the SuperPET. It has many of the features of a true operating system. These include:

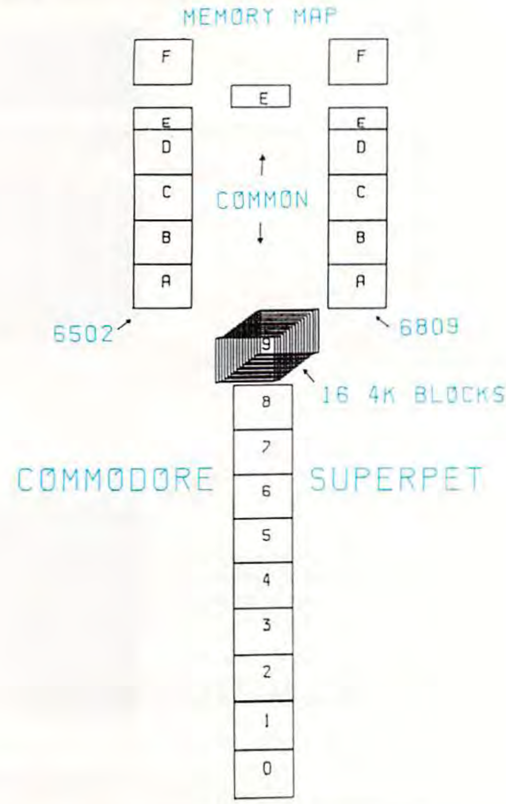
— The facility to load 6809 machine language programs into the SuperPET RAM including bank-switched memory.

— Full-screen monitor facilities to examine and modify arbitrary locations in memory; bytes are displayed in both hexadecimal and character formats.

— A built-in disassembler to facilitate examination of programs residing in the machine.

— A built-in library of functions used by the high level language processors, but available to the machine language programmer.

Figure 1.



SOFTWARE HARDWARE FOR THE VIC 20®

SOFTWARE FOR THE VIC 20

TAPE CASSETTES

- **VicTerm A** — **$19.95**
  Be smart! Operate your VIC 20 as a dumb terminal.
- **viCalc I** — **$14.95**
  First in a series of useful calculator programs. 10 memorys with arithmetic, 4 stacked data registers. Math functions plus compound interest tables.
- **VPM** — **$14.95**
  Securities Portfolio Management. Important records on tape for 25 securities.
- **AMOK** — **$18.95**
  The halls of AMOK are populated by robots that follow one instruction ... get the intruder. That's you. To save yourself, you must be quick on the draw and fast on your feet.
- **SIMON** — **$9.95**
  Test dexterity and memory by repeating the flashing colors and tones—different every time.
- **PAC IT IN** — **$18.95**
  Fast action color game. Two players can ZAP ROBOTS before they "**PAC IT IN**."
- **SUPER ADDITION OR SUBTRACTION** — **each $9.95**
  Blackboard feature gives useful math practice with interest enhanced by color. Correct steps shown with carries.

HARDWARE FOR THE VIC 20

- **UMI RS232 COMMUNICATIONS INTERFACE** — **$49.95**
  For printers and telephone communications.
- **UMI 3K MEMORY EXPANSION** — **$79.95**
  With addressable and switchable ROM slots for ROM programs to 16K.
- **UMI 8K RAM EXPANDER** — **$99.95**
  Provides 11775 Bytes (characters) of user memory.
- **UMI EXPANSION CHASSIS** — **Price to be announced**
  For additional memory or program cartridges.

SOON TO BE AVAILABLE

SOFTWARE CARTRIDGES

- **SPIDERS OF MARS**
- **SATELLITES AND METEORITES**
- **ROBOT BLASTERS**
- **ASTRO TRANSPORTERS**
- **3D INVADERS**

ORIGINAL! EXCITING! UNIQUE!

DEALER INQUIRIES INVITED

Clip Coupon for **FREE CATALOG**
Catalog available in Spanish (Version Disponible en Español)
MasterCard/VISA Accepted

United Microware Industries Incorporated
3431 H Pomona Boulevard
Pomona, Calif. 91768
Phone (714) 594-1351

VIC 20 is a Registered Trademark of Commodore Business Machines

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 ______________________

— A serial-line setup routine to set the baud rate, stopbit, and parity used to establish the host-protocol parameters for mainframe communications with the HOSTCM program.

— A "passthru" terminal mode to communicate directly with a remote computer via the serial RS232 line. ©

SuperPET: A Preview

Bill MacLean
BMB Compuscience
Milton, Ontario

The day is near when we will all start to see a lot of SuperPETs, so we thought it was time to describe the system and try to speculate on the significance of the design. In reality, the SuperPET is two different computers in the same cabinet. It is the old (today!) 8032 with which we are all familiar, with a few new wrinkles. It is also a brand new 6809 based computer system with outstanding potential in its own right. We will try to evaluate it in each of these two categories.

SuperPET — 6502

Looked at one way, the only difference between the SuperPet and the 8032 is the RAM capacity and the I/O system. There is 64K of additional RAM in the SuperPET. It is mapped as 16 4K blocks all residing in the block $9000-$9FFF. This may seem impossible, but it is done by write-only register at $EFFC (61436) which allows the programmer to select which of the 16 blocks can be read and written to in the $9000 block. For example, to select the 15th block, the register at $EFFC numbered 0-15.) In addition to this extra RAM, there is an intelligent USART (Universal Synchronous, Asynchronous Receiver Transceiver) or RS-232C port located at the base address $EEF0 (61424). This device allows the programmer the option of software selecting such serial communications parameters as Baud rate, word length, number of stop bits, parity, etc. The USART used is the 6551 system. It is a chip currently being manufactured by MOS Technology and is simulated in software on the VIC 20.

The first question everyone seems to ask about the RAM capacity is "will my Visicalc use the extra memory" or "will my BASIC programs now access the additional memory?" The answer is *no*. It would require substantial modification for any of the existing operating systems to use this additional memory. However, this is true of any conventional memory expansion scheme that takes one over the 65K boundary on an 8-bit processor. That was the bad news, now for the good. Many of your favorite programs will be modified to run on the SuperPET and will utilize the extended memory. The concept of *paged* memory is a very powerful one. Once the step has been taken to design systems using this concept, virtually unlimited RAM can be used, guaranteeing much easier expansion in the future as the price of RAM drops.

SuperPET — 6809

You may well ask what is a 6809, and why is there one in a Commodore product? This microprocessor is a logical extension of the 6800 and the 6502, just as the 6502 was a logical extension of the 6800. It is a pseudo 16-BIT processor. This means that all of the internal registers and the stack pointer(s) are 16 bits long. This allows comfortable addressing anywhere in the address bus range. There are some other major differences, particularly suitable to the creation of position-independent code. It is in the areas of compactness of run-time code, the use of the stack, and the position-independent code capabilities that the 6809 shines. These characteristics made it much more attractive to design the SuperPET software systems using the 6809.

Describing the potential of this system is a little difficult. Under the Watcom operating system, the machine is really several distinct computers. This operating system consists fundamentally of a 22K ROM set occupying the top half of the 65K address range just as the CBM system does. This area contains all system libraries, I/O routines, monitor (6809) etc. The languages (BASIC, FORTRAN, APL, PASCAL, and Assembler) all reside on a diskette in drive 1 and are called from a menu. The languages each load into the $9000 blocks and execute there. This is one of the first unusual things to notice about the system. No matter how large the interpreters are, they don't take more than 4K out of the memory map. This means that the lower 32K of 8032 memory is available to each of the languages for the program and variable storage. The facility for creating software that runs out of this paged memory is included in the assembler development system supplied with the SuperPET. This is one of the most significant contributions to the future software growth of this product.

The Manuals

On the subject of documentation, there are five manuals included in the system, one for each language. These manuals are tutorial examples. Of

Every PET Needs a Friend.

CURSOR is the best friend your Commodore PET will ever have. Since July, 1978 we have published 150 of the most user-friendly programs for the PET available anywhere. When we write or edit a program, we spend lots of time fussing about how it will treat you. We pay attention to lots of little things that help make using a computer a pleasure instead of a pain.

Naturally, **CURSOR** programs are technically excellent. Each program that we purchase is extensively edited or re-written by a professional programmer. But *imagination* is just as important as being user-friendly and technically good! We delight in bringing you off-beat, unusual programs that "show off" the abilities of your PET or CBM.

CURSOR is user-friendly, technically great and full of imaginative programs. And every issue of **CURSOR** is still available! We continue to upgrade previously published programs so that they'll work on the three varieties of Commodore ROM's (Old, New, and 4.0). New issues also work on the 80 column CBM.

For only $4.95 you can buy a sample issue and judge for yourself. Or send $18 for a four-issue subscription. Each **CURSOR** comes to you as a C-30 cassette with five programs and a graphic Front Cover, ready to LOAD and RUN on your PET.

Who knows? After your PET meets **CURSOR**, things may never be the same!

AUTHORIZED DISTRIBUTORS:

Great Britain
AUDIOGENIC, Ltd.
P.O. Box 88
Reading, Berkshire

Holland
COPYTRONICS
Bergemeester
Van Suchtelenstraat 46
7413 XP Deventer

Japan
SYSTEMS FORMULATE CORP.
Shin-Makicho Bldg. 1-8-17
Yaesu, Chuo-ku, Tokyo 103

Australasia
MICROCOMPUTER HOUSE, LTD.
133 Regent Street
Chippendale, Sydney
N.S.W. Australia 2008

THE CODE WORKS

Box 550
Goleta, CA 93116
805-683-1585

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

course, since the languages are standards, many texts are available to help the beginner.

We mentioned libraries above. It has appeared that Microsoft and Commodore have gone out of their way in the past to hide their code entry points

This bridging of the tremendous gap between traditional computerists and micro users can only benefit both sides.

from legitimiate users. I am happy to say that this is not the case with the Watcom system. The entire operating system is jump table oriented and the table entries and setup conditions are described in the very complete manuals which accompany the system. I sat down and wrote a complete disk handler with error trapping and all in under 200 bytes using the available calls and documentation.

The greatest significance of the SuperPET is the perception people will have of it. To the micro user, it is a logical extension of the Commodore product line, but to the more traditional computer community, it is something entirely different. This is probably the first small computer at a reasonable cost that supplies the operating system and languages of larger systems and the communications interface to effectively use them. The impact of having the traditional computer community using this type of machine will be felt very rapidly in software availability. People used to programming in FORTRAN (or especially APL) will be able to put complex software systems, which have been used for years, on the micro. This bridging of the tremendous gap between traditional computerists and micro users can only benefit both sides. There are an awful lot of accumulated man hours of experience from which we should be able to profit. ©

80 COLUMN GRAPHICS

The Integrated Visible Memory for the PET has now been redesigned for the new 12" screen 80 column and forthcoming 40 column PET computers from Commodore. Like earlier MTU units, the new K-1008-43 package mounts inside the PET case for total protection. To make the power and flexibility of the 320 by 200 bit mapped pixel graphics display easily accessible, we have designed the Keyword Graphic Program. This adds 45 graphics commands to Commodore BASIC. If you have been waiting for easy to use, high resolution graphics for your PET, isn't it time you called MTU?

The image on the screen was created by the program below.

```
 10 VISMEM: CLEAR
 20 P=160: Q=100
 30 XP=144: XR=1.5*3.1415927
 40 YP=56: YR=1: ZP=64
 50 XF=XR/XP: YF=YP/YR: ZF=XR/ZP
 60 FOR ZI=-Q TO Q-1
 70 IF ZI<-ZP OR ZI>ZP GOTO 150
 80 ZT=ZI*XP/ZP: ZZ=ZI
 90 XL=INT(.5+SQR(XP*XP-ZT*ZT))
100 FOR XI=-XL TO XL
110 XT=SQR(XI*XI+ZT*ZT)*XF: XX=XI
120 YY=(SIN(XT)+.4*SIN(3*XT))*YF
130 GOSUB 170
140 NEXT XI
150 NEXT ZI
160 STOP
170 X1=XX+ZZ+P
180 Y1=YY-ZZ+Q
190 GMODE 1: MOVE X1,Y1: WRPIX
200 IF Y1=0 GOTO 220
210 GMODE 2: LINE X1,Y1-1,X1,0
220 RETURN
```

K-1008-43M Manual only $10 (credited toward purchase)
k-1008-43 Complete ready to install package $495

Mastercharge and Visa accepted

Write or call today for our full line catalog describing all MTU 6502 products, including our high speed 8" Floppy Disk Controller for up to 4 megabytes of PET storage.

MTU
Micro Technology Unlimited
2806 Hillsborough Street
P.O. Box 12106
Raleigh, NC 27605, U.S.A.
(919) 833-1458

NOW 80 COLUMN PETS CAN HAVE MTU HIGH RESOLUTION GRAPHICS

Japanese Micros: A First Look

Andy Gamble
Columbia College, Vancouver, Canada

Surprise, surprise! The Japanese passion for things small and electronic has finally turned to the world of the computer. The expertise the Japanese brought to transistor radios, stereos, and — dare I say it? — cars, now moves into microcomputer technology. A familiar story?

Let me backtrack a little. I first visited Japan in spring 1980, and, although this was just a holiday, I couldn't resist investigating what they had to offer in the way of micros. (I think this particular disease has been termed "microholism.") The strange thing was that I couldn't find hide nor hair of the little beasts. It seems that personal computers were almost unknown in Japan then.

At last I found the Bit Inn in Tokyo: it wasn't that it was hidden, just low-profile. It seemed a typical computer shop by western standards: some PETs, Apples, TRS-80s. But what was that over there? Yes, a true Japanese micro, the Sharp MZ-80. And was that a machine made by NEC? More about these later.

Try as I might, I couldn't find much further evidence of personal computers in Japan (I feel that I'm going to be contradicted on this) until my second visit a year later. By 1981, the whole story seemed to have changed. It wasn't that there were significantly more computer shops around, but most large department stores (and they get really large in Japan) now had their own micro department.

Here, then, is a brief rundown on some machines available today in Japan. The prices given are the result of a rough conversion of yen to dollars which, as we all know, is subject to change. It should also be noted that Japanese consumer products are often sold at a discount of up to 20 percent.

SHARP MZ-80B

Designers of personal computers should take a long, close look at the Sharp MZ-80B. Its modern silver-gray case contains a 10" monitor, a cassette drive with "logical" tape control, and a full-size keyboard. In a word, it looks really, ahem, sharp.

The Z-80A microprocessor runs at 4Mhz and can support up to 64K of RAM. High-resolution graphics capability enables the Japanese katakana symbols to be written. The cursor keys are exceptionally appealing. Ten user-defined function keys complement the keyboard layout. The machine is RS-232 and IEEE-488 compatible.

Apart from the usual BASIC, other languages and operating systems available include FORTRAN 80, COBOL 80, a BASIC compiler, FORTH and CP/M. The prices is around $1400. Sharp also manufactures the MZ80BP5 dot-matrix printer, an 80 cps machine selling for $700, the MZ-80CR card reader, capable of reading 150 cards per minute and the MZ-80BF dual mini-floppy disk drive with 572 Kbyte storage for $1500. A truly impressive system.

NEC PC-8000 Series

NEC is clearly aiming the $800 PC-8001 micro at the business market, but interest in the machine is very strong from the educational and scientific-professional fields. The PC-8001 supports high-resolution color — the monitor is $450 extra — and has perhaps the best color I have ever seen. The CPU is a PD780C-1, which apparently is similar to the Z80A, and runs at 4MHz. The Japanese seem to be big on function keys: the PC-8001 has five.

This series also includes the 8023 printer, a dot-matrix type at $900, and two dual disk drives of 286Kbytes at about $1300. Is the cheaper one an "expansion" disk drive? I couldn't find out.

CASIO FX-9000P

Casio is a well known name in North America for its range of calculators and watches, but in Japan also for computers and electronic musical instru-

NEW! EDUCATIONAL SOFTWARE FOR CHILDREN

from MICROGRAMS, INC.

- Designed to supplement curriculum in Grades K-8
- Classroom tested
- Programmed with sound and advanced motivational graphics
- Tutorial and Review Programs available for Math, Reading, Language, Science, Social Studies, Vocabulary, Spelling, and special interest areas
- Testing and Grading programs for Teachers

Expect more from your PET®. . .

MICROGRAMS

INCORPORATED

P.O. BOX 2146, LOVES PARK, IL 61130

PHONE 815/965-2464

☐ Please send me a free catalog.

☐ Please send me a sample program and a free catalog. I have enclosed $2.00 for postage and handling.

NAME ______________________

ADDRESS ______________________

CITY ______________________

STATE ______________________ ZIP __________

PET is the registered trademark for Commodore Business Machines, Santa Clara, CA.

ments. The FX-9000P is another well designed machine with an integral monitor. The Z80A microprocessor runs at 2.75MHz and supports up to 32K RAM with high-resolution graphics. Most BASIC keywords are available from the keyboard with a single stroke of a key — a nice time saver. There are several statistical commands. BASIC itself, and expansion RAM (16K dynamic), is obtained by plug-in ROM-packs inserted into the front of the machine. The base price is $800, with the ROM-packs costing around $90 each.

Fujitsu Micro 8

Hold onto your hats. This machine is especially impressive. For around $1200 you get the following: Not one, but two microprocessors, both 6809s, which address up to 128K. Available RAM is only 32K, but that's because the resident BASIC is huge and color is supported with high-resolution graphics. Some idea of the power involved can be guessed from the graphic commands CIRCLE, CONNECT, SYMBOL and PAINT. UCSD Pascal and FLEX are also available.

For the Japanese, one of the delights this computer offers is the ability to reproduce the three different Japanese scripts: katakana (used for foreign words), hiragana, and kanji. When you realize that katakana and hiragana each contain about 50 characters, and a small, usable subset of the kanji (Chinese characters) would amount to several hundred characters, you cannot fail to be impressed. Normal English letters, numerals, and symbols are also there, of course.

Bubcom 80

Apart from the now usual Z80-based, 64K user RAM, 640x200 pixel high-resolution color display (optional) and eight function keys, I would have to admit that this machine is rather strange. It has 64K of bubble memory. That's right, bubbles. Eight inch disk drives with up to 1.2 Mbytes of memory are available. Are we looking at the future generation of personal computers here, with so much memory they can remember your shopping lists for 10 years? Prices start at $1800.

The Rest

Whereas you'd expect electronic concerns like Sharp to be producing computers, it came as a bit of a shock to discover how many hi-fi companies were also in the game. HITACHI, for instance, manufacturers the MB-6890, whose 6809 CPU can handle 32K of color with five function keys (again) and an RS232 interface. Plug-in boards are accepted in a similar manner to the Apple, all for $1500. The same series includes the MP-1040 printer at $900 and the MP-3540 dual disk drive at $1500.

TEAC produces the 4MHz Z80A-driven PS-85: the case for the monitor also contains the two disk drives. FORTRAN is available on this system.

SANYO makes several monochrome monitors, including the popular DDN-120C at $230.

As for "foreign machines," there's still a great interest shown in Commodore, Apple, and Radio Shack products. It's no secret that Commodore introduced the VIC into Japan first, where it's had time to create a strong following. Be on the lookout for some impressive VIC programs to surface from Japan. The VIC-20 is called VIC-1001 there, but the only difference that I can see is the yen symbol where the English pound sign is, and the katakana characters replacing one set of graphics characters on the keys. The VIC-1001 costs about $350 in its 5K form, more than holding its own against competitively priced Japanese models.

Exactly who buys these machines? No doubt the computer companies would like to say that scientists, professionals, and educators do: the advertising is certainly aimed at those people. And, true, it is not yet common for people to have computers for fun, or for their own erudition, as it is in North America. But then why is it that the software offered for sale includes so many games? The truth is that Japan is a nation besotted with video games (America is fast catching up), and this is reflected in the programs that are produced. Not just passable — *perfect* imitations of the popular video games are to be found for all the major machines. Most Japanese computers support the required high-resolution graphics and sound for this.

In The Coming Years

The Japanese have recently announced that their government and private sector will cooperate on a ten-year plan: the development of a "fifth generation" computer. The goal is the development of a machine which will perform with great power. It is hoped that the machine will have the ability to write its own programs, understand spoken human commands at a quite sophisticated level, and accomplish other tasks which are far beyond current technology. By "fifth generation," the Japanese seem to mean, essentially, artificial intelligence.

What impact will the present Japanese microcomputers have on world markets in the next few years? Will they manage to surge ahead in high-speed processor technology, in memory size and density, in artificial intelligence? Computerists around the world will be paying attention to the Japanese efforts in coming years.

LETTER QUALITY WORD PROCESSOR PRINTER/TYPEWRITER

FOR NEC, APPLE, TRS 80, COMMODORE, ATARI, H.P., OSBORNE 1

OLYMPIA ES100

- 92 character electronic keyboard
- 8 character buffer memory
- Dual pitch, 10 and 12
- 17.5 C.P.S.
- All settings from keyboard
- Auto. correction
- Daisy type print mechanism
- Cartridge ribbons
- 14 1/8 inches writing line
- 1400 dealers nationwide

REN TEC ES

- Installation in 15 minutes using existing ES100 cables
- CMOS logic for minimal drain on ES100 power supply
- Hi or low true status bits
- Accepts RS232 serial with 7 crystal controlled Baud rates
- Accepts Centronics parallel interface
- Selectable auto. line feed

$1495.00* TYPEWRITER & INTERFACE **$295.00*** INTERFACE

RENAISSANCE TECHNOLOGY CORPORATION

3347 VINCENT ROAD
PLEASANT HILL, CALIFORNIA 94523
(415) 930-7707

NEC Dot Matrix Printer	795.00
100 CPS	
Bidirectional printing	
Friction and tractor feed	
Parallel Interface	
Single ribbon cartridge	
ATARI 10-Key Accounting Pad	124.95
NEC Monitors	
12'' Green Screen	285.00
12'' RGB Color	1095.00
12'' Composite Video	430.00

DEALER INQUIRIES WELCOME

Perfectly Balanced

educational software

from

MICRO-ED

for

PET® and VIC®

Send for our free catalog*

*please specify PET or VIC

MICRO-ED, Inc. • P.O. Box 24156

Minneapolis, MN 55424

or telephone us at (612) 926-2292

VIC and PET are the registered trademarks for Commodore Business Machines

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**
4116-200 ns RAM (NEC)				**8 for 19**
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

THE STAR MODEM

From 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.

VIC Chipaway - a breakout game using Atari paddles	7
VIC Moon Mission	7
VIC Micro-Coder - a machine language utility program	10
VIC Blue Meanies from Outer Space	9
VIC 21 - casino style Blackjack	9
VIC Slither/Super Slither	9
VIC Biorhythm	9
VIC Space Math	9
VIC Car Chase	9
VIC viCalc - sophisticated visible calculator	14
VIC Simon - audible and visual memory game	9
VIC 3K RAM (with 2 ROM Sockets)	65

CP/M Handbook (with MP/M) Zaks	11.85
Programming the 6502 - Zaks	10.00
Microprocessor Interfacing Techniques	13.00
PET/CBM Personal Computer Guide (Osborne)	12.00
PET and the IEEE-488 Bus (GPIB)	12.25
6502 Assembly Language Prog. - Levanthal	13.50
Z-80 Assembly Language Prog. - Levanthal	12.75
8080A/8085 Assembly Language Programming	12.75
Z8000 Assembly Language Programming	16.00
6809 Assembly Language Programming	13.50
6800 Assembly Language Programming	12.75
The 68000 Microprocessor Handbook	5.75
BASIC and the Personal Computer (Dwyer)	11.00

commodore

CBM-PET SPECIALS

	list	SALE
8023 Printer - 136 col, 150 cps bi-directional	(995)	775
8300 (Diablo 630) Daisy Wheel - 40 cps bi-directional	(2250)	1725
8032 80 x 25 CRT, business keyboard	(1495)	1100
Micro Mainframe	(1995)	1600
8096 Board (extra 64K RAM for 8032)	(500)	400
8050 Dual Disk Drive - 1 megabyte	(1795)	1375
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)	560
4022 Tractor Feed Printer	(795)	630
C2N External Cassette Deck	(75)	65
VIC 20 Color Computer	(299)	269
VIC 1515 Graphic Printer	(395)	315
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

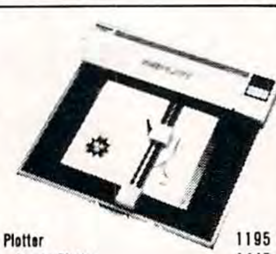
WordPro 3 Plus - 32K CBM, disk, printer	200
WordPro 4 Plus - 8032, disk, printer	325
OZZ Data Base System for CBM 8032	335
VISICALC for PET, ATARI, or APPLE	155
SM-KIT - Super PET ROM Utilities	**40**
Programmers Toolkit - PET ROM Utilities	35
PET Spacemaker II ROM Switch	36
2 Meter PET to IEEE or IEEE 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

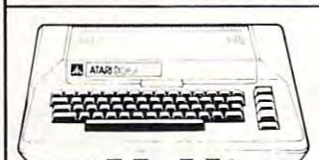
ALL BOOK and SOFTWARE PRICES DISCOUNTED
OSBORNE/McGraw-Hill, HAYDEN, SYBEX,etc.

