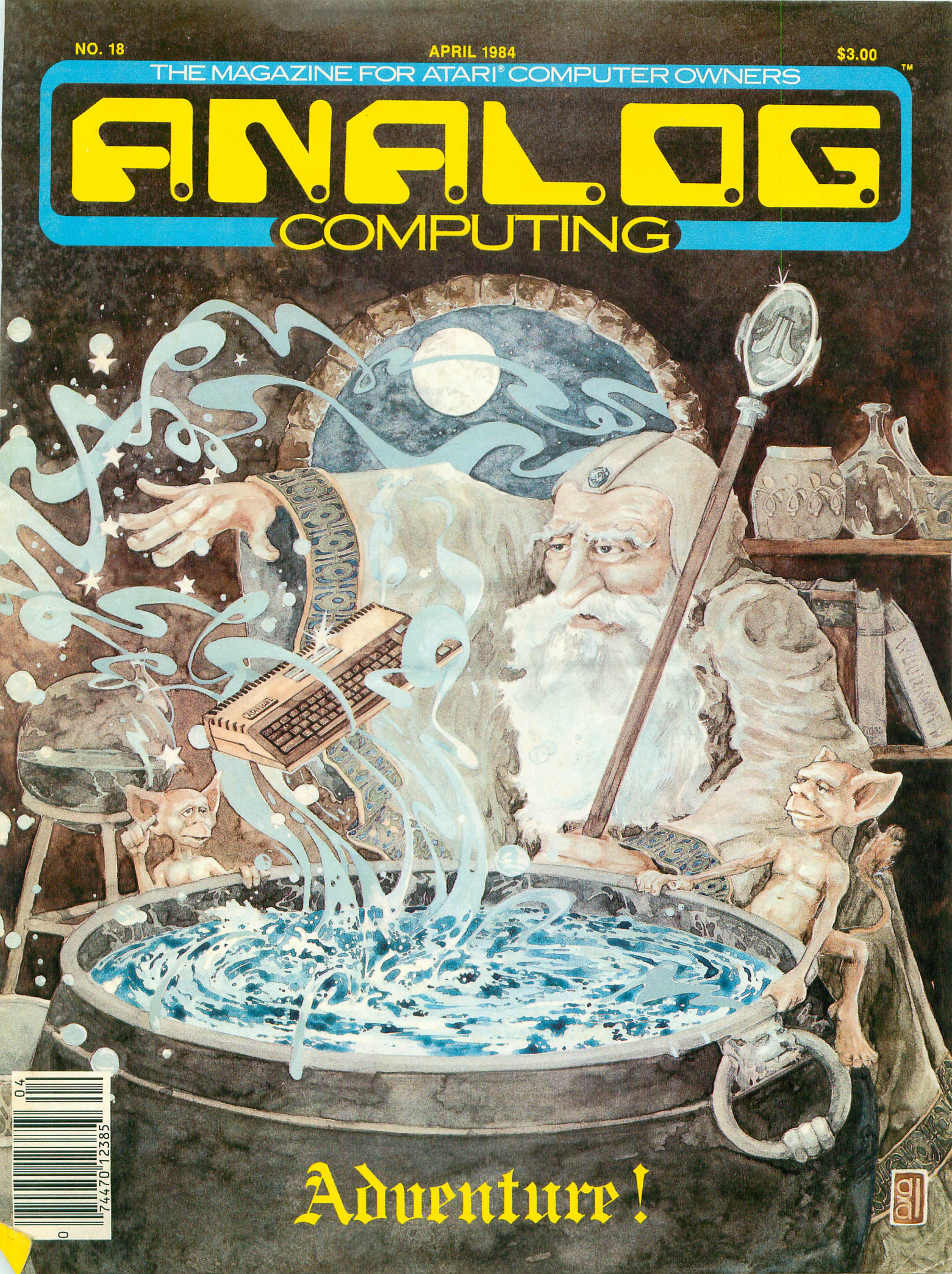
NO. 18 APRIL 1984 $3.00

THE MAGAZINE FOR ATARI® COMPUTER OWNERS™

ANALOG COMPUTING



Adventure!

REMEMBER.

ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison

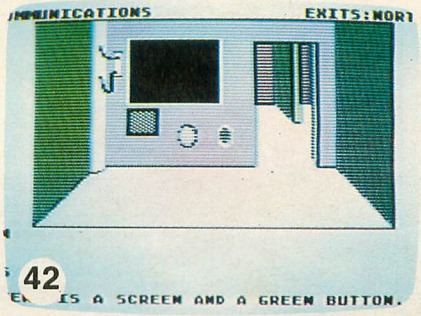
CIRCLE #101 ON READER SERVICE CARD.

NO. 18 APRIL 1984

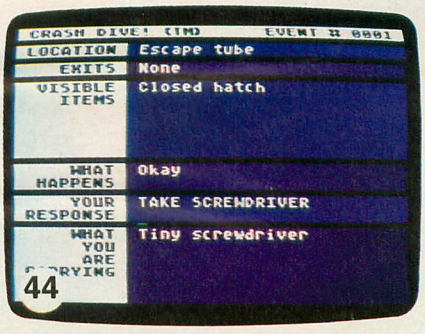
THE MAGAZINE FOR ATARI® COMPUTER OWNERS

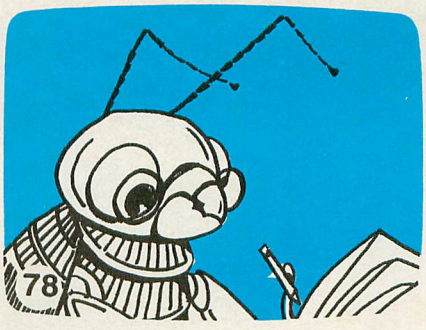
ANALOG COMPUTING™

FEATURES

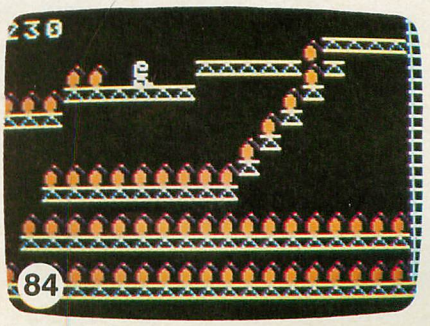


PRODUCT REVIEWS





COLUMNS



ANALOG COMPUTING (ISSN 0744-9917) is published monthly for $28 per year by ANALOG Magazine Corp., 565 Main Street, Cherry Valley, MA 01611. Second-class postage paid at Worcester, MA and additional mailing offices.

Today's most innovative educational software began here 60 years ago.

At Scholastic, we have something no other educational software company has: 60 years' experience making learning fun for kids.

We began in the schoolrooms of the 20's with the first national news magazine written especially for young people, The Scholastic. Since then, our one magazine has grown into 37, and we've become the largest educational publisher of books and magazines in the English-speaking world.

Now we've put everything we've learned from five generations of school children into the most innovative family of educational software available today. Scholastic Wizware.™

Our experience makes Wizware different from all other educational software. It turns learning subjects like geography, writing and spelling into exciting adventures for your children. And because every Wizware game is *interactive,* kids become deeply involved in what they're learning.

What's more, our experience has taught us the importance of teaching things most other educational software leaves out. Like teamwork, imagination, critical thinking and problem solving. You'll find them all in Wizware.

But by far the most important result of our experience is that your children will thoroughly enjoy learning with Wizware.

There are now Wizware programs for teaching everything from creative writing to computer programming. Here are a few of the ways we bring learning into the Computer Age.

Spelldiver.™
It's the most ingenious way ever devised for teaching spelling and improving reading skills. Deep beneath the sea lie giant words covered by a strange seaweed called lettermoss. Kids must face ferocious sharks and pesky flippernippers to remove the lettermoss and decipher the words.

Agent U.S.A.™
Agent U.S.A. turns geography into an exciting race to save the nation from the ruthless Fuzzbomb. Along with learning geography, children hone their problem-solving and reasoning abilities.

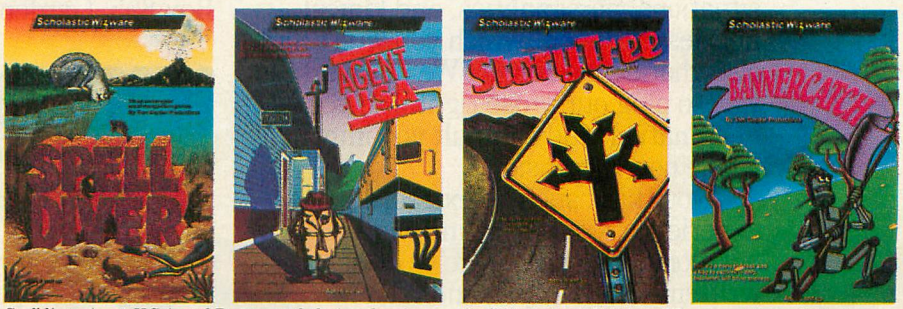
Story Tree.™
Story Tree can bring out the Mark Twain in every child. Budding authors create their own mystery and adventure stories. A remarkable feature lets them weave alternate choices into every turn of the plot, and challenges their imaginations to the maximum.

Bannercatch.™
Based on the classic game of Capture-the-Flag, this is the most sophisticated and fun strategy game for kids available today. Not only do players learn how to devise complex strategies, they also learn how to work together to solve difficult problems.

Scholastic Wizware. Our experience makes all the difference.

Look for Wizware at your local computer store. Or contact Scholastic Inc., 730 Broadway, New York, NY 10003, 212-505-3000.

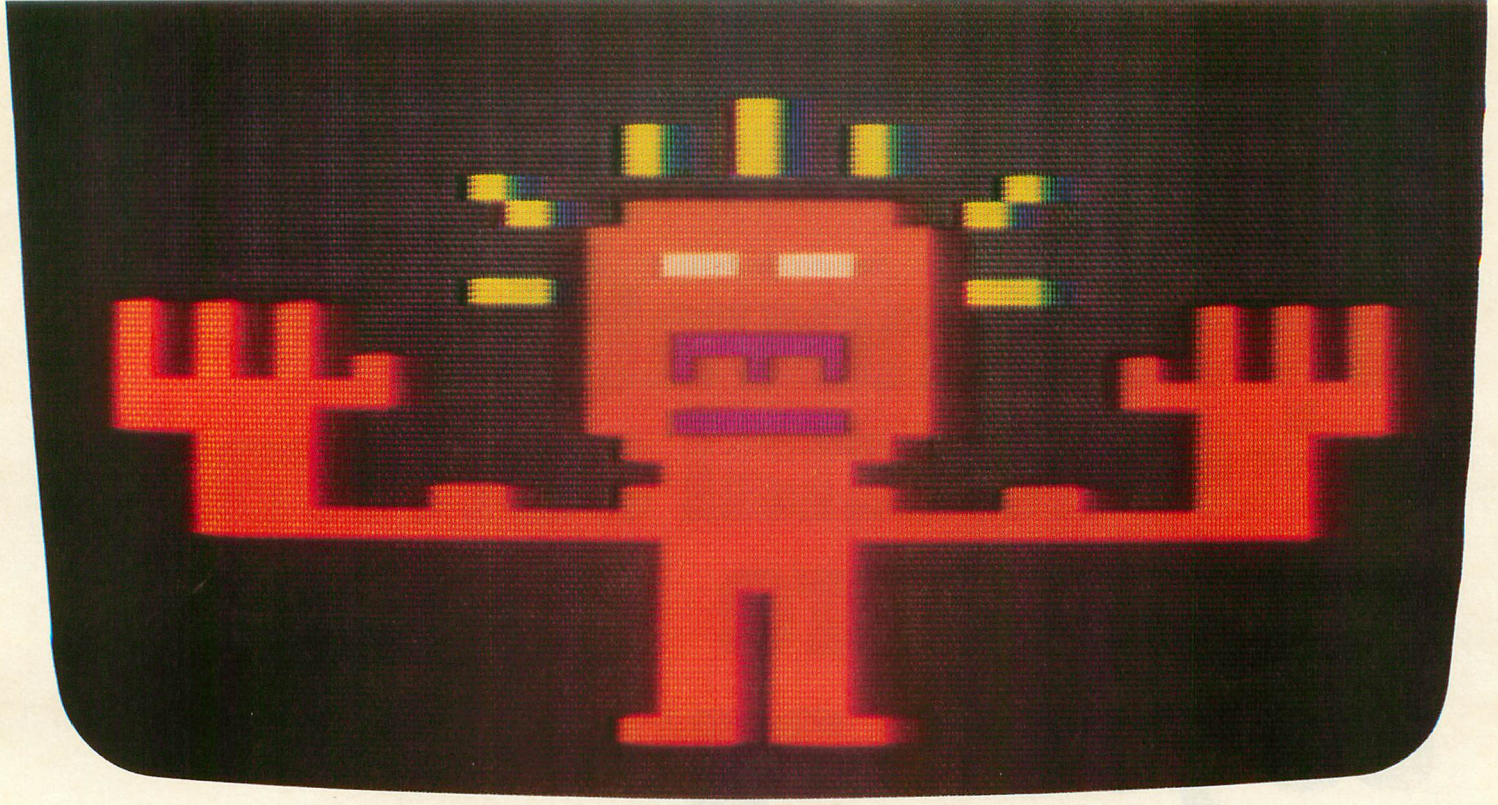
Scholastic™ Wizware



Spelldiver, Agent U.S.A. and Bannercatch designed and developed by Tom Snyder Productions, Inc. Story Tree designed and developed by George Brackett.

Spelldiver, Agent U.S.A. and Bannercatch available for Atari 800/1200/XL. Commodore, Apple and IBM versions available soon. Story Tree available for Apple.

CIRCLE #102 ON READER SERVICE CARD.



WOULD YOU SHELL OUT $1000 TO MATCH WITS WITH THIS?

SUSPENDED,™ The WITNESS,™ PLANETFALL,™ ENCHANTER,™ and INFIDEL™—has become an instant best-seller. For the simple reason that Infocom offers you something as rare and valuable as anything in software—real entertainment.

Meet your match. Meet Infocom games: perhaps the best reason in software for owning a personal computer.

In fact, people have been known to purchase computers and disk drives solely for the purpose of playing our games. And they haven't been disappointed. Because Infocom's prose stimulates your imagination to a degree nothing else in software approaches. Instead of putting funny little creatures on your screen, we put you inside our stories. And we confront you with startlingly realistic environments alive with situations, personalities, and logical puzzles the like of which you won't find elsewhere. The secret? We've found the way to plug our prose right into your imagination, and catapult you into a whole new dimension.

If you think such an extraordinary experience is worth having, you're not alone. Everything we've ever written—ZORK® I, II, and III, DEADLINE,™ STARCROSS,™

At last, you can fritter away your evenings playing a computer game without feeling like you're frittering away your computer investment.

Step up to Infocom. All words. No pictures. The secret reaches of your mind are beckoning. A whole new dimension is in there waiting for you.

(For more information on Infocom games contact: Infocom, Inc., P.O. Box 855, Garden City, NY 11530.)

INFOCOM™

The next dimension.

For your: Apple II, Atari, Commodore 64, CP/M 8," DEC Rainbow, DEC RT-11, IBM, MS-DOS 2.0, NEC APC, NEC PC-8000, Osborne, TI Professional, TI 99/4A, TRS-80 Model I, TRS-80 Model III.

CIRCLE #103 ON READER SERVICE CARD.

ANALOG STAFF

Editors/Publishers
MICHAEL DESCHENES
LEE H. PAPPAS

Managing Editor
JON A. BELL

Technical Editor
BRIAN MORIARTY

Contributing Editors
JOEL GLUCK
BRADEN GRIFFEN, M.D.
TONY MESSINA

Art Director
BOB DESI

Contributing Artists
GARY LIPPINCOTT
LINDA RICE

Technical Division
CHARLES BACHAND
TOM HUDSON
KYLE PEACOCK

Advertising Manager
MICHAEL DESCHENES

Distribution
PATRICK J. KELLEY

Production/Distribution
LORELL PRESS, INC.

Contributors
RAY BERUBE
CLIFF CHAPUT
MARK COMEAU
CARL FIRMAN
SALLY FORTH
NORMAN HILL
MICHAEL LONG
STEVE PANAK
CLINTON PARKER

ANALOG Magazine Corp. is in no way affiliated with Atari. Atari is a trademark of Atari, Inc.

ABOUT THE COVER

This issue's stunning cover was the work of Gary Lippincott, a multi-talented artist who resides in Spencer, Massachusetts. When Gary breezed into our offices a few months ago with his portfolio, out tumbled some of the finest fantasy artwork we had ever seen. "Aha!" we surmised, "the perfect person to do the cover for our Adventure issue!" We gave Gary a quick sketch of what we wanted, and two weeks later Gary came back with our cover. Rendered in both watercolor and tempera paint, it stands as a representation of a classic adventure character — a sorceror and his apprentices — engaged in an unusual form of high technology alchemy. Keep your eyes peeled for more of Gary's artwork on future covers of **ANALOG Computing**.

ADVERTISING SALES

JE Publishers Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038

Los Angeles	—	(213) 467-2266
San Francisco	—	(415) 864-3252
Chicago	—	(312) 445-2489
Denver	—	(303) 595-4331

Gerald F. Sweeney & Associates
P.O. Box 662
New York, NY 10113
(212) 242-3540

ANALOG Computing
Home Office
Michael DesChenes
National Advertising
(617) 892-9230

Address all advertising materials to:
Michael DesChenes — Advertising Production
ANALOG Computing
565 Main Street, Cherry Valley, MA 01611

ANALOG COMPUTING (ISSN 0744-9917) is published monthly for $28 per year by **ANALOG** Magazine Corp., 565 Main Street, Cherry Valley, MA 01611, Tel. (617) 892-3488. Second-class postage paid at Worcester, MA and additional mailing offices. POSTMASTER: Send address changes to **ANALOG COMPUTING**, P.O. Box 615, Holmes, PA 19043. No portion of this magazine may be reproduced in any form without written permission of the publisher. Program listings should be provided in printed form. Articles should be furnished as typed copy in upper and lower case with double spacing. By submitting articles to **ANALOG COMPUTING**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **ANALOG**. If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned, please supply self-addressed, stamped envelope. U.S.A. Newstand distribution by Eastern News Distributors, Inc., 111 Eighth Ave., New York, NY 10011.

Contents copyright © 1984 ANALOG Magazine Corp.

IN THIS ISSUE

by Brian Moriarty

As our superb cover by Gary Lippincott clearly shows, this month's issue is devoted to Adventure. We've selected new products from the biggest names in the adventure business (including Sierra On-Line, Adventure International, Infocom, Quality Software and others) and passed them to our reviewing staff for consideration. Many reviews that we hoped to include didn't fit, so look for the spillover in next month's issue.

Virtually all commmercial Atari adventures require at least 32K of memory to play. This slams the door of adventure smack in the face of several hundred thousand Atari users who only have 16K, including all owners of the 600XL system. This month's feature game is an attempt to give the 16K crowd a taste of what they're missing. **Crash Dive!** is a little too tricky to be called a beginner's game, and you need a system bigger than 16K to type in the BASIC version. But the boot-load version fits snugly in a 600XL; and though modesty prevents me from telling you how great and exciting it is, I think you'll find the effort of borrowing a bigger machine to type it in well worth it.

We haven't forgotten the joystick crowd in this issue, either. Mark Comeau's **Munch'In Climb'In** uses colorful player/missile graphics and sound effects to send you on a hungry rampage across a network of girders and ladders. It's a very challenging game that proves once again how much you can do with Atari BASIC with a little ingenuity — and Tom Hudson's PMG routines!

Speaking of Tom, don't miss this month's **BASIC Training** installment. Would-be game designers will find this discussion of graphics vectoring most enlightening. Part II of Clint Parker's **Introduction To Action!** features a high-speed plot routine that makes his kaleidoscope demo really zoom along. Dr. Griffin looks at a couple of interesting "simulated computer" programs in his **Griffin's Lair** educational column, and Sally Forth sheds light on the use of arrays in her favorite language. You may notice that Joel Gluck's **Our Game** isn't listed on the contents page. No, he hasn't suffocated under a pile of mail! He's taken a short vacation from the rigors of **Our Game**, both the column and sorting through his correspondence, and will return next issue with **Our Game**.

Fans of Tom Hudson's **Boot Camp** column get a special bonus this month. Besides his usual poking around in the 6502 registers, Tom presents an extremely powerful little utility for machine-language debugging. **HBUG** is just the ticket for **MAC/65** owners who need a fast, painless way to find out why their "flawless" M/L masterpieces don't run. (I personally have little use for **HBUG**, since all of my program errors are caused by bugs in the hardware. Isn't that right, Tom?)

Looking for another utility to add to your growing collection? Norman Hill submitted a nice program that keeps track of magazine articles for you. **File'em** is great when you need to reference a certain product review but can't remember where you saw it. Use it in conjunction with Issue 15's **ANALOG Index** and missing articles will become a thing of the past. You can even use **File'em** to organize articles from other magazines — but why bother?

On a serious note.

Last November, I was invited to appear as a guest speaker at the 2nd anniversary party of the Tulsa Regional Atari User's Group. TRACE is a very large and active organization, with an excellent newsletter and a devoted, enthusiastic membership.

One of TRACE's many special interest groups is devoted to helping the hundreds of handicapped people in the Tulsa area. While in Tulsa, I witnessed a demonstration of the inexpensive hardware and software they have developed to make it easier for the handicapped to communicate with the outside world. Their apparatus consists of a little rubber tube connected to a switchbox. By gently blowing or sipping on the tube, I was able to "type" words and messages on a TV screen. It sounds simple, and it is — but the value of TRACE's Sip-And-Puff Communicator to a severely handicapped person cannot be overestimated.

We're delighted to be able to share TRACE's invention with the rest of the Atari Community. Take a few minutes to read Michael Long's article on **Communications for the Handicapped**. It could make life a lot easier for a handicapped person in your area, perhaps even someone you know. □

Precision Software Tools
The Right Tools For The Right Job

OSS, the leader in software tools for home computers, introduces a new line of application tools that you can use at home yet are versatile enough for the professional.

The Write Tool™

Get all the power you need in a word processor. Compare our features. Compare our value. Buy The Write Tool at the right price. Use with cassettes or disks. On cartridge, $49.95.

Move Over Atari Writer!

The Postal Tool™

Manage your list like never before. Add and change names easily. Prepare bulk mailings. Organize your lists. Select and sort by multiple categories.

The Sort Tool™

Easily sort almost any file. Produce key files, tag files, or fully-sorted files. Sort lines, fixed-length records, or variable-length records. Try The Sort Tool, fast and efficient.

Both The Postal Tool and The Sort Tool on disk only, $59.95.

The Print Tool™

Give your current word processor professional features. Indexing. Outlines. Footnotes. Table of contents. 35mm slides. And much more. Make your documents appear their best. Use The Print Tool. On disk only, $59.95.

See your dealer first. MC/VISA, C.O.D. accepted.

All products available for computers made by Atari, Inc.

OSS
Precision Software Tools

1173D S. Saratoga/Sunnyvale Rd.
San Jose, CA 95129 • (408) 446-3099

The Write Tool, The Print Tool, The Postal Tool, The Sort Tool and OSS are trademarks of Optimized Systems Software, Inc.

CIRCLE #104 ON READER SERVICE CARD.

READER COMMENT

In reference to my program, **BASIC Cassette Recovery**, I would like to note that the instruction to 'RUN' the program after typing it in and saving it was missing from the article.

Another problem has arisen, which I was not aware of. Although the program seems to run without error here on my machine, and several others (Thanks for calling...), there are machines that it won't run properly on. I am, frankly, at a loss to explain this. As a result, I plan to write a more comprehensive program to do the same thing, which should solve the problem. In the mean time, when the program asks you to press return, press SYSTEM RESET instead.

The question has arisen about the possibility of recovering a bootable cassette program. I would like to let your readers know that it isn't possible to recover these programs because there is no way of knowing what the program did (from the point of view of a recovery program). The problem is akin to reconstructing a house from a foundation and no plans. You can get close, but close doesn't count in a machine language program.

In closing, I'd just like to say, "Keep up the good work" with an excellent magazine.

Bob Fine
Tomkins Cove, NY

Action! Update

Thank you for your very nice review of **Action!** in Issue 16, but I would like to clear up a couple of details. First, just to make sure your readers aren't misled; I do not work for OSS.

Secondly, a few comments about listing number seven. Although I feel it is fair to code it the way you did, I feel your readers are missing out on some of the power of **Action!** by doing such a straightforward translation from the BASIC version. If you declare SCREEN as a BYTE ARRAY:

```
BYTE ARRAY SCREEN
```

and replace the

```
POKE (SCREEN+J,255)
```

with:

```
SCREEN(J)=255
```

you will not only get a cleaner looking program, but one that runs faster (21 jiffies, 34% faster). Since **Action!** supports 16 bit numbers, you could declare SAVMSC as a CARDinal and replace:

```
SCREEN=SAVMSCL+256*SAVMSCH
```

with:

```
SCREEN=SAVMSC
```

Now, this doesn't have much impact on the timing of this program, but it would on any program that does a large number of 16 bit references.

Clinton Parker
Action! Computer Services

Software Piracy

I am sitting here re-reading Greg Walforf's letter in Issue 13 and Alex Leaven's reply in Issue 15, on the matter of Piracy. Unfortunately, in my opinion, too much has been left *un*said...

Re G.W.'s letter; It must be the umpteenth time I've read *this* particular justification of piracy. It's become a litany. One wonders just how much guilt these people must carry, that they feel contiually constrained to write letters excusing their thievery and to assuage their burdened consciences.

Me? Unfortunately, like A.L., I sit on the other side of the fence, developing software for *free*. That's how it sometimes appears as I rip open an envelope bearing a royalty payment only to find a check perhaps *one-fifth* what it rightfully should be. Forget for the moment fellows like Budge & Wetmore, the deservedly well-paid Superstars who don't have to worry about mundane things like putting food on the table and buying shoes for an active four year old child. For the vast majority, professional programming is a very tough career, replete with growing pains, slow progress, long and *brutally* hard hours and the tantalizing dream of success.

The success does *not* come overnight...

Dan Gorlin spent the better part of one year developing a brilliant graphical *tour-de-force* that also happens to be an eminently playable game. The effort that went into this masterpiece defies tenacity and is a touch more than genius. D.G. *owns* **Choplifter**, having paid for it with a commensurate share of labor and sweat. I, for one, pay tribute by paying to play.

But what of the characters of G.W.'s ilk? Far be it from them to pay for something more easily obtained. So be it...the pirate says to hell with Broderbund & Gorlin; they're making too much anyway. Broderbund, having spent a sizeable sum on development and promotion with no guarantees, does *not* agree! Gorlin, while being a generous sort, is not indisposed to being paid for his labors.

G.W. also suggests a lower

program pricepoint would eradicate the impetus to pirate. Well, how then will Greg justify his purchase of the Happy Enhancement? Will it become a paperweight? The warped logic of so-what-it's-only-one-copy is absolutely blind to the truth. In reality, when the *average* pirate makes a copy for *two* friends, and so on and so on, eventually hundreds of illegal copies may be in circulation. This is the crux of the matter. Of course, one copy makes little difference. In practice, however, the "one" copy is eventually multiplied by thousands, and the final effect is patently clear. Witness the fact that most of Atari's releases in early 1983 were available on illegal Bulletin Boards *before* Atari actually published them. I, myself, had the distinct displeasure of seeing one of my own games being demonstrated in a local store by a neighborhood pirate before the game was actually released by Adventure International! Is there any doubt that A.I. and I received *less* than what we were entitled to?

Please, please, no more letters justifying piracy. I and no one else can prevent the onslaught of the rationalizations that permit piracy to continue. We are, indeed, talking about *theft*, reprehensible and damaging to the developers of the software. Most of us are merely human beings (Budge is probably an E.T.) fighting to survive in a rapidly changing environment, hopefully to prosper. It ain't easy. If you need any further proof of your folly, just look at the market for Atari software. It has changed radically. Far fewer products are being released as publishers realize that perhaps anywhere from 3 to 10 illegal copies are being made for each copy legally sold. Many stores and distributors are now cutting back to a large extent on Atari software, due in part to the effect piracy has had on sales.

We are at a crossroads. Make no mistake, the future of the Atari as a serious development tool may be at stake. If you feel you *must* have the program, pay for it. Any alternative is *non compos mentis*.

Alan M. Newman
Brooklyn, NY

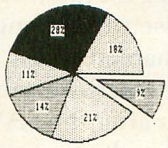
Create-A-Font

First, I would like to take this opportunity to thank you for publishing the most valuable and informative magazine available to the serious Atari owner. I have learned more about programming from your articles than from all the books I've read combined (and that's quite a stack). My special thanks to Tom Hudson, for well documented source code listings that are easily modified and expanded. Keep up the good work!

The **Create-A-Font** program which appeared in last issue was by far the most creatively designed BASIC utility program I've ever seen. It is such a pleasure to use that what was once a chore is now almost like playing a game. One thing about the program did disturb me, however. When modifying the alphabetic characters, the menu portion of the screen would eventually become unreadable. I decided to insert a display list interrupt in the program to switch character sets in the lower part of the screen. Now, the menu is always perfectly readable, even if all the characters are modified. Add the following lines of code:

```
45 DIM DLI$(12):RESTORE 20
80:FOR X=1 TO 12:READ N:DL
I$(X)=CHR$(N):NEXT X:RESTO
RE
110 CHBAS=X*C256:A=USR(ADR
(CLEAR$),PM-C256*C2,C256*C
4):POKE 705,148:POKE 710,C
0:POKE 712,148
155 GOSUB 2000
182 POKE 54286,192
2000 IF PEEK(DL-C8)<>2 THE
N 2070
2010 POKE 1791,X
2020 POKE DL-C8,PEEK(DL-C8
)+128
2030 ADL=ADR(DLI$)
2040 POKE 513,INT(ADL/C256
)
2050 POKE 512,ADL-(PEEK(51
3)*C256)
2060 POKE 54286,192
2070 RETURN
2080 DATA 72,173,255,6,141
,10,212,141,9,212,104,64
```


GRAPHICS HARDCOPY



$34.95

*Dumps anything on the screen of your Atari** Computer to a printer. All graphics and text modes supported. Great for graphs, pictures, custom fonts, etc. Includes scaling and grey scale. Works with EPSON*, NEC*, OKIDATA*, and GEMINI* connected to the Atari 850 Interface Module. Includes software on disk (only) and extensive User Manual. Order Part No. "GH" $34.95.*

SCREEN PRINTER INTERFACE
Same as Graphics Hardcopy but includes a cable to connect your 36 pin (Centronics) style) parallel printer to controller ports 1 and 2, thus eliminating the need for the 850 Interface Module. Compatible with BASIC, Editor/Assembler, PILOT and Atari Writer. Includes software on disk (only), printer cable, and extensive User Manual. Order Part No. "APG" $79.95

(209) 667-2888 VISA MASTERCARD C.O.D.

MACROTRONICS, inc.®

1125 N. GOLDEN STATE BLVD.
TURLOCK, CA. 95380

**Atari is a registered trademark of Atari Computer Inc.
* NEC, Epson, Okidata and Gemini are registered trademarks of NEC Information Systems, Epson America Inc., Okidata Corp., and Star Micronics respectively.

CIRCLE #105 ON READER SERVICE CARD.

NONVIOLENT FAMILY FUN AT AFFORDABLE PRICES!

Dawn of Civilization

A million years ago mankind was faced with two great challenges - adequate food supply and defense against the threat of invasion from unknown forces. **DAWN OF CIVILIZATION** gives you a chance to help a tribe of early people feed itself, fend off invaders, and reproduce to insure continuation throughout generations. Written in **100% machine language** for one to four human or computer players, this game is easy to learn, but difficult to master! Your skillful play can insure the continuation of mankind - can you meet the chalenge?

ALL ATARI® - 16K disk or tape $10.95

After Pearl

Can you duplicate the amazing effort put forth by the American forces after Pearl Harbor? **AFTER PEARL** is an exciting strategic simulation of the war in the Pacific. The game comes with a 40 turn campaign covering the entire period from Pearl Harbor to the capture of Iwo Jima. The country's forces are at your command, and 113 aircraft carriers, battleships, and cruisers are included in the action. AFTER PEARL will challenge you, either in its solitaire version or against a human opponent. **100% machine language** programming keeps the action in AFTER PEARL as fast as real life!

ALL ATARI® - 48K disk or tape $19.95

SUPERware

ORDER NOW! 2028 Kingshouse Rd. Silver Spring, MD 20904 **(301) 236-4459** Dealer Inquires Invited

MC, VISA, AMX, Check, M.O. Accepted
Include $2.00 shipping, MD residents add 5% tax.
Write SUPERware for your **FREE** copy of BASIC MAGIC I - programming tricks for the ATARI.

CIRCLE #106 ON READER SERVICE CARD.

You should now be able to alter any character with **Create-A-Font** without disturbing the menu. During disc or cassette SAVing or LOADing of character sets, the DLI will be momentarily disabled, but this will clear after the SAVE/LOAD is completed.

Randolph Constan
East Islip, NY

I've been an Atari 800 owner for about 18 months and your magazine has become a standard tool in helping me get the most from my system. My applications for the Atari at this time are graphics and word processing with other business uses planned in the future.

I would like to use my system in the design and execution of programmed slide presentations using multiple projectors and dissolve units. I've been looking into the equipment of some manufacturers and have seen several that offer software and cards for the Apple computers, but nothing seems to be available for the Atari. It would seem possible to interface with the joystick ports and use the sound capabilities for the computer to generate the audio "cues" on tape to control the projectors, and store the information on disk for future presentations.

Perhaps you or your readers can provide some ideas and suggestions for this concept as I'm sure this would be an interesting application for the Atari computer, and a challenge for the serious programmer.

Very truly yours,
David LaComb
Utica, N.Y.

Lock Ups

I've had my Atari 800 computer for about 10 months. I bought my first issue of **ANALOG** (July/August 1983) and typed in **Cat and Mouse**. After I saved it on cassette, I ran the program and to my dismay my computer locked up. Awhile later, a friend was over and was typing in a game from a book. He ran the program after every line to see how it was coming along. When he reached the line with the USR command he typed it in and ran it (the program). The computer locked up. After seeing this we proceded to take out the line with the USR command in **Cat and Mouse**. The game ran fine except for the lack of a displayed score. I have had a few other lock ups but not all of the USR commands were followed by an ADR command, such as **Fast Repeat** in your February 1984 issue. **Stuntman** from your **Compendium** also locked up but ran fine without the USR command. I'm not sure what to do. I hardly ever take my basic cartridge out of my computer. Could I have something wrong with my basic cartridge because of leaving it in? Or is my ROM messed up somehow? I really enjoy your magazine and look forward to it every month. I've written to find out what is wrong and to see if my computer needs possible repairs.

Kirk Lampert
Seville, Ohio

Both "Stuntman" and "Cat and Mouse" work as listed. These programs contain USR functions with control characters.If any of these characters are mistyped, a lockup will almost certainly occur. Use the C:CHECK or D:CHECK Program, included in the **Compendium**, *to check your typing.*

—TH

In response to Chris Johnson's letter in issue #16, I would just like to say that **Starbowl Football** is one of the best computer games I have played. There are ways to beat the pro team consistently. One, go after the interception, the only way you can score is to have the ball. Secondly, find a play that is reliable for 10 or more yards. Hint: try programming the top receiver for a slant-in to catch the ball, a middle rush, and a screen on the bottom receiver. If the linebacker is on a blitz, a caught pass will result in 6 points.

Gamestar's Baseball is equally as exciting and I hope to purchase it soon.

Thank you,
P. Curtis
Frankfort, MI

Send letters to:

READER COMMENT
P.O. BOX 23
WORCESTER, MA 01603

COST EFFECTIVE SOFTWARE
BY
"The Best Little Software House In Texas"

HOMEBASE by SOFT SECTRE is a versatile database program for the home, small business or lab. In a 'USER FRIENDLY' MENU driven format HOMEBASE utilizes TWENTY COMMANDS: CREATE, ADD, LIST, CLIST, CHANGE, CONCATENATE, SEARCH, SUM, SORT, DELETE, LABELS, LOAD, PRINT, SAVE, DIRECTORY, HELP, DRIVE, AUDIO, LOWER CASE, END. An optional SECURITY CODE prevents unathorized data file retrieval and manipulation. Optional AUDIO FEEDBACK signals the end of a command response. The ATARI version uses a MACHINE LANGUAGE SORT!

HOMEBASE is ideal for: MAILING LISTS (we use it ourselves); household, insurance and equipment INVENTORIES; MAINTENANCE schedules; a PRESCRIPTION list; refund COUPON and GROCERY lists. Ready-to-use data files for these tasks are included. Why buy several programs? Purchase the ONE program that will handle all of your database needs. HOMEBASE by SOFT SECTRE will SAVE you TIME and MONEY!

SELECTRE by SOFT SECTRE is included in the disk based package at no additional charge (a $29.95 value!). Disk DIRECTORIES and PROGRAMS are loaded with numeric key input allowing even the most inexperienced person to easily operate the computer.

ATARI OR PET DISK 32K $49.95
PET TAPE 16K $49.95
Add $2.00 for shipping.

SOFT SECTRE
P.O. BOX 1821, PLANO, TX 75074
Send for FREE catalog! VISA/MasterCard WELCOMED
CIRCLE #108 ON READER SERVICE CARD.

MORE DISK DRIVE FOR YOUR MONEY .....

In fact, with the ASTRA 1620, you get two superb Disk Drives for the price of one. The ASTRA 1620 is Single or Double Density (software selectable) and completely compatible with ATARI DOS or OSA+ DOS. When used as Double Density, the ASTRA 1620 has the same capacity as Four ATARI 810® Disk Drives.

ONLY $595

DOUBLE OR SINGLE DENSITY

The ASTRA 1620 can be either single or double density, depending on the software selected. One drive can be configured for single density and the other drive for double density, or any combination desired. The ASTRA 1620 is compatible with virtually any software available for ATARI® Disk Drives. The ASTRA 1620 is smooth, quiet and fast. In Single Density mode, the ASTRA 1620 stores 88K bytes of programs or files. In Double Density, the ASTRA 1620 stores 176K bytes, simply twice as much.

TWO DRIVES ... Yes, two superb disk drives in the same size enclosure normally used for one drive. The ASTRA 1620 measures 7⅞" wide x 11⅞" deep x 5⅞" high.

Two drives will open a new dimension of computing for you. The program disk can be in one drive and the data disk can be in the other. This will eliminate time consuming disk changes. Backing up disks and copying files will never be easier. Just follow the instructions on the screen and walk away. The job will be completed within minutes. We have simplified copying from single to double density. With two drives, it's just as easy as copying in one mode. No disk switching!

Two double density drives give you the power that much larger and more expensive computers have without giving up any of the features available on the ATARI® Home Computer.

EASY TO USE ... The ASTRA 1620 comes complete with everything you need. Just plug it in, chain it up, and turn it on.

The ASTRA 1620 comes with OSA+ DOS (The best disk operating system available for the ATARI® computer!). The OSA+ DOS is completely compatible with all existing ATARI DOS files. Because the OSA+ user manual is very complete and technical, we include our own simplified user manual. Between the two furnished manuals, you have the information necessary to perform any task required of your disk drive.

The ASTRA 1620 also contains a data cord, power transformer, and operator manual.

ASTRA SYSTEMS

5230 Clark Avenue, Suite 19
Lakewood, California 90712

Phone
(213) 804-1475

CIRCLE #109 ON READER SERVICE CARD.

The Second ANALOG Guide to Atari Computer Publications

by Lee Pappas

ANALOG continues its *Guide To Atari Computer Publications* in the effort to keep our readers up to date on new books of interest. This issue lists books released since the previous guide printed in issue 12 (July/August 83). And don't forget the **ANALOG Compendium** ($14.95), containing over 50 programs covering graphics, disk and programming utilities, and the finest magazine games ever.

HARDY, JACK B. **ADVENTURES WITH THE ATARI.** 356pp. RESTON84, 14.95
Ever wish you could write your own adventure game? Well this book not only shows you how, but includes several listings for games (some using graphics) such as Escape, The Hunter, Time Crime, and The Creator. Some listings are in Atari BASIC, while others are written in PILOT and Microsoft BASIC to show how some languages are better than others for certain applications. Additional information covers flowcharting, mapping, and game design.

FERNANDEZ, JUDI N., DONNA TABLER, and *RUTH ASHLEY.* **6502 ASSEMBLY LANGUAGE PROGRAMMING.** 277pp. WILEY83, 12.95
Examples and diagrams assist in understanding the complex structure of machine language. Frequent questions give this book a textbook resemblance with answers provided at the end of each chapter.

LEVENTHAL, LANCE A., **ASSEMBLY LANGUAGE PROGRAMMING,** 640pp. OSBORNE, 18.95
One of the finest instructional guides to learning the 6502 language of the Atari computers. An easy to understand guide into the complex and challenging world of assembly coding.

EVANS, CARL M. **ATARI BASIC FASTER AND BETTER.** 300pp. IJG83, 19.95
A hefty book dedicated to Atari BASIC, 16 chapters cover extensive use of string manipulation, USR (machine language subroutines) calls, sound usage, screen handling and data structure. How to set up your own AUTORUN.SYS files, disk catalogs, and scrolling fields are also covered. Program listing, sound effect demos, and a multitude of charts and tables will get you on your way to advanced programming in BASIC.

SEYER, PHILIP C., **ATARI PLAYER-MISSILE GRAPHICS IN BASIC.** 173pp. RESTON83, 14.95
Philip Seyer has done an admirable job documenting, in an easy to understand format, how to go about plotting and moving the Atari's player-missile graphics. Tables and program listings illustrate how to accomplish these, along with additional information that will help the average user get more out of their computer.

PHILIPS, GARY, and JERRY WHITE, **THE ATARI USER'S ENCYCLOPEDIA,** 267pp. BOOK CO84, 19.95
If you're a regular reader of **ANALOG**, then this book should appeal to you. An amazing compilation takes us into the world of Atari-computerland, very complete and up to date. This book makes enjoyable reading, along with being highly informative. Written with the aid of Jerry White, one of the most knowledgeable Atari-ites around.

LAMOITIER, JEAN-PIERRE, **BASIC EXERCISES FOR THE ATARI,** 251 pp. SYBEX83, 12.95
Contains many program listings in BASIC including chapters on games, financial computations and flowcharts. Mathematical programs cover geometry, integers, and statistics.

(submitted by Larry A. Campbell of St. Louis, Mo.)

THE BEST ATARI SOFTWARE. Spiralbound, 192pp. By the Editors of Consumer Guide, 83. .
Written by several people involved in the Atari computing world, this book covers what the authors feel is some of the better pieces of Atari-computer compatible software. The catagories include word processing, home, business, education, modem (networking) programs, utility/programing aids, and entertainment. Entertainment is broken down into strategy and arcade games. Again, this is one of those books that must be taken with a grain of salt, as opinions always vary greatly.

CANE, MIKE, **THE COMPUTER PHONE BOOK.** 451pp. NEW AMERICAN83, 9.95
Besides many pages devoted to the explanation of what online systems are and how to use them, over one hundred pages list phone numbers, locations, system baud rates, and background information on the systems. Though not written specifically for the Atari systems, this book may be a help to those who are actively involved in telecommunications.

ORWIG, GARY W., and *WILLIAM S. HODGES,* **THE COMPUTER TUTOR: ATARI HOME COMPUTER EDITION.** LITTLE BROWN83, 15.50
Applicable to the entire Atari computer series, this book contains listings of an educational nature.

HOGAN, THOM, **DISCOVER FORTH.** 146pp. OSBORNE83, 16.95
For the beginner or experienced programmer, this book acts as both a learning guide and reference tool into the learning and programming of FORTH.

HELLER, DAVID L., JOHN JOHNSON, and *ROBERT KURCINA,* **DR. C. WACKO'S MIRACLE GUIDE TO DESIGNING AND PROGRAMMING YOUR OWN ATARI COMPUTER ARCADE GAMES.** 235pp. ADDISON-WESLEY83, 24.95
This book has a fresh look as though it were written in an insane asylum. The book reeks with twisted graphics and humor, and certainly gets its intended point across: teaching you how to program your own arcade-style games. Chapters cover animation, player-missile graphics, zounds (sounds), character graphics and movement. The $24.95 price tag includes an Atari-compatible disk containing many of the useful demos in the book (listings are also provided in the text).

HELLER, DAVID and DOROTHY, **FREE SOFTWARE FOR YOUR ATARI.** 208pp. ENRICH/OHAUS83, 8.95
As the title suggests, this book shows where and how to get software through educational sources, magazines, users groups, and BBSs (Bulletin Board Systems). The chapter on BBSs, for instance, explains how they work, lists many sources to call, and how to set up your own system. Also included is a discussion with a couple of Atari executives. This guide to free software will set you back $8.95 though.

BANSE, TIMOTHY P., **HOME APPLICATIONS AND GAMES FOR THE ATARI.** 134pp. LITTLE BROWN83, 14.50
A collection of somewhat brief programs, all compatible with the 400/800 and entire XL series. These 24 programs consist of games, utilities and practical programs such as Jet Jockey, Heat Loss Survey, Ghost Town Vampire Girls, and Calorie Counter. Documentation accompanies each program, along with a string and variable table to assist you in modifying any of the listings. A cassette is available from the author for an additional $9.95.

DITLEA, STEVE, **HOME COMPUTER SOFTWARE GUIDE.** 196pp. OSBORNE84, 11.95
Covers all the popular computers along with the Atari. Eight chapters "review" and discuss software catagories such as entertainment, education, communications, finance, word processing and investment. Occasional screen display examples and charts assist in software selection, though the book is not all inclusive for the Atari computers.

SWANSON, PAUL S., **INTRODUCTION TO GRAPHICS ON YOUR ATARI COMPUTER.** 250pp. OSBORNE83, 16.95
Covering the entire Atari computer line, this book shows you how to create simple to complex graphics on the Atari. Chapters cover redefined character sets, animation, plotting, drawing, mapping modes, multiple screens, machine language, music and sound.

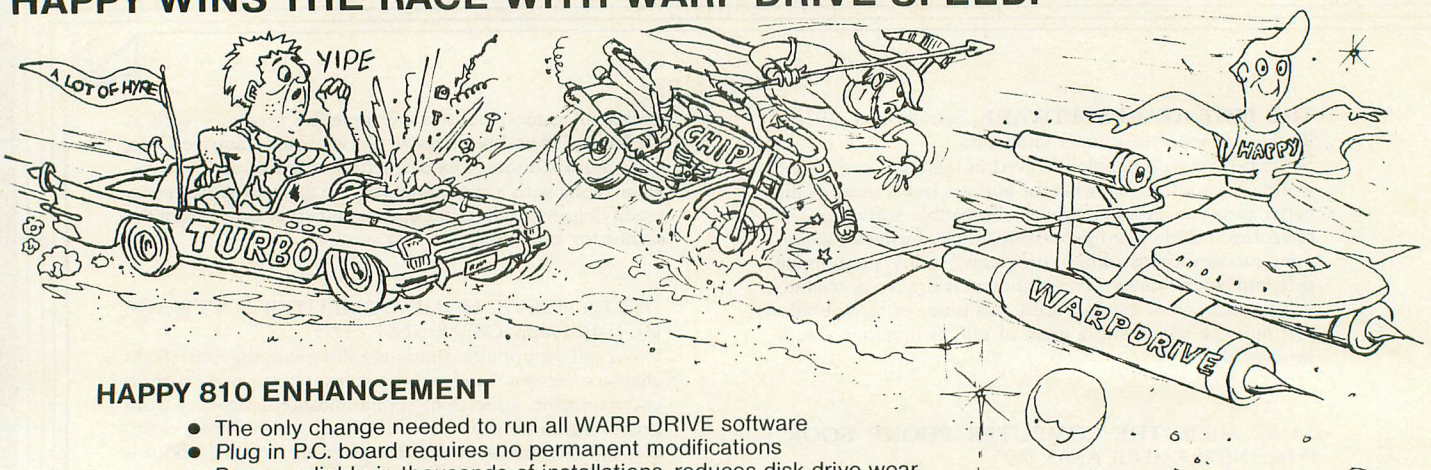
MAYER, NANCY KOZAK, Ed.D., **RAINY DAY ACTIVITIES FOR THE ATARI.** 156pp. RESTON84, 12.95
Written for 3-9 year olds, this plastic-spiralbound contains Atari BASIC written games for youngsters including those on numbers, words, names, and music. In addition, this book is laid out in a workbook format designed to be used along with the programs listed within.

WILLIS, JERRY, MERL MILLER and NANCY MORRICE. **THINGS TO DO WITH YOUR ATARI COMPUTER.** 230pp. NEW AMERICAN83, 3.95
Chapters cover Arts and Crafts, Fun and Games, Business and Professional Uses, Telecommunications and other topics. Many photos show screen displays and hardware, including a section on popular games from many different companies. A nice resource for those just purchasing an Atari home computer.

THE USER'S GUIDE TO ATARI (400/800/1200XL COMPUTERS, SOFTWARE & PERIPHERALS). 240pp. By the Editors of Consumer Guide, POCKET83, 3.95
For those who know little to nothing about their Atari computers, this paperback defines things like the OPTION, SELECT, START and BREAK keys, how to do simple math without programming, and what the cursor is. It also explains computer setup, disk and cassette storage, and a rather outdated section on Atari and compatible peripherals.

WEBER, JEFFREY R. and *STEPHEN J. SZCZECINSKI,* **USER'S HANDBOOK TO THE ATARI 400/800 COMPUTERS.** Many examples, 319pp. WEBER83, 13.95
A concise guide to all of the Atari computers, this book includes many charts and examples covering: Atari BASIC, graphics, DOS, and sound. The book also has chapters on the 810 disk drive, 850 interface module, and the older Atari printers.

HAPPY WINS THE RACE WITH WARP DRIVE SPEED!



HAPPY 810 ENHANCEMENT

- The only change needed to run all WARP DRIVE software
- Plug in P.C. board requires no permanent modifications
- Proven reliable in thousands of installations, reduces disk drive wear
- Comes completely assembled and tested, just plug in and use
- Full one year parts and labor guarantee, compatible with existing software
- High quality printed circuit board with gold connectors

NO ONE ELSE HAS THIS PERFORMANCE
Unenhanced whole disk (ATARI rev B format) read time: 112 seconds
Unenhanced whole disk (ATARI rev C fast format) read time: 89 seconds
ENHANCED 810 whole disk (any format) read time with standard software: 68 seconds
ENHANCED 810 whole disk (any format) read time with WARP DRIVE software: 43 seconds
Standard software whole disk write and verify time: 238 seconds
WARP DRIVE software whole disk write and verify time: 62 seconds

NEW HAPPY WARP DRIVE SOFTWARE

WARP SPEED HAPPY BACKUP PROGRAM
- Completely automatic: nothing to figure out, insert disks and press return
- Only program on the market guaranteed to backup any disk
- Can write to a blank disk: format write and verify in one operation
- Automatic program tracing: copies only the tracks that are used
- Efficient memory utilization: reduces the number of disk insertions
- Requires only one ENHANCED disk drive, backups will work on a standard drive

WARP SPEED MULTI DRIVE HAPPY BACKUP PROGRAM
- Same features as above plus support of multiple ENHANCED drives
- Can be used with up to 4 ENHANCED drives
- Source and all destination drives read and write in parallel
- Format write and verify 3 complete disks in less than 3 minutes

WARP SPEED HAPPY COMPACTOR PROGRAM
- Reduces the number of disks required to backup your library
- Combines up to 8 self booting disks into 1 disk with a menu
- Compacted disks run only on an ENHANCED drive
- Pays for itself by saving on disks
- Single or dual ENHANCED drive operation

HAPPY WARP DRIVE DOS
- Improves ATARI DOS 2.0S to use warp speed reading and write with verify
- Use all features of BASIC, PILOT, FMS, and DUP at top warp speed
- Warp speed I/O software module available separate from DOS

HAPPY WARP DRIVE SECTOR COPY PROGRAM
- Standard format whole disk read, write and verify in 105 seconds
- Use with sngle or dual drives, mix ENHANCED and NON-ENHANCED drives

HAPPY CUSTOMIZER PROGRAM (sold separately $99.95)
- Creates custom format disks of any specification
- Any type bad sector, duplicate sector numbers, or interleave
- Easy to use but requires an advanced level user to interpret the results

REVIEWED IN POPULAR MAGAZINES

A.N.A.L.O.G. COMPUTING—July/August 1983 *"...The installation instructions for the Happy 810 Enhancement are among the best I have ever seen. ...The Happy 810 Enhancement is one of the most powerful hardware modifications available to ATARI computer owners."*

ANTIC—July 1983 *"The difference between a normal ATARI 810 disk drive and one equipped with Happy is like the contrast between mass transit and the automobile. A car costs you more initially, but improves the quality of your life. Similarly, if you use your disk drive a lot, installing Happy will markedly enhance your programming life."*

SPECIAL SUGGESTED RETAIL PRICE BEFORE FEBRUARY 28, 1984: *Get the HAPPY 810 ENHANCEMENT with the single and multi drive HAPPY BACKUP PROGRAM, plus the HAPPY COMPACTOR PROGRAM, plus the HAPPY DRIVE DOS, plus the HAPPY SECTOR COPY, all with WARP DRIVE speed, including our diagnostic for $249.95. Existing registered ENHANCEMENT owners may upgrade to WARP DRIVE speed for $15.00 with no hardware changes!*

Price includes shipping by air mail to U.S.A. and Canada. Foreign orders add $10.00 and send an international money order payable through a U.S.A. bank. California orders add $16.25 state sales tax. Cashiers check or money order for immediate shipment from stock. Personal checks require 2-3 weeks to clear. Cash COD available by phone order and charges will be added. No credit card orders accepted. ENHANCEMENTS for other ATARI compatible drives coming soon, call for information. Please specify -H model for all drives purchased new after February 1982, call for help in ENHANCEMENT model selection. Dealers now throughout the world, call for the number of the dealer closest to you.

ATARI 810 is a registered trademark of Atari, Inc.

HAPPY COMPUTERS, INC. ● P. O. Box 1268 ● Morgan Hill, California 95037 ● (408) 779-3830

CIRCLE #110 ON READER SERVICE CARD.

Communications for the Handicapped

16K Cassette or 24K Disk

by Michael Long

Imagine yourself flat on your back, unable to move a muscle. Imagine yourself fully aware of all that happens around you, but unable to respond or communicate in any way. Sounds like the opening of a *Twilight Zone* episode, doesn't it? Yet people find themselves in this condition every day as a result of an automobile accident or other injury. This article will show you how you can help to end this isolation in your own community, with the help of your Atari.

The **Puff-and-Sip Communication Program** in **Listing 1** will allow anyone who can operate two switches to create text on a television screen using an Atari computer. The switches can be actuated by any means you choose, but for demonstration purposes we will use a joystick plugged into port one. Later I will describe a simple input device that the handicapped can use, but first let's get the program up and running.

Type in **Listing 1**, save it and RUN it. You will see a title screen for a few seconds while the computer gets everything set up. Then you will see a screen that allows you to vary the speed of the program. Pull back on the joystick and you will see a square cursor moving across the numbers at the bottom of the screen. Speed 1 is very slow. Each succeeding number doubles the speed. I have been able to operate with a joystick at speed 4, but speed 3 seems the best compromise with other devices. When the cursor is on 3, push forward on the stick.

You should now see a black screen with a cursor in the upper left corner and a menu across the bottom that looks like this:

S TAOSWIHCBFPMR P

There is another cursor around the "S" on the left. The green "S" stands for space. If you push on the stick a space will be added to the message and the message cursor will move one space to the right. Try it.

If you pull on the stick the menu cursor will move to the right across the list of orange letters. If you move the cursor off the right edge of the screen it will wrap to the left edge and the menu will "flip" to show the rest of the alphabet. Choose one of the orange letters and push on the stick. The letter you have chosen will be added to the message and the message cursor will move one space to the right, ready for another character.

Now that you know how to put characters on the screen, let's look at the rest of the menus. Pull on the stick until the cursor is on the green "P" on the far right and push. The menu "flips" to a punctuation menu that looks like this:

S .,;:?!‴ FB C N

The green "S" is for "space," and the punctuation marks work just like the letters. The green "F," "B," and "C" are commands. Put the cursor on the "F" and the message cursor moves forward one space each time you push the stick. Put the cursor on the

"B" and the cursor moves backward one space for each push. The "C" clears the screen, but only after giving you a chance to back out if you selected it by mistake.

Again, if you move the cursor off the right edge it will wrap to the left and the rest of the punctuation menu will appear. Put the cursor on the "N" on the right and the number menu appears.

The number menu also includes several puncuation marks that might be useful with numbers; it looks like this:

S 0123456789$%. L

Put the cursor on the "L" and push, and you are back to the letter menu.

Menu selection.

You may have noticed that you don't always get the same menu after you enter a character. Here's how the menu selection works and why.

The letter menus are held in one long string (DIG$) and selected according to the last character entered. You may have noticed that you can construct a message fairly quickly if the letters you want are near the start of the menu, but if they are near the end it can take quite a while. There are tables available in cryptography books that rank the alphabet in order of probability as the initial letter of a word, or following any particular letter. This is how the letter menus are arranged. There is a 90% probability that the required letter will be on the first page of the menu and a 50% probability that it will be in the first five characters.

There are other peculiarities in the way that the menus change. These are based on common sense rather than probability:

(1) After any character is selected, the cursor will always return to the "space." This is based on the assumption that there will be more spaces in a message than any other single character.

(2) After any punctuation mark is selected, the menu returns to the initial letter menu. Puncuation marks seldom follow each other. They are usually followed by a space.

(3) When a cursor control command ("F" or "B") is selected, the cursor remains on the same character. This makes it easier to move through the message to make corrections.

(4) When a number is selected, the letter menu does not automatically appear. Numbers are often used in groups (e.g.; $125.62).

Puff-and-sip.

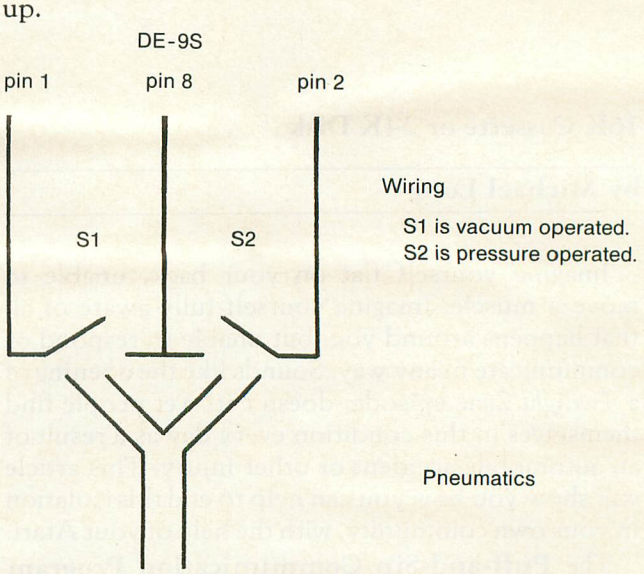
Now that you know how the program works, how could a person who cannot move his hands operate it? He (or she) certainly can't use a joystick. One answer is a "puff-and-sip" switch. Here's how it works and how to make one. You will need the following parts:

1. Two air-actuated microswitches, one pressure-actuated and one vacuum-actuated. These should be normally open.
2. One DE-98 joystick connector.
3. Twelve feet of 22-gauge wire.
4. Four feet of surgical tubing.
5. One Y-type connector for the tubing.
6. One small box to hold the switches.

The air-actuated microswitches can be a little hard to find. If you can't find them locally, contact:

Dumont Instrumentation, Inc.
75 Mall Drive
Commack, NY 11725
(516) 864-8500

Connect one side of the vacuum-activated switch to pin 1 of the DE-9 connector, and one side of the pressure activated switch to pin 2. Connect the other side of both switches to pin 8 (GND). Connect the air inputs of both switches to the "arms" of the Y-connector with tubing. Finally, connect a few feet of tubing to the base of the Y and box the whole thing up.



Check the connections to make sure they are correct and plug the box into controller port 1. If you blow into the tube, the effect is the same as pulling back on the stick. If you inhale on the tube, you push on the stick.

The **Puff-and-Sip Communication Program**, as listed here, will allow a quadraplegic patient with minimal computer skills to talk to a friend or therapist. It is certainly not the only use for this idea. With a little more work, the program could drive a printer, or be the basis for a word processor. It could be used as an input subroutine for a modem program, or even for an operating system that would allow the full power of the computer to be used without any keyboard input. It is offered here as an example of what you can do with average programming skills to improve the quality of life for the handicapped people in your community.

I would be interested in hearing from anyone with interest in this area. I can be contacted at (918) 834-0503, or on SIG*ATARI, CIS # 72435,35.

Partial variable list.

CURS — The horizontal position of the message cursor.

CURSLOC — The horizontal position of the menu cursor.

CURSIMG — The address where the cursor image is stored.

DIG$ — The letter menus in probability order.

MENU$ — The menu currently in use.

MINDEX — A pointer to the first or second half of the menu.

NUM$ — The number menu.

PMBASE — The address of the start of player/missile memory.

PUFF — The value of STICK(0) to be interpreted as a puff.

PUNC$ — The punctuation menu.

SIP — The value of STICK(0) to be interpreted as a sip.

T$ — The menu entry for the next menu.

VERT — The address of the P/M vertical move routine.

VP — The vertical position of the cursor as an offset from PMBASE.

How it works.

Lines 100-230 (Main Loop). Prints the current menu, reads STICK(0), updates the menu cursor, and calls the subroutines to get characters from the current menu.

Lines 300-340 (Get Letters). Call the subroutine to print the current character to the screen and sets MENU$ equal to the next menu.

Lines 400-530 (Get Punctuation). Calls the subroutines to print the current character to the screen, move the message cursor backward or forward, or clear the screen. Sets MENU$ equal to the next menu.

Lines 600-630 (Get Numbers). Calls the subroutine to print the current character to the screen and sets MENU$ equal to the next menu.

Lines 700-760 (Move Message Cursor). The subroutine to move the message cursor. Uses a ML routine for vertical moves.

Lines 800-850 (Clear Screen). Prompts for confirmation, clears the current message, and moves the message cursor to the upper left.

Line 1000 (Print Character). Prints the current character to the current message cursor position and calls the message cursor move routine.

Lines 20000-20120 (Set Speed). Prints the speed selection screen, reads STICK(0), and sets SPEED.

Lines 30000-30010 (Begin Initialization). Moves RAMTOP down four pages to make room for P/M memory, sets the left margin to zero, and sets graphics mode 2.

Lines 30100-30110 (Customize Display List). Changes the text window to large text (GR. 2). This allows using PRINT #6 to print to the message window, and PRINT to print to the menu window.

Lines 30200-30230 (Title Page). Prints the title page.

Lines 30300-30350 (Set-up P/M Graphics). Pokes the cursor images into P/M memory and sets the initial colors and horizontal positions.

Lines 30400-30720 (Initialize Variables). Sets the initial values of variables.

Lines 30800-30910 (Cursor Vertical Move Routine). Sets up the ML routine to move the cursor vertically. The routine was written by David H. Markley and published in *COMPUTE!'S First Book of Atari Graphics*, page 154. □

```
1 REM ******************************
2 REM *            ATARI           *
3 REM *       Puff-and-Sip         *
4 REM *       Communication        *
5 REM *          Program           *
6 REM *        Version 2.0         *
7 REM *          (c)1983           *
8 REM *       Michael Long         *
9 REM ******************************
10 GOSUB 30000:GOSUB 20000:? #6;"K";:P
OKE 705,8
95 REM * MAIN LOOP
100 ? " s ";MENU$(MINDEX,MINDEX+12);"
";T$;:? :? :?
110 POKE 77,0:FOR DELAY=1 TO SPEED:NEX
T DELAY:SOUND 0,0,0,0:LOOP=110
120 A=STICK(0):IF A<>PUFF THEN GOTO 17
0
130 IF CURSLOC=184 AND MINDEX=1 THEN C
URSLOC=56:MINDEX=14:LOOP=100:GOTO 160
140 IF CURSLOC=184 AND MINDEX=14 THEN
CURSLOC=56:MINDEX=1:LOOP=100:GOTO 160
150 CURSLOC=CURSLOC+8
160 POKE 53248,CURSLOC:GOTO LOOP
170 IF A<>SIP THEN GOTO LOOP
180 IF CURSLOC=56 THEN ? #6;" ";:MENU$
=DIG$(1,26):T$="p":MINDEX=1:LOOP=100:G
OSUB 700:GOTO 230
190 IF CURSLOC=64 OR CURSLOC=176 THEN
GOTO 230
200 IF T$="p" THEN GOSUB 300:GOTO 230
210 IF T$="n" THEN GOSUB 400:GOTO 230
220 IF T$="1" THEN GOSUB 600
230 SOUND 0,100,10,8:POKE 53248,CURSLO
C:GOTO LOOP
295 REM * GET LETTERS
300 IF CURSLOC=184 THEN MENU$=PUNC$:T$
="n":GOTO 330
310 GOSUB 1000
320 C=(ASC(MENU$(B,B))-64)*26:MENU$=DI
G$(C+1,C+26)
330 CURSLOC=56:MINDEX=1:LOOP=100
340 RETURN
395 REM * GET PUNCTUATION
400 IF CURSLOC=184 THEN MENU$=NUM$:T$=
"1":GOTO 520
410 IF MINDEX=1 THEN GOTO 430
420 GOSUB 1000:GOTO 510
```

Getting down to Atari® BASICS

Basic Atari® BASIC

(Coan/Kushner) A complete guide to Atari BASIC adapted from the best selling **Basic Apple BASIC.** Contains over 80 programs–all conveniently indexed–that explain and offer hands-on practice in BASIC programming.
#6526, $14.95.