ZENITH DATA SYSTEMS

Z19 Video Terminal (VT-52 compatible)	725
ZVM-121 Green Phosphor Monitor	115
Z89 with 48K	2150
Extra 16K RAM Board	115
Z47 Dual 8" Drive	2775

Synertek Systems

SYM-1 Microcomputer SALE	**205**
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



ATARI SPECIALS

800 Computer	749	DOS 2	21
400 - 16K	329	Pilot	68
810 Disk Drive	449	Microsoft BASIC	68
825 Printer	629	Educ. Series	20% off
850 Interface	139	**MISSILE COMMAND**	**32**
822 Printer	359	**ASTEROIDS**	**32**
Paddle Pair	17	**STAR RAIDERS**	**32**
Joystick Pair	17	Space Invaders	17
16K RAM	85	Music Composer	45
Assembler/Editor	46	Chess	30
TeleLink	20	Super Breakout	30

Write for prices on other Atari items

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

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

Microchess	17.00
Checker King	17.00
Gammon Gambler	17.00
Time Trek	13.45
Bridge Partner	13.45
Visicalc	155.00

Hayden Software

Complex Mathematics	12.70
Engineering Mathematics	12.70
General Mathematics	12.70
MCAP Circuit Analysis Program	21.00
Energy Miser	24.50

Jukebox Series for PET by L C Cargile
Excellent 4 part harmony music--write for list

Automated Simulations (EPYX) Fantasy Games

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.

Telecommunications. What Is It?

Michael E. Day
Chief Engineer, Edge Technology

Editor's Note: **COMPUTE!** *is pleased to announce Michael Day's new monthly column,* Telecommunications. *Mike is a recognized expert on this exciting aspect of computing. — RCL*

Telecommunications has a rather ominous ring to it, but, in reality, it simply means communication at a distance. Specifically, as the term is currently used, it means that some sort of equipment is required to allow the communication to occur.

Some time ago, before there was a personal computer or microcomputer, computers were very expensive. Because of this, only large companies could afford to buy and use them. Also, because of the very high cost, these buyers had to use the computer as much as possible in order to justify the cost of the computer.

Companies which were not able to utilize the computer all of the time found that they could *share* their computer with other companies who did not have a sufficient work load to justify purchasing a computer of their own.

At first, the information to be processed was hand carried to the computer site. This placed limits on the type and volume of work that could be shared. It became apparent that there was a need to communicate with the computer from a location away from the computer. It wasn't too bad when the computer was only a couple of rooms away, but from across town, existing equipment just did not work.

There was, however, a communications system in existence which had been built specifically to solve that very problem — the telephone system. Unfortunately it was designed for voice communications, not data communications.

Computers require binary (on/off) signals to communicate. The telephone system was designed to handle the continuously varying signals (analog) which make up voice signals. Because of this difference, the computer could not be directly connected to the phone lines.

The Solution

To help solve this, Bell Labs came up with a device that they called a MODEM. This allowed computers to be attached to the telephone network.

It was done by using a continuously varying signal (a carrier) that the telephone system could handle. Then, that signal is changed (MOdulated) by the computer's binary signals by a fixed amount.

At the other end of the phone line, another MODEM receives the signal and measures the change to convert (DEModulate) the signal back into the binary signals that the other, "listening" computer can understand.

The method of changing the carrier signal used is called FSK (Frequency Shift Keying).

When the computer is not sending anything, it is normally sending an ON (Binary 1) condition to the MODEM. When the MODEM hears the ON, it will transmit a signal at 1270 Hz. When the computer starts to send something it will send an OFF (Binary 0) signal to the MODEM. Hearing this, the MODEM will change the frequency it is transmitting to 1070 Hz.

At the other end of the phone line, the other MODEM does just the opposite: it hears the 1270 Hz signal that the first MODEM sent when it was transmitting the OFF signal and sends a corresponding OFF signal to its attached computer. When the first MODEM changes its signal to 1070 Hz, it recognizes this *shift* in frequency and changes the signal it is sending to its computer from an OFF condition to an ON condition.

The changing of the frequency continues as the computer sends a stream of OFFs and ONs to the other computer until it is done. It then reverts back to the continuous ON condition. This is done so that the other MODEM knows that you are still there, but you just don't have anything to transmit at this time.

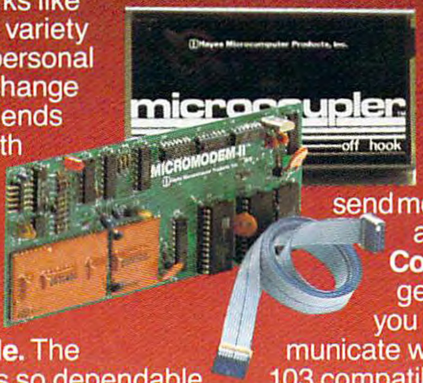
This changing, or *shifting*, of the transmitted signal is why it is called frequency shift keying.

One And Two Way Talk

Sending in only one direction is called *simplex* communication. Simplex communication does not expect (and in fact does not even allow!) any communication in the reverse direction. Television is a form of simplex communication: you can watch it all you want, but you cannot directly communicate back to the station through your TV set (at least not yet) no matter how much you might want to at times.

So here we run into a problem. What if the

With the Hayes direct-connect Micromodem II,[1]™ your Apple II can communicate by phone with the outside world. You can access information networks like The Source for a variety of business and personal applications, exchange programs with friends anywhere in North America, and even use your Apple II when you're away from your home or office.



Dependable. The Micromodem II is so dependable it comes with a *two-year* limited warranty. That's another reason why it's the largest selling direct-connect modem for Apple II computers.

Programmable. Automatic dialing and answering? Of course! We include programs on disk that dial phone numbers for you, send messages while you're away, and much more!

Complete. You get everything you need to communicate with other Bell 103 compatible modems at 110 or 300 baud. The serial interface is built-in, and we even include our FCC-approved Microcoupler™ that plugs directly into any modular telephone jack in the U.S.—you don't even need the phone!

S-100, too. The Hayes Micromodem 100 gives S-100 microcomputers all the advantages of our Micromodem II!

Put the outside world *inside* your computer with a data communications system from Hayes. Available at computer stores nationwide—call or write for the location nearest you. And don't settle for anything less than Hayes.

Hayes™

The Hayes Micromodem II opens up your Apple II to the outside world.

Hayes Microcomputer Products Inc. 5835 Peachtree Corners East, Norcross, Georgia 30092 (404) 449-8791

Micromodem II, Micromodem 100 and Microcoupler are trademarks of Hayes Microcomputer Products Inc.
[1]TM Apple Computer Inc. Micromodem II can also be used with the Bell & Howell computer.

computer you are communicating with has to send something back? What if you requested that it send a program to you? Since your MODEM is still sending the 1070 Hz signal, the other MODEM can't send the same frequency — it would be drowned out by your signal. You could turn off your transmitter, allowing you to receive the 1070 Hz signal. In fact, some MODEMs do just that, but it requires that you know that you are finished sending and also have a way of turning your transmitter off. What's more, since only one computer can communicate at one time, there is no way of telling the computer that is transmitting to shut up.

Because of these limitations, this partial two way form of communication is called *half duplex*. This term is used because both computers can communicate. It is a dual communication, but only one side can communicate at any one time.

An easier way to solve the problem would be to allow both computers to communicate to each other whenever needed. A different set of frequencies is used for one of the computers to transmit with so that there is no conflict.

When Bell Labs designed the MODEM, they chose a set of frequencies that would minimize any interference with the first set. The frequencies they chose were 2225 Hz for the ONE or ON condition, and 2025 Hz for the ZERO or OFF condition. Since both computers can fully use the communications equipment at the same time, this is referred to as *full duplex*.

Since two different sets of frequencies are used, some means of deciding which MODEM should use which frequencies had to be determined. The original application was for a remote terminal to communicate with the big computer across town.

Talking Full Duplex

Normally, the big computer would be directly attached to the MODEM so that person who wanted to use it could call it up on the telephone and the big computer would automatically answer the phone. It was decided that the set of frequencies that were chosen for the big computer to use would be called the *answer* frequencies. Since the person who called the computer originated the call, the set of frequencies that were chosen for that side's MODEM are called the *originate* frequencies.

The frequencies that were chosen were based on the use to which the MODEMs were to be put. Since the person who called the computer would normally be monitoring the information sent by the computer, it was decided to assign the less reliable set of frequencies to the answer MODEM. The person monitoring it could always ask for the information to be sent again if something went wrong. Conversely, since a computer was not as capable of detecting errors in a transmission, it was decided to give the most reliable set of frequencies to the originate MODEM.

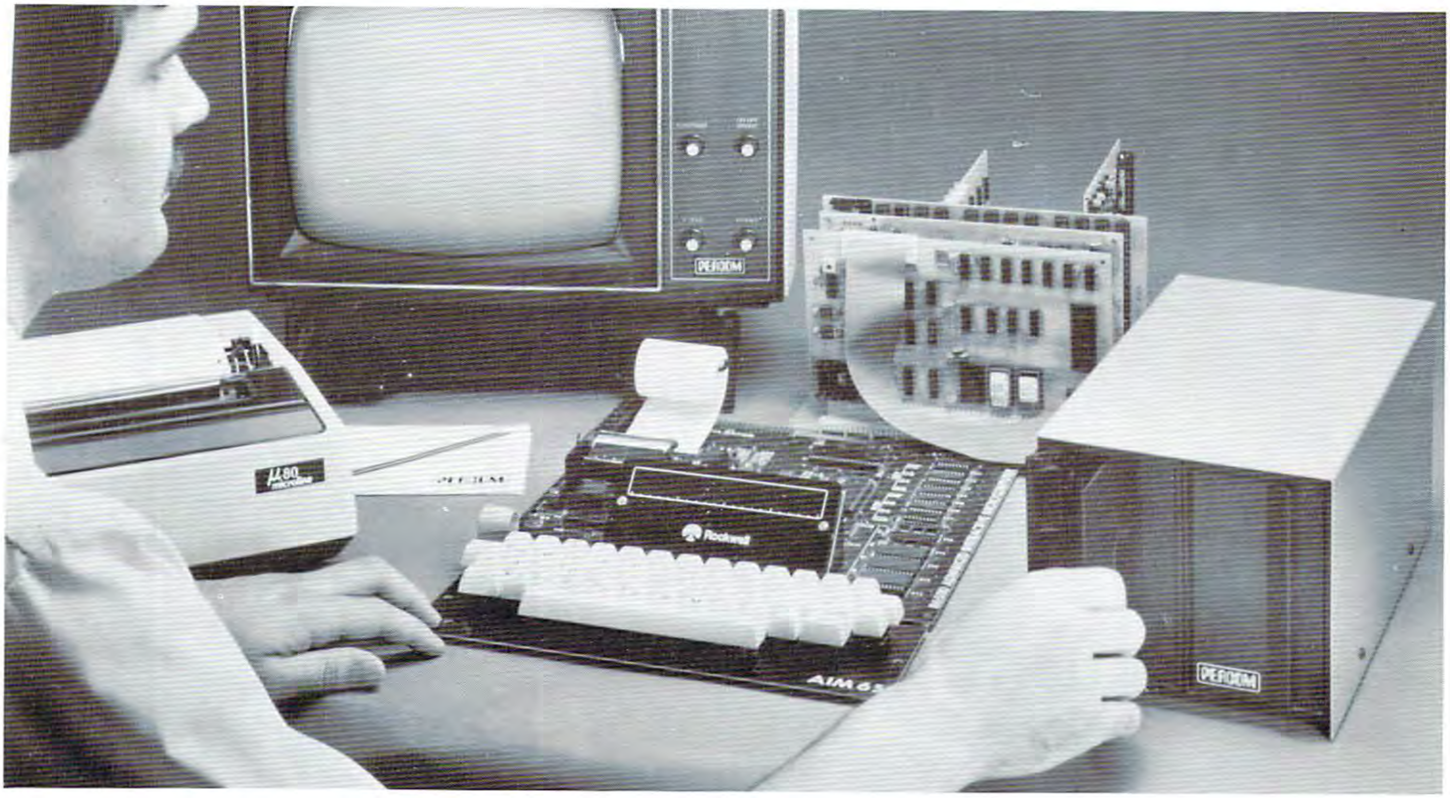
The frequencies chosen for the answer MODEM were 2025 Hz and the frequencies chosen for the originate MODEM were 1070 Hz and 1270 Hz.

Most acoustic MODEMs (sometimes referred to as acoustic couplers) are fixed as originate MODEMs. Some, however, have a switch on them which allows them to operate as an answer MODEM.

Direct-connect MODEMs (the type that is needed if automatic phone answering is desired) come in several different versions. One is *autoanswer only* (it will automatically answer the phone and connect itself to the phone line and then let the computer know that it has done so).

Another style that is available is the *autoanswer/manual originate*. This version allows the same operation as the previous version and, in addition, it can be used as an originate MODEM. Another type that is available is the *autoanswer/autodial* MODEM. This MODEM includes all of the previous capabilities and also allows the computer to make its own calls without operator assistance. Of course, this means that the computer must know how to do this: software must be in the computer to instruct it.

Next month we will explore *asynchronous* communications, another deceptively "ominous" word. ©



Introducing the M line . . .

Now! Drive Systems for AIM, KIM and SYM Computers – from PERCOM.

Assembled and tested systems start at only $599.95, including the drive controller circuit card, disk-operating system, interconnecting cable, drive and comprehensive users manual.

- **The right storage capacity** – Available in 1-, 2- and 3-drive systems, with either 40- or 80-track drives.
- **Flippy storage** – Flippy drives (optional) let you flip a diskette and store data and programs on the second recording surface.
- **High Storage Capacity** – Formatted, one-side storage capacity is 102 Kbytes (40-track drive), 205 Kbytes (80-track drive).
- **Proven Controller** – The drive controller design is the same as the design used in the Percom 680X LFD mini-disk system. This system – introduced in 1977 – has given reliable service in thousands of applications. Two versions are available: the MFD-C65 for the AIM-65 expansion bus, and the MFD-C50 for the System-50 (SS-50) bus.
  - Includes an explicit data separator circuit that's reliable even at the highest bit densities.
  - Provides for on-card firmware.
  - Includes a motor inactivity time-out circuit.
  - Capable of handling up to four drives.
  - Capable of reading both hard- and soft-sectored diskettes.
- **DOS included** – The MFD disk-operating system works with the AIM monitor, editor, assembler, Basic and PL/65 programs; interface is direct, through user I/O and F1, F2 keys.
- **Reliability assurance** – Drives are burned-in 48 hours, under operating conditions, to flag and remove any units with latent defects.
- **Full documentation** – Comprehensive hardware and software manuals are included with each system. These manuals cover details from design to operation and applications.

Now! Expand your AIM-65 with Low-cost System-50 Modules.

The Percom M65/50 Interface Adapter connects your M-65 bus to Percom's System-50 (SS-50) motherboard, allowing you to expand your AIM, KIM or SYM with proven System-50 modules. You can add disk storage, memory modules,even a video display system. The M65/50 provides buffer-amplification of address, data and control lines. On-card decode circuitry lets you allocate address space either to the computer or to the expansion motherboard. Price: only $89.95.

System Requirements: AIM-65, KIM or SYM computer with expansion bus and four Kbytes RAM (min).

Yes . . . I'd like to know more about Percom MFD drive systems. Rush me free literature.

Send to
PERCOM DATA COMPANY, Inc., Dept. 65-C
11220 Pagemill Rd. Dallas, TX 75243

name ______________________

address ______________________

city ______________ state ______

zip ______ phone number ______

MAIL TODAY!

PERCOM

PERCOM DATA COMPANY, INC.
11220 PAGEMILL RD. DALLAS, TX 75243
(214) 340-7081

Toll-Free Order Number: **1-800-527-1222**

PRICES AND SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE.

DYNACOMP

Quality software for:*

ATARI	**TRS-80 (Level II)****
PET	**NORTH STAR**
APPLE II Plus	**CP/M Disks/Diskettes**

DYNACOMP OFFERS THE FOLLOWING

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

CARD GAMES

BRIDGE 2.0 (Available for all computers) — **Price: $17.95 Cassette/$21.95 Diskette**
An all-inclusive version of this most popular of card games. This program both BIDS and PLAYS either contract or duplicate bridge. Depending on the contract, your computer opponents will either play the offense OR defense. If you bid too high, the computer will double your contract! BRIDGE 2.0 provides challenging entertainment for advanced players and is an excellent learning tool for the bridge novice. See the software review in 80 Software Critique. Rated #1 by Creative Computing.

HEARTS 1.5 (Available for all computers) — **Price: $15.95 Cassette/$19.95 Diskette**
An exciting and entertaining computer version of this popular card game. Hearts is a trick-oriented game in which the purpose is not to take any hearts or the queen of spades. Play against two computer opponents who are armed with hard-to-beat playing strategies. HEARTS 1.5 is an ideal game for introducing the uninitiated (your spouse) to computers. See the software review in 80 Software Critique.

STUD POKER (Atari only) — **Price: $11.95 Cassette/$15.95 Diskette**
This is the classic gambler's card game. The computer deals the cards one at a time and you (and the computer) bet on what you see. The computer does not cheat and *usually* bets the odds. However, it sometimes bluffs! Also included is a five card draw poker betting practice program. This package will run on a 16K ATARI. Color, graphics, sound. See review in COMPUTE.

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.

CRIBBAGE 2.0 (TRS-80 only) — **Price: $14.95 Cassette/$18.95 Diskette**
This is simply the best cribbage game available. It is an excellent program for the cribbage player in search of a worthy opponent as well as for the novice wishing to improve his game. The graphics are superb and assembly language routines provide rapid execution. See the software review in 80 Software Critique.

THOUGHT PROVOKERS

MANAGEMENT SIMULATOR (Atari, North Star 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 review in 80 Software Critique.

BACKGAMMON 2.0 (Atari, North Star 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.

CHECKERS 3.0 (PET only) — **Price: $16.95 Cassette/$20.95 Diskette**
This is one of the most challenging checkers programs available. It has 10 levels of play and allows the user to change skill levels at any time. Although providing a very tough game at level 4-8, CHECKERS 3.0 is practically unbeatable at levels 9 and 10.

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.

LEM LANDER (32K Apple Disk only) — **Price: $16.95 Diskette**
Pilot your LEM LANDER to a safe landing on any of nine different surfaces ranging from smooth to treacherous. The game paddles are used to control craft attitude and thrust. This is a real-time high res challenge!

FOREST FIRE! (Atari only) — **Price: $16.95 Cassette/$20.95 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.

NOMINOES JIGSAW (Atari, Apple and TRS-80 only) — **Price: $16.95 Cassette/$20.95 Diskette**
A jigsaw puzzle on your computer! Complete the puzzle by selecting your pieces from a table consisting of 60 different shapes. NOMINOES JIGSAW is a virtuoso programming effort. The graphics are superlative and the puzzle will challenge you with its three levels of difficulty. Scoring is based upon the number of guesses taken and by the difficulty of the board set-up. See review in ELECTRONIC GAMES.

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.

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: $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!

**ATARI, PET, TRS-80, NORTHSTAR, CP/M and IBM are registered tradenames and/or trademarks.*

***Except where noted, all model I software is available for the Model III. TRS-80 diskettes are not supplied with DOS or 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.

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.

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.

MOVING MAZE (Apple and Atari only) — **Price: $10.95 Cassette/$14.95 Diskette**
MOVING MAZE employs the games paddles to direct a puck from one side of a maze to the other. However, the maze is dynamically (and randomly) built and is continually being modified. The objective is to cross the maze without touching (or being hit by) a wall. Scoring is by an elapsed time indicator, and three levels of play are provided.

ALPHA FIGHTER (Atari only) — **Price: $14.95 Cassette/$18.95 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: $16.95 Cassette/$20.95 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: $16.95 Cassette/$20.95 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.

GIANT SLALOM (Atari only) — **Price: $14.95 Cassette/$18.95 Diskette**
This real-time action game is guaranteed addictive! Use the joystick to control your path through slalom courses consisting of both open and closed gates. Choose from different levels of difficulty, race against other players or simply take practice runs against the clock. GIANT SLALOM will run on 16K systems.

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.

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.

ADVENTURE

CRANSTON MANOR ADVENTURE (North Star and CP/M only) — **Price: $21.95 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.

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!

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!

Price: $329.95 (Please add $4.00 for shipping and handling)

TNT Software

The following DYNACOMP programs are available for use with TNT.

STUD POKER (Atari, 24K)
NOMINOES JIGSAW (Atari, 24K)
TEACHER'S PET I (Atari and North Star)
BRIDGE 2.0 (North Star)
CHOMPELO (Atari, 24K)

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.

Please specify 'TNT' versions when ordering.

ABOUT DYNACOMP

DYNACOMP is a leading distributor of small system software with sales spanning the world (currently in excess of 40 countries). During the past two 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 and A.N.A.L.O.G. 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.

BUSINESS and UTILITIES

SPELLGUARD™ (CP/M only) — **Price: $219.95 Disk**

SPELLGUARD is a revolutionary new product which increases the value of your current word processing system (WORDSTAR, MAGIC WAND, ELECTRIC PENCIL, TEXTED EDITOR II and others). Written entirely in assembly language, SPELLGUARD™ rapidly assists the user in eliminating spelling and typographical errors by comparing each word of the text against a dictionary (expandable) of over 20,000 of the most common English words. Words appearing in the text but not found in the dictionary are "flagged" for easy identification and correction. Most administrative staff familiar with word processing equipment will be able to use SPELLGUARD™ in only a few minutes.

MAIL LIST 2.2 (Apple, Atari and North Star diskette only) — **Price: $34.95**

This program is unmatched in its ability to store a maximum number of addresses on one diskette (minimum of 1100 per diskette, more than 2200 for "double density" systems!). Its many features include alphabetic and zip code sorting, label printing (1, 2, or 3 up), merging of files and a unique keyword seeking routine which retrieves entries by a virtually limitless selection of user defined codes. Mail List 2.2 will even find and delete duplicate entries. A very valuable program!

FORM LETTER SYSTEM rel. 2 (Atari, North Star and Apple Diskettes only) — **Price: $34.95**

FORM LETTER SYSTEM (FLS) is the ideal program for creating and editing form letters and address lists. It contains an easy-to-use text editor which produces fully justified text. Special codes are used in the address list to obtain personalized salutations. Form letters are produced by automatically inserting each address into a predetermined portion of your letter. FLS is completely compatible with MAIL LIST 2.2, which may be used to manage and sort your address files.

FLS and MAIL LIST 2.2 are available as a combined package for $59.95.

SORTIT (North Star only) — **Price: $29.95 Diskette**

SORTIT is a general purpose sorting program written in 8080 assembly language. This program will sort sequential data files generated by NORTH STAR BASIC. Primary and optional secondary keys may be numeric or one to nine character strings. SORTIT is easily used with files generated by DYNACOMP's MAIL LIST program and is very versatile in its capabilities for all other BASIC data file sorting.

PERSONAL FINANCE SYSTEM (Atari and North Star only) — **Price: $34.95 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.

FAMILY BUDGET (Apple 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.

DFILE (Atari and North Star diskettes only) — **Price: $19.95**

This handy program allows North Star and Atari disk users to maintain a specialized data base of all files and programs in the stack of disks which invariably accumulates. DFILE is easy to set up and use. It will organize your disks to provide efficient locating of the desired file or program.

FINDIT (North Star only) — **Price: $19.95**

This is a three-in-one program which maintains information accessible by keywords of three types: Personal (eg: last name), Commercial (eg: plumbers) and Reference (eg: magazine articles, record albums, etc). In addition to keyword searches, there are birthday, anniversary and appointment searches for the personal records and appointment searches for the commercial records. Reference records are accessed by a single keyword or by cross-referencing two or three keywords.

SHOPPING LIST (Atari and North Star 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: $59.95 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.

EDUCATION

HODGE PODGE (Apple only, 48K Applesoft or Integer BASIC) — **Price: $19.95 Cassette/$23.95 Diskette**

Let HODGE PODGE be your child's baby sitter. 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 9. HODGE PODGE is a non-intimidating teaching device which brings a new dimension to the use of computers in education.

TEACHER'S PET I (Available for all computers) — **Price: $11.95 Cassette/$15.95 Diskette**

This is the first of DYNACOMP's educational packages. Primarily intended for pre-school to grade 3, TEACHER'S PET provides the young student with counting practice, letter-word recognition and three levels of math skill exercises.

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.

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

DYNACOMP CASSETTES

DYNACOMP now offers high quality DYNACOMP brand name C-20 cassettes for computer use. Each cassette is guaranteed to be defect-free.

Box of 10 cassettes: $15.95 postpaid
Box of 20 cassettes: $29.95 postpaid

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 format) 8" CP/M floppy disks for systems running under MBASIC.

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 Hanning 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 and curve 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 and results 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 (Available for all computers) — **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): $23.95 + $1.50 postage
See reviews in KILOBAUD and Dr. Dobbs.

ROOTS (Available for all computers) — **Price $10.95 Cassette/$14.95 Diskette**

In a nutshell, ROOTS simultaneously determines all the zeroes of a polynomial having real coefficients. There is no limit on the degree of the polynomial, and because the procedure is iterative, the accuracy is generally very good. No initial guesses are required as input, and the calculated roots are substituted back into the polynomial and the residuals displayed.

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.

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.

Shipping and Handling Charges
Within North America: Add $1.50
Outside North America: Add 10% (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 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.

Editor's Note: Although some of the applications here are specific to PET computers, the analysis of the Boolean operators is valuable knowledge for any computerist. — RTM

Bits, Bytes And Basic Boole

David O. Williams
Toronto, Canada

Of all the operators which are included in Microsoft BASIC (or in most other implementations of the language) few cause as much confusion among novice programmers as the Boolean words OR, AND and NOT. What does the computer mean when you ask it to PRINT 3 OR 5 and it answers 7? How can 3 AND 5 equal 1? And what can be the possible use of something called NOT 3 which equals -4? In this article, I will describe the meanings and some of the uses of these operators.

I assume that most readers have some idea of what is meant by *binary numbers*. They are numbers which are written in a notation which has only two digits, zero and one. A single zero or a single one means exactly the same as in "regular" (i.e. decimal) numbers. However, to write the next number after one we have to start a new column, just as happens in decimal when we write the next number after nine. Thus, the first few whole numbers in both decimal and binary notations look as follows:

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

and so on.

By examining this little table you can see that each column of a binary number is related to a power of two, in the same way that the columns in decimal numbers are related to powers of ten. The right-hand digit represents 2^0 which equals one. The next column represents 2^1 which equals 2. The third column represents 2^2 or 4, and so on. Thus, 110 in binary represents 0 times 2^0, plus 1 times 2^1, plus 1 times 2^2, which, in decimal notation, all adds up to six. Take a minute right now to check that all the other numbers in the table can be expressed as sums of powers of two in the same way, and that these powers are reflected in the pattern of 1's in the binary representation of the number.

There is a trivial point about all numbers, binary or decimal, which we ought to get clear before going on to look at the Boolean operators. The number 000101 is exactly the same as the number 101, in either notation. The zeroes before the first non-zero digit (called leading zeroes) are often not written, but in a sense they exist anyway. They represent the fact that the number in question includes zero times high powers of two (or ten). In discussing Boolean operations it is helpful to have the same number of digits in all the numbers in any example. We can do this by writing in some leading zeroes where appropriate.

The OR

Now let's look at the example of the Boolean OR operator mentioned in the first paragraph. 3 OR 5 equals 7. In binary form, with the numbers written beneath each other, it looks like this:

```
   011
OR 101
   111
```

Here are a couple of other examples, each with numbers of four bits (that's just short for *B*inary dig*ITS*). Before reading on, see if you can spot the pattern and work out what the OR operator does. Hint: look at the ones in the numbers, and think of the English word OR.

```
   1011      0101
OR 1001   OR 0011
   1011      0111
```

If you still haven't seen the pattern, look at each column in each example. There are no "carries" in Boolean operations, so the columns do not affect each other at all.

I hope you discovered that the answer to an OR operation has a one in each position where one OR the other (OR both) of the starting numbers had a one. That is all the OR operator does. It combines two binary numbers to give a third according to this rule. Perhaps you also noticed that the third example I gave you was the same as the first, except that the two starting numbers were in reversed order. Of course this had no effect on the result.

The AND

Now you should have no difficulty in deciphering the AND operation.

```
    101          10010110
AND 011      AND 01010101
    001          00010100
```

The answer in each case has a one *if and only if* the first AND the second starting number had a one in the corresponding position.

As an exercise, I suggest you try thinking of some pairs of numbers (restrict yourself to fairly small positive integers) and predict what the results

COMPETENCY EXAM PREPARATION SERIES

This comprehensive set of programs consists of simulated exam modules, a thorough diagnostic package, and a complete set of instructional programs. It is designed to teach concepts and operations, provide drill and practice and assess achievement levels through pre and post testing. The Competency Exam Preparation Series provides a structured, sequential, curriculum encompassing mathematical, reading and writing instruction.

The C.E.P.S. program is designed for individual student use or use in a classroom setting. Programs provide optional printer capability covering worksheet generation and performance monitoring. C.E.P.S. are available in three software formats.

National Proficiency Series $1,299.00
N.Y.S. Regents Competency Test, Preparation Series $1,299.00
California Proficiency Assessment Test, Preparation Series $1,299.00

If desired separate Mathematics and Verbal packages are available for $799.00 ea. A Spanish language version of the Mathematics Instruction Package is available at no extra charge.

COLLEGE BOARD PREPARATION SERIES 81/82

for TRS-80 NORTHSTAR ™
PET, APPLE, OSI, ATARI

Each program confronts the user with a virtually limitless series of questions and answers. Each is based on past exams and presents material of the same level of difficulty and in the same form used in the S.A.T. Scoring is provided in accordance with the formula used by College Boards.

S.A.T., P.S.A.T., N.M.S.Q.T. - Educator Edition set includes 25 programs covering Vocabulary, Word Relationships, Reading Comprehension, Sentence Completion, and Mathematics. Price $229.95

Independent Tests of S.A.T. series performance show a mean total increase of 70 points in students' scores.

G.R.E. Series - Educator Edition includes 28 programs covering Vocabulary, Word Relationships, Reading Comprehension, Sentence Completion, Mathematics, Analytical Reasoning and Logical Diagrams. Price $289.95

ODYSSEY IN TIME

This spectacular adventure game adds a new dimension of excitement and complexity to **Time Traveler.** Players must now compete with the powerful and treacherous adversary in their exacting quest for victory.

To succeed they must vanquish this adversary in combat that rages across 24 time periods.

Odyssey In Time includes all the challenges of Time Traveler plus 10 additional eras, including those of Alexander the Great, Emperor Asoka of India, Attila the Hun, Genghis Khan. Each game is unique, and may be **interupted and saved** for later play.

available for APPLE & TR-80 PET, 32K - $39.95

ISAAC NEWTON

Perhaps the most fascinating and valuable educational game ever devised — ISAAC NEWTON challenges the players to assemble evidence and discern the underlying "Laws of Nature" that have produced this evidence. ISAAC NEWTON is an inductive game that allows players to intervene actively by proposing experiments to determine if new data conform to the "Laws of Nature" in question. Players may set the level of difficulty from simple to fiendishly complex.

In a classroom setting the instructor may elect to choose "Laws of Nature" in accordance with the complete instruction manual provided.

For insight into some of the basic principles underlying **ISAAC NEWTON** see GODEL, ESCHER, BACH by Douglas R. Hofstadler, Chapter XIX and Martin Gardner's MATHEMATICAL GAMES column in **Scientific American,** October, 1977 and June, 1959. $24.95.

TIME TRAVELER

Confronts players with complex decision situations and the demand for real time action. Using the **Time Machine**, players must face a challenging series of environments that include: The Athens of Pericles, Imperial Rome, Nebuchadnezzar's Babylon, Ikhnaton's Egypt, Jerusalem at the time of the crucifixion, The Crusades, Machiavelli's Italy, The French Revolution, The American Revolution, and The English Civil War. Deal with Hitler's Third Reich, Vikings, etc. At the start of each game players may choose a level of difficulty . . . the more difficult, the greater the time pressure. To succeed you must build alliances and struggle with the ruling powers. Each game is unique.

$24.95

Send $2.00 for complete Catalogue.

$5.00 Discount Coupon included in Catalogue.

PROGRAMS AVAILABLE FOR
TRS-80, APPLE II & PET
(unless otherwise indicated)
☐ disk or ☐ cassette (please specify)

Krell Software Corp.

Send check or money order to
21 Milbrook Drive, Stony Brook, NY 11790
(516) 751-5139

All programs require 16K TRS-80 programs require LEVEL II BASIC APPLE programs require APPLESOFT BASIC

NY State Residents Add Sales Tax

would be of ANDing and ORing them. Then test your predictions with your computer. Remember that the numbers will have to be in decimal notation for the BASIC interpreter in the machine to understand them. At a "deeper" level, however, the numbers are actually processed in binary form.

Perhaps the commonest use of Boolean operators is in what might be called "gentle POKEing".

The NOT

The NOT operation is even simpler, in binary notation, than the ones we have already looked at. There is only one starting number. Here are a couple of examples:

NOT <u>10010110</u>   NOT <u>01101001</u>
01101001   10010110

The digits in the answer are simply the opposites of the ones in the starting number. Each one becomes a zero, and each zero a one. Of course, doing this operation twice gets you back to your starting number. If you look closely at the two examples you'll see that they do just that.

A PET Complication

There is a complication, however, which has to do with the way the PET handles negative numbers. This is done by having an extra bit attached to a number, and this is either a zero or a one according to whether the number is positive or negative. When you NOT a number ALL the bits are reversed, including the sign bit. So the NOT of a positive number is negative, and vice versa. Zero is counted as a positive number, and its NOT is -1. In fact the following simple rule always applies:

NOT(X) = -(X+1)

Remember this rule as a curiosity, if you like. For practical purposes, it is more important to remember the fundamental fact that NOT reverses all the bits. Ones become zeroes and zeroes ones.

Only one more point needs to be considered before we can say we understand the Boolean operators, at least in theory. How are fractions handled? The answer is: they're not. If you ask the PET to PRINT 2.5 OR 5.1 the answer 7 will be returned. The fractional parts of the input numbers were simply dropped, just as if you had used the INT( ) operator. The PET does all the Boolean operations on 16-bit integers, including the sign bit. Fractions can't be fitted into this format, so they are dropped. Also, very large numbers, which would need more bits, cannot be used in Boolean operations. They lead to ILLEGAL QUANTITY ERRORs. In practice, this is not a serious restriction.

Programming With Booleans

By now, I hope, you feel you understand what the Boolean operators do, but you are probably still wondering whether they ever turn out to be useful in realistic programming situations. Indeed they do and, to illustrate this fact, we'll use the rest of this article to explore several uses.

The first of these, which is so simple as to be almost trivial, is a test which distinguishes between odd and even numbers. Every now and again (we'll see an actual example later) it is useful to make such a test on a number which crops up in the middle of some calculation. Of course there are many ways to do it, but none simpler than:

X=N AND 1

If N is odd, X will equal one. If N is even, X will be zero. The reason this works is simple if you think of binary numbers. The right-hand digit of an odd number is always one and, of an even number, is always zero. ANDing the number with 1 has the effect of looking only at this digit (a technique which is aptly known as *masking*), so the answer is zero or one, according to this digit.

Gentle POKEing

Perhaps the commonest use of Boolean operators is in what might be called "gentle POKEing". The POKE command, as I expect you know, forces a byte of memory (that's a set of eight bits) to be loaded with a number which the command contains. Thus the command:

POKE 59468,14

causes the byte at memory location number 59468 to be loaded with the bits 00001110. (That's fourteen in binary notation.) This command allows for no flexibility. Every bit of the number is exactly specified.

But sometimes we may not want to POKE all eight bits in the address. We may want one particular bit to be forced to become a one, but all the other bits should be left at whatever values they have already. For example, suppose we want the letter in the top left-hand corner of the PET video screen to be displayed in reverse field (black on white), but we do not want to change the actual letter from whatever it is already. The command:

POKE 32768,PEEK(32768) OR 128

does the trick. (Try it!). The command forces the eighth bit in the address ($128 = 2^7$) to become a one (remember the OR operator), but leaves all the other bits alone, so the letter is not changed. The eighth bit is the one which signifies reverse field, so making it a one has the effect we want. Of course,

now we have this character in reverse field, we may want to change it back again. We want to force the eighth bit to become a zero.

POKE 32768,PEEK(32768) AND NOT 128

is the command we want. This uses two Boolean operators. First the number NOT 128 is calculated. It has a one in every position *except* for the eighth bit, which is zero. Then this number is ANDed with the previous contents of the address. This forces a zero into the eighth bit, and leaves the rest unchanged. As an exercise, try predicting the effect of the following command, then try it several times in succession on your PET.

N=32768 : P=PEEK(N) : POKE N,((P OR 128) AND NOT (P AND 128))

Manipulating the reverse-field bit in the video display makes a good demonstration of "gentle POKEing," and it is sometimes useful in programs. However, the technique is more often used in connection with some rather complex trickery to make the computer do these things which it ordinarily could not do. There are some addresses in a computer's memory which contain bits which act as internal *flags* which store information about the machine's status. Often several flags which refer to very different aspects of the computer's condition are grouped together in a single byte.

For example, for the PET, there is a byte which contains a flag which shows whether the Play key on the tape deck has been pressed, but the same byte also contains bits which select a particular row of keys on the main keyboard. Normally all these flags are looked after by the PET's internal routines, and the programmer does not have to think about them. However, advanced programmers sometimes find reasons to change the contents of a flag. This means that the flag is made to contain the "wrong" information, and the PET starts to behave in abnormal ways because of this. Sometimes this abnormal behaviour can be put to good use. Anyway, whatever its purpose, changing a single flag bit in a byte which also contains other flags which are to be left unchanged is a common use of "gentle POKEing."

A very different use of Boolean operations is to translate between the ASCII numbers of characters and the numbers which are used to represent these same characters in the PET's video memory. To demonstrate this, first try keying in and running the following little program:

```
10 GET G$ : IF G$="" THEN 10
20 POKE 32768,ASC(G$)
30 GOTO 10
```

When a program is running, if you press a key on the keyboard a character will appear in the top left-hand corner of the screen. However, this character will often not be the same as the one on the key you pressed. The program is poking the ASCII numbers into video memory, but these are often the wrong numbers. The video numbers are related to the ASCII numbers, but they are often not the same. The following little Boolean formula translates the ASCII numbers into the correct POKE numbers:

P=(A AND 63)+(128 AND A)/2

If you think about it, (remember that dividing by two shifts each bit in a binary number one place to the right) you will find that this formula has the effect of dropping the seventh bit of the number and of moving the eighth bit into the seventh position. This is exactly the way in which the PET changes ASCII before putting numbers into the video memory. You can incorporate this into the little program as follows:

```
10 GET G$ : IF G$="" THEN 10
20 A=ASC(G$)
30 P=(A AND 63)+(128 AND A)/2
40 POKE 32768,P
50 GOTO 10
```

When you run this program you will find that you always get the same character appearing in the top left-hand corner of the screen as the one on the key you pressed. This trick is sometimes useful in programming. It allows a message to be POKEd onto the screen without using the PRINT command, and thereby disturbing the position of the cursor.

The opposite trick, of translating screen characters into ASCII, can also be useful. For example, you might want to copy whatever is on the screen onto a printer. The printer wants ASCII, but the message, or whatever, is in screen characters. The translation formula has to generate the appropriate seventh bit of the ASCII number, which is not present in the screen representation. The following formula correctly translates every character except one:

A=(63 AND P)+2*(64 AND P)+2*((NOT P) AND 32)

The last part of the formula is the one in which the seventh bit is generated. The one character which it does not translate correctly is π . This character is a Commodore addition to the standard ASCII set, and is handled by special routines in the PET. If you encounter this character, you will also have to write special routines to translate it.

Fine Plotting

My last example of Boolean operations will be the process of "fine plotting" on the 40-column PET. (I have never tried this on the 80-column machine, but I assume that something similar can be done.) On the 40-column PET there are 1000 character positions on the screen, arranged in 40 columns and 25 rows. Suppose you want to draw (or have the computer draw) an abstract shape, such as a mathematical graph, on the screen. A simple way to do this is to set up a pattern in which a single

character, such as a reverse-field blank, is printed or POKEd into some of the 1000 positions. The overall pattern of these characters on the screen is arranged to have the appearance of the wanted shape. However, the small number of columns and rows gives a crude graph, and for this reason this technique is known as "coarse plotting."

A great improvement can be made by "fine plotting," which effectively increases the number of columns to 80 and the number of rows to 50. This is done by constructing the shape out of some of the PET's graphic characters, such as those on the comma, semicolon and question-mark keys. In each of these characters, the full-size character position (which I shall call the "frame") can be imagined as being divided vertically and horizontally into four quarters or "squares." In each character, some of the squares in the frame are white and the others black, where the word "some" means any number from zero to four. There are a total of sixteen characters which, together, cover all the possible combinations of white and black square in a frame. There are thus 4000 positions for squares on the screen and, by appropriately choosing and combining the 16 characters, any pattern of white and black squares can be generated. The process of fine plotting generates the pattern which most closely represents a wanted shape or graph.

A moment's thought shows that there are some interesting problems involved in doing this. Suppose you want to make a particular square white. You cannot simply calculate an address in the video memory and POKE it to some fixed number. Each address in the memory corresponds to a frame, and hence to four squares. The number to which the address should be POKEd must therefore take into account all four squares. Making one square white must not accidentally make any of the other squares change from white to black or vice versa.

The important clue in recognizing an elegant way of making the required calculations is the similarity between the requirements of fine plotting and those of "gentle POKEing." In each case there is a combination of items (squares in a frame, or bits in a byte) and we want to alter just one item while leaving all the others undisturbed. We saw earlier that the Boolean OR instruction can be used to force a single bit in a byte to become a one, and that AND NOT can be used to make a bit a zero. Perhaps, by putting the 16 graphic shapes into one-to-one correspondence with the 16 possible four-bit binary numbers, so that each bit represents the status (white or black) of a particular square, and by using Boolean operators on the numbers, we can work out a way of modifying the status of one square in a frame without disturbing the others.

Of course it can be done, but you may be surprised by how easily. Here are the necessary routines:

```
  10 GOTO 1000
 100 REM PLOT/UNPLOT ROUTINE
 110 N=N1+40*INT(R/2)+INT(C/2)
 120 M=(C AND 1)+2*(R AND 1)
 130 POKE N,SC(F,SB(PEEK(N)),M)
 140 RETURN
1000 REM INITIALIZATION
1010 DIM SA(15), SB(255), SC(1,15,3)
1020 FOR A1=0 TO 15
1030 READ SA(A1)
1040 SB(SA(A1))=A1
1050 NEXT A1
1060 FOR A$=0 TO 3
1070 A2=
1080 FOR A3=0 TO 15
1090 SC(1,A3,A1)=SA(A3 OR A2)
1100 SC(1,A3,A1)=SA(A3 AND NOT A2)
1110 NEXT A3,A1
1120 N1=32768
1130 DATA 32,126,124,226,123,97,255,236,108,127,
     225,251,98,252,254,160
```

When you RUN the above, it will initialize the arrays, but will do nothing visible on the screen. When it has finished, try entering:

```
F=0 : C=40 : R=25 : GOSUB 100
```

You should see a small white square appear in the middle of the screen. The value of F is a flag. If F=0 the routine will plot a point, i.e. make a square appear white. If F=1 the point will be "unplotted," i.e. made dark. The values of C and R are the coordinates of the point. C is the column number, which goes from zero at the left of the screen to 79 at the right. R is the row number, which is zero at the top of the screen and 49 at the bottom.

Try entering:

```
F=0 : FOR R=0 TO 49 : C=R : GOSUB 100 : NEXT
```

A diagonal line of little squares will be drawn across the screen. Enter the same thing but with F=1 (and without allowing the screen to scroll!) and the line will be neatly erased.

Write a program including these routines, and you will be able neatly and quickly to fine-plot any shape you want.

But how does it work? You work it out! There are lots of things in the routines which you ought to recognize from earlier in this article. In line 120 there are tests for odd and even numbers. In lines 1090 and 1100 there are the OR and AND NOT operators which we know are used to force bits to become one or zero. From line 1070, A2 is clearly a power of two. A little experimentation will show you that the numbers in the DATA line are the screen memory numbers corresponding to the sixteen characters which are used. Look at the order of these characters in the DATA and see how they are READ in lines 1020 to 1059. You will find that the shapes of the characters (i.e. the patterns of light and dark squares in the frame) are related to the bit-patterns in the corresponding (binary) values of A1. ©

Model EP-2A-79

EPROM Programmer

North Star · Apple · S-100 · SS-50 · STD-Bus · Atari · Pet · Kim-1

TRS-80 · H-8 · H-89 · Ohio · Scientific · SWTP · Aim-65 · Sym-1

Three years in the field with unsurpassed performance. Software is available for the EP-2A-79 for most all of the microcomputers including the popular CP/M, FLEX, HDOS operating systems. Write or call for specific hardware/software interfacing. Driver packages available for F-8, 6800, 6809, 8080,8085, Z-80, 1802, 6502 and 2650 based systems.

EP-2A-79 115V 50/60 HZ $169.00

Personality Modules

PM-0	TMS 2708	$17.00	PM-5	2716, 2758	$17.00
PM-1	2704, 2708	17.00	PM-5E	2816	35.00
PM-2	2732	33.00	PM-8	MCM68764	35.00
PM-2A	2732A	33.00	PM-9	2764	35.00
PM-3	TMS 2716	17.00	SA-64-2	TMS 2564	39.00
PM-4	TMS 2532	33.00	SA-64-3	2764	39.00

Optimal Technology, Inc.

Phone (804) 973-5482
Blue Wood 127 Earlysville, VA 22936

Micro Co-op

Micro Co-op is a software co-operative. We publish a bi-monthly newsletter describing various new and useful software products as they become available. We also try to give reasonable comparisons of existing products that help you make more educated buying decisions. Our members may purchase most software through us below retail price.

For more information, write for a free copy of our newsletter.

Micro Co-op
(Specify Apple or Atari) Newsletter
P.O. Box 432
West Chicago, IL 60185
(312) 231-0912

MEMOREX
FLEXIBLE DISCS

WE WILL NOT BE UNDERSOLD!! Call Free **(800)235-4137** for prices and information. Dealer inquiries invited and C.O.D.'s accepted.

PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. In Cal. call
(800)529-5935 or
(805)543-1037

VISA

DISK DRIVE WOES?
PRINTER INTERACTION?
MEMORY LOSS?
ERRATIC OPERATION?

Don't Blame The Software!

Power Line Spikes, Surges & Hash could be the culprit! Pat. #4,259,705
Floppies, printers, memory & processor often interact! Our patented ISOLATORS eliminate equipment interaction AND curb damaging Power Line Spikes, Surges and Hash.

- **ISOLATOR (ISO-1) 3 filter isolated 3-prong sockets; integral Surge/Spike Suppression; 1875 W Maximum load, 1 KW load any socket $62.95**
- **ISOLATOR (ISO-2) 2 filter isolated 3-prong socket banks; (6 sockets total); integral Spike/Surge Suppression; 1875 W Max load, 1 KW either bank $62.95**
- **SUPER ISOLATOR (ISO-3), similar to ISO-1 except double filtering & Suppression $94.95**
- **ISOLATOR (ISO-4), similar to ISO-1 except unit has 6 individually filtered sockets $106.95**
- **SUPER ISOLATOR (ISO-11) similar to ISO-2 except double filtering & Suppression $94.95**
- **CIRCUIT BREAKER, any model (add-CB) .... Add $ 8.00**
- **CKT BRKR/SWITCH/PILOT (-CBS) .... Add $16.00**

AT YOUR DEALERS

Master-Card, Visa, American Express
Order Toll Free 1-800-225-4876
(except AK, HI, PR & Canada)

ESP Electronic Specialists, Inc.
171 South Main Street, Natick, Mass. 01760
Technical & Non-800: 1-617-655-1532

ATTENTION! AIM 65 USERS

AND OTHER 6500 MICRO PROCESSORS

Convert Your 65 Into A More Useful Tool With The New And Reasonable

VD640 VIDEO INTERFACE

Save Expensive Thermol Paper • Save Time • Save Money

FEATURES

- Uses 2K 2716 or 1K 2758 E Prom
- 1K Ram
- Pin compatible with AIM expansion connector. Can be pin adapted to fit Kim & Sim.
- Single +5 volt from connector.
- Reverse video & switch to reverse field.
- Upper & lower case characters.
- Uses standard T.V. with optional modulator.
- 7x11 Dot Format
- 40 characters X 16 lines.
- 32 special graphics characters.
- Software switch for programmable hardware screen flashing.
- Uses Motorola MC6845 CRTC
- Operating system in Eprom interacts with AIM monitor.
- Component layout silkscreened on P.C. Board.

Full operating instructions and schematic with each kit.

- [ ] VD640 Silkscreened P.C. board w parts list only ..... $ 39.00 each
- [ ] VD640 Kit-inc's. Silkscreened P.C. board with sockets, all components and Programmed Eprom. (P.C. Connector Board & mod. opt.) .... $ 119.00 each
- [ ] VD640 Fully assembled with sockets all components and connectors, tested and guaranteed 1 year (modulator optional) .. 149.00 each
- [ ] Optional R.F. Modulator (allows use of std. T.V.) $39.00
- [ ] Optional P.C. Connector Board 15.00

Insert quantity wanted in box next to kit desired and mail with check or money order to: Sierra Pacific; 1112 Wellington Dr.; Modesto, CA 95350. (Calif Residents add 6% sales tax.) (Allow 4 to 6 weeks delivery.)

The Practical Side Of Assembly Language

Part II: Loops And Arrays

Bruce D. Carbrey, Raleigh, NC

In the last installment (**COMPUTE!** #14) I presented some ideas on how to represent and use flags in 6502 assembly language programs. This time I will discuss methods for programming loop control structures for the manipulation of arrays of data. Let's start by writing a loop which simply initializes all elements of an array to zero. In BASIC, you might write:

```
100 DIM AR(50)
110 FOR I=1 TO 50
120 AR(I)=0
130 NEXT I
```

If you are a neophyte assembly language programmer and try to translate this program segment on a line-by-line basis, you might wind up with something like this:

```
AR     *=*+  50      ;SPACE FOR ARRAY AR (50 BYTES)
I      *=*+  1       ;LOOP COUNTER VARIABLE I
       ...
       LDA   #1
       STA   I       ;INITIALIZE LOOP COUNTER
LOOP   LDA   #0
       LDX   I       ;RECALL CURRENT LOOP
                      COUNTER
       STA   AR,X    ;SET ELEMENT OF ARRAY TO 0
       INX           ;ADVANCE TO NEXT ELEMENT
       STX   I       ;STORE INDEX REGISTER
       CPX   #50     ;CHECK AGAINST LIMIT
       BNE   LOOP    ;REPEAT UNTIL DONE
```

If you run this program you'll be dismayed to find out that it only sets the last 49 elements of the array to 0 and skips the first element, because the first element of the array should be indexed with a zero, not a one.

Rule #7. To access the first element of an assembly language array, you should use an index of 0, not 1. The last element of an array of size N is indexed by N–1.