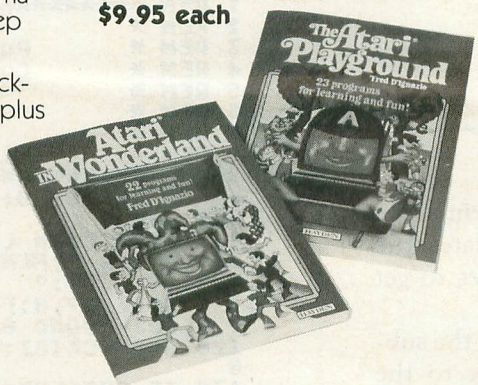
Qwerty's Alphabet Adventure
Qwerty's Number Adventure

(Shadow Lawn Press) Introduces your children to Qwerty–a fun-loving caterpillar who serves as a guide through the alphabet and number concepts. Colorful animation and a fun storyline will keep children entranced for hours.

Available as book/software package only, containing 1 cassette plus documentation.
Alphabet #6525
Number #6527
$14.95 each

The Atari® Playground
Atari® in Wonderland

(D'Ignazio) Imaginative way to teach microcomputing to youngsters. The Playground contains 23 unique programs in which children play in a spelling bee, draw with a computer crayon, chase wild letters and ghosts. Wonderland introduces children to word and number skills, angle measuring while riding a computer roller coaster, and counting in French and Spanish. Includes plenty of suggestions and tips for making each program even more fun!
Playground #5770
Wonderland #5771
$9.95 each



I Speak BASIC to My Atari®

(Jones) Part of the widely acclaimed **I Speak BASIC** series, the student text offers a gentle introduction to BASIC programming and operation of the Atari. Includes plenty of programming exercises and examples.

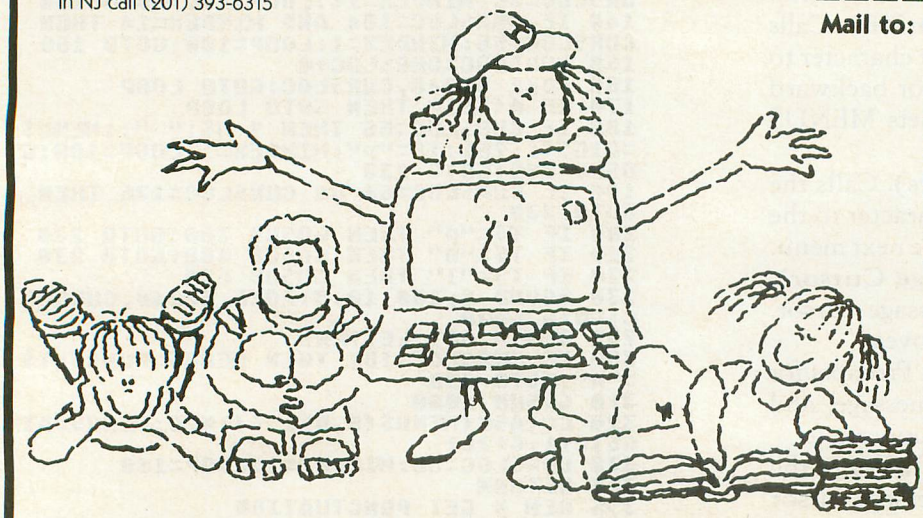
Teacher's Manual available for educators. Order on school letterhead. A 20% discount available to educators.
Student text #6178, $9.75.
Teacher's manual #6168, $18.75.

Stimulating Simulations for the Atari®

(Engel) Another in the popular **Stimulating Simulations** series. Provides all the unique challenges and spine-tingling chills that have made this series a favorite. Includes "Devil's Dungeon," "Lost Treasure," "Forest Fire," "Monster Chase," and "Diamond Thief."
#5197, $7.50.

HAYDEN

Order by Phone
1-800-631-0856
In NJ call (201) 393-6315



Mail to: Dept. AN44 • Hayden Book Company
10 Mulholland Dr. • Hasbrouck Hts., NJ 07604

Please send me the book(s) indicated below by code number. If I am not completely satisfied I may return the book(s) undamaged, within 10 days for a complete refund. **I am enclosing $2.00 to cover postage and handling.**

☐ Enclosed is my check or money order
☐ Bill my ☐ Visa ☐ MasterCard

Name
Address
City
State/Zip
Visa/MasterCard # Expires
Signature

Residents of NJ and CA must add sales tax.
Prices subject to change.

Atari is a registered trademark of Atari, Inc., a division of Warner Communications and is not affiliated with Hayden Book Company.

CIRCLE #111 ON READER SERVICE CARD.

```
430 IF CURSLOC=136 OR CURSLOC=160 THEN
 GOTO 530
440 IF CURSLOC=144 AND PEEK(85)<>19 TH
EN POKE 85,PEEK(85)+1:GOSUB 720:GOTO 5
30
450 IF CURSLOC=144 AND PEEK(85)=19 THE
N POKE 84,PEEK(84)+1:POKE 85,0:GOSUB 7
00:GOTO 530
460 IF CURSLOC=152 AND PEEK(85)<>0 THE
N POKE 85,PEEK(85)-1:GOSUB 740:GOTO 53
0
470 IF CURSLOC=152 AND PEEK(85)=0 AND
PEEK(84)=0 THEN GOTO 530
480 IF CURSLOC=152 AND PEEK(85)=0 THEN
 POKE 84,PEEK(84)-1:POKE 85,19:GOSUB 7
50:GOTO 530
490 IF CURSLOC=168 THEN GOSUB 800:GOTO
 510
500 GOSUB 1000
510 MENU$=DIG$(1,26):T$="p"
520 CURSLOC=56:MINDEX=1:LOOP=100
530 RETURN
595 REM * GET NUMBERS
600 IF CURSLOC=184 THEN MENU$=DIG$(1,2
6):T$="p":LOOP=100:GOTO 620
610 GOSUB 1000
620 CURSLOC=56:MINDEX=1
630 RETURN
695 REM * CURSOR MOVE ROUTINES
700 IF PEEK(84)=10 THEN POKE 84,9:POKE
 85,19:GOTO 760
710 IF PEEK(85)=0 THEN GOTO 730
720 CURS=CURS+8:GOTO 760
730 CURS=48:X=USR(VERT,CURSIMG,PMBASE+
VP,PMBASE+VP+8):VP=VP+8:GOTO 760
740 CURS=CURS-8:GOTO 760
750 CURS=200:X=USR(VERT,CURSIMG,PMBASE
+VP,PMBASE+VP-8):VP=VP-8
760 POKE 53249,CURS:RETURN
795 REM * CLEAR SCREEN ROUTINE
800 SOUND 0,100,10,8:POKE 704,0
810 ? "    ARE YOU SURE?     PUFF=YES
 SIP=NO":? :? :FOR DELAY=1 TO SPEED/2:
NEXT DELAY:SOUND 0,0,0,0
820 A=STICK(0):IF A=PUFF THEN POKE 704
,8:GOTO 850
830 IF A=SIP THEN POKE 704,8:RETURN
840 GOTO 820
850 ? #6;"K";:CURS=48:POKE 53249,CURS:
X=USR(VERT,CURSIMG,PMBASE+VP,PMBASE+65
6):VP=656:RETURN
995 REM * PRINT CHARACTER
1000 B=MINDEX+(CURSLOC-76)/8:? #6;CHR$
(ASC(MENU$(B,B))+128);:GOSUB 700:RETUR
N
19995 REM * SET SPEED
20000 ? #6;"K":? #6;"      SET SPEED
                "
20010 ? #6;"      puff=change
                "
20020 ? #6;"      sip =set
                "
20030 POSITION 0,9:? #6;" SLOW
   FAST"
20040 ? "       1 2 3 4 5":? :?
20050 POKE 704,8
20060 FOR DELAY=1 TO 100:NEXT DELAY:A=
STICK(0):IF A<>PUFF THEN GOTO 20090
20070 CURSLOC=CURSLOC+16:IF CURSLOC>16
0 THEN CURSLOC=96
20080 POKE 53248,CURSLOC
20090 IF A<>SIP THEN GOTO 20060
20100 B=ABS(CURSLOC-160)/16:SPEED=25*2
^B
20110 SOUND 0,100,10,8:CURSLOC=56:A=ST
ICK(0):IF A=15 THEN POKE 53248,CURSLOC
:RETURN
20120 GOTO 20110
29995 REM * INITIALIZATION
30000 A=PEEK(106)-4:POKE 106,A
30010 POKE 82,0:GRAPHICS 2
30095 REM * CUSTOMIZE DISPLAY LIST
30100 DL=PEEK(560)+PEEK(561)*256
30110 POKE DL+15,71:POKE DL+18,7:POKE
DL+19,65:POKE DL+20,PEEK(DL+22):POKE D
L+21,PEEK(DL+23)
30195 REM * TITLE PAGE
30200 ? #6;"K":? #6;"        ATARI"
30210 ? #6;"    PUFF-AND-SIP":? #6;"
  COMMUNICATION"
30220 ? #6;"       PROGRAM":? #6;" "
30230 ? #6;"       (C)1983":? #6;"    M
ICHAEL LONG"
30295 REM * SET UP P/M GRAPHICS
30300 POKE 54279,A:PMBASE=A*256
30310 FOR I=PMBASE+512 TO PMBASE+768:P
OKE I,0:NEXT I
30320 RESTORE 31000:FOR I=PMBASE+608 T
O PMBASE+615:READ A:POKE I,A:NEXT I
30330 RESTORE 31000:FOR I=PMBASE+656 T
O PMBASE+663:READ A:POKE I,A:NEXT I
30340 POKE 53248,128:POKE 53249,48:POK
E 704,0:POKE 705,0
30350 POKE 559,46:POKE 53277,2:POKE 62
3,4
30395 REM * INITIALIZE VARIABLES
30400 DIM DIG$(702),PUNC$(26),NUM$(26)
,MENU$(26),T$(1)
30410 CURSLOC=128:MINDEX=1:CURS=48:VP=
656:PUFF=13:SIP=14
30420 DIG$(1,26)="TAOSWIHCBFPMRELNDUGY
JVQKXZ":REM * INITIAL LETTERS
30430 DIG$(27,52)="NTRLSIDCYMGPVBUKFWX
ZEHJAOQ":REM * A *
30440 DIG$(53,78)="ELYOAURISBJMTCVDEGH
KNPQWXZ":REM * B *
30450 DIG$(79,104)="OEHATIRLUKCYSMNDQB
FGJPVWXZ":REM * C *
30460 DIG$(105,130)="EIAOUSRYDMGLVNWJC
BTFHQKPXZ":REM * D *
30470 DIG$(131,156)="RNSDALECTMVWPIXFG
YUOBKQHJZ":REM * E *
30480 DIG$(157,182)="OIERFATULYSMNBCDG
HJKPQVWXZ":REM * F *
30490 DIG$(183,208)="ERHOAIUNSTLGYMDBC
FJKPQVWXZ":REM * G *
30500 DIG$(209,234)="EAIOTRUYNSLMCWDBH
FGJKPQVXZ":REM * H *
30510 DIG$(235,260)="NSTOCLDEARMGVFPBZ
KXIUQHYJW":REM * I *
30520 DIG$(261,286)="UOEAIRBCDFGHJKLMN
PQSTVWXYZ":REM * J *
30530 DIG$(287,312)="EISNAHLOKYRBDFMTU
CGJPQVWXZ":REM * K *
30540 DIG$(313,338)="EILAYODSUTGFMKVPR
CBWNHJQXZ":REM * L *
30550 DIG$(339,364)="EAIOPMBUSYNCRFLTV
DGHJKQWXZ":REM * M *
30560 DIG$(365,390)="DTGESICAONYUMFLVK
WHRJQBPZX":REM * N *
30570 DIG$(391,416)="NRFUMLTSWVPDOCBIA
GKYEHJXZQ":REM * O *
30580 DIG$(417,442)="REOALPUITHSMYCNBD
FGJKQVWXZ":REM * P *
30590 DIG$(443,468)="UABCDEFGHIJKLMNOP
QRSTVWXYZ":REM * Q *
30600 DIG$(469,494)="EIOASTNYMDRKUGCLV
PBFHWXJQZ":REM * R *
30610 DIG$(495,520)="TEISOAUHPCLMKYWDR
FNBQGJVXZ":REM * S *
30620 DIG$(521,546)="HEIOARSTYULWMCNFV
ZBPDGJKQZ":REM * T *
30630 DIG$(547,572)="NRTSLCEBPGAMIDFUO
KVYHXZJQW":REM * U *
30640 DIG$(573,598)="EIAOYDSMUVBCFGHJK
LNPQRTWXZ":REM * V *
30650 DIG$(599,624)="AIEHOSNRLDYTBCFGJ
KMPQUVWXZ":REM * W *
30660 DIG$(625,650)="PTECIAXHMOBDFGJKL
NQRSUVWYZ":REM * X *
30670 DIG$(651,676)="EOSIAMPRLNTUCWDYB
KZFGHJQVX":REM * Y *
30680 DIG$(677,702)="EAIOZLUBCDFGHJKMN
PQRSTVWXY":REM * Z *
30690 MENU$=DIG$(1,26)
30700 PUNC$(1,6)=".,;:?!":PUNC$(7,7)=C
HR$(34):PUNC$(8,26)="'  fb c()+-*/=<>&@
#^"
30710 NUM$="0123456789$%.0123456789$%.
"
30720 T$="p"
30795 REM * CURSOR VERTICAL MOVEMENT
30800 VERT=1536
30810 RESTORE 31100:FOR I=0 TO 43:READ
 A:POKE VERT+I,A:NEXT I
```

```
30895 REM * CURSOR IMAGE ROUTINE
30900 CURSIMG=VERT+44
30910 RESTORE 31200:FOR I=0 TO 8:READ
A:POKE CURSIMG+I,A:NEXT I
30920 RETURN
30995 REM * CURSOR DATA
31000 DATA 255,129,129,129,129,129,129
,255
31095 REM * VERTICAL MOVE DATA
31100 DATA 104,162,5,104,149,220,202,1
6,250,198,220,198,222,160,0,177,224,17
0
31110 DATA 168,165,223,240,9,169,0,145
,222,136,208,249,138,168,165,221,240,7
,177,224,145,220,136,208,249,96
31195 REM * DATA FOR CURSOR IMAGE
31200 DATA 8,255,129,129,129,129,129,1
29,255
```

•

CHECKSUM DATA

(See p. 30)

```
1 DATA 255,115,185,694,793,837,47,293,
271,709,510,265,726,191,916,6807
140 DATA 29,137,441,84,509,366,774,778
,690,571,932,814,794,656,352,7927
340 DATA 597,733,448,819,869,135,778,7
57,350,564,885,665,795,455,353,9203
530 DATA 598,933,100,800,197,600,307,5
49,150,429,875,441,773,828,171,7751
800 DATA 925,633,569,883,729,628,721,7
76,874,673,889,757,175,93,208,9533
20060 DATA 640,804,219,221,324,887,189
,671,614,262,47,840,63,161,839,6781
30210 DATA 501,681,258,299,37,988,627,
646,517,873,353,238,456,463,249,7186
30440 DATA 264,441,883,838,931,900,911
,869,901,918,915,947,926,935,901,12480
30590 DATA 938,993,923,890,972,984,979
,968,959,972,874,199,332,662,113,11758
30800 DATA 58,230,407,991,210,57,85,59
1,113,861,940,458,667,5668
```

•

A LIMITED NUMBER OF BACK ISSUES IS AVAILABLE.

ISSUES

2, 8, 11, 12, 13, 14 & 15

$4.00 EACH

ANALOG BACK ISSUES
P.O. BOX 23
WORCESTER, MA 01603

***ATTENTION* ATARI DISK DRIVE OWNERS**

Back up your valuable software.

THE CHIP
THE CHIP with Archiver/Editor Software for the Atari 810 and 1050 disc drives. Includes Disassembler & Sector Editor. Includes Custom Formatter and Mapper. Backs up virtually any disk. $129.95 POST PAID
Available soon for PERCOM and other Atari compatible disc drives.

HAPPY 910 ENHANCEMENT
Complete with Warp Speed software package. Plug-in installation — no soldering. Backs up any disc.
Regular Price $249.95
LIMITED SPECIAL OFFER $199.95
Soon available for other disc drives.

HAPPY OWNERS
Update your enhancement with Happy Version Archiver/Editor. Makes Happy drives compatible with the chip.
$39.95 POST PAID

All software for the Atari DISCOUNTED 30% or MORE.
Discounts on all Atari compatible hardware.
Send for free brochure on any of the above or for details on our software discounts.

SOUTHERN SOFTWARE
A DIVISION OF SOUTHERN SUPPLY COMPANY
1879 RUFFNER ROAD BIRMINGHAM, AL 35210
24 HOUR PHONE 205-956-0986
Order before 11 A.M. for same day shipping.

CIRCLE #112 ON READER SERVICE CARD.

NO TAXES?

1984 TAX PLANNER

ATARI® 400/800/XL

DON'T GUESS-TA-MATE!
Now you can instantly know your exact tax situation and manipulate it throughout the year.

The TAX PLANNER allows you to plan and update your tax position throughout the year. Do you want to know the exact tax impact of buying a home, selling stock, contributing to an IRA, buying rental property etc. . .? Can also be used to estimate quarterly tax payments.

The TAX PLANNER includes these schedules and more

Form 1040	Schedule G - Income Averaging
Schedule A - Itemized Deductions	Schedule W - Married Deduction
Schedule B - Interest & Dividends	Form 2106 - Employee Expense
Schedule C - Business Income	Form 2119 - Sales of Residence
Schedule D - Capital Gains/Losses	Form 3906 - Moving Expense
Schedule E - Supplemental Income	Form 4684 - Casualties & Thefts

$49.95

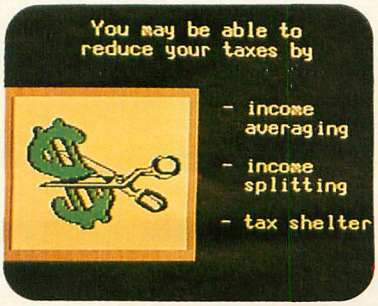
MasterCard VISA

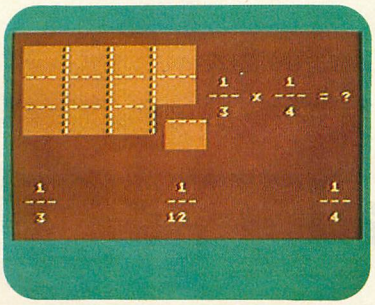
MiccaSoft
406 Windsor
New Braunfels, TX 78130
(512) 629-4341
Requires 48K and Disk Drive
VISA/MC/Check/Money Order
Dealer Inquiries Welcome - Texas Residents add 5% Sales Tax

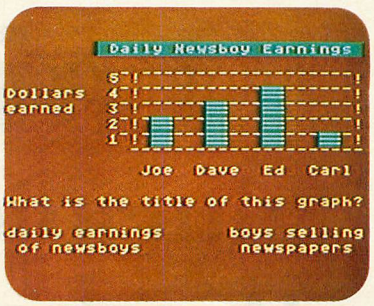
CIRCLE #113 ON READER SERVICE CARD.

NEW! For Your Atari Computer

TRS 80 Color, I, III, 4 or Apple Computer







Over 1000 Programs with Full Time Audio Narration, Pictures & Text!

Interactive Tutorial Programs

All along you've heard there isn't much of a selection of low cost microcomputer courseware. Well, do you want to learn Physics, Psychology, or Philosophy? Accounting or Auto Mechanics? Sociology, Supervision or Statistics? Economics or Electronics? How about English as a Second Language? Would you like to teach your children Math or Reading? If you do, then we have the educational programs for you—and at affordable prices.

It's true there aren't many companies that offer full-length courses in subjects other than reading and arithmetic, and what is offered seems to be drills, tests, games, or simulations. What you really want is a course that covers the subject with, say, 16 full-length lessons called tutorial programs, where you interact with an expert programmer backed by a staff of experts. That's exactly what we have.

Is there something wrong with our software? Well, we don't ask you to enter your name so we can drop it into some later text. We don't ask you to type in your answer and refuse to accept it if it's not spelled just right. And, we don't branch around a lot when you make an error. Our programs simply let you know if you're wrong by proceeding only when you select the right multiple-choice answer.

This proven learning-by-positive-reinforcement method lets you proceed quickly and smoothly through the programs, without a lot of cute tricks. But, if you're a devoted computerist or game freak you may be disappointed at the lack of motion in most of the pictures, or our special visual effects. We have some dandies, like the mushroom cloud that rises over Hiroshima in our History series, but your learning is not distracted by needless special effects.

OVER 1000 PROGRAMS

We have 64 courses of 16 half-hour programs: 1024 programs! All are easy-reading, upper- and lower-case. All are in color. All are illustrated by frequent graphics composed of special and regular characters. And, best of all, every frame of every program is accompanied by high-quality, full-time audio narration by professional voice talent. David Stanton, James Mathers, Pam Barrymore. Recorded and played back, not synthesized or digitized!

— If your child is having trouble with Reading or Math, or if he or she is exceptional, and could benefit from professionally programmed lessons or courses, then you need our educational programs for your Atari, Apple, or TRS 80 to help your child.

— If you want to build your math skills, we have programs on Numbers and their meanings, Addition, Subtraction, Multiplication, Division, Fractions, Decimals, Percents, Angles, Graphs, Word Problems, Algebra, Statistics. There are ten series in all, 160 math programs. Again, all with color, pictures, and a friendly tutor's voice—all the time.

— If your verbal skills need polishing, try our Talk & Teach programs on the alphabet, spelling skills, and every level of vocabulary; Sight Words or Learn by Phonics; develop Reading Comprehension skills from stories and articles. There's even a series on the Great Classics.

— For self-development, try a 16-program course in Economics or Psychology. Or Supervision, Sociology, Counseling (Personal, School, or Employee), Writing, Business, Philosophy, Government, World History, U.S. History, or Accounting.

— For vocational skills, study our 16-program courses in Electronics, Fluid Power, Auto Mechanics, Shop, Carpentry, Construction, Meat Processing, Military Skills (64 programs), and there are many more.

There are 832 programs for your Atari, Apple and TRS 80 Mod I/III/4 and 160 for the Color Computer. Send for a free catalog.

Would you like to do something more than play games on your computer? Tired of squeaks and robotics and want to hear a human voice? Do you want to further your education or help your children along in theirs? Now you can. For just **$8.80**, we'll send you one cassette with 2 programs from the course of your choice, 100% guaranteed to work in your Atari, Apple, or TRS 80 computer, and you can try us out. Better yet, get one full course of 16 programs on 8 cassettes for only **$59.90**. Atari Master Cartridge, **$9.95**; Apple T/T Board, **$99.00**; and T/T Player, **$79.00**. That's less than books and tuition for most college classes. And we offer a 10 day, 100% exchange allowance. Does your alma mater? Send us your check, or call us at **1-800-654-3871** with your Visa/Master Card number. We'll send your educational software pre-paid. Please allow 15 days delivery. Or see your computer dealer. He may have some of our courses in stock. You've got nothing to lose but your games!

We're Your Educational Software Source.

Course	No. of Programs
Reading	256
Mathematics	128
Comprehension	48
History	32
Algebra	16
Spelling	16
Government	16

16 Programs in each of the following:
Carpentry-Electronics-Health Services
Office Skills-Statistics-First Aid/Safety
Economics-Business-Accounting-Psychology
And Many More!

For your Atari 400/600/800/1200, you will need the Atari cassette recorder and the Dorsett 4001 Educational Master Cartridge, $9.95. For your Apple II, you will need the Dorsett M402 T/T plug-in board, $99.00, and the M401 stereo cassette player, $79.00. All programs listed are available for TRS 80, I, III, 4, which require the M203 speaker converter, $99.00, and 401 stereo cassette player, $79.00.

Send for a catalog of over 1000 programs for Atari, TRS 80, Apple, etc.

For more information, write or call: **TOLL FREE 1-800-654-3871**

Dealer inquiries welcome

VISA MasterCard

DORSETT
Educational Systems, Inc.
P.O. Box 1226, Norman, OK 73070

CIRCLE #114 ON READER SERVICE CARD.

Griffin's Lair

Educational Programs Review

by Braden E. Griffin, M.D.

This month we are going to look at a couple of programs which deal with "computer literacy." This is a term frequently bantered about, and I am not really sure what it means. Kind of like the coach who was talking about one of his star football players and said, "He doesn't know the meaning of the word fear. Come to think of it, he doesn't know the meaning of a lot of words!" I do know that I am sick of those commercials which disdain "computerese." What a terrible word. Well, if you are one of those who think a benchmark is a basketball stigmata consisting of an indentation of the posterior aspect of the thighs, a result of endlessly waiting to enter the game, you are in big trouble.

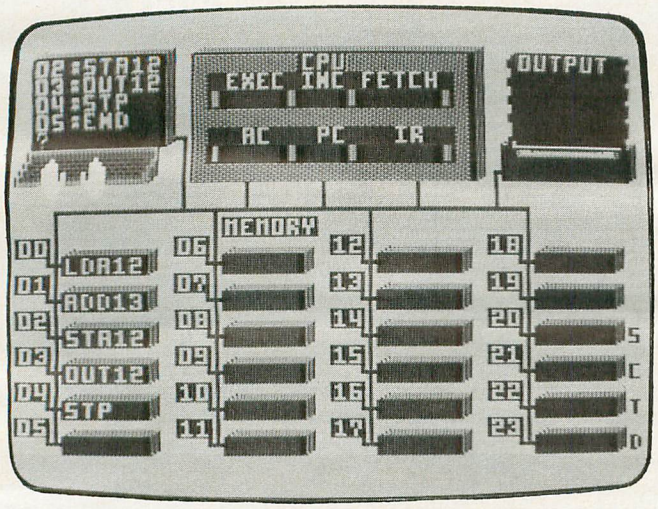
Anyway, here are two programs which may help you along the road to being more computer literate, no matter what your age or bias. They are both "mint," and I hope my reviews do them justice.

SIMULATED COMPUTER II
Carousel Software Inc.
877 Beacon Street
Boston, MA 02215
32K Disk/24K Cassette $29.95

Every month, ANALOG has at least one article with a game where the reader can type in a BASIC program listing (full of DATA statements), save it on disk or cassette, add a program which automatically boots the program, and eventually end up with an assembly language program which does not require BASIC. This allows the programmer to provide us with real-time simulations and arcade-style games which run thousands of times faster than BASIC. At the end of each of these articles is included the assembly language listing for those interested in seeing how the program works. I am interested. I look, but I do not see. What I see is a lot of letters, a few numbers, and a plethora of semicolons. No problem. I will just learn about assembly language. Since this language speaks to the computer more directly than BASIC, i.e., does not require a translation every time a command is executed, understanding it should help me better understand how the computer works. So, I borrowed a few books and copied a few articles about assembly language and sat down in front of a fire ready to expand my horizons. The first thing I encountered was an explanation of numbering systems. I once studied the binary system in school and I understand it. I can even convert a "real" number to a binary number, given enough time. (As you all know, we use the decimal system with the base 10 because we have ten fingers. Computers use the binary system with the base 2, so they must have two fingers. I bet you can guess which two fingers the

computer has!) Just as I was getting a feel for binary, the author started talking about hexadecimal systems with a base 16. Whoa! I could go no further. Every time I tried to go on I was overwhelmed. I could not see the forest for the trees (to coin a phrase) — just too much detail. Hello, fire.

Enter **Simulated Computer II** by Jim Wieder and Scott Steketee. Here is a program that shows how the computer works and, with unique simplicity, introduces many of the concepts of assembly language. By using the decimal system and scaling down in other areas, this simulation provides an enjoyable way for children (ages 12 and up) to learn about the world of the computer. Twenty-four memory locations are used instead of the thousands available in a real computer. In addition, there are only eleven types of instructions. Most importantly, the instructions are executed at the rate of two per second, a veritable snail's pace.



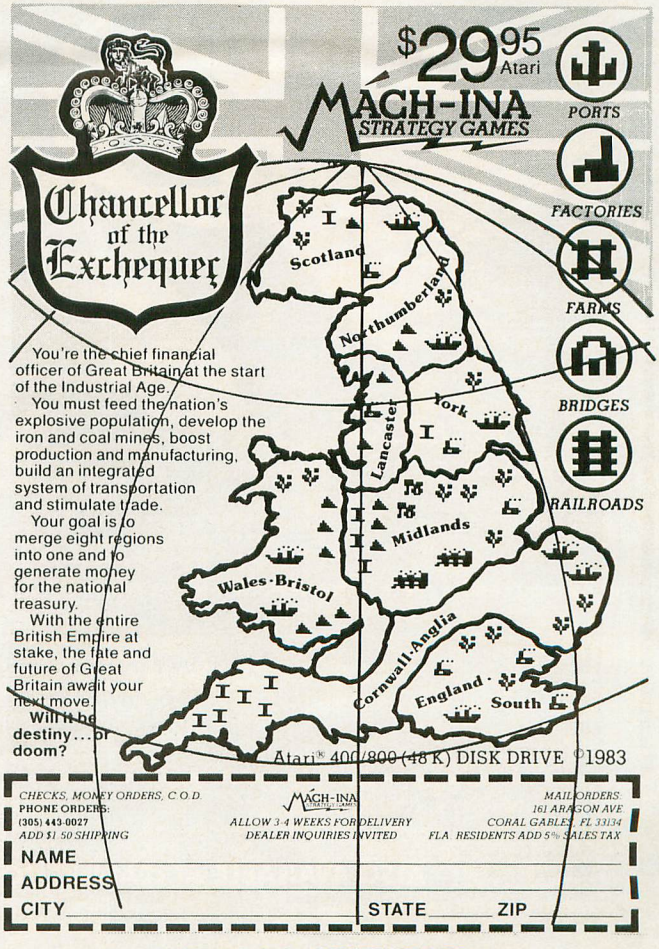
Simulated Computer II.

When the program is loaded, a screen display with all the major parts of the computer is seen. The INPUT DEVICE, a keyboard and monitor, is situated in the upper left corner. As letters or numbers are entered from the actual keyboard, they are seen being typed by animated fingers and are displayed on the simulated monitor. The OUTPUT DEVICE is depicted as a printer. The CPU contains six boxes which show how the computer's "brain" maintains control. Below this is seen twenty-four boxes representing the memory locations of the computer where instructions and numbers are placed.

The eleven available instructions are represented by three-letter mnemonics. One has the capability to LDAxx, or load the accumulator (AC — where all of the arithmetic is done) with the value in memory location xx or, conversely, to STAxx, or store the value of the accumulator into location xx. ADDxx, SUBxx, MULxx, and DIVxx operations can be performed on the accumulator using stored values. One may input (INPxx), output (OUTxx), and jump (JMPxx) to and from various memory locations. SKP lets one skip instructions depending on the value in the AC. STP is the mnemonic for stop.

When a program is written using these instructions and then run, the fireworks begin. Before one's very eyes, electricity begins flowing between the different system components through wires. Boxes light up with numbers and letters appearing and then rapidly changing. One has just witnessed a simulation of what happens in a real computer. Using the RUNSTP command, the program can be viewed one step at a time. The conversion of the mnemonic instructions to numbers is the first thing seen; e.g., LDA13 would be converted to 113. This means that operations code number 1(LDA), the instruction which loads the accumulator, is to be executed with the value found in memory location 13. Then the program counter (PC) is set to the first instruction and FETCHes it from the memory location "pointed to" by the PC. The instruction is then placed in the instruction register (IR), the program counter is incremented (INC), and then the instruction is EXECuted. This process continues until the program is completed. Pretty nifty, huh?

(Continued on page 25)



CIRCLE #115 ON READER SERVICE CARD.

ATARI SINGS YOUR FAVORITE SONGS!!!

THE Original VOICE BOX Speech Synthesizer by the ALIEN GROUP has received rave reviews:

MICRO COMPUTING–"The VOICE BOX injects an endearing personality to your computer. The possibilities are enormous."

COMPUTE–"The VOICE BOX offers more human-like tones and does not blank out the screen."

CREATIVE COMPUTING–"English text and phonetic code may be freely intermixed rather than requiring separate modes as is the case without exception with every other speech system. A mode called talking face displays an animated face with impressive lip sync animation."

ANTIC–"There is a great potential for teaching children to spell and an added dimension to games overall. I believe the VOICE BOX is well worth the price tag."

ANALOG–"For ATARI owners who want to add speech to their programs, the Alien Group VOICE BOX is probably the best choice."

POPULAR SCIENCE–"The speech quality is excellent. Besides creating speech, the software has a bit of fun with graphics."

and on the new VOICE BOX II.....

TIME MAGAZINE–"Machine of the Year" "The VOICE BOX by the Alien Group enables an ATARI to say aloud anything typed on its keyboard in any language. It also sings "Amazing Grace" and "When I'm 64" or anything else that anyone wants to teach it.

INCORPORATE THE SINGING HUMAN FACE INTO YOUR PROGRAMS AND GAMES

VOICE BOX II
Speech & Singing Synthesizer

The New **VOICE BOX II** for **ATARI** plugs into the serial port of the **ATARI** 400/800 with sound coming out of the TV/monitor. 48K DISK is required. It has all of the features of the original **VOICE BOX** plus many exciting new hardware and software features:

- The ability to sing with voice and 3 part music.
- A library of 30 famous songs.
- A comprehensive music system that allows the user to easily enter or modify new songs.
- Software that can convert the bottom two rows of the **ATARI** keyboard into a piano with a range of 3½ octaves using the shift and control keys.
- Programmable musical sound effects such as tremolo, vibrato, glissando and click track.
- A singing *human* face with lip-sync animation designed by Jerry White.
- A talking or singing ***ALIEN*** face with software that allows the user to change the face and 8 mouth patterns as he sees fit.
- The ability to speak with inflection and feeling.
- Can speak in a foreign language with correct foreign spelling as input.
- A talk and spell program by Ron Kramer. Users can program any vocabulary for this spelling game. In fact, this program can even speak in a foreign language like French, where the user must spell the correct word in English, or vice versa.
- **GREEN GOBLINS**–A talking arcade game by John Wilson.
- Random Sentence Generator–An amusing grammar game that helps teach school children to identify parts of speech and recognize a variety of sentence structures.
- **NUMBER SPEAK**–A subroutine by Scott Matthews that converts up to a 9 digit number into normal English pronunciation. Ideal for building your own math games.
- **STUD POKER**–A talking poker game by Jerry White.
- The screen never blanks out while talking or singing.
- Singing or speaking subroutines can be incorporated into your programs, requiring as little as 100 bytes of RAM plus 5 bytes for each word.
- Price $169.00 includes **VOICE BOX II** and all of the above software.
- Inquire about our discounts for educational institutions.

To order by mail send a check or money order to the ALIEN GROUP for $169.

THE ALIEN GROUP
27 West 23rd Street (212) 741-1770
New York, NY 10010

ALSO AVAILABLE AT LEADING COMPUTER STORES THROUGHOUT THE WORLD.

Atari is a registered trademark of Warner Communications.

CIRCLE #116 ON READER SERVICE CARD.

An extensive tutorial leads one through the programming process in a most painless fashion. The last four memory locations have special functions which permit one to use sound, color, and even turtle graphics. The instructions are easily understood and quite explicit in explaining the capabilities of the computer. There is a section with examples, challenging one to write a program which draws a square spiral, evaluates an algebraic expression, or makes decisions. Sample solutions to these and other problems are appended. Error messages are included and, as an integral part of the operations, add further realism. The use of a joystick is also optionally provided.

Simulated Computer II is easy to use, fun to play with and educational to boot. It is not intended to teach assumbly language. It does, however, allow you to grasp the concepts of how it is used and how the computer follows these instructions. In spite of occasional grammatical errors in the accompanying booklet, this is an excellent product and well worth the investment. I still have no intention of competing with my son's idol, Tom Hudson, but at least now I will know what he is talking about. Now, if I only could find out why he buries all those symbols denoting string variables — $$$$!

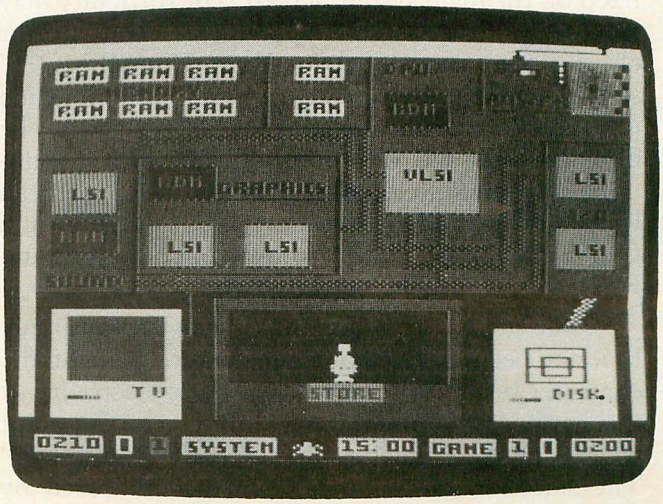
D-BUG
ELECTRONIC ARTS
2755 Campus Drive
San Mateo, CA 94403
48K Disk $35.00

The name Electronic Arts is quickly becoming synonymous with quality. This educational game, designed and programmed by Childware, does nothing to lessen this reputation.

When **D-Bug** is booted, the first thing to appear is the system screen comprised of the various parts of a computer. One has the option of selecting the two player game, or competing against the computer. One of three characters is chosen: Moe Dem, Dot Matrix, or M.A.X. the Robot (my personal favorite). If the computer is the opponent, it assumes the role of a fourth character, Charlie Fixit. Coincidentally, at the continually remodelled home of **ANALOG**, Charlie is the one I always am seeking out to "fix it." At this point, one has the opportunity to move to the different sections on the screen and, with one push of the joystick button, may enter and explore the individual system components. Moving to GAME on the control panel allows one to begin playing.

Gotcha!

Not to be confused with the familiar and nervously anticipated golf handicap, Gotcha! is a simple board-style game with the object being to capture or force an opponent to capture particular shapes. Using the joystick to maneuver, points are awarded every time one's own shape is captured. Suddenly, while you're engrossed in manipulating the enemy into a compromising position, the screen flashes and bizarre things begin to happen. Something has gone awry inside the computer. In order to continue the game, the player must locate the problem and repair it. Here is where the debugging takes place.



D:BUG.

In the beginner levels, the game screen, showing the symptom, alternates with the system screen which is flashing the specific area containing the bug. For example, the game screen may be missing the bottom middle third of its display, and the MEMORY section of the system screen will be seen flashing brightly. One enters the problem area with the previously selected character and moves over each part. If the part is loose, the character will have a gray flicker. This is easily fixed by jumping up and down on the affected part, a feat accomplished by positioning the character on it and pressing the joystick button. If a red flicker appears, the part is "hot" and must be replaced with a new one purchased at Charlie Fixit's Store. Sometimes, one gets no flickering, and a tester must be "rented." In this case, a part has gone bad, and using the tester is like buying a new replacement. There is a time limit for each repair job, and if it is exceeded before the job is completed, one's opponent has a chance to fix it. If the problem cannot be found, or if neither player can fix it, Charlie may be hired to do the job, but at a very high cost (ahem!). If the computer is the opponent, it gets to use Charlie for free. As the levels of difficulty increase, fewer hints are provided. In level 3, the flashing of the section containing the bug is eliminated. Level 4 adds to this difficulty by creating more than one bug at a time. The most difficult level, the fifth, provides no gray or red flickers, requiring the player to find loose and hot parts based solely on the symptom seen on the game screen.

(Continued next page.)

Pffft!

There are seven areas within the computer where problems may arise. "Snow" flickering on and off the screen may be due to a loose monitor cable, or a boot error may result from a faulty disk drive door. These bugs in MONITOR and DISK DRIVE are fixed on the main system screen. The difficulty may be in the POWER SUPPLY, where a bad fuse or clogged filter may be the culprit. The MEMORY section contains eight RAM chips, each of which produces its own symptom if found to be bad. The CPU contains the VLSI (Very Large Scale Integration) chip, and if it is loose, all kinds of bad stuff can happen. The other three sections, SOUND, I/O, and GRAPHICS, all contain a variety of chips (LSI and ROM) which may go on the blink.

Power surges, or "transients," may appear at any time and begin to chase one's character. If caught, the character becomes charged with static electricity. If this happens too often, it may build up to the point where simply touching a chip may cause it to "blow." The static charge build-up may be dissipated by entering the POWER SUPPLY area and touching the GROUND symbol.

Additional features include the ability to change the graphic characters in Gotcha!, initially butterflies and boats, by entering the GRAPHICS seciton. The sounds contained in **D-Bug** may also be changed by entering the area that controls this function. It is possible to play the game without interruptions by selecting the NO BUG control. The game itself is not all that exciting, so this is a seldom used option.

What the fool?

A frequently encountered problem in educational programs is that the disaster which occurs when a wrong answer is given — like the screen blowing up, or being eaten by a monster — may be more exciting than a correct answer. Young children sometimes purposely make mistakes just to see the consequences. In this program, the problem must be righted if the bug appears during one's turn. Because of the cost of parts and time, the debugging process deducts points from one's total. If one were to make quick selections during the Gotcha! game, the odds seem to favor the bug occurring during the opponent's turn. This is particularly true when playing against the computer. Such strategy would enable a player to build up points during the game phase while the opponent sacrifices points repairing the bug. The only problem is that it is much more fun (and educational) to search for and repair bugs than it is to watch someone else do it. Consequently, I found my children (and me!) slow-poling during the game just so the bug would occur while it was still their turn. I am not sure if this is a shortcoming or not. While it may eliminate some of the competitive edge, it may, in turn, enhance intellectual pursuit.

The educational goals of this package are numerous. Learning the names of the different parts of the computer and what functions they control will certainly be achieved. The development of deductive reasoning skills by working from symptom to cause is definitely encouraged. Children ten years and older will be able to use **D-Bug** with little difficulty. Superb graphics and an innovative approach make **D-Bug** an outstanding educational tool for kids of all ages. It is exciting and fun. So much so, they will never suspect that it is educational. Sometimes, we have to fool them into learning. Big Bird does it all the time — **D-Bug** will too. □

MOVING?

DON'T MISS A SINGLE ISSUE.

Let us know your new address right away. Attach an old mailing label in the space provided and print your new address where indicated.

QUESTION ABOUT YOUR SUBSCRIPTION?

Check the appropriate boxes below:

☐ New subscription. Please allow 4-8 weeks for your first copy to be mailed. ☐ Renewal subscription. Please include a current address label to insure prompt and proper extension. ☐ 1-year $28.00. This rate limited to the U.S.A. and its possessions. ☐ Payment enclosed or ☐ Bill me.

MAIL TO: ANALOG COMPUTING
P.O. Box 615, Holmes, PA 19043

Name _____
Address _____
City _____ State _____ Zip _____

ATTACH LABEL HERE

(If label is not handy, print OLD address in this space.)

DRAGONRIDERS OF PERN.™
FLY THE UNFRIENDLY SKIES.

Shooting down the menacing and constantly multiplying Threads isn't easy, but it's only one of the challenges in this official computer game version of Anne McCaffrey's famous book series.

Your strategy will be put to the test as you try to negotiate alliances with Pern's Lord Holders in an attempt to form the most powerful Weyr on the planet. Should you take a firm stance or compromise? Will asking a Craftmaster for assistance increase your chances for success? Maybe you should invite prospective allies to a Wedding or even a Dragon Hatching. Remember to check the Lord Holders personality traits first. It may be critical to your success.

Numerous screens combine to create truly unique and challenging game play. There's even a practice screen to sharpen your Thread Fighting skills.

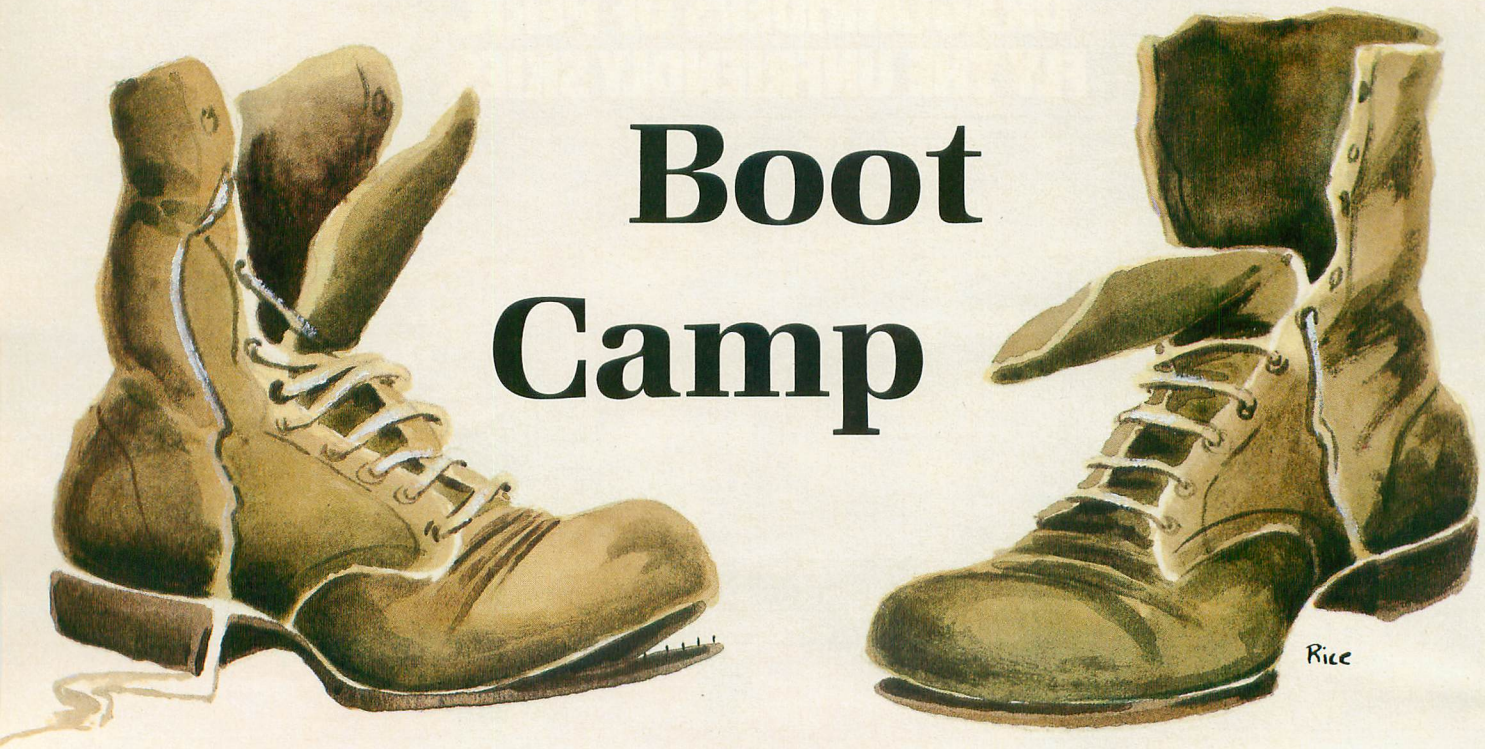
If you liked the books, you'll love the game. After all, how often do you get the chance to actually fly a dragon?

One to four players, joystick and keyboard controlled.

EPYX™
COMPUTER SOFTWARE

Strategy Games for the Action-Game Player

CIRCLE #117 ON READER SERVICE CARD.



Boot Camp

by Tom Hudson

Before beginning my regular **Boot Camp** material, I'd like any users of the **MAC/65** assembler to take a look at this issue's **HBUG** debug package (see page 78).

I received a letter from Allen J. Henninger of Linden, PA in January. He informed me that most of the **Boot Camp** examples failed to operate properly when he used **MAC/65**'s debug utility, **BUG/65**. I looked into the problem and, sure enough, Mr. Henninger was right.

When using **BUG/65**, BRK instructions cause a fatal system crash. Programs executing infinite loops can only be stopped via the SYSTEM RESET key. There are ways to circumvent the BRK lockup problem, but there's no way to stop an infinite loop and find where the program was executing.

If you use **MAC/65**, I strongly suggest that you type in **HBUG**. It'll help you check the operation of the programs shown in **Boot Camp**, avoiding nasty lock-ups.

The solutions.

If you solved last issue's multi-byte math problems, give yourself a pat on the back. Successful completion of these programming puzzles indicates that you're well on your way to becoming proficient in 6502 assembly language.

Whether you solved the problems or not, take a look at the following possible solutions. There are many ways to solve any programming problem, and these examples may show you a different approach.

```
10  *=$600
20  SED                 ;DECIMAL MODE
30  LDA OLDBAL          ;GET LOW BYTE
40  SEC                 ;FIRST SUBTRACT
50  SBC WITHD           ;SUBTRACT LOW
60  STA NEWBAL          ;STORE RESULT
70  LDA OLDBAL+1        ;GET MED BYTE
80  SBC WITHD+1         ;SUBTRACT MED
90  STA NEWBAL+1        ;STORE RESULT
0100  LDA OLDBAL+2      ;GET HI BYTE
0110  SBC #0            ;SUBTRACT DUMMY
0120  STA NEWBAL+2      ;STORE RESULT
0130  BRK               ;ALL DONE!
0140 OLDBAL .BYTE $73,$86,$10
0150 WITHD  .BYTE $85,$42
0160 NEWBAL *=*+3
0170  .END
```

Figure 1.

Figure 1 shows the solution to the first problem given last month. You were asked to subtract the two-byte BCD variable WITHD from the three-byte variable OLDBAL, placing the result in the three-byte variable NEWBAL; OLDBAL = 108673 and WITHD = 4285.

As you can see from **Figure 1**, both OLDBAL and WITHD are defined using the .BYTE directive. Standard data storage formats are used, so the values are defined from low-order to high-order. That is, 108673 is defined as .BYTE $73,$86,$10. The variable NEWBAL is simply set up as *=*+3, reserving three bytes for the result of the operation.

The program itself uses the usual multi-byte subtract structure for the first two subtract operations. The third subtract uses a "dummy" value of

zero for the third byte of WITHD, since it is one byte shorter than OLDBAL. This insures that any borrows from lower-order bytes will be processed properly.

Try executing this program on your computer. After it is finished, examine the three-byte NEWBAL to be sure it contains 104388 (108673 - 4285). NEWBAL is located at memory location $0622-0624. If you display these locations, you will see something like **Figure 2**.

```
0622 88 43 10
```

Figure 2.

You will note that the number 104388 contained in NEWBAL is stored in low-order to high-order format, just like OLDBAL and WITHD.

Solution two.

The second problem I assigned last month asked you to subtract each byte of the ten-byte TABLE2 from the corresponding byte of TABLE1, placing the results in the ten-byte TABLE3. The initial values for TABLE1 and TABLE2 are:

```
TABLE1 .BYTE $10,$18,$40,$86,$9A
       .BYTE $A0,$BC,$C0,$F0,$F8
TABLE2 .BYTE $00,$08,$14,$2F,$9A
       .BYTE $90,$0B,$22,$65,$78
```

If done properly, TABLE3 should contain the following values when the program is finished:

```
$10,$10,$2C,$57,$00,$10,$B1,$9E,$8B,$80
```

A possible solution to this problem is shown in **Figure 3**.

```
10        *=$600
20        CLD             ;BINARY MODE!
30        LDX #9          ;10 BYTES TO DO
40 SUBLP  LDA TABLE1,X    ;GET BYTE 1
50        SEC             ;SINGLE-BYTE!
60        SBC TABLE2,X    ;SUBTRACT BYTE2
70        STA TABLE3,X    ;AND STORE IT
80        DEX             ;NEXT BYTE
90        BPL SUBLP       ;DO ALL 10 BYTES
0100      BRK             ;ALL DONE!
0110 TABLE1 .BYTE $10,$18,$40,$86,$9A
0120        .BYTE $A0,$BC,$C0,$F0,$F8
0130 TABLE2 .BYTE $00,$08,$14,$2F,$9A
0140        .BYTE $90,$0B,$22,$65,$78
0150 TABLE3 *=*+10
0160        .END
```

Figure 3.

As you can see from **Figure 3**, this problem can be solved by simply indexing through all ten bytes of the tables in the loop SUBLP. Within this loop, the X register points to the desired byte of each table. Each time the loop is executed, the byte from TABLE2 is subtracted from the corresponding byte of TABLE1, and the result is placed in the proper location in TABLE3. Note that each subtract is preceded by the SEC (set carry) instruction, so that the subtracts will be treated as single-byte operations.

If you're still having trouble with multi-byte math, go back and re-read last issue's column. It may also be a good idea to review the math basics from ANALOG #15's **Boot Camp**.

Ups and downs.

There are two handy instructions we haven't covered yet that can sometimes be considered math instructions. These are INC (increment memory by 1) and DEC (decrement memory by 1).

INC n	(ZERO PAGE)
INC nn	(ABSOLUTE)
INC n,X	(ZERO PAGE INDEXED X)
INC nn,X	(INDEXED X)
DEC n	(ZERO PAGE)
DEC nn	(ABSOLUTE)
DEC n,X	(ZERO PAGE INDEXED X)
DEC nn,X	(INDEXED X)

The INC instruction simply adds 1 to the value contained in the memory byte referenced and places the result back into the memory location. The accumulator is not affected, but the SIGN and ZERO

(Continued on page 31.)

SUPERIOR QUALITY PERIPHERALS from Convologic

The **BYTEWRITER** is a multi-use EPROM burner that interfaces to your computer through joystick ports 1 and 2 or 3 and 4 (software selectable). The **BYTEWRITER** reads, writes and verifies EPROMS: 2516, 2716, 2532, 2732, 2564, 2764, 68764, 27128. No personality modules. Will read operating system ROMS.

Operational software will support inspect and change, constant store, cassette and disk I/O. Source Code available for **$35.00.**

BYTEWRITER – $189.00

The **BYTEREADER** option interfaces cartridges to **BYTEWRITER.**

BYTEREADER – $30.00

The **NEW 1850** expansion unit provides a versatile interface between your 400/800 and peripherals. Residing on the SIO bus with your disk and/or cassette drive, it can provide for communication to Centronics compatible printers and RS-232 asynchronous devices, such as a modem. An internal 2K buffer region is allocated to meet the requirements of printer and serial activity.

NEW 1850 – $149.00

Cartridge boards dual socketed for 2764's, to make 8K or 16K – Cartridges – **$7.50,** Shells for cart. **$5.00,** 2764 EPROM – **$10.00.**

Ask for free brochure.

Each accessory is fully assembled with a 90 Day Limited Warranty. Mail Cashier's Check or Money Order for 4 to 6 weeks delivery. Shipping and handling $4.00 domestic, $6.00 foreign, C.O.D. orders $3.00 extra $30.00 minimum order to:

MasterCard VISA

CONVOLOGIC, INC.

ORDERS ONLY 1-800-227-3800 Ext. 7024

Send for FREE Brochure – Dealer Inquiries Encouraged.

421 Bay Tree Lane, Longwood, FL 32779, (305) 869-6630

CIRCLE #118 ON READER SERVICE CARD.

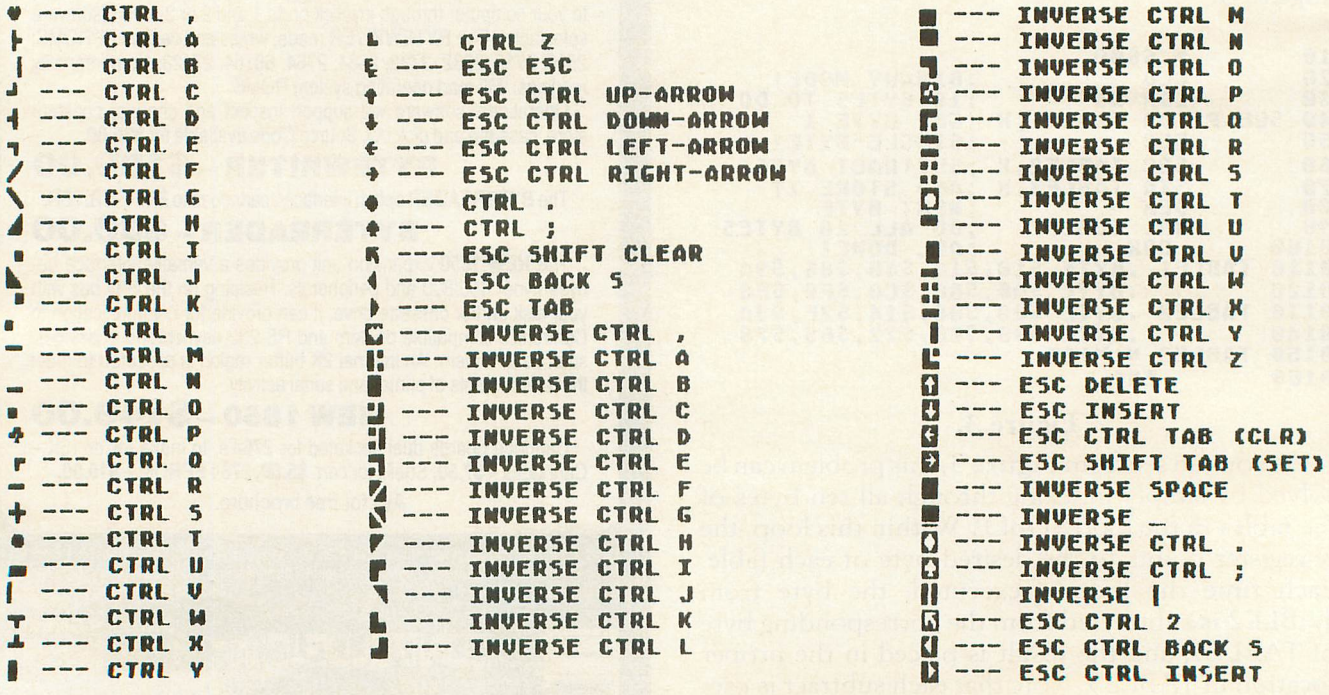
WHAT IS D:CHECK/C:CHECK?

Most program listings in **ANALOG** are followed by a table of numbers appearing as DATA statements, called "CHECKSUM DATA." These numbers are to be used in conjunction with D:CHECK and C:CHECK, which appeared in the **ANALOG Compendium** and Issue No. 16.

D:CHECK and C:CHECK are programs by Istvan Mohos and Tom Hudson. They are designed to find and correct typing errors when entering programs from the magazine. For those readers who do not have a copy of either article, send a pre-addressed, stamped, business-sized envelope to:

D:CHECK ARTICLE
P.O. BOX 23
WORCESTER, MA 01603

Some program listings reproduced in A.N.A.L.O.G. may contain "strange" characters not shown on the ATARI keyboard. These are special characters which use the CTRL, ESC and "ATARI LOGO" (INVERSE) keys. Shown below is a list of these characters and the keystrokes used to get them. □



flags reflect the result of the operation. **Figure 4** shows an example of the INC operation.

```
10        *=$0600
20        LDA #5          ;5 IN ACCUMULATOR
30        STA VALUE       ;AND IN VALUE
40        INC VALUE       ;VALUE = 6
50        INC VALUE       ;VALUE = 7
60        INC VALUE       ;VALUE = 8
70        BRK             ;ALL DONE!
80 VALUE  *=*+1
90        .END
```

Figure 4.

This program will place the value 5 in the accumulator and the location labeled VALUE. It then increments VALUE 3 times. When finished, the accumulator will still contain 5, but VALUE will contain 8.

If the INC operation is performed on a byte containing $FF, the byte's value will "wrap around" to zero. Note that this instruction is not a true math instruction because the carry resulting from the byte wraparound is NOT shown in the status flags.

The DEC instruction is similar to the INC instruction, but operates in reverse. Instead of adding 1 to the value of the byte, DEC subtracts 1. **Figure 5** shows an example of the use of the DEC instruction.

```
10        *=$600
20        CLD             ;BINARY MODE
30        LDA #5          ;SET COUNTER...
40        STA COUNT       ;TO 5
50        LDA #7          ;SET ADDVAL...
60        STA ADDVAL      ;TO 7
70 LOOP   LDA ADDVAL      ;GET ADDVAL
80        CLC             ;SINGLE-BYTE ADD
90        ADC ADDVAL      ;ADD TO ITSELF
0100      STA ADDVAL      ;SAVE RESULT
0110      DEC COUNT       ;HIT ZERO YET?
0120      BNE LOOP        ;NO! LOOP BACK
0130      BRK             ;ALL DONE!
0140 ADDVAL *=*+1
0150 COUNT  *=*+1
0160      .END
```

Figure 5.

In **Figure 5**, we're using the variable COUNT as a simple counter to control the addition of ADDVAL. We will add ADDVAL to itself 5 times. When finished, ADDVAL will be multiplied by 32. Let's walk through this example.

Line 20 clears the decimal mode so that we'll be working in binary mode.

Lines 30-40 initialize COUNT to 5.

Lines 50-60 initialize ADDVAL to 7. When complete, this program will multiply 7 by 32, with a result of 224 ($E0) in the accumulator.

Lines 70-100 add ADDVAL to itself, placing the result back in ADDVAL. This has the effect of multiplying ADDVAL by 2 each time it is done.

Line 110 decrements COUNT by 1. When COUNT reaches zero, the ZERO flag will be set. This will be our signal to stop.

Line 120 checks the ZERO flag to see if all five multiplies have been done. If the ZERO flag is NOT set, the program will branch (BNE) back to the label LOOP.

Line 130 BREAKS the program when all five iterations of the loop are complete.

Lines 140-150 define the one-byte storage areas ADDVAL and COUNT.

As you can see, the INC and DEC instructions can come in handy when you need a counter or want to add or subtract without affecting the accumulator. We have used the X and Y registers to perform counter functions, but if these registers are in use, you can always set up a byte and use the INC and DEC instructions instead.

Bit-flipping.

When you get deeper into assembly language, you'll need to manipulate bytes in ways that BASIC can't. Now we'll look at four instructions that allow a wide variety of ways to manipulate and test the contents of the accumulator. These instructions are AND, BIT, ORA and EOR.

```
      BYTE 1:  0 1 1 0 1 0 1 1
AND   BYTE 2:  1 0 1 1 0 0 0 1
      ---------------------------
      RESULT:  0 0 1 0 0 0 0 1
```

Figure 6.

Figure 6 shows how the AND function works. As you can see, two bytes are used as inputs to the function. The corresponding bits of these two bytes are examined. If the bit of the first byte is 1 AND the bit of the second byte is 1, the result for that bit will be 1. Otherwise, that bit of the result will be set to 0. This process is repeated for all eight bits.

In 6502 assembly language, the AND function has the following eight formats:

AND #n	(IMMEDIATE)
AND nn	(ABSOLUTE)
AND n	(ZERO PAGE)
AND (n,X)	(PRE-INDEXED INDIRECT)
AND (n),Y	(POST-INDEXED INDIRECT)
AND n,X	(ZERO PAGE INDEXED X)
AND nn,X	(INDEXED X)
AND nn,Y	(INDEXED Y)

In each of these formats, the accumulator is ANDed with the memory byte indicated in the operand. The result of the AND function is placed in the accumulator. The SIGN and ZERO flags are set according to the result.

The AND function is most often used to mask off certain bits of the accumulator or test bits to see if they are on.

Let's say you want to get a random number that does not exceed 7. You could use the code:

Your "ONE STOP" Source for Computer Books

You can rely on PACE for ONE STOP shopping for all your Micro Computer needs. We have thousands of books, programs and accessories—covering all the major brands of computers—even 60 different magazines! The ONE STOP friendly store.

pace

VISA/MASTERCARD ORDERS CALL (312) 595-3860

BOOKS ON THE ATARI®

☐ **EASY GUIDE TO 400/800**
4795-000125 160 Pgs $ 7.95

☐ **BASIC EXERCISES**
4795-000101 251 Pgs $12.95

☐ **YOUR FIRST PROGRAM**
4795-000130 150 Pgs $ 9.95

☐ **COMPUTE!'s 1st BOOK of ATARI**
4105-000000 184 Pgs $12.95

☐ **INSIDE ATARI DOS**
4105-000002 108 Pgs $19.95

☐ **COMPUTE!'s 2nd Book of ATARI**
4105-000006 250 Pgs $12.95

☐ **COMPUTE!'s 1st BOOK of ATARI GRAPHICS**
4105-000008 248 Pgs $12.95

☐ **MAPPING THE ATARI** Explanations of Atari memory locations (400/800)
4105-000009 194 Pgs $14.95

☐ **ATARI BASIC SOURCEBOOK**
4105-000015 296 Pgs $12.95

☐ **COMPUTE!'s 1st BOOK of ATARI GAMES**
4105-000014 232 Pgs $12.95

☐ **COMPUTERS FOR KIDS - ATARI**
4198-000022 72 Pgs $ 4.95

☐ **THE CREATIVE ATARI** How-To for the beginner or expert.
4198-000034 243 Pgs $15.95

☐ **31 NEW ATARI PROGRAMS FOR HOME, SCHOOL AND OFFICE**
4080-000018 95 Pgs $ 8.95

☐ **101 PROGRAMMING TIPS AND TRICKS FOR THE ATARI**
4080-000022 126 Pgs $ 8.95

☐ **STIMULATING SIMULATIONS for the ATARI**
4410-005197 118 Pgs $ 6.50

☐ **ATARI IN WONDERLAND** 22-programs for learning and fun.
4410-005771 139 Pgs $ 9.95

☐ **YOUR ATARI COMPUTER** Complete how-to guide book.
4665-000065 458 Pgs $17.95

☐ **ATARI GAMES & RECREATIONS** BASIC techniques & educational games.
4690-000242 338 Pgs $14.95

☐ **ATARI PILOT FOR BEGINNERS**
4690-000301 229 Pgs $14.95

☐ **INSIDE ATARI BASIC**
4690-003082 183 Pgs $12.95

☐ **THE VISICALC BOOK FOR ATARI**
4690-008393 298 Pgs $14.95

☐ **MOSTLY BASIC: ATARI Vol #1**
4760-022075 181 Pgs $12.95

☐ **MOSTLY BASIC: ATARI Vol #2**
4760-022092 $12.95

☐ **ATARI SOUND & GRAPHICS**
4925-009593 234 Pgs $ 9.95

BOOKS ON THE ATARI®

☐ **PROGRAM WRITING WORKBOOK**
4080-000814 $ 4.95

☐ **32 BASIC PROGRAMS for ATARI** Business, educational and games.
4250-000084 $19.95

☐ **HANDS ON BASIC - ATARI**
4525-049194 $19.95

☐ **KIDS & THE ATARI**
4560-000055 219 Pgs $19.95

☐ **400/800 DISK GUIDE**
4665-000095 $ 7.95

☐ **INTRO TO ATARI GRAPHICS**
4665-000111 $16.95

☐ **THE ATARI ASSEMBLER**
4690-000236 $14.95

☐ **ATARI: A BEGINNER'S GUIDE**
4690-000271 $12.95

☐ **ATARI: A USER'S GUIDE**
4690-000323 $15.95

☐ **ATARI BASIC**
4690-049791 $13.95

☐ **HOW TO WIN ATARI COMPUTER GAMES**
4768-049558 $ 8.95

☐ **EASY GUIDE TO 400/800**
4795-000125 $ 7.95

☐ **YOUR FIRST ATARI PROGRAM**
4795-000130 $ 9.95

☐ **ATARI BASIC**
4925-006496 $10.95

☐ **ATARI BASIC REFERENCE GUIDE** Large card with all BASIC info needed
4925-087041 $ 2.95

☐ **BASIC EXERCISES FOR THE ATARI**
4795-000101 251 Pgs $12.95

☐ **HOW TO USE ATARI COMPUTERS:** An Introduction to 400/800/1200
4040-000235 65 Pgs $ 2.95

☐ **UNDERSTANDING GRAPHICS**
4040-000224 49 Pgs $ 2.95

☐ **BASIC CODING SHEETS** - Graph sheets, pad of approx 40 sheets
4080-000806 $ 2.95

COMPUTER DUST COVERS
For ATARI®

☐ 4108-000800 400 $11.98
☐ 4108-000810 410 $ 4.98
☐ 4108-000820 800 $12.98
☐ 4108-000830 810 $ 7.98
☐ 4108-000845 820 $ 8.98
☐ 4108-000850 822 $ 7.98
☐ 4108-000860 825 $11.98
☐ 4108-000865 825 w/rk $11.98
☐ 4108-000877 1200 $11.98

Also for: APPLE®, COMMODORE®, IBM®, TEXAS INSTRUMENTS®, RADIO SHACK® and *MORE!*

NOW. . . PACE has two stores! Our main store, at 345 East Irving Park Road in Wood Dale, just West of O'Hare Airport, (phone 312-595-3860), and our new location at 1 FIRST NATIONAL PLAZA in Downtown Chicago, (phone 312-372-2464).

USE THIS AD AS YOUR HANDY ORDER FORM

TO ORDER: Just check the block by the title of the book or item you want. Send, along with your NAME, STREET ADDRESS, CITY, STATE and ZIP CODE, and your Certified Check, Money Order, Personal Check (allow 2 weeks to clear), or your VISA or MASTERCARD NUMBER, EXPIRATION DATE and INTERBANK NUMBER (Min. Charge is $25) to:

pace micro software centers

DEPARTMENT: A - C
LOCK BOX 328, BENSENVILLE, IL 60106
Phone: (312) 595-3860

Please add $2.50 postage and handling on all orders. Illinois residents add 6% tax. Foreign orders add 10% (Minimum $5.00).

PLEASE SEND ME A COMPLETE BOOK & SOFTWARE LISTING FOR:
☐ APPLE® ☐ ATARI® ☐ COMMODORE® ☐ IBM ☐ RADIO SHACK®
☐ TEXAS INSTRUMENTS® ☐ TIMEX®/SINCLAIR® ☐ Other _____

CIRCLE #119 ON READER SERVICE CARD.


```
GETRND LDA RANDOM
       CMP #8
       BCS GETRND
```

This code gets a random number and checks to see if it is greater than 7. If it is, the program loops back to GETRND and tries again. This routine works, but it may need to try several times before it gets a good value.

We can perform the same function easily with the AND instruction. By using the AND instruction, only one try is necessary. It even takes less memory than the previous example. The code is:

```
LDA RANDOM
AND #07
```

This code MASKS the contents of the accumulator with the value 7. **Figure 7** shows three possible outcomes of the procedure. As you can see, none of them exceed 7.

```
      BYTE:  1 0 0 1 1 1 0 1
AND   MASK:  0 0 0 0 0 1 1 1
     ------------------------
    RESULT:  0 0 0 0 0 1 0 1  =  5

      BYTE:  1 1 1 1 0 1 1 1
AND   MASK:  0 0 0 0 0 1 1 1
     ------------------------
    RESULT:  0 0 0 0 0 1 1 1  =  7

      BYTE:  0 0 0 1 0 0 0 0
AND   MASK:  0 0 0 0 0 1 1 1
     ------------------------
    RESULT:  0 0 0 0 0 0 0 0  =  0
```

Figure 7.

This is just one example of the use of the AND operation. We'll cover more uses in the future.

A companion to the AND function is the BIT (bit test) instruction. It performs almost the same function as AND, but changes only the status flags. BIT does *not* affect the contents of the accumulator. The primary function of the BIT operation is to test the contents of the accumulator. BIT has the following formats:

BIT nn (ABSOLUTE)
BIT n (ZERO PAGE)

Besides not changing the accumulator as a result of the AND operation, BIT handles the status flags differently. The ZERO flag is handled the same as AND. The SIGN and OVERFLOW flags are set to bits 7 and 6 of the operand, respectively. This is a strange twist, and I've not yet encountered a situation where I've used this odd flag setting. The following code shows a typical use of the BIT instruction.

(Continued on page 34.)

As Easy As Falling Off . . .

HIGH QUALITY PRODUCTS AT AFFORDABLE PRICES

ALOG COMPUTING

ADVANCED SOFTWARE DESIGNS FOR HOME COMPUTERS

Dealer and distributor inquiries invited.

We at **ALOG Computing** wish to thank you for the response you have given us on our first product the **ALOG Pagewriter**. (See Comments) Now we have two other utility programs available for Atari Home Computer users. We have tried to make them simple and fun to use. We hope you enjoy them.

All our programs work on ATARI 400/800, 800XL, and 1200XL computers.

THE ALOG PAGEWRITER

Turns your ATARI computer and 80 column printer into a very easy to use electronic typewriter. Because the entire page layout is displayed while the user is typing and editing, the **ALOG PAGEWRITER** is ideal for simple word processing tasks such as letters, notes, memos, or the kid's book reports and term papers. The average learning time is about five minutes.

KEY FEATURES

• Uses standard Atari editing keys (e.g. INSERT, DELETE, TAB, etc.) • A HELP screen with command summary • Visible and fully adjustable margins • FILL command for right justification • Automatic return option (with word moving) • Line 'Split' and 'Splice' commands • Store 10 pages with no disk swapping • Not copy protected

REQUIRED EQUIPMENT:

• At least 32K of RAM • One disk drive • 80 column printer

THE ALOG DISPLAYMAKER

A versatile color graphics program in Forth that you can use for making graphs, charts, displays or just have fun drawing pictures.

KEY FEATURES

Graphics Mode: • Circles, Ellipses, Rectangles • "Rubber Band" lines • Semi-automatic fill • Color pallette for easy variation • Zoom magnification

Text Mode: • Three sizes of characters • Three colors • Regular ATARI character set • Special character set • "Built-in" character editor (gives unlimited number of characters)

Disk Storage of 8 displays

"Slide Show" display features

Graphics dump to many dot graphics printers
(2 sizes) — Epson*, Gemini 10, Prowriter and others.
*requires Graphtrax option

REQUIRED EQUIPMENT

• At least 48K of RAM • One disk drive • Optional (joystick and graphics printer)

THE ALOG MAILLIST

A simple, easy to use data base program specifically designed for making and keeping mailing lists and printing out mailing labels.

KEY FEATURES

• Multilevel fast search and sort • Complete disk utility package • Split & merge files • Supports 1 or 2 disk drives • Prints continuous form labels 1 to 2 inches high • Uses standard ATARI Editing Keys • Holds 130 records in memory and 500 records per disk • Make your own backup copies

REQUIRED EQUIPMENT

• At least 32K RAM (48K recommended). • One disk drive (works fine with two) of any manufacture that works with ATARI DOS 2.0S • ATARI BASIC Cartridge • ATARI DOS 2.0S • A printer with adjustable tractor feed.

COMMENTS

from users of the Alog Pagewriter (used with written permission)

"Great product. I wrote two pages the first time I sat down with it. I wish I had it a couple of years ago."
James Tanaka, Monterey Park, CA

"Refreshingly simple but adequate word processor at an affordable price — good work."
Jay Carccarese, San Jose, CA

"Super, I am a Valforth programmer and I love it."
P.R.S., San Antonio, Texas

"Good, simple text processor for letters — great to be able to back up master disk."
Larry Cox, Floyds Knobs, IN

"Excellent, it makes letter writing a dream."
M.H., Oaktown, VA

"Great price and easily learned."
Noel Brooks, Great Falls, MT

*"As I have used **PAGEWRITER** I have grown to love it. After using WORDSTAR and LETTER PERFECT a truly user kind word processor is a wonderful experience."*

"One feature which I feel sets it apart from all others is the capability to visualize the page. This feature is indispensable when designing tables for a report."

"Thanks again for an excellent piece of software."
John C. Goodman, Marblehead, PA

"Excellent for one or two pages of text. It's simple."
Richard E. Lane, Vandenberg AFB, CA

"Very Easy to use. It's very useful for letters"
Florian C. Pulver, Riverside, California

"I have found your product to be quite easy to operate and understand, both in the written instructions and manual operation. I have the ATARIWRITER and BANK STREET WRITER and I find myself using your product more than either one of these two products."

"I have also found that for the price of your product versus that of the other companies that your product has paid for itself twice over."
Edward Locke, Mentor, OH

"Very easy to use and perfect for my needs."
J.B. Karluk, Throop, PA

All programs are only $39.95 each. To order direct send check or money order to **ALOG Computing, 1040 Veronica Springs Road,** Santa Barbara, CA 93105. We pay shipping. For information, credit card orders or C.O.D., call our distributor: **COMSTAR (805) 964-4660.**

CIRCLE #120 ON READER SERVICE CARD.

(ATARI is a trademark of Atari, Inc.)

```
        LDA BYTE
        BIT TESTBT
        BNE BITON
          .
          .
          .
BITON     .
          .
          .
BYTE    *=*+1
TESTBT  .BYTE $01
        .END
```

This code uses the bit mask TESTBT to see if the 1 bit of the memory location labeled BYTE is set. The value contained in BYTE is placed in the accumulator, then the BIT instruction is executed. Since TESTBT is the location used by the BIT operand, the accumulator will be ANDed with $01. If the 1 bit of the accumulator is set, the result of the BIT operation will be a NOT EQUAL condition. In this case, the BNE instruction would cause the program to branch to the location BITON. Otherwise, the program would fall through to the code after the BNE instruction.

I personally don't use BIT instructions much. Unfortunately, the designers of the 6502 didn't allow for an immediate format of this instruction. As a result, you must set up all the masks you use somewhere in memory, making the operation a bit more cumbersome.

This OR that.

Another bit-manipulating instruction used fairly often is the ORA (OR accumulator) operation. The formats of this instruction are:

ORA #n	(IMMEDIATE)
ORA nn	(ABSOLUTE)
ORA n	(ZERO PAGE)
ORA (n,X)	(PRE-INDEXED INDIRECT)
ORA (n),Y	(POST-INDEXED INDIRECT)
ORA n,X	(ZERO PAGE INDEXED X)
ORA nn,X	(INDEXED X)
ORA nn,Y	(INDEXED Y)

Unlike the AND operator, which only sets the result bit when both input bits are 1, the OR operator sets the result bit when EITHER input bit is 1. **Figure 8** shows how the OR function works.

```
    BYTE 1:  1 0 1 1 0 1 1 0
OR  BYTE 2:  0 1 0 1 0 0 1 0
    RESULT:  1 1 1 1 0 1 1 0
```

Figure 8.

As you can see, the OR operation sets the result bit if either bit 1 OR bit 2 is set. If both of the bits are off, the result bit will also be off. Like the AND operation, the ORA operation affects only the SIGN and ZERO flags.

The OR operation is used to turn on specific bits in a byte, most often in graphics handlers. The following code demonstrates how the OR instruction works.

```
10      *=$600
20      LDA #$4C        ;$4C IN ACCUM.
30      ORA #$11        ;OR WITH $11
40      ORA OR3         ;OR WITH $80
50      BRK             ;ALL DONE!
60 OR3  .BYTE $80
70      .END
```

Figure 9.

Line 20 loads the accumulator with $4C (01001100 binary).

Line 30 ORs the accumulator with $11 (00010001 binary). After this OR operation, the accumulator will contin $5D (01011101) binary).

Line 40 ORs the accumulator with the contents of the memory location OR3. Since OR3 is defined as $80, the accumulator will be OR'd with 10000000 binary. After this instruction is executed, the accumulator will contain $DD (11011101 binary).

Line 50 stops the execution of the program. At this point you can see that the accumulator contains $DD.

An ANALOG exclusive.

The last accumulator manipulation instruction we're going to look at this time is EOR (exclusive-OR). This instruction works like OR except that when BOTH input bits are set, the result bit will be turned off. The following example shows how EOR works:

```
    BYTE 1:  1 0 1 1 0 0 1 1
EOR BYTE 2:  1 0 0 1 1 0 1 0
    RESULT:  0 0 1 0 1 0 0 1
```

The EOR instruction is commonly used in graphics routines, and also for flipping the setting of bits in program flags. Let's see how the EOR instruction lets us flip bits. The following example shows the EOR function flipping all the bits of a byte to the opposite binary settings:

```
    BYTE 1:  1 0 1 1 0 0 0 1
EOR BYTE 2:  1 1 1 1 1 1 1 1
    RESULT:  0 1 0 0 1 1 1 0
```

No matter what the contents of byte 1, if it is exclusive-OR'd with $FF (binary 11111111), the result of the operation will be the mirror-image of the first byte. The 6502 code necessary for this operation is:

```
LDA #$B1
EOR #$FF
```

What if we only want to flip a certain bit? The following example shows the flipping of only the 4 bit of byte 1:

```
      BYTE 1:  1 0 1 1 0 0 0 1
EOR   BYTE 2:  0 0 0 0 0 1 0 0
      RESULT:  1 0 1 1 0 1 0 1
```

As you can see, the bit has been flipped to a 1. The equivalent 6502 code for this example is:

```
LDA #$B1
EOR #$04
```

The EOR operation is easy to use. All you need to do is determine which bits you want to flip and exclusive-OR the accumulator with the appropriate byte. Like the AND and ORA operation codes, EOR sets the SIGN and ZERO flags according to the result of the operation.

Problem time.

Here are some good bit-manipulation problems for you to solve for next month.

In each of the following problems, you are given bit patterns before and after a bit manipulation operation. You must determine (1) the operation (AND, ORA, EOR) and (2) the second bit pattern used to obtain the result. Some problems have 2 possible answers. These are indicated with a (2) to the right of the problem. If you've read carefully, these should be a snap to solve.

BYTE 1	OPN	BYTE 2	RESULT	ANS
01000011			01000001	(2)
11001011			10100010	
11110000			01000000	(2)
01010101			11111111	(2)
11001000			01111100	
11111111			11110001	(2)
00100100			10111000	
01000111			00010010	

Until next time, try developing some problems of your own. It's a good idea to try some addressing modes other than the ones used in this column. Next month, we'll find out how to do simple multiplication and division! □

Address all letters to:

BOOT CAMP
P.O. Box 23
Worcester, MA 01603

THIS ONE'S A KEEPER.
Unlike games that get "stale", you'll USE this program – again & again for years to come.

There's nothing else like
DECISIONS ... DECISIONS™
A TOOL FOR MAKING LOGICAL CHOICES.

Combines scientific decision-making principles with your computer's power, to shine new light on business, personal, and family decisions.

» EASILY SORT OUT CONFUSING INFORMATION.
» REMOVE UNCERTAINTY FROM DIFFICULT CHOICES.
» CLARIFY AND QUANTIFY YOUR IDEAS.
» SHOW OFF YOUR COMPUTER'S PRACTICAL SIDE!

Leads you step by step through any problem; analyzes your choices with swift computer accuracy; displays the results in easy to interpret graphic form, on your TV screen.

DECISIONS...DECISIONS program disk for 48K ATARI and complete reference manual, in handsome binder, only $37.50 including shipping, add $2.25 tax in Calif. ORDER NOW - you'll be glad you did. Send check to:

LATERAL SOFTWARE
Dept. 2
P.O. Box 605
Stanton, CA 90680

CIRCLE #121 ON READER SERVICE CARD.

HERE COMES THE FUN!

PLAY

BEAT THE BEATLES

on Interactive Software

48K DISK OR CASSETTE $24.95

AVAILABLE NOW. ORDER DIRECT FROM:
INTERACTIVE SOFTWARE
P.O. BOX 991
BALA-CYNWYD, PA 19004

FOR ALL ATARI BRAND COMPUTERS

©1983 INTERACTIVE SOFTWARE. (215) 725-9756

NOT AFFILIATED WITH ATARI, INC.
CIRCLE #122 ON READER SERVICE CARD.

ULTIMA I
by Lord British
SIERRA ON-LINE
Sierra On-Line Building
Coarsegold, California 93614
48K Disk $39.95

by Steve Panak

Ultima I may not be the most difficult adventure game, but it is quite possibly one of the most addictive. One of the most famous adventure games for the Apple computer has finally been translated for the Atari, and, much to my horror, I was unable to get a full night's sleep until I had completed it.

The game follows the traditional D & D theme. You must first create your character, assigning him various attributes, such as strength and intelligence, and then you steer him through the **Ultima** universe. Each character is created from a "prototype seed" which is first copied onto a blank disk. Then a menu is presented for you to assign attribute values and select a race and class (fighter, wizard, etc.) for your character. Throughout the adventure, you will learn to love your creation, as he gains experience, gold, and rises in level until he is able to battle the evil Mondain.

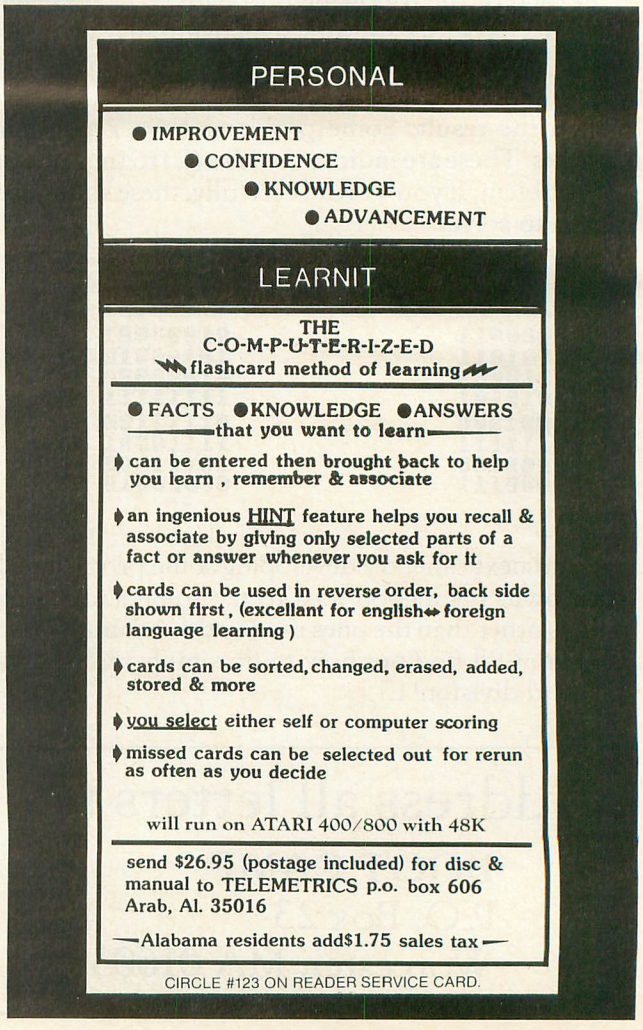
Ultima I is epic in its scope, as it encompasses four continents, as well as space and time. As you increase your wealth and power you are able to purchase boats, aircars, and finally a spacecraft. Weapons, too, increase in power, as does your armor. Wizards and clerics can utilize increasingly more powerful magic spells. You must steer your character through towns, in which you purchase supplies; castles, where you ask the king for quests and rescue a princess; and dungeons, where you complete your quests and gain experience and gold by defeating a multitude of monsters. I won't give away the nature of the quests to avoid spoiling the adventure, but I will say that the battles in the dungeons are probably the best part.

The hi-res graphics are most impressive. There is fine scolling as you move over the continents. The dungeons are likewise impressive, and are drawn with linear lines, simulating vector graphics like those found in the arcade version of **Tempest**. Each of the many monsters has a distinctive appearance and specific strengths and weaknesses. Movement is extremely easy with the joystick option. Even when using the joystick, the keyboard must be used, as the joystick only controls movement on one plane and allows you to attack. Often times, very late at night, you will find yourself repeatedly thinking that you will go into the dungeon "just one more time."

Unfortunately, the game is not without its drawbacks. Much time is spent waiting for the disk to supply more data when you move from one playing area to another. Also, many disk swaps are necessary; the game would be much more playable if two disk drives are used. Puzzle quality and difficulty are low. Little, if any, actual problem-solving is necessary, as most problems simply involve going somewhere or killing something. Although the graphics are great, the space sequence is poor arcade action at best, although a slight strategy is necessary. There is a save game feature which is always a plus. The end of the game is a bitter disappointment, somewhat of an anti-climax, when compared to what I expected.

The documentation is fairly limited, which leaves a lot for the player to figure out on his own. This is a convenient reference card, to remind you of the one-letter commands to control the action.

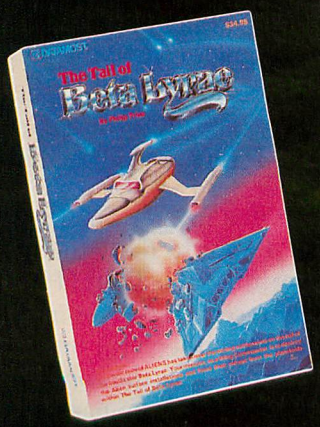
All things considered, I can give a conditional recommendation for **Ultima I**. It will please those people who revel in arcade action. It may disappoint those who prefer high difficulty and logical puzzles which must be solved. However, I enjoyed the game, although I belong to the latter group. Rating on a scale from 1 to 10, **Ultima I** rates as follows: puzzle quality-3.5; documentation-5; graphics-8; overall playability-7.5. □



From out of our minds

The Tail of Beta Lyrae™

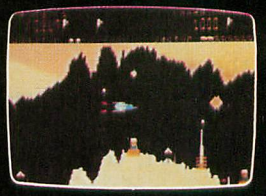
By Philip Price



THE TAIL OF BETA LYRAE will amuse and amaze you for many months to come. Why? Because it CHANGES as time goes by. So you'll never get bored. There's even original toe-tapping, finger-snapping music to entertain you while you play. Only $34.95

For the Atari Home Computers.*

Visa/Mastercard accepted. $2.00 shipping/handling (California residents add 6½% sales tax).



ACTUAL GAME SCREEN

DATAMOST, INC.™

The most out of our minds.

8943 Fullbright Avenue, Chatsworth, CA 91311-2750
(818) 709-1202

*Atari is a trademark of Atari, Inc.

Copyright © 1984
DATAMOST, Inc.
All Rights Reserved

CIRCLE #124 ON READER SERVICE CARD.

Ask Sally Forth

by Sally Forth

Dear Sally,

The **mUse** *article on page 112 of Issue #14 states that BASIC arrays eat up RAM. Each array element requires six bytes, meaning a 10 x 10 array uses 600 bytes. Is FORTH as wasteful?*

If FORTH uses six bytes for each array element, can I simulate arrays by using strings? The **mUse** *article says that BASIC strings are more RAM-efficient, using just "one byte per DIM allotment."*

(Unsigned)

All numbers in Atari BASIC are stored in a 6-byte floating-point format. The reason for using this format is that it allows BASIC to express very large and small numbers easily. The disadvantages of floating point are that it takes a lot of memory space to store each number, and a lot of processing time to handle them.

FORTH is considerably more flexible than Atari BASIC when it comes to storing numbers. With FORTH, you can define your own storage formats that are as efficient or wasteful as you like, limited only by the capacity of the machine.

The "default" format for a FORTH number is a 2-byte signed integer. This format allows you to express any whole number between –32768 and +32767. No fractions or decimal points are allowed; that is the price you pay for using only one-third of the storage space required by Atari BASIC.

How do you set up an array in FORTH? The method you use depends on what you're going to do with the array. Let's assume that you need an array called BOX with 10 elements in it. In Atari BASIC, you would type:

```
DIM BOX(9)
```

Notice that BOX is DIMensioned to 9, not 10. The reason is that BASIC numbers array elements starting with 0. If you DIMed BOX to 10 instead of 9, you'd actually be reserving space for 11 elements instead of 10. BASIC is wasteful enough without having useless array elements hogging up RAM!

The BOX array could be defined in FORTH like this:

```
VARIABLE BOX 18 ALLOT
```

Not too hard to grasp, except for that mysterious 18 ALLOT. What has the number 18 got to do with a 10-element array? It works like this: the FORTH word VARIABLE automatically reserves 2 bytes in the dictionary under the name BOX, enough for one signed integer. The 18 ALLOT phrase tells the compiler to reserve 18 additional bytes under the name BOX. Those 18 bytes are enough to hold 9 more 2-byte numbers. That gives BOX a total storage capacity of 20 bytes, or 10 2-byte numbers. See?

Now let's fill our BOX arrays with numbers. We'll assign each number a value equal to its position in the array, so that BOX (0)=0, BOX (1)=1, BOX (2)=2, etc. One way to accomplish this in BASIC is:

```
10 DIM BOX(9)
20 FOR I=0 TO 9
30 BOX(I)=I
40 NEXT I
```

In FORTH, you could try:

```
VARIABLE BOX 18 ALLOT

10 0 DO
     I I 2* BOX + !
     LOOP
```

The above FORTH example works okay, but it isn't as efficient as it could be. Because all of the numbers in the array are less than 255, we could use a single byte to store each one instead of two. And since the numbers are predefined, why not stuff them into the array at the same time the array is defined?

Here's an example that will show you what I mean.

```
LABEL BOX

0 C,  1 C,  2 C,  3 C,  4 C,
5 C,  6 C,  7 C,  8 C,  9 C,
```

The word LABEL simply marks a place in the dictionary and gives it the name BOX. The following sequence of C, words simply compiles the byte values 0 - 9 directly into the dictionary. You end up with a "string" of 10 bytes numbered 0 - 9, already initialized and ready to go without any external loops! How's that for efficiency?

I'll leave you with a semi-useful example of how to set up and access arrays in FORTH. Let's set up two arrays, XPOINTS and YPOINTS, each of which containing the fixed X and Y coordinates of ten points to be plotted sequentially. I'll show you the code that will set up the arrays and plot the points, in both BASIC and in FORTH.

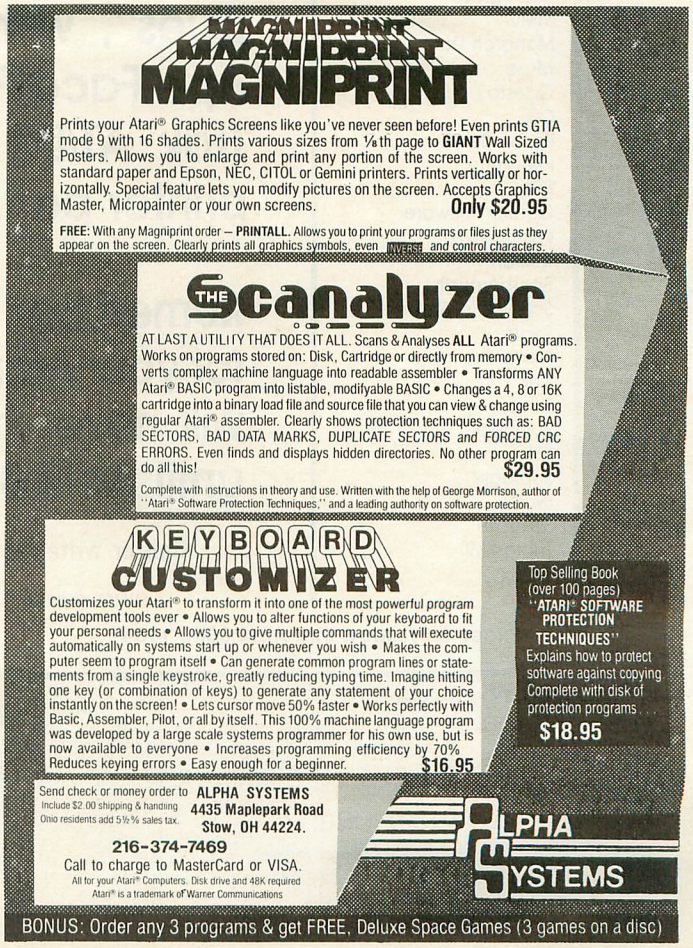
First, the BASIC code:

```
10 DIM XPOINTS(9),YPOINTS(9)
20 FOR I=0 TO 9
30 READ X:XPOINTS(I)=X:NEXT I
40 FOR I=0 TO 9
50 READ Y:YPOINTS(I)=Y:NEXT I
60 GRAPHICS 3:COLOR 1
70 FOR I=0 TO 9
80 PLOT XPOINTS(I),YPOINTS(I)
90 NEXT I
100 REM * X-COORDINATE DATA
110 DATA 20,16,20,24,12,28,16,20,24,20
120 REM * Y-COORDINATE DATA
130 DATA 6,8,8,8,9,9,10,10,10,12
```

Here's the same thing in FORTH:

```
( First set up the X and Y
( coordinate tables )

LABEL XPOINTS

20 C,  16 C,  20 C,  24 C,  12 C,
28 C,  16 C,  20 C,  24 C,  20 C,

LABEL YPOINTS

6 C,   8 C,   8 C,   8 C,   9 C,
9 C,   10 C,  10 C,  10 C,  12 C,

( The following word will read the
( coordinates out of the tables and
( PLOT them )

: PLOTEM

  3 GRAPHICS
  1 COLOR
  10 0 DO
       I XPOINTS + @ ( fetch x-coord )
       I YPOINTS + @ ( and y-coord   )
       PLOT          ( and PLOT them )
       LOOP
  ;                  ( simple, eh?   )
```

Remember: whenever possible, initialize an array by compiling the values directly into the dictionary. And use bytes instead of words when your data values are less than 255. You'll save all sorts of RAM space, which can then be used to accomplish more interesting things. □



CIRCLE #125 ON READER SERVICE CARD.

Great Software • Great Service • ~~Great~~ incredible Prices

Activision
Adventure International
Atari
Atari Program Exchange
Avalon Hill
Big Five
Bram
Broderbund
Budgeco
CBS
CDY Consulting
Computari
Continental
Datasoft
Designware
Eastern Software
Educational Software
Edu-ware
Electronic Arts
Epyx
First Star
Funsoft
Gamestar
Hayden
Imagic
Infocom
In-Home Software
Innovative Design
JV Software
L&S Computerware
LJK
Koala
Master Control Software
Microprose
MMG
Monarch Data Systems
Muse
Odesta
OSS
Parker Bros.
Penguin
PDI
Quality Software
Reston
Roklan
Scarborough
Scholastic Wizware
Sega
Sierra On-line
Sim
Sirius
Spinnaker
SSI
Sublogic
Swifty
Synapse
Thorn-EMI
Tronix
Versa

For almost a year, we've worked hard to bring you our selection of quality Atari software titles at low, low prices.

This month, we'll let you pick the specials. Every Atari program we sell — and that's well over one thousand programs — is 30%-50% off the suggested list price until April 30th.

During April, we'll also help you overcome the "Atari 850 Interface Blues", with superspecials on the Ape Face Printer Interface and the Trak AT/D2 Disk Drive with built-in printer port.

Remember, our prices on software for your Atari from almost everybody will never be lower than they'll be until April 30th!

Call or write for FREE CATALOG (specify Atari)

CIRCLE #126 ON READER SERVICE CARD.

Prices only good thru April 30th, 1984

Mail order prices do not apply at our retail location.

HOW TO ORDER:
Use mail or phone. For immediate shipment pay by Money Order, Certified Check, Cashiers Check, VISA, or Master Card (add 4% for credit card). Personal checks are accepted but require 1-3 weeks to clear. 10% deposit required for C.O.D.

SHIPPING & HANDLING:
$2.50 for orders under $50; free shipping over $50. Mass. residents add 5% sales tax. Prices are mail order only and are subject to change without notice.

SOFTWARE PLUS

(617) 369-1992
for orders and information
phone hours: 10-6 EST Mon.—Sat.

45 Walden St., Box AA, Concord, Mass. 01742

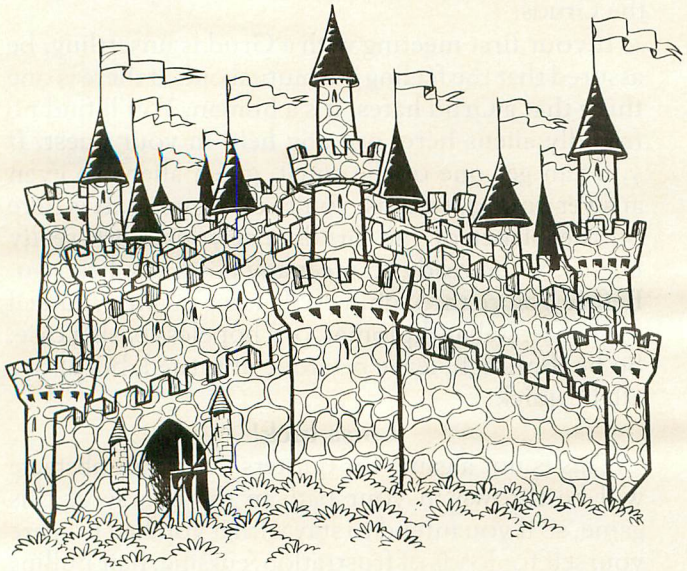
CIRCLE #126 ON READER SERVICE CARD.

EXODUS: ULTIMA III
by Lord British
ORIGIN SYSTEMS
P.O. Box 99
North Andover, Massachusetts 01845
48K Disk $59.95

by Cliff Chaput

It all started when I bragged to my friend, "Hey, Ted! I'm going to review **Ultima III**!"

"Neat, Kiffy," he replied. (I hate it when he calls me Kiffy.) "But I thought you didn't like role-playing games and medieval themes."



Role-playing? Medieval? Yech. Well, I thought, I'm stuck with it. I might as well review it. I shoved the disk in the drive and booted it with bad expectations. After viewing a modest title screen, the program began. Wow! I had to admit, the introduction was quite fabulous. Some heavy-duty programming must have gone into the title's "materialization." And there's a cute little dragon fight thrown in there. Very well done!

After that I ran into a problem. For those of us who are buying an **Ultima** program for the first time, beware: You must be prepared with a blank diskette. On this will go a copy of *the back side* of the **Ultima III** disk, your player disk.

Once you've got that over with, you may enter the magical land of Sosaria. The finely-detailed graphics really get you into the game. The landscape and the players are works of art. Take a moment to really appreciate them. The music is quite beautiful as well, not to mention appropriate.

First off, you get your player or group of four players prepared to battle, steal, and cast. Once your ensemble is ready, you journey around the island, exploring forests, entering towns and castles, searching for some unknown creature named Exodus. Hidden in mountain caves are dungeons, and once you enter one of those, it's like going through a maze (one might suggest keeping a map). Finding moongates, stealing money, killing trolls — it's all part of the fun. This is the type of game that'll keep you going for months. A true programming triumph.

But, as all programs do, **Ultima III** has its problems. For instance, the music can get very monotonous. Although there is a sound toggle key, it only turns off the movement and battle sounds, and the music goes merrily on.

And the ultimate of all problems, "The Glitch!" Now I'm not trying to stir up hot coals or anything, but my copy of **Ultima III** does not work. I wouldn't mention this if I thought it was an outside case, but good ole Ted is on his third **Ultima III**, and he says that it's going sour, too. I would really hate to knock such a wonderful product on a one in a million chance, but four out of four? That's not a good record.

In short, **Ultima III** is a must-have for your software collection. You might want to ask around to see if all those **Ultima III**s glitch; I doubt if they all do. You may have some problems at first, but once you get started, you'll never be able to stop. □

EPSON*, NEC*, PROWRITER*, GEMINI*, OKIDATA 92*
OKIDATA 82A/OKIGRAPH, M-T SPIRIT, DMP-80, PANASONIC KXP-1070

NEW!
ATARI*

The only self-booting grafix handler for dumps in **horizontal** format — all mach. lang. — **Lister** incl. — all modes — mixed modes — change aspect ratios, etc. while running other programs — assem ed — basic or no cartridge — demos, utilities, fonts, included — dump, create forms, stationery, calendars, requires interface. **$29.95**

diskwiz-II

Now for single/double density. Repair, explore, alter, duplicate, map, speedcheck, bad sector (810), block move, trace, special print capabilities, disassembler, new speed, new ease, new functions, special printing functions, excellent repair tool w/instr. — even better than before! The best repair/editor/duplicator at any price — still at the lowest price. (Updates avail. for a small fee.) **$29.95**

1st Class Postage Paid
California Residents add 6%, Foreign Orders add $2.50
C.O.D. add $3.00 — No credit cards
Prices subject to change
(213) 376-4105

ALLEN MACROWARE
P.O. Box 2205
Redondo Beach, CA 90278

* *Indicates Trademark of non-related company*

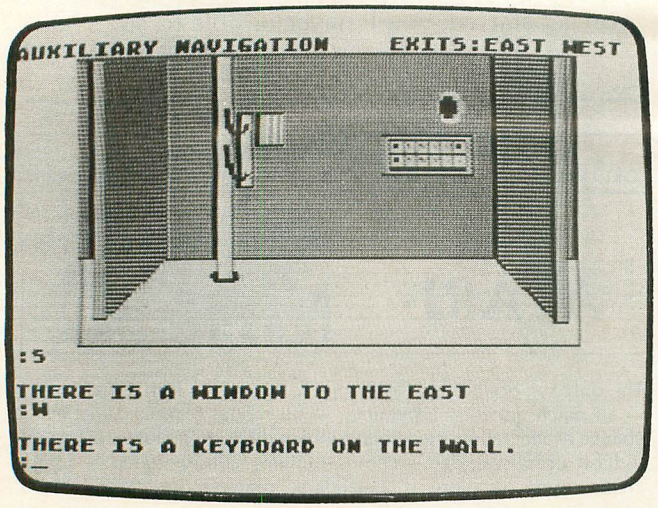
CIRCLE #127 ON READER SERVICE CARD.

GRUDS IN SPACE
by Chuck Somerville and Joe Dudar
SIRIUS SOFTWARE
Sacramento, California 95827
48K Disk $39.95

by Patrick J. Kelley

So, my friend, you wanna be a Space Jockey, huh? You wanna visit exotic places, strange new worlds, and hobnob with inscrutible aliens, huh? You say you wanna battle horrible maneating monstrosities and be a Galactic Hero with medals and all that, do ya? Don't ask for much, do ya, friend? I suppose you even wanna be a Rich Man for all your troubles, too.

Under most circumstances, these requests would be next to impossible to achieve one at a time, but put them all together and the odds of them happening to one person would be astronomical. However, if you're willing to invest a few dollars, a bit more time, and a lot more patience, you can get as close to all this as any Earthly mortal can. Prepare to match wits with **Gruds In Space**.



Gruds in Space.

The premise for this new graphics/text adventure from Sirius Software is the stuff of classic yarns. You are a lone wolf space pilot, approached by Earth's Armed Space Force to carry out a do-or-die mission, with the fate of our dear planet's fighting men in the balance. It seems that your ship is the only one in the solar system close enough and fast enough to fly a daring rescue mission. Your destination: The airless moon of Baranok, and site of a pitched battle. Unless precious fuel is rushed to the waiting Earth star force, the future looks pretty grim for ol' Terra and her boys in uniform. You are the last hope, and should you decide to accept this mission, you may be signing your own death warrant. Grimly you accept, solely out of patriotism and strong moral fiber. (Or is it because of the $1,000,000 cash reward waiting for you if you succeed?) Nevertheless, the game is afoot. But before you begin to spend that cool million, you better hear some more facts.

Let the game begin.

Firing up the Hyperwarp on board your ship, you set off in search of the fuel, and your place in the sun. Ahead of you lies the majestic planet Saturn, and the beginning of a great adventure. For it is not just a simple matter of just getting the fuel. First, you must beam down to one of Saturn's moons and establish contact with the natives in a local mining colony. All of your skills as a detective and a diplomat must be brought into play here, as you barter for the location of the fuel, hunt for clues, and come face to face with the Gruds.

If your first meeting with a Grud is unsettling, be assured that the feeling is a mutual one. If there is one thing that a Grud hates, it's a human. You'll find no friendly aliens here, nor any help in your quest. If you can get one of the squat, green aliens to even answer your questions, you're doing good. It is here on the planet of the Gruds that your work really begins, so be ready to meet any adversity head on. Before you're through, you'll either qualify yourself as an interstellar diplomat, or a hopeless basket case. So be thorough, leave no stone unturned, but above all, be quick.

Lotsa luck!

This is not a game for quitters. Many an obstacle will be thrown in your path in the course of this game, so if you intend to stay with it you had prepare yourself for loads of frustration, cursing, hair pulling and rewards beyond your wildest dreams. As a novice adventurer, I found myself on the verge of abandoning the game many a time, but a cool head prevailed and I'm not sorry at all. This is an enjoyable game that will provide many hours of thrills and delight, along with a real sense of accomplishment that cannot be found in your average shoot 'em up or blow 'em to Hades derivative.

But perhaps the best feature of **Gruds In Space** is the graphics. This is the most detailed and animated graphics/text adventure that I've ever seen, and belies a lot of love in its creation. Blinking eyes, twinkling stars, flashing lights and leering monsters fill every frame with a real character, and the continuity of shape and color are truly amazing. This game sets a standard that many other so-called "graphic" adventures fall far short of, both in concept and execution. In some cases, the animation is so well integrated that it becomes more than just an enhancement to the adventure, but a feature unto itself. I won't say any more to give away the plot or spoil the myriad surprises in store for you, so I'll just close by saying that if every picture is worth a thousand words, then **Gruds In Space** must qualify for at least a hundred volumes. □

MILES ACCOUNTING SYSTEM II™

The Finest Accounting Software Available for the Atari Today

It's good for business.

Who says the Atari is just a games machine? Not us. And we have the software to prove it.

There's no longer any reason to spend thousands for a "business computer" when your Atari and Miles Accounting System II can handle any accounting situation your business can throw at it.

Miles Computing has the hands-down, best accounting software available for the Atari today. We think you'll agree.

The Critics' Choice.

The critics agree. Here's what some of them said about Miles Payroll System:

"The package is flexible and capable of holding and manipulating a wide spectrum of information. In fact, it is hard to think of an area the program does not cover."
—*Desktop Computing*

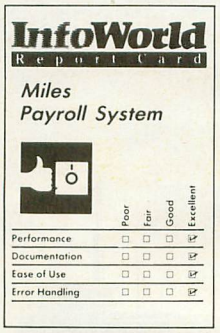
"The performance of Miles Payroll System is excellent. It handles all its promised features quickly and easily. ...It is an easy-to-learn program that is very powerful."
—*InfoWorld*

"The documentation is excellent—well-written, clear and concise."
—*Desktop Computing*

"So much for the Atari's reputation of being a games-only machine."
—*Desktop Computing*

Hardware Requirements: Atari computer with 48k, any 80 column (minimum) printer, and 2 disk drives (single or double density). Double density drives will greatly increase the capacity of Miles Accounting System II.

Price: $145 per module. Buy any 3 modules and get the 4th one free.



"In all respects—features, performance, ease of use, documentation and support—this is a professionally conceived and executed program. We look forward to other business software from Miles Computing that will serve the needs of Atari owners."
—*InfoWorld*

The Complete Accounting System.

There are six separate modules in the system. Each functions individually as a complete stand-alone program. All six modules can also be easily linked resulting in an extremely powerful integrated accounting system.

G/L General Ledger System***

Allows interactive maintenance of the chart of accounts file, including add, change/inquiry and delete. Handles up to thirteen accounting periods, supports multiple profit centers, prints source cross-reference reports, financial statements (Income Statement, Balance Statement, Statement of Cash Flow).

A/R Accounts Receivable System**

This comprehensive system processes and maintains records from customer invoices and cash receipts, calculates service charges, maintains sales history and credit rating records. Allows aging. Handles both open items and balance forward customers.

A/P Accounts Payable System***

Keeps accurate status of all outstanding obligations: prints cash requirements report, allows flexable payment selection, including partial payments, prints A/P checks and check register, prints vendor analysis report.

PAYROLL Payroll System

Cumulative totals are maintained for each employee, as well as complete reporting, check writing, and W-2 reporting. Allows weekly, biweekly, semimonthly or monthly pay periods, handles Federal, State and City taxes, FICA, SDI, Group Insurance, Federal and State Unemployment Insurance, maintains cumulative totals and Worker's Compensation, prints paychecks and W-2's. Gives 941 information.

I/C Inventory Control System**

Provides complete control of your resale inventory: not in stock items, items on order, items at or below reorder point, complete Vendor Item Report, suggested Purchase Order to Vendor. Allows for inventory costing by either average cost, LIFO or FIFO, and handles multiple pricing per items.

OE/I Order Entry Invoicing System**

For entry of sales orders and shipping data, and printing customer orders, invoices and shipping papers. May also used to maintain address records, generating back orders for partially filled orders. Orders are automatically printed when shipping dates are entered into the system. Provides O/E and editing, handles credit memos, prints picking tickets, price lists and stocking status reports.

** Available end of 2nd quarter 1984
*** Available 3rd quarter 1984

See for yourself.

Ask your local computer dealer to show you one of our self-running demos, or call us directly.

MILES Computing
Miles ahead of the pack.

MILES COMPUTING
7136 Haskell Avenue Suite 300
Van Nuys, California 91406
(818) 994-7901

CIRCLE #128 ON READER SERVICE CARD.

CRASH DIVE!

by

Brian Moriarty

24K Cassette 32K Disk

You're on maintenance duty aboard the USS **Sea Moss**, *patrolling the icy North Atlantic waters with an arsenal of twenty nuclear missiles.*

The **Sea Moss** *is no ordinary sub. She's the first to carry the Navy's new experimental sonar-jammer that can make her "invisible" to even the most sophisticated enemy sensors. The 50-kiloton cruisers in her missile bay are the pride of the Pentagon: fast, silent, incredibly accurate.*

The enemy would love to get their hands on the **Sea Moss** *and her secrets. It's not likely to happen, though. The only way they could possibly breach the hull would be from the inside — and your fellow crewmembers have been carefully handpicked for their unswerving patriotism and utter lack of imagination. No "moles" in this bunch of sailors. No, sir!*

The intercom in the equipment bay clicks to life. "I've got a bad line in the forward escape tube," a voice from the command deck crackles. "Wanna come up here and take a look at it?" You grab a screwdriver, scoot up a ladder and slam the hatch of the escape tube behind you.

It's all over in a few seconds. The General Quarters klaxxon blares to life. You hear the shrieks and choked coughing of friends as they stumble through the passages outside, and a single hoarse shout: "Gas!" Some poor sucker pounds weakly on the escape hatch. Then the alarm cuts off as suddenly as it began. Everything is silent as death. Frozen with fear, you sit trembling in the airtight escape tube, knowing that now it's just you and the **Sea Moss** *against whoever shut off the alarm.*

The game.

Crash Dive! is a machine-language text adventure that pits you in a race against time. As the sole survivor of a terrible act of naval sabotage, you must find a way to keep your ship out of the hands of The Enemy. No sacrifice is too great to achieve this important goal. The question is, how do you get rid of a giant nuclear submarine and everything in it?

As the start of the game, the *Sea Moss* is assumed to be cruising along the surface of the ocean. Your mission is as follows:

1. Find a way to survive in the submarine's poisoned atmosphere.
2. Get the sub under water, so that enemy ships will not be able to reach it easily. You have a limited number of moves after the game begins to accomplish this, or the Enemy will capture the sub and kill you on the spot!
3. Find a way to completely destroy the *Sea Moss*.

Some of these goals will be relatively easy to accomplish. Others will require careful thought and a little bit of resourcefulness. Don't forget that there may be somebody left alive on the *Sea Moss* besides yourself — and that somebody might not be very friendly!

We'll discuss the details of playing **Crash Dive!** in a moment. First, let's take a look at the program itself, and how to get it up and running on your computer.

Typing it in.

Listing 1 is an Atari BASIC program that will create an auto-booting version of **Crash Dive!** on disk or cassette. The DATA statements are listed in hexadecimal (base 16) in order to make the program as small as possible. It makes typing a little more difficult, but it's a necessary evil.

Listing 1 will not fit in a 16K Atari system. You'll need at least 24K of memory if you're using cassette, or 32K if you're using disk. However, the machine-language file created by **Listing 1** does fit in 16K. If you only have 16K in your computer, ask a friend with a larger system to help you type in and RUN the BASIC listing. After the boot tape or disk is made, you'll be able to enjoy the game on your 16K system.

Listing 2 is the assembly-language source code for **Crash Dive!**, created with the MAC/65 Macro Assembler. You do *not* have to type **Listing 2** into your computer to play the game (thank goodness!). It's provided for those readers interested in learning how the program works.

Follow the instructions below to make either a cassette or disk version of **Crash Dive!**

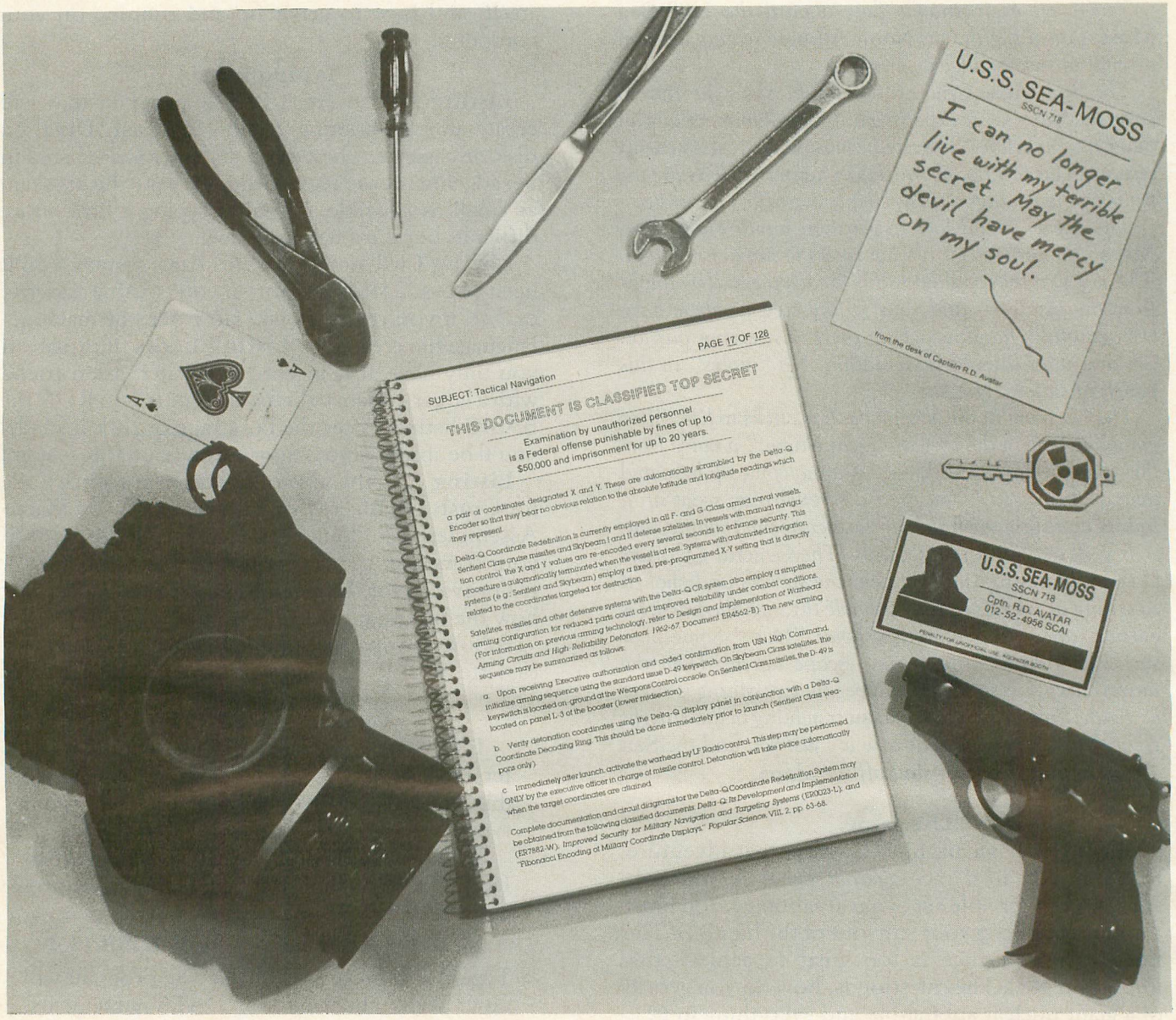
Cassette instructions.

1. Carefully type **Listing 1** into your computer (remember, you need at least 24K to do this). Use **C:CHECK** (page 30) to verify your typing.
2. When **C:CHECK** says the program is perfect, type RUN and press RETURN. The program will prompt you with:

MAKE CASSETTE (0) OR DISK (1)?

Type 0 and press RETURN. The program will now begin checking the DATA statements, printing the line number of each as it goes. It will alert you if it finds any problems. Fix any incorrect lines and re-RUN the program as necessary until all typos are eliminated.

3. When all DATA lines are correct, the computer will "beep" twice and prompt you to READY CASSETTE AND PRESS RETURN. Insert a blank cassette into your recorder, press the PLAY and RECORD buttons simultaneously and hit RETURN. The message WRITING FILE will appear, and the program will create a machine-language boot-tape version of **Crash Dive!**, printing the line number of each DATA statement as it goes. When the READY prompt reappears, the game is recorded and ready to play. CSAVE the BASIC program on a separate tape before continuing.
4. To play **Crash Dive!**, rewind the boot tape created by the BASIC program to the beginning. Turn your computer OFF and remove all cartridges. Press the PLAY button on your recorder



and turn ON your computer while holding down the START key. If you have a 600XL or 800XL computer, you must hold down the START and the OPTION keys together when you turn on the power. The computer will "beep" once. Hit the RETURN key and **Crash Dive!** will load and run automatically.

Disk instructions.

1. Type **Listing 1** into your computer and use **D:CHECK2** (see page 30) to verify your typing.

2. When **D:CHECK** says the BASIC code is perfect, type RUN and press RETURN. The program will ask:

MAKE CASSETTE (0) OR DISK (1)?

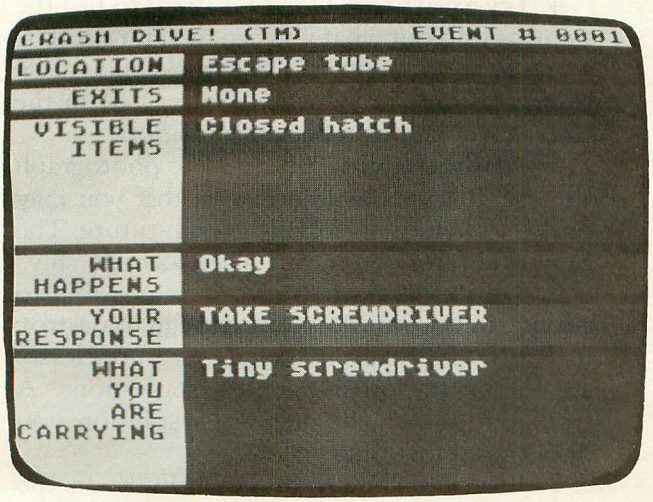
Type 1 and press RETURN. The program will begin checking the DATA statements, printing the line number of each statement as it proceeds. The program will alert you if it finds any problems. Fix incorrect lines and re-RUN the program as necessary until all typos are eliminated.

3. When all DATA lines are correct, the program will prompt you to INSERT DISK WITH DOS, PRESS RETURN. Put a disk containing Atari DOS 2.0S into drive #1 and press RETURN. The message WRITING FILE will appear and the program will create a binary AUTORUN.SYS file on the disk, displaying the line number of each DATA statement as it goes. When the READY prompt reappears, **Crash Dive!** is ready to play. Be sure the BASIC program is SAVEd out to a disk before continuing.

4. To play the game, insert the disk containing the AUTORUN.SYS file into drive #1. Turn your computer OFF, remove all cartridges and turn the computer back ON. **Crash Dive!** will load and run automatically.

Assuming everything went okay, you should now be looking at the **Crash Dive!** title screen, which includes the following prompt:

Press START to play new game
Press OPTION to restore old game



Crash Dive!

You haven't played the game before, so press the START key. Your TV screen should now look like the screen shown above. Note that the screen is divided into seven distinct sections or *windows*. From top to bottom, they are:

Event Window. The EVENT # counter in the top right corner keeps track of how many "events" have transpired since the start of the game. In general, each movement or other action you take during the course of the game counts as one event.

Location Window. This window contains a brief description of your current location.

Exit Window. The Exit Window tells you which directions you can go from your current location. Six vectors of movement are allowed: N (North), S (South), E (East), W (West), U (Up) and D (Down). "North" is towards the front of the submarine, "South" is aft, and so forth. If you can't move from a given location for some reason, the Exit Window will read "None."

Visible Items Window. This window displays a list of all the things you can see at your current location. Up to six items may be present in a location at any one time. Unoccupied locations will contain "Nothing."

What Happens Window. The What Happens Window reports on the results of your actions and lets you know if anything interesting is happening on board the *Sea Moss*. Keep an eye on this window — it may offer valuable information you will need to complete the adventure.

Response Window. This 2-line window is the communications link between you and the game. The commands and sentences you type into the Response Window tell the computer how you want to proceed. Each new line scrolls up into the top half of the window after you hit RETURN, so that you can see what you just typed. A blinking green underline keeps track of your position.

Inventory Window. Look here for a list of all the items you are carrying. Up to six items may be held at one time. If you're empty-handed, the window will say "Nothing."

Talking to Crash Dive!

Like most text adventures, **Crash Dive!** understands two-word sentences in the form:

VERB NOUN

The single space character between the verb and the noun is required. Don't worry about capitals vs. lower-case, numbers or funny characters — **Crash Dive!** has a "smart" keyboard handler that will snarl at you if you try entering anything illegal.

The best way to learn how to talk to **Crash Dive!** is to play with it. Let's use the opening screen as an example. The Location Window says you're in the escape tube, with no obvious exits and nothing in your inventory. You can "see" a closed hatch and a tiny screwdriver. Type the sentence TAKE SCREWDRIVER and you'll see the screwdriver vanish from the Visible Items list and reappear in your inventory. Simple, right?

You can interact with objects on the screen much like you can in real life. Type EXAMINE SCREWDRIVER and the What Happens Window will tell you that it "Seems ordinary." Now try EXAMINE HATCH and learn something interesting about the escape hatch. If you try to TAKE HATCH, you'll find out what happens when you attempt something impossible. DROP SCREWDRIVER will put the tiny screwdriver back in the Visible Items list.

You may be tempted to type OPEN HATCH, but if you read my little introductory tale carefully you'll know better than to try it. Think about your situation for a while and you'll discover a way to explore the rest of the *Sea Moss* without suffocating!

Commands.

Crash Dive! also understands a limited number of single-character commands. These are used to control your movement around the sub, and to perform special "internal" game functions. The following commands are recognized by **Crash Dive!**:

Movement Commands

N - North S - South E - East
W - West U - Up D - Down

Internal Commands

X - Mark Game Position
Q - Quit/Restart Game
A - Again (Repeat Last Sentence)

The movement commands are easy to use. Just consult the Exit Window to see which vectors are available, and type the initial of the direction you want to

go. The program will scold you if you type an illegal direction.

Saving your game.

The "X" (Mark Game Position) command is used when you want to save the current status of your game. Type X/RETURN and you'll see the following prompt:

SAVE GAME TO DISK OR CASSETTE?

If you're using a disk drive, insert a disk containing Atari DOS 2.0S into drive #1 and press the "D" key. Your game will be saved out in a few seconds and you'll return to the main screen.

If you're using cassette, insert a blank tape into your recorder and press the "C" key. The computer will "beep" twice. Press the PLAY and RECORD keys on the recorder simultaneously and hit RETURN. The game will be saved and you'll return to the main screen.

Starting over.

The "Q" (Quit/Restart) command is used when you want to restart the game from the beginning, or restore a game you have previously saved to disk or tape. Type Q/RETURN and you'll see the familiar **Crash Dive!** title screen. Press the START key if you want to start over from scratch. Press OPTION and you'll be asked:

RESTORE FROM DISK OR CASSETTE?

If your game was saved on disk, insert the game disk into drive #1 and press the "D" key. Your game will automatically resume at exactly the point where you left it.

If your game was saved on tape, cue the tape to the beginning of the saved game and press the "C" key. The computer will "beep" once. Press the PLAY key on the recorder and hit RETURN. Your game will resume at the point where you left it.

The A command.

The third and last command recognized by **Crash Dive!** is "A," which means Again. This command re-executes the last sentence you typed as if you had typed it in again yourself. The A command only repeats your last sentence (verb/noun); it will *not* repeat single-character commands.

Hints for successful play.

1. **Draw a map.** You'll have a hard time remembering the layout of the *Sea Moss* unless you draw a map. There are no mazes in this adventure, but a map will help you recall where interesting items are located and how the various rooms are connected.

2. **Examine everything.** Objects may have important features that will not be evident unless you examine them closely. Most of the items you discover in the game are essential to your success (though I may have left a couple of red herrings lying around. . .).

3. **Save your game frequently.** Use the X command to save your current status after important discoveries and breakthroughs, and before trying anything that might be dangerous. Otherwise you may find yourself starting all over again in the escape tube.

4. **Try anything.** Don't be afraid to test the game to find out what you can or can't do. The worst that can happen is that you will be captured and killed by enemy agents, shot in the back or cooked by a blast of radiation.

5. **Study the clue photo.** The photograph on page 46 contains information that you may find very helpful in solving the adventure. The game will refer you to this photo occasionally.

6. **Don't give up hope.** It *is* possible to survive in the *Sea Moss* long enough to destroy it. Really and truly it is! If you're hopelessly stuck, ask for other people's suggestions. A fresh outlook might uncover a solution you didn't think of yourself.

7. **Use C:CHECK or D:CHECK** on the program before you try to use it. It only takes one byte in the wrong place to make **Crash Dive!** totally unplayable.