You may also recognize that it is not necessary to allocate space or save the variable I for the loop. Since it is only needed to control the loop, it can be simply kept in the index register (I chose the X register; the Y register will serve equally well). We can correct and improve the loop as follows:

```
AR     *=*+  50      ;SPACE FOR ARRAY AR (50 BYTES
       ...
       LDA   #0      ;CONSTANT TO FILL ARRAY
                      WITH
       TAX           ;X=0=INITIAL INDEX TO
                      ARRAY AR
LOOP   STA   AR,X    ;ZERO OUT ONE ARRAY
                      ELEMENT
       INX           ;ADVANCE TO NEXT ELEMENT
       CPX   #50     ;CHECK INDEX AGAINST LIMIT
       BNE   LOOP    ;REPEAT UNTIL DONE
```

You may notice some other subtle improvements in this program segment. The A register is only loaded with 0 once, outside the loop, since it does not change inside the loop. This will make the program run faster by eliminating 49 unneeded repetitions of the LDA #0 instruction.

Rule #8. Move code which does not need to be repeated out of the loop if possible.

Also note that one byte of code was saved by using TAX to initialize the X register instead of LDX #0. Naturally if we were filling our array with something other than 0, this trick won't work. We now have a correctly-functioning loop which is equivalent of our BASIC program (strictly speaking, is not exactly equivalent because the BASIC interpreter uses floating point arithmetic which uses four or five bytes for each array element instead of one).

Can our loop be further improved in terms of efficiency? Consider this alternative:

```
AR     *=*+  50      ;SPACE FOR ARRAY (50 BYTES
       ...
       LDA   #0      ;CONSTANT TO FILL THE ARRAY
                      WITH
       LDX   #49     ;INDEX TO LAST ELEMENT OF
                      THE ARRAY
LOOP   STA   AR,X    ;SET AN ELEMENT OF THE
                      ARRAY TO 0
       DEX           ;BACKUP TO PREVIOUS
                      ELEMENT
       BPL   LOOP    ;REPEAT UNTIL DONE
```

This code segment fills the loop backwards, filling the last element first and the first element last. Once the 0th element has been filled, the index register is decremented to -1 ($FF) and the BPL LOOP instruction will exit the loop. Notice that we have eliminated the CMP instruction from the loop, saving two cycles.

Rule #9. Moving backwards through an array will usually be more efficient.

If you try to make the array bigger than 128 elements you will be in trouble! Suppose you increase the dimension of AR to 200. In this case your loop will be executed only once because on the first pass, the DEX instruction will change the index from 199 to 198. But 198 has the hexadecimal representation $C6, which has bit 7 (the sign bit) set to 1. Therefore the 6502 will consider this a negative number (-58 decimal) and the BPL instruction will let control "fall through." Therefore, our BPL instruction will only work right up to +127 decimal, which is the largest *signed* 8-bit number. We can remedy this problem for index values up to 255 with a slightly more "tricky," but equivalent, method:

```
AR     *=*+  200     ;SPACE FOR ARRAY (200 BYTES)
       ...
       LDA   #0      ;CONSTANT TO FILL ARRAY
                      WITH
       LDX   #200    ;INDEX TO LAST ELEMENT OF
                      THE ARRAY+1
```

NEW!

IBM COMPATIBLE 8" FLOPPY DISK for CBM/PET,* Aim, Sym

PEDISK II

LOW COST – HIGH PERFORMANCE FLOPPY DISK

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:

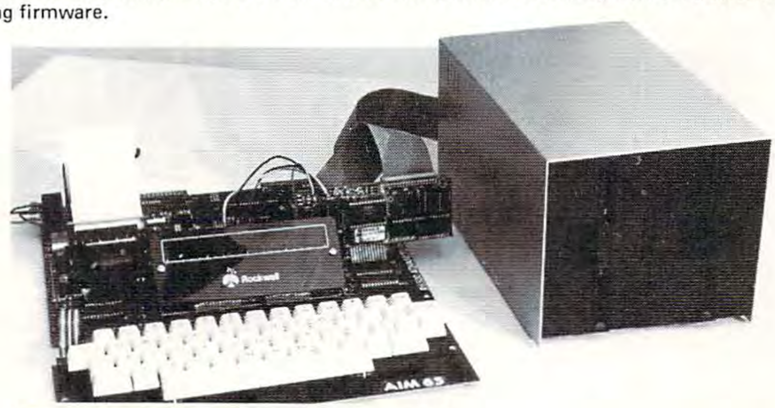
Model 877-1 Single Drive, IBM standard (295K)	$1,095.00
Model 877-2 Dual Drive, IBM standard (590K)	$1,695.00

Payable in U.S. Dollars

PEDISK II is a high performance mass storage peripheral to enhance your computer's storage capability. Total storage to 850K bytes is available. The PEDISK II system consists of a small disk controller electronic board that mounts inside the computer and an external disk drive assembly. PEDISK II offers the fastest mass storage system available for the Commodore PET. With a data transfer rate of 250000K bits per second, the PEDISK II loads data directly to memory. This is up to four times faster than any 488 bus-type mass storage device.

One, two, or three drives connect to the PEDISK controller board. The user can choose 5¼" or 8" disk drives. An important feature of the PEDISK II system is its ability to exchange data with other computers. The standard 8" IBM 3740 format allows data exchange with large computers. Data files, mail lists, etc. available on large computers can be processed by the PET/PEDISK system. Remote PET/PEDISK computers can provide data entry for large machines.

PDOS II software links directly to the standard BASIC and operates with BASIC-type commands for easy interfacing. A full set of disk utility commands completes the powerful disk operating firmware.



DISK COMMANDS:

BASIC

- !LOAD – reads a program file to the computer
- !SAVE – stores a BASIC program file to the disk
- !OPEN – forms a sequential or relative data file
- !INPUT – reads a data record from a file on the disk
- !PRINT – stores a data record to a file on the disk
- !CLOSE – ends a sequential or relative data file
- !LIST – displays a directory of all files on the disk
- !RUN – reads a program file and executes

MONITOR-DOS

- D – displays contents of memory or diskette.
- G – go to program and execute.
- H – help user with listing of all commands.
- K – kill a file on the diskette (erase file).
- L – read program to the computer memory.
- M – memory examine and change monitor.
- N – name a file differently (rename).
- P – print directory of all files on the disk.
- R – return to BASIC mode.
- S – save program or data from memory to the disk.
- U – utility: format, copy, compress, patch diskette.
- X – execute program after loading.

fullFORTH +

FULL FEATURE "FORTH" FOR 6502 SYSTEMS

INTERPRETER – can be executed directly in an interpretive mode to speed testing and debugging.

CROSS-COMPILER – words can be individually compiled and tested, the entire program can also be cross-compiled for maximum efficiency.

COND. ASSEMBLER – Machine language modules can be intermixed and conditionally assembled to fullFORTH.

STRING HANDLING – variable length constants and variables are allowed. Processes compare, move, concatenate and sub-string words.

FLOATING POINT – process 5 or 9 digit integer and floating point numbers for arithmetic operations.

SCREEN EDITOR – contains a unique full cursor visible screen editor.

SPECIFY PEDISK II, PET 2040 or 4040 DISK, OR APPLE $75.00

FOR INFORMATION, SEE YOUR DEALER OR:

cgrs MICROTECH

P.O. BOX 102 • LANGHORNE, PA 19047 • (215) 757-0284
U.S.A.

DEALER INQUIRIES INVITED

*PET IS A TRADEMARK OF COMMODORE

```
LOOP   STA   AR-1,X     ;SET AN ELEMENT OF THE
                         ARRAY TO 0
       DEX              ;BACKUP TO PREVIOUS
                         ELEMENT
       BNE   LOOP       ;REPEAT UNTIL DONE
```

Here we have replaced the BPL instruction with a BNE instruction so that the loop will terminate one pass earlier, but will not be stymied by index values greater than 127. Since our last pass through the array will now have an index of 1 instead of 0, we must compensate by changing the destination for our indexed STA instruction to AR-1. Therefore the last element set will be AR-1 + 1 = AR + 0. Finally, the starting index must be bumped from 199 to 200 for the same reason. Note that a starting index of 0 will clear a full 256 byte array.

The same technique can be used to move a block of data of up to 256 bytes from one known location to another:

```
ARA    *=*+  200        ;ARRAY A CONTAINING 200
                         ELEMENTS
ARB    *=*+  200        ;ARRAY B CONTAINING 200
                         ELEMENTS
       ...
       LDX   #200       ;NUMBER OF ELEMENTS TO
                         MOVE
LOOP   LDA   ARA-1,X    ;FETCH ELEMENT OF ARRAY A
       STA   ARB-1,X    ;INSTALL IN ARRAY B
       DEX              ;DECREMENT TO PREVIOUS
                         ELEMENT
       BNE   LOOP       ;REPEAT UNTIL DONE
       ...
```

What happens if you have more than 256 bytes? Throw away your 6502 and get a processor with a 16-bit index register? Nope. The indirect,X and indirect,Y addressing modes will solve this problem.

Rule #10. To use arrays of more than 256 bytes or arrays whose location is not known at assembly time, plan on using indirect,X or indirect,Y addressing.

Unlike the absolute, indexed addressing modes, indirect,X and indirect,Y are not equivalent. You may remember that indirect,X addressing uses pre-indexing and indirect,Y uses post-indexing. As a practical matter, indirect,X addressing will almost always be used with a permanent index of 0, simulating simple indirect addressing. This mode lends itself to manipulating large data arrays in non-time-critical portions of a program. For example, the following loop initializes a 1000-element array to 0:

```
ARRAY  *=*+  1000              ;ROOM ARRAY OF 1000
                                ELEMENTS
PTR    *=*+  2                 ;POINTER TO AN ARRAY
                                ELEMENT
       ...
CLR1K  LDA   #ARRAY&$FF        ;LOW 8 BITS OF ADDRESS
                                OF START OF ARRAY
       STA   PTR               ;INITIALIZE POINTER
       LDA   #ARRAY/256        ;HIGH 8 BITS OF AD-
                                DRESS OF START OF
                                ARRAY
       STA   PTR+1             ;INITIALIZE HIGH BYTE
                                OF POINTER
       LDX   #0                ;PERMANENTLY LOAD
                                X WITH 0
LOOP   LDA   #0
       STA   (PTR),X           ;ZERO BYTE POINTED
                                TO BY PTR
       INC   PTR               ;BUMP POINTER UP TO
                                NEXT ELEMENT
       BNE   CHECK             ;BRANCH IF NOT
                                CROSSING PAGE
                                BOUNDARY
       INC   PTR+1             ;ELSE BUMP HI-ORDER
                                BYTE OF POINTER
CHECK  LDA   PTR
       CMP   #ARRAY+1000&$FF   ;CHECK POINTER
                                AGAINST LIMIT
       BNE   LOOP              ;REPEAT IF NOT DONE
       LDA   PTR+1
       CMP   #ARRAY+1000/256   ;ELSE CHECK HI BYTE
                                OF POINTER
       BNE   LOOP              ;REPEAT IF NOT DONE
```

Some assemblers use the notation #‹ARRAY to mean the low byte of the address of ARRAY and #›ARRAY for the high byte instead of #ARRAY&$FF and #ARRAY/256. Clearly this program segment is quite a bit "messier" than the one for arrays of less than 256 bytes. When planning sizes for arrays, you should remember this and try to limit arrays to 256 bytes or less whenever practical.

Luckily, the indirect,Y addressing mode is considerably more powerful than indirect,X. For our final problem, let's use the indirect,Y mode to build a subroutine to move a large block of data from one place to another in memory as fast as possible. The source address, destination address, and number of bytes to be moved are to be specified as input to the routine as three 16-bit variables in page 0:

```
FROM   *=*+  2   ;POINTER TO STARTING ADDRESS OF
                  ARRAY TO MOVE
TO     *=*+  2   ;POINTER TO STARTING ADDRESS
                  OF DESTINATION
COUNT  *=*+  2   ;NUMBER OF BYTES TO COPY
```

In an earlier example we saved execution time by removing the need for a compare inside the loop. We can apply the same principle to speed up our block move by sub-dividing the routine into two loops, one which moves entire pages (1 page = 256 bytes), and one which moves the final fractional page. This allows us to avoid any compares in the part which moves entire pages(which is part of the routine executed the most when copying large blocks). This will also let us use both 8-bit index registers to maximum effectiveness by allocating one for counting pages and index registers to maximum effectiveness by allocating one for counting pages and the other for indexing bytes within the page. The resulting routine (shown in Program 4) can easily be converted into a block-fill routine instead by removing FROM and all lines that refer to it, and presetting A to 0 (or the value with which to fill the array).

Rule #11. To deal with large arrays, split your program into two loops, one to operate on entire pages and one to operate on the "leftover"

fractional page.

The routine in Program 4 moves data at about 16.1 machine cycles per byte for large blocks, which means a 16K byte array can be moved in 0.26 seconds using a 6502 with a 1MHz clock. In certain applications where speed is of paramount importance, you may wish to improve even this super-fast copy routine. Can it be done? Yes, if you are willing to trade some increased program size for increased execution speed. Again, we employ the same gen-

Program 2: Keyboard Driver with Alpha Lock Flag Using 0 = False and Non-0 = True

Editor's Note: Part of this program was not printed in **COMPUTE!** *#14. we reprint it entirely here. — RTM*

```
                  ;
                  ;       SUBROUTINE INCH: KEYBOARD DRIVER FOR ASCII-ENCODED
                  ;       KEYBOARD WITH PARALLEL INTERFACE.
                  ;
                  ;       ADDRESSES SHOWN ARE FOR 6530 ON KIM-1 COMPUTER.
                  ;       KEYBOARD DATA LINES TO PORT A BITS 0 TO 6,
                  ;       NEGATIVE-GOING STROBE TO BIT 7.
                  ;
                  ;       ON ENTRY: IF ALFALK IS NON-0, THEN ALL LOWERCASE LETTERS WILL
                  ;       BE RETURNED AS THE EQUIVALENT UPPERCASE ALPHA.
                  ;       ON RETURN: REGISTER A = ASCII CODE FOR KEY PRESSED;
                  ;       X AND Y PRESERVED.
                  ;
1700              PAD     =       $1700       ;KIM PORT A DATA REGISTER ON 6530
1701              PADD    =       $1701       ;KIM PORT A DATA DIRECTION REGISTER
                  ;
0000                      *=      $1780       ;PROGRAM ORIGIN
                  ;
1780 A900         INCH    LDA     #$00
1782 8D0117               STA     PADD        ;SET PORT DIRECTION = INPUTS
1785 AD0017       INCH1   LDA     PAD         ;TEST PORT
1788 30FB                 BMI     INCH1       ;WAIT FOR STROBE PULSE
178A 2C0017       INCH2   BIT     PAD
178D 10FB                 BPL     INCH2       ;WAIT FOR END OF STROBE
                  ;
                  ;       IF ALPHA-LOCK FLAG IS SET, FOLD ANY LOWERCASE LETTERS TO
                  ;       EQUIVALENT UPPERCASE LETTERS.
                  ;
178F 48           FOLD    PHA                 ;SAVE CHARACTER TEMPORARILY
1790 ADA317               LDA     ALFALK      ;RECALL "ALPHA LOCK" FLAG
1793 F00C                 BEQ     FOLD2       ;BRANCH IF NO FOLDING DESIRED
1795 68                   PLA                 ;ELSE RECALL CHARACTER
1796 C97B                 CMP     #$7B        ;LOWER CASE "Z" + 1
1798 B006                 BCS     FOLD1       ;BRANCH IF PUNCTUATION
179A C961                 CMP     #$61        ;LOWER CASE "A"
179C 9002                 BCC     FOLD1       ;BRANCH IF NOT LOWER CASE ALPHA
179E E920                 SBC     #$20        ;ELSE FOLD TO EQUIVALENT UPPERCASE
17A0 60           FOLD1   RTS
                  ;
17A1 68           FOLD2   PLA                 ;RECALL CHARACTER
17A2 60                   RTS
                  ;
                  ;       ALPHA LOCK FLAG (DEFAULT = ALLOW LOWER CASE)...
                  ;
17A3              ALFALK  .BYTE   0           ;"ALPHA LOCK" FLAG;  NON-0=UPPERCASE ONLY.
                  ;
0000                      .END

NO ERROR LINES
```

Program 4: Block-Move Memory Routine

```
GENERAL BLOCK-MOVE SUBROUTINE                        MTU 6502 ASSEMBLER 1.0

0000                      .PAGE  'GENERAL BLOCK-MOVE SUBROUTINE'
0000                      *=     0        ;ZERO PAGE ORIGIN
0000           FROM       *=*+   2        ;STARTING ADDRESS OF BLOCK TO BE COPIED
0002           TO         *=*+   2        ;STARTING ADDRESS OF DESTINATION
0004           COUNT      *=*+   2        ;NUMBER OF BYTES TO BE MOVED
0006           ;
0006                      *=     $2000    ;ORIGIN FOR PROGRAM
2000           ;
2000           ;          THIS ROUTINE COPIES A BLOCK OF ANY SIZE FROM ONE
2000           ;          LOCATION TO ANOTHER.
2000           ;
2000           ;          ON ENTRY: FROM (2 BYTES) IS THE STARTING ADDRESS OF
2000           ;          THE BLOCK TO BE COPIED; TO (2 BYTES) IS THE DESIRED
2000           ;          STARTING DESTINATION ADDRESS FOR THE COPY; COUNT
2000           ;          (2 BYTES) IS THE NUMBER OF BYTES TO COPY.
2000           ;
2000           ;          ON RETURN: NO REGISTERS PRESERVED; FROM, TO AND COUNT
2000           ;          ARE CLOBBERED.
2000           ;
2000           ;          NOTE: THE DESTINATION BLOCK MAY OVERLAP THE SOURCE
2000           ;          BLOCK ONLY IF "TO" IS AT A LOWER ADDRESS THAN "FROM".
2000           ;
2000 A000      BLKMOV     LDY    #0       ;INITIAL INDEX WITHIN A PAGE
2002 A605                 LDX    COUNT+1  ;NUMBER OF PAGES TO BE MOVED
2004 F00E                 BEQ    FRCMOV   ;BRANCH IF ONLY A FRACTIONAL PAGE
2006           ;
2006           ;          THIS LOOP COPIES ENTIRE PAGES...
2006           ;
2006 B100      PAGMOV     LDA    (FROM),Y ;FETCH A BYTE FROM SOURCE
2008 9102                 STA    (TO),Y   ;COPY TO DESTINATION
200A C8                   INY             ;BUMP POINTER
200B D0F9                 BNE    PAGMOV   ;REPEAT TILL ENTIRE PAGE MOVED
200D E601                 INC    FROM+1   ;BUMP HI BYTE OF POINTERS
200F E603                 INC    TO+1
2011 CA                   DEX             ;DECREMENT COUNT OF PAGES TO COPY
2012 D0F2                 BNE    PAGMOV   ;REPEAT TILL ALL WHOLE PAGES COPIED
2014           ;
2014           ;          THIS LOOP COPIES THE FINAL FRACTION OF A PAGE...
2014           ;
2014 A604      FRCMOV     LDX    COUNT    ;RECALL NUMBER OF BYTES LEFT TO COPY
2016 F008                 BEQ    DONEMV   ;BRANCH IF COUNT IS EXACT PAGE MULTIPLE
2018 B100      FRLOOP     LDA    (FROM),Y ;FETCH A BYTE FROM SOURCE
201A 9102                 STA    (TO),Y   ;COPY TO DESTINATION
201C C8                   INY             ;BUMP INDEX
201D CA                   DEX             ;DECREMENT COUNT OF BYTES LEFT
201E D0F8                 BNE    FRLOOP   ;REPEAT UNTIL DONE
2020           ;
2020 60        DONEMV     RTS
2021           ;
2021                      .END

    0 ERRORS IN PASS 2
```

eral principle of loop optimization:

Rule #12. To optimize loop execution speed, try to remove unnecessary compares from within the loop.

About the only way we can remove more compares from Program 4 is to "unwind" part of the loop and, instead, write some of the loop code "in-line." Since we know the first loop will always move exactly 256 bytes, we can move two bytes at a time instead of one before checking for a page crossing:

```
        ...
PAGMOV  LDA  (FROM),Y  ;FETCH A BYTE FROM SOURCE
        STA  (TO),Y    ;COPY TO DESTINATION
        INY            ;BUMP POINTER
        LDA  (FROM),Y  ;FETCH A BYTE FROM SOURCE
        STA  (TO),Y    ;COPY TO DESTINATION
        INY            ;BUMP POINTER
        BNE  PAGMOV    ;REPEAT TILL ENTIRE PAGE
                        MOVED
        ...
```

This loop now takes 14.5 cycles per byte moved versus 16 cycles for the equivalent loop of Program 4, because the three cycle BNE instruction is only executed for every other byte moved. The loop can be unwound still further to move four, eight or more bytes per pass, but the speed improvement gained drops off rapidly as more code is written inline.

In the next installment I will explore some techniques for optimizing jumps and subroutine calls. ©

Part One:

Introduction To Binary Numbers

Charles Brannon
Greensboro, NC

To use machine language, or even to be truly computer literate, requires an understanding of binary numbers. The reason is simplicity — a computer can only understand two states, whether it is +5 or -5 volts, yes or no, or on or off. These simple relationships are expressed in a computer's world as merely a one or a zero. Because such a number has only two elements, one and zero, it is called binary. Mathematically, binary numbers are called base two numbers. We shall attempt to understand the computer more fully in this mathematical way.

The numbers we commonly use, whether we call them integers, counting numbers, or real numbers, are understood to be in base ten. Sometimes base ten is called *decimal* from the Latin word *decem*, meaning ten. We can look at any number on a digit-by-digit basis:

5	3	2	7
1000	100	10	1
10^3	10^2	10^1	10^0

So the number 5,327 can be expressed as $5x10^3 + 3x10^2 + 2x10^1 + 7x10^0$. Remember that any number raised to the zero power is one, so that we have 5x1000 +3x100+2x10+7x1 = 5000+300+20 +7, which totals 5,327. It is simply a matter of multiplying each digit by a power of ten. For other number bases, only two things change — the base itself, and the number of digits used. Let's say we have 4302, but in base *five*. It can be shown as being equal to $4x5^3 + 3x5^2 + 0x5^1 + 2x5^0 = 4x125 + 3x25 + 0x5 + 2$, which, when totaled, is equal to 581 in base ten. It can be seen that 10 (pronounced ONE-ZERO) is equal to five. In fact, 10 is always equal to the base itself. Therefore, 10 in base two must be equal to two. Hey! There went our first binary number. In base five, the only digits are (0,1,2,3,4). The digit 5 is not present because it is represented by 10 (remember?). When we jump all the way down to base two, the only digits we'll have are zero and one. That's just what a computer needs. Any number can be converted to decimal in the same way we did for that base five number. Let's take the binary number 1101. It can be expressed as $1x2^3 + 1x2^2 + 0x2^1 + 0x2^1 + 1x2^0 = 8 + 4 + 0 + 1 = 13$. Because we are always multiplying by either a zero or a one, we really only have to "sum up" the ones to get the value. For example:

0	0	1	0	1	1	0	1
128	64	32	16	8	4	2	1

This would give us 32 + 8 + 4 + 1, or 45. Remember, we skip the zeros. The numbers at the bottom could be extended as far left as necessary. Just multiply the current value by two to get the next one.

On most microcomputers, the numbers have only eight digits. Each digit is called a *bit*, which is short for Binary digIT. Eight bits together comprise a *byte*. Since numbers are stored in "little boxes" called *memory locations*, the memory size of your computer now means something. If you have a 16K computer, that means it has roughly 16,000 of these boxes. Each box can store one byte. Since any character (the letter "a" or the number "9") can be stored as a number from zero to 255 (the highest number that can fit in eight bits), memory is often referred to as "characters" of storage. Therefore, if your computer can display 25 lines of 40 characters for a total of 1000 characters, it would take 1000 bytes to store one screen of information.

You should now be able to convert binary numbers to decimal. Now we'll work on going the other way. Basically, the trick is to break the number down into the powers of two. 61 probably has a 32 in it, but not 64 or 128. If it also has 16, then we have 32 + 16 = 48. Subtracting 48 from 61 gives us 13. Of the last possibilities (8,4,2,1), we choose 8 + 4 + 1 = 13. Therefore, we can now total 32 + 16 + 8 + 4 + 1 to get 61. Now we "fill in" the bits to form the binary number.

BITS	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**1**
POWERS OF TWO	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**

We put a one above the powers of two we used, and a zero above the rest. So now we know that $61_{10} = 00111101_2$.

The previously mentioned method will give you a feeling for how binary numbers work, but it is sometimes easier to use the "division method" to convert a decimal number to binary.

The Division Method

	Number	Remainder
1. Write down your number.	37	
2. Divide it by two.	18	
3. Write down the remainder.		1
4. Continue...	9	0
	4	1
	2	0
5. When you get to one, two	1	0
can go into one zero times,		
with a remainder of one.	0	1

6. Read the remainders from the bottom up. We have: 100101

7. You can check the number:

0	**0**	**1**	**0**	**0**	**1**	**0**	**1**
128	**64**	**32**	**16**	**8**	**4**	**2**	

32 + 4 + 1 = 37.

This method takes the guessing out of conversion. However, the easiest way of all is to use your computer to do the conversion. A short program is included at the end of this article which will convert a decimal number to binary. Because it is written to run on any BASIC-speaking computer, you may want to modify it and add any special features unique to your computer.

To reinforce your knowledge, I strongly suggest that you do the exercises included at the end of this article. (Without your computer!) Next month, we'll get into working with these binary numbers — adding and subtracting them.