8. **Don't call ANALOG.** We are absolutely not giving out adventure hints over the telephone! If you're really stuck, send me a self-addressed, stamped envelope at the following address:

CRASH DIVE CLUES
c/o Brian Moriarty
ANALOG Computing Magazine
P.O. Box 23
Worcester, Mass. 01603

BASIC Listing.

```
10 REM *** CRASH DIVE ***
20 TRAP 20:? "MAKE CASSETTE (0), OR DI
SK (1)";:INPUT DSK:IF DSK>1 THEN 20
30 TRAP 40000:DATA 0,1,2,3,4,5,6,7,8,9
,0,0,0,0,0,0,10,11,12,13,14,15
40 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
E 1000:TRAP 120:? "CHECKING DATA"
50 LINE=LINE+10:? "LINE:";LINE:READ DA
T$:IF LEN(DAT$)<>90 THEN 220
60 DATLIN=PEEK(183)+PEEK(184)*256:IF D
ATLIN<>LINE THEN ? "LINE ";LINE;" MISS
ING!":END
70 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
EX(D1)*16+HEX(D2)
80 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
EAD CHKSUM:GOTO 50
90 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
TOTAL=TOTAL-1000
100 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
 THEN 50
110 GOTO 220
120 IF PEEK(195)<>6 THEN 220
130 IF PASS=0 THEN 170
140 IF  NOT DSK THEN 160
150 PUT #1,224:PUT #1,2:PUT #1,225:PUT
 #1,2:PUT #1,128:PUT #1,31:CLOSE #1:EN
D
```

```
160 FOR X=1 TO 25:PUT #1,0:NEXT X:CLOS
E #1:END
170 IF  NOT DSK THEN 200
180 ? "INSERT DISK WITH DOS, PRESS RET
URN";:DIM IN$(1):INPUT IN$:OPEN #1,8,0
,"D:AUTORUN.SYS"
190 PUT #1,255:PUT #1,255:PUT #1,128:P
UT #1,31:PUT #1,190:PUT #1,58:GOTO 210
200 ? "READY CASSETTE AND PRESS RETURN
";:OPEN #1,8,128,"C:":RESTORE 230:FOR
X=1 TO 40:READ N:PUT #1,N:NEXT X
210 ? :? "WRITING FILE":PASS=2:LINE=99
0:RESTORE 1000:TRAP 120:GOTO 50
220 ? "BAD DATA: LINE ";LINE:END
230 DATA 0,55,88,31,127,31,169,0,141,4
7,2,169,60,141,2,211,169,0,141,231,2,1
33,14,169,56,141,232,2
240 DATA 133,15,169,128,133,10,169,31,
133,11,24,96
1000 DATA A2008E4402E886092065E44CB81F
7070707042403C90100290100290100202020202
0202901002029010020280B0,119
1010 DATA 1002020202020270418E1FA2FF9A
20B22AA20CA00620772AA906A035205D2AA219
A00720772AA917A035205D2A,527
1020 DATA A208A00920772AA922A035205D2A
A205A01020772AA93BA035205D2AA203A01220
772AA95AA035205D2AA9068D,277
1030 DATA 2C3C8D2D3CA9228D2F02207C2AAD
1FD0C907F0F9AE1FD0E007D0F9C906F00AC903
F00C20802A4C142020E92F4C,59
1040 DATA 9B20A205A01420772AA97DA03520
5D2A207C2A20252BC944F00DC943F00F20382E
20802A4C472020422E4C6420,682
1050 DATA 20492EA9039D4203A9049D4A03A9
009D4B032056E430DAA210A9C09D4403A93A9D
4503A9489D4803A9019D4903,753
1060 DATA A9079D42032056E430BA20382E20
B22AA98E8D3002A91F8D3102A99CA035205D2A
A90C8553A9C1A035205D2AA9,28
1070 DATA D7A035205D2A20152B20152B2015
2B20152B201D2BA903A036205D2A201D2BA924
A036205D2A20152B20152BA9,715
1080 DATA 278553A9C48DC002A9308D01D0A9
CA8D02D0A9408D03D0A2FF8E0ED08E0FD08E10
D0E88E04D0E88E6F02A9038D,442
1090 DATA 89D08D0AD08D0BD0A9F08581A91E
8582A90D8552A085A22CA906205CE4A9318D00
02A92A8D0102A9C08D0ED4AD,551
1100 DATA C23A4CA823A2FF9A20BC2CEEC03A
D003EEC13AADC23AD01BA92020D82DF01420B2
2AA208A00320772AA928A03A,724
1110 DATA 205D2A4C192EADC03AC920D01EAD
C13AD019ADC73AD01420B22AA207A00320772A
A96DA039205D2A4C192EADCB,209
1120 DATA 3AF00ACED23AD005A9008DCB3AAD
CC3AF020A91C20D82DF019ADCB3AD01420B22A
A208A00320772AA911A03920,112
1130 DATA 5D2A4C192EADD33A101420B22AA2
09A00320772AA900A03A205D2A4C192EADC23A
C912D00DA90420EB2DD006CE,122
1140 DATA D33A4C0022A9018DD33AADCE3AD0
35AD0AD229F88DC33AAD0AD229F88DC43AADC9
3AF020ADC73A18D869088DC7,759
1150 DATA 3A10148DCE3A20112DA955A03820
5D2AA917A03A205D2AA9228D2F0220E92B2011
2DA684E001D027AD0006A208,480
1160 DATA DDD62EF010CA10F820092DA94DA0
36205D2A4CD02EBD53238591BD5C2385926C91
00A920858A858BA201BD0006,261
1170 DATA C920F012E8E48490F420092DA959
A036205D2A4CD02E868DE002900 2A202BD0006
9589CA10F8A200868E868CA0,417
1180 DATA 00B98900DDDF2ED008E8C8C00390
F2B00DE68EA68CE8E8E8E05A90E1B0BEA58EC9
1CD00AA92DA037205D2A4CD0,704
1190 DATA 2EA68DE8A000BD0006998900E8C8
C00390F4A200868F868CA000B98900DD6E2FD0
08E8C8C00390F2B018E68FA6,490
1200 DATA 8CE8E8E8E07E90E120092DA962A0
36205D2A4CD02EA58E85A90AAABD362F8591E8
BD362F8592A58F85AAC927F0,358
1210 DATA 0DC928F013AABDE23B85906C9100
A941A037205D2A4CD02EA956A037205D2A4CD0
2E8F8F8F8F8F8F65FE712323,354
1220 DATA 2323232323232400C20752AA98F
A036205D2AA222A00C20772A8680207C2A2025
2BC959F008A20C209E2A4CD0,17
1230 DATA 2E4CB81FB593100AA97DA036205D
2A4CD02ED8859F207C24A59F8DC23A20DC23A0
05B1A0999300B1A299990088,0
1240 DATA 10F3A201209E2AA00120752AAEC2
3ABD3E32BC5432205D2A205F2D202A2D20972D
4CC92D0A85900A1865908590,906
1250 DATA 1869DA85A0A93A690085A118A590
695E85A2A93B690085A360207C2420B22AA205
A00B20772AA966A03A205D2A,355
1260 DATA A9228D2F02207C2A20252BC944F0
0DC943F00F20382E20802A4C1A2420422E4C37
2420492EA9039D4203A9089D,771
1270 DATA 4A03A9009D4B032056E430DAA210
A9C09D4403A93A9D4503A9489D4803A9019D49
03A90B9D42032056E430BA20,981
1280 DATA 382E4C9B20A5A9858EA5AA858F4C
1923ADC23A20DC23A005B9930091A0B9990091
A28810F360A58FC916B034C,138
1290 DATA 572EA59020D82DD0034C8D2EA590
20EB2DF0034C842EA590C914D00AA900A03820
5D2A4CCA2EC92AD027ADC23A,4
1300 DATA D022A91B20D82DF00AA941A03A20
5D2A4CCA2EA90720EB2DD00AA985A03A205D2A
4CCA2E20D62DF0034C722EA6,581
1310 DATA A5A4A4B59999D43AA9FF9599205F
2D20972D4CC92D20D82DF0034C962EADC23AC9
11F01820E92DF0034C7B2EA6,958
1320 DATA A5A4A4B9D43A9599A9FF99D43AD0
D0207925D0E3A4A4A6A5B9D43A9D5E3BA9FF99
D43AA988A039205D2AA590C9,566
1330 DATA 2AF009205F2D20972D4CCA2EADCD
3AD0F2A926A2058DCD3A9DE23BA26C9D5E3B20
7925A6A5A9159D5E3BD0D7A2,873
1340 DATA 6DBD5E3B3007E8E07190F68A6086
A5A90060C916B0034C572EC91CF00EC920F00A
C92AF0034C692E4C92244C08,768
1350 DATA 2520EB2DF00AA59020D82DF0034C
842EA590C909D026ADD13AF0034CB12E20E92D
F0034C7B2EA993A037205D2A,709
1360 DATA A9188DD13AA6A59599205F2D4CCA
2EC90FD0152044DAADC73A85D420FB2DA9DBA0
38205D2A4CCA2EC911D04BAD,74
1370 DATA C23AC90BF008ADC33AAEC43AD006
ADC53AAEC63A85A786A8A9D3A038205D2AA211
A00920772A2044DAA5A785D4,613
1380 DATA 20FB2DA9D7A038205D2AA211A00A
20772A2044DAA5A885D420FB2DD051C920D026
ADD03AF0034CB12E20E92DF0,873
1390 DATA 034C7B2EA993A037205D2AA9218D
D03AA6A59599205F2D4CCA2EC915D017ADCF3A
D009A9C8A038205D2AD015A9,815
1400 DATA BAA038205D2AD00C0AAABD9F26E8
BC9F26205D2A4CCA2EA8373F398437C937DA37
84373F39DA37183884378437,371
1410 DATA 843784378437DA378437B6378437
84375B3800388437843 77D387D388437953884
37843784377D387D38843784,576
1420 DATA 3784378437B63784378437953 8DA
37843774380A209DD0527F006CA10F84C572E4C
A8250224080F111718 1A1E1F,303
1430 DATA 20D82DD0034CA82EA59020EB2DF0
034C842EA590C90AD02AA203ADC83AF012A900
8DC83AA9029DE23B8599205F,433
1440 DATA 2D4CCA2EAD5F3BC907F0034CBA2E
8EC83AA924D0E4C90BD021AEC93AF00EA9008D
C93AA9F3A038205D2AD00BE8,884
1450 DATA 8EC93AA902A039205D2A4CCA2EC9
0CD00DADC53AD81869088DC53A4C0F26C90DD0
0DADC63A38D8E9088DC63A4C,754
1460 DATA 0F26C90ED03BADC53ACDC33AD033
ADC63ACDC43AD02BADCA3AF02620B22AA90E8D
C6028DC802A9008DC502A20C,86
1470 DATA A00B20772AA955A03A205D2AA922
8D2F024CD4274CBA2EC90690034C572E20EB2D
F0034C842EA590D014A9228D,370
1480 DATA CC3A8599A9048598205F2D202A2D
4CC92DC901D009A93FA039205D2AD031C903D0
09A9C9A037205D2AD024C905,904
1490 DATA D009A9C7A036205D2AD017C922F0
0CC923F008C925F004C927D00AA953A039205D
2A4CCA2E4C572EA91520D82D,664
1500 DATA F0034CC32EA59020EB2DF00AA590
20D82DF0034C842EADCF3AD00AA9C8A038205D
2A4CCA2EA955A038205D2AA9,732
1510 DATA 008DCF3AA590C901D01EA99FA039
205D2AA9238599A2019DE23BE89DE23BA90385
96205F2D202A2D4CCA2E20D8,686
```

```
1520 DATA 2DF0034C962EA590C91AD00AA9AF
A039205D2A4CCA2EC918F0034C692EADC23AC9
13D0F6A9278599A2069DE23B,950
1530 DATA A9158594205F2D202A2D4CC92D20
EB2DF0034C842EA58FC91890034CA82EA590C9
14D0034C9224C903F0034C57,422
1540 DATA 2EA91D20D82DD00FA9258599A204
9DE23BA9118594D0C0A91620D82DD00AA9C4A0
39205D2A4CCA2E4CC32EA58F,549
1550 DATA C926F0034C692EADCB3AD00BA909
8DCB3A8DD23A4CC92D4C8D2EA92120D82DF003
4CC32EA59020EB2DF0034C84,612
1560 DATA 2EA590C901D00AA9DBA039205D2A
4CCA2EC906F0034C572EA9288599A2079DE23B
8DCA3A4CD52820EB2DF0034C,609
1570 DATA 842EA590C907F0034C572EA91920
D82DD02FADC83AF01420B22AA209A00320772A
A9E9A039205D2A4C192EA929,111
1580 DATA 859AA2089DE23BA9028D883BA203
9DE23B4CD5284CC32E20D82DF0034C962EA590
C91EF0034C572EA4A4A9FF99,475
1590 DATA D43AA941A038205D2AA91420EB2D
D00FA6A5A92A9599A2169DE23BE89DE23B205F
2D20972D4CCA2E20D82DD003,458
1600 DATA 4CA82EA91E20D82DF0034CC32EA5
9020EB2DF0BA4C842EC91CF007C920F0034C57
2E4C9224488A489848A685BD,514
1610 DATA 4F2ABC562A8D0AD48D18D08C0DD0
E68568A868AA68407060706070006000000000
00F000A2008D44038C4503A9,744
1620 DATA 098D4203A97F8D48038E49034C56
E4A20D86558454604A919D002A9648D00D2A9AA
8D01D2A9008514A514C905D0,425
1630 DATA FAA2008E01D2CA8EFC0260BDC32B
8586BDD62B8587A018A9009186B810FB60A940
8D0ED4856AA200A90C8D4203,815
1640 DATA 2056E4A2008E4B03A9038D4203A9
F48D4403A9348D4503A90C8D4A032056E4A900
8D00D48D2F028D0DD08D0ED0,166
1650 DATA 8D0FD08D10D0A2089DC002CA10FA
A90E8DC502A9748DC402A2018EF002E88652A9
7085108D0ED260A9EDA03520,568
1660 DATA 5D2A60A9F8A0352052D2A60ADFC02
C9FFF0F9A8A2FF8EFC0229C0F00620802A4C25
2B98A20DDD712BF0F2CA10F8,534
1670 DATA B97F2BC920F010C99BF00CC97EF0
08C96190DCD838E920A07F8483A4838C1FD0A2
08CA10FDC68310F2601C2C27,419
1680 DATA 3C36370F2002222606070E6C6A3B
8A8B6B2B2A6F8070759B692D3D7680638C8D62
787A34803361B3532312C20,818
1690 DATA 2E6E806D2F81728065797F747771
398030377E383C3E6668648082677361204000
6075759DC5ED153D658DB5DD,540
1700 DATA 052D557DA5CDF51D3C3C3C3C3C3D
3D3D3D3D3E3E3E3E3E3E3FD8A218A9209D
0006CA10FAA00C20752AA2FF,621
1710 DATA 86808EFC02E8868420252BC920F0
08C97EF004C99BD0062080204CF42B20FB2C20
56E4E68420252BC99BF032C9,70
1720 DATA 7ED010C68430E320FB2C2056E4A5
84F0BBD0E520FB2C2056E4E684A584C91890D7
20802A20252BC99BF006C97E,18
1730 DATA F0D4D0F020FB2C86802056E4A018
18B92D3E99053E2A2A2A2A2903AAB9053E291F
1DBF2B990006A900992D3E88,266
1740 DATA 10DF60A9008585A580F029D8A555
0A0A1869308D00D0AD2B02F00BA9F08D5B2A85
81A93C8582A581C682D008A0,526
1750 DATA 1E848249F085818D5B2A4C5FE4A9
B085AB85AC85ADA99B85AFADC03A85D4ADC13A
85D520AAD920E6D8D8A0FFC8,224
1760 DATA B1F310FBA203B1F3098095ABCA88
10F6A221A00020772AA9ABA000205D2A60A20B
8E4203A2008E48038E490360,941
1770 DATA A945A0362052D2A60A209209E2AE8
209E2AA0094C752AA20D209E2AE8E01390F860
A202209E2AA000A20086A6B5,496
1780 DATA 93300ABD592D999D3CE6A6C8C8E8
E00690EDA5A6D00CA00220752AA90CA039205D
2A602E33253735 24A203209E,231
1790 DATA 2AE8E00990F8A00320752AA20086
A6868CB599300CE6A6AABD9E34BCC934205D2A
A68CE8E00690E7A5A6D007A9,45
1800 DATA 25A037205D2A60201F2DA00D2075
2AA20086A6868CBDD43A300CE6A6AABD9E34BC
C934205D2AA68CE8E00690E6,2
1810 DATA A5A6D007A925A037205D2A602011
2DA9A4A036205D2A4CCA2EA9FFA205DDD43AF0
05CA10F88A6086A4A90060A9,123
1820 DATA FFA205D599F005CA10F98A6086A5
A9006020AAD920E6D8D8A0FFC8B1F310FB297F
91F3C8A99B91F3A5F3A4F44C,196
1830 DATA 5D2AA20AA00520772AA92AA03920
5D2AA218A00720772AA960A039205D2A4CE81F
A210A90C9D42034C56E4A9FA,384
1840 DATA A0344C4D2EA9F7A034A2109D4403
989D450360A96BA036205D2AD06AA9C7A03620
5D2AD061A902A037205D2AD0,92
1850 DATA 58A9D9A036205D2AD055A9EDA036
205D2AD04CA9BCA036205D2AD043A9A9A03620
5D2AD03AA913A037205D2AD0,87
1860 DATA 31A96BA037205D2AD022A978A037
205D2AD019A984A037205D2AD010A9E3A03820
5D2AD007A9A4A038205D2A20,589
1870 DATA 7C2A4C47212080 2A4C3A224E5345
57554451584154414B474554505 54C44524F52
454D4C4F4F45584153454152,910
1880 DATA 4541505553505 2454F5045434C4F
5553454B494C53484F464952425245534D4149
4E53554E53484F4C554E4C43,348
1890 DATA 5554504F554C554247524557454 1
474F2092249224C32E08258C25A825A825A825
F5260F270F27DA27A82E692E,68
1900 DATA C32E45284528A82EA82EA228DE28
292945297F29C929072A072A232A484154444F
4F4C4F43534341475241 5452,464
1910 DATA 4141495253574943414253494743
4150475245524544474F4C53494C5748494741
555045525343 4F4449534455,818
1920 DATA 43534C4F424F4C554E4950495353
43524E4F5449442043555443415257524 54D41
534741534B4E495348414D41,182
1930 DATA 4E5355494B4559425245425554 49
4E56A213A9009DC03ACA10FAA200A9FF9D5E3B
9DDA3AE8E08490F5A2059DD4,490
1940 DATA 3A95999593CA10F6A2018ECF3A8E
C23AA9208DC53AA9A88DC63AA225BD48309DE2
3BCA10F7A227BD6E30BC9630,254
1950 DATA 99DA3ACA10F4A21FBDBE30BCDE30
995E3BCA10F46000010102030405060708090A
0B0C0D0E0F10101112131414,334
1960 DATA 151617181911A1B1C1C1D1E1F2021
02000404020601 0F0604090705060906 0A0813
09150F0D0C0F0C13100E040F,1
1970 DATA 0E130F1412091313 0B010C0D1418
191C1D2024252627 2D323638393B3F47494A51
565A5B5C5D5E63666E727475,366
1980 DATA 767B7E8214070016080917011902
0A0F0B10111F110C0D1B1C031E1A1D12040513
0E2006000106070C1213181E,480
1990 DATA 2A2B3031363C3D4243444484E5455
5A60666C727374787E536F6E617220737068 65
72659B4573636170652074 75,551
2000 DATA 62659B4163636573732074756E6E
656C9B4361707461696E27732071756172746 5
72739B466F72776172642070,149
2010 DATA 617373616765 9B526164696F2072
6F6F6D9B4C6F6E6720636F727269646F729B53
6F6E6172207374617469 6F6E,831
2020 DATA 9B42616C6C61737420636F6E7472
6F6C9B436F6D6D616E642073746174696F6E9B
4E617669676174696F6E2063,480
2030 DATA 656E7465729B557070657220 6D69
7373696C65206261799B546F727065646F2072
6F6F6D9B576561706F6E7320,124
2040 DATA 6C6F636B65729B53686F77657220
7374616C6C739B437265772773207175617274
6572739B47616C6C65799B56,873
2050 DATA 656E74696C6174696F6E20647563
749B46616E20726F6F6D9B4D697373696C6520
636F6E74726F6C9B45717569,566
2060 DATA 706D656E74206261799B4C6F7765
72206D697373696C65206261799BFE0B172538
4853616F7F8FA1B3C0CFDDED,642
2070 DATA F4050E1E2C3031313131313131313
3131313131313131313232323236C6F736564
206861746368 9B4C6F636B65,863
2080 DATA 6420646F6F729B426C616E6B2073
63616E6E65729B436C6F736564206772617465
9B547261697 46F7220776974,440
2090 DATA 6820706973746F6C9B436C6F7365
64206169726C6F636B9B4C6F636B6564206172
6D696E672073776974636 89B,37
```

```
2100 DATA 506F7765722063616626C659B5369
676E9B44656164206361707461696E9B477265
656E20627574746F6E9B5265,632
2110 DATA 6420627574746F6E9B476F6C6420
627574746F6E9B53696C766572206275747 46F
6E9B57686974652062757474,308
2120 DATA 6F6E9B44657074682067617567 65
9B5065726973636F70659B4469676974616C20
646973706C61799B44756374,33
2130 DATA 20646F776E20746F2066616E2072
6F6F6D9B536C6F7420696E206169726C6F636B
9B426F6C7465642D646F776E,417
2140 DATA 20736F6E617220756E69749B5069
73746F6C9B54696E792073637265776472697 6
65729B537569636964652 06E,80
2150 DATA 6F74659B536563757269747920 49
449B4361626C652063757474657273 9B436172
649B5772656E63689B476173,769
2160 DATA 206D61736B9B44756C6C206B6E69
66659B5368616D706F6F9B5461637469637320
6D616E75616C9B5261646961,400
2170 DATA 74696F6E2073756974 9B4B65799B
4F70656E2068617463689B4F70656E20646F6F
729B4163746976652073636 1,15
2180 DATA 6E6E65729B4F70656E2067726174
659B4465616420747261697 46F729B4F70656E
206169726C6F636B9B416374,661
2190 DATA 697661746564207377697463689B
536576657265642063616 26C659B526164696F
6163746976652073 6F6E6172,261
2200 DATA 20756E69749B6A7783919EB2C1D6
E2E7F4010C1826333F49596F7F969DAEBBC7D5
DAE1EAF5FD0C1B1F2A34434E,994
2210 DATA 5B6879873232323232323232323232
3233333333333333333333333333333333333333
3333333434343434343434343434,534
2220 DATA 3434453A9B433A9B44313A47414D
452E4441549B4272769616E204D6F7269617274
7927739B435241534820444 9,305
2230 DATA 56459B2843293139383420414E41
4C4F4720436F6D707574696E679B5072657373
20A0D3D4C1D2D4A020746F20,786
2240 DATA 706C6179206E65772067616D659B
5072657373 20A0CFD0D4C9CFCEA020746F2072
6573746F7265206F6C642067,850
2250 DATA 616D659B526573746F7265206672
6F6D20C469736B206F7220C361737365747465
3F9BA0C3D2C1D3C8A0C4C9D6,478
2260 DATA C5A1A0A8D4CDA9A0A0A0A0A0A0C5
D6C5CED4A0A3A0A0A0A0A0A09BA0CCCFC3C1D4
C9CFCEA020A0A0A0A0C5D8C9,405
2270 DATA D4D3A09BA0A0D6C9D3C9C2CCC5A0
20A0A0A0A0C9D4C5CDD3A09BA0A0A0A0A0A0A0
A0A0A09BA0A0A0A0A0D7C8C1,221
2280 DATA D4A09BA0A0C8C1D0D0C5CED3A020
A0A0A0A0A0D9CFD5D2A020A0D2C5D3D0CFCED3
C5A09BA0A0A0A0A0A0D9CFD5,241
2290 DATA A020A0A0A0A0A0A0C1D2C5A020A0
C3C1D2D2D9C9CEC7A09B53796E7461783A9B42
616420636F6D6D616E649B42,402
2300 DATA 6164207665726 29B426164206E6F
756E9B5468617427732069 6D706F737369626C
659B43616E277420676F2074,759
2310 DATA 6861742077617 99B547970652059
20746F207175697420676 16D653A9B4F6B6179
9B416C726561647920686F6C,156
2320 DATA 64696E672069749B49736E277420
6865726 59B43616E277420646F207468617420
7965749B596F757220617 26D,487
2330 DATA 7320617265206 6756C6C219B4E6F
7420656E6F75676820726F6F6D206865726 59B
4265206D6F726520737065 63,761
2340 DATA 6966696 39B596F7520646F6E2774
2068617665206974 9B4E6F7468696E679B5479
7065204E2053204520572055,874
2350 DATA 206F722044 9B52656665722 0746F
2069742062792063 6F6C6F729B547970652049
20666F7220696E76656E746F,64
2360 DATA 72799B446F65736E27742068656C
709B5768792062 6F746865723F9B5365656D73
206F7264696E6172799B596F,700
2370 DATA 7520666F756E6420736F6D657468
696E67219B4974277320616972746967687 49B
456E656D7920617070726F61,158
2380 DATA 6368696E67219B536372657765 64
20696E20706C6163659B4C6F6F6B732064616E
6765726F75739B4E65656473,705
2390 DATA 206865792074 6F20616374697661
74659B426F6C747320617265207469674 20
2620727573747 99B44414E47,919
2400 DATA 45523A2052616469617469 6F6E20
7A6F6E65219B427261696E7320626C6F776E20
6F75749B5368616D706F6F20,222
2410 DATA 616C6C207573656420757 09B4241
4E47219B41636365707473207365637572697 4
792049420636172649B4974,470
2420 DATA 20676C6F77739B5365652070686F
746F20696E20414E414C4F472023313 89B4163
65206F662053706164657321,307
2430 DATA 9B4561736965722073616 9642074
68616E20646F6E659B4F6E6C7920312062756C
6C65749B4E6F2062756C6C65,667
2440 DATA 74739B58203D9B59203D9B464154
484F4D539B4E6F7468696E672068617070656E
739B537562206C6576656C73,57
2450 DATA 206F66669B5375622064697665 73
9B4E6F6E659B4120636C6F7564206F6620706F
69736F6E6F7573206761739B,556
2460 DATA 6B696C6C7320796F7520696E7374
616E746C79219B4C6F636B2069732076657279
207365637572659B416C7265,51
2470 DATA 616479206F70656E9B796F752061
726520444541449B456E656D79206361707475
726573207468652073756220,300
2480 DATA 616E649B49742066616C6C732064
6F776E2074686520706970659B4C6F636B2064
657374726F796564219B5472,745
2490 DATA 792065786 16D696E696E67207468
696E67739B536372657764726976657227732 0
746F6F2074696E799B4B6579,326
2500 DATA 20776F6E2774206669749B41206A
6F6C74206F662068696768207 66F6C74616765
9B54726169746F722073686F,638
2510 DATA 6F747320796F7520616E649B5375
622068697473206 26F74746F6D219B4120626C
617374206F662072616496F,941
2520 DATA 6163746976697479 9B426F6C7473
20776F6E2774206C657420796F759B436F6E67
726174756C6174696F6E7321,539
2530 DATA 9B536176520676 16D6520746F20
C469736B206F7220C361737365747465 3F9B43
6F6E6E65637465642074 6F20,18
2540 DATA 6361626C659B0000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000,676
```

•

CHECKSUM DATA

(See p. 30)

```
10 DATA 205,351,496,811,423,729,200,60
3,555,573,694,613,29,205,197,6684
160 DATA 749,198,962,93,491,30,155,941
,287,88,522,600,216,516,706,6554
1060 DATA 900,534,157,861,15,983,935,8
81,220,633,824,685,180,903,116,8827
1210 DATA 944,700,783,728,657,542,707,
886,830,993,213,263,281,961,898,10386
1360 DATA 420,104,44,85,96,544,679,976
,333,222,350,953,865,898,28,6597
1510 DATA 911,24,703,994,156,61,875,18
2,961,52,490,763,912,733,907,8724
1660 DATA 951,36,677,622,150,85,45,928
,938,343,792,708,780,219,815,8089
1810 DATA 905,315,845,974,48,770,505,4
86,805,645,202,458,956,165,618,8697
1960 DATA 134,577,302,470,410,679,737,
552,510,779,876,285,582,825,549,8267
2110 DATA 598,676,767,688,464,638,859,
679,393,228,920,433,555,654,755,9307
2260 DATA 384,174,307,915,532,571,503,
510,508,671,692,543,607,411,645,7973
2410 DATA 278,309,691,821,709,697,376,
584,566,567,528,718,673,490,8007
```

•

(Assembly language listing starts next page.)

```
; ------------------------
; CRASH DIVE! (TM)
; by Brian Moriarty
; ANALOG Computing #18
; (C)1984 ANALOG Computing
; ------------------------
;
; MACRO DEFINITIONS
; -----------------
;
;         POSITION MACRO
;         --------------
;
; SYNTAX:
; POSITION xpos,ypos
;
        .MACRO POSITION
        .IF %0<>2 .OR %1>39 .OR %2>23
          .ERROR "POS parameters"
          .ELSE
          .IF %1=13
            LDY #%2
            JSR POSIT13
            .ELSE
            LDX #%1
            LDY #%2
            JSR POSIT
            .ENDIF
          .ENDIF
        .ENDM
;
;         PRINT MACRO
;         -----------
;
; SYNTAX:
; PRINT <addr of EOL-terminated string>
;
        .MACRO PRINT
        .IF %0<>1
          .ERROR "PRINTE parameters"
          .ELSE
          LDA # <%1
          LDY # >%1
          JSR EPRINT
          .ENDIF
        .ENDM
;
;         TEXT MACRO
;         ----------
;
; SYNTAX:
; TEXT <"string">
;
        .MACRO TEXT
        .IF %0<>1 .OR %1>127
          .ERROR "TEXT parameters"
          .ELSE
          .BYTE %*1,EOL
          .ENDIF
        .ENDM
;
; SYSTEM  EQUATES
; ---------------
;
;         ZERO-PAGE
;         ---------
;
BOOT?   =   $09     ; OS boot flag
POKMSK  =   $10     ; interrupt mask
RTCLOK  =   $14     ; system clock
LMARGN  =   $52     ; left margin
RMARGN  =   $53     ; right margin
ROWCRS  =   $54     ; cursor row
COLCRS  =   $55     ; cursor column
RAMTOP  =   $6A     ; # pages of RAM
FR0     =   $D4     ; floating point register
CIX     =   $F2     ; FP index register
INBUFF  =   $F3     ; FP pointer
;
;         PAGES 2-3
;         ---------
;
VDSLST  =   $0200   ; DLI vector
SRTIMR  =   $022B   ; key repeat timer
SDMCTL  =   $022F   ; DMA control
SDLSTL  =   $0230   ; D-list addr
COLDST  =   $0244   ; coldstart flag
GPRIOR  =   $026F   ; PM0 priority
PCOLR0  =   $02C0   ; player 0 color
PCOLR1  =   $02C1   ; player 1 color
PCOLR2  =   $02C2   ; player 2 color
PCOLR3  =   $02C3   ; player 3 color
COLOR0  =   $02C4   ; playfield 0 color
COLOR1  =   $02C5   ; playfield 1 color
COLOR2  =   $02C6   ; playfield 2 color
COLOR4  =   $02C8   ; background color
CRSINH  =   $02F0   ; cursor inhibit
CH      =   $02FC   ; keypress register
ICCOM   =   $0342   ; CIO command
ICBADR  =   $0344   ; CIO addr
ICBLEN  =   $0348   ; CIO length
ICAUX1  =   $034A   ; AUX byte 1
ICAUX2  =   $034B   ; AUX byte 2
;
;         CTIA/GTIA
;         ---------
;
HPOSP0  =   $D000   ; h-pos player 0
HPOSP1  =   $D001   ; "          " 1
HPOSP2  =   $D002   ; "          " 2
HPOSP3  =   $D003   ; "          " 3
SIZEP0  =   $D004   ; width player 0
SIZEP1  =   $D009   ; "           " 1
SIZEP2  =   $D00A   ; "           " 2
SIZEP3  =   $D00B   ; "           " 3
GRAFP0  =   $D00D   ; graphics player 0
GRAFP1  =   $D00E   ; "              " 1
GRAFP2  =   $D00F   ; "              " 2
GRAFP3  =   $D010   ; "              " 3
COLPF2  =   $D018   ; color register 2
CONSOL  =   $D01F   ; console keys
;
;         POKEY
;         -----
;
AUDF1   =   $D200   ; frequency channel 1
AUDC1   =   $D201   ; vol/dist channel 1
AUDCTL  =   $D208   ; audio control
RANDOM  =   $D20A   ; random # generator
IRQEN   =   $D20E   ; interrupt enable
;
;         ANTIC
;         -----
;
DMACTL  =   $D400   ; DMA control
WSYNC   =   $D40A   ; wait for horz sync
NMIEN   =   $D40E   ; NMI enable
;
;         FLOATING POINT
;         --------------
;
FASC    =   $D8E6   ; FP-to-ATASCII
IFP     =   $D9AA   ; integer-to-FP
ZFR0    =   $DA44   ; clear FR0
;
;         OS ROUTINES
;         -----------
;
CIOV    =   $E456   ; CIO entry
SETVBV  =   $E45C   ; set v-blank vector
SYSVBV  =   $E45F   ; OS VBI entry
SIOINV  =   $E463   ; SIO init
;
; INTERNAL EQUATES
; ----------------
;
;         MEMORY ALLOCATION
;         -----------------
;
INLINE  =   $0600   ; text input buffer
GAMEDATA =  $3AC0   ; working area
EDLIST  =   $3C20   ; start of E: display list
SCREEN  =   $3C40   ; start of screen RAM
;
; Game play database
;
        *=  GAMEDATA
;
EVENT   *=  *+2     ; event #
PLACE   *=  *+1     ; location
SLAT    *=  *+1     ; ship's latitude
SLON    *=  *+1     ; ship's longitude
MLAT    *=  *+1     ; missile latitude
MLON    *=  *+1     ; missile longitude
DEPTH   *=  *+1     ; sub's depth
GREEN   *=  *+1     ; green button on/off
RED     *=  *+1     ; red button on/off
SWITCH  *=  *+1     ; arming switch on/off
BREATH  *=  *+1     ; holding breath?
HATCH   *=  *+1     ; hatch opened?
UNIT    *=  *+1     ; unit dropped in rm 17?
CRASHED *=  *+1     ; sub crashed?
BULLET  *=  *+1     ; bullet used?
SUIT    *=  *+1     ; suit searched?
CAPTAIN *=  *+1     ; captain searched?
BHOLD   *=  *+1     ; breath holding counter
TRAITOR *=  *+1     ; traitor wait counter
HOLDING8S *= *+6    ; current inventory
VECTORS *=  *+132   ; current vector table
OBJECTS *=  *+132   ; current object table
NTRANS  *=  *+38    ; current translation matrix
;
;         MISCELLANEOUS
;         -------------
;
EOL     =   $9B
SPACE   =   $20
;
;         ZERO-PAGE
;         ---------
;
        *=  $80
;
; Cursor control
;
CURSEN  *=  *+1     ; cursor on/off flag
CSHAPE  *=  *+1     ; current cursor shape
BLINK   *=  *+1     ; cursor blink timer
;
; Keyboard handler
;
CLICK   *=  *+1     ; key click counter
LENGTH  *=  *+1     ; line length register
;
; DLI control
;
DLICOL  *=  *+1     ; DLI color index
;
; Screen clearing
;
CLPOINT *=  *+2     ; screen clear pointer
CINDEX  *=  *+1     ; window clear index
;
; Parser
;
PBUFF   *=  *+3     ; parsing buffer
PDEX    *=  *+1     ; scanning index
LBREAK  *=  *+1     ; pos. of space char
VCODE   *=  *+1     ; verb code #
UCODE   *=  *+1     ; untranslated noun code #
NCODE   *=  *+1     ; translated noun code #
DOVECT  *=  *+2     ; verb execution vector
CURVECT *=  *+6     ; current room vectors
CUROBJS *=  *+6     ; current room objects
NEWPLACE *= *+1     ; next room code #
VPNT    *=  *+2     ; addr of current v-buffer
OPNT    *=  *+2     ; addr of current o-buffer
INVPOS  *=  *+1     ; position in inventory buffer
ROOMPOS *=  *+1     ; position in room buffer
ANY?    *=  *+1     ; flag for empty room/inv
LATSHOW *=  *+1     ; latitude to display
LONSHOW *=  *+1     ; longitude to display
VLAST   *=  *+1     ; last verb
ULAST   *=  *+1     ; last noun
;
; Event counter
;
EBUFF   *=  *+5     ; printing buffer
;
;         INITIALIZATION
;         --------------
;
        *=  $1F80
;
ENTRY
        LDX #0
        STX COLDST
        INX         ; = 1
        STX BOOT?
        JSR SIOINV  ; for sound init
        JMP TITLE
;
;         CUSTOM DISPLAY LIST
;         -------------------
;
DLIST
        .BYTE $70,$70,$70,$70 ; blank 32 lines
        .BYTE $42, <SCREEN, >SCREEN ; text w/LMS
        .BYTE $90,$10 ; blank 4 w/DLI
        .BYTE $02     ; text (location)
        .BYTE $90,$10 ; blank 4 w/DLI
        .BYTE $02     ; text (exits)
        .BYTE $90,$10 ; blank 4 w/DLI
        .BYTE $02,$02,$02,$02,$02,$02 ; text (v-items)
        .BYTE $90,$10 ; blank 4 w/DLI
        .BYTE $02,$02 ; text (events)
        .BYTE $90,$10 ; blank 4 w/DLI
        .BYTE $02,$02 ; text (response)
        .BYTE $80,$80,$10 ; blank 4 w/DLIs
        .BYTE $02,$02,$02,$02,$02,$02 ; text (i-items)
        .BYTE $70     ; blank 8
        .BYTE $41, <DLIST, >DLIST ; JVB
;
;         TITLE SCREEN
;         ------------
;
TITLE
        LDX #$FF
        TXS
        JSR NEWSCREEN ; reset E:
          POSITION  12,6
          PRINT   T0  ; "Brian Moriarty's"
          POSITION  25,7
          PRINT   T1  ; "CRASH DIVE"
          POSITION  8,9
          PRINT   T2  ; "(C)1984 ANALOG Computing"
;
STARTOP
          POSITION  5,16
          PRINT   T3  ; "Press  START  to play new game"
          POSITION  3,18
          PRINT   T4  ; "Press  OPTION  to restore old game"
;
        LDA #6
        STA EDLIST+12 ; modify
        STA EDLIST+13 ; display list
;
        LDA #$22
        STA SDMCTL  ; restore screen
        JSR BEEP
;
; Wait for selection
;
POLL
        LDA CONSOL
        CMP #7      ; scan until a
        BEQ POLL    ; key is pressed
LET80
        LDX CONSOL
        CPX #7
        BNE LET80
        CMP #6      ; START pressed?
        BEQ NEWGAME ; yes, do a new game
        CMP #3      ; OPTION pressed?
        BEQ RESTORE ; yes, restore old game
BADPOLL
        JSR BOOP    ; else
        JMP POLL    ; resume scan
;
; Start new game
;
```

```
NEWGAME
        JSR INITDATA
        JMP PLAYSCREEN
;
; Restore a previous game
;
RESTORE
         POSITION  5,20
         PRINT   T5  ; "Restore from Disk or Tape?"
        JSR BEEP
DORT
        JSR GETKEY
        CMP #'D     ; disk?
        BEQ GETDISK
        CMP #'C     ; tape?
        BEQ GETAPE
DORTERR
        JSR CLOSE1
        JSR BOOP
        JMP DORT
;
; Get game from disk
;
GETDISK
        JSR DPOINT
        JMP READOPEN
;
; Get game from tape
;
GETAPE
        JSR TPOINT
;
READOPEN
        LDA #3      ; OPEN command
        STA ICCOM,X
        LDA #4      ; READ
        STA ICAUX1,X
        LDA #0
        STA ICAUX2,X
        JSR CIOV
        BMI DORTERR
;
; Get game data thru IOCB #1
;
GETDATA
        LDX #$10
        LDA # <GAMEDATA
        STA ICBADR,X
        LDA # >GAMEDATA
        STA ICBADR+1,X
        LDA #$48
        STA ICBLEN,X
        LDA #$01
        STA ICBLEN+1,X
        LDA #7      ; GET RECORD
        STA ICCOM,X
        JSR CIOV
        BMI DORTERR
        JSR CLOSE1
;
;         PLAYSCREEN INIT
;         ---------------
PLAYSCREEN
        JSR NEWSCREEN ; reset E:
;
; Enable custom display list
;
        LDA # <DLIST
        STA SDLSTL
        LDA # >DLIST
        STA SDLSTL+1
;
; Print title & labels
;
         PRINT  T6  ; "CRASH DIVE! (TM) EVENT #"
        LDA #12     ; set right margin
        STA RMARGN  ; for wraparound
         PRINT  T7  ; "LOCATION/EXITS"
         PRINT  T8  ; "VISIBLE ITEMS"
        JSR BAR
        JSR BAR
        JSR BAR
        JSR BAR     ; 4 blank bars
        JSR SAYWHAT ; "WHAT"
         PRINT  T11 ; "HAPPENS/YOUR RESPONSE"
        JSR SAYWHAT ; "WHAT"
         PRINT  T12 ; "YOU ARE CARRYING"
        JSR BAR
        JSR BAR     ; more blank bars
        LDA #39
        STA RMARGN  ; reset
;
        LDA #$C4    ; green
        STA PCOLR0  ; cursor
;
; Setup P/M borders & mask
;
        LDA #48
        STA HPOSP1  ; position left
        LDA #202    ; and
        STA HPOSP2  ; right borders
        LDA #64     ; and
        STA HPOSP3  ; title cover
        LDX #255
        STX GRAFP1  ; set up
        STX GRAFP2  ; side borders and
        STX GRAFP3  ; title mask
        INX         ; = 0
        STX SIZEP0  ; set cursor width
        INX         ; = 1
        STX GPRIOR  ; set player priority
        LDA #3
        STA SIZEP1  ; set border
        STA SIZEP2  ; and mask
        STA SIZEP3  ; widths
;
        LDA #$F0    ; init
        STA CSHAPE  ; cursor shape
        LDA #30     ; and
        STA BLINK   ; blink timer
;
        LDA #13
        STA LMARGN  ; init left margin
;
        LDY # <IMMVBI ; set VBI vector,
        LDX # >IMMVBI ; specifying
        LDA #6      ; immediate-mode
        JSR SETVBV
;
        LDA # <DLI
        STA VDSLST  ; tell OS where
        LDA # >DLI  ; we put our
        STA VDSLST+1 ; DLI service routine
        LDA #$C0    ; set bits 6 & 7 of NMIEN
        STA NMIEN   ; to enable DLIs and VBIs
;
        LDA PLACE
        JMP REENTRY
;
;         EVENT GENERATOR
;         ---------------
;
NEXTEVENT
        LDX #$FF    ; empty stack
        TXS         ; just in case!
        JSR SHOWEV  ; show event #
        INC EVENT   ; update
        BNE RADIO   ; event
        INC EVENT+1 ; counter
;
; Handle room 0
;
RADIO
        LDA PLACE
        BNE ENEMY
        LDA #32     ; got suit?
        JSR OWNIT?
        BEQ ENEMY
        JSR NEWSCREEN
         POSITION  8,3
         PRINT  T79 ; "A blast of radioactivity"
        JMP KILLS
;
; Check for enemy takeover
;
ENEMY
        LDA EVENT
        CMP #32
        BNE LUNGS
        LDA EVENT+1
        BNE LUNGS
        LDA DEPTH
        BNE LUNGS
        JSR NEWSCREEN
         POSITION  7,3
         PRINT  T67 ; "Enemy captures the sub and"
        JMP KILLS
;
; Limit breath-holding
;
LUNGS
        LDA BREATH  ; holding it?
        BEQ POISON  ; no - skip this
        DEC BHOLD   ; else decrement count
        BNE POISON  ; continue if NZ
        LDA #0      ; else
        STA BREATH  ; release hold
;
; Handle poison atmosphere
;
POISON
        LDA HATCH   ; hatch opened?
        BEQ DOTRAIT ; not yet
        LDA #28     ; got the
        JSR OWNIT?  ; mask?
        BEQ DOTRAIT ; yup - you're safe
        LDA BREATH  ; holding breath?
        BNE DOTRAIT ; good thing, otherwise ...
        JSR NEWSCREEN
         POSITION  8,3
         PRINT  T62 ; "A cloud of poisonous gas"
        JMP KILLS
;
; Handle traitor
;
DOTRAIT
        LDA TRAITOR
        BPL NTRAIT
        JSR NEWSCREEN
         POSITION  9,3
         PRINT  T77 ; "Traitor shoots you and"
        JMP KILLS
NTRAIT
        LDA PLACE   ; is this
        CMP #18     ; room 18?
        BNE TRESET  ; we're safe
        LDA #4      ; is the traitor
        JSR INROOM? ; lurking?
        BNE TRESET  ; thankfully not
        DEC TRAITOR ; else reduce wait time
        JMP DOCRASH
TRESET
        LDA #1
        STA TRAITOR
;
; Change sub coords, handle crash
;
DOCRASH
        LDA CRASHED
        BNE PARSER
        LDA RANDOM
        AND #$F8
        STA SLAT
        LDA RANDOM
        AND #$F8
        STA SLON
        LDA RED
        BEQ PARSER
        LDA DEPTH
        CLC
        CLD
        ADC #8
        STA DEPTH
        BPL PARSER
        STA CRASHED
        JSR CLWH
         PRINT  T45 ; "BANG!"
         PRINT  T78 ; "Sub hits bottom!"
;
;         INPUT PARSER
;         ------------
;
PARSER
        LDA #$22
        STA SDMCTL
        JSR GETLINE ; put line into INLINE
        JSR CLWH
        LDX LENGTH
        CPX #1      ; if length is 1
        BNE DOCLAUSE ; check for legality
;
; Check for a legal single-char command
;
        LDA INLINE  ; get the character
        LDX #8      ; init search index
LEGSING
        CMP SCOMS,X
        BEQ EXSING  ; matched! go do it
        DEX         ; otherwise
        BPL LEGSING ; keep searching
        JSR SYNERR  ; error, so print
         PRINT  T14 ; "Invalid command"
        JMP BADPARSE ; and try again
;
; Execute a single-char command
;
EXSING
        LDA SVECTL,X ; fetch the lsb
        STA DOVECT  ; and
        LDA SVECTH,X ; msb of the
        STA DOVECT+1 ; execution addr
        JMP (DOVECT) ; and do it!
;
; Find the 1st space character
; in the user's response
;
DOCLAUSE
        LDA #SPACE
        STA PBUFF+1
        STA PBUFF+2
        LDX #1
FIND1
        LDA INLINE,X ; length is in X
        CMP #SPACE  ; is it a space?
        BEQ ENDV    ; yes!
        INX         ; else keep scanning
        CPX LENGTH
        BCC FIND1
BADVERB
        JSR SYNERR  ; verb is no good, so print
         PRINT  T15 ; "Verb not recognized"
        JMP BADPARSE ; and try again
;
; Space char found, so record its
; position and move the first half
; of the clause into the parsing buffer
;
ENDV
        STX LBREAK
        CPX #2
        BCC TOVB
        LDX #2
TOVB
```

```
        LDA INLINE,X
        STA PBUFF,X
        DEX
        BPL TOVB
;
; Check for a legal verb
;
        LDX #0      ; init verb index
        STX VCODE
VNEXT
        STX PDEX
        LDY #0      ; init buffer char index
VSCAN
        LDA PBUFF,Y ; get a char from buffer
        CMP VERBS,X ; match?
        BNE NEXTRY  ; nope - try another verb
        INX
        INY
        CPY #3
        BCC VSCAN   ; if all 3 chars match
        BCS LEGALV  ; the verb is legal
NEXTRY
        INC VCODE
        LDX PDEX
        INX
        INX
        INX
        CPX #NV*3+3 ; out of verbs?
        BCC VNEXT   ; nope - keep scanning
        BCS BADVERB ; else verb is worthless
;
; Verb is legal, so fetch its execution
; vector
;
LEGALV
        LDA VCODE   ; fetch verb #
        CMP #28     ; if it's 80,
        BNE MOVE2   ; warn user:
         PRINT  T29 ; "Type N S E W U or D"
        JMP BADPARSE
;
; Move the second half of the
; clause into the parsing buffer
;
MOVE2
        LDX LBREAK  ; fetch pos. of space char
        INX         ; plus 1
        LDY #0      ; init buffer char index
MOVEN
        LDA INLINE,X ; fetch character
        STA PBUFF,Y ; stuff into buffer
        INX
        INY
        CPY #3      ; until 3 characters
        BCC MOVEN   ; have been moved
;
; Check for a legal noun
;
        LDX #0      ; init noun index
        STX UCODE
NNEXT
        STX PDEX
        LDY #0      ; init buffer char index
NSCAN
        LDA PBUFF,Y ; get a char from buffer
        CMP NOUNS,X ; match?
        BNE NEXTRY2 ; nope - try another noun
        INX
        INY
        CPY #3
        BCC NSCAN   ; if all 3 chars match
        BCS LEGALN  ; the noun is legal
NEXTRY2
        INC UCODE
        LDX PDEX
        INX
        INX
        INX
        CPX #NN*3+3 ; out of nouns?
        BCC NNEXT   ; nope - keep scanning
        JSR SYNERR  ; else noun is garbage
         PRINT  T16 ; "Noun not in vocabulary"
        JMP BADPARSE
;
; Noun's code # is in UCODE;
; verb's code # is in VCODE;
; verb execution addr is in DOVECT
;
LEGALN
        LDA VCODE   ; fetch
        STA VLAST
        ASL A       ; execution addr
        TAX         ; and
        LDA VVECTS,X ; save it
        STA DOVECT  ; in DOVECT
        INX
        LDA VVECTS,X
        STA DOVECT+1
        LDA UCODE
        STA ULAST
        CMP #39     ; was it BUTTON?
        BEQ DOBUTT
        CMP #40     ; was it INVENTORY?
        BEQ DOINVE
        TAX
        LDA NTRANS,X ; translate noun
        STA NCODE
        JMP (DOVECT) ; execute verb
;
; Handle BUTTON
;
DOBUTT
         PRINT  T30 ; "Refer to it by color"
        JMP BADPARSE
;
; Handle INVENTORY
;
DOINVE
         PRINT  T31 ; "Type I for inventory"
        JMP BADPARSE
;
;        EXECUTE SINGLE-CHAR COMMANDS
;        ----------------------------
;
;        COMMAND VECTOR TABLES
;        ---------------------
;
SVECTL
        .BYTE  <DOM, <DOM, <DOM, <DOM, <DOM
        .BYTE  <DOM, <DOQ, <DOX, <DOA
SVECTH
        .BYTE  >DOM, >DOM, >DOM, >DOM, >DOM
        .BYTE  >DOM, >DOQ, >DOX, >DOA
;
;        HANDLE "Q" (QUIT)
;        -----------------
;
DOQ
         POSITION  13,12
         PRINT  T19 ; "Type Y to quit game:"
         POSITION  34,12
        STX CURSEN  ; enable cursor
        JSR BEEP
        JSR GETKEY
        CMP #'Y
        BEQ DOQUIT
        LDX #12
        JSR ERASE
        JMP BADPARSE
DOQUIT
        JMP TITLE
;
;        HANDLE MOVEMENT
;        ---------------
;
; ENTRY: Vector (0-5) in X
;
DOM
        LDA CURVECT,X
        BPL EXMOVE
CANTGO
         PRINT  T18 ; "You can't go that way."
        JMP BADPARSE
EXMOVE
        CLD         ; for safety
        STA NEWPLACE ; save destination
        JSR SAVELOC ; save status
        LDA NEWPLACE ; get destination,
        STA PLACE   ; make it current, and
REENTRY
        JSR BPOINT  ; point to the new buffers
;
; Get new buffer data
;
        LDY #5
RLOOP
        LDA (VPNT),Y
        STA CURVECT,Y
        LDA (OPNT),Y
        STA CUROBJS,Y
        DEY
        BPL RLOOP
;
; Refresh screen
;
SHOWPLACE
        LDX #1
        JSR ERASE   ; clear location window
         POSITION  13,1
        LDX PLACE   ; get loc #
        LDA RDLS,X  ; fetch lsb and
        LDY RDHS,X  ; msb of text addr and
        JSR EPRINT  ; print it
;
        JSR SHOWVIS ; display visible items
        JSR SHOWVECTS ; display new vectors
        JSR SHOWINV ; show inventory
        JMP POKAY   ; congratulations!
;
;        POINT TO NEW BUFFERS
;        --------------------
;
; ENTRY: Buffer # (0-23) in A
;
BPOINT
        ASL A       ; * 2
        STA NCODE   ; save it
        ASL A       ; * 4
        CLC
        ADC NCODE   ; *2 + *4 = *6
        STA NCODE   ; save it
        CLC
        ADC # <VECTORS
        STA VPNT
        LDA # >VECTORS
        ADC #0
        STA VPNT+1
        CLC
        LDA NCODE
        ADC # <OBJECTS
        STA OPNT
        LDA # >OBJECTS
        ADC #0
        STA OPNT+1
        RTS
;
;        HANDLE "X" (SAVE GAME)
;        ----------------------
;
DOX
        JSR SAVELOC ; save current status
        JSR NEWSCREEN
         POSITION  5,11
         PRINT  T82 ; "Save game to Disk or Cassette?"
        LDA #$22
        STA SDMCTL
        JSR BEEP
SAVEPOLL
        JSR GETKEY
        CMP #'D
        BEQ DSAVE
        CMP #'C
        BEQ CSAVE
BADWRITE
        JSR CLOSE1
        JSR BOOP
        JMP SAVEPOLL
;
; Save to disk
;
DSAVE
        JSR DPOINT
        JMP GSAVE
;
; Save to cassette
;
CSAVE
        JSR TPOINT
;
GSAVE
        LDA #3
        STA ICCOM,X
        LDA #8
        STA ICAUX1,X
        LDA #0
        STA ICAUX2,X
        JSR CIOV
        BMI BADWRITE
;
; Write out game data
;
WRITE
        LDX #$10
        LDA # <GAMEDATA
        STA ICBADR,X
        LDA # >GAMEDATA
        STA ICBADR+1,X
        LDA #$48
        STA ICBLEN,X
        LDA #$01
        STA ICBLEN+1,X
        LDA #11
        STA ICCOM,X
        JSR CIOV
        BMI BADWRITE
        JSR CLOSE1
        JMP PLAYSCREEN
;
;        HANDLE "A" (AGAIN)
;        ------------------
;
DOA
        LDA VLAST   ; restore old verb
        STA VCODE
        LDA ULAST   ; and noun
        STA UCODE
        JMP LEGALN  ; and do it again!
;
;        SAVE LOC STATUS
;        ---------------
;
SAVELOC
        LDA PLACE
        JSR BPOINT
        LDY #5
SLOOP
        LDA CURVECT,Y
        STA (VPNT),Y
        LDA CUROBJS,Y
        STA (OPNT),Y
        DEY
        BPL SLOOP
        RTS
;
;        VERB EXECUTORS
;        --------------
;
; ENTRY: Translated noun code in A
;        and in NCODE;
;        untranslated code in UCODE
```

```
;
;         TAKE
;         ----
DOTAKE
        LDA UCODE
        CMP #22       ; is it moveable?
        BCS DT0       ; yes
        JMP IMPOSS
DT0
        LDA NCODE
        JSR OWNIT?    ; already have it?
        BNE DT1
        JMP ALREADY
DT1
        LDA NCODE
        JSR INROOM?   ; is it here?
        BEQ DT2
        JMP NOTHERE
DT2
        LDA NCODE
        CMP #20       ; bolted unit?
        BNE DT3
         PRINT  T41   ; "Bolts are tight & rusty"
        JMP GOODPARSE
DT3
        CMP #42       ; free unit?
        BNE DT3B
        LDA PLACE     ; room 0?
        BNE DT3B
        LDA #27       ; got wrench?
        JSR OWNIT?
        BEQ DT3A
         PRINT  T80   ; "Bolts won't let you"
        JMP GOODPARSE
DT3A
        LDA #7        ; power cable
        JSR INROOM?
        BNE DT3B
         PRINT  T83   ; "Connected to cable"
        JMP GOODPARSE
DT3B
        JSR INVSPACE? ; arms full?
        BEQ DT4
        JMP ARMSFULL
DT4
        LDX ROOMPOS   ; get object position
        LDY INVPOS    ; and inv position
        LDA CUROBJS,X ; pick it up
        STA HOLDINGS,Y ; add to inventory
        LDA #$FF
        STA CUROBJS,X ; leave a blank slot
SHOWALL
        JSR SHOWVIS   ; show room
        JSR SHOWINV   ; and inventory
        JMP POKAY     ; done!
;
;         DROP
;         ----
DODROP
        JSR OWNIT?    ; do you have it?
        BEQ DD0
        JMP DONTHAVE
DD0
        LDA PLACE
        CMP #17       ; is this room 17?
        BEQ DROP17    ; special handling
DD1
        JSR ROOMSPACE? ; enough room for it?
        BEQ DROPIT
        JMP ROOMFULL
DROPIT
        LDX ROOMPOS
        LDY INVPOS
        LDA HOLDINGS,Y
        STA CUROBJS,X
        LDA #$FF
        STA HOLDINGS,Y
        BNE SHOWALL
;
; Handle room 17
;
DROP17
        JSR ROOMIN18? ; space in room 18?
        BNE DD1       ; no - drop it in 17
        LDY INVPOS    ; get inv position
        LDX ROOMPOS   ; and pos in room 18
        LDA HOLDINGS,Y ; pick up item
        STA OBJECTS,X ; and put in 18
        LDA #$FF      ; clear inventory
        STA HOLDINGS,Y ; slot
         PRINT  T68   ; "It falls down the pipe"
        LDA NCODE
        CMP #42       ; dropped the unit?
        BEQ DROPUNIT  ; special handling
D17B
        JSR SHOWVIS
        JSR SHOWINV
        JMP GOODPARSE
;
; Handle UNIT in 17
;
DROPUNIT
        LDA UNIT      ; init = 0
        BNE D17B
        LDA #38       ; update traitor
        LDX #5        ; status in
        STA UNIT      ; unit flag,
        STA NTRANS,X  ; translation matrix
        LDX #108      ; and object
        STA OBJECTS,X ; matrix
        JSR ROOMIN18? ; find room
        LDX ROOMPOS   ; for pistol
        LDA #21       ; and
        STA OBJECTS,X ; drop it in 18
        BNE D17B
;
; Find empty slot in room 18
;
ROOMIN18?
        LDX #109      ; skip 1st object
RN18
        LDA OBJECTS,X
        BMI RN18A     ; found a blank!
        INX
        CPX #113      ; scan to end
        BCC RN18      ; of room
        TXA           ; return NZ status
        RTS
RN18A
        STX ROOMPOS   ; save room pos
        LDA #0        ; set zero status
        RTS
;
;         REMOVE
;         ------
DOREMOVE
        CMP #22       ; moveable?
        BCS DRM
        JMP IMPOSS
DRM
        CMP #28       ; mask?
        BEQ GODROP
        CMP #32       ; suit?
        BEQ GODROP
        CMP #42       ; unit?
        BEQ GOGRAB
        JMP BESPEC
GOGRAB
        JMP DOTAKE
GODROP
        JMP DODROP
;
;         LOOK/EXAM
;         ---------
DOLOOK
        JSR INROOM?   ; is it in room?
        BEQ LOOKOK    ; if not,
        LDA NCODE
        JSR OWNIT?    ; do you have it?
        BEQ LOOKOK
        JMP NOTHERE   ; guess not
LOOKOK
        LDA NCODE
        CMP #9        ; captain?
        BNE LK0
;
; Search captain
;
EXAMCAP
        LDA CAPTAIN
        BEQ EC0
        JMP SEEMSORD
EC0
        JSR ROOMSPACE?
        BEQ EC1
        JMP ROOMFULL
EC1
         PRINT  T35   ; "Found something!"
        LDA #24
        STA CAPTAIN
        LDX ROOMPOS
        STA CUROBJS,X
        JSR SHOWVIS
        JMP GOODPARSE
;
LK0
        CMP #15       ; gauge?
        BNE LK1
;
; Read depth gauge
;
READGAUGE
        JSR ZFR0
        LDA DEPTH
        STA FR0
        JSR VPRINT
         PRINT  T55   ; "Fathoms"
        JMP GOODPARSE
;
LK1
        CMP #17       ; display?
        BNE LK2
;
; Read navigation displays
;
READISP
        LDA PLACE
        CMP #11       ; missile room?
        BEQ SHOWMD
        LDA SLAT
        LDX SLON
        BNE DISHOW
SHOWMD
        LDA MLAT
        LDX MLON
DISHOW
        STA LATSHOW
        STX LONSHOW
         PRINT  T53   ; "X ="
         POSITION  17,9
        JSR ZFR0
        LDA LATSHOW
        STA FR0
        JSR VPRINT
         PRINT  T54   ; "Y ="
         POSITION  17,10
        JSR ZFR0
        LDA LONSHOW
        STA FR0
        JSR VPRINT
        BNE LKX
;
LK2
        CMP #32       ; suit?
        BNE LK3
;
; Examine suit
;
EXAMSUIT
        LDA SUIT
        BEQ ES0
        JMP SEEMSORD
ES0
        JSR ROOMSPACE?
        BEQ ES1
        JMP ROOMFULL  ; "Not enough room here."
ES1
         PRINT  T35   ; "Found something!"
        LDA #33       ; key
        STA SUIT
        LDX ROOMPOS
        STA CUROBJS,X
        JSR SHOWVIS
        JMP GOODPARSE
;
LK3
        CMP #21       ; pistol?
        BNE LK4
;
; Examine pistol
;
EXAMPIST
        LDA BULLET
        BNE PX0       ; init = 1
         PRINT  T52   ; "No bullets"
        BNE LKX
PX0
         PRINT  T51   ; "Only 1 bullet"
        BNE LKX
;
LK4
        ASL A         ; * 2
        TAX           ; use as an index
        LDA LKLK,X    ; fetch lsb
        INX           ; and
        LDY LKLK,X    ; msb of text addr
        JSR EPRINT    ; print text
LKX
        JMP GOODPARSE ; and exit
;
;         EXAM TEXT LOOKUP TABLE
;         ----------------------
LKLK
        .WORD T36,T64,T34,T38
        .WORD T39,T34,T64,T39
        .WORD T42,T34,T34,T34
        .WORD T34,T34,T39,T34
        .WORD T37,T34,T34,T46
        .WORD T41,T34,T34,T48
        .WORD T48,T34,T49,T34
        .WORD T34,T34,T48,T48
        .WORD T34,T34,T34,T34
        .WORD T37,T34,T34,T49
        .WORD T39,T34,T47
;
;         READ
;         ----
DOREAD
        LDX #9
RDLOOP
        CMP READS,X
        BEQ READOK
        DEX
        BPL RDLOOP
        JMP IMPOSS
READOK
        JMP DOLOOK
;
; READable nouns
;
READS
        .BYTE 2,36,8,15,17
        .BYTE 23,24,26,30,31
;
```

Tired of high prices, poor service and hidden charges you get from other mail order companies . . .

Try RISING SUN SOFTWARE!