Exercises

1. Convert to decimal:

a) 10101 b) 110011

c) 0111100 d) 11111111

2. Convert to binary

a) 52 b) 234

c) 66 d) 15

3. Extend the chart to 16 bits:

— — — — — — — — 128 64 32 16 8 4 2 1

```
1ØØ REM  TO CONVERT A DECIMAL NUMBER
11Ø REM  TO BINARY
12Ø REM
13Ø PRINT "ENTER THE DECIMAL NUMBER:"
14Ø INPUT D
15Ø FOR I=7 TO Ø STEP -1
16Ø P=2↑I
17Ø IF INT(D/P)=1 THEN PRINT "1"; : D=D-P: GOTO 19Ø
18Ø PRINT "Ø";
19Ø NEXT I
2ØØ PRINT
21Ø END
```

©

Editor's Note: In February of this year we ran the following Reader's Feedback plea:

> I am a high school science teacher. I am a novice Apple Computer programmer. I would appreciate **COMPUTE!** articles designed to enhance the programming ability of novice Apple programmers... In-depth articles of Apple POKEing, PEEKing, and CALLS would be very helpful....

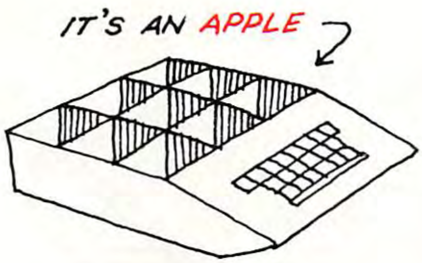
In response, we received the following from Gary who Kathleen and I had the pleasure of meeting at this year's West Coast Computer Faire. Gary, 11, gave us permission to run the response as an article. We think it's an excellent piece for beginners. — RCL

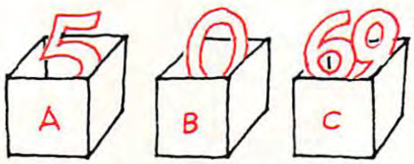
An Apple Primer

Gary Lin
San Jose, CA

Having trouble with PEEK's, POKE's, and CALL's? Here's a rough explanation:

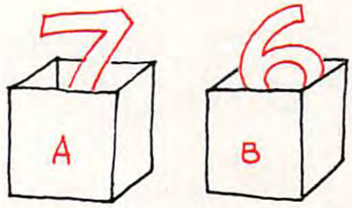
1. Imagine the memory of an Apple is divided up to a bunch of boxes.



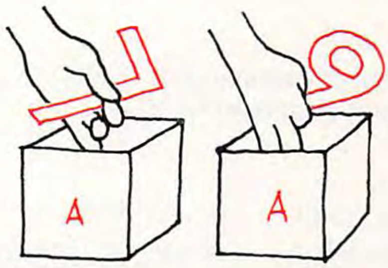


The Apple stores numbers in those boxes. Each box can have only one number assigned to it. Each box has its own personal address.

The Apple, like a mailman, gets a number and delivers it to the box. In this example, Box 29 gets the number 5.



The Apple stores numbers in the boxes of its memory. Okay, suppose the Apple sends a 9 to Box A. Box A holds the number 7.



The Apple takes out the 7 and throws it away. Now Box A is clear. ⇨ Then it puts a 9 into Box A.

In reality, there are no boxes. Instead there are addresses. Addresses, like boxes, can be assigned a number. For example, address 2 may hold the number 15. The Apple's addresses are numbered in hex, a complicated numbering scheme*. Only the Apple understands HEX, and humans need to know the decimal equivalent.

Don't worry about hex, most beginners say, "What? hex, are you kidding?"

*Base 16

Okay, each address holds a number. If the Apple didn't assign a number to an address, the address automatically holds a 0.

Suppose you type, in BASIC, "COW":

The Apple does a long process and sticks "COW" into its memory. It converts "COW" (or whatever you type in) into little numbers and assigns the numbers to some address. Somewhere, in an address, is "COW."

Well, you can do it a different way!

The Secret

Introducing the amazingly, one and only,

POKE!

Okay, POKE is a command that tells the computer to stick a number into an address.

Let's type in POKE 135,6. Here's what the computer does: It converts 6 into hex and runs over to the address. Then 6 is placed in.

POKE A, B

A is the address (decimal) where you're going to stick a number. B is the number. Try it yourself!

You tell me. What do I do with some number in some address. Here's the next biggie:

PEEK!

Suppose ten is stored someplace, maybe address A. We want to know what is in address A. So we type

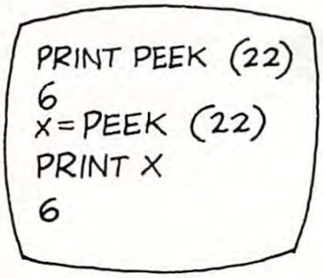
"PEEK (100)" (100 is the address for A). The computer figures out what 100 is in hex and goes to that address and picks out the number stored there.

It runs back and converts the number to decimal. To show what is at 100, we PRINT PEEK (100) and it'll print it.

PRINT PEEK (A)

need to show the number at address.

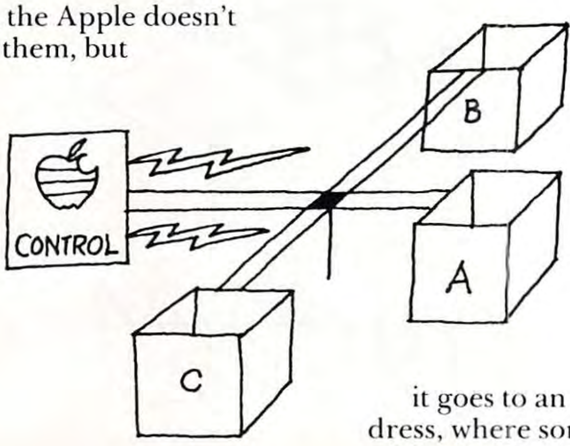
A is the address where you want to know what's there.



You can assign to a variable (another address, actually) the number which is at location 22 (decimal).

The last, but not least: **CALL!**

No, the Apple doesn't call them, but



it goes to an address, where something lies, (usually a program) and starts RUNning itself.

A BRIGHT NEW STAR FROM ANDROMEDA!

NEW ROM BOARD FOR THE APPLE II*
$125.00 WITH UTILITY ROM.

With Andromeda's new ROM Board, you can plug many useful utility programs into your Apple II. Because ROM memory never forgets, you can access these utilities instantly without having to load them from disk.

The ROM Board comes with the utility ROM, which gives you five powerful options to apply to your Applesoft* programs. With the Utility ROM, you can do automatic line numbering, control a program list with a page mode, restore a crashed Applesoft* program in memory, alphabetize a disk catalogue and create a disk without DOS, giving you an extra 8K on your disk. Any of Soft Control Systems' other ROMS can be used, such as the Dual DOS in ROM, and 'Your'ple ROM.

You can install 2K PROMS, 4K PROMS, or even 2K RAM chips in each of the two memory sockets. So you can even have the Read - Write capability of RAM to develop PROM Programs yourself, or just have an extra 2K RAM for your machine - Language programs. Two 2732 PROMS allow a total of 8K of memory on the Board.

Now with One Year Warranty.

Don't forget the Andromeda 16K RAM Expansion Board $195.00

ANDROMEDA
INCORPORATED
P.O. Box 19144
Greensboro, NC. 27410
919 852-1482

Distributed By:

COMPUTER DATA SERVICES
P.O. Box 696
Amherst, NH. 03031
603 673-7375

*Apple II and Applesoft are trademarks

You say, "that's nice, but what's so big about it?"

The Big Trick of

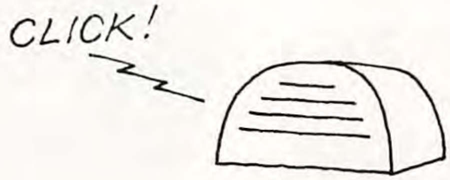
PEEKs, POKEs, and CALLs

Certain addresses in the Apple do nice things, depending on what's stored there.

Like POKE 50,127 (Type it in!)

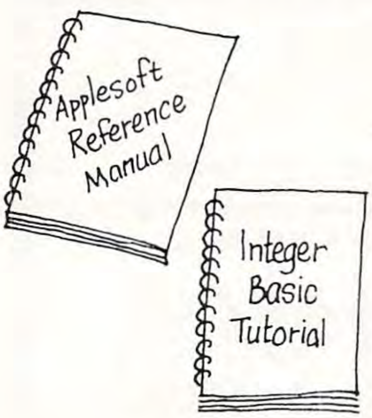
The computer sees 127 is in location 50 and the built-in command tells it to do something. What it does depends on the value stored. Try POKE 50,63 and POKE 50,255. You see, address 50 tells the computer to do something. PEEK does the same, but PEEK doesn't stick a value in — it just activates whatever is at that location.

Like PEEK (-16336)
(Listen carefully!)



CALLs also do stuff like CALL -936 clears the screen. CALL -151 enters Monitor. There are hundreds of POKEs, PEEKs, and CALLs that do special things. That's why people love them.

To find out more, look in these books:



and

I hope this helps you understand it better, although it's not much of a primer. If you have any questions write me:

Gary Lin
1598 Lock Lomond
San Jose, CA 95129

©

Page Flipper: Five Hires And Four Lores Pages For The Apple

Richard Cornelius
Department of Chemistry
Wichita State University

Five high resolution pages? Four pages of text or low resolution? The facility to copy, overlay or xcopy from one page to another? Yes, all of these and more are available on the (48K) Apple II Plus, and here is the program, PAGE FLIPPER, which demonstrates their use.

Simple arithmetic tells us that space is available on the 48K Apple for more than two high resolution pages. Each hires page occupies 8K (8192 bytes) of memory. Let's digress for a moment to see why that much space is required. The resolution on the Apple is 280 dots across by 192 dots high, which means that the Apple must store information regarding 53760 dots. Each dot is controlled by one bit, so we need $53760 \div 8 = 6720$ bytes to record all of the on/off information for all of the dots on the screen. In addition, we need to record information about the color of the dot.

On the Apple screen, the colors in a horizontal series of seven dots are controlled by a single color control bit. If the color bit is off, then the colors in the seven dots can be any of those given by HCOLOR values of 0 to 3, depending upon the locations of the dots. When HCOLORs 4 to 7 are selected, the color bit is on. The on/off control bits for the seven dots plus the color bit make up one byte. Since each byte controls only seven dots, we need $53760 \div 7 = 7680$ bytes. Some space in the 8K reserved for each page is not used, but there is simply no way to store all of the necessary information in, say, 6K. Even the space on each hires page that is not used to store graphics information cannot readily be used for other purposes because it is fragmented. After every set of 120 bytes that is used for information storage, there follows a set of eight bytes that is not used. Thus 512 bytes of unused space on each high resolution memory page is divided into 64 pieces of eight bytes each.

If we use 8K of memory to store the information on a single hires page, then an Apple with 48K could store enough information for six pages. Since we need to leave some room for a program, we must limit ourselves to five pages. Hires page one is located beginning at 8K and continuing up to 16K, and hires page two occupies memory from 16K to 24K. These two pages are the ones that are readily accessible through the Applesoft commands HGR and HGR2. Pages three, four, and five can be defined to begin at 24, 32, and 40K.

The area in memory to which the HPLOT and DRAW commands write is controlled by a POKE to position 230. POKEing a 32 specifies page one, 64 says page two, and 96, 128, and 160 are used to direct writing to pages three, four, and five. To see that this works, first scroll to the bottom of the Apple screen and then enter the HGR command. Now tell the computer HCOLOR=1 and HPLOT 0,0 to 279, 191 to put a line on the screen. Next POKE 230,64 to direct plotting to hires page two, set HCOLOR=2 and PLOT 279,0 to 0,191. No line appears on the screen because you are viewing page one and the plotting appeared on page two. If you POKE -16299,0 you will switch the Apple to display page two and presto, you see the second line that you plotted.

The Flipper

Unfortunately, we cannot so simply switch to see the other hires pages. Instead we must actually move the information on these pages down to page one or two so that it can be displayed. To accomplish this, we use a short machine language program for speed. This machine language routine is given in Program 1. You don't need to understand any machine language in order to use this little program because it is entered into memory by POKE statements in PAGE FLIPPER, it is executed by a CALL 768, and its function is controlled by POKE statements. Depending upon the POKEd values the routine can a) erase a page, b) copy information from one page to another, discarding the information originally on the destination page, c) overlay a page onto a different one so that the images from the two pages are superimposed, or d) "xcopy" the contents of one page onto another. "Xcopy" is most easily described as being analogous to the XDRAW routine which handles shapes in Applesoft. If you XDRAW a shape on top of some existing image, you get a composite of both the first image and the shape. If you XDRAW again, the shape disappears and you are left with only the original image. In PAGE FLIPPER, if you "xcopy" the contents of one page onto another and then xcopy it again, you are left with the original image also. For those interested in the machine language, "xcopy" uses an exclusive or (EOR) while overlay uses an inclusive or (ORA).

PAGE FLIPPER can also manipulate the pages of memory which store text or low resolution (lores) graphics. Just as the Apple has two hires pages, it also has two text/lores pages which begin at 1K and 2K. Much less information is required to store the letters that appear on the screen than is needed to store a screenful of hires graphics, so each text/hires page occupies only 1K of memory. Since the text screen offers 40 characters across and 24 down, there are 960 "boxes" where characters can be displayed. The contents of each little box is controlled by one byte of memory, so 960 bytes are all that is needed.

As is the case for the hires screens, the unused memory within the 1K allocated for a text/lores page is fragmented into many 8-byte pieces. The image on the lores screen corresponds to the same information as the text screen, but the image displayed is different when the Apple is in lores mode. Each byte which specifies a character on the text screen determines the colors (COLOR 0 to 15) of two blocks on the lores screen which occupy the same screen location as the corresponding character. Four bits (a nibble) determine the color of the upper block, and four bits determine the color of the lower block. In PAGE FLIPPER page three of text/lores is at 3K and page four is at 4K. Page four is used only to save the instructions. A schematic map of memory usage in the program is given in Figure 1. When the machine language routine is used to move any of the text/lores pages, it has less to move than when it operates on any of the hires pages, so another POKE statement is used to specify the size of the page that is being moved.

In addition to the POKEs used to adapt the machine language routine to different purposes, POKE statements are also used to control the display mode of the Apple. These POKEs are outlined in the Apple manuals. All of the POKE positions used in PAGE FLIPPER are listed in Table 1.

Easily Moved Pages

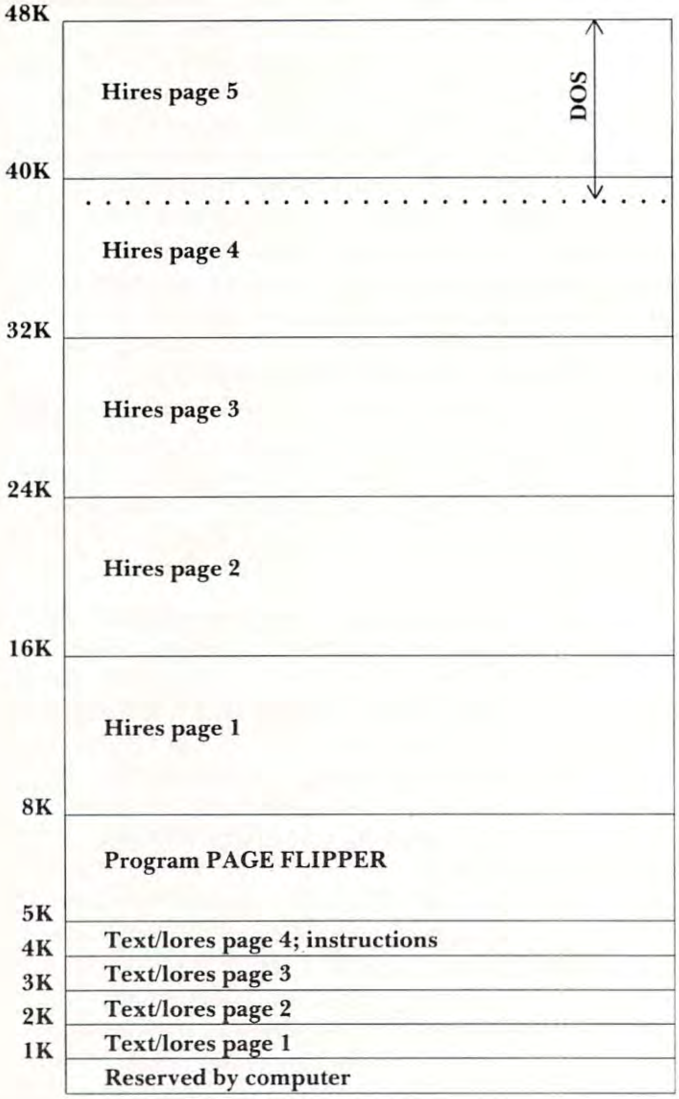
Now that the memory layout and details have been explained, let's look at the PAGE FLIPPER program itself. The first thing to notice is that the program is divided into two parts. The initial part completes a few tasks and then loads the second part. This division is necessary for two reasons. One reason is that the division leaves that part of the program which does most of the work (the second part) small enough that room is left for three pages of text/lores memory plus a fourth page for the directions. The other, more critical, reason is that the first program POKEs certain values into the correct positions so that the second program loads above text/lores page four. Normally an Applesoft program loads starting at 2K, but we want to be able to copy images into that area without overwriting our program.

The first executable statements in the initial part of the program POKE into memory the machine language transfer routine so that it will be there when it is needed. Beginning in line 2000, the instructions are printed, but they are printed on text page one while hires graphics page two is displayed so the user doesn't see them yet. In line 2190, these instructions are moved into the memory area for hires page one for safekeeping while the second half of the program is loaded later. After

Table 1. POKE Positions and Functions

Positions	Values to be POKEd	Function
103,104	1,20	sets spot for beginning of program to above text/lores page 4
230	32,64,96,128, or 160	makes HPLOT write on hires page 1, 2, 3, 4, or 5
768 to 804	values in statement number 1040	puts machine language transfer routine into memory
773	32 x hires page no. or 4 x lores page no.	determines page from which image will be taken
781	32 x hires page no. or 4 x lores page no.	determines page to which image will be written
785	32 for hires, 4 for text/lores	sets size of page to be transferred
790,791	169,0 177,6 17,8 81,8	erase page [LDA #$00] copy a page [LDA ($06),Y] overlay [ORA ($08),Y] xcopy [EOR ($08),Y]
5120,5121,5122	0,0,0	allows execution of program loaded above text/lores page 4
-16297	0	display hires if in graphics mode
-16298	0	display lores if in graphics mode
-16299	0	display page 1 (hires or text/lores)
-16300	0	display page 2 (hires or text/lores)
-16301	0	mix text and graphics if in graphics mode
-16302	0	full screen graphics if in graphics mode
-16303	0	text mode
-16304	0	graphics mode

Figure 1. Schematic RAM Memory Map



this information is moved, text page one is cleared and the introductory screen image is printed. The next to last action in the initial part is moving the pointers which specify where Applesoft programs begin so that when the last statement runs the second part of the program, it will load above text/lores page four.

For convenience in reference, the statements in the second part of the program are numbered higher than those in the first part, but this numbering scheme is not required for its operation. As part of the initialization routine, HIMEM is set to 8192 in order to prevent string variables from being written onto one of the hires pages. The commands IN#0 and PR#0 disconnect DOS so that DOS can be erased by using the machine language transfer routine. Until DOS is disabled hires pages four and five cannot be used. The remainder of the initialization routine plots random lines on the hires pages two through five in different colors, copies the instructions from hires page one to text page one, and then clears and plots random lines on hires page one.

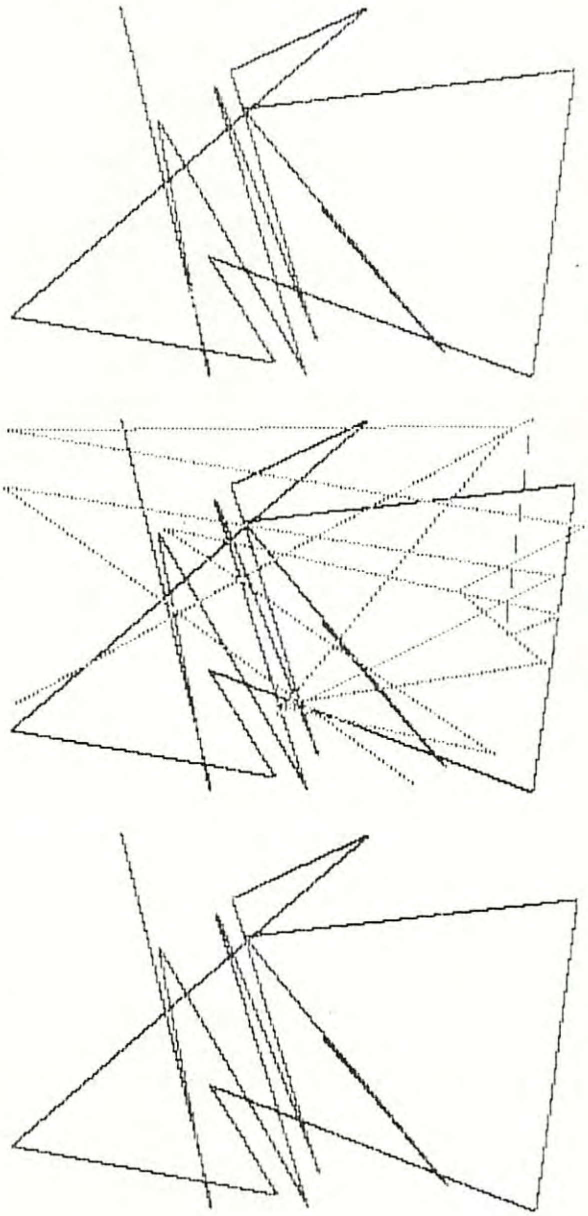
When the program reaches the input routines, we can see the power of being able to readily move pages in memory. The directions that are displayed on the screen are given in Figure 3 except that the inverse characters on the screen are represented only by underlines in the figure. All of the input is controlled by GET statements so that the user never needs to hit return. At any time, an "I" will move the instructions to text page one and display them. A "T" puts the display into text mode, "L" gives lores graphics, and an "H" changes to hires graphics. The commands "M" and "F" specify mixed and full screen graphics. "Q" is used to quit the program. The "Q" reboots DOS which was cleared to make room for hires pages four and five. The command "D" followed by a one or two shows page one or two which may be text, lores, or hires depending on which keys have been pressed previously. An "E" followed by a number can erase any one of the hires pages 1-5 or text/lores pages 1-3 depending upon the current mode of display

Figure 2. Xcopying with Page Flipper

A) Random lines on page 2.

B) Page 3 xcopied onto page 2.

C) Page 3 xcopied onto page 2 again.



A)

B)

C)

Figure 3. Instruction Page

Options: Instructions

TEXT, LOW — OR HIGH-RES GRAPHICS

MIXED OR FULL SCREEN GRAPHICS

QUIT AND REBOOT

The following commands must be followed by one or two page numbers (represented by X and Y). Accessible pages are Hires 1-5 (1-2 for display) and Text/Lores 1-3 (1-2 for display).

DISPLAY X

ERASE X

COPY X ONTO Y

XCOPY X ONTO Y

OVERLAY X ONTO Y

(hires or text/lores). The commands "C", "X", and "O" each need to be followed by two numbers. The first number gives the source page and the second number the destination page for the transfer routine.

As an example of what PAGE FLIPPER can demonstrate, press the following sequence of letters, observing what (if anything) happens after each entry; L, H, D, 2, X, 3, 2, X, 3, 2. This sequence changes the display first to lores page one, then the white random lines of hires page one, and then to the green random lines of hires page two (Figure 3a). The X command xcopies the violet lines of hires page three onto two to give a result like in Figure 3b and then again xcopies three onto two. The result is exactly the image that was first on page two. Next try to xcopy first the green lines of page two, then the violet lines of page three, the orange lines of page four, and the blue lines of page five onto page one: D, 1, X, 2, 1, X, 3, 1, X, 4, 1, X, 5, 1. Now that all of these lines are on page one, xdraw pages two through five onto page one once again. Color by color, all of the lines disappear except the white ones that were originally on page one. Now try your hand with the lores screens in the same manner. Remember that at any time you can press "I" to return to the instruction page.

This program by itself is fun to play with, but does not actually accomplish much. The accomplishment will come when you make use of the concepts pulled together here and put them in your own programs to improve them. Imagine the enhanced graphics capabilities that you will have with five hires pages. Get to work!