ENTERTAINMENT

ARTWORX
Strip Poker (D) $25.00
S.P. Data Disks 20.00

ATARI (ROM only)
Centipede $32.00
Defender 32.00
Dig Dug 32.00
Donkey Kong 35.00
Eastern Front 35.00
Galaxian 32.00
Joust 35.00
Missle Command 27.00
Ms. Pac-Man 35.00
Pac-Man 32.00
Pengo 32.00
Qix 32.00
Robotron 32.00
Space Invaders 27.00
Star Raiders 32.00

BIG FIVE
Miner 2049er (R) $35.00

BRODERBUND
A.E. (D) $25.00
Arcade Machine (D) ... 42.00
Choplifter (D) 25.00
Choplifter (R) 32.00
Oper. Whirlwind (D) ... 25.00
Sea Fox (D) 21.00
Sea Fox (R) 28.00
Sky Blazer (D) 28.00

BUDGECO
Raster Blaster (D) $21.00

DATAMOST
Airstrike (D/C) $28.00
Bilestoad (D) 28.00
Mating Zone (D) 25.00

DATAMOST
Micropainter (D) $25.00
Pooyan (D/C) 21.00
Zaxxon (D/C) 28.00

DON'T ASK
Poker Sam (D/C) $20.00
S.A.M. (D) 42.00

EDU-WARE
Prisoner 2 (D) $28.00
Rendezvous (D) 28.00

ELECTRONIC ARTS
Archon (D) $28.00
Hard Hat Mack (D) 25.00
M.U.L.E. (D) 28.00
Murder on the Zunderneuf (D) 28.00
Pinball Construction Set (D) 28.00
Worms? (D) 25.00

EPYX
Jumpman (D) $28.00
Jumpman Jr. (D) 28.00

INFOCOM (disk only)
Deadline $35.00
Enchanter 35.00
Planet Fall 35.00
Starcross 28.00
Suspended 35.00
The Witness 35.00
Zork I 28.00
Zork II 28.00
Zork III 28.00

INTELLIGENT STATEMENTS
Pro. Blackjack (D) ... $49.00

LIGHTNING SOFTWARE
Master Type (D) $28.00

MUSE
Castle Wolfenstein ... $21.00

ODESTA
Chess (D) $49.00
Checkers (D) 35.00
Odin (D) 35.00

PARKER BROTHERS
Astrochase (R) $35.00
Chess (R) 42.00
Frogger (R) 34.00
Popeye (R) 35.00
Q-Bert (R) 35.00
Risk (R) 42.00
Super Cobra (R) 35.00
Tutankam (R) 35.00

ROKLAN
Deluxe Invaders (R) .. $28.00
Gorf (D) 28.00
Gorf (R) 32.00
Wizard of Wor (R) 32.00
Wizard of Wor (D) 28.00

SIERRA ON-LINE
Crossfire (R) $25.00
Crossfire (D/C) 21.00
Frogger (D/C) 25.00
Mission Asteroid (D) .. 20.00
Sammy Lightfoot (R) .. 27.00
Ultima II (D) 42.00
Ulysses (D) 28.00
Wiz & Princess (D) 25.00

HARDWARE DEALS

MODEMS
Apple Cat II .. $299.00
Micromodem II 265.00
Micromodem w/terminal prog. 295.00
212 Apple Cat . 580.00

MONITORS
Amdek
Color I $299.00
Color II RGB ... 599.00
Color III RGB .. 399.00
RGB Card 149.00

USI
Pi1 9" Green . $119.00
Pi2 12" Green . 149.00
Pi3 12" Amber. 159.00
Pi4 9" Amber .. 129.00
Color 1400 .... 299.00

PRINTERS
C. Itoh
GX-100 ..... $229.00
Prowriter 399.00
Okidata
Microline 80 ... 349.00
Microline 82A . 449.00
Microline 92 ... 549.00

STRATEGIC SIMULATIONS
Battle for Normandie (D/C) .. $28.00
Battle of Shiloh (D/C) 28.00
Combat Leader (D) 28.00
Cosmic Balance (D/C) 28.00
Cosmic Balance II (D/C) 28.00
Cytron Masters (D) 28.00
Galactic Gladiator (D) . 28.00
Knights of the Desert (D/C0 ..... 28.00
Shattered Alliance (D) . 28.00
Tigers in the Snow (D/C) 28.00

SYNAPSE
Blue Max (D/C) $25.00
Dimension X (D/C) 25.00
Ft. Apocalypse (D/C) .. 25.00
Necromancer (D/C) ... 25.00
Pharoah's Curse (D/C) 25.00
Shadow World (D/C) .. 25.00
Shamus (D/C) 25.00
Shamus (R) 32.00
Shamus II (D/C) 25.00
Survivor (D/C) 25.00
Zepplin (D/C) 25.00

THORN EMI
Hockey (R) $28.00
Jumbo Jet Pilot (R) 35.00
River Rescue (R) 28.00
Soccer (R) 35.00
Submarine Commander (R) 35.00

UTILITIES & LANGUAGES

ADVENTURE INTERNATIONAL
Diskey (D) $35.00

ATARI
Assembler Editor (R) . $45.00
Atari Basic (R) 42.00
Macro Assembler (R) .. 68.00
Microsoft Basic II (R) .. 68.00
PILOT (R) 60.00

DATASOFT
BASIC Compiler (D) .. $63.00
Lisp Interpreter (D) ... 70.00
Edit 6502 (R) 125.00

OPTIMIZED SYSTEMS
Basic A+ (D) $56.00
Bug 65 (D) 25.00
Mac/65 (D) 56.00

EDUCATION

ATARI
Conversational Languages
French (C) $42.00
German (C) 42.00
Italian (C) 42.00
Spanish (C) 42.00
Educator Kit 117.00
Inv. to Programming 1 ..... 18.00
Inv. to Programming 2 ..... 23.00
Inv. to Programming 3 ..... 23.00
Juggles House (D) 23.00
Juggles House (C) 18.00
Juggles Rainbow (D) .. 23.00
Juggles Rainbow (C) .. 23.00
My First Alphabet (D) . 27.00
Programmer Kit 53.00
Tough Typing (cass) .. 18.00

EDU-WARE
Compu-Read (D) $21.00
Compu-Read (C) 15.00

SPINNAKER
Face Maker (D) $28.00
Hey Diddle Diddle (D) . 21.00
Kindercomp (D) 21.00
Most Amazing Thing (D) 28.00
Rhymes & Riddles (D) . 21.00
Snooper Troops 1 (D) . 32.00
Snooper Troops 2 (D) . 32.00

RISING SUN SOFTWARE
4200 PARK BLVD.
OAKLAND, CALIFORNIA 94602
(415) 482-3391

Ordering Information: We'll accept any form of payment—cash, personal check, money order, VISA/MasterCard, or C.O.D. Send cash at your own risk. Add $2.00 for UPS shipping; $3.00 for Blue Label Air. California residents add applicable sales tax. ALL orders shipped same day received. If we are out of stock on a particular item we will include a special bonus with your order when shipped.

master charge

CALL TOLL FREE 24 HOURS (ORDERS ONLY)
(800) 227-1617, ext. 114 (Outside California)
(800) 772-3545, ext. 114 (Inside California)

VISA

CIRCLE #129 ON READER SERVICE CARD.

```
;          PUSH
;          ----
DOPUSH
          JSR OWNIT?
          BNE DPH
          JMP WHYBOTH
DPH
          LDA NCODE
          JSR INROOM?
          BEQ DPH0
          JMP NOTHERE
DPH0
          LDA NCODE
          CMP #10       ; green button?
          BNE DPH1
;
; Handle green button push
;
          LDX #3
          LDA GREEN
          BEQ SCANON
          LDA #0
          STA GREEN
          LDA #2        ; blank scanner
SCANNER
          STA NTRANS,X
          STA CUROBJS
          JSR SHOWVIS
          JMP GOODPARSE
SCANON
          LDA OBJECTS+1 ; check cable
          CMP #7
          BEQ ONGREEN
          JMP NOTHAP    ; "Nothing happens"
ONGREEN
          STX GREEN
          LDA #36       ; active scanner
          BNE SCANNER
;
DPH1
          CMP #11       ; red button?
          BNE DPH2
;
; Handle red button push
;
          LDX RED
          BEQ REDON
          LDA #0
          STA RED
           PRINT  T59 ; "Sub levels off"
          BNE REDX
REDON
          INX           ; = 1
          STX RED
           PRINT  T60 ; "Sub dives!"
REDX
          JMP GOODPARSE
;
DPH2
          CMP #12       ; gold button?
          BNE DPH3
;
; Handle gold button
;
          LDA MLAT
          CLD
          CLC
          ADC #8
          STA MLAT
          JMP SHOWMD
;
DPH3
          CMP #13       ; silver button?
          BNE DPH4
;
; Handle silver button
;
          LDA MLON
          SEC
          CLD
          SBC #8
          STA MLON
          JMP SHOWMD
;
DPH4
          CMP #14       ; white button?
          BNE SORRY
;
; Handle white button
;
PUSHWHITE
          LDA MLAT      ; missile =
          CMP SLAT      ; sub?
          BNE SORRY
          LDA MLON      ; missile =
          CMP SLON      ; sub?
          BNE SORRY
          LDA SWITCH    ; missile armed?
          BEQ SORRY
          JSR NEWSCREEN
          LDA #14
          STA COLOR2
          STA COLOR4
          LDA #0
          STA COLOR1
           POSITION  12,11
           PRINT   T81 ; "Congratulations!"
          LDA ##22
          STA SDMCTL
SOLVED
          JMP SOLVED
SORRY
          JMP NOTHAP    ; "Nothing happens."
;
;          OPEN
;          ----
DOOPEN
          CMP #6        ; can't be OPENed
          BCC OP0       ; if NCODE > 6
          JMP IMPOSS
OP0
          JSR INROOM?   ; is it here?
          BEQ OP1
          JMP NOTHERE   ; nope
OP1
          LDA NCODE     ; hatch?
          BNE OP2
HOPEN
          LDA #34
          STA HATCH     ; mark hatch as opened
          STA CUROBJS   ; change in current objects
          LDA #4        ; open path to
          STA CURVECT+5 ; room 4
          JSR SHOWVIS
          JSR SHOWVECTS
          JMP POKAY
OP2
          CMP #1        ; door?
          BNE OP3
           PRINT  T64 ; "Lock is very secure"
          BNE OPX
OP3
          CMP #3        ; grate?
          BNE OP4
           PRINT  T38 ; "Screwed in place"
          BNE OPX
OP4
          CMP #5        ; airlock?
          BNE OP5
           PRINT  T23 ; "Can't do that yet"
          BNE OPX
OP5
          CMP #34       ; open hatch?
          BEQ ALOPEN
          CMP #35       ; open door?
          BEQ ALOPEN
          CMP #37       ; open grate?
          BEQ ALOPEN
          CMP #39       ; open airlock?
          BNE OP6
ALOPEN
           PRINT  T65 ; "Already open!"
OPX
          JMP GOODPARSE
OP6
          JMP IMPOSS
;
;          SHOOT
;          -----
DOSHOOT
          LDA #21       ; do you have
          JSR OWNIT?    ; the pistol?
          BEQ SHT0
          JMP EASIER
SHT0
          LDA NCODE
          JSR INROOM?
          BEQ SHT1
          LDA NCODE
          JSR OWNIT?
          BEQ SHT1
          JMP NOTHERE
SHT1
          LDA BULLET
          BNE SHT3
           PRINT  T52 ; "No bullets"
          JMP GOODPARSE
SHT3
           PRINT  T45 ; "BANG!"
          LDA #0
          STA BULLET
          LDA NCODE
          CMP #1        ; Locked door?
          BNE SHOOTX
           PRINT  T71 ; "Lock destroyed!"
          LDA #35       ; change door status in
          STA CUROBJS   ; object matrix
          LDX #1        ; and
          STA NTRANS,X  ; in the
          INX           ; translation
          STA NTRANS,X  ; table
          LDA #3        ; open west wall
          STA CURVECT+3 ; to room #3
          JSR SHOWVIS   ; show door change
          JSR SHOWVECTS ; and new vector
SHOOTX
          JMP GOODPARSE
;
;          INSERT
;          ------
DOINSERT
          JSR OWNIT?
          BEQ INS0
          JMP DONTHAVE
INS0
          LDA NCODE
          CMP #26       ; card?
          BNE INS1
           PRINT  T72 ; "Try examining things"
          JMP GOODPARSE
INS1
          CMP #24       ; ID?
          BEQ INS2
INSX
          JMP BESPEC
INS2
          LDA PLACE
          CMP #19       ; room 19?
          BNE INSX
          LDA #39       ; update object
          STA CUROBJS   ; matrix
          LDX #6        ; and
          STA NTRANS,X  ; translator
          LDA #21       ; open south wall
          STA CURVECT+1 ; to room 21
INEXIT
          JSR SHOWVIS
          JSR SHOWVECTS
          JMP POKAY
;
;          UNSCREW
;          -------
DOUNSCREW
          JSR INROOM?
          BEQ UNS0
          JMP NOTHERE
UNS0
          LDA UCODE
          CMP #24       ; nothing you can carry
          BCC UNS1      ; is unscrewable
          JMP WHYBOTH
UNS1
          LDA NCODE
          CMP #20       ; bolted unit?
          BNE UNS2
          JMP DOTAKE
UNS2
          CMP #3        ; closed grate?
          BEQ UNS3
          JMP IMPOSS
UNS3
          LDA #29       ; do you have
          JSR OWNIT?    ; the knife?
          BNE UNS4      ; nope
          LDA #37       ; patch
          STA CUROBJS   ; object
          LDX #4        ; and
          STA NTRANS,X  ; translator tables
          LDA #17       ; open south wall to
          STA CURVECT+1 ; room 17
          BNE INEXIT
UNS4
          LDA #22       ; screwdriver?
          JSR OWNIT?
          BNE UNSX
           PRINT  T73 ; "Blade's too tiny"
          JMP GOODPARSE
UNSX
          JMP EASIER
;
;          HOLD
;          ----
DOHOLD
          LDA UCODE
          CMP #38       ; breath?
          BEQ DHLD0
          JMP BESPEC
DHLD0
          LDA BREATH    ; already
          BNE DHLD1     ; holding
          LDA #9        ; it?
          STA BREATH    ; if not, set timer
          STA BHOLD     ; to 8 events
          JMP POKAY
DHLD1
          JMP ALREADY
;
;          UNLOCK
;          ------
DOUNLOCK
          LDA #33       ; key?
          JSR OWNIT?
          BEQ UNL0
          JMP EASIER
UNL0
          LDA NCODE
          JSR INROOM?
          BEQ UNL1
          JMP NOTHERE
UNL1
          LDA NCODE
          CMP #1        ; locked door?
          BNE UNL2
           PRINT  T75 ; "Key doesn't fit"
          JMP GOODPARSE
```

```
UNL2
        CMP #6      ; locked switch?
        BEQ UNL3
        JMP IMPOSS
UNL3
        LDA #40     ; unlocked
        STA CUROBJS
        LDX #7
        STA NTRANS,X
        STA SWITCH
        JMP INEXIT
;
;        CUT
;        ---
DOCUT
        JSR INROOM?
        BEQ DCT0
        JMP NOTHERE
DCT0
        LDA NCODE
        CMP #7      ; hummer?
        BEQ DCT1
        JMP IMPOSS
DCT1
        LDA #25     ; cutters?
        JSR OWNIT?
        BNE DCT2
        LDA GREEN
        BEQ CUTIT
        JSR NEWSCREEN
         POSITION  9,3
         PRINT  T76 ; "A jolt of high voltage"
        JMP KILLS
CUTIT
        LDA #41     ; severed
        STA CUROBJS+1
        LDX #8
        STA NTRANS,X
        LDA #2      ; blank scanner
        STA OBJECTS+42
        LDX #3
        STA NTRANS,X
        JMP INEXIT
DCT2
        JMP EASIER
;
;        POUR
;        ----
DOPOUR
        JSR OWNIT?
        BEQ DPR
        JMP DONTHAVE
DPR
        LDA NCODE
        CMP #30     ; shampoo?
        BEQ DOSHAM
        JMP IMPOSS
DOSHAM
        LDY INVPOS  ; remove
        LDA #$FF    ; shampoo from
        STA HOLDINGS,Y ; inventory
         PRINT  T44 ; "Shampoo all used up"
        LDA #20     ; is bolted unit
        JSR INROOM? ; nearby?
        BNE POURX   ; nope
        LDX ROOMPOS ; else change to
        LDA #42     ; free unit
        STA CUROBJS,X
        LDX #22     ; patch
        STA NTRANS,X ; translation
        INX         ; table
        STA NTRANS,X
POURX
        JSR SHOWVIS
        JSR SHOWINV
        JMP GOODPARSE
;
;        LUBRICATE
;        ---------
DOLUBE
        JSR OWNIT?
        BNE DOL
        JMP WHYBOTH
DOL
        LDA #30     ; got the shampoo?
        JSR OWNIT?
        BEQ DOL1
DOL0
        JMP EASIER
DOL1
        LDA NCODE
        JSR INROOM?
        BEQ DOSHAM
        JMP NOTHERE
;
;        WEAR
;        ----
DOWEAR
        CMP #28     ; mask?
        BEQ WEAROK
        CMP #32     ; suit?
        BEQ WEAROK
        JMP IMPOSS
WEAROK
        JMP DOTAKE
;
IMPOSS
         PRINT  T17 ; "That's impossible."
        BNE GOODPARSE
;
NOTYET
         PRINT  T23 ; "You can't do that yet."
        BNE GOODPARSE
;
BESPEC
         PRINT  T26 ; "Be more specific."
        BNE GOODPARSE
;
ARMSFULL
         PRINT  T24 ; "You're carrying too much!"
        BNE BADPARSE
;
ROOMFULL
         PRINT  T25 ; "Not enough room here."
        BNE BADPARSE
;
NOTHERE
         PRINT  T22 ; "It isn't here."
        BNE BADPARSE
;
ALREADY
         PRINT  T21 ; "You already have it."
        BNE BADPARSE
;
DONTHAVE
         PRINT  T27 ; "You don't have it."
        BNE BADPARSE
;
NOHELP
         PRINT  T32 ; "Doesn't help."
        BNE GOODPARSE
;
WHYBOTH
         PRINT  T33 ; "Why bother?"
        BNE GOODPARSE
;
SEEMSORD
         PRINT  T34 ; "Seems ordinary."
        BNE GOODPARSE
;
NOTHAP
         PRINT  T56 ; "Nothing happens."
        BNE GOODPARSE
;
EASIER
         PRINT  T50 ; "Easier said than done"
;
GOODPARSE
        JSR BEEP
        JMP NEXTEVENT
;
BADPARSE
        JSR BOOP
        JMP PARSER
;
;        SUBROUTINES
;        -----------
;
;        DLI ROUTINE
;        -----------
DLI
        PHA         ; save A,
        TXA
        PHA         ; X
        TYA         ; and
        PHA         ; Y registers
        LDX DLICOL  ; fetch color index
        LDA DCOLS,X ; fetch new color
        LDY CURSES,X ; and cursor shape
        STA WSYNC   ; wait for scan
        STA COLPF2  ; change color
        STY GRAFP0  ; and player shape
        INC DLICOL  ; update index
        PLA
        TAY         ; restore Y,
        PLA         ; X
        TAX         ; and
        PLA         ; A registers
        RTI         ; back to mainline
;
DCOLS
        .BYTE $70,$60,$70,$60,$70,$00,$60
CURSES
        .BYTE $00,$00,$00,$00,$00
CURSOR
        .BYTE $00,$00
;
;        PRINT MACRO SUBROUTINE
;        ----------------------
; ENTRY: Addr of EOL-terminated string
;        in A/Y registers (LSB/MSB).
;
EPRINT
        LDX #0      ; IOCB #0 (E:)
        STA ICBADR  ; lsb of string addr
        STY ICBADR+1 ; msb of string addr
        LDA #$09    ; PUT LINE command
        STA ICCOM
        LDA #$7F
        STA ICBLEN  ; lsb max string length
        STX ICBLEN+1 ; msb (0)
        JMP CIOV
;
;        POSITION MACRO ROUTINE
;        ----------------------
; ENTRY: X-pos in X, Y-pos in Y.
;
POSIT13
        LDX #13
POSIT
        STX COLCRS
        STY ROWCRS
        RTS
;
;        AUDIO PROMPTS
;        -------------
BEEP
        LDA #25     ; high tone
        BNE BTONE
;
BOOP
        LDA #100    ; low tone
BTONE
        STA AUDF1
        LDA #$AA    ; dist/vol = 10
        STA AUDC1
        LDA #0
        STA RTCLOK
SOUND
        LDA RTCLOK
        CMP #5      ; 5 jiffies
        BNE SOUND
        LDX #0
        STX AUDC1   ; silence!
        DEX
        STX CH      ; clear last key
        RTS
;
;        ERASE A STATUS LINE
;        -------------------
;
; ENTRY: Target line (0-18) in X.
;
ERASE
        LDA LADRSL,X ; lsb of line addr
        STA CLPOINT
        LDA LADRSH,X ; msb of addr
        STA CLPOINT+1
        LDY #24     ; clear 25 characters
        LDA #0
CLLOOP
        STA (CLPOINT),Y
        DEY
        BPL CLLOOP
        RTS
;
;        INIT SCREEN
;        -----------
NEWSCREEN
        LDA #$40
        STA NMIEN
        STA RAMTOP  ; set system to 16K
;
; Close IOCB #0 (E:)
;
        LDX #0      ; IOCB #0 (E:)
        LDA #$0C    ; CLOSE command
        STA ICCOM
        JSR CIOV    ; slam!
;
; Re-open E: with new RAM size
;
        LDX #0      ; E: again
        STX ICAUX2  ; zero this byte
        LDA #3      ; OPEN command
        STA ICCOM
        LDA # <EADR ; point to "E:"
        STA ICBADR
        LDA # >EADR
        STA ICBADR+1
        LDA #$0C    ; allow read/write
        STA ICAUX1
        JSR CIOV    ; do the OPEN
;
        LDA #0
        STA DMACTL
        STA SDMCTL  ; shut off ANTIC
        STA GRAFP0
        STA GRAFP1
        STA GRAFP2
        STA GRAFP3  ; blank out borders
;       LDX #8      ; all colors black
COLOFF
        STA PCOLR0,X
        DEX
        BPL COLOFF
        LDA #14
        STA COLOR1  ; except text
        LDA #$74
        STA COLOR0
;
        LDX #1
        STX CRSINH  ; disable system cursor
        INX         ; = 2
        STX LMARGN  ; fix margin
        LDA #$70
        STA POKMSK
        STA IRQEN   ; disable BREAK key
        RTS         ; and return
```

MORE UTILITY.

TOP-DOS. . . the best DOS for your Atari computer.

TOP-DOS is a powerful, user-friendly Disk Operating System (DOS) that gives you more functions and features than any other DOS written for Atari computers. Yet TOP-DOS is compatible with Atari software.

Here are a few of the features and commands you can expect to find in TOP-DOS.

POWERFUL

Turns large jobs into small ones.

You'll find all the familiar features of Atari DOS 2.0 in TOP-DOS but with added functions and six new commands.

Turn your major disk operations into small tasks with the ability to create a file of commands that can be executed in a single line. Or restore deleted files with one command.

FLEXIBLE

Tailor TOP-DOS to your needs.

TOP-DOS is flexible to meet the needs of virtually any user.

For example, the file directory can be listed in multi-column format to help you find your files. The amount of your involvement in file manipulations can be changed with the many options available. System parameters can be tailored to your preference with the SET command.

What independent reviewers say of TOP-DOS:

"TOP-DOS gives the user many advanced features found only in systems running on much larger machines. . . . This DOS is fast, easy to use, and extremely powerful, but most of all, a bargain. . . a best buy in the software marketplace."
Larry Dziegielewski, ANTIC

"The UNDELETE command is superb —one of the best things to come along in a long time."
Bill Howey, A.N.A.L.O.G.

"The first thing that really impressed me was the full use of the screen, giving me access to a large amount of work history."
Peter Ellison, ROM

FRIENDLY

TOP-DOS interacts with you.

Use computer prompts when you need help. As you gain experience switch to one-line commands for more speed.

You'd expect to pay more for all these features. However, TOP-DOS is only $49.95.

And there's no risk. TOP-DOS is guaranteed to please you or return it within 30 days for a full refund.

SAVE $10

If you order TOP-DOS before June 1, 1984 you will save $10 off the regular price. Order today and get more from your disk drive. Send a check or money order to ECLIPSE, 1058-A Marigold Court, Sunnyvale, CA 94086. Or call (408) 246-8325.

ECLIPSE

TOP-DOS handles double-density and double-sided Percom-compatible disk drives, and includes all the features of its predecessor, DOS-MOD. ATARI is a registered trademark of Atari, Inc. TOP-DOS and DOS-MOD are trademarks of ECLIPSE. Prices are subject to change without notice. Shipping charges are prepaid. California residents add 6.5% sales tax ($39.95 + 2.60 = $42.55; after June 1, 1984: $49.95 + 3.25 = $53.20).

CIRCLE #130 ON READER SERVICE CARD.

```
;
;         BLANK BAR
;         ---------
BAR
          PRINT  T9
          RTS
;
;         "WHAT" BAR
;         ----------
SAYWHAT
          PRINT  T10
          RTS
;
;         KEYBOARD INPUT HANDLER
;         ----------------------
; These routines are based in part on
; Steve Howard's "Alternative Keyboard Handler"
; (ANALOG Computing #15, pp. 96-103).
;
;         FETCH A KEYPRESS
;         ----------------
GETKEY
          LDA CH
          CMP #$FF      ; key pressed?
          BEQ GETKEY    ; not yet - keep scanning
;
; Analyze keycode
;
ANALYZE
          TAY           ; save key for later
          LDX #$FF
          STX CH        ; reset key
          AND #$C0      ; bit 6 or 7 set?
          BEQ LEGAL?    ; nope
;
; Handle a bad keypress
;
BADKEY
          JSR BOOP      ; razz user and
          JMP GETKEY    ; try again
;
; Look for illegal keys
;
LEGAL?
          TYA           ; restore keycode
          LDX #13
KLOOP
          CMP ILLEGAL,X
          BEQ BADKEY    ; razz if illegal key
          DEX
          BPL KLOOP
;
; Get ATASCII equivalent
;
          LDA ATASCI,Y
;
; Screen out numbers, pass EOL and BS
;
          CMP #SPACE    ; space bar?
          BEQ CLK1      ; that's okay
          CMP #EOL      ; RETURN?
          BEQ CLK1      ; fine by me
          CMP #$7E      ; backspace?
          BEQ CLK1      ; love 'em
          CMP #'a
           BCC BADKEY
          CLD
          SEC
          SBC #$20      ; convert to upper case
;
; Click the speaker
;
CLK1
          LDY #$7F
          STY CLICK
CLK2
          LDY CLICK
          STY CONSOL    ; tick!
          LDX #8        ; click freq
DELAY
          DEX
          BPL DELAY
          DEC CLICK
          BPL CLK2      ; 128 times
;
          RTS           ; ATASCII code in A
;
;         ILLEGAL KEYS
;         ------------
ILLEGAL
          .BYTE $1C     ; escape
          .BYTE $2C     ; tab
          .BYTE $27     ; atari
          .BYTE $3C     ; caps
          .BYTE $36     ; <
          .BYTE $37     ; >
          .BYTE $0F     ; =
          .BYTE $20     ; ;
          .BYTE $02     ; ;
          .BYTE $22     ; .
          .BYTE $26     ; /
          .BYTE $06     ; +
          .BYTE $07     ; *
          .BYTE $0E     ; -
;
;         ATASCII CONVERSION TABLE
;         ------------------------
; We use our own table because the
; location of the ROM-based table varies
; depending on which computer you have.
;
ATASCI
          .BYTE $6C,$6A,$3B,$8A,$8B,$6B,$2B,$2A
          .BYTE $6F,$80,$70,$75,$9B,$69,$2D,$3D
          .BYTE $76,$80,$63,$8C,$8D,$62,$78,$7A
          .BYTE $34,$80,$33,$36,$1B,$35,$32,$31
          .BYTE $2C,$20,$2E,$6E,$80,$6D,$2F,$81
          .BYTE $72,$80,$65,$79,$7F,$74,$77,$71
          .BYTE $39,$80,$30,$37,$7E,$38,$3C,$3E
          .BYTE $66,$68,$64,$80,$82,$67,$73,$61
;
;         INTERNAL CONVERSION TABLE
;         -------------------------
INTATA
          .BYTE $20,$40,$00,$60
;
;         Y-OFFSET TABLES
;         ---------------
; These two tables contain the
; starting address of each status line
; (absolute screen line address + 13).
; LADRSL holds the LSBs, LADRSH the MSBs.
;
LADRSL
          .BYTE  <SCREEN+53,  <SCREEN+53,  <SCREEN+93,  <SCREEN+133
          .BYTE  <SCREEN+173,  <SCREEN+213,  <SCREEN+253,  <SCREEN+293
          .BYTE  <SCREEN+333,  <SCREEN+373,  <SCREEN+413,  <SCREEN+453
          .BYTE  <SCREEN+493,  <SCREEN+533,  <SCREEN+573,  <SCREEN+613
          .BYTE  <SCREEN+653,  <SCREEN+693,  <SCREEN+733
LADRSH
          .BYTE  >SCREEN+53,  >SCREEN+53,  >SCREEN+93,  >SCREEN+133
          .BYTE  >SCREEN+173,  >SCREEN+213,  >SCREEN+253,  >SCREEN+293
          .BYTE  >SCREEN+333,  >SCREEN+373,  >SCREEN+413,  >SCREEN+453
          .BYTE  >SCREEN+493,  >SCREEN+533,  >SCREEN+573,  >SCREEN+613
          .BYTE  >SCREEN+653,  >SCREEN+693,  >SCREEN+733
;
;         FETCH INPUT LINE
;         ----------------
GETLINE
;
; Clear line input buffer
;
          CLD
          LDX #24
          LDA #SPACE
CLINL
          STA INLINE,X
          DEX
          BPL CLINL
;
; Get first character of line
;
GETONE
           POSITION  13,12
          LDX #$FF
          STX CURSEN    ; turn on PM8 cursor
          STX CH        ; clear key
          INX
          STX LENGTH    ; zero line length
          JSR GETKEY    ; fetch a keycode
          CMP #SPACE    ; first char not be
          BEQ BADONE    ; a space
          CMP #$7E      ; a backspace
          BEQ BADONE
          CMP #EOL      ; or an EOL
          BNE PUT1
;
; Handle bad first character
;
BADONE
          JSR BOOP      ; razz user and
          JMP GETONE    ; try again
;
; Print 1st char
;
PUT1
          JSR SETCIO    ; to E:
          JSR CIOV
          INC LENGTH
;
; Get rest of input line
;
REST
          JSR GETKEY    ; grab another keycode
          CMP #EOL      ; if it's an EOL,
          BEQ GOTEOL    ; line entry complete
          CMP #$7E      ; backspace?
          BNE PUTNEXT   ; no - send to screen
;
; Handle a backspace
;
BACKS
          DEC LENGTH    ; if 1st char of line
          BMI BADONE    ; signal error
          JSR SETCIO    ; E:
          JSR CIOV      ; let CIO do backspace
          LDA LENGTH    ; if length=0,
          BEQ GETONE    ; handle as 1st char
          BNE REST      ; else continue
;
; Print latest character
;
PUTNEXT
          JSR SETCIO
          JSR CIOV      ; print character
          INC LENGTH    ; next position
          LDA LENGTH
          CMP #24       ; end of line?
          BCC REST      ; not yet - get another key
;
; Too many chars in line!
;
GETLAST
          JSR BOOP      ; a warning razz
          JSR GETKEY
          CMP #EOL      ; must have an EOL
          BEQ GOTEOL
          CMP #$7E      ; or a backspace
          BEQ BACKS
          BNE GETLAST ; insist!
;
; EOL recieved
;
GOTEOL
          JSR SETCIO    ; specify E:
          STX CURSEN    ; shut off cursor
          JSR CIOV      ; and send EOL
;
; Convert screen bytes to ATASCII
; and move to INLINE
;
          LDY #24
TOBUFF
          CLC
          LDA SCREEN+493,Y ; grab screen byte
          STA SCREEN+453,Y ; move to upper line
          ROL A
          ROL A
          ROL A
          ROL A
          AND #3        ; transform byte, and
          TAX           ; use as an index
          LDA SCREEN+453,Y ; restore original value
          AND #$1F      ; clear bits 5-7
          ORA INTATA,X ; merge with code table
          STA INLINE,Y ; send to buffer
          LDA #0
          STA SCREEN+493,Y ; clear response line
          DEY
          BPL TOBUFF
          RTS
;
;         IMMEDIATE VBI ROUTINE
;         ---------------------
; Positions and blinks cursor,
; resets DLI color index
;
IMMVBI
;
; Reset DLI color index
;
          LDA #0
          STA DLICOL
;
; Okay to update cursor?
;
          LDA CURSEN    ; if enable flag = 0,
          BEQ VEXIT     ; don't redraw cursor
;
; Calculate cursor X-position:
; XNEW = ( XOLD * 4 ) + 48
;
          CLD
          LDA COLCRS
          ASL A
          ASL A         ; times 4
          CLC
          ADC #48       ; plus 48
          STA HPOSP0    ; use as h-pos
;
; Don't blink cursor if a key
; is being pressed.
;
          LDA SRTIMR    ; 0 = no press
          BEQ BLINK?
;
          LDA #$F0
          STA CURSOR
          STA CSHAPE    ; force cursor on
          LDA #60       ; for at least
          STA BLINK     ; 1 second
;
BLINK?
          LDA CSHAPE
          DEC BLINK     ; next jiffy
          BNE VEXIT     ; don't blink until 0
;
; Blink the cursor
;
          LDY #30
          STY BLINK     ; reset timer
          EOR #$F0      ; flip the cursor shape
          STA CSHAPE    ; and save it for later
VEXIT
          STA CURSOR    ; plot the cursor
          JMP SYSVBV    ; sayonara
;
;         DISPLAY EVENT COUNTER
;         ---------------------
```

The computer adventure you've been waiting for...

THE RETURN OF HERACLES™

An exploration of Greek mythology translated into modern electronics
by Stuart Smith

The creative mind of Stuart Smith, author of ***Ali Baba and the Forty Thieves,*** brings to life the world of mythic Greece in fantastic color and sound! Carefully researched and skillfully programmed, ***Return of Heracles*** is computer entertainment at its best. Twelve difficult and dangerous tasks will be assigned to you by Zeus, and your heroes must accomplish them all. One or more players take on the role of an ancient Greek hero or heroine. There are 19 heroes to choose from, or choose them all! May the gods favor you!

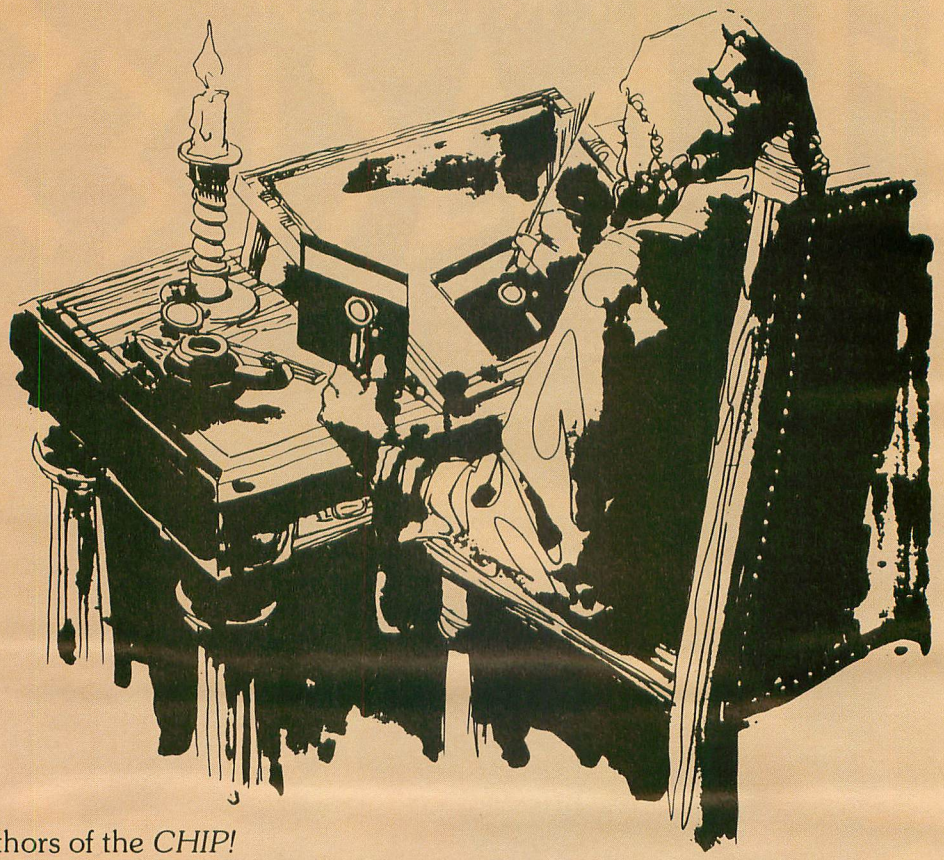
For Apple II, Atari, & Commodore 64 home computers. On diskette. Requires 48K. $32.95

Apple is a trademark of Apple Computer, Inc.
Atari is a trademark of Atari, Inc.
Commodore 64 is a trademark of Commodore Business Machines, Inc.

QS QUALITY SOFTWARE
21601 Marilla Street • Chatsworth, CA 91311
(818) 709-1721

CIRCLE #131 ON READER SERVICE CARD.

SpartaDOS and ARCHIVER II™



From the authors of the *CHIP!*

SpartaDOS is our all new command driven DOS fully compatible with all disk drives made for the Atari Home Computers.™ Supports all Atari compatible densities - both single and double sided, 5¼ and 8 inch.

UltraSpeed™ I/O allows read/write 2 to 4 times faster than standard!* Comprehensive utilities include: a sector copier, RS-232 handler, and DOS file translators.

ARCHIVER II allows UltraSpeed backup of all diskettes and compacting of multiple programs on each disk. Includes an exceptional editor, a speed check, and is fully automatic.

Available now for Atari 810 and 1050 drives! Coming soon for most other brands.

Suggested Retail Price $129.95 (plus shipping and installation)

See SpartaDOS and ARCHIVER II now at leading computer stores or order direct from:

Spartan Software
Div. of ICD, Inc.
828 Green Meadow Ave.
Rockford, IL 61107
815-229-2999 (1:00-9:00 p.m. CST)

*Speed depends on drive hardware. A chip replacement is required for most drives.

SpartaDOS, ARCHIVER II, and UltraSpeed are trademarks of ICD, Inc. Atari is a registered trademark of Atari, Inc.

CIRCLE #132 ON READER SERVICE CARD.

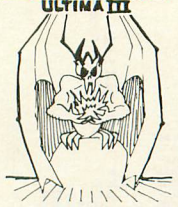
```
SHOWEV
;
; Initialize EBUFF
;
        LDA #'Ø
        STA EBUFF
        STA EBUFF+1
        STA EBUFF+2
        LDA #EOL
        STA EBUFF+4
;
; Convert event # to ATASCII
;
ECON
        LDA EVENT
        STA FRØ
        LDA EVENT+1
        STA FRØ+1
        JSR IFP     ; convert to floating point
        JSR FASC    ; then to ATASCII
        CLD
;
; Determine length of number
;
        LDY #$FF    ; init loop index
FINDE
        INY
        LDA (INBUFF),Y ; check characters
        BPL FINDE
;
; Change # to inverse video and
; move to EBUFF
;
        LDX #3      ; move 3 chars maximum
TOEB
        LDA (INBUFF),Y
        ORA #$80    ; set msb
        STA EBUFF,X ; put in EBUFF
        DEX
        DEY
        BPL TOEB
;
; Display contents of EBUFF
;
         POSITION  33,Ø
         PRINT  EBUFF
        RTS
;
;         SET CIO TO PUT CHAR MODE
;         ------------------------
SETCIO
        LDX #$ØB
        STX ICCOM
        LDX #Ø
        STX ICBLEN
        STX ICBLEN+1
        RTS
;
;         SYNTAX ERROR
;         ------------
SYNERR
         PRINT  T13 ; "Syntax:"
        RTS
;
;         CLEAR WINDOWS
;         -------------
; What Happens window
;
CLWH
        LDX #9
        JSR ERASE
        INX
        JSR ERASE
        LDY #9
        JMP POSIT13
;
; Inventory window
;
CLINV
        LDX #13
CLINV1
        JSR ERASE
        INX
        CPX #19
        BCC CLINV1
        RTS
;
;         DISPLAY ROOM VECTORS
;         --------------------
SHOWVECTS
        LDX #2
        JSR ERASE
        LDY #Ø
        LDX #Ø
        STX ANY?
SVL
        LDA CURVECT,X
        BMI VSKIP
        LDA VNAMES,X
        STA SCREEN+93,Y
        INC ANY?
        INY
        INY
VSKIP
        INX
        CPX #6
        BCC SVL
        LDA ANY?
        BNE VECTEX
         POSITION  13,2
         PRINT  T61 ; "None"
VECTEX
        RTS
;
; Vector initials
;
VNAMES
        .SBYTE "NSEWUD"
;
;         DISPLAY ROOM OBJECTS
;         --------------------
;
SHOWVIS
        LDX #3
CLVIS
        JSR ERASE
        INX
        CPX #9
        BCC CLVIS
         POSITION  13,3
        LDX #Ø
        STX ANY?
SHV1
        STX PDEX
        LDA CUROBJS,X
        BMI SHV2
        INC ANY?
        TAX
        LDA OBDLS,X
        LDY OBDHS,X
        JSR EPRINT
SHV2
        LDX PDEX
        INX
        CPX #6
        BCC SHV1
        LDA ANY?
        BNE SHEXIT
         PRINT  T28 ; "Nothing"
SHEXIT
        RTS
;
;         DISPLAY INVENTORY
;         -----------------
SHOWINV
        JSR CLINV
         POSITION  13,13
        LDX #Ø
        STX ANY?
SHI1
        STX PDEX
        LDA HOLDINGS,X
        BMI SHI2
        INC ANY?
        TAX
        LDA OBDLS,X
        LDY OBDHS,X
        JSR EPRINT
SHI2
        LDX PDEX
        INX
        CPX #6
        BCC SHI1
        LDA ANY?
        BNE SIEXIT
         PRINT  T28 ; "Nothing"
SIEXIT
        RTS
;
;         "OKAY" PROMPT
;         -------------
POKAY
        JSR CLWH
         PRINT  T20 ; "Okay"
        JMP GOODPARSE
;
;         SEARCH INVENTORY
;         ----------------
INVSPACE?
        LDA #$FF
OWNIT?
        LDX #5
OLOOP
        CMP HOLDINGS,X
        BEQ FOUND
        DEX
        BPL OLOOP
        TXA
        RTS
FOUND
        STX INVPOS
        LDA #Ø
        RTS
;
;         SEARCH ROOM
;         -----------
ROOMSPACE?
        LDA #$FF
INROOM?
        LDX #5
IRLOOP
        CMP CUROBJS,X
        BEQ FOUND2
        DEX
        BPL IRLOOP
        TXA
        RTS
FOUND2
        STX ROOMPOS
        LDA #Ø
        RTS
;
;         PRINT INTEGER IN FRØ
;         --------------------
;
VPRINT
        JSR IFP
        JSR FASC
        CLD
        LDY #$FF
VLOOP
        INY
        LDA (INBUFF),Y
        BPL VLOOP
        AND #$7F
        STA (INBUFF),Y
        INY
        LDA #EOL
        STA (INBUFF),Y
        LDA INBUFF
        LDY INBUFF+1
        JMP EPRINT
;
;         DEATH
;         -----
;
KILLS
         POSITION  10,5
         PRINT  T63 ; "kills you instantly!"
DEATH
         POSITION  24,7
         PRINT  T66 ; "YOU ARE DEAD"
        JMP STARTOP
;
;         CLOSE IOCB #1
;         -------------
;
CLOSE1
        LDX #$1Ø
        LDA #12
        STA ICCOM,X
        JMP CIOV
;
;         POINT TO FILENAMES
;         ------------------
;
DPOINT
        LDA # <FILE
        LDY # >FILE
        JMP POINT
TPOINT
        LDA # <CADR
        LDY # >CADR
POINT
        LDX #$1Ø
        STA ICBADR,X
        TYA
        STA ICBADR+1,X
        RTS
;
;         INIT DATABASE
;         -------------
;
INITDATA
;
; Set status flags
;
        LDX #19
        LDA #Ø
IDØ
        STA EVENT,X
        DEX
        BPL IDØ
;
; Clear vector/object matrix
;
        LDX #Ø
        LDA #$FF
ID1
        STA OBJECTS,X
        STA VECTORS,X
        INX
        CPX #132
        BCC ID1
;
; Clear working arrays
;
        LDX #5
ID2
        STA HOLDINGS,X
        STA CUROBJS,X
        STA CURVECT,X
        DEX
        BPL ID2
        LDX #1
        STX BULLET  ; 1 bullet
        STX PLACE   ; start in escape tube
        LDA #32
        STA MLAT
        LDA #168
        STA MLON
;
; Init tranlation table
;
        LDX #37
ID3
```

WE SPECIALIZE IN ATARI COMPUTERS SO WE KNOW WHAT WORKS BEST!

COMPUTER PALACE

CALL US AT (503) 683-5361 FOR INFORMATION ON THE LATEST AND BEST FOR ATARI

EXODUS ULTIMA III



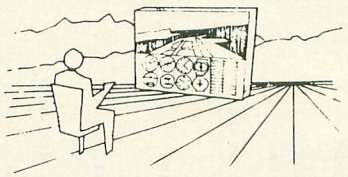
- 2-disk set!
- A lot of game for your money!

$52.50

New from Origin Systems.

Here is the latest in the famous series of Fantasy/Role-Playing games. This one is the best yet!

Flight Simulator II



So what if you haven't gotten your pilot's license yet . . . You can take off with this super-scrolling flight simulator and fly to your heart's content! The disk even contains a WW I flying ace dog fight game so that you can brush up on your aerial combat!

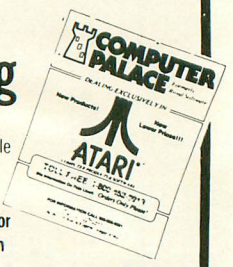
48K Disk From Sublogic **$44.90**

DIMENSION-X

What has unlimited power . . . Scrolling perspective . . . and super excitement? THIS DOES!!!

32K Disk/Tape from Synapse Only $31.50

FREE Catalog



With any order, or send $1 (refundable with purchase).

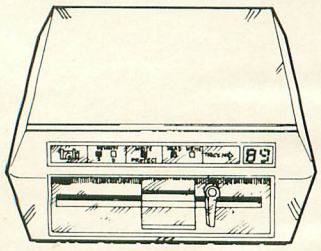
Send us your name & address for FREE FLYERS with Special Offers!

Use Your Credit Cards!

We honor VISA, American Express, & MasterCard.

"IF YOU DON'T HAVE OUR CATALOG... YOU'RE MISSING OUT!"

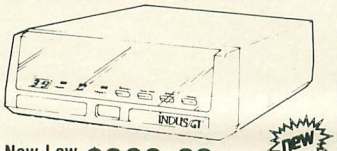
TRAK DISK DRIVE



- DOUBLE DENSITY
- BUILT-IN PARALLEL PRINTER INTERFACE & BUFFER

$398.00 Plus Frt.

INDUS-GT DISK DRIVE



New Low Price $398.00 + Frt.

- Quiet & Reliable
- Faster I/O than Atari 1050 Drive
- True Double Density Operation
- Includes Free Software: DOS—Word Processing—Data-Base—Spread Sheet
- Includes Carrying Case

ONE YEAR WARRANTY!!!

Now You Can Run CP/M On Your Atari!

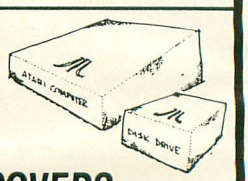
Priced from $299.00 Plus Freight (16K model, not CP/M compatible)

ATR-8000 Expander

- 64K, Z80 computer
- Runs any combination of 5¼" and 8" drives in single or double density
- Built-in serial and parallel ports
- Built-in printer buffer

64K CP/M model $449.00
2-DDDS 5¼" add-on drives 595.00
1-DDDS 5¼" add-on drive 379.00

Protect Your Investment with these durable and attractive



DUST COVERS

Custom Fitted For:

- ATARI 400,800,600XL,800XL,1200XL, 410,810,1010,1050,1025 and 1027
- EPSON MX-80 and FX-80
- GEMINI 10 • PROWRITER 8510
- PERCOM • RANA

Please specify **Only $8.95** Each

SPECIAL! 2 for $16.95

SUPER SPECIALS!

LIMITED QUANTITIES—WHILE SUPPLIES LAST!

(PLEASE LIST YOUR SECOND-CHOICE WHEN ORDERING)

MOUNTAIN KING

16K cartridge From CBS Reg. $39.95 **Only $17.95**

K-Razy Shoot Out

Reg. $39.95 **$17.95** cartridge from CBS

METEOR STORM

Reg. $29.95 **$9.95** 16K Disk from Royal Software

Boulders & Bombs

From CBS **$17.95** Reg. $39.95 16K CARTRIDGE

	Regular	SPECIAL
Chicken (D, T)	35.00	$19.95
Guardian of Gorn (D)	35.00	9.95
Genetic Drift (D)	30.00	14.95
Claim Jumper (D, T)	35.00	19.95
Protector II (D, T)	35.00	$19.95
David's Midnite Magic (D)	35.00	17.95
Race In Space (T)	30.00	19.95
Megalegs (T)	35.00	9.95

BOOKS & MAGAZINES

COMPUTE PUBLICATIONS:

1st Book of Atari	12.95
2nd Book of Atari	12.95
1st Book of Atari Graphics	12.95
1st Book of Robots & Computers	12.95
1st Book of Atari Games	12.95
Atari Basic Source Book	12.95
Inside Atari DOS	19.95
Mapping the Atari	14.95
Machine Language for Beginners	14.95

Machine Language Programming:

6502 Assembly Language Prgm.	18.95
6502 Assembly Language Subroutines	17.95
The Atari Assembler	14.95
De Re Atari (All About Atari)	17.90
Machine Language for Beginners	14.95

Basic Programming Applications:

Games Atari's Play	14.95
Kids and the Atari	19.95
Your Atari Computer	17.95

Magazines:

ANTIC-vol. 2, no. 11	2.50
Back issues-vol2, no.2,3,4,9,10	ea. 2.50
ANALOG-no.16	3.00
Back issues-no.13,14,15	ea. 3.00
HiRes-no.3	2.95
Back issues-no. 1, 2	ea. 2.95

SOFTWARE

Joust (C)	44.90
Robotron (C)	43.50
Blue Max (D, T)	31.50
Odesta Chess (D)	59.50
Pole Position (C)	44.90
Donkey Kong (C)	44.90
Monkey Wrench II (C)	54.00
Ultima II (D)	52.50
Dig Dug (C)	44.90
B/Graph (D)	89.90
The Tax Advantage (D)	62.50
Letter Perfect (C)	89.90
Spell Perfect (D)	69.50
Home Accountant (D)	67.50
Q-Bert (C)	44.90
Valforth (D)	54.50

HARDWARE (plus freight)

Astra 1620 Dual Drive	529.00
Rana 1000 Drive	349.00
Super-5 Printer	329.00
Atari 1027 Printer	319.00
Gemini & Epson Printers	CALL
Atari Graphics Tablet	79.10
Koala Pad Tablet	87.90
Power Pad Tablet	87.90
Axiom Printer Interface	99.00

COMPUTER PALACE

OPEN Monday - Saturday, 9 to 6

2160 W. 11th Avenue Eugene, Oregon 97402

USE YOUR CREDIT CARD & CALL

Toll Free 1-800-452-8013

(Orders Only, Please)

For Information Call (503) 683-5361

Shipping Note: UPS & Parcel Post-$2.90. UPS Air-$4.75 Shipping on hardware & paper supplies-extra. Call.

24 & 48 hour shipping available—Call (503) 683-5361

We specialize in *ATARI Compatible Products*
So we know what works best!!!
Call for the latest info... (503) 683-5361

CIRCLE #133 ON READER SERVICE CARD.

```
        LDA TRANS,X
        STA NTRANS,X
        DEX
        BPL ID3
;
; Init vector matrix
;
        LDX #39
ID4
        LDA VDATA,X
        LDY VSETS,X
        STA VECTORS,Y
        DEX
        BPL ID4
;
; Init object matrix
;
        LDX #31
ID5
        LDA ODATA,X
        LDY OSETS,X
        STA OBJECTS,Y
        DEX
        BPL ID5
        RTS
;
; Translator data
;
TRANS
        .BYTE 0,1,1,2,3,4,5,6
        .BYTE 7,8,9,10,11,12,13,14
        .BYTE 15,16,16,17,18,19,20,20
        .BYTE 21,22,23,24,25,26,27,28
        .BYTE 28,29,30,31,32,33
;
; Vector data
;
VDATA
        .BYTE 2,0,4,4,2,6,1,15
        .BYTE 6,4,9,7,5,6,9,6
        .BYTE 10,8,19,9,21,15,13,12
        .BYTE 15,12,14,16,14,4,15,14
        .BYTE 19,15,20,18,9,19,19,11
VSETS
        .BYTE 1,12,13,20,24,25,28,29
        .BYTE 32,36,37,38,39,45,50,54
        .BYTE 56,57,59,63,71,73,74,81
        .BYTE 86,90,91,92,93,94,99,102
        .BYTE 110,114,116,117,118,123,126,130
;
; Object data
;
ODATA
        .BYTE 20,7,0,22,8,9,23,1
        .BYTE 25,2,10,15,11,16,17,31
        .BYTE 17,12,13,27,28,3,30,26
        .BYTE 29,18,4,5,19,14,32,6
OSETS
        .BYTE 0,1,6,7,12,18,19,24
        .BYTE 30,42,43,48,49,54,60,61
        .BYTE 66,67,68,72,78,84,85,90
        .BYTE 96,102,108,114,115,116,120,126
;
;          DICTIONARY
;          ----------
;
; SINGLE-CHAR COMMANDS
; --------------------
SCOMS
        .BYTE "N"    ; 0 - North
        .BYTE "S"    ; 1 - South
        .BYTE "E"    ; 2 - East
        .BYTE "W"    ; 3 - West
        .BYTE "U"    ; 4 - Up
        .BYTE "D"    ; 5 - Down
        .BYTE "Q"    ; 6 - Quit
        .BYTE "X"    ; 7 - Save
        .BYTE "A"    ; 8 - Again
;
; COMPOUND VERBS
; --------------
VERBS
        .BYTE "TAK" ; 0 - take
        .BYTE "GET" ; 1 - get
        .BYTE "PUL" ; 2 - pull
        .BYTE "DRO" ; 3 - drop
        .BYTE "REM" ; 4 - remove
        .BYTE "LOO" ; 5 - look
        .BYTE "EXA" ; 6 - examine
        .BYTE "SEA" ; 7 - search
        .BYTE "REA" ; 8 - read
        .BYTE "PUS" ; 9 - push
        .BYTE "PRE" ; 10 - press
        .BYTE "OPE" ; 11 - open
        .BYTE "CLO" ; 12 - close
        .BYTE "USE" ; 13 - use
        .BYTE "KIL" ; 14 - kill
        .BYTE "SHO" ; 15 - shoot
        .BYTE "FIR" ; 16 - fire
        .BYTE "BRE" ; 17 - break
        .BYTE "SMA" ; 18 - smash
        .BYTE "INS" ; 19 - insert
        .BYTE "UNS" ; 20 - unscrew
        .BYTE "HOL" ; 21 - hold
        .BYTE "UNL" ; 22 - unlock
        .BYTE "CUT" ; 23 - cut
        .BYTE "POU" ; 24 - pour
        .BYTE "LUB" ; 25 - lubricate
        .BYTE "GRE" ; 26 - grease
        .BYTE "WEA" ; 27 - wear
        .BYTE "GO " ; 28 - go
;
NV      =   29      ; total # of verbs
;
; VERB VECTOR TABLE
; -----------------
VVECTS
        .WORD DOTAKE ; 0
        .WORD DOTAKE ; 1
        .WORD EASIER ; 2
        .WORD DODROP ; 3
        .WORD DOREMOVE ; 4
        .WORD DOLOOK ; 5
        .WORD DOLOOK ; 6
        .WORD DOLOOK ; 7
        .WORD DOREAD ; 8
        .WORD DOPUSH ; 9
        .WORD DOPUSH ; 10
        .WORD DOOPEN ; 11
        .WORD WHYBOTH ; 12
        .WORD BESPEC ; 13
        .WORD EASIER ; 14
        .WORD DOSHOOT ; 15
        .WORD DOSHOOT ; 16
        .WORD WHYBOTH ; 17
        .WORD WHYBOTH ; 18
        .WORD DOINSERT ; 19
        .WORD DOUNSCREW ; 20
        .WORD DOHOLD ; 21
        .WORD DOUNLOCK ; 22
        .WORD DOCUT ; 23
        .WORD DOPOUR ; 24
        .WORD DOLUBE ; 25
        .WORD DOLUBE ; 26
        .WORD DOWEAR ; 27
;
; COMPOUND NOUNS
; --------------
NOUNS
        .BYTE "HAT" ; 0 - hatch
        .BYTE "DOO" ; 1 - door
        .BYTE "LOC" ; 2 - lock
        .BYTE "SCA" ; 3 - scanner
        .BYTE "GRA" ; 4 - grate
        .BYTE "TRA" ; 5 - traitor
        .BYTE "AIR" ; 6 - airlock
        .BYTE "SWI" ; 7 - switch
        .BYTE "CAB" ; 8 - cable
        .BYTE "SIG" ; 9 - sign
        .BYTE "CAP" ; 10 - captain
        .BYTE "GRE" ; 11 - green
        .BYTE "RED" ; 12 - red
        .BYTE "GOL" ; 13 - gold
        .BYTE "SIL" ; 14 - silver
        .BYTE "WHI" ; 15 - white
        .BYTE "GAU" ; 16 - gauge
        .BYTE "PER" ; 17 - periscope
        .BYTE "SCO" ; 18 - scope
        .BYTE "DIS" ; 19 - display
        .BYTE "DUC" ; 20 - duct
        .BYTE "SLO" ; 21 - slot
        .BYTE "BOL" ; 22 - bolt
        .BYTE "UNI" ; 23 - unit
        .BYTE "PIS" ; 24 - pistol
        .BYTE "SCR" ; 25 - screwdriver
        .BYTE "NOT" ; 26 - note
        .BYTE "ID " ; 27 - ID
        .BYTE "CUT" ; 28 - cutters
        .BYTE "CAR" ; 29 - card
        .BYTE "WRE" ; 30 - wrench
        .BYTE "MAS" ; 31 - mask
        .BYTE "GAS" ; 32 - gas
        .BYTE "KNI" ; 33 - knife
        .BYTE "SHA" ; 34 - shampoo
        .BYTE "MAN" ; 35 - manual
        .BYTE "SUI" ; 36 - suit
        .BYTE "KEY" ; 37 - key
        .BYTE "BRE" ; 38 - breath
        .BYTE "BUT" ; 39 - button
        .BYTE "INV" ; 40 - inventory
;
NNN     =   41      ; total # of nouns
;
;         ROOM DESCRIPTIONS
;         -----------------
;
RD0
        TEXT  "Sonar sphere"
RD1
        TEXT  "Escape tube"
RD2
        TEXT  "Access tunnel"
RD3
        TEXT  "Captain's quarters"
RD4
        TEXT  "Forward passage"
RD5
        TEXT  "Radio room"
RD6
        TEXT  "Long corridor"
RD7
        TEXT  "Sonar station"
RD8
        TEXT  "Ballast control"
RD9
        TEXT  "Command station"
RD10
        TEXT  "Navigation center"
RD11
        TEXT  "Upper missile bay"
RD12
        TEXT  "Torpedo room"
RD13
        TEXT  "Weapons locker"
RD14
        TEXT  "Shower stalls"
RD15
        TEXT  "Crew's quarters"
RD16
        TEXT  "Galley"
RD17
        TEXT  "Ventilation duct"
RD18
        TEXT  "Fan room"
RD19
        TEXT  "Missile control"
RD20
        TEXT  "Equipment bay"
RD21
        TEXT  "Lower missile bay"
;
;         ROOM DESC. ADDRESS TABLES
;         -------------------------
;
RDLS
        .BYTE  <RD0, <RD1, <RD2, <RD3, <RD4, <RD5
        .BYTE  <RD6, <RD7, <RD8, <RD9, <RD10, <RD11
        .BYTE  <RD12, <RD13, <RD14, <RD15, <RD16, <RD17
        .BYTE  <RD18, <RD19, <RD20, <RD21
;
RDHS
        .BYTE  >RD0, >RD1, >RD2, >RD3, >RD4, >RD5
        .BYTE  >RD6, >RD7, >RD8, >RD9, >RD10, >RD11
        .BYTE  >RD12, >RD13, >RD14, >RD15, >RD16, >RD17
        .BYTE  >RD18, >RD19, >RD20, >RD21
;
;         OBJECT DESCRIPTIONS
;         -------------------
;
OBD0
        TEXT  "Closed hatch"
OBD1
        TEXT  "Locked door"
OBD2
        TEXT  "Blank scanner"
OBD3
        TEXT  "Closed grate"
OBD4
        TEXT  "Traitor with pistol"
OBD5
        TEXT  "Closed airlock"
OBD6
        TEXT  "Locked arming switch"
OBD7
        TEXT  "Power cable"
OBD8
        TEXT  "Sign"
OBD9
        TEXT  "Dead captain"
OBD10
        TEXT  "Green button"
OBD11
        TEXT  "Red button"
OBD12
        TEXT  "Gold button"
OBD13
        TEXT  "Silver button"
OBD14
        TEXT  "White button"
OBD15
        TEXT  "Depth gauge"
OBD16
        TEXT  "Periscope"
OBD17
        TEXT  "Digital display"
OBD18
        TEXT  "Duct down to fan room"
OBD19
        TEXT  "Slot in airlock"
OBD20
        TEXT  "Bolted-down sonar unit"
OBD21
        TEXT  "Pistol"
OBD22
        TEXT  "Tiny screwdriver"
OBD23
        TEXT  "Suicide note"
OBD24
        TEXT  "Security ID"
OBD25
        TEXT  "Cable cutters"
OBD26
        TEXT  "Card"
OBD27
        TEXT  "Wrench"
OBD28
        TEXT  "Gas mask"
OBD29
        TEXT  "Dull knife"
OBD30
        TEXT  "Shampoo"
OBD31
        TEXT  "Tactics manual"
OBD32
```

```
OBD33   TEXT  "Radiation suit"
OBD34   TEXT  "Key"
OBD35   TEXT  "Open hatch"
OBD36   TEXT  "Open door"
OBD37   TEXT  "Active scanner"
OBD38   TEXT  "Open grate"
OBD39   TEXT  "Dead traitor"
OBD40   TEXT  "Open airlock"
OBD41   TEXT  "Activated switch"
OBD42   TEXT  "Severed cable"
        TEXT  "Radioactive sonar unit"
;
;       DESCRIPTION ADDR TABLES
;       -----------------------
OBDLS
        .BYTE  <OBD0, <OBD1, <OBD2, <OBD3
        .BYTE  <OBD4, <OBD5, <OBD6, <OBD7
        .BYTE  <OBD8, <OBD9, <OBD10, <OBD11
        .BYTE  <OBD12, <OBD13, <OBD14, <OBD15
        .BYTE  <OBD16, <OBD17, <OBD18, <OBD19
        .BYTE  <OBD20, <OBD21, <OBD22, <OBD23
        .BYTE  <OBD24, <OBD25, <OBD26, <OBD27
        .BYTE  <OBD28, <OBD29, <OBD30, <OBD31
        .BYTE  <OBD32, <OBD33, <OBD34, <OBD35
        .BYTE  <OBD36, <OBD37, <OBD38, <OBD39
        .BYTE  <OBD40, <OBD41, <OBD42
;
OBDHS
        .BYTE  >OBD0, >OBD1, >OBD2, >OBD3
        .BYTE  >OBD4, >OBD5, >OBD6, >OBD7
        .BYTE  >OBD8, >OBD9, >OBD10, >OBD11
        .BYTE  >OBD12, >OBD13, >OBD14, >OBD15
        .BYTE  >OBD16, >OBD17, >OBD18, >OBD19
        .BYTE  >OBD20, >OBD21, >OBD22, >OBD23
        .BYTE  >OBD24, >OBD25, >OBD26, >OBD27
        .BYTE  >OBD28, >OBD29, >OBD30, >OBD31
        .BYTE  >OBD32, >OBD33, >OBD34, >OBD35
        .BYTE  >OBD36, >OBD37, >OBD38, >OBD39
        .BYTE  >OBD40, >OBD41, >OBD42
;
;       TEXT
;       ----
;
EADR    TEXT  "E:"
CADR    TEXT  "C:"
FILE    TEXT  "D1:GAME.DAT"
T0      TEXT  "Brian Moriarty's"
T1      TEXT  "CRASH DIVE"
T2      TEXT  "(C)1984 ANALOG Computing"
T3      TEXT  "Press  START  to play new game"
T4      TEXT  "Press  OPTION  to restore old game"
T5      TEXT  "Restore from Disk or Cassette?"
T6      TEXT  " CRASH DIVE! (TM)       EVENT #      "
T7      TEXT  " LOCATION        EXITS "
T8      TEXT  "  VISIBLE        ITEMS "
T9      TEXT  "             "
T10     TEXT  "     WHAT "
T11     TEXT  "  HAPPENS        YOUR    RESPONSE "
T12     TEXT  "     YOU          ARE    CARRYING "
T13     TEXT  "Syntax:"
T14     TEXT  "Bad command"
T15     TEXT  "Bad verb"
T16     TEXT  "Bad noun"
T17     TEXT  "That's impossible"
T18     TEXT  "Can't go that way"
T19     TEXT  "Type Y to quit game:"
T20     TEXT  "Okay"
T21     TEXT  "Already holding it"
T22     TEXT  "Isn't here"
T23     TEXT  "Can't do that yet"
T24     TEXT  "Your arms are full!"
T25     TEXT  "Not enough room here"
T26     TEXT  "Be more specific"
T27     TEXT  "You don't have it"
T28     TEXT  "Nothing"
T29     TEXT  "Type N S E W U or D"
T30     TEXT  "Refer to it by color"
T31     TEXT  "Type I for inventory"
T32     TEXT  "Doesn't help"
T33     TEXT  "Why bother?"
T34     TEXT  "Seems ordinary"
T35     TEXT  "You found something!"
T36     TEXT  "It's airtight"
T37     TEXT  "Enemy approaching!"
T38     TEXT  "Screwed in place"
T39     TEXT  "Looks dangerous"
T40     TEXT  "Needs key to activate"
T41     TEXT  "Bolts are tight & rusty"
T42     TEXT  "DANGER: Radiation zone!"
T43     TEXT  "Brains blown out"
T44     TEXT  "Shampoo all used up"
T45     TEXT  "BANG!"
T46     TEXT  "Accepts security ID card"
T47     TEXT  "It glows"
T48     TEXT  "See photo in ANALOG #18"
T49     TEXT  "Ace of Spades!"
T50     TEXT  "Easier said than done"
T51     TEXT  "Only 1 bullet"
T52     TEXT  "No bullets"
T53     TEXT  "X ="
T54     TEXT  "Y ="
TT55    TEXT  "FATHOMS"
T56     TEXT  "Nothing happens"
T59     TEXT  "Sub levels off"
T60     TEXT  "Sub dives"
T61     TEXT  "None"
T62     TEXT  "A cloud of poisonous gas"
T63     TEXT  "kills you instantly!"
T64     TEXT  "Lock is very secure"
T665    TEXT  "Already open"
T66     TEXT  "you are DEAD"
T67     TEXT  "Enemy captures the sub and"
T68     TEXT  "It falls down the pipe"
T71     TEXT  "Lock destroyed!"
T72     TEXT  "Try examining things"
T73     TEXT  "Screwdriver's too tiny"
T75     TEXT  "Key won't fit"
T76     TEXT  "A jolt of high voltage"
T77     TEXT  "Traitor shoots you and"
T78     TEXT  "Sub hits bottom!"
T79     TEXT  "A blast of radioactivity"
T80     TEXT  "Bolts won't let you"
T81     TEXT  "Congratulations!"
T82     TEXT  "Save game to Disk or Cassette?"
T83     TEXT  "Connected to cable"
;
```


Attention Programmers!