Program 1: Machine Language Transfer Routine

Location	Value	Operation	
0300-	A9 00	LDA	#$00
0302-	85 06	STA	$06
0304-	A9 20	LDA	#$20
0306-	85 07	STA	$07
0308-	A9 00	LDA	#$00
030A-	85 08	STA	$08
030C-	A9 20	LDA	#$20
030E-	85 09	STA	$09
0310-	A2 20	LDX	#$20
0312-	A0 00	LDY	#$00
0314-	B1 06	LDA	($06),Y
0316-	A9 00	LDA	#$00
0318-	91 08	STA	($08),Y
031A-	88	DEY	
031B-	D0 F7	BNE	$0314
031D-	E6 07	INC	$07
031F-	E6 09	INC	$09
0321-	CA	DEX	
0322-	D0 F0	BNE	$0314
0324-	60	RTS	

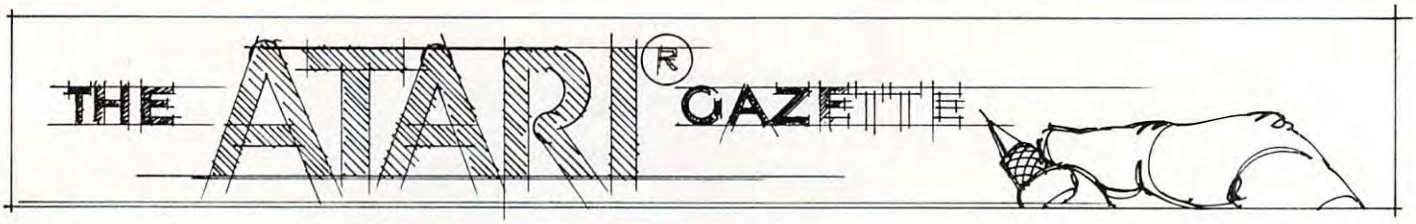
Program 2: "Page Flipper"

```
100  REM *** PAGE FLIPPER ***
110  REM     INITIAL PART

130  REM  BY DICK CORNELIUS
140  REM     CHEMISTRY DEPT.
150  REM     WICHITA STATE UNIV.
160  REM     WICHITA, KS  67208

170  REM     (316) 689-3120

180  REM  POKES USED BY THE MACHINE LANGUAGE TRANSFER ROUTINE:
190  REM   773:  SPECIFIES THE "FROM" PAGE
200  REM         32,64,96,128, OR 160 FOR HIRES PAGE 1,2,3,4, OR 5
210  REM         4,8,12, OR 16 FOR TEXT/LORES PAGE 1,2,3, OR 4
220  REM   781:  SPECIFIES THE "TO" PAGE (SEE VALUES ABOVE)
230  REM   785:  SIZE OF PAGE:  4-TEXT/LORES, 8-HIRES
240  REM   790:  DETERMINES (WITH 791) THE NATURE OF THE TRANSFER
250  REM         177-COPY, 81-XCOPY, 17-OVERLAY, 169-ERASE
260  REM   791:  0-ERASE, 6-COPY, 8-OVERLAY OR XCOPY

1000  REM
        **INITIALIZATION**

1010  REM    THE NEXT TWO STATEMENTS POKE INTO MEMORY THE MACHINE
1020  REM    LANGUAGE MEMORY TRANSFER ROUTINE
1030  FOR SPOT = 768 TO 804: READ CODE: POKE SPOT,CODE: NEXT
1040  DATA  169,0,133,6,169,32,133,7,169,0,133,8,169,64,133,9,162,32,
      160,0,177,6,81,8,145,8,136,208,247,230,7,230,9,202,208,240,96
2000  REM
        **PRINT INSTRUCTIONS**
```

```
2010  HIMEM: 8192
2020  HOME : HGR2
2030  HOME
2040  PRINT "OPTIONS:";
2050  HTAB 10: INVERSE : PRINT "I";: NORMAL : PRINT "NSTRUCTIONS"
2060  PRINT : HTAB 10: INVERSE : PRINT "T";: NORMAL : PRINT "EXT, ";:
      INVERSE : PRINT "L";: NORMAL : PRINT "OW- OR ";: INVERSE : PRINT
     "H";: NORMAL : PRINT "IGH-RES GRAPHICS";
2070  PRINT : HTAB 10: INVERSE : PRINT "M";: NORMAL : PRINT "IXED OR
     ";: INVERSE : PRINT "F";: NORMAL : PRINT "ULL SCREEN GRAPHICS"
2080  PRINT : HTAB 10: INVERSE : PRINT "Q";: NORMAL : PRINT "UIT AND
     REBOOT"
2090  PRINT : PRINT "THE FOLLOWING COMMANDS MUST BE FOLLOWED"
2100  PRINT "BY ONE OR TWO PAGE NUMBERS (REPRESENTED"
2110  PRINT "BY X AND Y). ACCESSIBLE PAGES ARE HIRES"
2120  PRINT "1-5 (1-2 FOR DISPLAY) AND TEXT/LORES 1-3";
2130  PRINT "(1-2 FOR DISPLAY).
2140  PRINT : HTAB 10: INVERSE : PRINT "D";: NORMAL : PRINT "ISPLAY "
     ;: INVERSE : PRINT "X": NORMAL
2150  PRINT : HTAB 10: INVERSE : PRINT "E";: NORMAL : PRINT "RASE ";:
      INVERSE : PRINT "X": NORMAL
2160  PRINT : HTAB 10: INVERSE : PRINT "C";: NORMAL : PRINT "OPY ";: INVERS
E : PRINT "X";: NORMAL : PRINT " ONTO ";: INVERSE : PRINT "Y": NORMAL
2170  PRINT : HTAB 10: INVERSE : PRINT "X";: NORMAL : PRINT "COPY ";:
      INVERSE : PRINT "X";: NORMAL : PRINT " ONTO ";: INVERSE : PRINT
     "Y": NORMAL
2180  PRINT : HTAB 10: INVERSE : PRINT "O";: NORMAL : PRINT "VERLAY "
     ;: INVERSE : PRINT "X";: NORMAL : PRINT " ONTO ";: INVERSE : PRINT
     "Y": NORMAL
2190  POKE 773,4: POKE 781,32: POKE 785,4: POKE 790,177: POKE 791,6: CALL
     768: REM MOVES PAGE 1 TO HIRES PAGE 1
3000  REM
        **PRINT FIRST SCREEN IMAGE**

3010  TEXT : HOME : VTAB 2: HTAB 13:
      PRINT "/PAGE FLIPPER/
3020  PRINT : PRINT : PRINT "THIS PRO
      GRAM ALLOWS THE EASY DISPLAY OF"
3030  PRINT "THE VARIOUS TEXT AND
      GRAPHICS PAGES AND"
3040  PRINT "DEMONSTRATES THE /XPLOT/
      UTILITY."
3050  PRINT : PRINT "WRITTEN BY DICK
      CORNELIUS
3060  PRINT " WICHITA STATE UNIVERSITY"
3070  PRINT " WICHITA, KS 67208"
3080  PRINT : PRINT : PRINT : PRINT
      "PLEASE WAIT A FEW SECONDS..."
3090  VTAB 21
4000  REM
        **LOAD SECOND HALF**

4010  POKE 104,20: POKE 103,1: REM MOVES
      STARTING POSITION FOR APPLESOFT
      PROGRAMS
4020  POKE 5120,0: POKE 5121,0: POKE
      5122,0: REM  PUTS ZEROS INTO
      STARTING POSITIONS
4030 D$ =  CHR$ (13) +  CHR$ (4)
4040  PRINT D$"RUN PAGE FLIPPER.FINAL
      PART"
```

```
5000  REM ***PAGE FLIPPER***
5010  REM     FINAL PART
5020  REM   UPDATED 6/22/81

5030  REM  BY DICK CORNELIUS
5040  REM     CHEMISTRY DEPT.
5050  REM     WICHITA STATE UNIV.
5060  REM     WICHITA, KS  67208

5070  REM     (316) 689-3120

6000  REM
        **INITIALIZATION**

6010  HIMEM: 8192: REM  KEEPS ALL VARIABLES STORED BELOW HIRES PAGE 1
6020 BELL$ =  CHR$ (7):MODE$ = "T/L"
6030  IN# 0: REM    THESE TWO COMMANDS
6040  PR# 0: REM    DISCONNECT DOS
6050  REM  THE FOLLOWING STATEMENTS CLEAR THE VARIOUS PAGES
6060  POKE 790,169: POKE 791,0
6070  POKE 785,128: POKE 773,32: POKE 781,64: CALL 768: REM CLEARS HIRE
     S PAGES 2-5
6080  REM DOS HAS NOW BEEN ERASED
6090  REM
        **DRAWING LINES ON DIFFERENT PAGES**

6100  FOR PAGE = 2 TO 5
6110  HCOLOR= PAGE - 1: IF PAGE > 3 THEN  HCOLOR= PAGE + 1
6120  GOSUB 8000: REM LINES PLOTTED ON HIRES PAGES 2-5 HERE
6130  NEXT
6140  POKE 773,32: POKE 781,4: POKE 785,4: POKE 790,177: POKE 791,6: CALL
     768: REM MOVES INSTRUCTIONS ONTO PAGE 1
6150  POKE 781,16: CALL 768: REM MOVES INSTRUCTIONS ONTO PAGE 4
6160  POKE 781,32: POKE 785,32: POKE 790,169: POKE 791,0: CALL 768: REM
     CLEARS HIRES PAGE1
6170 PAGE = 1: HCOLOR= 3: GOSUB 8000: REM LINES PLOTTED ON HIRES PAGE 1
7000  REM
        **INPUT ROUTINES**

7010  GET G$
7020  IF G$ = "I" THEN  PRINT BELL$;: GOSUB 7900
7030  IF G$ = "T" THEN  PRINT BELL$;: POKE  - 16303,0:MODE$ = "T/L"
7040  IF G$ = "L" THEN  PRINT BELL$;: POKE  - 16298,0: POKE  - 16304,0:
     MODE$ = "T/L"
7050  IF G$ = "H" THEN  PRINT BELL$;: POKE  - 16297,0: POKE  - 16304,0:
     MODE$ = "H"
7060  IF G$ = "M" THEN  PRINT BELL$;: POKE  - 16301,0
7070  IF G$ = "F" THEN  PRINT BELL$;: POKE  - 16302,0
7080  IF G$ = "Q" THEN 9000
7090  IF G$ = "D" THEN  PRINT BELL$;: GOSUB 7400
7100  IF G$ = "E" THEN  PRINT BELL$;: GOSUB 7500
7110  IF G$ = "C" OR G$ = "X" OR G$ = "O" THEN  PRINT BELL$;: GOSUB 760
     0
7120  GOTO 7010
7400  REM
        **DISPLAY**

7410  GET G$
7420  IF G$ = "1" THEN  POKE  - 16300,0: PRINT BELL$;: RETURN
7430  IF G$ = "2" THEN  POKE  - 16299,0: PRINT BELL$;: RETURN
7440  POP : GOTO 7030
7500  REM
```

```
        **ERASE**

7510  GOSUB 7800
7520  POKE 781,PLOC
7530  POKE 790,169:POKE 791,0
7540 SIZE = 32
7550  IF MODE$ = "T/L" THEN
      SIZE = 4
  **COPY, XCOPY, OR OVERLAY**
7560  POKE 785,SIZE
7570  CALL 768
7580  RETURN
7600  REM

7610 SIZE = 32
7620  IF MODE$ = "T/L" THEN
      SIZE = 4
7630  POKE 785,SIZE
7640 A1 = 177:A2 = 6
7650  IF G$ = "X" THEN A1 =
      81:A2 = 8
7660  IF G$ = "O" THEN A1 =
      17:A2 = 8
7670  GOSUB 7800
7680  POKE 773,PLOC
7690  GOSUB 7800
7700  POKE 781,PLOC
7710  POKE 790,A1
7720  POKE 791,A2
7730  CALL 768
7740  RETURN
7800  REM
        **PAGE SELECTOR**

7810  GET G$
7820 PAGE =  ASC (G$) - 48
7830 MAX = 5:MULT = 32
7840  IF MODE$ = "T" OR MODE$ = "T/L" THEN MAX = 3:MULT = 4
7850  IF PAGE < 1 OR PAGE > MAX THEN  POP : POP : GOTO 7030
7860 PLOC = PAGE * MULT: REM  PLOC IS PAGE LOCATION
7870  PRINT BELL$;: RETURN
7900  REM
        **GET INSTRUCTIONS**

7910 MODE$ = "T/L"
7920  POKE 773,16: POKE 781,4: POKE 785,4: POKE 790,177: POKE 791,6: CALL
     768
7930  POKE  - 16303,0: POKE  - 16300,0
7940  RETURN
8000  REM
        **PLOT RANDOM LINES**

8010  POKE 230,PAGE * 32
8020  HPLOT 280 *  RND (1),192 *  RND (1)
8030  FOR LINE = 1 TO 15
8040 X = 280 *  RND (1):Y = 192 *  RND (1)
8050  HPLOT  TO X,Y
8060  NEXT
8070  RETURN
9000  REM
        **REBOOT**

9010  PR# 6
```



Atari Data Management/ Database System

An Atari Database With Application Generation Features

Ronald Marcuse
Freehold, NJ

My initial excursion into the world of microcomputers began several months ago, timed to coincide with the arrival of the carton containing my Atari 800. It took perhaps a month for my data processing (meaning file processing) background to emerge from the myraid of games that I was coding and playing on the Atari, but this was inevitable. Don't misunderstand me, the games are fun, but I personally wanted more from the computer than just zapping Zylon Raiders or hitting a quarter-inch baseball across a 25 inch screen.

Luckily, a second carton arrived during this transition period — the 810 Disk Drive, complete with enough blank disks to store anything that my imagination conjured up. The cassette recorder, which had satisfied the storage needs of my embryonic stage, was too limited in I/O abilities for a confirmed file-processor.

In the ensuing month, information processing systems materialized in all shapes and sizes, covering such important data as telephone/addresses, appointments, and the "Star Wars" figure collection of my offspring (which turned out to be the largest of my numerous files)! When I saw myself coding a catalog system to keep track of all the other catalog/ information systems, I vowed to find a better way.

This leads us to the subject of the article — an Atari Data Management/Database system. There is a great deal of similarity between different systems that are designed to track different data; in fact, the similarities usually far outweigh the differences. The variance in file/record attributes does not require a markedly different approach in getting to and from the storage medium. If one were to store these file/record attributes in, say, a Data Dictionary, one could use the Dictionary to supply the parameters to drive generalized file/screen/ printer IO routines. One need only specify the attributes to generate the application. An information system needed to track paper clips could be implemented in minutes. That, in a nutshell, is the concept.

Converting To Microsoft BASIC

Before we get into a discussion of the software itself, a word about converting this program to the "other" forms of BASIC (e.g. MICROSOFT). Atari Basic has an intrinsic weakness in its handling of string variables as compared to Microsoft and others of that ilk. The inability to dimension a string array as well as the lack of the concatenation flexibility of LEFT$, MID$, and RIGHT$ has, if anything, caused additional complexity in the software. Sub-stringing on the Atari is of the form A$(B,C) where B and C are the starting and ending points of the stated (and DIMensioned) string A$. There are also numerous GOSUB NNNN + I's in the programming to allow retrieval of a particular string where A$(I) would have been much simpler.

The selection of file names would depend on the environment at which the conversion is aimed. Atari's requirement is of the form "Dn:FILENAME.EXT" where n is the drive number (1-4), FILENAME is a maximum of eight characters and the optional EXTender is limited to three. Within this particular effort, the Database files are generically formed as "D:filename.DB". The DOS functions represented by the various XIO commands are reproduced in Table 3. The Atari's TRAP statements are a mechanism to redirect program control during an error that would otherwise cause an abnormal termination.

The discussion of the Data Management/ Database software can best be handled by neatly dividing it up into its three main functions: 1. Data Dictionary; 2. File Management; and 3. Soft Utility. You flowcharters out there may find the diagram in Figure 1 interesting. A primary option menu, located in lines 100-220 of the program listing, controls the flow into and out of the three main functions. Note that the sort utility is *not* a resident

*CYBERSOFT is a registered trademark of Cybersoft Inc.

IF YOU OWN ATARI TAKE A CLOSER LOOK.

Introducing CYBERSOFT™ custom ATARI® programs. The Grand Masters of programming at a mail order price.

Now that you own ATARI®, the finest, most versatile, most powerful home computer on the market, you should have the software to make your ATARI sit up and really function.

We have the "best of the best" programs from the Grand Masters of micro-computer software: Automated Simulations, Crystal, Swifty Software, Adventure International, Datasoft, Personal Software, and Dynacomp. All sold by mail. And, with the CYBERSOFT ironclad guarantee*.

Space Chase *(Swifty Software) 16K* Nifty eye-hand coordination tests... If you clear one galaxy of "planets" and avoid destruction at the hands of an alien ship you are rewarded by an attack by two aliens and then three. Game keeps score and displays high score. Colorful graphics and good playing. Cassette $14.95

Fastgammon™ *(Quality Software) 8K* The best of the computer backgammon games. You can repeat games with the same dice rolls to try different strategies. You roll dice or let computer roll for you. This machine language program challenges experienced and beginners alike. Underutilizes ATARI® graphics but playing quality more than makes up. Cassette $19.95

Invasion Orion™ *(Automated Simulations) 32K* You are the fleet commander pitted against the Klaatu somewhere in space. You play the leading role in a number of carefully crafted scenarios. This game takes more skill than luck in order to be successful. The documentation is superb. Create your own senarios. Excellent graphics, good sound. Cassette $24.95

Rescue at Rigel *(Automated Simulations) 32K* Ten people from the Orion colony at Ultima Thule (including your favorite female) have been captured. Your mother ship must leave in 60 minutes. As "Sudden" Smith, you must find and release them, but it won't be easy in the maze of corridors, rooms and levels. Oh, by the way, the place is crawling with bad guys. Cassette $29.95

Sands of Mars *(Crystal) 40K* This is the one gamesters are all talking about. A two diskette game, with diskette one putting you in command of the "Starship Herman". You must stock her, select her crew, command the launch and negotiate your way through the hazards of the Universe. Diskette two holds your adventures on the Sands of Mars. The game is saveable. Two Diskettes $39.95

Star Trek 3.5 *(Adventure International) 24K* Lots of color, lots of action as you hyperwarp through the quadrants in your search for Klingons. This program will satisfy the "trekkie" in us all. Cassette $14.95

Text Wizard™ *(Datasoft) 32K* ATARI owners have waited a long time for a high quality, full blown word processor. Text Wizard was well worth the wait. This easy-to-use and comparatively inexpensive program has received raves from user groups across the country. This program fully utilizes the unique ATARI keyboard and ease of cursor movement. Justifies right, left and center. Pagination. Word search and substitute. Diskette and ATARI 825, Centronics 737 or Epson MX-80 printer. $99.95

Adventureland *(Adventure International) 24K* Designed for the novice at adventuring and a great way to get hooked on the world of adventure. As you make your way through an enchanted world you will encounter wild animals, spell-ridden critters and more. Your task is to avoid the perils as you try to collect the 13 lost treasures. Cassette $14.95

Mission Impossible *(Adventure International) 24K* Another adventure from the great Scott Adams. Beginning "Good morning, your mission is...", your task is to save a nuclear reactor from destruction. No magic help in this one. Addiction is possible. Cassette $14.95

Bridge 2.0 *(Dynacomp) 24K* Now you can practice your bridge at home before you have to embarrass yourself at the neighbors. Bridge 2.0 bids (Goren) and plays both contract and duplicate bridge. It even doubles if you get carried away with your bidding. No graphics, but clearly arranged format makes playing easy. Cassette $17.95

Visicalc™ *(Personal Software) 32K* If you use your ATARI for any business purpose or for your personal finances, you will want Visicalc. It probably sold more computers than any program other than Star Raiders. Visicalc creates a very large accountants ledger pad upon which you write with your ATARI. The fundamental difference is, if you change any figure(s) the balance of the figures are automatically adjusted. Extensive documentation outlines the full features of this fantastic business tool. Diskette $199.95

Nominoes Jigsaw *(Dynacomp) 24K* A 60 piece jigsaw puzzle you complete on your screen. Three levels of difficulty to confound you. Scoring is a function of how many tries and how difficult the puzzle. Cassette $17.95

*CYBERSOFT Ironclad Guarantee.

We fully guarantee the software we sell. If for any reason, you are not satisfied with your purchase, return the program within 30 days in its original packaging. We'll refund your money, no questions asked.

Our order system is as accurate and fast as our software. Just dial

1-800-247-2000 EXT. 505

(In Iowa 1-800-362-2133, extension 505). Or, you can write to CYBERSOFT at P.O. Box 505 Bellevue, Washington 98009. VISA and MasterCard gladly accepted.

Include $2.00 handling charge per order. Washington residents add 5.4% sales tax.

cybersoft

module and is called in by executing a RUN "D:DMSSORT" statement. This was done purposely to allow the package to run on an Atari configured with 24K RAM and a single disk drive. Those lucky (or rich?) enough to own 32K+ can easily incorporate the sort into the main program by following the procedures at the end of the article. Generally speaking, the File Management routines occupy lines 500-4600 while the Data Dictionary runs from 6000-9320, though several subroutines are shared by both.

The Dictionary

The Data Dictionary function, as the front-seat driver of the entire vehicle, deserves clarification first. Its primary function is to create and monitor the individual dictionary records containing the attributes of the database on that particular disk. Within this scheme, the ADD logic resides in lines 8000-8630, the INDEX in 6500-6600, the INQUIRY or LIST in 7000-7080, and the DELETE from 7500-7640. Subroutines shared by one or more of these modules are: PAUSE 6300; CREATE FILESPEC 6310-6320; PRINT DIC RECORD 8750-8870; LOAD VARIABLES FROM DIC RECORD 9000-9050; and READ DIC RECORD 9200-9320. Additionally, located from 9801-9846 are short subroutines of the variety GOSUB NNNN+I utilized in moving headers and data elements to and from larger strings. The format of the dictionary record (REC$) and other variables used are listed in Tables 1 and 2.

In operation, when creating a new application/database, one would specify a title (25 char. max), data set name (8 char. max) and the number of data elements (6 max) within the proposed record. A loop (lines 8110-8200) would step you through the entering of the heading (12 char.); size (30 max); and editing requirements of each data element. Editing criteria of N–Numeric; D–Date; and $–Dollar may be requested. A nominal limit of six data elements with 30 characters maximum for any one element and a total record length of 120 bytes has been imposed for no other reason than I happened to have liked those numbers on that day. If the Data Dictionary has *not* been initialized on the disk, you will be prompted during the add routine for permission to create it (lines 8610-8630).

There is a conspicuous absence of any UPDATE functions on the data dictionary menu. This was intentional. Changing the attributes of a given data base *after* it has been created and fed huge mounds of raw data may be detrimental to your health. (Picture the File Manager going after a 40 byte record that it now believes is 80 bytes long). If anyone out there adds an update here, please call me after the smoke clears.

The Manager

The File Management system, with its menu residing in lines (500-670), requires that the operator first select one of the databases on that disk (lines 750-892), with an index available by entering a single character of "I". The selection process reads the data dictionary file and, upon finding your request (file name is the key), moves the dictionary record into the applicable variables. Once loaded, these variables control the execution of the other File management modules. These are: ADD RECORD (lines 1000-1200); LIST RECORD/INQUIRY (2000-2440); UPDATE RECORD (3000-3310). Note that typing an "E" on any function menu will return program control to the module one step higher in the network. The variables used in these procedures are listed in Tables 1 and 2. There is a SEARCH procedure generated in lines 4000-4210 that can be utilized by both the INQUIRY/LIST and UPDATE functions. One need only specify the field number and then enter the value of the characters to be used. By entering fewer characters than the size of the field, one can perform a generic search, extracting, in turn, all records that satisfy those requirements.

The previously mentioned input editing is performed in lines 4400-4590. The selection of output medium (screen or printer) is done in lines 680-740. ASCII control characters (for an EPSON printer), based on the attributes of the resident Database, are generated and sent to the device thereby setting character size and tabs. The EPSON MX-80 has software selectable print lines of from 40 to 132 positions. Other printers without this feature may require a modified approach or a limitation in your record size.

The Sort

The SORT routine (Program 2) is an example of the "selection and exchange" variety. Logic similar to that contained in the database manager allows you to select the file to sort or list the index of that particular disk (lines 4000-4340). The choice of sort key and whether "ascending" or "descending" occurs in lines 5000-5070. The file is input and stored in the DIMensioned string X$ in lines 5100-5160. The variables I, J, and K are used as pointers during the loops through X$ as follows: I represents the sorted/not sorted boundary of X$, J is the current comparison to the previously selected lowest or highest value (DS$) and K is the location of this previous value.

An exchange between X$(K,K+L) and X$(I,I+L) occurs if an unsorted condition is detected on any loop. The sort terminates when the sorted boundary is equal to the size of X$. X$ is then written to the disk, the original file is deleted and the new file is renamed.

OK, at this point it would probably be beneficial for us to single-step through the code, but I think that we have neither the time nor the space to accomplish this feat. But we can and will examine at least one aspect of the structure, notably: the interaction of the subroutines that perform the input and handling of an existing Data Dictionary record. Entry to the READ DICTIONARY routine (9200-9310) is from several possible locations: LIST INDEX, both FMS and DIC (via line 6500); DICTIONARY INQUIRY/LIST (line 7080); and SELECT DATABASE (line 800). Two variables (loaded by the calling module) enable this one subroutine to perform several functions. "T" represents the "type" of request (0-index, 1-loop through to find and load data dic. record with key equal to value in FILD$,2-list single data dic. record with key FILD$, 9-browse through all data dic. records. The variable "R" is utilized to return control back to a specific line number in the event of a DISK I/O error being TRAPped to line 9500.

As the program executes each line within the routine, program control is modified based on the value of "T" during that call. Options include loading the full record into all of the Dictionary variables (9000-9050), listing the Dictionary record to the screen (8750-8870), chatting with the operator (9280-9290) and a "quickie" index. The subroutines mentioned above, as well as the balance of the program, work in a similar manner.

Does It All Work?

How well does this all work? Not too bad (would you believe me if I said that it was perfect)? There is one "bug" in Atari's DOS I that has forced me to process updates in a less efficient style. This problem centers around the inability to "rewrite" a record using the NOTE and POINT comands. (These are software pointers that allow random access to a disk file.) The optimum procedure for updating a disk record would be inputting the record, updating the data, and then returning the record to the same location in the file.