ANALOG Computing is interested in programs, articles, and software review submissions dealing with the Atari home computers. If you feel that you can write as well as you can program, then submit those articles and reviews that have been floating around in your head, awaiting publication. This is your opportunity to share your knowledge with the growing family of Atari computer owners. **ANALOG** pays between $30.00-$360.00 for all articles. All submissions for publication must be typed, upper and lower case with double spacing. Program listings should be provided in printed form, and on cassette or disk. By submitting articles to **ANALOG Computing**, authors acknowledge that such materials, upon acceptance for publication, become the exclusive property of **ANALOG**. If not accepted for publication, the articles and/or programs will remain the property of the author. If submissions are to be returned. please supply a self-addressed, stamped envelope. All submissions of any kind must be accompanied by the author's full address and telephone number. Send programs to: Editor, **ANALOG Computing**, P.O. Box 23, Worcester, MA 01603.

Is It a Game — Or Is It Real?

NATO COMMANDER

- Accelerated real-time simulation
- Full scrolling European Continent
- Demonstration Mode
- Save Game Feature
- Five Battle Scenarios
- Four Skill Levels

PLAYERS HAVE:

- Tactical and Strategic Air Power
- Tactical and Nuclear Forces
- Continuous Battlefield Status and News Service Updates
- Multiple Combat Units, Including: Infantry, Mechanized Armor and Attack Helicopters

Nato Commander is available for Atari and Commodore 64 computers, 48K disk or cassette. IBM-PC version will be available shortly.

Suggested retail price for this outstanding real-time combat simulation is $34.95.

If you can't find our games at your local store you can order by Master Card, Visa, Money Order, COD or check. Add $2.50 postage and handling. MD residents add 5% sales tax.

MicroProse Software

10616 Beaver Dam Road, Hunt Valley, MD 21030
(301) 667-1151

DEALER INQUIRIES INVITED

ATARI is the registered trademark of Atari, Inc.
Commodore 64 is the registered trademark of Commodore Business Machines, Inc.
IBM-PC is the registered trademark of International Business Machines, Inc.

CIRCLE #134 ON READER SERVICE CARD.

BASIC Training

by Tom Hudson

For the next couple of issues, we're going to be looking at some graphics concepts that will be new to many readers. Those BASIC programmers interested in writing their own games should find these routines very helpful.

What's a vector?

Generally, when computer techno-types are discussing graphics, you'll hear the word "vector" tossed around. Actually, as far as we're concerned, a vector is simply a direction.

Whenever the Atari computer performs a DRAWTO command, it calculates a vector (or direction) from the last point plotted (point 1) to the point being drawn to (point 2). The vector from point 1 to point 2 can be left, right, up, down or any other direction. Simple enough, right?

Many games, such as **Robotron**, **Missile Command** and even **Star Raiders** use the concept of vectors. In **Missile Command**, enemy missiles are approaching your cities, intending to turn them into smoking ruins. In order to draw the missile tracks on the screen, the computer must calculate a vector from the missile's starting point to the target city.

Many BASIC programmers would like to use vectoring in their games, but don't know how. If you want an electrified robot to chase Blaster Bob (TM) around on the screen, the robot must know which direction Bob is in. The following programs demonstrate several ways to accomplish this.

The first type of vector calculation most programmers discover is what I call X-Y matching. This is the simplest and fastest vector calculation, but as you will see, it has some drawbacks.

Let's say we're writing a game called **Get Bob**, starring our friend, Blaster Bob. Bob's being menaced by the usual electrified robot. Assuming we've chosen the X-Y matching vector schemes, let's see how the robot will act.

Each time the robot gets ready to move, it looks to see what direction Bob is in. This robot is not very sophisticated, and only knows if Bob is above or below, and left or right. The robot can only move in eight directions (the same as an Atari joystick). If Bob is directly above the robot, it will move straight up. If he is directly to the right of the robot, it will move right.

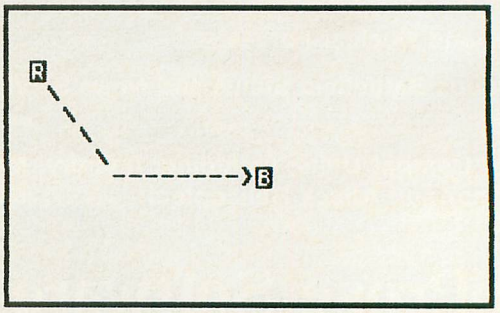


Figure 1.

Problems start when Bob is not at one of the eight directions the robot can move in. As **Figure 1** shows, if Bob is at an odd angle, the robot's path to him is jagged. This is the main drawback to the X-Y matching algorithm.

As I said before, the X-Y matching algorithm is the easiest vector method to program. **Figure 2** shows the basic code necessary to implement this type of vector.

Figure 2.

```
10 REM *** SIMPLE X-Y MATCHING ***
20 REM
30 GRAPHICS 6:COLOR 1
40 ? "ENTER FROM COORDS X,Y";:TRAP 40:
INPUT FX,FY:IF FX<0 OR FX>159 OR FY<0
OR FY>79 THEN ? "■":GOTO 40
50 PLOT FX,FY
60 ? "ENTER TO COORDS X,Y";:TRAP 50:IN
PUT TX,TY:IF TX<0 OR TX>159 OR TY<0 OR
 TY>79 THEN ? "■":GOTO 50
70 PLOT TX,TY
80 XV=SGN(TX-FX)
90 YV=SGN(TY-FY)
100 IF  NOT XV AND  NOT YV THEN 140
110 FX=FX+XV
120 FY=FY+YV
130 PLOT FX,FY:GOTO 80
140 ? "VECTOR COMPLETE":END
```

●

CHECKSUM DATA
(See p. 30)

```
10 DATA 498,253,28,496,104,443,136,595
,601,526,821,829,368,953,6651
```

●

Line 30 places the computer in graphics mode 6 and selects color 1. The program will graphically illustrate how the vector code works.

Lines 40-70 accept the X and Y coordinates of the starting and ending points of the line and plot them on the screen. In graphics mode 6, the X coordinate ranges from 0-159 and the Y coordinate from 0-79. If you exceed these values, the console speaker will beep and you must re-enter the coordinates.

Line 80 finds out if the point being drawn to is to the left or right of the starting point. By using the sign function (SGN), the X vector (XV) will be set to -1 (left), 0 (no movement) or 1 (right).

Line 90 finds out if the point being drawn to is above or below the starting point. The Y vector (YV) will contain -1 (up), 0 (no movement) or 1 (down).

Line 100 is the line end test. If both XV and YV are zero, FX and FY have reached the same point as TX and TY, and the line is finished.

Line 110 adds the X vector (XV) to the X coordinate of the starting point (FX).

Line 120 adds the Y vector (YV) to the Y coordinate.

Line 130 plots the new point and loops back to Line 80 to get the next vector.

Line 140 informs you when the vector drawing is complete.

Try entering different values for the starting and ending points, observing how the line acts with different endpoints. You can see that, although this method may be good for some applications, you certainly wouldn't want to use it for drawing pictures!

True vectors (part 1).

Let's say **Get Bob** was a tremendous success, and all the rabid video-gamers out there want to see more of Blaster Bob. Well, you sit down to write **Get Bob II**, but want the robot to be smarter, and head directly toward Bob, no matter which direction he's in. You need a true vector routine.

True vectors are more tricky than our simple X-Y matching. In order for a vector to go in the EXACT direction, we must get more information about the target point. **Figure 3** shows one type of true vector algorithm.

Figure 3.

```
10 REM *** TRUE VECTOR METHOD 1 ***
20 REM
30 GRAPHICS 6:COLOR 1
40 ? "ENTER FROM COORDS X,Y";:TRAP 40:
INPUT FX,FY:IF FX<0 OR FX>159 OR FY<0
OR FY>79 THEN ? "■":GOTO 40
50 PLOT FX,FY
60 ? "ENTER TO COORDS X,Y";:TRAP 50:IN
PUT TX,TY:IF TX<0 OR TX>159 OR TY<0 OR
 TY>79 THEN ? "■":GOTO 50
70 PLOT TX,TY
80 XD=SGN(TX-FX)
90 YD=SGN(TY-FY)
100 DELTAX=ABS(TX-FX)
110 DELTAY=ABS(TY-FY)
120 IF DELTAX>1 OR DELTAY>1 THEN DELTA
X=DELTAX/2:DELTAY=DELTAY/2:GOTO 120
130 XV=DELTAX*XD
140 YV=DELTAY*YD
150 IF INT(FX+0.5)=INT(TX) AND INT(FY+
0.5)=INT(TY) THEN 200
160 FX=FX+XV
170 FY=FY+YV
180 PLOT FX,FY
190 GOTO 150
200 ? "VECTOR COMPLETE":END
```

●

CHECKSUM DATA
(See p. 30)

```
10 DATA 356,253,28,496,104,443,136,541
,547,173,184,198,568,577,216,4820
160 DATA 836,844,146,725,943,3494
```

●

With this method, we find the differences in the X and Y coordinates of the two points, called DELTA X and DELTA Y.

Once these are found, we divide both by 2 repeat-

edly until they are BOTH less than or equal to 1. The resulting numbers are the vectors we will add to the starting coordinates in order to get to the ending point.

Lines 30-70 perform the same function as those in **Figure 2**.

Lines 80-90 get the general direction of the target point, just as in X-Y matching.

Lines 100-110 find the DELTA X and DELTA Y values. The absolute value function makes the numbers positive, since distances cannot be negative.

Line 120 divides both DELTAX and DELTAY by 2 until both are less than or equal to 1. This ensures that the line we draw will never move more than 1 pixel at a time.

Lines 130-140 calculate the final X and Y vectors (XV and YV) by multiplying the limited DELTA values by the X and Y directions (-1, 0 or 1).

Line 150 checks to see if the line has reached the endpoint yet. If both X coordinates are equal and both Y coordinates are equal, the line is complete, and the program skips to Line 200.

Lines 160-170 increment the line coordinates (FX and FY) by the proper vector amounts.

Line 180 plots the next point in the line on the screen.

Line 190 loops back to Line 150 for the next plot.

Line 200 prints the end message and stops the program's execution.

RUN this program several times with different endpoints and observe that the line generated is much better than that generated by X-Y matching. This is not the best line we can obtain, though. The next vector algorithm is the one used by the Atari operating system to draw lines in the graphics modes, and always gives good results.

True vectors (part 2).

When you tell your computer to draw a line from one part of the screen to another, you set in motion a fairly complex chain of events. The Atari operating system manual describes the algorithm on page 184. This is the vector algorithm used in virtually every computer, and is quite involved. Since a total understanding of this algorithm is not essential in order to use it, I'll describe the major points of interest and let the more advanced programmers take it from there.

COMPUTER DISCOUNT SALES

4251 West Sahara Avenue, Suite E, Las Vegas, Nevada 89102

(1-800-621-6131 ORDER LINE ONLY) 1-702-367-2215

Buy • Sell • Trade • New • Used • Demo. "Computer Hot Line"

ATARI

1200XL Reflects	Rebate $ 259
800 48k Reflects	Rebate 289
400 16K	99
810 Disk Drive	379
410 Recorder	72
830 Modem	129
850 Interface	199

Atari Visicalc $159

SPECIAL OF THE MONTH

ATARI SOFTWARE

Bookkeeper Kit	$165
Programmer Kit	52
Atari Writer	55
Data Perfect	75
Letter Perfect	105
Home Accountant	48
Assembler Editor	42
Atari Speed Reading	54
Home File Manager	36
Bookkeeper	102
Pac-Man	30
Centipede	30
Qix	30
Breakout	26
Deluxe Space Invaders	26
Asteroids	30
Juggles House (D.C.)	22
Juggles Rainbow	22

ATARI

Preppie II	$22
Frogger	23
Choplifter	23
Apple Panic	20
Zork I	27
Zork II	27
Deadline	34

APPLE

Home Accountant	$48
Multiplan	175
Compiler	119

HARDWARE

C. Itoh Prowriter	$379
Nec. 8023A	439
Banana Printer	209
Hayes 1200 Modem	489

COMM. 64 DISK

Temple Of Apshai	$33
Upper Reaches Apshai	18
Jump Man	33
Zork I	33
Zork II	33
Zork III	33
Frogger	30
Jawbreaker	24
Ft. Apocalyse	30
Pharoh's Curse	30
Type Attack	30

Comm 64	$219
1541 Disk Drive	249
1525 Printer	229
1701 Color Mont.	259
Hes Mon	29
Paper Clip w/p	115
Calc Result	140
Sysres-Utility	90
Renaissance	30
Vic-20	90
Datasette	64
1600 Modem	85
Word Processor	95
1311 Joystick	8
1312 Paddles	16
1210 3k Expander	34
Apple Panic	34
Serpentine	34
Sargon Chess	29
Gorf	29

UMI-VIC 20

Renaissance	$30
Meteor Run	30
Outworld	30
Super Amok	30
Video Vermin	30
Amok	15
Subchase	15

Atari Conversational Languages • French • Italian • Spanish • German • $42 each

Purchases can be made by check, money order, C.O.D., Carte Blanche and Diners Club.

CIRCLE #135 ON READER SERVICE CARD.

Figure 4.

```
10 REM *** TRUE VECTOR METHOD 2 ***
20 REM
30 GRAPHICS 6:COLOR 1
40 ? "ENTER FROM COORDS X,Y";:TRAP 40:
INPUT FX,FY:IF FX<0 OR FX>159 OR FY<0
OR FY>79 THEN ? "■":GOTO 40
50 PLOT FX,FY
60 ? "ENTER TO COORDS X,Y";:TRAP 50:IN
PUT TX,TY:IF TX<0 OR TX>159 OR TY<0 OR
 TY>79 THEN ? "■":GOTO 50
70 PLOT TX,TY
80 XD=SGN(TX-FX)
90 YD=SGN(TY-FY)
100 DELTAX=ABS(TX-FX)
110 DELTAY=ABS(TY-FY)
120 XACC=0
130 YACC=0
140 COUNT=DELTAX:IF DELTAY>DELTAX THEN
 COUNT=DELTAY
150 EPOINT=COUNT
160 IF COUNT=DELTAX THEN YACC=EPOINT/2
:GOTO 180
170 XACC=EPOINT/2
180 IF COUNT=0 THEN 260
190 XACC=XACC+DELTAX
200 IF XACC>=EPOINT THEN XACC=XACC-EPO
INT:FX=FX+XD
210 YACC=YACC+DELTAY
220 IF YACC>=EPOINT THEN YACC=YACC-EPO
INT:FY=FY+YD
230 PLOT FX,FY
240 COUNT=COUNT-1
250 GOTO 180
260 ? "VECTOR COMPLETE":END
```

CHECKSUM DATA

(See p. 30)

```
10 DATA 358,253,28,496,104,443,136,541
,547,173,184,646,652,402,653,5616
160 DATA 334,420,959,874,24,860,46,133
,418,724,961,5753
```

•

Figure 4 is a vector program using the Atari vector algorithm.

Lines 30-70 perform the same function as those in **Figure 2**.

Lines 80-90 get the X and Y direction values. Note that this function is essential to all types of vector algorithms.

Lines 100-110 find the DELTA values, the differences between the FROM and TO coordinates.

Lines 120-130 reset the X and Y accumulator variables (XACC and YACC) to zero. These variables are used as counters to tell the computer when to increment the X and Y coordinates.

CASADAPTER
"IT REALLY WORKS!"

CASADAPTER is a cassette interface that allows you use your own cassette recorder or stereo with the Atari 400/800/1200®. **CASADAPTER** will handle motor control, audio and data channels.

$34.95

48K **RAM BOARD FOR: THE ATARI 400®.**

$115.00

SCP **SAR-AN** COMPUTER PRODUCTS

12 Scamridge Curve
Buffalo, New York 14221
(716) 632-3441

Dealer Inquiries Invited
*Product of Gemini Software

Add $2.50 shipping.
Send check or money order
C.O.D. accepted
New York State Residents
add 7% tax
®Atari is a trademark of
Atari, Inc.

***MAGIC DUMP** is a screen dump utility that allows you to dump a Hi-Resolution graphics picture to a printer in a variety of different sizes. **MAGIC DUMP** is used in the RIGHT hand cartridge slot, so it is always ready to use.

MAGIC DUMP will work with all Epson printers and Gemini printers, the Centronics 739 printer, and the Prowriter or N.E.C. printers. **$59.95**

***GTIA DRAW** is a drawing program that uses all the features of Atari's® new GTIA chip. **GTIA DRAW** will give you three extra graphics modes, Mode 9 (16 luminances and one color), Mode 10 (8 luminances and color), or Mode 11 (16 colors and one luminance).

GTIA DRAW will allow you to:
- Add text to pictures
- Blank horizontal or vertical lines
- Shift the screen in any direction
- 'ZOOM' in on certain areas of a picture
- Fill portions of a screen
- Work on two different screens simultaneously

$49.95

***THE DRUMESISER** is a unique sound synthesizing tool that allows the creation of sounds such as a drum, piano, organ, harpsichord, or electronic synthesizers. The different types of sounds created are limited by the user's imagination and ambition.

THE DRUMESISER comes with an Editor, which allows you to create your own sounds, a Player, which will allow you to play the different instruments, and the Memory Options, which make it possible for you to playback any sounds that you have recorded.

$49.95

CIRCLE #136 ON READER SERVICE CARD.

XLENT
Software
Presents:
A MegaHit!

MegaFont
The Complete Program Lister
and Graphics Dumper

By Randy Dellinger and Richard Rognlie
(XLENT's Star Programmers)

Allows you to list ALL control and INVERSE characters to your printer in any of NINE fonts provided, or create your own with any font editor.

Dump your GRAPHICS 8 screens (created with Micro Painter, etc.) to the printer in any of THREE sizes (quarter, half, and full page.)

For use with Prowriter, NEC or Epson w/ Graftrax printers
48K disk only $19.95

★ ★ ★ ★ ★ ★ ★ ★ ★

COMPUTER
ASSISTED
MATH
PROGRAM

+ − × ÷

by
Johnny
Masuda

Originally written for Johnny's daughter, this is the BEST math drill program written for the ATARI. Easily played by elementary students. Multiple skill levels make it a challenge to older children. Use your keyboard or joystick.

32K disk or tape only $19.95

★ ★ ★ ★ ★ ★ ★ ★ ★

Mode Mixer 1&2
by Margie Bliss and Jerry Kwit

Lets you combine all ATARI graphics modes to create custom display lists. Mode Mixer 1 generates ERROR-FREE BASIC code for insertion into your programs. Mode Mixer 2 generates the actual Display List. Includes FREE strategic warship game. Excellent tutorial style documentation.

48K disk only $19.95

Add $2.00 for Shipping and Handling
Add $1.50 for C.O.D. orders
VA residents add 4% sales tax

Send Check or Money Order to:
XLENT Software
P.O. Box 5228
Springfield, VA 22150

24-Hour Phone: (703) 644-8881

Dealer Inquiries Welcome

CIRCLE #137 ON READER SERVICE CARD.

Line 140 sets COUNT to the larger of DELTAX and DELTAY. This tells the computer which axis (X or Y) has farthest to go. Count is decremented each time a point is plotted, and when it reaches zero, the line is complete.

Line 150 sets EPOINT (endpoint) to the value of COUNT. EPOINT is then used as the limit value for XACC and YACC.

Lines 160-170 set either XACC or YACC to a starting value of EPOINT/2. If the X axis has farthest to go, YACC is set, otherwise, the computer initializes XACC.

Line 180 is the start of the plot loop. If the COUNT variable is zero, the line has reached the endpoint and the draw is finished.

Line 190 adds the DELTAX value to the X accumulator.

Line 200 checks to see if the X accumulator value has exceeded the EPOINT limit. If it has, the EPOINT value is subtracted from the accumulator and the X coordinate is moved in the proper direction (contained in XD).

Lines 210-220 perform the functions of Lines 190-200, but for the Y coordinate.

Line 230 plots the new coordinate on the screen.

Line 240 decrements the COUNT variable. As noted before, when this variable reaches zero, the draw is complete.

Line 250 loops back to Line 180 for the next iteration of the draw loop.

Line 260 is simply the end message for the routine.

When this program is executed, you will see that this algorithm produces the best lines of the three vector routines presented here. It is slightly slower than the other methods, but that's the price you pay for the accuracy.

Follow the leader.

Up until now, poor Blaster Bob hasn't had much of a chance against the evil electrified robot — he can't move! In order to present a more fair challenge, we should at least let him move around, right?

This problem presents a slightly different challenge to our vector routines. They must calculate a new vector to the target point for each movement! You've probably guessed that this will usually be slower than drawing lines to stationary targets, and you're right.

Figures 5, 6 and 7 show "follower" routines using each of the vector methods discussed. Each program allows you to move the target around on the screen with a joystick plugged into port 1. The program then tries to hit the target. To speed the routines up, there is no coordinate limiting, so be careful not to run the target off the screen. Let's look at the advantages and disadvantages of each.

Figure 5.

```
10 REM *** FOLLOWER (X-Y MATCHING) ***
20 REM
30 GRAPHICS 6:COLOR 1
40 DIM XS(15),YS(15):FOR I=1 TO 15:REA
D X,Y:XS(I)=X:YS(I)=Y:NEXT I
50 DATA 0,0,0,0,0,0,0,0,1,1,1,-1,1,0,0
,0,-1,1,-1,-1,-1,0,0,0,0,1,0,-1,0,0
60 FX=0:FY=0
70 TX=80:TY=40
80 STIK=STICK(0)
90 TX=TX+XS(STIK)
100 TY=TY+YS(STIK)
110 PLOT TX,TY
120 XV=SGN(TX-FX)
130 YV=SGN(TY-FY)
140 IF  NOT XV AND  NOT YV THEN 190
150 FX=FX+XV
160 FY=FY+YV
170 PLOT FX,FY
180 GOTO 80
190 ? "GOTCHA!":END
```

•

CHECKSUM DATA
(See p. 30)

```
10 DATA 691,253,28,350,327,808,992,514
,642,729,181,347,357,548,833,7600
160 DATA 841,143,518,454,1956
```

•

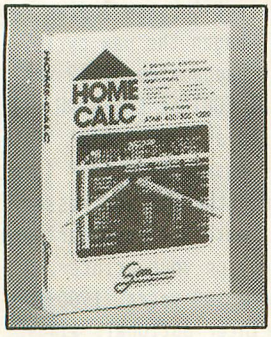
Figure 6.

```
10 REM *** FOLLOWER (VECTOR 1) ***
20 REM
30 GRAPHICS 6:COLOR 1
40 DIM XS(15),YS(15):FOR I=1 TO 15:REA
D X,Y:XS(I)=X:YS(I)=Y:NEXT I
50 DATA 0,0,0,0,0,0,0,0,1,1,1,-1,1,0,0
,0,-1,1,-1,-1,-1,0,0,0,0,1,0,-1,0,0
60 FX=0:FY=0
70 TX=80:TY=40
80 STIK=STICK(0)
90 TX=TX+XS(STIK)
100 TY=TY+YS(STIK)
110 PLOT TX,TY
120 XD=SGN(TX-FX)
130 YD=SGN(TY-FY)
140 DELTAX=ABS(TX-FX)
150 DELTAY=ABS(TY-FY)
160 IF DELTAX>1 OR DELTAY>1 THEN DELTA
X=DELTAX/2:DELTAY=DELTAY/2:GOTO 160
170 XV=DELTAX*XD
180 YV=DELTAY*YD
190 IF INT(FX)=INT(TX) AND INT(FY)=INT
(TY) THEN 240
200 FX=FX+XV
210 FY=FY+YV
220 PLOT FX,FY
230 GOTO 80
240 ? "GOTCHA!":END
```

•

(Continued next page.)

Why spend $200
for a spreadsheet when you can buy
HOME-CALC for under $40
Available for all Atari computers



*Car Expenses
*Budgets
*Finances
*Income Tax
*Investments
*Sports Records
*Payment Schedules
*Stock Portfolios
*Expense Accounts

Cassette - $29.95
Diskette - $39.95

Home-Calc is a powerful, inexpensive spreadsheet for home and business calculations. You can enter numbers, labels, or fomulas in each cell. Home-Calc allows you to add, subtract, multiply, divide, use the exponential guide, sum, and recalculate. The diskette version also features a replicate command. Maximum spreadsheet size is 510 cells for the 48k diskette version. (Disk version requires 24k memory; cassette requires 16k memory.)

See your dealer or order direct

Phone orders—(215) 825-4250

HOME-CALC cassette—$29.95	_____
HOME-CALC diskette—$39.95	_____
Shipping and handling	$2.00
C.O.D. and credit card add $1.50	_____
6% sales tax (PA, NJ residents only)	_____
TOTAL	_____

Name _____
Address _____
City _____ State _____ Zip _____
Phone _____

☐ MC# _____
☐ VISA# _____
Exp. Date _____

Atari 400/800/1200 are trademarks of Atari, Inc.

Dealer, Distributor, and Representative inquiries invited

CIRCLE #138 ON READER SERVICE CARD.

fresh ideas in Software . . .

Sim
COMPUTER PRODUCTS INC.
1100 E. HECTOR ST.
WHITEMARSH, PA. 19428

CHECKSUM DATA
(See p. 30)

```
10 DATA 420,253,28,350,327,808,992,514
,642,729,181,329,339,185,196,6293
160 DATA 214,580,589,779,820,828,130,5
05,441,4886
```

●

Figure 7.

```
10 REM *** FOLLOWER (VECTOR 2) ***
20 REM
30 GRAPHICS 6:COLOR 1
40 DIM X5(15),Y5(15):FOR I=1 TO 15:REA
D X,Y:X5(I)=X:Y5(I)=Y:NEXT I
50 DATA 0,0,0,0,0,0,0,0,1,1,1,-1,1,0,0
,0,-1,1,-1,-1,0,0,0,0,1,0,-1,0,0
60 FX=0:FY=0
70 TX=80:TY=40
80 STIK=STICK(0)
90 TX=TX+X5(STIK)
100 TY=TY+Y5(STIK)
110 PLOT TX,TY
120 XD=SGN(TX-FX)
130 YD=SGN(TY-FY)
140 DELTAX=ABS(TX-FX)
150 DELTAY=ABS(TY-FY)
160 XACC=0
170 YACC=0
180 COUNT=DELTAX:IF DELTAY>DELTAX THEN
 COUNT=DELTAY
190 EPOINT=COUNT
200 IF COUNT=DELTAX THEN YACC=EPOINT/2
:GOTO 220
210 XACC=EPOINT/2
220 IF COUNT=0 THEN 310
230 XACC=XACC+DELTAX
240 IF XACC>=EPOINT THEN XACC=XACC-EPO
INT:FX=FX+XD
250 YACC=YACC+DELTAY
260 IF YACC>=EPOINT THEN YACC=YACC-EPO
INT:FY=FY+YD
270 PLOT FX,FY
280 COUNT=COUNT-1
290 IF STICK(0)=15 THEN 220
300 GOTO 80
310 ? "GOTCHA!":END
```

●

CHECKSUM DATA
(See p. 30)

```
10 DATA 423,253,28,350,327,808,992,514
,642,729,181,329,339,185,196,6296
160 DATA 658,664,414,665,307,404,934,8
58,36,872,58,145,430,478,498,7421
310 DATA 434,434
```

●

Figure 5 is the X-Y matching version of the follower. It is the fastest of the three followers, but like its simple vector version, produces very poor lines. There's not much that can be done to improve this algorithm. If it's speed you want, and you don't mind the erratic behavior, this routine is fine.

Figure 6 uses the first vector method to follow the target. In my opinion, this routine produces the best results as far as the follower's path is concerned. When you run the program, you will see that the following point always moves smoothly, with nice curves. This algorithm looks "smartest" in programs.

The major drawback with this routine is speed. The farther away the target is, the slower the calculations. As the follower gets closer to the target, however, the routine speeds up considerably.

Figure 7 uses the system vector routines for the follower. This is a very interesting situation. Due to the iterative nature of this vector algorithm, the follower becomes confused when the target moves. When the target stops, the follower can "lock in" on a straight course toward the target. Line 290 checks the stick to see if the target is in motion. If it is, the program must recalculate the vector (GOTO 80).

Using vectors.

You can use these vector and follower routines in your own games. The primary vector calculation routines will work in any graphics mode. All you have to do is give them the "FROM" and "TO" coordinates.

The routines shown here leave the follower's path on the screen so that you can see how each algorithm works. They can be easily modified to only show a single pixel (or redefined character).

By using your imagination, you can come up with some challenging (and fun) games in BASIC with these routines. □

Ziza Presents

THE
CHINESE FORTUNE
CALENDAR

Are people born in the YEAR OF THE RAT compatible with those born in the YEAR OF THE DRAGON?

On February 2, 1984, the ancient Chinese Zodiac renewed itself. The YEAR OF THE RAT, began the new cycle. To commemorate this renewal (it happens every 12 yrs), ZIZA has released THE CHINESE FORTUNE CALENDAR. It reveals the animal which ruled at YOUR BIRTH! Enter your Birth date and get your answer PLUS your fortune! The fortune reveals your strengths and weaknesses. It also suggests those who will make the best partners! Press a key and you get a copy on your printer. A great idea for parties! Covers 100 years, 1900-2000. Atari only. 48K Diskette, printer optional. $19.95.

羊 兔 犬 牛

ZIZA PRESENTS, INC.
2257 Independence
Ann Arbor, MI 48104
313-973-0299

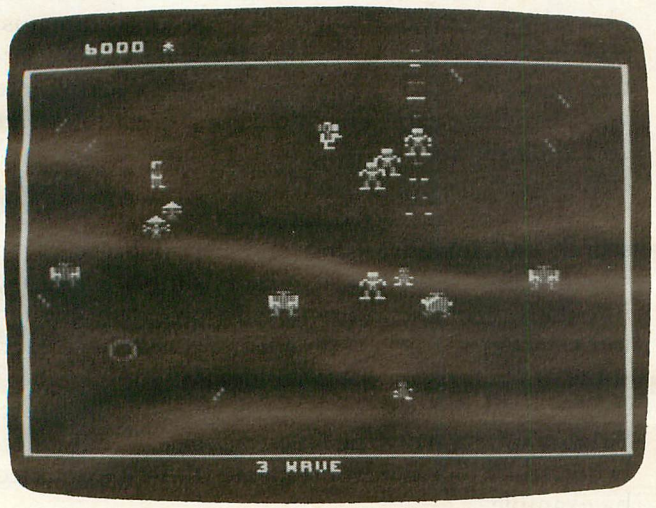
Dealer Inquiries Invited
Check or money order. Add $1.50 for shipping.
MI residents add 4% tax. Atari TM Atari Inc.

CIRCLE #139 ON READER SERVICE CARD.

ROBOTRON: 2084
ATARI, Inc.
P.O. Box 61657
San Jose, California 94086
16K Cartridge $49.95

by Kyle Peacock

The year is 2084. Man's technological achievements in the field of robotics have produced a new form of mechanized menace: the Robotron. This new series is so advanced, man is inferior to his own creation. Guided by their infallible logic, the Robotrons have concluded that man is inefficient and must be destroyed.



Robotron.

If the above scenario sounds like the story line to a *2001* sequel, guess again. **Robotron: 2084** has arrived on the Atari home computer.

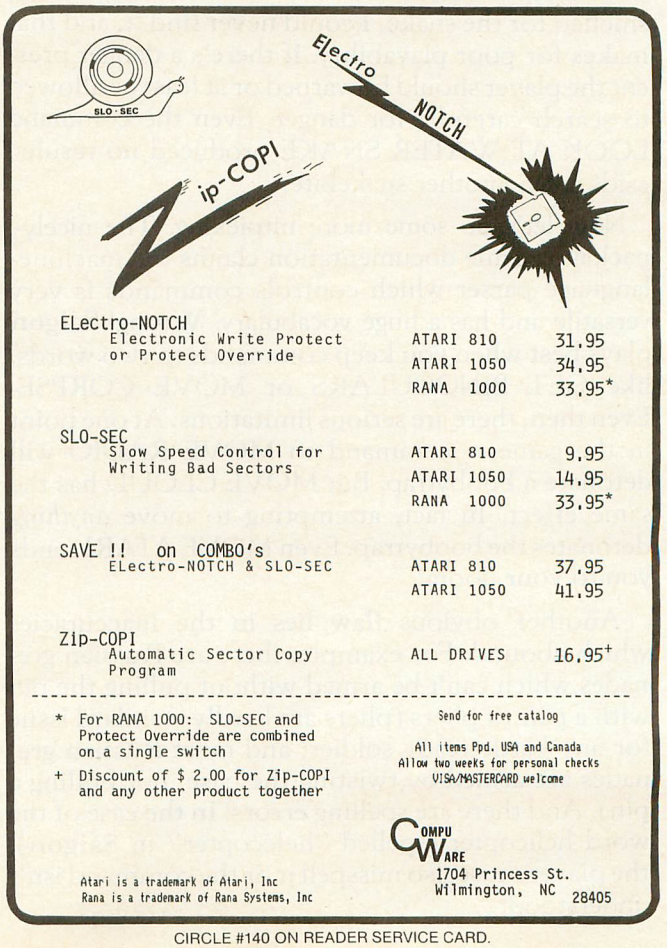
The Robotrons have all but eliminated mankind. Other than yourself, only one human family remains. Due to a genetic engineering error (a fancy name for "birth defect"), you possess superhuman powers. You have the capability to skrag the Robotrons and save the last human family.

Robotron is an extremely fast paced shoot-em-up. Your little poindexter character runs around the screen, dodging enemies and vaporizing anything that moves. The Robotrons have an instinct for their survival and your demise. Contact with any of the various species causes you to lose a life. Fortunately, you receive five lives initially and an extra life every 20,000 points.

One good aspect of **Robotron** is the different types of attackers. Ground Roving Unit Network Terminator (GRUNT) Robotrons are the most abundant. They are designed to eliminate you on contact. Hulk Robotrons march around in pursuit of the last human family. They replace Mommy, Daddy and Mikey with a set of skull and crossbones. Giant Brain Robotrons also prey on the family members. They can reprogram people into deadly Progs. A Prog is a sort of synthetic Robotron, just as lethal as its mechanical counterpart. Other menaces include Enforcer Embryos which release Enforcers, Cubic Quarks which continually bounce off walls, and Torture Tanks which move around on rotating threads.

One feature of **Robotron** that brings it closer to its arcade counterpart is the option for two joysticks. In this mode, one joystick controls the direction of travel, the other direction of fire. Atari designed the game packaging to accomodate this feature. If you don't have two joysticks, though, you can still play Robotron, so don't let this deter you.

Overall, **Robotron: 2084** isn't a bad game. I was hoping Atari would design a little better playability into the home computer version. I rarely played this game in the arcades. It was always in a class with **Defender** and **Stargate** as either having too many buttons, or too much "stuff" going on at once. Atari's **Robotron** rivals the arcade version in too many respects. It's just as tough and unplayable as the real thing. □



ELectro-NOTCH
Electronic Write Protect or Protect Override
ATARI 810 31.95
ATARI 1050 34.95
RANA 1000 33.95*

SLO-SEC
Slow Speed Control for Writing Bad Sectors
ATARI 810 9.95
ATARI 1050 14.95
RANA 1000 33.95*

SAVE !! on COMBO's
ELectro-NOTCH & SLO-SEC
ATARI 810 37.95
ATARI 1050 41.95

Zip-COPI
Automatic Sector Copy Program
ALL DRIVES 16.95+

* For RANA 1000: SLO-SEC and Protect Override are combined on a single switch

\+ Discount of $ 2.00 for Zip-COPI and any other product together

Send for free catalog

All items Ppd. USA and Canada
Allow two weeks for personal checks
VISA/MASTERCARD welcome

Atari is a trademark of Atari, Inc
Rana is a trademark of Rana Systems, Inc

COMPU WARE
1704 Princess St.
Wilmington, NC 28405

CIRCLE #140 ON READER SERVICE CARD.

SAIGON: THE FINAL DAYS
by Jyym Pearson
with Robyn Pearson and Norman Sailer
ADVENTURE INTERNATIONAL
P.O. Box 3435
Longwood, Florida 32750
16K Cassette $24.95

by Ray Berube

Hurray! Another 16K adventure for my memory-poor Atari 600XL. Another chance to spend a weekend challenging my grey cells while my wife repeatedly informs me that "Dinner is ready! — Getting cold!" and finally "Get your own supper!"

The setting of **Saigon: The Final Days** is the jungle and city environs of Saigon just prior to the American evacuation. The player is a prisoner of war who, with a little luck, will make his way out of the jungle, through the city and into the American compound in time to be evacuated. In other words, **Saigon** is a Green Beret primer.

My early encounters with a water snake, machine gun nest and boobytraps were fatal. I found myself back at the start — each time a little more frustrated. I was tired of being bitten by that snake! Then I discovered the first several flaws in the game. No matter how hard I looked, listened, touched or smelled for the snake, I could never find it, and that makes for poor playability. If there's a danger present the player should be warned or at least be allowed to search carefully for danger. Even the command LOOK AT WATER SNAKE produced no results, aside from another snakebite.

Now let's do some more nitpicking. The nicely-packaged game documentation claims the machine-language parser which controls commands is very versatile and has a huge vocabulary. Wrong! **Saigon** plays best when you keep commands to two words, like GET BINOCULARS or MOVE CORPSE. Even then, there are serious limitations. At one point in the game, a command to MOVE RADIO will detonate a boobytrap. But MOVE CLOUD has the same effect. In fact, attempting to move *anything* detonates the boobytrap. Even MOVE ATARI sends you to your doom!

Another obvious flaw lies in the inaccuracies which abound. For example, there are Russian grenades which can't be armed without pulling the pin with a pair of pliers (pliers are hardly standard issue for any battle-ready soldier; and most Russian grenades are armed by twisting the base, not pulling a pin). And there are spelling errors! In the case of the word helicopter (spelled "helecopter" in **Saigon**), the player must also misspell it or the command isn't understood!

Saigon is also not very imaginative. Most of the obstacles are physical ones: crossing jungle streams, opening locked doors, crawling through tunnels, getting tanks to start. There's very little to stimulate your thinking cap. The inclusion of a hint sheet coded by number is the game's greatest intellectual challenge.

The key to any good adventure is the ability to search and uncover clues and useful objects. An adventure without this feature is simply a journey into frustration. **Saigon** doesn't give you enough information to make intelligent decisions. In fact, some of the most important clues and tools are held deliberately out of reach until you die. In short, **Saigon** suffers from the *killer syndrome*. What exactly does the killer syndrome mean? It's best explained by example.

At one point in the game, the player survives a helicopter crash but finds himself paralyzed. Within the crashed helicopter are crucial clues and tools you need to advance in the adventure. But the game doesn't allow you to find these things until you are killed at the next obstacle! After your resurrection and return to the site of the crash, you're suddenly able to discover the clues you need. No command of any kind will let you access this vital information the first time through. Backtracking past this sequence doesn't work either. **Saigon** requires your death to reveal information vital to your survival. That's the killer syndrome, and it's an exercise in frustration!

Well, after all this, is **Saigon** worth your money? If you're one of the many users whose Atari has only 16K (as all 600XL owners are now, but hopefully not for long), there are very few text adventures available. Even with its flaws, I recommend you buy the game. Just have a good supply of patience on hand. However, if your Atari belongs to the smart set of 48K and up, then I can't recommend **Saigon**. Invest a little more money and buy an Infocom adventure, or even one of the original Scott Adams titles. You'll enjoy your investment rather than railing at your monitor. □

Books&Software

BOOKS for ATARI 400/600XL/800XL 1200XL

ATARI BASIC – Learning by Using
An excellent book for the beginner. Many short programs and learning exercises. All important features of the ATARI computers are described (screen drawings, special sounds, keys, paddles, joysticks, specialized screen routines, graphics, sound applications, peeks, pokes, and special stuff).
Order-No. 164 $7.95

Games for the ATARI Computer
This book describes advanced programming techniques like player-missile-graphics and use of the hardware-registers. Contains many ready to run programs in BASIC and one called GUNFIGHT in machine language.
Order-No. 162 $7.95

How to program your ATARI in 6502 Machine Language
Introduction to machine language for the BASIC programmer.
Order-No. 169 $9.95

FORTH on the ATARI – Learning by Using
Introduction, programs, applications, learning exercises
Order-No. 170 $7.95

All programs from book No. 170 on disk.
Order-No. 7319 $22.00 only!

A Look into the Future–ASTROLOGY on your ATARI 800. How to calculate your own horoscope. Including listing of the program.
Order-No. 171 $9.95

HACKERBOOK for your Atari computer
Tips+tricks-Very important subroutines in 6502 machine language. How to make bootable cassettes, disks, and EPROMs. Complete construction article and software on how to build an EPROM burner.
Order-No. 172 $9.95

SMALL BUSINESS SOFTWARE FOR ALL ATARI COMPUTERS

SUPERMAIL (500 addresses on 1 disk) Completely written in FORTH. Comes on autoboot disk. No cartridge, no DOS, no FORTH language required.
Order-No. 7312 $49.00

SUPERINVENTORY (1000 items p.disk) Completely written in FORTH. Same as above. (Disk only)
Order-No. 7320 $49.00

BUSIPACK-1 (written in FORTH). Complete order entry, inventory, mailing and invoicing. (Disk only).
Order-No. 7313 $98.00

Microcomputer Hardware Handbook
Order-No. 29 $ 14.95

ATCASH
Convert your ATARI 800 into a powerful cash register. (Disk only).
Order-No. 7303 $49.95

Invoicing program in BASIC
Order-No. 7201 (C) $29.95
Order-No. 7200 (D) $39.95

Mailing List in BASIC
Order-No. 7212 (C) $19.95
Order-No. 7213 (D) $24.95

Inventory control in BASIC
Order-No. 7214 (C) $19.95
Order-No. 7215 (D) $24.95

SOFTWARE IN MACHINE LANGUAGE FOR ATARI

ATMONA-1
Machine language monitor.
Order-No. 7022 (C) $19.95

ATMONA-2
This is a tracer (debugger) that lets you explore the ATARI RAM/ROM area. You can stop at previously selected address, opcode, or operand. Also very valuable in understanding the microprocessor. At each stop, all registers of the CPU may be changed. Includes Atmona-1.
Order-No. 7049 cassette $49.95
Order-No. 7050 disk $54.00

ATMAS
Macro-Assembler for ATARI-800/48K. One of the most powerful editor assemblers on the market. Versatile editor with scrolling. Up to 17k of source-code. Very fast, translates 5k source-code in about 5 seconds. Source code can be saved on disk or cassette (Incl. Atmona-1).
Order-No. 7099 disk $89.00
Order-No. 7999 cartridge $129.00

ATMS APPLICATION DISK
All programs and machine language subroutines from Book No.169 on disk.
Order-No. 7311 $20.00

ATAS
Same as ATMAS but without macro-capability. (32K and 48K RAM)
Order-No. 7098 $49.95

ATEXT-1
This wordprocessor is an excellent buy for your money. It features screen oriented editing, scrolling, string search (even nested), left and right margin justification. Over 30 commands. Text can be saved on disk or cassette.
Order-No. 7210 cassette $29.95
Order-No. 7216 disk $34.95
Order-No. 7217 cartridge $69.00

FORTH for the ATARI
FORTH from ELCOMP PUBLISHING, Inc. is an extended Fig-Forth-version, Editor and I/O package included. Utility package includes decompiler, sector copy, Hex-dump (ASCII), ATARI filehandling, total graphic and sound, joystick program and player missile. Extremely powerful!
Order-No. 7055(D) reg. $99.00 sale $39.95

Floating point package with trigonometric functions (0 – 90°).
Order-No. 7230 disk $29.95

Learn FORTH
A subset of Fig-Forth for the beginner. On disk (32k RAM) or cass. (16k RAM).
Order-No. 7053 reg. 79.00 sale $19.95

HARDWARE-ADD-ONs for ATARI

PRINTER INTERFACE
This construction article comes with printed circuit board and software. You can use the EPSON printer without the ATARI printer interface. (Works with gameport 3 and 4).
Order-No. 7211 $19.95

300 Baud serial interface (RS232+5V)
Software with connector and construction article.
Order-No. 7291 $19.95

EPROM BURNER for ATARI 400/800 KIT. Printed circuit board incl. software and extensive construction article.
Order-No. 7292 $49.00

EPROM BOARD (CARTRIDGE)
Holds two 4k EPROMs (2532). EPROMs not included.
Order-No. 7043 $29.95

EPROM BOARD KIT
Same as above but bare board only with description.
Order-No. 7224 $14.95

Astrology and Biorhythm for ATARI
Order-Nol 7223 D/C $29.95

Birth control with the Atari(Knaus Ogino)
Order-No. 7222 disk only! $29.95

The APPLE in your Hand, BRAND NEW! Book includes introduction to 6502 Machine Language and FORTH. BASIC programs never published before!
Order-No. 178 (200 pages) $12.95

CP/M –MBASIC and the OSBORNE computer
Business Applications, compl. listings of mailing list, datablock, inventory, invoicing and more.
Order-No. 177 $9.95

SUPERSOFTWARE f.the Commodore-64
BLIZTEXT! – The best wordprocessor for the C-64 in the whole universe. Includes terminal software for electronic mail.
Order-No. 4965 (62 pages manual) $89.00

MACROFIRE – Editor/Assembler for the C-64
The best macroassembler you can buy!
Order-No. 4963 $89.00

SUPERBOOKS for your C-64

The Great Book of Games, Vol. I
64 programs for the Commodore-64.
Order-No. 182 $9.95
Programs from this book on disk $19.95

MORE ON THE SIXTYFOUR
Tips, tricks, hints, very important subroutines.
Order-No. 182 $9.95
Programs from this book on disk $19.95

How to program in 6502 Machine Language on your C-64
Order-No. 184 $12.95

Commodore-64 Tune-up, Vol. I
How to expand your C-64.
Order-No. 185 $12.95

Small Business Programs for the C-64
Order-No. 186 $12.95

HARDWARE ADD-ONs for your C-64

Parallel printer interface, KIT
Order-No. 4990 $19.95

Universal Experimenter Board
Order-No. 4970 $9.95

Expansion Board (holds 4 exp. boards)
Order-No. 4992 $29.95

BOOKS, SOFTWARE, ADD-ONs for VIC-20, APPLE II, OSBORNE, TIMEX + OSI computers

Tricks for VICs (Book)
Order-No. 176 $9.95

Universal Experim.board for the VIC-20
(Save money with this great board). This board plugs right into the expansion slot of the VIC-20.
Order-No. 4844 reg. $19.95 sale $9.95

Programming in BASIC and machine language on the ZX-81 (82) TIMEX1000
Order-No. 174 (book) $9.95

The Custom APPLE + Other Mysteries
A complete guide to customizing the APPLE software and hardware.
Order-No. 680 $24.95
We also stock the boards which are used in the book No. 680 (bareboards).

HOFACKER

Elcomp Publishing, Inc, 53 Redrock Lane, Pomona CA 91766, Phone: (714) 623-8314, Telex: 29 81 91

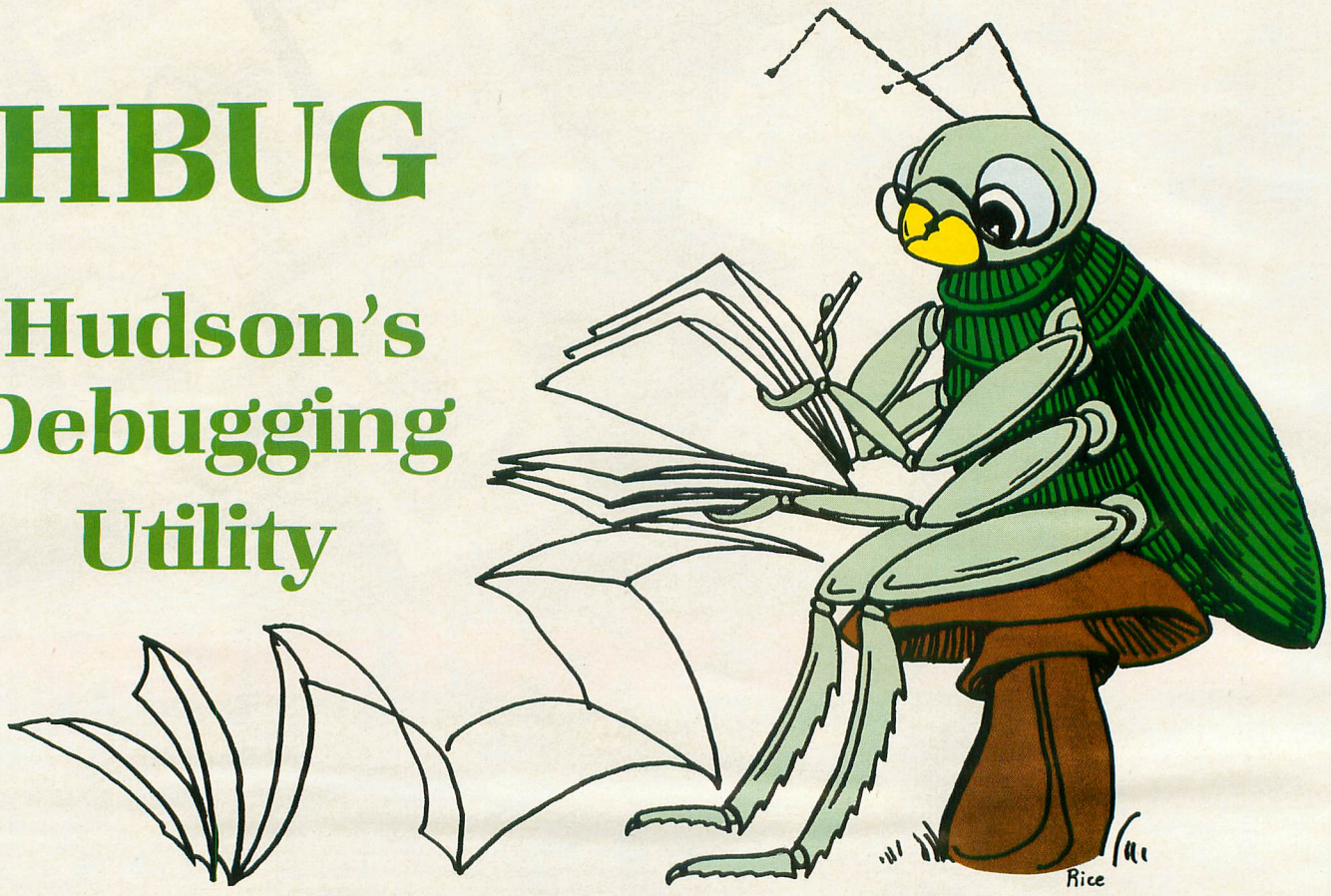
PAYMENT: Check, money order, VISA, MASTER CARD, Eurocheck, ACCESS, Interbank
Prepaid orders add $3.50 for shipping (USA)
$5.00 handling for C.O.D.
All orders outside USA: add 15% shipping, California residents add 6.5 % sales tax.

ATARI is a reg. trademark of ATARI Inc., VIC-20 + Commodore-64 are trademarks of Commodore Business Machines, APPLE is a reg. trademark of APPLE Computer, Inc., CP/M is a reg. tradem. of Digital Research

CIRCLE #141 ON READER SERVICE CARD.

HBUG

Hudson's Debugging Utility



16K Disk

by Tom Hudson

HBUG is a simple debugging utility designed for readers of **ANALOG**'s **Boot Camp** column who do not own the Atari Assembler Editor cartridge. The program's syntax is the same as DEBUG, allowing **Boot Camp** readers to execute the examples and observe the results.

Why HBUG?

Why write another debug package? I'm sure quite a few readers own the **MAC/65** assembler from Optimized Systems Software. This is a fine package (I use it myself, and recommend it highly). As you may know, **MAC/65** comes with its own debug package, **BUG/65**. **BUG/65** is a very powerful debugging tool, but it has a couple of shortcomings.

First, the only way to stop a program that is running in an infinite loop is to press SYSTEM RESET. As a result, there is no way to determine register contents or where the program was looping.

Second, many of **Boot Camp**'s illustrative programs rely on the use of the BRK instruction to stop execution at selected locations. **BUG/65** doesn't recognize the BRK instruction, and any attempt to perform a BRK locks up the system. **BUG/65** allows you to set breakpoints, but these are generally cumbersome to use.

HBUG overcomes these shortcomings, making life easier for those new to assembly language.

Typing it in.

Before you start typing anything in, take a look at the listings accompanying this article.

Listing 1 is the main data and data checking routine, written in Atari BASIC. This program will create a file on your disk called HBUG.COM.

Listing 2 is the assembly source code for HBUG, written with OSS's **MAC/65**. You don't have to type this in to use **HBUG**, but the adventuresome types out there (you know who you are) may like the assembly typing experience.

Follow the instructions below to create the HBUG.COM file.

1. Type **Listing 1** into your Atari and verify your typing with **D:CHECK2** (see page 30).

2. Type RUN and press RETURN. The program will begin checking the data lines, printing the line numbers as it goes. You will be alerted if the program finds any problems. Fix any incorrect lines and re-RUN the program as necessary until all errors are eliminated.

3. When all DATA lines are correct, you will be

prompted to INSERT DISK, PRESS RETURN. Place a disk in drive #1 and press RETURN. The message WRITING FILE will appear, and the computer will create the HBUG.COM file, printing each line number as it goes. When the READY prompt appears, you're ready to use **HBUG**. Make sure the BASIC program has been saved under a different filename before continuing.

Usually, you'll want to load the object code of the program to be executed before loading **HBUG**. If the program is already in memory, simply skip ahead to the **HBUG** loading instructions.

If you want to load the object code in Atari DOS, go to the DOS menu screen and type:

```
L RETURN
Filename.OBJ RETURN
```

If you're using OS/A+, type:

```
LOA Filename.OBJ RETURN
```

To load **HBUG** with Atari DOS, go to the DOS menu screen and type:

```
L RETURN
HBUG.COM RETURN
```

HBUG will load and run automatically.

To load **HBUG** with OS/A+, go to the OS/A+ input screen and type:

```
HBUG RETURN
```

HBUG will load and run automatically.

Up and running.

When running, **HBUG** supports 6 commands, which are a subset of the Atari assembler editor DEBUG program's commands. These commands are:

```
DOS      (RETURN TO DOS)
DR       (DISPLAY REGISTER CONTENTS)
CR<      (CHANGE REGISTER CONTENTS)
D        (DISPLAY MEMORY CONTENTS)
C        (CHANGE MEMORY CONTENTS)
G        (EXECUTE AT ADDRESS)
```

These commands are the most important ones, as far as we're concerned. I would have liked to include the "disassemble memory" command, but it would have increased the size of the program considerably. The use of these commands is described below.

If any invalid commands are entered, **HBUG** will "beep" at you and show the line you entered with the invalid character highlighted in inverse video.

DOS.

This command is simple — it transfers control to the disk operating system. Once there, you can perform any function allowed by DOS. If you're using OS/A+, typing RUN restarts **HBUG**. If you're using Atari DOS, you'll have to reload **HBUG** to start it up again.

Change registers.

You can set the registers (A, X, Y, Processor status [P], and Stack pointer [S]) to any values you like before executing a test program. Be careful about changing the P and S registers, since invalid values placed here can cause a system crash.

The format of the Change Registers command is:

```
CR< Ra, Rx, Ry, Rp, Rs

Where Ra = Accumulator
      Rx = X register
      Ry = Y register
      Rp = Processor status register
      Rs = Stack pointer
```

The values entered for the registers must be hexadecimal, from 0 through FF.

Figure 1 shows several forms of the CR command and their effect on the registers.

CR<1F,6E	Changes the Accumulator to $1F and the X register to $6E.
CR<,,,4C	Changes the P register to $4C. All other registers are unchanged.

Figure 1.

You can see that by inserting commas, you can leave certain registers unchanged while altering others.

Display registers.

This command is the companion of the CR command. Its format is:

```
DR
```

Whenever you want to see what the current settings of the user registers, simply type DR and press return. The computer will display something like this:

```
A=03 X=32 Y=01 P=33 S=E2
```

Display memory.

HBUG allows you to display the contents of any memory byte or group of bytes. The formats of this command are:

```
FORMAT 1: Dxxxx

FORMAT 2: D

FORMAT 3: Dxxxx,yyyy
```

Format 1 displays eight bytes of memory starting at address xxxx.

Format 2 displays the eight bytes of memory starting at the last specified address plus 8.

Format 3 displays the memory block from address xxxx to yyyy. The display can be paused by pressing CTRL-1 and stopped by pressing the BREAK key.

(Continued next page.)

This command differs from the cartridge command in that the ATASCII characters corresponding to the displayed bytes are shown to the right of the hex values.

Change memory.

You can change any memory byte in system RAM with the Change Memory command. Its format is:

```
Cxxxx< byte1, byte2, ... , byte n
```

This command changes the memory starting at address xxxx to the hex values following the "<" symbol. Like the CR command, the C command allows you to skip memory locations by using commas to indicate skipped bytes. **Figure 2** shows several examples of the Change memory command.

C 5000<1F,45,DE	Change location $5000 to $1F, $5001 to $45, and $5002 to $DE.
C 600<20,,,,F6	Change location $0600 to $20 and location $0604 to $F6. Leave all other locations as is.

Figure 2.

Use care when changing memory bytes! Be sure you're using the address you want, because careless changes could wipe out vital system data, causing a fatal lockup. Also, take care not to change any locations from $2000-26FF, since this is where HBUG is located.

Execute user program.

This command is what **HBUG** is all about. It allows you to execute assembly code you have placed in memory. Its format is:

```
Gxxxx
```

When the G command is entered, the computer (1) changes the 6502 registers to the values contained in the user registers (see CR and DR), and (2) jumps to the address specified by xxxx.

The user program will continue to execute until it is stopped by the BREAK key or tries to execute a 6502 BRK instruction. If either of these events occurs, the program will be interrupted, **HBUG** will place the 6502 register contents in the user registers and **HBUG** will take control. The location where the program was executing and the user registers will be displayed in the following format:

```
40F5    A=1B X=6F Y=2D P=04 S=EA
```

HBUG will perform as advertised unless the user program violates the **HBUG** rules, outlined below.

HBUG no-nos.

1. The user program should not change any memory locations from $2000-26FF. This is where **HBUG** is located, and any changes to this memory could send **HBUG**, your program, and the system off to never-never land.

2. The user program should not use the deferred vertical blank vector or shut off vertical blank interrupts. **HBUG** uses deferred vertical blank to detect the BREAK key during user program execution. Don't steal this vector unless you want to see **HBUG** crippled. If you must use vertical blank interrupts, try using the immediate vertical blank.

3. Don't touch the VBREAK vector ($0206-0207). These locations are used to detect the execution of a 6502 BRK instruction, and alteration of these bytes will once again cripple **HBUG**.