Unfortunately, an error in the DOS close routine causes a bad link on the file with a resultant file number mismatch and the loss of all sectors located after the rewritten data. This has been fixed in DOS II. The way around this problem involves rewriting the entire file as a "temporary" data set, stopping when you come to the record to be updated to post the changes, and then deleting the original and renaming the temporary. A little on the slow side, but it *does* work. Another enhancement planned is the inclusion of a sort routine in machine language. The "selection/exchange" sort is fairly quick, but one written in machine language (probably a *bubble* type) would run like a jackrabbit. Oh well, is a program ever really *finished*?

For those 32 + K's out there who must have the sort routine within the main program, move the sort call (lines 150 and 200) to 585 and 655, changing the RUN to a GOTO 5000. Add SORT lines 5000-5360 to the main program, changing line 5360 to GOTO 500. Also, DIMension variable X$ with a size of 8000, or as much as you can spare. The rest of the code can be put into the round file (and I don't mean a floppy). Keeping the sort separate gives you the ability to work with larger files, so think first before you merge the two together.

Typing The Programs In

Both program listings contain unprintable ASCII characters used for screen and printer control. I have taken the liberty of substituting other characters (enclosed by [ ]) in their place. A glance at Table 4 before you start keying may save you much aggravation later. Additionally, the lower case phrases in the programs should be typed as *upper case inverse video*. One final note: because the two programs call each other by name, you must SAVE the two as "D:DMSDB" and "D:DMSSORT".

Table 1. Data Dictionary Record (REC$)

Variable Name	Pos. Within REC$	Description
FILD$	1-8	Application File Name
APP$	9-33	Application Title
DL(1)		
:	34-45	Lengths of 6 (max) Data Elements
:		Within Record (2 char ea.)
DL(6)		
HD1$		
:	46-117	Heading Titles (12 char)
:		
HD6$		
DE(1)		Editing Criteria for Elements
:	118-123	0 = Alphanumeric 1 = Numeric (N)
:		2 = Date (D) 3 = Dollar ($)
DE(6)		
	124	Delimiter ("*")

Table 2. Other DMS/DB Variables

Name	Size	Description
RL	(var)	Application Data Set Record Length
NE	(var)	Number of Data Elements
DM$	8	DMS/DB Filespec (D:DMS.DB)
FIL$	14	Application Filespec (D:filename.DB)
I$	1	Operator Response to Questions
IN$	31	General Input of Data and Temp Stor
SF		
SL		
SS	(var)	Used by Search Function
SE		
SV$		
FD1$		
:	(var)	Application Data Elements
FD6$		
T	—	Tran Type Passed to I/O Routines
EOF	—	End of File Counter
I,J,K,L,N	—	Temp Stor for Looping, Length, etc.
ERR	—	Input Data Error Flag
R	—	Error Message Return Line #
P	—	Printer/Screen Indicator (1 = Print)
S$,X$,P$,N$	—	Messages, Prompts

Table 3. Atari DOS I XIO Commmands

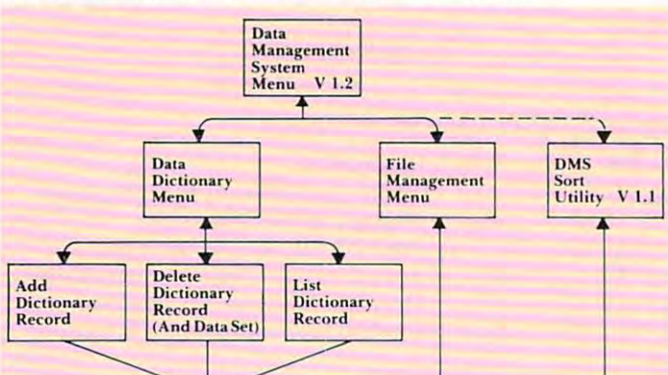
Command #	Description
XIO 32	Rename (XIO 32,#n,0,0,"D:oldname, newname")
33	Delete
35	Lock File
36	Unlock File

Note: The General Format is XIO nn,#n,0,0, "D:filename" where nn = XIO cmnd no. and n = IO control block

Table 4. Control Characters (Atari and EPSON MX-80 Printer)

Symbol	ASCII Val. (Dec)	Key Sequence (Atari)	Description
[A]	0	CON; , (comma)	Null; End of Tab Set Seq
[B]	9	CON; I	Horizontal Tab
[C]	11	CON; K	Vertical Tab
[D]	12	CON; L	Form Feed
[E]	14	CON; N	Print Double Width Characters
[F]	15	CON; O	Print Condensed Characters
[G]	18	CON; R	Cancel Condensed Mode
[H]	20	CON; T	Cancel Double Width Mode
[I]	27,68	ESC;D	Set Tab (followed by Tab Positions and NULL Char)
[J]	27,253	ESC;CON;2	Console Bell
[K]	27,125	ESC;CON;CLR	Clear Screen
[L]	27,29	ESC;CON; =	Move Cursor Down
[M]	27,158	ESC;CON;TAB	Clear Tab (screen)
[N]	27,159	ESC;SHFT;TAB	Set Tab (screen)
[O]	27,127	ESC;TAB	Tab (screen)

Figure 1. Data Management/Database System



Program 1.

```
10 REM * DMS - DATABASE PROTOTYPE    VER
 1.2  *
20 REM * 02/19/81  RONALD MARCUSE, FREEH
OLD NJ *
40 POKE 82,0:POKE 83,39:? "|{=}|":GOTO 1
00
50 DIM APP$(26),FIL$(14),FILD$(9),HD1$(1
2),HD2$(12),HD3$(12),HD4$(12),HD5$(12)
60 DIM HD6$(12),I$(1),REC$(132),IN$(31),
DM$(8),DL(6),DE(6)
65 DIM S$(15),X$(5),P$(24),N$(16):N$="|R
ECORD NOT FOUND|":DM$="D:DMS.DB"
70 P$="PRESS RETURN TO CONTINUE":S$="{DO
WN}SELECT OPTION:":X$="|ERROR|"
75 FOR I=1 TO 6:DL(I)=0:DE(I)=0:NEXT I
80 FOR I=1 TO 132:REC$(I,I)=" ":NEXT I:R
ETURN
100 GRAPHICS 0:? "{DOWN}  DATA MANAGEMEN
100 GRAPHICS 0:? "{DOWN}  DATA MANAGEMEN
T SYSTEM  - VER 1.2":CLR :DIM I$(1)
110 ? "{DOWN}      PRIMARY OPTION MENU":P
OKE 16,64:POKE 53774,64
120 ? "{DOWN} D - DATA DICTIONARY FUNCTI
ONS":? "    (DEFINE NEW APPLICATION & DA
TA)"
130 ? "{DOWN} R - EXEC FILE MANAGEMENT S
YSTEM":? "    (ACCESS EXISTING DATA BASE
)"
150 ? "{DOWN} S - CALL DMS SORT UTILITY"
:? "{DOWN} E - END (TERMINATE DMS)"
160 ? :? "SELECT OPTION:";:INPUT I$:IF I
$="R" THEN 500
180 IF I$="D" THEN 6000
200 IF I$="S" THEN RUN "D:DMSSORT"
210 IF I$="E" THEN GRAPHICS 0:END
220 ? "|INVALID|":GOTO 160
490 ? "{DOWN}SELECT DATA SET 1ST":GOSUB
6300
500 ? "{CLEAR}{DOWN}   DMS FILE MANAGEME
NT MENU"
520 IF NE<>0 THEN ? :? ,APP$:IF P=1 THEN
 ? ,"|PRINTER SELECTED|"
540 ? "{DOWN} S - SELECT DATA BASE":? "
A - ADD"
580 ? " L - LIST / INQUIRY":? " U - UPDA
TE"
590 ? " P - PRINTER (SCREEN DEFAULT)":?
" E - END (RETURN TO DMS MENU)"
600 ? :? "SELECT OPTION:";:INPUT I$:IF I
$="E" THEN 100
620 IF I$="S" THEN 750
630 IF I$="A" THEN 1000
640 IF I$="L" THEN 2000
650 IF I$="U" THEN 3000
660 IF I$="P" THEN 680
670 ? "|INVALID|":GOTO 600
680 IF NE=0 THEN 490
```

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

Compatible with Atari DOS. 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 microcomputer systems at a competitive price." **INFOWORLD**

DATA PERFECT T.M. LJK

APPLE & ATARI
DATA BASE MANAGEMENT **$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, DISASSEMBLER, 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 — APPLE & 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 $34.95

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

*Trademarks of: Apple Computer — Atari Computer — Epson America — Hayes Microcomputers — Personal Software — Videx — Bit 3 Inc. — M&R Ent. — Advanced Logic Systems — Vista Computers

```
690 ? :? "TYPE: P- PRINTER S- SCREEN":IN
PUT I$:IF I$<>"P" THEN P=0:GOTO 500
700 P=1:IN$="{R}{ESC}D      {,}":IF RL>55
 OR NE>4 THEN IN$(1,1)="{O}"
710 L=1:FOR I=2 TO NE:J=DL(I-1):IF J<12
THEN J=12
720 L=L+J+2:IN$(I+2,I+2)=CHR$(L):NEXT I
730 R=690:TRAP 9510:LPRINT IN$:TRAP 4000
0:GOSUB 80:GOTO 500
750 ? "{CLEAR}{DOWN}    DATA BASE SELECT
ION":CLR :GOSUB 50:R=760
760 ? "{DOWN}ENTER DATA SET NAME (I FOR
INDEX)":? :? P$;" (END)"
770 INPUT FILD$:IF LEN(FILD$)=0 THEN 500

780 IF LEN(FILD$)=1 AND FILD$="I" THEN G
OSUB 6500:GOTO 760
800 T=1:GOSUB 9200:IF EOF=0 THEN 500
820 ? "{DOWN}DATA SET LOADED":GOSUB 6300


830 RL=1:FOR N=1 TO NE:GOSUB 880+N:RL=RL
+DL(N):NEXT N
850 DIM DS$(RL),SV$(30):FOR I=1 TO RL:DS
$(I,I)=" ":NEXT I:GOTO 500
881 DIM FD1$(DL(N)):RETURN
882 DIM FD2$(DL(N)):RETURN
883 DIM FD3$(DL(N)):RETURN
884 DIM FD4$(DL(N)):RETURN
885 DIM FD5$(DL(N)):RETURN
886 DIM FD6$(DL(N)):RETURN
1000 IF NE=0 THEN 490
1005 R=500:TRAP 9500:XIO 36,#2,0,0,FIL$:
OPEN #2,9,0,FIL$:TRAP 40000
1010 ? "{CLEAR}{DOWN}   ";APP$:? "{DOWN}
 TO ADD RECORD, ENTER:"
1040 FOR N=1 TO NE
1050 GOSUB 9820+N:IF DE(N)<>2 THEN ? " (
";DL(N);" CHAR MAX)":GOTO 1070
1060 ? " (6 CHAR - MMDDYY)"
1070 INPUT IN$:L=LEN(IN$):IF N=1 AND L=0
 THEN 1180
1075 GOSUB 4400:IF ERR<>0 THEN 1070
1080 GOSUB 9810+N:NEXT N:GOSUB 4000
1170 ? #2;DS$:? "{DOWN}TRANSACTION ACCEP
TED"
1180 ? "{DOWN}TYPE E FOR FMS MENU":? P$
1190 INPUT I$:IF I$<>"E" THEN 1010
1200 CLOSE #2:XIO 35,#2,0,0,FIL$:GOTO 50
0
2000 IF NE=0 THEN 490
2020 ? "{CLEAR}{DOWN}   ";APP$:? "{DOWN}
 INQUIRY / LIST":IF P=1 THEN ? ,,"{UP}!P
RINTER SELECTED!"
2030 ? "{DOWN} A - LIST ALL RECORDS":? "
 S - SEARCH ON KEY"
2050 ? " E - END (RETURN TO MENU)":T=9
2060 ? S$;:INPUT I$:IF I$="E" THEN 500
2080 IF I$="A" THEN 2300
2090 IF I$="S" THEN T=1:? "{CLEAR}":GOSU
B 4100:GOTO 2300
2100 ? X$:GOTO 2060
2300 I=0:EOF=0:R=2000:TRAP 9500:OPEN #2,
4,0,FIL$
2310 TRAP 2420:INPUT #2,DS$:TRAP 40000:I
F T>8 THEN 2330
2320 IF SV$<>DS$(SS,SE) THEN 2310
2330 EOF=EOF+1:IF P<>1 THEN 2400
2340 L=1:J=1:IF I>0 THEN 2370
2350 R=2440:TRAP 9510:LPRINT "{I}{N}";AP
P$;"{T}{K}"
2360 LPRINT HD1$;"{I}";HD2$;"{I}";HD3$;"
{I}";HD4$;"{I}";HD5$;"{I}";HD6$;"{K}"
2370 FOR N=1 TO NE:K=DL(N):IF K<12 THEN
K=12
2375 REC$(J,J+DL(N)-1)=DS$(L,L+DL(N)-1):
L=L+DL(N):J=J+K+2:NEXT N
2380 TRAP 9510:LPRINT REC$(1,J):I=I+1:IF
 I<55 THEN 2310
2390 I=0:LPRINT "{L}":GOTO 2310
2400 GOSUB 4300:? :? P$;" (E TO STOP)"
2410 INPUT I$:IF I$<>"E" THEN 2310
2420 IF EOF=0 THEN ? N$
2430 IF EOF>0 THEN ? "REC COUNT- ";EOF
2435 IF P=1 THEN LPRINT "{L}"
2440 CLOSE #2:GOSUB 6300:GOTO 2020
3000 IF NE=0 THEN 490
3020 ? "{CLEAR}{DOWN}   ";APP$:? "{DOWN}
 UPDATE / DELETE";:GOSUB 4100
3040 R=500:TRAP 9500:OPEN #2,4,0,FIL$:OP
EN #3,8,0,"D:TEMP":EOF=0
3060 TRAP 3250:INPUT #2,DS$:TRAP 40000
3080 IF SV$<>DS$(SS,SE) THEN ? #3;DS$:GO
TO 3060
3100 GOSUB 4300
3120 ? "{DOWN}ENTER: FIELD # TO UPDATE;
D TO DELETE":? "PRESS RETURN TO WRITE RE
C"
3130 EOF=EOF+1:INPUT I$:IF LEN(I$)=0 THE
N ? #3;DS$:GOTO 3060
3135 IF I$="D" THEN 3060
3140 TRAP 3120:N=VAL(I$):TRAP 40000:IF N
<1 OR N>NE THEN 3120
3150 ? "{DOWN}ENTER NEW ";:GOSUB 9820+N:
? " "

3170 INPUT IN$:L=LEN(IN$):GOSUB 4400:IF
ERR<>0 THEN 3150
3180 GOSUB 9810+N:GOSUB 4000:GOTO 3100
3250 FOR I=1 TO LEN(IN$):IN$(I,I)=" ":NE
XT I:IN$(1,7)="D:TEMP,"
3260 IN$(8,7+LEN(FIL$)-2)=FIL$(3)
3270 CLOSE #2:CLOSE #3:XIO 36,#2,0,0,FIL
$:XIO 33,#2,0,0,FIL$
3280 XIO 32,#3,0,0,IN$:XIO 35,#3,0,0,FIL
$
3290 IF EOF=0 THEN ? N$
3300 ? "{DOWN}MORE UPDATES? (Y OR N)":IN
```

```
PUT I$:IF I$="Y" THEN 3000
3310 GOTO 500
4000 FOR I=1 TO LEN(DS$):DS$(I,I)=" ":NE
XT I
4010 L=1:FOR N=1 TO NE:GOSUB 9840+N:DS$(
L,L+DL(N)-1)=IN$
4020 L=L+DL(N):NEXT N:DS$(RL)="%":RETURN

4100 ? " SELECT KEY:":? :FOR I=1 TO NE:?
 " ";I;" ";
4110 GOSUB 9820+I:? " ":NEXT I:? :IF SF=
0 THEN 4140
4120 ? "PRESS RETURN FOR KEY:":? " 1ST "
;SL;" POS OF ";
4130 GOSUB 9820+SF:? " (";SV$;")":? " ":
? " OR, ";
4140 ? "ENTER KEY FIELD #";
4160 TRAP 4200:INPUT SF:TRAP 40000:IF SF
<1 OR SF>NE THEN ? X$:GOTO 4140
4170 ? "ENTER VALUE OF ";:GOSUB 9820+SF:
? " (1-";DL(SF);")"
4180 INPUT SV$:SL=LEN(SV$):IF SL<1 OR SL
>DL(SF) THEN ? X$;" LEN":GOTO 4170
4190 SS=1:IF SF>1 THEN FOR I=2 TO SF:SS=
SS+DL(I-1):NEXT I
4200 IF SF=0 THEN ? X$:GOTO 4160
4210 SE=SS+SL-1:RETURN
4300 L=1:FOR N=1 TO NE:IN$=DS$(L,L+DL(N)
-1)
4310 GOSUB 9810+N:L=L+DL(N):NEXT N
4320 ? "{CLEAR}{DOWN}  ";APP$:? :FOR N=1
 TO NE
4330 ? N;" ";:GOSUB 9820+N:GOSUB 9840+N:
? " ";IN$:NEXT N:RETURN
4400 ERR=0:IF L<1 OR L>DL(N) THEN ERR=1
4410 ON DE(N) GOSUB 4500,4520,4560:IF ER
R<>0 THEN ? X$
4420 RETURN
4500 FOR I=1 TO L:J=ASC(IN$(I,I)):IF J<4
8 OR J>57 THEN ERR=1
4510 NEXT I:RETURN
4520 GOSUB 4500:IF L<>6 THEN ERR=1
4530 IF IN$(1,2)<"01" OR IN$(1,2)>"12" T
HEN ERR=1
4540 IF IN$(3,4)<"01" OR IN$(3,4)>"31" T
HEN ERR=1
4550 RETURN
4560 IF IN$(L-2,L-2)<>" " THEN ERR=1
4570 TRAP 4590:I=VAL(IN$(1,L)):TRAP 4000
0

4580 RETURN
4590 ERR=1:GOTO 4580
6000 ? "{CLEAR}{DOWN}   DMS  DATA DICTIO
NARY MENU":CLR :GOSUB 50
6050 ? "{DOWN} A - ADD NEW APPLICATION /
 DATA BASE"
6060 ? " I - LIST DICTIONARY INDEX":? "
L - LIST CURRENT DB DESCRIP"
6080 ? " D - DELETE DATA BASE":? " E - E
ND (RETURN TO DMS MENU)"
6100 ? :? S$;:INPUT I$
6120 IF I$="A" THEN 8000
6130 IF I$="I" THEN R=6000:GOSUB 6500:?
:? P$;:INPUT I$:GOTO 6000
6140 IF I$="L" THEN 7000
6150 IF I$="D" THEN 7500
6160 IF I$="E" THEN 100
6170 ? X$:GOTO 6100
6300 FOR I=1 TO 150:NEXT I:RETURN
6310 FOR L=1 TO LEN(FILD$):IF FILD$(L,L)
<>" " THEN NEXT L
6320 L=L-1:FIL$(1,2)="D:":FIL$(3,2+L)=FI
LD$(1,L):FIL$(3+L)=".DB":RETURN
6500 ? "{CLEAR}{DOWN}  DATA SET INDEX":?
 :T=0:GOSUB 9200:RETURN
7000 ? "{CLEAR}{DOWN}   DATA DICTIONARY
INQUIRY":? "{DOWN} A - ALL FILES":T=9
7020 ? " S - SINGLE FILE":? " E - END (R
ETURN TO MENU)"
7030 ? S$;:INPUT I$:IF I$="E" THEN 6000
7040 IF I$="A" THEN 7080
7050 IF I$="S" THEN ? "ENTER FILE NAME":
T=2:GOTO 7070
7060 ? X$:GOTO 7030
7070 INPUT FILD$:IF LEN(FILD$)=0 THEN 70
60
7080 R=7000:GOSUB 9200:GOTO 7000
7500 ? "{CLEAR}{DOWN} TO DELETE DICTIONA
RY ELEMENT AND":? " RELATED FILE, TYPE F
ILE NAME:"
7510 ? P$;" (CANCEL)":INPUT FILD$:IF LEN
(FILD$)=0 THEN 6000
7520 R=7500:TRAP 9500:OPEN #2,4,0,DM$:OP
EN #3,8,0,"D:TEMP":EOF=0
7530 TRAP 7600:INPUT #2,REC$:TRAP 40000
7540 IF FILD$<>REC$(1,LEN(FILD$)) THEN ?
 #3;REC$:GOTO 7530
7550 EOF=EOF+1:GOSUB 9000:GOSUB 8750
7560 ? "TYPE D TO DELETE";:INPUT I$:IF I
$<>"D" THEN ? "SAVED":? #3;REC$:GOTO 753
0

7570 ? "DELETED":XIO 36,#4,0,0,FIL$:XIO
33,#4,0,0,FIL$:GOTO 7530
7600 IF EOF=0 THEN ? N$
7610 CLOSE #2:CLOSE #3:XIO 36,#2,0,0,DM$
:XIO 33,#2,0,0,DM$
7620 XIO 32,#3,0,0,"D:TEMP,DMS.DB":XIO 3
5,#3,0,0,DM$
7640 GOTO 6000
8000 ? "{CLEAR}{DOWN}  ADD TO DATA DICTI
ONARY":? "{DOWN} ENTER APPLICATION NAME
(1 TO 25 CHAR)"
8010 INPUT APP$:L=LEN(APP$):IF L<1 OR L>
25 THEN ? X$:GOTO 8010
8020 ? "ENTER FILE NAME (1 TO 8 CHAR)":?
 "  FILESPEC WILL BE 'D:XXXXXXXX.DB'"
```

```
8030 INPUT FILD$:L=LEN(FILD$):IF L<1 OR
L>8 THEN ? X$:GOTO 8030
8040 IF FILD$(1,1)<"A" OR FILD$(1,1)>"Z"
 THEN ? X$:GOTO 8030
8060 GOSUB 6310
8080 ? "{DOWN} (MIN REC LEN=10, MAX=120)
":? "{DOWN}ENTER # OF DATA ELEMENTS IN R
EC (2-6)"
8090 TRAP 8080:INPUT NE:TRAP 40000
8100 IF NE<2 OR NE>6 THEN ? X$:GOTO 8080

8110 RL=0:FOR I=1 TO NE:? "{CLEAR}{DOWN}
  FOR DATA ELEMENT # ";I;" OF ";NE;". EN
TER:":?
8120 ? "HEADING (1-12)":INPUT IN$:L=LEN(
IN$):IF L<1 OR L>12 THEN ? X$:GOTO 8120
8125 GOSUB 9800+I
8130 ? "ELEMENT LENGTH (1 TO 30)":? "  (
TOT REC LEN IS ";RL;")"
8140 TRAP 8130:INPUT L:TRAP 40000:IF L<=
0 OR L>30 THEN ? X$:GOTO 8130
8150 DL(I)=L:RL=RL+L
8160 ? "EDITING? (N:NUMERIC, D:DATE, $:D
OLLAR)":? "RETURN TO SKIP"
8170 INPUT I$:DE(I)=0:IF I$="N" THEN DE(
I)=1
8180 IF I$="D" THEN DE(I)=2
8190 IF I$="$" THEN DE(I)=3
8200 NEXT I
8220 IF RL<10 OR RL>120 THEN ? "REC LEN=
";RL;"  ";X$:GOTO 8080
8300 REC$(1,8)=FILD$:REC$(9,33)=APP$
8320 FOR I=1 TO 6:REC$(32+I*2,33+I*2)=ST
R$(DL(I))
8340 REC$(117+I,117+I)=STR$(DE(I)):NEXT
I
8360 REC$(46,57)=HD1$:REC$(58,69)=HD2$:R
EC$(70,81)=HD3$
8380 REC$(82,93)=HD4$:REC$(94,105)=HD5$:
REC$(106,117)=HD6$:REC$(124)="%"
8390 GOSUB 8750:? "{DOWN}TYPE Y TO CREAT
E DATA BASE"
8400 INPUT I$:IF I$<>"Y" THEN 6000
8420 TRAP 8600:XIO 36,#2,0,0,DM$:OPEN #2
,9,0,DM$
8430 ? #2;REC$:CLOSE #2:XIO 35,#2,0,0,DM
$:TRAP 40000
8450 ? "DATA BASE CREATED"
8460 OPEN #3,8,0,FIL$:CLOSE #3:XIO 35,#3
,0,0,FIL$:GOTO 6000
8600 STATUS #2,ST:IF ST<>170 THEN R=8390
:GOTO 9500
8610 ? DM$;" NOT ON DISK":? "TYPE Y TO I
NITIALIZE"
8620 INPUT I$:IF I$<>"Y" THEN 6000
8630 OPEN #2,8,0,DM$:GOTO 8430
8750 ? "{CLEAR}{CLR TAB}{TAB}{CLR TAB}{T
AB}{CLR TAB}{TAB}{CLR TAB}{TAB}{CLR TAB}
{TAB}{CLR TAB}{DOWN}  DATA DICTIONARY RE
CORD"
8760 ? "{DOWN} FILE NAME - ";FILD$:? " A
PPLICATION - ";APP$
8780 ? "{DOWN} EL{SET TAB}EM #  {SET TAB
}HEADING      LE{SET TAB}NGTH  {SET TAB}
 EDIT?":?
8800 FOR I=1 TO NE:? "{TAB}";I;"{TAB}";:
GOSUB 9820+I:? "{TAB}";DL(I);"{TAB}";
8820 IF DE(I)=0 THEN ? " "
8830 IF DE(I)=1 THEN ? "NUMERIC"
8840 IF DE(I)=2 THEN ? "DATE"
8850 IF DE(I)=3 THEN ? "DOLLAR"
8870 NEXT I:? :? " RECORD LENGTH = ";RL:
RETURN
9000 FILD$=REC$(1,8):APP$=REC$(9,33)
9010 RL=0:NE=0:FOR I=1 TO 6
9020 DL(I)=VAL(REC$(32+I*2,33+I*2)):DE(I
)=VAL(REC$(117+I,117+I))
9030 IF DL(I)=0 THEN 9050
9040 NE=NE+1:RL=RL+DL(I):IN$=REC$(34+I*1
2,45+I*12):GOSUB 9800+I
9050 NEXT I:GOSUB 6310:RETURN
9200 TRAP 9500:OPEN #2,4,0,DM$:EOF=0
9210 TRAP 9300:INPUT #2,REC$:TRAP 40000:
IF T>8 THEN 9250
9220 IF T=0 THEN ? " ";REC$(1,8);" ";REC
$(9,33):GOTO 9210
9230 IF FILD$<>REC$(1,LEN(FILD$)) THEN 9
210
9250 EOF=EOF+1:GOSUB 9000:IF T>1 THEN GO
SUB 8750
9280 IF T>1 THEN ? :? P$:IF T>8 THEN ? "
 TYPE E TO END"
9290 IF T>1 THEN INPUT I$:IF T>8 AND I$<
>"E" THEN 9210
9300 CLOSE #2
9310 IF T>0 AND EOF=0 THEN ? N$:GOSUB 63
00
9320 RETURN
9500 STATUS #2,K:? "|{=}CHECK DISK DRIVE
|",X$;K:? P$;:CLOSE #2:INPUT I$:POP :GOT
O R
9510 STATUS #7,K:? "|{=}CHECK PRINTER|",
X$;K:? P$;:INPUT I$:GOTO R
9801 HD1$=IN$:RETURN
9802 HD2$=IN$:RETURN
9803 HD3$=IN$:RETURN
9804 HD4$=IN$:RETURN
9805 HD5$=IN$:RETURN
9806 HD6$=IN$:RETURN
9811 FD1$=IN$:RETURN
9812 FD2$=IN$:RETURN
9813 FD3$=IN$:RETURN
9814 FD4$=IN$:RETURN
9815 FD5$=IN$:RETURN
9816 FD6$=IN$:RETURN
9821 ? HD1$;:RETURN
```

```
9822 ? HD2$;:RETURN
9823 ? HD3$;:RETURN
9824 ? HD4$;:RETURN
9825 ? HD5$;:RETURN
9826 ? HD6$;:RETURN
9841 IN$=FD1$:RETURN
9842 IN$=FD2$:RETURN
9843 IN$=FD3$:RETURN
9844 IN$=FD4$:RETURN
9845 IN$=FD5$:RETURN
9846 IN$=FD6$:RETURN
```