If you follow these instructions and avoid careless alteration of **HBUG**'s memory, **HBUG** should help you debug the programs that **BUG/65** couldn't. □

BASIC Listing.

```
10 REM *** HBUG ***
20 DATA 0,1,2,3,4,5,6,7,8,9,0,0,0,0,0,
0,0,10,11,12,13,14,15
30 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
E 1000:TRAP 110:? "CHECKING DATA"
40 LINE=LINE+10:? "LINE:";LINE:READ DA
T$:IF LEN(DAT$)<>90 THEN 160
50 DATLIN=PEEK(183)+PEEK(184)*256:IF D
ATLIN<>LINE THEN ? "LINE ";LINE;" MISS
ING!":END
60 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
EX(D1)*16+HEX(D2)
70 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
EAD CHKSUM:GOTO 40
80 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
TOTAL=TOTAL-1000
90 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
THEN 40
100 GOTO 160
110 IF PEEK(195)<>6 THEN 160
120 IF PASS=2 THEN PUT #1,224:PUT #1,2
:PUT #1,225:PUT #1,2:PUT #1,0:PUT #1,3
2:CLOSE #1:END
130 ? "INSERT DISK, PRESS RETURN";:DIM
 IN$(1):INPUT IN$:OPEN #1,8,0,"D:HBUG.
COM"
140 PUT #1,255:PUT #1,255:PUT #1,0:PUT
 #1,32:PUT #1,159:PUT #1,37
150 ? :? "WRITING FILE":PASS=2:LINE=99
0:RESTORE 1000:TRAP 110:GOTO 40
160 ? "BAD DATA: LINE ";LINE:END
1000 DATA A9008D0ED48D8325BA8E7925AD06
028D1326AD07028D1426A9248D0702A9478D06
02A224A907DA907205CE4A940,79
1010 DATA 8D0ED420B82320132420EF23A200
A9878D4403A9258D4503A9058D4203A97F8D48
038E49032056E430E2A200BD,272
1020 DATA 8725C99BF003E810F6E000F0CC8E
8225A200BD8725C920D0148AA8B98825998725
C8CC8225D0F4CE82254C6C20,99
1030 DATA E8EC8225D0DFAD8225D0034C3320
A9008D7E25AC7E25B9D4248D7F25BED324A000
B98725DDC824D018C8E8EC7F,862
1040 DATA 25D0F1AD7E250AAABDDA248D8025
8DDB248D81256C8025EE7E25AD7E25C907D0C5
20F6234C3320B98725C99BF0,412
1050 DATA 034CD520A2E4A062A907205CE4AD
13268D0602AD14268D07026C0A00086829EF48
2820B823AD7A25203024AD7D,291
1060 DATA 258D0025AD7C258DFF24AD7B2520
3024AD7D258DFE24AD7C258DFD2420512120013
24A2FDA0242017244C3320B9,530
1070 DATA 8725C99BF0034CD5202051214205
A0252017244C3320A000B97525203024BE6025
AD7C259D0525AD7D259D0625,388
```

```
1080 DATA C8C005D0E660A204BD75259D0726
CA10F7A200A9008D0D268D0C26B98725C99BF0
0AC92CF00620BD214C8321AD,330
1090 DATA 0C26F006AD0D269D0726B98725C9
9BD00EA204BD07269D7525CA10F74C3320E8C8
E005D0C14CD5208E8525A20F,194
1100 DATA DD6525F006CA10F84CEF210E0D26
0E0D260E0D260E0D268A0D0D268D0D26EE0C26
AD0C26C903B005AE8525C860,803
1110 DATA 68684CD520B98725C99BD01DA5D4
18690885D4A5D5690085D5A9008D1126A9078D
10264C50224CD52020612 3AE,555
1120 DATA 1226F0F5AE0E2686D4AE0F2686D5
C99BF0DAC92CD0E3C8206123AE1226F0DAC99B
D0D6AD0E2638E5D48D1026AD,656
1130 DATA 0F26E5D58D1126A920A22C9D3225
CA10FAA91BA20E9D4F25CACA10F9A5D4203024
AD7D258D3525AD7C258D3425,403
1140 DATA A5D5203024AD7D258D3325AD7C25
8D3225A000B1D4C99BF006BE2A259D4F252030
24BE2225AD7D259D3825AD7C,975
1150 DATA 259D3725C8AD102638E9018D1026
AD1126E9008D1126100620 0C244C3320C008D0
C4200C2410034C3320A5D418,354
1160 DATA 690885D4A5D5690085D54C5022B9
8725C93CF016206123AE1226F04BC93CD047AE
0E2686D4AE0F2686D5C8A200,468
1170 DATA A9008D0D268D0C26B98725C99BF0
0AC92CF00620BD214C0523AD0C26F00D8C8625
8AA8AD0D2691D4AC8625B987,181
1180 DATA 25C99BD0034C3320E8C84CFD224C
D520206123C99BD01DAD1226F018AE79259AAD
782548AE7625AC7725AD7525,182
1190 DATA EE8325286C0E264CD520A9008D0E
268D0F268D1226B98725C99BF008C92CF004C9
3CD00160A20FDD6525F008CA,864
1200 DATA 10F868684CD5200E0E262E0F260E
0E262E0F260E0E262E0F260E0E262E0F268A0D
0E268D0E26EE1226AD1226C9,437
1210 DATA 05B0D2C84C6C23D820D923A200A9
E68D4403A9248D4503A9038D4203A90C8D4A03
8E4B032056E460A2078E8425,861
1220 DATA A90C9D42032056E4CE8425AE8425
10F060A2EAA0244C1724A9801987259987 25A2
EFA0242017 24A287A0254C17,511
1230 DATA 24A232A0254C1724A2E9A0248E44
038C4503A200A9098D4203A97F8D48038E4903
2056E46048290FAABD65258D,595
1240 DATA 7D25684A4A4A4AAABD65258D7C25
60AD8325D0026840D8A9008D83258E76258C77
25688D7525688D78256838E9,252
1250 DATA 028D7A2568E9008D7B258E8525BA
8E7925AE8525584CFD20AD8325F043A511D03F
A9808511A9008D8325BABD01,185
1260 DATA 018D7725BD02018D7625BD03018D
7525BD04018D7825BD05018D7A25BD06018D7B
258A1869068D7925A9209D06,879
1270 DATA 01A9FD9D05014C62E4444F534452
43523C4443470003050809 0A0BDB203A216E21
F421DD223A23453A9B9B4842,393
1280 DATA 55479BFD494E505554204552524F
52219B2020202020202020202020202020413D2020
20583D202020593D20202050,148
1290 DATA 3D20202053 3D20209B000306090C
0F1215010305070908 0D0F2020202020202020
2020202020202020202020 20,456
1300 DATA 2020202020202020202020202020
2020202020202020202020209B060B10151A3031
323334353637383941424344,282
1310 DATA 4546000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000,421
```

●

CHECKSUM DATA

(See p. 30)

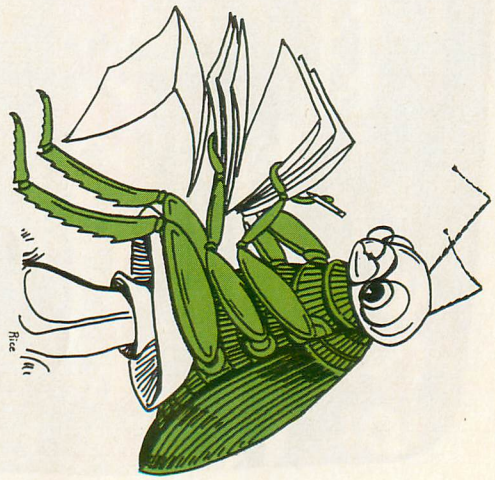
```
10 DATA 445,957,808,431,727,198,599,55
3,272,701,611,112,74,561,36,7085
160 DATA 165,899,670,50,200,953,769,78
8,545,908,10,936,728,156,4,7781
1140 DATA 914,480,888,896,938,985,945,
703,671,650,801,931,906,613,145,11466
1290 DATA 871,748,340,1959
```

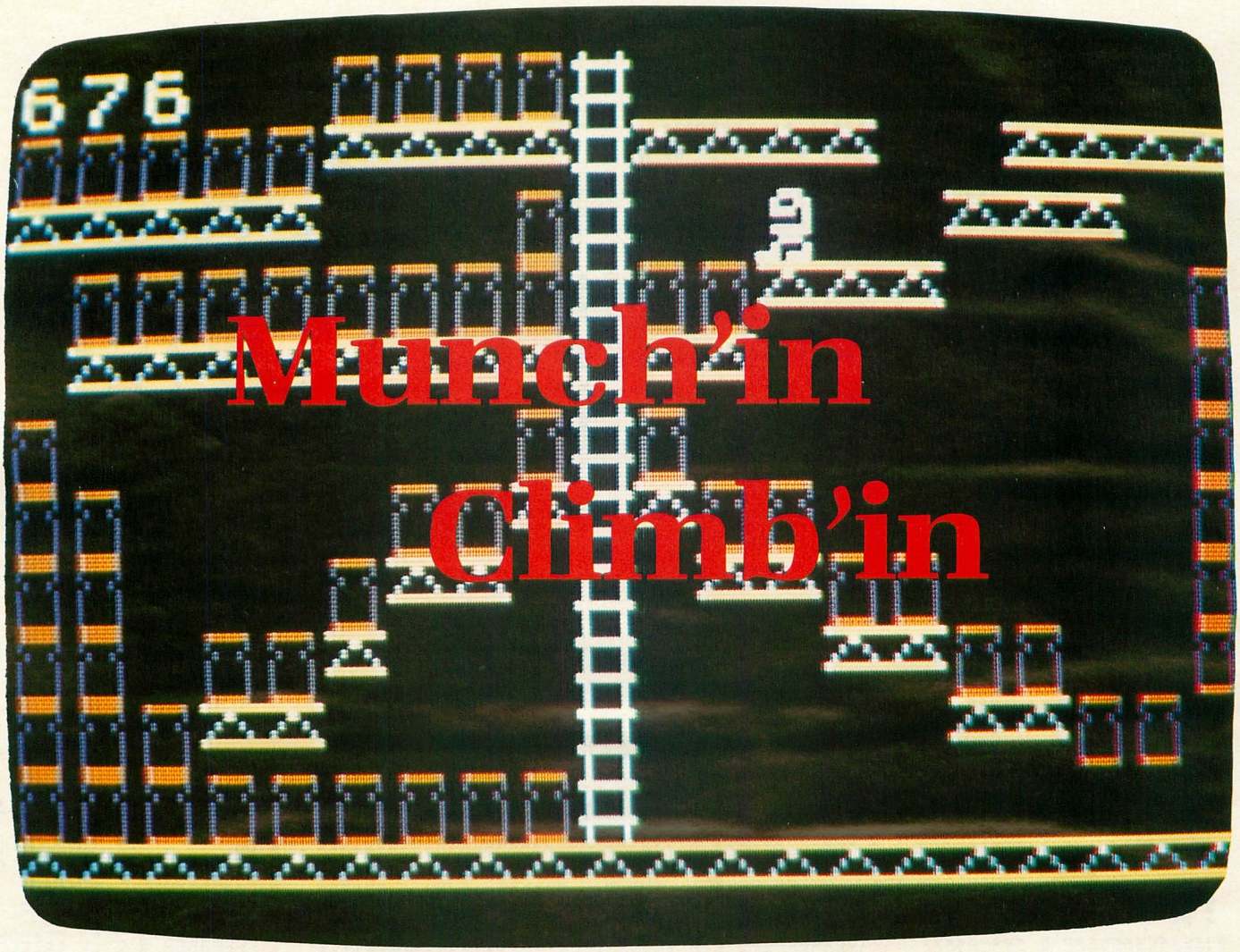
Assembly Listing.

```
                ;OPT NO LIST
;
;------------------------------
;HUDSON'S DEBUG PROGRAM (HBUG)
;------------------------------
;
DOSVEC  =   $0A     ;DOS run address
BRKKEY  =   $11     ;BREAK key status
STACK   =   $0100   ;hardware stack
VBREAK  =   $0206   ;BRK inst vector
RUNAD   =   $02E0   ;prog run addr
ICCOM   =   $0342   ;CIO command
ICBADR  =   $0344   ;CIO buffer address
ICBLEN  =   $0348   ;CIO buffer length
ICAUX1  =   $034A   ;CIO aux. byte 1
ICAUX2  =   $034B   ;CIO aux. byte 2
NMIEN   =   $D40E   ;interrupt enable
CIOV    =   $E456   ;CIO entry point
SETVBV  =   $E45C   ;VBLANK setup
XITVBV  =   $E462   ;VBLANK exit
;
;Page zero usage
;
CML     =   $D4     ;my two-byte
CMH     =   $D5     ;address work area
;
;--------------------
;Program entry point
;--------------------
;
HBUG
        LDA #$00    ;turn off...
        STA NMIEN   ;interrupts
        STA EXEC    ;and execute flag
        TSX         ;get stack pointer
        STX USERS   ;put in user area
        LDA VBREAK  ;save old BRK
        STA BRKSVL  ;vector in my
        LDA VBREAK+1 ;work area
        STA BRKSVH  ;for later
        LDA # >BRKHAN ;now point to
        STA VBREAK+1 ;my BRK inst.
        LDA # <BRKHAN ;routine
        STA VBREAK
        LDX # >VBI  ;set up
        LDY # <VBI  ;vertical
        LDA #7      ;blank
        JSR SETVBV  ;interrupt
        LDA #$40    ;turn on the
        STA NMIEN   ;VBLANK interrupt
        JSR NEWSCR  ;open GR.0 screen
INPUT   JSR PRTCR   ;carriage return
        JSR PROMPT  ;print HBUG prompt
INPT2   LDX #0
        LDA # <MYBUFF ;point to
        STA ICBADR  ;my input
        LDA # >MYBUFF ;buffer
        STA ICBADR+1
        LDA #5      ;GET RECORD command
        STA ICCOM
        LDA #$7F    ;my buffer's
        STA ICBLEN  ;length
        STX ICBLEN+1
        JSR CIOV    ;get input!
        BMI INPT2   ;go back if error
;
;This section 'squishes' all the
;spaces out of the input line.
;
        LDX #0      ;first find end
FINDBE  LDA MYBUFF,X ;of line
        CMP #$9B    ;CR?
        BEQ GOTEND  ;yes!
        INX         ;no, next char.
        BPL FINDBE  ;keep looking!
GOTEND  CPX #0      ;CR first char?
        BEQ INPUT   ;yes, try again.
        STX ENDPTR  ;save end index
        LDX #0      ;start w/1st char
SQUISH  LDA MYBUFF,X ;get the character
        CMP #32     ;space?
        BNE NOSQSH  ;no!
        TXA         ;move index
        TAY         ;to Y register
SQSHLP  LDA MYBUFF+1,Y ;shift all
        STA MYBUFF,Y ;characters
        INY         ;back to remove
        CPY ENDPTR  ;the space.
        BNE SQSHLP
        DEC ENDPTR  ;line 1 shorter
        JMP SQUISH  ;keep squishing!
NOSQSH  INX         ;next char
        CPX ENDPTR  ;end?
        BNE SQUISH  ;no, keep going!
        LDA ENDPTR  ;was line all spaces?
        BNE GETCMD  ;no, get command
        JMP INPUT   ;get another input!
;
;Now find command & process it
;
GETCMD  LDA #0      ;start with 1st
        STA CMDPTR  ;command.
CHKCMD  LDY CMDPTR  ;get command pointer
        LDA CMDST+1,Y ;get index of
        STA CEND    ;command text end
        LDX CMDST,Y ;and start.
        LDY #0      ;point to 1st char
CMDCMP  LDA MYBUFF,Y ;is buffer char
        CMP CMDTXT,X ;= command char?
        BNE NOTCMD  ;no!
        INY         ;yes, try next char
        INX         ;next command char
        CPX CEND    ;end of command?
        BNE CMDCMP  ;no, keep comparing!
        LDA CMDPTR  ;yes, mult command
        ASL A       ;index by 2 to point
        TAX         ;into jump table
        LDA CMDADR,X ;get command routine
        STA CMDJMP  ;address low byte
        LDA CMDADR+1,X ;and high byte
        STA CMDJMP+1 ;and save.
        JMP (CMDJMP) ;jump to routine!
NOTCMD  INC CMDPTR  ;try next command
        LDA CMDPTR  ;get pointer
        CMP #7      ;more commands?
        BNE CHKCMD  ;yes!
GOTERR  JSR ERRMSG  ;oops! bad command!
        JMP INPUT   ;get another input
;
;This section handles the DOS
;command.  It shuts off the HBUG
;VBLANK and BRK vectors and JMPs
;to DOS.
;
```

```
GODOS   LDA MYBUFF,Y ;get next char
        CMP #$9B     ;CR?
        BEQ DOSOK    ;yes, go to DOS
        JMP GOTERR   ;no, invalid command!
DOSOK   LDX # >XITVBV ;point VBLANK
        LDY # <XITVBV ;back to the
        LDA #7       ;system exit
        JSR SETVBV   ;point.
        LDA BRKSVL   ;restore old
        STA VBREAK   ;system
        LDA BRKSVH   ;BRK vector from
        STA VBREAK+1 ;save area
        JMP (DOSVEC) ;go to DOS!
;
;Show regs after BREAK key or
;BRK instruction.
;
SHOBRK  PHP          ;get processor
        PLA          ;status in A,
        AND #$EF     ;turn off BRK flag
        PHA          ;and put back in
        PLP          ;status register!
        JSR NEWSCR   ;re-open screen
        LDA USRPCL   ;get prog counter low
        JSR BINHEX   ;convert to hex
        LDA HEX2     ;get low digit
        STA BRK4     ;put in line
        LDA HEX1     ;get high digit
        STA BRK3     ;put in line, too
        LDA USRPCH   ;get PC high
        JSR BINHEX   ;convert to hex
        LDA HEX2     ;put low digit
        STA BRK2     ;in line
        LDA HEX1     ;and high digit
        STA BRK1     ;in line.
        JSR SETREG   ;set rest of line
        JSR PRTCR    ;carriage return
        LDX # <BRKTXT ;point to
        LDY # >BRKTXT ;BREAK text
        JSR PRINT    ;print it
        JMP INPUT    ;and get input
;
;Display registers (DR)
;
SHOREG  LDA MYBUFF,Y ;is next char
        CMP #$9B     ;a CR?
        BEQ DROKAY   ;yes, it's OK!
        JMP GOTERR   ;otherwise ERROR
DROKAY  JSR SETREG   ;get registers
        LDX # <DRTXT ;point to the
        LDY # >DRTXT ;DR text
        JSR PRINT    ;print it
        JMP INPUT    ;and get input
;
;Set up register disp area
;
SETREG  LDY #0       ;start w/user byte 0
SETLP   LDA USER,Y   ;get data
        JSR BINHEX   ;convert to hex
        LDX DRPTR,Y  ;get its position
        LDA HEX1     ;get first digit
        STA DRTXT,X  ;put in line
        LDA HEX2     ;get second digit
        STA DRTXT+1,X ;put in line
        INY          ;next byte
        CPY #5       ;done 5?
        BNE SETLP    ;not yet!
        RTS          ;all done!
;
;Change registers (CR)
;
CHGREG  LDX #4       ;first copy user
INITMP  LDA USER,X   ;registers
        STA TMPUSR,X ;to temporary
        DEX          ;hold area
        BPL INITMP
        LDX #0       ;1st user byte
CRSTRT  LDA #0       ;zero out...
        STA BHOLD    ;byte hold
        STA HDIG     ;and digit count
CRLOOK  LDA MYBUFF,Y ;get input char
        CMP #$9B     ;CR?
        BEQ STOTRY   ;yes, all done.
        CMP #$2C     ;comma?
        BEQ STOTRY   ;yes, try store
        JSR CTOBIN   ;convert it to binary
        JMP CRLOOK   ;and do next one
STOTRY  LDA HDIG     ;any digits?
        BEQ NXTCCK   ;no, skip it!
        LDA BHOLD    ;yes, save the byte
        STA TMPUSR,X ;in temporary table
NXTCCK  LDA MYBUFF,Y ;was this
        CMP #$9B     ;a CR?
        BNE NXTCHR   ;no, keep going.
        LDX #4       ;otherwise,
SETUSR  LDA TMPUSR,X ;copy the
        STA USER,X   ;temporary table
        DEX          ;back to the
        BPL SETUSR   ;user registers
        JMP INPUT    ;get another line
NXTCHR  INX          ;next user register
        INY          ;next character
        CPX #5       ;done 5 regs?
        BNE CRSTRT   ;no, loop back.
        JMP GOTERR   ;UH-OH! too many!
;
;Convert char. to binary #
;
CTOBIN  STX XHOLD    ;save X register
        LDX #$0F     ;set scan index
HDSCAN  CMP HEXDIG,X ;compare ASCII
        BEQ GOTHD    ;got it!
        DEX          ;next char
        BPL HDSCAN   ;scan all 16
        JMP CBERR    ;not in table!
GOTHD   ASL BHOLD    ;shift current #
        ASL BHOLD    ;left 4 times to
        ASL BHOLD    ;multiply it by
        ASL BHOLD    ;16
        TXA          ;get this digit
        ORA BHOLD    ;and add it to
        STA BHOLD    ;the hold area.
        INC HDIG     ;one more digit
        LDA HDIG     ;are there more
        CMP #3       ;than 2?
        BCS CBERR    ;yes--too big!
        LDX XHOLD    ;restore X reg
        INY          ;next buffer char
        RTS          ;and exit
CBERR   PLA          ;remove return addr
        PLA          ;from stack
        JMP GOTERR   ;and do error routine
;
;Display memory contents
;
SHOMEM  LDA MYBUFF,Y ;is character...
        CMP #$9B     ;a CR?
        BNE GETSAD   ;no, get address.
        LDA CML      ;get last address
        CLC          ;and add 8 to it
        ADC #8       ;since no address
        STA CML      ;was specified
        LDA CMH
        ADC #0
        STA CMH
SHOW8   LDA #8       ;show only 8
        STA COUNTH   ;bytes
        LDA #7
        STA COUNTL
        JMP SHOWLN   ;go show 'em!
SMERR   JMP GOTERR   ;jump to err routine
GETSAD  JSR GET4     ;get 4-byte address
        LDX G4DIGS   ;any digits found?
        BEQ SMERR    ;no!
        LDX ADL      ;save address
        STX CML      ;in page zero
        LDX ADH      ;work area
        STX CMH
        CMP #$9B     ;CR after address?
        BEQ SHOW8    ;yes, show 8 bytes.
        CMP #$2C     ;comma?
        BNE SMERR    ;no, bad command
        INY          ;next char
        JSR GET4     ;get end address
        LDX G4DIGS   ;got any digits?
        BEQ SMERR    ;no!
        CMP #$9B     ;CR after end addr?
        BNE SMERR    ;no!
        LDA ADL      ;now subtract
        SEC          ;end address
        SBC CML      ;from start
        STA COUNTL   ;to get number
        LDA ADH      ;of bytes to
        SBC CMH      ;display.
        STA COUNTH
SHOWLN  LDA #32      ;clear out
        LDX #44      ;display line
CLRML   STA SHOM1,X
        DEX
        BPL CLRML
        LDA #$1B     ;and set up ESC
        LDX #14      ;characters
SETESC  STA ASCII,X  ;in the ASCII
        DEX          ;display area
        DEX
        BPL SETESC
        LDA CML      ;convert the
        JSR BINHEX   ;current address
        LDA HEX2     ;to ascii hex
        STA SHOM4    ;characters
        LDA HEX1     ;and put in
        STA SHOM3    ;the memory
        LDA CMH      ;display line.
        JSR BINHEX   ;this is done
        LDA HEX2     ;2 times, for
        STA SHOM2    ;the low and high
        LDA HEX1     ;bytes of the
        STA SHOM1    ;address
        LDY #0       ;start showing!
BILDLN  LDA (CML),Y  ;get mem byte
        CMP #$9B     ;CR?
        BEQ NO9B     ;yes, don't show it!
        LDX ASCPOS,Y ;put in ASCII
        STA ASCII,X  ;display area
NO9B    JSR BINHEX   ;convert byte to hex
        LDX SMPOS,Y  ;get position
        LDA HEX2     ;get low char
        STA SMDATA+1,X ;and store
        LDA HEX1     ;get high char
        STA SMDATA,X ;and store.
        INY          ;next byte
        LDA COUNTL   ;now decrement
        SEC          ;the byte count
        SBC #1       ;by 1
        STA COUNTL
        LDA COUNTH
        SBC #0
        STA COUNTH
        BPL MORESM   ;more? yes!
        JSR PDATA    ;no more, print line
        JMP INPUT    ;and get next command
MORESM  CPY #8       ;done 8 bytes?
        BNE BILDLN   ;no, loop back
        JSR PDATA    ;done 8, print 'em
        BPL NOSTOP   ;no BREAK key
        JMP INPUT    ;get next command
NOSTOP  LDA CML      ;increment display
        CLC          ;address by 8
        ADC #8
        STA CML
        LDA CMH
        ADC #0
        STA CMH
        JMP SHOWLN   ;and loop back.
;
;Change memory contents (Cnnnn<)
;
CHGMEM  LDA MYBUFF,Y ;get char
        CMP #$3C     ;'<'?
        BEQ CMDFLT   ;yes, default address
        JSR GET4     ;get the address
        LDX G4DIGS   ;got any digits?
        BEQ CMERR    ;no!
        CMP #$3C     ;next char '<'?
        BNE CMERR    ;no!
        LDX ADL      ;save the
        STX CML      ;change memory
        LDX ADH      ;address on
        STX CMH      ;page zero.
CMDFLT  INY          ;next buffer char
        LDX #0       ;1st memory byte
CMSTRT  LDA #0       ;zero out...
        STA BHOLD    ;byte hold
        STA HDIG     ;and digit count
CMLOOK  LDA MYBUFF,Y ;get char
        CMP #$9B     ;CR?
        BEQ CMTRY    ;yes, all done!
        CMP #$2C     ;comma?
        BEQ CMTRY    ;yes, store last byte
        JSR CTOBIN   ;convert char to binary
        JMP CMLOOK   ;and get next
CMTRY   LDA HDIG     ;any digits?
        BEQ NEXTCM   ;no! don't store
        STY YHOLD    ;save Y register
        TXA          ;move X...
        TAY          ;to Y
        LDA BHOLD    ;get byte to store
        STA (CML),Y  ;and store it!
        LDY YHOLD    ;get Y back
NEXTCM  LDA MYBUFF,Y ;get last char
        CMP #$9B     ;was it CR?
        BNE NXTLOC   ;no, continue
        JMP INPUT    ;all done!
NXTLOC  INX          ;next mem byte
        INY          ;next input char
        JMP CMSTRT   ;loop back!
CMERR   JMP GOTERR   ;go to error routine
;
;Execute at address (Gnnnn)
;
EXECUT  JSR GET4     ;get the run address
        CMP #$9B     ;is that all?
        BNE EXERR    ;no!
        LDA G4DIGS   ;got any digits?
        BEQ EXERR    ;no!
        LDX USERS    ;put user stack
        TXS          ;pointer in S
        LDA USERP    ;put user status
        PHA          ;on stack
        LDX USERX    ;set user X reg
        LDY USERY    ;and user Y reg
        LDA USERA    ;and accumulator
        INC EXEC     ;set execute flag
        PLP          ;get status off stack
        JMP (ADL)    ;go to run address!
EXERR   JMP GOTERR   ;go to error routine
;
;Get 4-character address
;
GET4    LDA #0       ;zero out...
        STA ADL      ;address low byte
        STA ADH      ;address high byte
        STA G4DIGS   ;digit count
G4LOOP  LDA MYBUFF,Y ;get char
        CMP #$9B     ;CR?
        BEQ G4END    ;yes, all done!
        CMP #$2C     ;comma?
        BEQ G4END    ;yes, all done!
        CMP #$3C     ;'<'?
```

```
        BNE TESTIT  ;no, check digit
G4END   RTS         ;exit!
TESTIT  LDX #$0F    ;set hex digit pointer
G4SCAN  CMP HEXDIG,X ;is it this char?
        BEQ GOTG4D  ;yes!
        DEX         ;try next hex digit
        BPL G4SCAN  ;loop if more.
G4ERR   PLA         ;discard return
        PLA         ;address,
        JMP GOTERR  ;show error.
GOTG4D  ASL ADL     ;this code
        ROL ADH     ;shifts the
        ASL ADL     ;current address
        ROL ADH     ;left 4 bits
        ASL ADL     ;in order to
        ROL ADH     ;multiply it
        ASL ADL     ;by 16.
        ROL ADH
        TXA         ;get this digit
        ORA ADL     ;and add it to
        STA ADL     ;the address.
        INC G4DIGS  ;one more digit
        LDA G4DIGS  ;how many total?
        CMP #5      ;more than 4?
        BCS G4ERR   ;yes! error!
        INY         ;ok, do next character
        JMP G4LOOP  ;and loop back.
;
;Set up new graphics 0 screen
;
NEWSCR  CLD
        JSR NOIOCB  ;close all IOCB's
;
;now open screen!
;
        LDX #0
        LDA # <EADR ;'E:' filename
        STA ICBADR
        LDA # >EADR
        STA ICBADR+1
        LDA #3      ;OPEN command
        STA ICCOM
        LDA #12     ;I/O
        STA ICAUX1
        STX ICAUX2  ;zero aux byte
        JSR CIOV    ;open it!
        RTS         ;and return.
;
;Close all IOCB's
;
NOIOCB  LDX #7      ;first close
        STX SAVEX   ;all IOCB's.
CLOOP   LDA #12     ;CLOSE command
        STA ICCOM,X
        JSR CIOV    ;close it!
        DEC SAVEX   ;next IOCB
        LDX SAVEX   ;more IOCB's?
        BPL CLOOP   ;yes!
        RTS         ;all done!
;
;Show HBUG prompt
;
PROMPT  LDX # <HMSG ;point to
        LDY # >HMSG ;HBUG message
        JMP PRINT   ;and print it!
;
;Show error message
;
ERRMSG  LDA #$80    ;set high bit
        ORA MYBUFF,Y ;inverse the
        STA MYBUFF,Y ;invalid character
        LDX # <ERRTXT ;point to
        LDY # >ERRTXT ;error message
        JSR PRINT   ;print it
        LDX # <MYBUFF ;point to
        LDY # >MYBUFF ;input buffer
        JMP PRINT   ;print it, too!
;
;Print memory display line
;
PDATA   LDX # <SHOM1 ;point to memory
        LDY # >SHOM1 ;display line
        JMP PRINT   ;print it!
;
;Print carriage return only
;
PRTCR   LDX # <CR   ;point to CR and
        LDY # >CR   ;fall thru to print
;
;General-use print routine
;
PRINT   STX ICBADR  ;save print area lo
        STY ICBADR+1 ;and high
        LDX #0      ;zero X reg.
        LDA #9      ;PUT RECORD command
        STA ICCOM
        LDA #$7F    ;set up...
        STA ICBLEN  ;buffer length
        STX ICBLEN+1
        JSR CIOV    ;print it!
        RTS         ;and exit.
;
;Binary-to-hex converter
;
BINHEX  PHA         ;save byte
        AND #$0F    ;get low 4 bits
        TAX         ;put in index
        LDA HEXDIG,X ;lookup hex
        STA HEX2    ;and save
        PLA         ;get byte again
        LSR A       ;shift right
        LSR A       ;4 times
        LSR A       ;to get
        LSR A       ;high 4 bits
        TAX         ;put in index
        LDA HEXDIG,X ;lookup hex
        STA HEX1    ;and save
        RTS         ;all done!
;
;Handle 6502 BRK interrupt
;
BRKHAN  LDA EXEC    ;executing?
        BNE SAVREG  ;yes!
        PLA         ;no, restore accum.
        RTI         ;and return from int.
SAVREG  CLD         ;no decimal mode!
        LDA #0      ;reset the
        STA EXEC    ;executing flag
        STX USERX   ;save X
        STY USERY   ;and Y
        PLA
        STA USERA   ;and accumulator
        PLA
        STA USERP   ;and status reg.
        PLA
        SEC         ;now get program
        SBC #2      ;counter from stack
        STA USRPCL  ;and subtract 2
        PLA         ;to get BREAK
        SBC #0      ;address.
        STA USRPCH
        STX XHOLD   ;save X reg.
        TSX         ;now store stack pntr
        STX USERS   ;in the user area.
        LDX XHOLD   ;restore X
        CLI         ;clear interrupt
        JMP SHOBRK  ;and show break info
;
;Handle BREAK key in VBI
;
VBI     LDA EXEC    ;executing?
        BEQ NOBKEY  ;no!
        LDA BRKKEY  ;BREAK pressed?
        BNE NOBKEY  ;no!
        LDA #$80    ;reset BREAK
        STA BRKKEY  ;press flag
        LDA #0      ;and
        STA EXEC    ;execute flag
        TSX         ;get stack pntr
        LDA STACK+1,X ;get Y register
        STA USERY   ;and save it
        LDA STACK+2,X ;get X register
        STA USERX   ;and save it
        LDA STACK+3,X ;get accumulator
        STA USERA   ;and save it
        LDA STACK+4,X ;get status register
        STA USERP   ;and save it
        LDA STACK+5,X ;get program
        STA USRPCL  ;counter
        LDA STACK+6,X ;and
        STA USRPCH  ;store it!
        TXA         ;move stack pntr
        CLC         ;to A, add 6 to
        ADC #6      ;get true value
        STA USERS   ;and save it!
        LDA # >SHOBRK ;change return
        STA STACK+6,X ;address to
        LDA # <SHOBRK ;go to SHOBRK
        STA STACK+5,X ;after VBLANK.
NOBKEY  JMP XITVBV  ;all done!
;
;Data areas
;
;Command text & pointers
;
CMDTXT  .BYTE "DOSDRCR<DCG"
CMDST   .BYTE 0,3,5,8,9,10,11
CMDADR  .WORD GODOS
        .WORD SHORE0
        .WORD CH0RE0
        .WORD SHOMEM
        .WORD CH0MEM
        .WORD EXECUT
;
;Miscellaneous text
;
EADR    .BYTE "E:",$9B
CR      .BYTE $9B
HMSG    .BYTE "HBUG",$9B
ERRTXT  .BYTE $FD,"INPUT ERROR!",$9B
BRKTXT
BRK1    .BYTE 32
BBRK2    .BYTE 32
BRK3    .BYTE 32
BRK4    .BYTE 32
        .BYTE "    "
DRTXT   .BYTE "    A=   X=   Y=   "
        .BYTE "P=   S=  " $9B
SMPOS   .BYTE 0,3,6,9,12,15,18,21
ASCPOS  .BYTE 1,3,5,7,9,11,13,15
SHOM1   .BYTE 32
SHOM2   .BYTE 32
SHOM3   .BYTE 32
SHOM4   .BYTE 32
FILL1   .BYTE 32
SMDATA  .BYTE "              "
        .BYTE "              "
FILL2   .BYTE 32
ASCII   .BYTE "                "
        .BYTE $9B
DDRPTR    .BYTE 6,11,16,21,26
HEXDIG  .BYTE "0123456789ABCDEF"
;
;Misc. memory usage
;
USER    ;       user registers
USERA   .BYTE 0     ;accumulator
USERX   .BYTE 0     ;X register
USERY   .BYTE 0     ;Y register
USERP   .BYTE 0     ;status register
USERS   .BYTE 0     ;stack pointer
USRPCL  *=  *+1     ;program counter low
USRPCH  *=  *+1     ;program counter high
HEX1    *=  *+1     ;hex digit 1
HEX2    *=  *+1     ;hex digit 2
CMDPTR  *=  *+1     ;command pointer
CEND    *=  *+1     ;command end index
CMDJMP  *=  *+2     ;command jump addr
ENDPTR  *=  *+1     ;end of input pointer
EXEC    *=  *+1     ;user prog execute flag
SAVEX   *=  *+1     ;X reg. hold area
XHOLD   *=  *+1     ;another X reg hold
YHOLD   *=  *+1     ;Y reg hold
MYBUFF  *=  *+128   ;input buffer
TMPUSR  *=  *+5     ;user reg temp storage
HDIG    *=  *+1     ;hex digit count
BHOLD   *=  *+1     ;byte hold area
ADL     *=  *+1     ;address low
ADH     *=  *+1     ;address high
COUNTL  *=  *+1     ;disp mem count,
COUNTH  *=  *+1     ;low and high
G4DIGS  *=  *+1     ;4-digit count
BRKSVL  *=  *+1     ;BRK interrupt...
BRKSVH  *=  *+1     ;vector save
;
;Tell computer where to run HBUG
;
        *=  RUNAD
        .WORD HBUG
;
;That's all, folks!
;
        .END
```





16K Cassette 24K Disk

by Mark Comeau

Your little muncher is one of the hungriest guys around. He'll eat cherries, pumpkins, milk cans and anything else he can find on the grid of girders and ladders. Trouble is, lunchtime is limited to only a few precious seconds. Can you gobble down an entire grid full of goodies before the clock runs out?

You can move your muncher up or down ladders and across girders with a joystick plugged into port #1. You can even fly around to grab those hard-to-reach snacks by jumping off the grid and pushing the stick in the desired direction. Be careful, though! If you sail over a prize you'll have to climb back up the grid and jump for it again, costing valuable seconds.

You get one point for every snack you munch. Some of the tidbits are so tasty, you'll have to pause over them for a moment before your drooling muncher will start to munch. Gobble up every prize on a grid and you'll go on to a more difficult grid; any time left over gets added to your score. Press the red fire button to exit a grid. There are eight different types of food and eight levels of play. If you manage to complete all eight levels, you'll be rewarded with an intermission and then get a chance to play again with a 10-second handicap.

See if you're good enough to complete all eight levels of **Munch'In Climb'In**. It *is* possible, but very, very difficult! □

Program breakdown.

Line #	Function
40-120	Init Tom Hudson's PMG routine
130-220	Screen set-up and init
230-330	Main movement subroutine
340-360	Death!
370-510	Score display
520-880	Eight screen set-up routines
890-1000	Intermission
1280-1420	High score display
1430-1740	Character set init

```
10 REM * MUNCH'IN CLIMB'IN
20 REM * by Mark Comeau
30 REM * ANALOG Computing #17
40 GOSUB 1440:DIM P2$(C8),SCORE(C10),H
SCORE(C10),LG$(C16):FOR T=C1 TO C10:SC
ORE(T)=C0:NEXT T
50 DIM PMMOV$(100),P0$(C8),P1$(C8),A$(
C2),BUG(C5):MOVE=ADR(PMMOV$)
60 RESTORE 1640:FOR X=C1 TO 100:READ N
:PMMOV$(X)=CHR$(N):NEXT X
70 FOR I=C1 TO C8:READ N:P0$(I)=CHR$(N
):NEXT I
80 FOR I=C1 TO C8:READ N:P1$(I)=CHR$(N
):NEXT I
90 FOR I=C1 TO C8:READ N:P2$(I)=CHR$(N
):NEXT I
100 FOR I=C1 TO C16:READ N:LG$(I)=CHR$
(N):NEXT I
110 PMBASE=INT((PEEK(145)+C3)/C4)*C4:P
OKE 54279,PMBASE:PMB=PMBASE*256:PMD=AD
R(P0$)
120 POKE 559,46:POKE 53277,C3:ALT=ADR(
P1$):LAD=ADR(P2$):POS=LAD:POKE 704,C14
:LG=ADR(LG$)
130 GRAPHICS C18:POKE 756,PEEK(106)+C1
:POKE 559,46:POKE 77,C0:POKE 710,53
140 PRI=PRI+C1
150 IF PRI=205 THEN PRI=197
160 SCORE=SCORE+DOT:IF QWER=C1 THEN SC
ORE=SCORE+TIM:QWER=C0
170 IF MEN=C0 THEN MEN=C4:SCRE=C1:STM=
C0:PRI=196:GOTO 370
180 COLOR 68:PLOT C4,C6:DRAWTO C3+MEN,
C6
190 GOSUB 1390
200 ON SCRE GOSUB 520,590,680,750,820,
1010,1100,1200,890
210 FOR V=C5 TO C0 STEP -C1:FOR T=C0 T
O C3:SOUND T,255-T,14,V:NEXT T:NEXT V
220 X=200:Y=96:COLOR 32:DOT=C0:TIM=TIM
-STM
230 S=STICK(C0)
240 X=X+(S=C7 AND X<200)*C8-(S=C11 AND
 X>48)*C8:GX=(X-48)/C8:GY=(Y-C16)/C8:L
OCATE GX,GY,C:LOCATE GX,GY+C1,D
250 IF S=C7 THEN POS=PMD
260 IF S=C11 THEN POS=ALT
270 IF C=99 THEN Y=Y+(S=C13 AND Y<96)*
C8-(S=C14 AND Y>C16)*C8:POS=LAD
280 IF D=32 OR D=PRI THEN Y=Y+C8
290 IF C<>PRI THEN 310
300 SOUND C0,100,C14,C14:SOUND C0,C0,C
0,C0:PLOT GX,GY:DOT=DOT+C1:IF DOT=MAX
THEN SCRE=SCRE+C1:QWER=C1:GOTO 130
310 A=USR(MOVE,C0,PMB,POS,X,Y,C8)
320 TIM=TIM-C2:POSITION C0,C0:? #C6;TI
M;" ":POKE 704,C14:IF TIM=C0 THEN MEN=
MEN-C1:GOTO 340
330 GOTO 230
340 P=PEEK(560):A=USR(MOVE,C0,PMB,ALT,
X,Y,C8):FOR AM=C1 TO C10
350 FOR T=C0 TO C3:POKE 53256,T:R=INT(
RND(C0)*30):POKE 712,R:POKE 560,P+T:SO
UND C0,R,C8,C14:NEXT T:NEXT AM
360 POKE 53256,C0:SOUND C0,C0,C0,C0:GO
TO 130
370 DOT=C0:IF SCORE>SCORE(C10) THEN GO
TO 1280
380 IF STRIG(C0)=C0 THEN SCORE=C0:GOTO
 130
390 POKE 53256,C3:A=USR(MOVE,C0,PMB,LG
,150,50,C16)
400 SOUND C0,C0,C0,C0:GRAPHICS C18:POK
E 559,46
410 FOR T=C0 TO C3:SOUND T,255-T,C14,C
5:NEXT T
420 ? #C6;" MUNCH'IN CLIMB'IN":POSITIO
N C0,C3
430 ? #C6;"m                    c"
440 ? #C6;"u                    l"
450 ? #C6;"n                    i"
460 ? #C6;"c                    m"
470 ? #C6;"h                    b"
480 ? #C6;? #C6;"  BY  MARK COMEAU"
490 POSITION C3,C11:? #C6;"PRESS TRIGG
ER":POSITION C5,C5:? #C6;SCORE:SCORE=C
0:POKE 704,C14
500 POKE 710,INT(RND(C0)*255):POKE 709
,INT(RND(C0)*255):IF STRIG(C0)=C0 THEN
 POKE 704,C0:POKE 53256,C0:GOTO 130
510 GOTO 500
520 TIM=400:MAX=69
530 COLOR 66:PLOT C1,C9:DRAWTO C19,C9:
PLOT C2,C7:DRAWTO C10,C7:DRAWTO C15,C2
:PLOT C0,C11:DRAWTO C19,C11
540 PLOT C0,C4:DRAWTO C18,C1
550 COLOR PRI:PLOT C1,C8:DRAWTO C19,C8
:PLOT C2,C6:DRAWTO C10,C6:DRAWTO C14,C
2:PLOT C0,C10:DRAWTO C18,C10
560 PLOT C0,C3:DRAWTO C18,C0
570 COLOR 99:PLOT C19,C0:DRAWTO C19,C1
0
580 RETURN
590 TIM=800:POS=ALT:MAX=79
600 COLOR 99:PLOT C9,C0:DRAWTO C9,C10
610 COLOR 66:PLOT C0,C11:DRAWTO C19,C1
1:PLOT C8,C1:DRAWTO C0,C2:PLOT C10,C6:
DRAWTO C19,C11
620 PLOT C1,C4:DRAWTO C8,C4:PLOT C8,C6
:DRAWTO C0,C11:PLOT C10,C1:DRAWTO C19,
C1:DRAWTO C10,C4
630 COLOR PRI:PLOT C0,C10:DRAWTO C8,C1
0:PLOT C10,C10:DRAWTO C19,C10:PLOT C8,
C0:DRAWTO C0,C1
640 PLOT C10,C5:DRAWTO C19,C10:PLOT C1
,C3:DRAWTO C8,C3:PLOT C8,C5:DRAWTO C0,
C10:PLOT C8,C2
650 PLOT C10,C0:DRAWTO C14,C0:PLOT C19
,C3:DRAWTO C19,C8:DRAWTO C16,C10
660 PLOT C10,C3:DRAWTO C15,C1:PLOT C0,
C5:DRAWTO C0,C10:PLOT C1,C6:DRAWTO C1,
C9
670 RETURN
680 TIM=550:MAX=81
690 COLOR 99:PLOT C8,C1:DRAWTO C8,C10:
PLOT C19,C0:DRAWTO C19,C10
700 COLOR 66:PLOT C0,C11:DRAWTO C19,C1
1:PLOT C1,C2:DRAWTO C18,C10:PLOT C1,C4
:DRAWTO C14,C10:PLOT C1,C6
710 DRAWTO C10,C10:PLOT C18,C1:DRAWTO
C6,C1:PLOT C18,C3:DRAWTO C8,C3:PLOT C1
8,C5:DRAWTO C10,C5:PLOT C18,C7
720 DRAWTO C12,C7:COLOR PRI:PLOT C1,C1
:DRAWTO C18,C9:PLOT C1,C3:DRAWTO C14,C
9:PLOT C1,C5:DRAWTO C10,C9
730 PLOT C18,C0:DRAWTO C6,C0:PLOT C18,
C2:DRAWTO C8,C2:PLOT C18,C4:DRAWTO C10
,C4:PLOT C18,C6:DRAWTO C12,C6
740 RETURN
750 TIM=1800:MAX=135:POS=ALT
760 COLOR 99:PLOT C9,C0:DRAWTO C9,C10
770 COLOR 66:PLOT C0,C11:DRAWTO C19,C1
1:PLOT C8,C1:DRAWTO C0,C5:PLOT C10,C1:
DRAWTO C19,C5
780 COLOR PRI:PLOT C0,C10:DRAWTO C8,C1
0:PLOT C10,C10:DRAWTO C19,C10:PLOT C8,
C0:DRAWTO C0,C4
790 PLOT C10,C0:DRAWTO C19,C4:W=C2:FOR
 T=C8 TO C0 STEP -C1:W=W+0.5:PLOT T,W:
DRAWTO T,C10:NEXT T
800 W=C2:FOR T=C10 TO C19:W=W+0.5:PLOT
 T,W:DRAWTO T,C10:NEXT T
810 RETURN
820 TIM=450:MAX=80
830 COLOR 99:PLOT C0,C1:DRAWTO C0,C10:
PLOT C19,C0:DRAWTO C19,C10
840 COLOR 66:PLOT C0,C11:DRAWTO C19,C1
1:PLOT C1,C2:DRAWTO C18,C5:PLOT C1,C4:
DRAWTO C18,C7
850 PLOT C3,C2:DRAWTO C18,C1:PLOT C3,C
8:DRAWTO C15,C8:PLOT C1,C6:DRAWTO C6,C
11
860 COLOR PRI:PLOT C1,C10:DRAWTO C18,C
10:PLOT C1,C1:DRAWTO C18,C4:PLOT C1,C3
:DRAWTO C18,C6:PLOT C11,C0
870 DRAWTO C18,C0:PLOT C10,C1:PLOT C3,
C7:DRAWTO C15,C7:PLOT C1,C5:DRAWTO C6,
C10
880 RETURN
890 POS=ALT:STM=STM+C20
900 FOR T=C0 TO C3:SOUND T,C0,C0,C0:NE
XT T
910 COLOR 99:PLOT C8,C6:DRAWTO C8,C6:P
LOT C10,C6:DRAWTO C10,C6:COLOR 66:PLOT
 C8,C5:DRAWTO C10,C5
```

```
920 PLOT C8,C7:DRAWTO C10,C7:IF X)120
THEN FOR X=X TO 120 STEP -C1:GOSUB 100
0:NEXT X:GOTO 940
930 FOR X=X TO 120:GOSUB 1000:NEXT X
940 IF Y)64 THEN FOR Y=Y TO 64 STEP -C
1:GOSUB 1000:NEXT Y:GOTO 960
950 FOR Y=Y TO 64:GOSUB 1000:NEXT Y
960 FOR T=C0 TO 255:POKE 704,T:SOUND C
0,T,C14,C14:NEXT T
970 FOR T=255 TO C0 STEP -C1:SOUND C0,
T,C14,C14:NEXT T
980 FOR R=C1 TO C20:SOUND C0,C10,C13,C
14:SOUND C0,C0,C0,C0:FOR V=C14 TO C1 S
TEP -C1:SOUND C0,28-V,C2,V:NEXT V
990 NEXT R:SCRE=C1:POP :GOTO 130
1000 A=USR(MOVE,C0,PMB,POS,X,Y,C8):RET
URN
1010 TIM=950:MAX=92
1020 COLOR 66:PLOT C0,C1:DRAWTO C10,C1
1:PLOT C0,C3:DRAWTO C8,C11:PLOT C0,C5:
DRAWTO C6,C11:PLOT C0,C7
1030 DRAWTO C4,C11:PLOT C0,C9:DRAWTO C
2,C11:PLOT C0,C11:DRAWTO C19,C11:PLOT
C18,C1:DRAWTO C5,C5
1040 COLOR PRI:PLOT C0,C0:DRAWTO C10,C
10:PLOT C0,C2:DRAWTO C8,C10:PLOT C0,C4
:DRAWTO C6,C10:PLOT C0,C6
1050 DRAWTO C4,C10:PLOT C0,C8:DRAWTO C
2,C10:PLOT C0,C10:PLOT C14,C4:DRAWTO C
14,C10:PLOT C18,C0:DRAWTO C5,C4
1060 PLOT C13,C6:DRAWTO C13,C10:PLOT C
12,C6:DRAWTO C12,C10:PLOT C15,C4:DRAWT
O C15,C10
1070 PLOT C16,C3:DRAWTO C16,C10:PLOT C
17,C3:DRAWTO C17,C10:PLOT C18,C3:DRAWT
O C18,C10
1080 COLOR 99:PLOT C0,C0:DRAWTO C0,C10
:PLOT C19,C0:DRAWTO C19,C10
1090 RETURN
1100 TIM=800:MAX=107
1110 COLOR 99:PLOT C0,C1:DRAWTO C0,C10
:PLOT C19,C0:DRAWTO C19,C10:COLOR 66:P
LOT C0,C11:DRAWTO C19,C11
1120 PLOT C1,C2:DRAWTO C8,C3:PLOT C1,C
4:DRAWTO C7,C5:PLOT C1,C6:DRAWTO C6,C7
:PLOT C1,C8:DRAWTO C5,C9
1130 PLOT C18,C1:DRAWTO C10,C1:PLOT C1
8,C2:DRAWTO C10,C3:PLOT C18,C4:DRAWTO
C11,C5:PLOT C18,C6:DRAWTO C12,C7
1140 PLOT C18,C8:DRAWTO C13,C9
1150 COLOR PRI:PLOT C1,C10:DRAWTO C18,
C10:PLOT C1,C1:DRAWTO C8,C2:PLOT C1,C3
:DRAWTO C7,C4
1160 PLOT C1,C5:DRAWTO C6,C6:PLOT C1,C
7:DRAWTO C5,C8:PLOT C18,C0:DRAWTO C9,C
0:DRAWTO C9,C10:PLOT C14,C2
1170 DRAWTO C10,C2:PLOT C18,C3:DRAWTO
C11,C4:PLOT C18,C5:DRAWTO C12,C6:PLOT
C18,C7:DRAWTO C13,C8:PLOT C10,C5
1180 DRAWTO C10,C10:PLOT C8,C5:DRAWTO
C8,C10:PLOT C11,C6:DRAWTO C11,C10:PLOT
 C7,C6:DRAWTO C7,C10
1190 RETURN
1200 POS=ALT:TIM=700:MAX=79
1210 COLOR 99:PLOT C0,C0:DRAWTO C0,C11
:PLOT C19,C0:DRAWTO C19,C9:PLOT C9,C5:
DRAWTO C9,C8
1220 COLOR 66:PLOT C0,C11:DRAWTO C19,C
11:PLOT C1,C2:DRAWTO C18,C5
1230 PLOT C18,C1:DRAWTO C8,C1:PLOT C3,
C8:PLOT C4,C10:PLOT C15,C7
1240 PLOT C18,C9:DRAWTO C1,C9:PLOT C10
,C5:DRAWTO C10,C8:COLOR PRI:PLOT C1,C1
0:DRAWTO C19,C10:PLOT C1,C1
1250 DRAWTO C18,C4:PLOT C18,C0:DRAWTO
C8,C0:PLOT C3,C7:PLOT C4,C9:PLOT C15,C
6:PLOT C9,C4:DRAWTO C4,C9
1260 PLOT C8,C4:DRAWTO C3,C9:PLOT C3,C
1:DRAWTO C3,C8:PLOT C14,C5:DRAWTO C11,
C8:PLOT C1,C2:DRAWTO C1,C8
1270 RETURN
1280 POKE 704,C0:POKE 53256,C0:GRAPHIC
5 C18
1290 SETCOLOR C2,C0,C0:Q=C16:T=65:POKE
 752,C1:? #C6;" MUNCH'IN CLIMB'IN ":?
#C6;"       top ten"
1300 POKE 559,46:FOR W=C1 TO C10:IF SC
ORE>SCORE(W) THEN 1320
1310 NEXT W
1320 FOR T=C1 TO C10:HSCORE(T)=SCORE(T
):NEXT T
1330 FOR T=W TO C9:SCORE(T+C1)=HSCORE(
T):NEXT T:SCORE(W)=SCORE
1340 FOR T=C1 TO C10:POSITION C0,T+C1:
? #C6;SCORE(T):NEXT T:POSITION C5,W+C1
:? #C6;"YOUR SCORE"
1350 FOR T=255 TO C0 STEP -C4:FOR V=C0
 TO C3:SOUND V,T+V,C14,C14:NEXT V:NEXT
 T
1360 FOR V=C0 TO C3:SOUND V,C0,C0,C0:N
EXT V
1370 IF STRIG(C0)=C0 THEN 390
1380 GOTO 1370
1390 IF STRIG(C0)=C0 THEN RETURN
1400 FOR T=C0 TO C3:SOUND T,250-(T*C3)
,C14,C7:NEXT T
1410 POSITION C5,C5:? #C6;SCORE:FOR T=
C1 TO 600:NEXT T:POKE 53248,C0:POSITIO
N C5,C5
1420 ? #C6;"          ":POSITION C3,C6:
? #C6;"       ":RETURN
1430 REM * INITIALIZATION
1440 GRAPHICS 18:POSITION 6,5:? #6;"ST
AND BY"
1450 RESTORE 1720:READ C0,C1,C2,C3,C4,
C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15
,C16,C17,C18,C19,C20
1460 DIM XFR$(38):RESTORE 1730:FOR X=C
1 TO 38:READ Z:XFR$(X)=CHR$(Z):NEXT X
1470 POKE 106,PEEK(106)-C5:GRAPHICS C1
8:START=(PEEK(106)+C1)*256:POKE 756,ST
ART/256:POKE 752,C1
1480 Z=USR(ADR(XFR$)):RESTORE 1520
1490 READ X:IF X=-C1 THEN RESTORE 1520
:POKE 559,34:RETURN
1500 FOR Y=C0 TO C7:READ Z:POKE X+Y+ST
ART,Z:NEXT Y
1510 GOTO 1490
1520 DATA 264,0,12,18,57,124,124,124,5
6
1530 DATA 272,255,24,36,66,255,0,0,0
1540 DATA 288,28,38,58,2,62,28,34,238
1550 DATA 296,0,12,18,57,124,124,124,5
6
1560 DATA 304,14,17,60,126,126,126,126
,60
1570 DATA 312,126,66,36,66,66,66,66,12
6
1580 DATA 320,16,16,56,56,124,124,124,
56
1590 DATA 328,0,0,90,60,90,126,36,24
1600 DATA 336,129,189,126,219,255,189,
66,60
1610 DATA 344,32,50,24,12,22,19,56,124
1620 DATA 352,129,90,60,126,126,60,90,
129
1630 DATA 280,66,255,66,66,66,255,66,6
6,-1
1640 DATA 216,104,104,104,133,213,104,
24,105,2,133,206,104,133,205,104,133,2
04,104,133,203,104,104,133,208
1650 DATA 104,104,133,209,104,104,24,1
01,209,133,207,166,213,240,16,165,205,
24,105,128,133,205,165,206,105
1660 DATA 0,133,206,202,208,240,160,0,
162,0,196,209,144,19,196,207,176,15,13
2,212,138,168,177,203,164
1670 DATA 212,145,205,232,169,0,240,4,
169,0,145,205,200,192,128,208,224,166,
213,165,208,157,0,208,96
1680 DATA 56,100,92,64,124,56,68,119
1690 DATA 28,38,58,2,62,28,34,238
1700 DATA 24,60,60,60,24,36,102,0
1710 DATA 28,28,38,38,58,58,2,2,62,62,
28,28,34,34,238,238
1720 DATA 0,1,2,3,4,5,6,7,8,9,10,11,12
,13,14,15,16,17,18,19,20
1730 DATA 104,169,0,133,203,133,205,16
9,224,133,206,165,106,24,105,1,133,204
,160
1740 DATA 0,177,205,145,203,200,208,24
9,230,204,230,206,165,206,201,228,208,
237,96
```

•

CHECKSUM DATA
(See p. 30)

```
10 DATA 593,823,296,246,947,170,217,22
0,223,61,708,285,974,959,573,7295
160 DATA 326,531,555,835,258,723,867,1
94,213,387,665,518,215,25,321,6633
310 DATA 555,797,707,807,464,390,847,5
66,160,46,239,500,996,16,9,7099
460 DATA 5,2,768,129,310,702,468,395,9
42,124,943,125,613,588,139,6253
610 DATA 625,854,905,769,141,97,612,49
7,480,967,541,245,363,605,41,7742
760 DATA 159,451,928,98,113,598,481,46
6,346,44,128,272,619,289,450,5442
910 DATA 522,832,732,576,675,60,471,61
8,318,685,719,738,97,702,527,8272
1060 DATA 801,821,46,788,547,532,766,6
30,892,191,188,748,32,791,700,8473
1210 DATA 244,155,187,124,820,26,792,5
08,681,525,536,418,354,970,175,6515
1360 DATA 717,465,730,300,562,409,564,
677,866,636,536,443,726,648,724,9003
1510 DATA 734,225,971,219,235,321,362,
475,953,856,982,350,613,875,962,9133
1660 DATA 231,410,983,711,636,14,415,1
47,639,4186
```


RENT ATARI® BUY
EXCITEMENT

INDUS DISK DRIVES $349/$60 Mo.

400/800/810 Heavy Duty Dust Covers - $3.49

NEW & EXCITING

Ultima (D) $49
Asylum (D) $29
Pogo Joe (D) $29
Danger Ranger (D) $29
Dunzhin (C/D) $27
Kaiv (C/D) $27
Wylde (C/D) $27
Heracles (C/D) $33
Beneath Apple Manor (C/D) . $30

SPECIALS

	Sell	Rent
EPYX		
Armor Assault (D)	$14	$5
Crypt of Undead (D)	$ 9	$5
King Arthurs Heir (D)	$ 9	$5
Nightmare (D)	$12	$5
Temple of Apshai (D)	$16	$5
DATASOFT		
Graphic Master (D)	$15	$5
Micro Painter (D)	$15	$5
Shooting Arcade (C/D)	$12	$5
Canyon Climber (C/D)	$12	$5
SYNAPSE		
Shamus (C/D)	$12	$5
Protector II (C/D)	$11	$5
Nautilus (C/D)	$11	$5
Dodge Racer (C/D)	$ 9	$5

100's more as low as $3.49

Call toll-free outside Texas: 1—800—433-2938
— Inside Texas call: 817—292-7396

WEDGWOOD RENTAL
5316 Woodway Drive
Fort Worth, Texas 76133
CIRCLE #142 ON READER SERVICE CARD.

Double Byte

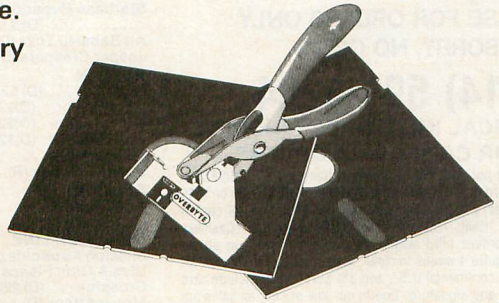
Why pay more for double-sided disks when you can use both sides of a single-sided disk. How? With OVERBYTE!

OVERBYTE is a self-aligning, heavy duty metal hold punch which doubles the capacity of a normal single-sided disk.

OVERBYTE punches a write-protect notch on the left side of a single-sided disk allowing the reverse side to be used.

OVERBYTE can be used with Apple, Rana, Franklin, Atari, Commodore, and Victor 9000 disk drives (and other compatible drives).

Disks are expensive. Double your memory at half the cost!



OVERBYTE ONLY $19.95

CIRCLE #143 ON READER SERVICE CARD.

Send To:

OVERBYTE

8621 Laurel Canyon Blvd.
Sun Valley, CA 91352
(213) 504-0309

Please rush me OVERBYTE for $19.95

Please include $3.00 for shipping & handling (California residents add 6½% sales tax.)

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE ( ) _____

☐ MASTERCARD ☐ VISA ☐ CHECK

☐ MONEY ORDER _____

CREDIT CARD NUMBER _____

EXPIRATION DATE _____

SIGNATURE (for charge cards) _____

ATARI®

600XL . . . . . CALL

1200XL . . . . . CALL *

*Reflects $100 Atari Rebate

1050 DRIVE	**$335**	**MICROBITS INFC**	**$78**
1025 PRINTER	**$399**	**80 COLUMN BD**	**$249**
1020 COLOR PTR	**$219**	**TECHNICAL NOTES**	**$25**
1027 PRINTER	**$279**	**REAL TIME CLK**	**$38**
1010 RECORDER	**$72**	**AXIOM PRINTER INFC**	**$80**
MICRO PERIF. MODEM/INFC			**$129**

DRIVES

ASTRA 1620	**$469**
RANA 1000	**$295**
PERCOM 88-S1	**$298**
PERCOM 40-S1	**$418**
PERCOM 40-S2	**$718**
PERCOM 44-S1	**$510**
PERCOM 44-S2	**$929**

MEMORIES

48K RAM (INTEC)	**$95**
64K RAM (INTEC)	**$119**
48K RAM (MOSAIC)	**$109**
64K RAM (MOSAIC)	**$145**
128K RAM DISK	**$299**
32K RAM (MOSAIC)	**$68**

ATARI SOFTWARE

ADVENTURE INT'L
Adv. 1-12 each (C) $18
Preppie (C/D) $20
Preppie II (C/D) $23
Diskey (D) $33
Sea Dragon (C/D) $23
Bug Off! (C/D) $20
Tutti-Frutti (C/D) $17
Saga 1-3 (ea.) (D) $27
Lunar Lndr . (C) $11 (D) $15
Galactic Empire (C) . . $14
Galactic Trader (C) . . . $14
Stratos (C/D) $23
Rear Guard (C) $14 (D) $17

APX
Eastern Front (C/D) . . $23
747 Land Sim. (C/D) . . $17
Fig-Forth (C) $30
Family Cash Flow (D) . $17
Downhill (C/D) $17
Avalanche (C/D) $17
Outlaw/Howitzer (C/D) $17
Salmon Run (C/D) $17
Data Management (D) $17
Galahad/Holy Grail (D) $21
Adv. Music System (D) $21

ATARI INC.
Microsoft Basic II (R) . $62
Mickey in Great Outdoors (C/D) $36
Paint (D) $30
Speed Reading (C) . . . $54
Qix (R) $30
Dig Dug (R) $30
Atari Writer (R) $68
Time Wise (D) $23
Visicalc (D) $139
Juggles House (C/D) . $22
Juggles Rnbw (C/D) . . $22
Pilot (Home) (R) $55
Galaxian (R) $30
Defender (R) $30
ET $34
Microsoft Basic (D) . . $62
Macro Ass. & Edit (D) $62
Assembler Editor (R) . $42
Basic Cartridge (R) . . . $45
Pac Man (R) $30
Centipede (R) $30
Caverns of Mars (D) . . $28
Star Raiders (R) $30
Conv. Lang. Ea. (C) . . . $42
Music Composer (R) . . $31
Super Breakout (R) . . . $26
My First Alphabet (D) . $26
Prog.2 & E (ea.)(C) . . . $21
Word Processor (D) . $102
Pilot (Educ.) (R) $92
Touch Typing (C) $19
Home File Mngr (D) . . . $36
Bookkeeper (D) $102
Family Finance (D) . . . $35
Prog. 1 (C) $18
Scram (C) $18
Asteroids (R) $27
Space Invaders (R) . . . $27
Missile Command (R) . $27
Telelink (R) $21
Superman III (R) $34
Basketball (R) $24
Donkey Kong (R) $34

AVALON HILL
VC (D) $17
B-1 Nuc. Bomber (C) . . $12
Legionnaire (C) $23
Empire of Overmind . . . . . (D) $23 (C) $20
Tanktics . . . (D) $20 (C) $17
Comptr Stock & Bonds . . . . . (D) $17 (C) $14
Dnieper River Line (D) $20
Voyager (D) $17
Controller (D) $20
GFS Sorceress . . . . . (D) $23 (C) $20
Telegard . . (D) $19 (C) $16
Vorrak (D) $18
Lord of Karma (C) $14

AVALON HILL(Cont'd)
Moon Patrol (C) $17
Planet Miners (C) $12
Conflict 2500 (C) $12
Space Station Zulu . . . . . (D) $17 (C) $14
Flying Ace . (D) $21 (C) $18
Gypsy . . . . . (D) $18 (C) $15
Galaxy . . . . (D) $17 (C) $14
Guns of Fort Def. (C) . $14
Nuke War (C) $12
Andromeda Conquest . . . . . (D) $16 (C) $13
Close Assit (D) $23 (C) $20

BRODERBUND
Sky Blazer (D) $22
Bank St. Writer (D) . . . $46
A.E. (D) $23
Arcade Machine (D) . . $39
Choplifter . (D) $23 (R) $29
Labyrinth (C/D) $20
Serpentine . (D) $23 (R) $27
Steller Shuttle (C/D) . . $17
Apple Panic (C/D) $20
Genetic Drift (C/D) . . . $20
David's Midnight (D) . . $23
Sea Fox (D) $20
Track Attack (D) $20
Operatn Whrlwnd (D) . $23
Match Boxes (C/D) . . . $20

CDY CONSULTING
Pogoman (C/D) $27
Mad-Netter (C/D) $23
It-Is-Balloon (C/D) $23

CBS
Mountain King (R) . . . $27
Boulders & Bombs (R) $27
Krazy (each) (R) $27

CONTINENTAL SOFT.
Home Accountant (D) $48
Tax Advantage (D) . . . $39

DATA MOST
Pig Pen (D) $20
Mating Zone (D) $20
Roundabout (D) $20
Bilestoad (D) $20
Night Raiders (D) $20
Monster Smash (D) . . . $20

DATASOFT
Text Wizard (D) $65
Graphic Master (D) . . . $27
Micro Painter (D) $23
Lisp Interpreter (D) . . . $79
Graphics Gen.(D) $17
Basic Compiler (D) . . . $65
Zaxxon (C/D) $27
Teletalk (D) $34
Pac. Coast Hwy (C/D) . $20
Clowns/Balloons (C/D) $20
Spell Wizard (D) $53
Canyon Climber (C/D) $20
Sands of Egypt (D) . . . $27
O'Riley's Mine (C/D) . . $23
Rosen's Brigade (C/D) $23
Moon Shuttle (D) $27

DON'T ASK
Sam (D) $39
P.M. Animator (D) $23
Teletari (D) $27
Poker Sam (D) $17
Word Race . (D) $17 (C) $14

EDU-WARE
Prisoner II (D) $27
Spelling Bee (D) $27
Rendevous (D) $27
Compu/Read . . . . . (D) $21 (C) $15
Compu/Math-FR . . . . . (D) $27 (C) $21
Compu/Math-Dec . . . . . (D) $27 (C) $21

EDUCATIONAL SOFT.
Tricky Tutorial 1,2,3 or 4 (C/D) $15
Tricky Tutorial 5,6 or 7 (C/D) $22

Printers/Etc.