Program 2.

```
11 REM **    DMS SORT UTILITY        **
12 REM **  RM  02/22/81   VER 1.2    **
20 ? "|{=}|":GRAPHICS 0:POKE 82,0:POKE 1
6,64:POKE 53774,64:GOTO 4000
50 DIM X$(8500),DS$(124),APP$(25),FIL$(1
4),FILD$(8),HD1$(12),HD2$(12)
60 DIM HD3$(12),HD4$(12),HD5$(12),HD6$(1
2),I$(1),IN$(12),DL(6)
70 FOR I=1 TO 6:DL(I)=0:NEXT I
80 RETURN
4000 CLR :? "{CLEAR}{DOWN}  DMS DATA BAS
E SORT":GOSUB 50
4010 ? :? " SELECT DATA SET:":? " (I - I
NDEX, D - DMS, E - END)"
4030 INPUT FILD$:L=LEN(FILD$):IF L=1 AND
 FILD$="D" THEN RUN "D:DMSDB"
4040 T=1:IF L=0 THEN 4010
4050 IF L=1 AND FILD$="E" THEN GRAPHICS
0:END
4060 IF L=1 AND FILD$="I" THEN ? "{CLEAR
}{DOWN} DATA SET INDEX":? :T=0
4200 R=4000:TRAP 9500:OPEN #2,4,0,"D:DMS
.DB":EOF=0
4210 TRAP 4300:INPUT #2,DS$:TRAP 40000
4220 IF T=0 THEN ? " ";DS$(1,8);" ";DS$(
9,33):GOTO 4210
4230 IF FILD$<>DS$(1,LEN(FILD$)) THEN 42
10
4240 FILD$=DS$(1,8):APP$=DS$(9,33)
4250 EOF=EOF+1:RL=1:NE=0:FOR I=1 TO 6
4260 DL(I)=VAL(DS$(32+I*2,33+I*2)):IF DL
(I)=0 THEN 4290
4280 NE=NE+1:RL=RL+DL(I):IN$=DS$(34+I*12
,45+I*12):GOSUB 9800+I
4290 NEXT I
4300 CLOSE #2:IF T=0 THEN 4010
4310 IF EOF=0 THEN ? "|NOT FOUND|":GOTO
4010
4330 FOR L=1 TO LEN(FILD$):IF FILD$(L,L)
<>" " THEN NEXT L
4340 L=L-1:FIL$(1,2)="D:":FIL$(3,2+L)=FI
LD$(1,L):FIL$(3+L)=".DB"
5000 IF NE=0 THEN 490
5010 ? "{CLEAR}{DOWN}  ";APP$:? "{DOWN}
SELECT SORT KEY:":? :FOR I=1 TO NE
5020 ? I;" ";:GOSUB 9820+I:? :NEXT I:? "
PRESS RETURN TO CANCEL";
5030 TRAP 4000:INPUT SF:TRAP 40000:IF SF
<1 OR SF>NE THEN ? "|INVALID|":GOTO 5030

5040 T=2:? "ASCENDING OR DESCENDING? (A
OR D)";:INPUT I$:IF I$="D" THEN T=1
5050 SL=DL(SF):? "LOADING ";FIL$:? "SORT
 ON ";SL;" CHAR OF ";:GOSUB 9820+SF:?
5060 SS=1:IF SF>1 THEN FOR I=2 TO SF:SS=
SS+DL(I-1):NEXT I
5070 SE=SS+SL-1:R=5000:EOF=0:N=1:L=RL-1
5100 TRAP 9500:OPEN #2,4,0,FIL$
5110 TRAP 5150:INPUT #2,DS$:TRAP 40000
5120 EOF=EOF+1:X$(N,N+L)=DS$:N=N+RL:GOTO
 5110
5150 CLOSE #2:? "{DOWN}REC LOADED= ";EOF
;", RAM (BYTES)= ";N-1:? "BEGIN SORT"
5160 I=1:N=N-RL
5200 K=0:DS$=X$(I,I+L):J=I+RL
5210 ON T GOTO 5220,5240
5220 IF X$(J+SS-1,J+SE-1)>DS$(SS,SE) THE
N DS$=X$(J,J+L):K=J
5230 GOTO 5250
5240 IF X$(J+SS-1,J+SE-1)<DS$(SS,SE) THE
N DS$=X$(J,J+L):K=J
5250 J=J+RL:IF J<=N THEN 5210
5280 IF K<>0 THEN X$(K,K+L)=X$(I,I+L):X$
(I,I+L)=DS$
5290 I=I+RL:IF I<N THEN 5200
5300 ? "|{=}|SORT COMPLETED"
5320 TRAP 9500:XIO 36,#2,0,0,FIL$:OPEN #
2,8,0,FIL$:TRAP 40000:EOF=0
5330 FOR I=1 TO N STEP RL:? #2;X$(I,I+L)
:EOF=EOF+1:NEXT I
5340 CLOSE #2:XIO 35,#2,0,0,FIL$:? "REC
COUNT= ";EOF
5350 ? "SORT THIS FILE AGAIN? (Y OR N)";
:INPUT I$:IF I$="Y" THEN 5010
5360 GOTO 4000
9500 STATUS #2,K:? "|{=}CHECK DISK DRIVE
|","|ERROR| ";K:CLOSE #2:? "PRESS ENTER"
;:INPUT I$:GOTO R
9801 HD1$=IN$:RETURN
9802 HD2$=IN$:RETURN
9803 HD3$=IN$:RETURN
9804 HD4$=IN$:RETURN
9805 HD5$=IN$:RETURN
9806 HD6$=IN$:RETURN
9821 ? HD1$;:RETURN
9822 ? HD2$;:RETURN
9823 ? HD3$;:RETURN
9824 ? HD4$;:RETURN
9825 ? HD5$;:RETURN
9826 ? HD6$;:RETURN
```

©

A Program For Writing Programs On The Atari 400/800 Computers

David D. Thornburg
Innovision
Los Altos, CA 94022

If you write a lot of programs for your computer, you may have developed a few building blocks which you like to use over again in different applications. While it is a pretty easy task to bring these building blocks into a new program unchanged, you probably need to personalize these program segments for most applications.

Have you ever wished that there were a simple way to get the computer to write customized program segments for you?

Depending on how fancy you want to get, a "Program Writing Program" (PWP) can be made to construct a fairly detailed set of BASIC statements on the basis of your answers to a series of questions. To illustrate how such programs work, I will describe a simple PWP for the Atari computers.

PWP1 — A BASIC Example

This program creates a series of DATA statements which incorporate a word or data list entered from the keyboard. This is a very handy utility, since the user only types the raw data to be used (word lists, numbers, etc.), and the computer generates the proper BASIC program lines with line numbers, the word DATA, and all the field separators. As the DATA statements are composed, they are written to a file on a disk or cassette for retrieval later.

As you can see from the listing below, this program is quite simple.

```
10 DIM A$(40)
20 PRINT "PWP1 — A DATA STATEMENT
   WRITER"
30 PRINT "ENTER YOUR DATA AND PRESS
   RETURN"
40 PRINT "ENTER *** WHEN DONE"
50 OPEN #1,8,0,"D1:WORDS.LST"
60 C=30000
70 PRINT #1;C;"DATA";
80 FOR I=1 TO 8
90 INPUT A$
100 IF A$="***" THEN GOTO 170
110 PRINT #1;A$;
120 IF I<>8 THEN PRINT #1;",";
130 NEXT I
140 PRINT #1
150 C=C+10
160 GOTO 70
170 PRINT #1;A$
180 CLOSE #1
190 END
```

In line 10 we define a string variable which will receive text from the keyboard. After printing the instructions (lines 20-40), the computer opens the file WORDS.LST on disk drive 1. (If you use the cassette recorder instead, you must modify the program as shown below.) Line 60 sets the starting line number to 30000. The beginning of the DATA statement is printed to the file by line 70. This consists of the line number (stored in variable C) followed by the word DATA. Next, a loop using lines 80-130 accepts data from the keyboard and appends up to eight entries after the word DATA, placing commas (,) between them. In line 100 the program checks to see if you are done entering data, at which time the program prints the three asterisks (line 170) and closes the data file (line 180). Otherwise, if there is more data to be entered, the new data is printed to the file and a comma is placed after the entry (line 120) unless it is the last entry in the line. Once eight entries have been made, the DATA statement is completed with a carriage return (by using the PRINT statement in line 140). The line number is increased by 10 (line 150) and the next DATA statement is started (line 160).

...a 'Program Writing Program' (PWP) can be made to construct a fairly detailed set of BASIC statements on the basis of your answers to a series of questions.

If you are using a cassette instead of a disk drive to store the new program, you must make the following changes:

```
50 OPEN #1,8,0,"C:"
52 FOR I=1 TO 64: PRINT #1;"R.";: NEXT I
54 PRINT #1
```

In line 50, the file for the new program is opened on the cassette recorder. Lines 52 and 54 print 128 characters to this file to make sure that there is no gap between the tape header and the first 128 byte block of data. By printing the characters "R." 64

LE SHHHSTICK™

Finally! The ultimate goal in microcomputer hardware, achieved.

A joystick with such remarkable realism and control that even the thought of other joysticks virtually vanish.

Inside Le Stick™ are motion detectors which will maneuver your sights in any direction you want with simple one handed movements. The large red push button on top provides a quick and accurate firing mechanism for better response time.

From Star Raiders to graphic draw programs — Le Stick™ eliminates all the frustrations you experience with the conventional two handed joysticks or keyboards.

The joystick of the future. See and test it at your Datasoft™ dealer today, or contact us for more information.

LE STICK $39.95
FOR ATARI 400/800, ATARI VIDEO GAME, COMMODORE VIC AND THE TRS-80 COLOR COMPUTER SYSTEMS

Datasoft Inc.™
COMPUTER SOFTWARE
19519 Business Center Dr., Northridge, CA 91324
(213)701-5161

Character Generator
for Atari 400/800 Computers

This new program for the Atari Computer enables you to create high resolution graphics in the text mode. Redefines a character set virtually any way you want. Ready for whatever language or graphics set you can devise — from ancient Japanese to Modern Russian. **(Diskette) $19.95 / (Cassette) $15.95.**

Mailing List
for Atari 800 Computer

Business application software for the Atari Computer. A versatile disc based system. An efficient micro format equal to those used for many large computers. Sorts by Zip Codes, Names, and stores special data for your personal use. **(Diskette) $24.95.**

Order through your local software dealer, or send check or money order plus $2.00 shipping and handling to Datasoft.

times, we generate a meaningless REM statement which will be ignored automatically when the new program is loaded back into the computer.

Running PWP1

When you RUN this program, you will hear the disk drive turn on and off as the data file is created. If you use the cassette recorder, be sure a blank tape is inserted and press the PLAY and RECORD buttons when you hear the two "buzzes" from the computer. Once you have done this, press the RETURN key and notice that the tape is starting to move.

Once the program is ready to accept data, you will see a question mark on the screen. At this point, start entering your data, pressing RETURN after each entry. Entries of a list of words might look like this:

```
?THIS
?IS
?A
?LIST
?OF
.
.
.
?COMPUTER
?***
```

Once the three asterisks are entered, the data file will be closed, and the program is finished.

Be sure to save PWP1 on disk or tape because we need to erase it is order to look at the new program.

Loading The New Program

If you try to load your new program using LOAD "D:WORDS.LST", or CLOAD (from tape), you will have a most unpleasant surprise. BASIC programs which are SAVE'd or CSAVE'd are stored in a compact "tokenized" format. The program written by PWP1 is saved in the form of text strings, just as if it had been entered from the keyboard. To bring this program into the computer, we need to use the ENTER command. Be sure to type NEW and press RETURN before typing ENTER, otherwise the new program will be *added* to the program already resident in the computer. You should either type ENTER "D:WORDS.LST" or ENTER "C:", depending on whether you used a disk or a cassette. If you used a disk, the program will load automatically. If you used the cassette, you should press the PLAY button when you hear the "buzz" and press RETURN again on the computer. The cassette version will print READY twice before it is done (the first READY shows up when the dummy REM statement has been received). If you now type LIST you should see something like this:

```
30000 DATA THIS,IS,A,LIST,OF,WORDS,WHICH,
      WILL
30010 DATA BE,PLACED,INTO,A,SERIES,OF,
      DATA,STATEMENTS
30020 DATA BY,THE,COMPUTER,***
```

Obviously, the word list will be made up of whatever words *you* used when you run PWP1.

Next, you should write the rest of the program which uses these DATA statements. The following short program is an example which prints words from this list randomly.

```
10 DIM A$(40)
20 N=0
30 READ A$:N=N+1:IF A$<>"***" THEN 30
40 RESTORE
50 C=INT(N*RND(1)+1)
60 FOR I=1 TO C
70 READ A$
80 NEXT I
90 PRINT A$
100 GOTO 40
```

Once you have finished this program (which includes all the DATA statements written by the computer), you should save it using CSAVE or SAVE, and you are all done.

Using the concepts shown in PWP1, you should now be able to write your own Program Writing Programs. You can use PWP's to create all kinds of program segments. If you are really industrious, you might even want to make a PWP to write complete BASIC programs. If you do the job well enough, it just may be the last program *you* ever need to write! ©

Insight: Atari

Bill Wilkinson
Cupertino, CA

In my September column, I mentioned that the subject of graphics and I/O would make a nice series of columns. I wondered if there would be enough interest in the topic to justify the writing effort. Since that time, I have (in the course of talking to our customers) discovered that not only is there interest in the topic, but there is also a woeful lack of information and an abundance of misinformation regarding Atari's OS. So, with this column, we start a three or four part series on assembly language I/O.

Also, this month's column includes a list of major, known bugs in Atari BASIC and how to get around them.

Atari I/O, Part One: Interfacing To OS

Before I get started with the hairy details, I would like to state that Atari has the *best operating system* in the low-end microcomputer market. There is a simple reason for this: Atari has the *only* operating system on the market! Now, admittedly, I am being a purist when I make this contention, but the truth is that the Atari is the only machine I know of that has a *true* operating system in ROM. And, no, neither my company (Optimized Systems Software) nor I were involved in the creation of that operating system; the credit must go straight to Atari.

The operating system is contained in ROM and is identical on both the Atari 400 and Atari 800. The 10K bytes of ROM you may have noticed contain not only the operating system, but also the upper/lower case character set, the floating point mathematical operations, the power-on and cartridge select logic, and the device drivers. Device drivers? Aren't those part of the operating system? No! An emphatic "no." And that's what enables me to say that Atari has the only true operating system.

Believe it or not, the operating system on the Atari occupies less than 700 bytes. And yet it is as complete in its own way as UNIX is on a large time sharing machine. How many times have you read a magazine and seen lists of addresses of I/O subroutines for XYZ computer? You must use this address to output a character to the screen, another to get a character from the keyboard, yet another to talk to the line printer, and disk I/O? A nightmare! Not so Atari. One and only one address need be remembered: Hex E456, Decimal 58454. (Yes, I know, why not E400 or F000 or some such. Well, I didn't say Atari was perfect, only good.)

With only one address that matters, you can imagine that it should be easy for Atari to come out with new versions of the OS without affecting any other programs. They have, and they are, and only programs that have "cheated" (gone outside the OS rules) are in trouble. So, don't get yourself in trouble; follow the OS rules.

Finally, to avoid duplication of effort, I would refer you to the massive program listing for "SHOOT" in **COMPUTE!** #16: the first two pages and first column of the next two pages constitute most of the useful equates when using the Atari from assembly language. We are concerned with the "Operating System Equates" (first column, second page) and "Page three RAM assignments" (third column, first page). I will use the mnemonics given in that listing throughout this series of articles. (Those of you who own our "OS/A+" system will find these equates, with some mnemonics altered slightly for consistency, in the file "SYSEQU.ASM". You may use the EASMD pseudo-op ".INCLUDE #D:SYSEQU.ASM" to include them in any assembly programs. This will save you some typing.)

...the operating system on the Atari occupies less than 700 bytes.

The Structure Of The IOCB's

When a program calls the OS through location $E456, OS expects to be given the address of a properly formatted IOCB (Input Output Control Block). For simplicity, Atari has predefined eight IOCB's, each 16 bytes long, and the program specifies which one to use by passing the IOCB number times 16 in the 6502's X-register. Thus, to access IOCB number four, the X-register should contain $40 on entry to OS. Notice that the IOCB number corresponds directly to the file number in BASIC (as in PRINT #6, etc.). Actually, the IOCB's are located from $0340 to $03BF (refer to the "SHOOT" listing).

When OS gets control, it uses the X-register to inspect the appropriate IOCB and determine just what it was that the user wanted done. Table 1 gives the Atari standard names for each field in the IOCB along with a short description of the purpose of the field. Study the Table before proceeding.

The user program should *never* touch fields ICHID,ICDNO, ICSTA and ICPTL/ICPTH. In addition, unless the particular device and I/O request requires it, the program should not change ICAX1 through ICAX6. The most important field is the one-byte command code, ICCOM, which tells the operating system what function is desired.

The OS itself only understands a few fundamental commands, but Atari wisely provided for extended commands necessary to some de-

Now for adults.

Textwizard™ transforms Atari into a powerfully serious word processor.

Textwizard™ is no kids game. It's a sophisticated and complete word processing program for the **Atari 800 computer.

Here is instant control over the creation, editing and formating of any writing. Insert words. Replace phrases. Delete sentences. Move paragraphs. Automatically repositions surrounding text. *Textwizard™ does it all with over 50 simple commands.

On your command, Textwizard™ will search out and correct mistakes throughout your document. And it protects you from common operating errors by warning you before you make them.

Textwizard™ will print out as many original "copies" as you need. In boldface, elongated or condensed lettering, upper or lowercase, with proportional spacing. And you can store your work on a diskette.

So, whether you write legal briefs, computer programs, or The Great American novel, Textwizard™ can make your work easier. And though it's no toy, it's fun to use.

At $99.95, you don't have to write a best-seller to afford Textwizard.™ Textwizard™ is another of the many creative products from Datasoft.

Datasoft Inc.™
COMPUTER SOFTWARE

Software for people who aren't easy to please.

19519 Business Center Drive / Northridge, California 91324 / (213) 701-5161

Check your local software dealer or
Send check or money order with $2.00 postage/handling. California residents, add 6% sales tax.

*Textwizard will perform on a 32K system with one or more disc drives. It is compatible with the Atari® 825, Centronics® 737 and Epson® MX-80 printers.
** Atari is a registered trademark of Atari Computers Inc.

vices (XIO in BASIC). In any case, each one of these fundamental commands deserves a short description.

OPEN Open a device (synonyms: file, IOCB, channel) for read and/or write access. OS expects ICAX1 to contain a byte that specifies the mode of access: ICAX1 = 4 for read access, 8 for write access, and 12 for both read and write access. (Note: the disk file manager and the screen device handler allow other modes, and they will be discussed in a later section.) The name of the device (and, for the disk, the file) must be given to OS; this is accomplished by placing the ADDRESS of a string containing the name in ICBAL/ICBAH.

...the drivers account for over 5K bytes of the ROM code. The screen handler, with all its associated editing and Graphics modes, occupies about 3K bytes of that.

CLOSE Terminate access to a device/file. Only the command must be given.

STATUS Request the status of a device/file. The device can interpret this request as it wishes, and pass back a (hopefully) meaningful status. As with OPEN, the ADDRESS of a filename must be placed in ICBAL/ICBAH.

GET TEXT A powerful command, this causes the OS to retrieve ("GET") bytes one at a time from a device/file already OPENed until either the buffer space provided by the user is exhausted or an Atari RETURN character (Hex 9B, Decimal 155) is encountered. The user specifies the buffer to use by placing its ADDRESS in ICBAL/ICBAH and its size (length) in ICBLL/ICBLH.

PUT TEXT The analog of GET TEXT,OS outputs characters one at a time until a RETURN is encountered or the buffer is empty. Requires ICBAL/ICBAH and ICBLL/ICBLH to be specified.

GET DATA Extremely flexible command, this causes OS to retrieve, from the device/file previously OPENed, the number of bytes specified by ICBLL/ICBLH into the buffer specified by ICBAL/ICBAH. *No checks whatsoever are performed on the contents of the transferred data.*

PUT DATA Similar to GET DATA, except that OS will output ICBLL/ICBLH bytes from the buffer specified by ICBAL/ICBAH. Again, no data checks are performed.

Table 2 provides the OS commands and their usage of the various fields of the IOCB's. For convenience, the disk file manager extended commands are also shown, but I must withhold discussion of them until next month.

Device names on the Atari computers are very simplistic; they consist of a single letter (optionally followed by a single numeral). Traditionally (and, in the case of disk files, of necessity) the device name is followed by a colon. You have probably seen these device names in your various Atari manuals, but a quick summary might be convenient:

E: The keyboard/screen editor device. The normal console output.
K: The keyboard alone. Use this device to bypass editing of user input.
S: The screen alone. Can be either characters (à la E:) or graphics.
P: The printer. The standard device driver allows only one printer.
C: The cassette recorder.
D: The disk file manager, which also usually requires a file name.

Other device names are possible (e.g., for RS-232 interfaces) and, in fact, the ease with which other devices may be added is another reason for my claim that Atari has a *true* operating system. The structure of device drivers is material for a later article, but I should like to point out that the OS ROM includes drivers for all the above except the disk. In fact, the drivers account for over 5K bytes of the ROM code. The screen handler, with all its associated editing and Graphics modes, occupies about 3K bytes of that.

Actually, the next column will begin to delve deeper into the ways of using OS, but for those of you anxious and brave enough to get started now we present a very simple example program:

```
PUTMSG                      ; A ROUTINE TO PRINT A
                              MESSAGE
        LDX   #$00          ; WE USE IOCB NUMBER 0, THE
                              CONSOLE (E:)
        LDA   #PUTREC
        STA   ICCOM,X       ; THE COMMAND IS 'PUT TEXT
                              RECORD'
        LDA   #MSG&255
        STA   ICBAL,X       ; LOWER BYTE OF ADDRESS
                              OF 'MSG'
        LDA   #MSG/256
        STA   ICBAH,X       ; UPPER BYTE OF ADDRESS
        LDA   #255
        STA   ICBLL,X       ; LOWER BYTE OF LENGTH OF
                              MSG
        STA   ICBLH,X       ; UPPER BYTE, LENGTH IS ALL
                              OF MEMORY
                            ; BUT 'PUTREC' WILL STOP
                              WITH THE 'RETURN' CHAR
        JSR   CIOV          ; CALL THE OS TO DO THE
                              WORK
        TYA                 ; MOVES RETURNED ERROR
                              CODE TO A-REGISTER
        BMI   ERROR         ; ANY NEGATIVE VALUE IS
                              SOME SORT OF ERROR
```