GEMINI 10X	**$279**	**PROWRITER**	**$345**
GORILLA	**$199**	**SMITH TPI**	**$488**
CITOH		**SILVER REED P**	**$669**
Prowriter	$345	**QUME 11/40 +**	**$1299**
Prowriter II	$629	**OKI-DATA**	
Starwriter	$1149	Microline 82A	CALL
Printmaster	$1448	Microline 83A	CALL
NEC		Microline 84P	CALL
8023 A-C	$409	Microline 92	CALL
3510	$1375	Microline 93	CALL
3530	$1579	**DIABLO**	
3550	$1779	620R	$939
7710/7730	$1998	630R	$1719

MONITORS

AMDEK		**NEC**	
Color I	$289	GRN (JB1260)	$115
V300	$139	GRN (JB1201)	$155
V300A	$149	Color Composite	$298
Color II	$449	RGB Color	$598

MODEMS

HAYES		**NOVATION**	
Smartmodem	$209	J-Cat	$99
Smartmodem 1200	$498	Apple Cat II	$259
Micromodem II	$259	D-Cat	$149
		ANCHOR AUTOMATION	
		Mark I or II Modem	$78

SPECIALS

Gemini 15X Printer	$399
Axiom AT-100 Printer (with interface	$229
Astra Double Density Dual Drive	$469
Rana 1000 Drive	$319
Bit-3 80 Column Board	$245
Mannesmann Talley 160L Printer	$589
Atari 400 Keyboard (In Home)	$35
Programmer Kit . . $48	Entertainer Kit . . . $64
Wico Joystick . . . $23	Wico Trackball . . . $49

COSMIC COMPUTERS
UNLIMITED

727 BREA CANYON RD., SUITE 16
WALNUT, CA 91789
ORDER LINES OPEN MON-SAT 8 am - 8 pm

(800) 626-7642

PLEASE FOR ORDERS ONLY
SORRY, NO COD'S

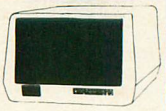
(714) 594-5204

FOR TECHNICAL INFO, ORDER INQUIRIES, OR FOR CALIFORNIA ORDERS

Add $2.50 shipping per software order in continental U.S. Add $5.00 shipping per software order for AK, HI, FPO-APO. Add $10.00 or 15% (whichever is greater) per software order for non-U.S. Call for cost of hardware shipping. Calif. residents add 6½% sales tax. Cashiers checks or money orders filled within 24 hours for items in stock. Personal checks require 4 weeks to clear. MasterCard and Visa OK for software only within continental U.S., add 3% surcharge. Include card no., expiration date and signature. Due to our low prices, all sales are final. All defective returns must have a return authorization number. Please call to obtain one before returning goods for replacement or repair. Prices & availability subject to change.

CIRCLE #144 ON READER SERVICE CARD.

*ASTRA 1620
DISK DRIVE SYSTEM



MORE FOR YOUR MONEY
DOUBLE OR SINGLE DENSITY
TWO DRIVES

SPECIAL $469

EPYX-AUTO. SIMULATION
Rescue at Rigel (C/D) . $20
Datestones/Ryn (C/D) $14
Upr Rchs Apshai (C/D) $14
Ricochet (C/D) $14
Keys/Acheron (C/D) . . $14
Curse of Ra (C/D) $14
Danger/Drindisti (C/D) $14
Jumpman (C/D) $27
Escp Vulcans Isle (D) . $20
Crypt of Undead (D) . . $20
Nightmare (D) $20
Armor Assault (D) $27
Monster Maze (R) $27
Alien Garden (R) $27
Plattermarna (R) $27
Morloc's Tower (C) . . . $14
King Arthur's Heir . . . . . (D) $20 (C) $14
Hellfire Warrior (C/D) . $27
Invasion Orion (C/D) . . $17
Temple of Aps. (C/D) . $27
Star Warrior (C/D) $27
Dragon's Eye (D) $20
Crush Crumble (C/D) . $20

GEBELLI
Firebird (R) $27
Embargo (R) $30
Candy Factory (D) $20
Match Racers (C/D) . . $20
Dr. Goodcode (D) $20

HAYDEN
Go . . . . . (D) $23 (C) $20
Sargon II . . (D) $23 (C) $20
Bulldog Pinball (C) . . . $20

HES
Coco (C/D) $34
Coco II (C/D) $27

INFOCOM
Suspended (D) $34
Zork I, II or III (D) $27
Starcross (D) $27
Deadline (D) $34
Witness (D) $34

INHOME
Baseball . . (R) $23 (D) $23
Crypts of Terror . . . . . (D) $23 (C) $20

INNOVATIVE DESIGN
Pool 1.5 (D) $23
Juggler (D) $20
Speedway Blast . . . . . (D) $20 (R) $27
Pool 400 (R) $27

JV SOFTWARE
Jrny to Plnts (C/D) $20
Action Quest (C/D) . . . $20
Ghost Encount. (C/D) . $20

LJK
Letter Perfect (D) . . . $104
Data Perfect (D) $74
Letter Perfect (R) . . . $137
Edit 6502 (R) $137

MONARCH DATA SYS.
ABC Compiler (D) $48

ODESTA
Chess (D) $45
Checkers (D) $34
Odin (D) $34

OPTIMIZED SYSTEMS
C-65 (D) $58
Bug-65 (D) $23
Max-65 (D) $58
Basic A + (D) $58
Action (D) $65
Speed Read Plus (D) . . $40

PHOENIX
Birth of Phoenix (D) . . $16
Adv. In Time (D) $20

QUALITY
Fastgammon (C) $14
Name That Song . . . . . (D) $13 (C) $11
Starbase Hyperion . . . . . (D) $17 (C) $14
Ali Baba/40 Thvs (D) . . $20
Jeeper Creepers (D) . . $20

ROKLAN
Gorf . . . . . (D) $27 (R) $30
Wizard/Wor . (D) $27 (R) $30
Dlx Invaders (D) $23 (R) $27
Anti Sub Ptrl (D) $20 (C) $14
Telecom (D) $49
Space Journey (R) . . . $30

SENTIENT
Gold Rush (D) $23
Cyborg (D) $23

SIERRA ON-LINE
Mission Asteroids (D) $17
Ulys.& Gldn Fleece (D) $27
Crossfire . . (D) $20 (R) $23
Mouseattack (D) $23
Jawbreaker (D) $20 (R) $23
Threshold (D) $27
Softporn (D) $20
Ultima II (D) $39

SIERRA ON-LINE
Marauder (D) $23
Lunar Leeper (D) $20
Wiz & Princess (D) . . . $22
Frogger (C/D) $23

SIRIUS
Alpha Shield (R) $27
Wavy Navy (D) $23
Bandits (D) $23
Space Eggs (D) $20
Sneakers (D) $20
Way Out (D) $27
Type Attack (D) $27
Repton (D) $27
Critical Mass (D) $27
Fast Eddy (R) $23
Worm War (R) $23

SPINNAKER
Snooper Troop 1,2 (D) . $30
Kindercomp (D) $20
Rhymes & Riddles (D) $20
Hey Diddle Diddle (D) . $20
Srch Amzng Thngs (D) $27
Story Machine (D) $23
Face Maker (D) $23

STRATEGIC SIM.
Cosmic Balance (D) . . $27
Cosmic Balance II (D) $27
Tigers In Snow (C/D) . . $27
Battle of Shiloh (C/D) . $27
Battle of Norm. (C/D) . $27
Cytron Masters (D) . . . $27
Knights/Desert (C/D) . $27
Combat Leader (C/D) . $27

SUBLOGIC
Pinball (C/D) $20

SWIFTY
J. White Music Lessons . . . . . (C/D) $20
Space Shuttle (D) $20

SYNAPSE SOFTWARE
File Mngr 800 + ... $65
Protector II (D) $23 (R) $29
Shamus . . . (D) $23 (R) $29
Fort Apocalypse (C/D) $23
Shamus II (C/D) $23
Necromancer (C/D) . . . $23
Pharoh's Curse (C/D) . $23
Slime (C/D) $23
Page 6 (D) $23
Reptilian (C/D) $23
Picnic Paranoia (C/D) . $23
Claim Jumper (C/D) . . $23
Drelbs (C/D) $23
Shadow World (C/D) . . $23
Survivor (C/D) $23
T.N.T. (C/D) $23
N.Y.C. (C/D) $23
Slamball (C/D) $23
Blue Max (C/D) $23
Nautilus (C/D) $23
Chicken (R/D) $23
Quasimodo (C/D) $23
River Quest (C/D) $23

SYNERGISTIC
Programr Wkshp (D) . . $23
Graph Workshop (D) . . $27
Disk Workshop (D) . . . $23
Worlock's Revenge (D) $23

THORN EMI
Soccer (R) $34
Jumbo Jet (R) $34
Submarine Comm. (R) $34
Kick Back (R) $34
Pool (C) $22
Humpty Dumpty (C) . . $22
Hckry Dckry Dck (C) . . $22
Darts (C) $22

USA
Atari World (D) $39
3-D Sprgrphcs (C/D) . . $27
Survival Adv. (C/D) . . . $17

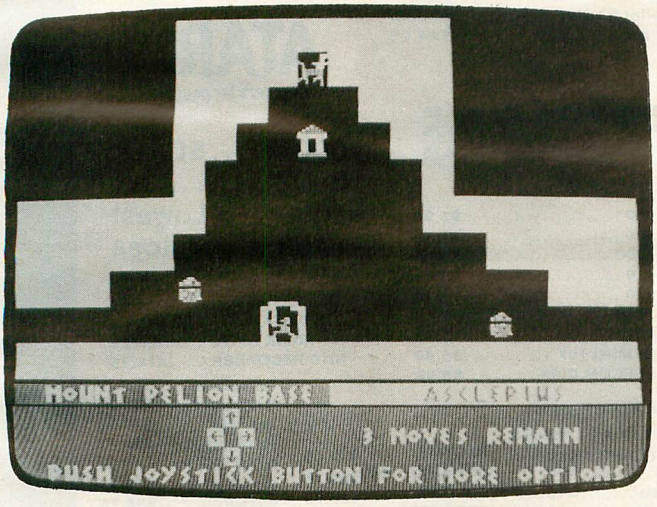
VERSA
Mind Bgglrs I (D) $14 (C) $12
Globe Master (D) $20

MISCELLANEOUS
Financial Wizard (D) . . $41
Castle Wolfenstein (D) $20
Master Type (D) $27
Astro Cha[illegible] (D) $22
Miner 2049e[illegible] $34
Cypher Bowel (C) $33
Bug Attack (C/D) $20
Raster Blaster (D) $20
Air Strike (C/D) $27
Prism (D) $18
Megalegs (C) $23
Cap'n Cosmo (D) $19
Spy's Demise (D) $14
Galac. Chase (D) $20 (C) $18
Kid Grid (C/D) $20
Battle Trek (D) $20
Snapper (D) $19
T&F-PMP Property Management (D) . . $149
Millionaire (D) $39

THE RETURN OF HERACLES
by Stuart Smith
QUALITY SOFTWARE
6660 Reseda Blvd., Suite 105
Reseda, CA 91335
48K Disk $32.95

by Michael Des Chenes

I have never been interested in adventures. If you were to ask me what my favorite adventure is, I would have to say **Ali Baba** from Quality Software. I heard rumors that Stuart Smith, author of **Ali Baba**, was working on a similar adventure with more features. I didn't hear anything from Quality Software for over a year, so I figured that I would never see their next adventure. Without warning, a package arrived at the **ANALOG** offices from Quality Software, **The Return of Heracles**. Finally, an adventure that even I could get excited about!



The Return of Heracles.

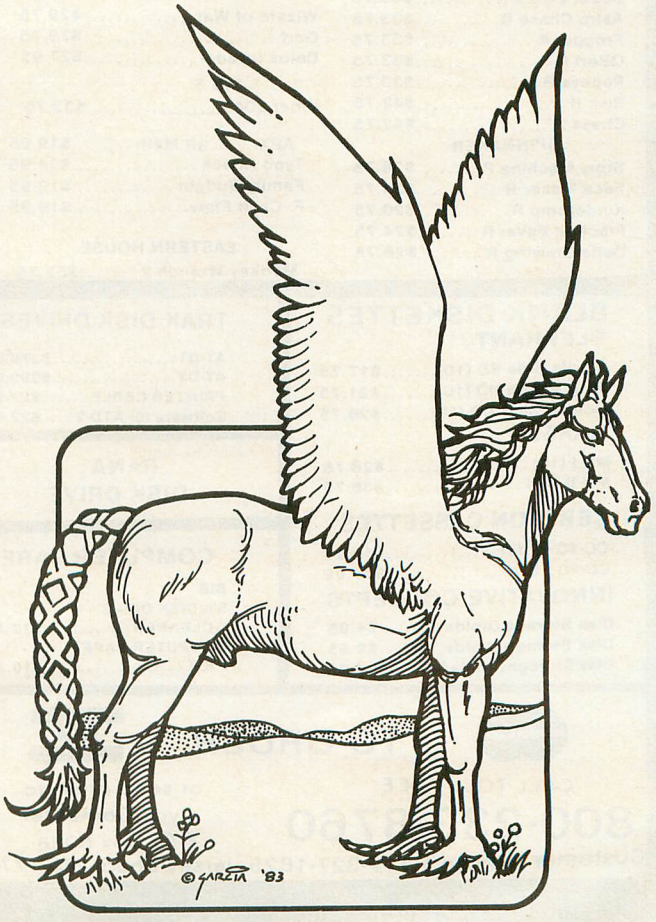
Heracles is a one to four player graphics adventure using only the joysticks for control. The game is an exploration of Greek mythology. Each human player takes on the role of up to four of the over twenty available heroes, heroines, or creatures. You move about in the world of mythic Greece, earning fame and fortune, sharpening your abilities, fighting strange beasts, and trying to complete the twelve tasks given to you by Zeus, powerful Father of the Heavens.

Many of the players may work together, taking turns. If your player is killed you may add any of the other available players at any point in the game. There is a total of 247 characters in the game. Each character has its own personality, shown in a character table in the instruction book. Strength, vigor, dexterity, speed, target size, armor, melee and hand-to-hand combat ratings are different for each character. For example, an Adder has the strength of 7 and the speed of 2, but a Boar has the same strength but a higher speed of 10. Choose your characters wisely and you'll live longer.

Your character can become stronger and more agile during the game by collecting money, finding a trading outpost and purchasing weapons, poisons and training. But don't get to greedy! The more money you carry around with you, the slower your player moves because of the added weight.

With such a larger number of characters to choose from, you'll never get bored playing the game over and over. I find the game to be more enjoyable than most adventures because of the ability to play with or against another human player. Zeus tells you how well you're doing each time you complete one of his tasks. Each task is worth a certain number of points. Finishing all twelve tasks gives you an additional bonus and completes the game.

Even if adventures bore you, I think you'll find **The Return Of Heracles** an entertaining game. If you're already hooked on adventures, I think you'll like this new twist in adventure gaming. The graphics, sound, playability and the amount of research into Greek mythology that went into this game are top notch. □



Lyco Computer Marketing & Consultants

TO ORDER CALL US — **TOLL FREE 800-233-8760** — In PA 1-717-327-1824

PRINTER INTERFACING — Available for IBM PC, Apple, Atari, Vic 20 & Vic 64

PRINTER PAPER AVAILABLE

SAVE on these in-stock PRINTERS

LETTER QUALITY

SMITH CORONA TP2...$449.00

DIABLO 630 ..$1719.00

ALPAHCOM 42 .....$89.00
ALPHACOM 81 .....$129.00
NEC 8023 .....$369.00
NEC 8025 .....$699.00
NEC PC-8200 COMPUTER .....$CALL

EPSON

RX-80 .....$SAVE$
RX-80FT.....ON
FX-80 .....In-Stock
FX-100.....EPSON
MX-80FT .....PRINTERS
MX-100.....$$CALL$$

MANNESMANN TALLY

SPIRIT 80 .....$CALL
MT 160L.....$CALL

OKIDATA

80 .....$SAVE$
82A.....CALL for
83A.....LOWEST
84 .....PRICES
92 .....on these
93.....In-Stock
PACEMARK 2350...PRINTERS

ATARI 850 REPLACEMENTS IN-STOCK

CITOH

GORILLA GX100 .....$179.00
PROWRITER 8510 ...$339.00
PROWRITER II $659.00
8600 .....$1025.00
STARWRITER.....$1099.00
PRINTMASTER $1499.00

STAR MICRONTICS

GEMINI 10X.....$269.00
GEMINI 15X.....$CALL
DELTA 10.....$479.00

MODEMS

ANCHOR MARK I ..$79.00
ANCHOR MARK II..$79.00
HAYES SMART ...$239.00
HAYES MICRO II $309.00

Micro Bit
MPP-1000..... $129.75

NOVATION
CAT $144.00
D-CAT..... $155.00
J-CAT $115.00
APPLE CAT II.... $279.00
212 APPLE CAT . $589.00

MONITORS

Sakata Color $229.00
Amdek Color I $275.00
Amdek 300 Green..... $149.00
Amdek 300 Amber $149.00
Gorilla Green $99.00

PARKER BROTHERS

Tutankham R.....$33.75
Super Cobra R $33.75
Astro Chase R.....$33.75
Frogger R $33.75
QBert R $33.75
Popeye R.....$33.75
Risk R $42.75
Chess R.....$42.75

SPINNAKER

Story Machine R $26.75
Face Maker R $24.75
Kinderomp R.....$20.75
Fraction Fever R $24.75
Delta Drawing R.....$26.75

SSI

Battle of Shilo C/D.....$26.75
Tigers in the Snow C/D....$26.75
Battle for Normandy C/D ..$26.75
Knights of the Desert C/D .$26.75
Cosmic Balance C/D $26.75

ON-LINE

Frogger $24.95
Wizard & Prin $26.95

ROKLAN

Wizard of War.....$29.75
Gorf $29.75
Delux Invader $27.95

BIG 5

Miner 2049.....$32.75

APX
Typo Attack $24.95
Family Budget $19.95
F. Cash Flow..... $19.95

3R Math..... $19.95

EASTERN HOUSE

Monkey Wrench 2 .....$52.75

BRODERBUND

Bank Street Writer D..... $49.75
AE D.....$24.75
Apple Panic D.....$23.75
Choplifter ROM $32.75
David's Midnight.....$24.75
Stellar Shuttle C/D.....$18.75
Ft. Apocalypse.....$24.75

SYNAPSE

BLUE MAX C/D $24.75
Ft. APOCALYPSE C/D...$24.75
PHAROAH'S CURSE C/D...$24.75

DUST COVERS

800 .....$3.99
400 .....$3.99
1200.....$3.99
410 .....$3.99
810 .....$3.99
1050.....$5.99
PROWRITER $5.99
GEMINI 10X.....$5.99
PERCOM DISK $5.99

ALIEN GROUP

Voice Box 2 . $99.75

DON'T ASK
Sam $41.75
Abuse $15.95
Teleatri.....$27.95
Poker Sam . .$24.95

PACMAN..... $29.75
DONKEY KONG $29.75
DIG DUG..... $29.75
DEFENDER.... $29.75
CENTIPEDE ... $29.75

EPYX

GATEWAY TO ASPHI R.....$28.75
JUMPMAN JR R $28.75
PIT STOP R $28.75
GATEWAY TO

ATARI®
Computers for people.

600XL ...$CALL
800XL.....for
1400XL...Lowest
1450 .....Prices

1020 PRINTER.....NOW
1025 PRINTER.....IN
1027 PRINTER .....STOCK
1050 DISK DRIVE ....$SAVE$
1010 RECORDER.....$74.75

DEADLINE ..$34.75
ENCHANTER ..$34.75
INFIDEL .$34.75
PLANETFALL ..$34.75
STAR CROSS..$34.75
SUSPENDED ..$34.75
WITNESS ...$34.75
ZORK I ...$34.75
ZORK II $34.75

BLANK DISKETTES

ELEPHANT

Single Side SD (10).....$17.75
Single Side DD (10).....$21.75
Double Side DD (10).....$26.75

MAXELL

MD I (10).....$28.75
MD II (10) $38.75

CERTRON CASSETTES

CC-10 12 for $15.99
CC-20 12 for $17.99

INNOVATIVE CONCEPTS

Disk Storage (holds 10)....$4.95
Disk Storage (holds 15)....$9.95
Disk Storage (holds 50)....$26.95

TRAK DISK DRIVES

AT-D1 .....$379.00
AT-D2.....$399.00
PRINTER CABLE $22.95
Software for ATD-2 ....$22.95

RANA DISK DRIVE

COMPUTER CARE

BIB
5¼ DISK DRIVE CLEANER.....$12.75
COMPUTER CARE KIT $19.75

HARD DISK DRIVES for APPLE IBM-PC

5MEG.....$1349.00
10MEG ...$1599.00
15MEG ...$1999.00
20MEG ...$2359.00

*Add $ 30.00 for TRS 80 Drives

TEXAS INSTRUMENT

Disk Drive...$245.00

PERCOM

FOR ATARI COMPUTERS

AT88S1 ....$329.00
AT88S2 ....$535.00
AT**S1PD...$439.00
RFD40S1...$399.00
RFD40S2...$675.00
RFD44S1...$449.00
AT88 doubler

MasterCard — VISA

TO ORDER

CALL TOLL FREE
800-233-8760
Customer Service 1-717-327-1825

or send order to
Lyco Computer
P.O. Box 5088
Jersey Shore, PA 17740

POLICY

In-stock items shipped within 24 hours of order. Personal checks require four weeks clearance before shipping. No deposit on C.O.D. orders. Free shipping on prepaid cash orders within the continental U.S. PA residents add sales tax. All products subject to availability and price change. Advertised prices show 4% discount offered for cash, add 4% for Master Card or Visa. DEALER INQUIRIES INVITED.

Introduction to Action! Part 2.

by Clinton Parker

Part I of this series presented a brief introduction of Action! data types and control structures using a small example program. In this part, I will expand on that example to demonstrate the use of ARRAYs in the Action! language, and increase the speed at which it runs.

This increase in speed is accomplished by providing a specialized PLOT routine instead of using the one provided in the cartridge library. The PLOT routine in the cartridge (the same one used by the OS) was written to be very flexible so that it could handle all the different graphics modes and check for illegal values. The problem with this generality is that it doesn't plot points on the screen all that fast. Since all the points plotted in KAL are in graphics mode 24, it seems reasonable to write a PLOT routine just for that mode.

All right, we now see that having our own PLOT routine would be useful, but how do we go about writing one? First, we'll start by looking at how the Atari represents graphics mode 24 data by means of a simple example. Imagine a small piece of graph paper 24 by 12. Label the top left square 0,0 and the bottom right square 23,11. Draw a line from top to bottom between squares 7 & 8 and 15 & 16, and then number these divisions starting with 0,1,2 for the first line; 3,4,5 for the next line (1) and ending with 33,34,35 for the last line (11). What you should have is **Figure 1.** Except for the screen being much larger, this is exactly how the Atari generates a graphic 24 display. Each 8 square division on the graph paper represents an 8-bit byte of memory.

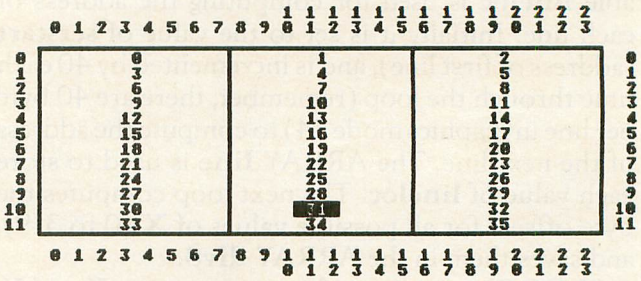


Figure 1.

If we plot point 10,10 on our sheet of graph paper, we note that it is in division 31 and is the 2nd square of that division (first square of a division is 0). The computer does a similar calculaton when we tell it to plot point 10,10. It first determines which byte of the screen memory we want and then it determines which bit in that byte is to be set.

Now this isn't as hard as it looks, because there are several tricks that can be used to make these calculations simple. We can calculate the offset of the first division (byte) of each line by multiplying the number of divisions (3 for our example, 40 for a graphics 24 display) by the line number. We can then calculate which division (byte) we want on that line by dividing the column by 8 (8 spaces per section, 8 bits per byte). Finally, we can compute which square (bit) is to be changed by the remainder of this division. Thus, for 10,10 example we have:

line offset = 30 (10*3)
division offset = 1 (10/8)
square offset = 2 (10 MOD 8)

We now have enough information to design our PLOT routine. Remember that we are writing our own routine to increase the speed of plotting points. Multiplication and division are slow operations, so if we avoid doing these operations when we are plotting, it will greatly increase the speed of our plot routine. As turns out, we can avoid doing these operations by precomputing the line offsets and byte offsets at the beginning of the program and then use those offsets in our plot routine. We do this by storing the precomputed offsets in ARRAYs. In the plot routine, we'll use Y as an index into the line offset ARRAY (line) and X as an index into the byte offset ARRAY (div8).

Walking through.

The PROCedure **Init()** is responsible for generating the precomputed line and byte offsets. It starts by setting up the display with:

```
Graphics(24)
SetColor(1,0,14) : SetColor(2,0,0)
```

The next block of code computes the line offsets (192 of them for graphics mode 24). The variable **scrstart** is defined to be location 88. This location contains the starting address of the screen. The variable **lineloc** is used for computing the address of each line. Initially it is set to the value of **scrstart** (address of first line), and is incremented by 40 each time through the loop (remember, there are 40 byte per line in graphics mode 24) to compute the address of the next line. The ARRAY **line** is used to store each value of **lineloc**. The next loop computes the byte offsets for all possible values of **X** (0 to 319), and saves them in the ARRAY **div8**.

PROC **Plot()** is passed two arguments, **X** and **Y**, which define the point to be plotted. The byte that is to be modified on the screen is computed by adding the line address of **Y** to the byte offset of **X** as follows:

```
pos = line(Y) + div8(X)
```

The BYTE POINTER **pos** now contains the address of the byte we want to modify. Next, we determine if we are plotting a point or erasing one by:

```
IF color 0 THEN
```

If **color** is non-zero, we want to plot a point. This is done by setting the correct bit of the byte pointed to by **pos**. This is what

```
pos^ == % m1(X&7)
```

does. This may look very complicated, but it isn't. X&7 computes which bit is to be modified (same as X MOD 8, but much faster). This is used as the index for the ARRAY **m1**. ARRAY **m1** is declared to contain a set of 8 masks. Each mask represents the bit to be modified for that index. Thus, when **mi(X&7)** is or'ed into the byte pointed to by **pos**, it sets only the bit to be plotted without affecting the other bits of that byte.

In a similar manner, if **color** is zero

```
pos^ == & m2(X&7)
```

erases point **X,Y** on the screen. ARRAY **m2** is declared to contain 8 masks which, when and'ed with the byte pointed to by **pos**, erase a single bit without effecting the other bits of that byte.

Using this **Plot** routine instead of the built-in routine increases the execution speed of **Kal** by about a factor of 3. Since none of the X values used in **Kal** exceeds 255, you can change the declaration of **Plot** to be:

```
PROC Plot (BYTE x, y)
```

This will make this version of **Kal** run about 4 times faster than using the built in **Plot** routine, but it will no longer work for all legal values of **X**.

If you haven't followed all of this, don't worry. I didn't go into any details about bit-wise operations (& and %) to keep the description brief. You can still enjoy the results (assuming you have an Action! cartridge). You can even use these two PROCs (**Init** and **Plot**) in other programs that you write yourself. □

Listing 1.

```
;          KAL.ACT

; Copyright 1984 BY Clinton Parker
; All Rights Reserved

; last modified February 18, 1984

TYPE REC=[CARD cnt,ax,bx,cx,ay,by,cy]
REC p, e
CARD period, npts, persistence

CARD ARRAY line(192)
BYTE ARRAY div8(320)
BYTE ARRAY m1(0)=[128 64 32 16 8 4 2 1]
BYTE ARRAY m2(0)=[$7F $BF $DF $EF $F7 $F
B $FD $FE]

PROC Plot(CARD x, BYTE y)
  BYTE POINTER pos

; get address of byte to modify
  pos = line(y) + div8(x)

; modify only one bit of that byte
  IF color#0 THEN ; plot
    pos^ ==% m1(x & 7)
  ELSE ; erase
    pos^ ==& m2(x & 7)
  FI
RETURN

PROC Init()
  CARD i, scrstart=88
  BYTE POINTER lineloc
```

Put a Monkey Wrench into your ATARI 800

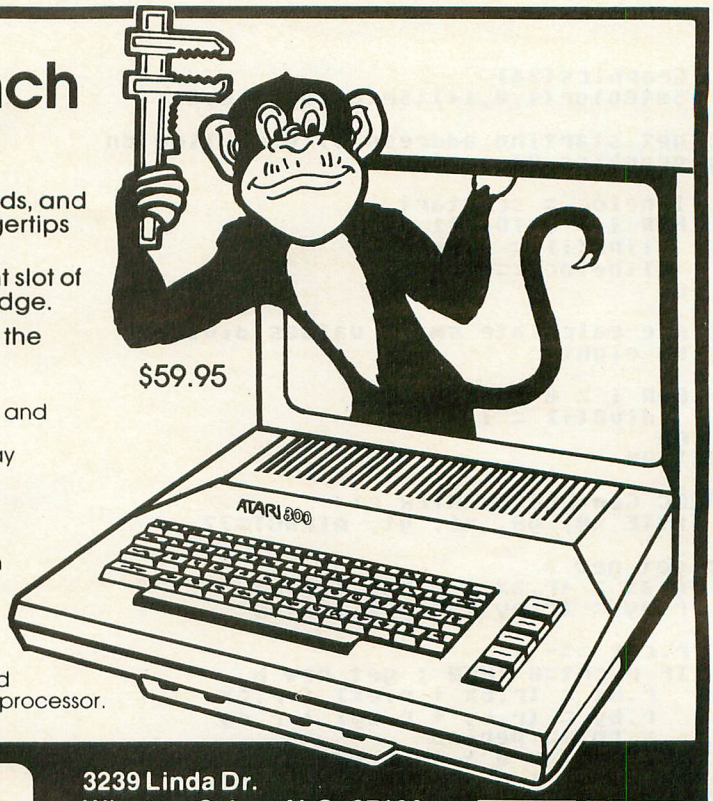
Cut your programming time from hours to seconds, and have 18 direct mode commands. All at your fingertips and all made easy by the MONKEY WRENCH II.

The MONKEY WRENCH II plugs easily into the right slot of your ATARI and works with the ATARI BASIC cartridge.

Order your MONKEY WRENCH II today and enjoy the conveniences of these 18 modes:

- Line numbering
- Renumbering basic line numbers
- Deletion of line numbers
- Variable and current value display
- Up and down scrolling of basic programs
- Location of every string occurrence
- String exchange
- Move lines
- Copy lines
- Special line formats and page numbering
- Disk directory display
- Margins change
- Memory test
- Cursor exchange
- Upper case lock
- Decimal conversion
- Machine language monitor
- Hex conversion

The MONKEY WRENCH II also contains a machine language monitor with 16 commands that can be used to interact with the powerful features of the 6502 microprocessor.



Eastern House

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

VISA master card THE INTERBANK CARD

CIRCLE #146 ON READER SERVICE CARD.

ATARI

A LIFETIME OF DISCOVERY BEGINS WITH ATARI COMPUTERS

1-800-824-7506

DISK DRIVES

RANA 1000	CALL
TRAK AT1 (Single/Double Density)	CALL
TRAK ATD1 (Single Density/Parallel Interface)	CALL
TRAK ATD2 (Single/Double Density/Parallel Interface)	CALL
ASTRA 1620	CALL
INDUS GT	CALL

RAM (MEMORY) BOARDS

ATARI 16K	$ 29.00
Intec 64K Board (400)	109.00
Intec 48K Board (400)	86.00
Intec 32K Board (400)	59.00
Microbits 64K (600)	129.00

MC
VISA
C.O.D.
(Add $2.50)

COMPUTER CREATIONS, Inc.
P.O. Box 292467
Dayton, Ohio 45429
For Information Call:
(513) 294-2002
(Or to order in Ohio)

All Orders Add $3.00 Shipping and Handling. Ohio Residents Add 6% for Sales Tax.
ASK FOR OUR FREE CATALOG

PRINTERS

Gemini STX-80 (80 Column Thermal)	$169.00
Gemini 10X Printer (80 Column)	279.00
Gemini 15 Printer (136 Column)	399.00
Delta 10 Printer (80 Column)	499.00
Powertype	379.00
Radix 10	649.00
Radix 15	769.00
Epson RX-80 (80 Column) Dot Matrix	379.00
Epson RX-80FT (80 Column)	469.00
Epson FX-80 (80 Column) Dot Matrix	579.00
Epson FX-100 (136 Column) Dot Matrix)	749.00
Axiom AT-100 (80 Column) Inc. Int. Cable	229.00
Silver-Reed EXP 500 Daisy Wheel Printer (80 Column)	429.00
Silver-Reed EXP 550 Daisy Wheel Printer (136 Column)	679.00

PRINTER INTERFACE CABLES

APE FACE	$ 65.00
MPP-1150	79.00
Star Universal	69.00

MONITORS

Gorilla™ Hi Res 12" Non-Glare Screen	$ 85.00
Gorilla™ Hi Res 12" Non-Glare Amber Screen	99.00
BMC Green Screen	79.00
Sakata SC100 Color Screen	249.00
Monitor Cable	12.00

GENERIC DISKS

Generic 100% Defect-Free/Guaranteed MINI-FLOPPY DISKS

Diskettes (1 Box Min.) - 10 per box			Bulk Diskettes with Sleeves - Price per Disk		
	SS/DD	DD/DD		SS/DD	DD/DD
1 or 2 Boxes	17.49/box	20.99/box	10-29	1.59	1.99
3 - 9 Boxes	15.99/box	19.99/box	30 - 99	1.49	1.89
10+ Boxes	14.99/box	18.99/box	100+	1.45	1.79

ATARI HOME COMPUTERS

ATARI 600XL™ Home Computer (16K RAM)	CALL
ATARI 800XL™ Home Computer (64K RAM)	CALL
ATARI 1200XL™ Home Computer (64K RAM)	CALL
ATARI 1400XL™ Home Computer (64K RAM)	CALL
ATARI 1450XLD™ Home Computer (64K RAM)	CALL

ATARI ADDITIONAL EQUIPMENT ..... CALL

MODEMS

MPP 1000C	$129.00
Signalman Mark II Modem	79.00
Hayes Stack Smartmodem (300 BAUD)	239.00
Hayes Stack Smartmodem (1200 BAUD)	549.00
Novation J CAT	119.00
Novation 103 Smart CAT	189.00
Novation Auto CAT	199.00

CIRCLE #147 ON READER SERVICE CARD.

```
  Graphics(24)
  SetColor(1,0,14):SetColor(2,0,0)

; get starting address of each line on
; graphics 24 screen

  lineloc = scrstart
  FOR i = 0 TO 191 DO
    line(i) = lineloc
    lineloc ==+ 40
  OD

; pre-calculate small values divided
; by eight

  FOR i = 0 TO 319 DO
    div8(i) = i / 8
  OD
RETURN

PROC Gen(REC POINTER r)
  BYTE x0, y0, x1, y1, ATRACT=77

; get new a
  r.ax = (r.ax + r.bx) ! r.bx
  r.ay = (r.ay + r.by) ! r.by

  r.cnt ==- 1
  IF r.cnt=0 THEN ; get new b
    r.bx = (r.bx + r.cx) ! r.cx
    r.by = (r.by + r.cy) ! r.cy
    r.cnt = period
    ATRACT = 0 ; turn off attact mode
  FI

  x0 = r.ax RSH 9
  y0 = r.ay RSH 9
  IF x0<=y0 AND y0<96 THEN
    x1 = 191 - x0
    y1 = 191 - y0
    Plot(x0+64, y0):Plot(x0+64, y1)
    Plot(y0+64, x0):Plot(y0+64, x1)
    Plot(x1+64, y0):Plot(x1+64, y1)
    Plot(y1+64, x0):Plot(y1+64, x1)
  FI
RETURN

PROC Kal()
  CHAR CH=764

  Init()

; change for different patterns:
  persistence = 2500
  period = 10000  p.cnt = period
  p.ax= 5221  p.bx=64449  p.cx=3
  p.ay=57669  p.by=64489  p.cy=3

; copy plot record to erase record
  MoveBlock(e, p, REC)

; handle persistence
  color = 1
  FOR npts = 1 TO persistence DO
    Gen(p)
  UNTIL CH#255 OD

; draw patterns until key drepressed
  WHILE CH=255 DO
    color = 1   Gen(p)
    color = 0   Gen(e)
  OD

; ignore key and restore screen
  CH = 255 : Graphics(0)
RETURN
```


VERY LOW PRICES GET YOUR ATTENTION
VERY GOOD SERVICE KEEPS IT

AT RCE WE NOT ONLY PROVIDE OUR PATRONS WITH LOW PRICES . . . WE BACK THEM WITH SUPPORT! ! ! FACTORY AUTHORIZED SERVICE CENTER SUPPORT FOR OVER TWENTY DIFFERENT BRANDS OF HOME ELECTRONICS INCLUDING . . . ATARI, FOURTH DIMENSION, MICRO-SCI, SANYO, FRANKLIN, PANASONIC AND U.S. PIONEER. APPLE WARRANTY SERVICE AVAILABLE.

FACTORY AUTHORIZED SERVICE COMBINED WITH PRICES LIKE THESE:

ATARI HARDWARE

	LIST	RCE
600XL COMPUTER	$199	$CALL
800XL COMPUTER	$299	$CALL
1400XL COMPUTER	$599	$CALL
1450XL COMPUTER	$999	$CALL
1010 RECORDER	$100	$75
810 DISK DRIVE	$599	$429
850 INTERFACE	$219	$165
1020 PRINTER	$299	$239
1025 PRINTER	$549	$439

MONITORS

GREEN SCREEN
BMC 12" $89
ZENITH 12" $99
SANYO 12" $205

BLACK & WHITE
SANYO 9" $145
SANYO 12" $189

AMBER SCREEN
ZENITH 12" $115
AMDEK 12" $179

PRINTERS

STAR MICRONICS
GEMINI 10 ..... $CALL
GEMINI 15 ..... $CALL

MODEMS
HAYES SMARTMODEM 300 .. $209
MICROBITS MPP 1000 ..... $169

INTERFACES
MICROBITS MPP 1100 ..... $89

DISK DRIVES
RANA $319

RCE COMMANDER 2400

SPECIFY 400 or 800 version
2400 - 1 ..... $199
2400 - 2 ..... $169

OUR PRICES ARE ALWAYS GOING DOWN
CALL FOR LATEST REDUCED PRICE

SEND FOR RCE'S FREE ATARI HARDWARE AND SOFTWARE CATALOG

WE ALSO CARRY A FULL LINE OFAPPLE/FRANKLIN AND IBM COMPATIBLE SOFTWARE!!

RCE

ORDER TOLL-FREE
800-547-2492
IN OREGON: (503)479-4711

RALSTON CLEARWATERS ELECTRONICS
536 N.E. 'E' STREET GRANTS PASS, OR 97526
ALL BRANDS ARE REGISTERED TRADE MARKS
FOR CUSTOMER SERVICE CALL: (503)479-4711 or (503)479-4150

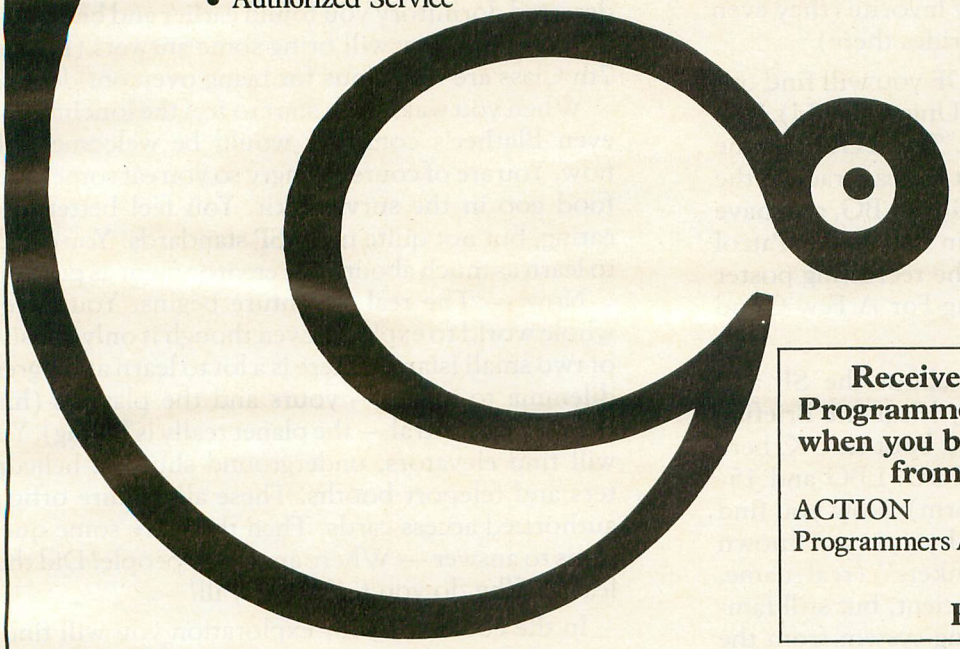
TERMS:
SHIPPING: Add 6% of total transaction for UPS brown (ground) or 9% for UPS blue (air), Parcel Post, or any special arrangements. Minimum shipping charge - $6.00
PAYMENT: Cashier's checks, certified checks, money orders, and bank wires honored immediately. Visa & Master Charge accepted. Allow 20 days for personal checks to clear.
REFUNDS: 10% restocking charge on all returns or exchanges. No refunds on opened software. Call first.
GUARANTEE: All products with full manufacturer's warranty. Sanyo and Apple warranty available. We have full repair and service facilities for all electronic repairs with HP, Dynascan, Pioneer, Sanyo and Apple trained and certified technicians. For any technical service call them for instant advice or questions right on their benches at (503) 479-4150.
REPAIRS: Call for details on quality guaranteed discount repair and reconditioning service.
We have been repairing electronic equipment for 12 years and love it!

CIRCLE #148 ON READER SERVICE CARD.

RAM Has More In Store

We're The Experienced Generation Of ATARI Experts!

- Software—Over 1000 Titles!
- Hardware
- Educational & Business Consultations
- Authorized Service
- Instruction
- 24-Hour Bulletin Board at (617) 371-1855
- Shipments Everywhere



Receive a FREE Programmers Aid Disk when you buy ACTION from OSS!

ACTION	$99
Programmers Aid Disk	30
	$129 Value
	For Only $99!

Disk Drives From:
• ATARI • RANA • TRAK • INDUS •

New! Programmable Keyboard Only $29.95!

- Works with ATARIWRITER, Basic, Assembler Editor, Program-Text Editor, and many other application programs.
- One programmable key can return a **word, phrase, or any combination of keystrokes** at speeds of up to 60 characters per second!

New England's #1 ATARI Source

RAM COMPUTER CENTER

427 Great Road, Acton, MA 01720 (617) 263-0418

CIRCLE #149 ON READER SERVICE CARD.

PLANETFALL
by Steve Meretzky
INFOCOM, Inc.
55 Wheeler St.
Cambridge, MA 02138
32K Disk $49.95

by Carl Firman

Would you like to send a post card from another planet to one of your friends? You would! Well, you'll find three of them in the Official Document File (ODF) that comes with the latest (and best) science fiction adventure from Infocom, **Planetfall**. These are real 110th-century post cards; the one from "Historic Ramos II" is my favorite (they even have old fashioned rocket ship rides there).

As you prowl through the ODF you will find out more about the Third Galactic Union (TGU). The TGU exists in the year 11,344. The TGU has the honor of being the LBG (Largest Bureaucracy in the Galaxy). You can't stand the TGU's LBG, and have decided to escape by enlisting in the Stellar Patrol (SP); you know, the one with the recruiting poster that says, "The Patrol's Looking For A Few Good Organisms."

If you want to know more about the SP and **Planetfall**, I will have to get my Magnetic Briefing Disk (MBD) out of my ODF and boot it. OK, here we are in 11,344, ODF's, the TGU, LBG and 15-page income tax forms (short form). Now you find out that **Planetfall** was written by a person known as the "Bearded Oracle of Yonkers" (real name, Steve Meretzky) and uses the ancient, but still famous, Infocom Interlogic operating system from the 20th century. With Interlogic, you can now communicate with your computer using complete sentences and a 600 word plus vocabulary — all text (bureaucracies like the TGU love text), no pictures.

You also discover that you are an Ensign 7th Class in the SP and that you have an official, authorized I.D. card to prove it. It will quickly become obvious that the recruiting posters that you read did not tell you everything. Ensign 7th Class is the lowest of the low. You're a deck-swabbing, brass-polishing member of the Special Assignment Task Force on board the SPS *Feinstein*.

Life is cruel on the *Feinstein*, mostly because of a certain Ensign 1st class Blather. When you meet Blather, your first impulse will be to render his molecules into a mass off disassociated atoms. Well, keep smiling at Blather; his time as a functioning group of associated molecules is short!

Shortly you find yourself plummeting towards the surface of a planet in one of the *Feinstein*'s escape pods. Through the pods viewport, you will see the whole SPS *Feinstein* disassociate — including Blather and all your shipmates. You are the sole survivor!

Well at least you won't have Blather to worry about any more. In fact, you are about to land on a water world with two small islands. Ah, dreams of a tropical paradise like Teegy 5.

Splashdown! Dreams of paradise are shattered quickly; as you climb to the top of the first island, you find yourself alone. A little exploring and you discover a large, deserted, technological complex. Where are all the people? Why aren't they here? Well, you're not discouraged — are you? After all, you've got your official (authorized also) mop, bucket, survival kit, and of course your Towel* (towels are very important on strange planets). Night comes, and with it tiredness — you head for the deserted dormitory you found earlier and bed down. Surely tomorrow will bring some answers (Ensigns 7th Class are notorious for being overconfident).

When you wake, you start to feel the loneliness — even Blather's company would be welcome right now. You are of course hungry so you eat some of the food goo in the survival kit. You feel better after eating, but not quite up to SP standards. You decide to learn as much about your environment as possible.

Now — The real adventure begins. You have a whole world to explore. Even though it only consists of two small islands, there is a lot to learn and a great dilemma to solve — yours and the planet's (hint from TGU central — the planet really is falling). You will find elevators, underground shuttles, helicopters and teleport booths. These all require official authorized access cards. Then there are some questions to answer — Where are all the people? Did they leave? Why do you feel slightly ill?

In the course of your exploration you will find a room full of robots. One of them is "Floyd". Floyd is slightly cockeyed with a strange lopsided grin — but he will prove to be as staunch and true a companion as E.T. was to Eliot. In fact Floyd is critical to your survival — so take good care of him.

You have one immediate priority — food. You have to eat, and the quickest way to die in the 110th-century is to starve. The second thing you will need is access cards, and Floyd can help you here when he's not reciting the first six hundred digits of Pi. Access cards will provide food and transportation — seek them out. There are also computers to use (even some game tapes), alien laboratories and a library; all will provide additional clues to your dilemma. Exercise good Ensign 7th Class logic and you will be a "Hero" and save this water world from destruction. I am going to put the MDB back in your ODF — you'll probably find it mixed up with the post cards.

There is one cardinal rule for people stranded on strange worlds to remember; when all seems lost always remember where your towel* is and everything will be fine. □

File 'em

A Magazine Indexing Program

32K Disk

by Norman Hill

How often have you thought that a certain hardware or software product you've heard about might now be of use to you? You remember that there was a review of such an item in one of the magazines, but that was several months ago. It will be necessary to search through a pile of back issues in order to find it, and, also, it may be that several similar items have been reviewed in different issues. A comparison is always helpful.

A commercial database manager would be a useful way of coming up with the necessary information. However, if you are like me, you prefer to design your own program. **File'em** is not as sophisticated as a professional DB, but it should be more than adequate for your needs.

File'em is designed to store information about magazine articles on disk and to retrieve that information when required.

After typing in and D:CHECKing the program, SAVE it to disk. The program is set up to add or retrieve information from already existing data files. Since such files do not exist, it will be necessary to set up new files before RUNning the program.

Insert a newly formatted disk into the drive and type in the following program:

```
10 OPEN #1,8,0,"D:FILEEM1.DAT"
20 CLOSE #1
30 OPEN #1,8,0,"D:FILEEM2.DAT"
40 CLOSE #1
50 OPEN #1,8,0,"D:FILEEM3.DAT"
60 CLOSE #1
70 OPEN #1,8,0,"D:FILEEM4.DAT"
80 CLOSE #1
90 OPEN #1,8,0,"D:FILEEM5.DAT"
100 CLOSE #1
110 OPEN #1,8,0,"D:FILEEM6.DAT"
120 CLOSE #1
130 OPEN #1,8,0,"D:FILEEM7.DAT"
140 CLOSE #1
150 OPEN #1,8,0,"D:FILEEM8.DAT"
160 CLOSE #1
```

When this program is RUN, the eight data files will be set up on your data disk.

Now load FILEEM and type RUN. You will be asked if you wish to make entries or retrieve information. Obviously, you will type "1" at this stage.

The next prompt will ask you to put the **File'em** data disk into drive #1 and press RETURN. You will now be asked in turn for the Publication, Date, Title and Page Number.

When these entries have been made, you will be asked for the language. Since many magazine programs are now being given in a mixture of BASIC and Assembler, number 8 allows for that possibility. For reviews and informational articles, where no computer language is used, enter 9. You may also prefer to alter some of the entries given in the program. Since Lisp is rarely used in microcomputer programming, you may prefer to leave number 4 blank, to be filled in later with some new language that may become popular in the future. I do not use Lisp, FORTH, PILOT or Pascal, but it is useful to have articles on file which involve these languages. Perhaps some day I will become interested in these languages and will wish to retrieve information concerning them.

The next prompt is for "Type of Article." Number 8 is available for any type that does not fit the earlier descriptions.

Next you will be asked for two keyword numbers from a given list. You may, of course, enter the same number twice. For an article dealing with Player/Missile Graphics you could enter 8,16 or 16,16 or 16,32. Number 32 is again a relief number if nothing else seems to fit. Numbers 29,30,31 are vacant and may be used for other keyboards.

Finally, you are asked if you have any more entries: If you enter Y, you will be returned to the first prompt. If you enter N, the program will end. Since the "Type of Article" prompt calls for a particular data file, each record is stored on disk as it is made. The next record entered may use a different file. If an error is made while making entries, and it is noticed before the record entry is completed, a simple SYSTEM RESET will let you restart the program and remake the entry.

When retrieving information, you will be asked for Language, Type of Article, and 2 Keywords. If you enter 1 for BASIC; 7 for General Utility; 32,32 for keywords, you will be presented with all General Utilities in BASIC. 7, 32 or 7, 7 for keywords will return all Financial Utilities in BASIC. Note that 32 is again a general escape keyword.

Output may be to the screen or to the printer. The printer output is designed for an 80-column printer. If you have a 40-column printer, the output instructions will obviously have to be modified. Lines 1380-1460 in the BASIC program are designed to format the printer output. □

```
10 REM ******************************
20 REM *          FILE 'EM          *
30 REM *             BY             *
40 REM *       NORMAN  HILL         *
50 REM ******************************
60 DIM T$(25),P$(10),D$(8),M$(1),PR$(1
),BZ$(10),AZ$(10),N$(13):N$="D:FILEEM0
.DAT":GRAPHICS 17
80 PRINT #6:PRINT #6;" DO YOU WISH TO"
90 PRINT #6:PRINT #6;" 1 MAKE ENTRIES"
100 PRINT #6:PRINT #6;"       or"
110 PRINT #6:PRINT #6;" 2 get informat
ion"
120 PRINT #6:PRINT #6:PRINT #6;"TYPE N
UMBER REQUIRED"
130 OPEN #2,4,C0,"K"
140 GET #2,A:IF A<49 OR A>50 THEN 140
150 CLOSE #2:IF A=50 THEN 840
160 GRAPHICS 0:PRINT "PLACE 'FILEEM.DA
T' DISK IN DRIVE 1":PRINT
170 PRINT "PRESS RETURN WHEN READY";
180 INPUT P$:IF A=50 THEN 290
190 PRINT CHR$(125):REM CLEAR SCREEN
200 REM -----------------
210 REM ENTER INFORMATION
220 REM -----------------
230 PRINT "CHECK EACH ENTRY CAREFULLY"
240 PRINT "BEFORE PRESSING RETURN"
250 PRINT "PUBLICATION (MAX 10 CHARACT
ER$)":INPUT P$
260 PRINT "DATE";:INPUT D$
270 PRINT "TITLE(MAX 25 CHARACTER$)":I
NPUT T$
280 PRINT "PAGE NUMBER";:INPUT P
290 GRAPHICS 1
300 PRINT #6;"     LANGUAGE$"
310 PRINT #6:PRINT #6;" 1 BASIC":PRINT
 #6:PRINT #6;" 2 microsoft"
320 PRINT #6:PRINT #6;" 3 assembler":P
RINT #6:PRINT #6;" 4 LISP"
330 PRINT #6:PRINT #6;" 5 FORTH":PRINT
 #6:PRINT #6;" 6 pilot"
340 PRINT #6:PRINT #6;" 7 pascal":PRIN
T #6:PRINT #6;" 8 BASIC & ASSEM":PRINT
 #6:PRINT #6;" 9 OTHER"
350 PRINT "ENTER LANGUAGE NUMBER";:INP
UT L
370 GRAPHICS 1:? #6;" TYPE OF ARTICLE"
380 PRINT #6:PRINT #6;" 1 DISK UTILITY
":PRINT #6:PRINT #6;" 2 education"
390 PRINT #6:PRINT #6;" 3 game":PRINT
#6:PRINT #6;" 4 INFORMATION"
400 PRINT #6:PRINT #6;" 5 REVIEW":PRIN
T #6:PRINT #6;" 6 home utility"
410 PRINT #6:PRINT #6;" 7 general util
ity":PRINT #6:PRINT #6;" 8 OTHER"
420 ? "ENTER TYPE NUMBER";:INPUT T
430 IF A=50 THEN 530
440 N$(9,9)=STR$(T):OPEN #1,9,C0,N$
530 GRAPHICS C0:SETCOLOR 2,4,2
540 PRINT "▶▶KEYWORD$":PRINT
550 PRINT " 1 ADVENTURE"," 2 ARCADE"
560 PRINT " 3 CLOCK",," 4 COMPILER$"
570 PRINT " 5 CONSOLE"," 6 DISK"
580 PRINT " 7 FINANCE$"," 8 GRAPHIC$"
590 PRINT " 9 GRAPH$","10 G.T.I.A."
600 PRINT "11 HARDWARE","12 HOME UTILI
TIE$"
610 PRINT "13 JOYSTICK","14 MEMORY"
620 PRINT "15 PADDLE","16 PL/MI$$"
630 PRINT "17 PLAYFIELD","18 PLOTTER$"
640 PRINT "19 PRINTER","20 SOUND"
650 PRINT "21 SIMULATION","22 STRING$"
660 PRINT "23 TAPE",,"24 VOICE"
670 PRINT "25 WORD-PROC.","26 O/S"
680 PRINT "27 REGISTER$","28 MODEM"
690 PRINT "29",,"30"
700 PRINT "31",,"32 OTHER"
710 PRINT :PRINT "ENTER 2 KEYWORD NUMB
ER$";:INPUT K1,K2
720 GRAPHICS C0:IF A=50 THEN 910
730 REM -------------
740 REM PRINT TO DISK
750 REM -------------
760 ? #1;P$:? #1;D$:? #1;T$:? #1;P;","
;L;",";T;",";K1;",";K2
770 CLOSE #1
780 ? :? "ANY MORE ENTRIE$(Y/N)";
790 INPUT M$:IF M$="Y" THEN 190
800 END
810 REM --------------
820 REM TYPE OF OUTPUT
830 REM --------------
840 GRAPHICS C0:PRINT "▶DO YOU WANT"
850 PRINT :PRINT "▶(1)DISPLAY":PRINT "
▶    OR":PRINT "▶(2)PRINTOUT":PRINT
860 PRINT "▶TYPE NUMBER REQUIRED"
870 OPEN #2,4,C0,"K"
880 GET #2,B:IF B<49 OR B>50 THEN 880
890 B=B-48:CLOSE #2
900 GOTO 160
910 N$(9,9)=STR$(T):OPEN #1,4,C0,N$
1000 IF B=1 THEN 1090
1010 REM -------------------
1020 REM HEADING$ TO PRINTER
1030 REM -------------------
1040 PRINT :PRINT "▶IS PRINTER ON?"
1050 PRINT :PRINT "PRESS RETURN WHEN R
EADY";:INPUT PR$
1060 PRINT CHR$(125)
1070 LPRINT "  PUBLICATION        DATE
     PAGE          TITLE"
1080 LPRINT "  -----------        ----
     ----          -----"
1090 TRAP 1340
1100 REM ------------------------
1110 REM GET INFOMATION FROM DISK
1120 REM ------------------------
1130 INPUT #1;P$:INPUT #1;D$:INPUT #1;
T$:INPUT #1;P,LI,TI,C1,C2
1140 IF LI=L OR L=9 THEN 1160
1150 GOTO 1130
1160 IF TI=T OR T=8 THEN 1180
1170 GOTO 1130
1180 IF K1=32 OR K1=C1 OR K1=C2 THEN 1
200
1190 GOTO 1130
1200 IF K2=32 OR K2=C1 OR K2=C2 THEN 1
220
1210 GOTO 1130
1220 IF B=2 THEN 1380
1230 REM ---------------
1240 REM PRINT TO SCREEN
1250 REM ---------------
1260 PRINT P$,D$,"PAGE ";P:PRINT T$
1280 GOTO 1130
1290 REM ----------------
```

BACKUP HARDWARE & SOFTWARE

24 HOUR SHIPPING

THE HAPPY™ 810 ENHANCEMENT KIT WITH WARP SPEED SOFTWARE

CALL FOR SPECIAL LOW PRICE

24 HOUR SHIPPING

BACK UP ANY ATARI™ DISK

RIGHT-WRITE

This device will allow you to write to side 2 of any disk. Install this box to your Atari™ 810 disk drive in 5 minutes. Just plug in one cable — no cutting or soldering required.

Push the button and a red led will light, allowing you to write a disk without notching out a hole in the disc.

Easy plug-in installation instructions included.

Fully tested and assembled.

ONLY **$24.95**

THE TRANSFER PACK

1) Disk file to tape
2) Boot tape to disk file
3) Tape to tape

Very powerful and low priced.

We have a multi-stage cassette Transfer Program to the Transfer Pack. This program will do most multi-stage tapes. It will transfer them to disk files.

ALL 4 FOR ONLY **$24.95**

NOW AVAILABLE *THE ULTIMATE BOOK*

On copyguarding. Master code/cracker reveals all. Hardware and software tricks. Informative book and disk with examples and programs.

ALL FOR **$24.95**

THE REPRODUCER

Reproduce copy-guarded disks on the Atari™ 1050 disk drive. Our Reproducer is a simple plug-in modification for the 1050 drive. With our software you will be able to write bad sectors.

Reproducer and software: **$49.95**

We are working on new products and software.

For information call (516) 588-6019.

Add $5 for Shipping & Handling.

PHONE ORDERS: (516) 467-1866
PRODUCT INFO: (516) 588-6019

GARDNER COMPUTING COMPANY
P.O. BOX 388, HOLBROOK, N.Y. 11741
CIRCLE #150 ON READER SERVICE CARD.

We accept C.O.D. orders, money orders and ship within 24 hours. (Personal checks will have to clear before shipping.)

ATARI

PC GALLERY MAIL ORDER

P.O. BOX 3354, CHERRY HILL, N.J. 08034

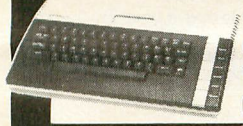
ATARI

TOLL FREE (800) 992-3300

FOR ORDER ONLY

FOR INFORMATION & N.J. CALL (609) 596-1944

ATARI HOME COMPUTERS



Atari 600XL	CALL
Atari 800 XL	Call
1050 Disk Drive	Call
Communicator II	$135
1010 Recorder	$78
1020 Printer	$239
1025 Printer	$345
1027 Daisy Wheel Printer	$274
830 Modem	$99
CX419 Bookkeeper	$195
1030 Modem	$CALL

SOFTWARE

Atariwriter	$71	Tennis	$35
Speed Reading	$53	Joust	$37
Pilot	$55	Ms. Pac Man	$35
Assembler Editor	$47	Dig Dug	$35
Logo	$72	Qix	$35
Family Finance	$36	Mickey Mouse	$37
Home Filing Manager	$39	Paint	$35

Microsoft Basic 2 $61.00
Donkey Kong $35.00
Pole Position $36.50

PRINTERS

EPSON

RX-80	CALL
RX-80 FT	CALL
FX-80 & FX-100	CALL
Printer Interface	$79

OKIDATA

92	$CALL

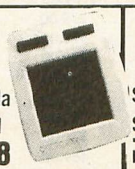
STAR

Gemini 10X	$296
Gemini 15X	$399

Smith Corona

TP-2	$445

Zork I, II, III	$26.50
Infidel	$34.00
Enchanter	$34.00
Qbert	$35.00
Flight Simulator	$43.00
Rally Speedway	$36.00
Congo Bongo	$30.00
Odesta Chess	$49.00
Archon	$28.00
Mule	$28.00



Koala Pad $68

Home Accountant	$48
Tax Advantage	$48

MODEM

Hayes

Smart	$204.95
Smart 1200	$534.95

IN-HOME KEY BOARD FOR 400 ..... $64.95

RAM BOARD- INTEC
48K..... $85 64K..... $99

MONITORS

AMDEK (300)	$154.95
COLOR I	$295.00

WICO

The Boss $14.95 Joystick $20.00
Famous Red Ball $25.95
Track Ball $35.95

PERCOM Disk Drive $CALL

DISKS

Elephants

SS/SD	$17.50
SS/DD	$21.00
Fuji SS/DD	$20
Verbatim SS/DD	$24
Maxell (MD1)	$26

COMPUTER COVER

$7.99 ea.

Features heavy duty canvas with vinyl interior - waterproof.

ATARI 800	$7.99
ATARI 810	$7.99
EPSON MX 80	$7.99
EPSON MX 80FT	$7.99
OKIDATA µ82A,92	$7.99

SURGE PROTECTOR UL LISTED

NEW!!

6 Outlet	$62.95
4 Outlet	$37.95

We carry a large selection of Atari software, call for FREE catalog with order ...or send $1 (refundable with order) UPS shipping 3% for hardware and $3 for software. N.J. resident add 6% sales tax.

VISA MasterCard

Note: Atari, Epson, IBM and Okidata are registered trademarks.

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

*MASTER CARD OR VISA ADD 3%

CIRCLE #151 ON READER SERVICE CARD.

```
1300 REM PRINT TO PRINTER
1310 REM ----------------
1320 LPRINT :LPRINT "   ";P$;"       ";
D$;"     ";BZ$;"      ";T$
1330 GOTO 1130
1340 CLOSE #1:END
1350 REM ---------------------
1360 REM FORMAT PRINTER OUTPUT
1370 REM ---------------------
1380 BZ$="           "
1390 LN=LEN(P$)
1400 IF LN<10 THEN P$(LN+1)=BZ$(1,10-L
N)
1410 LN=LEN(D$)
1420 IF LN<8 THEN D$(LN+1)=BZ$(1,8-LN)
1430 AZ$=STR$(P)
1440 BZ$(4-LEN(AZ$))=AZ$
1450 BZ$=BZ$(1,3)
1460 GOTO 1320
```

●

CHECKSUM DATA

(See p. 30)

```
10 DATA 335,264,977,647,343,485,942,74
,32,322,876,320,609,512,964,7702
170 DATA 489,689,877,871,795,877,52,13
4,77,461,134,324,916,457,479,7632
320 DATA 649,309,648,771,873,523,127,6
77,260,960,419,53,58,172,581,7080
560 DATA 539,903,813,110,746,437,352,7
94,287,902,969,408,403,247,3,7913
710 DATA 645,18,375,934,381,506,668,32
6,75,40,616,393,622,102,783,6484
860 DATA 784,346,666,632,717,39,537,14
1,261,143,812,88,447,369,907,6889
1090 DATA 679,438,659,440,839,405,711,
449,713,488,715,493,710,551,634,8924
1240 DATA 513,636,349,717,885,915,880,
776,715,346,179,444,181,998,22,8556
1400 DATA 846,993,526,230,67,3,721,338
6
```

●

In issue no. 15 of ANALOG **Computing**, we presented a complete index of articles which had appeared up to that point. **File'em** may be used to index these articles.

For those readers who would like a copy of issue no. 15, send $4.00 to:

Back Issues
P.O. Box 23
Worcester, MA 01603

ATARI

A LIFETIME OF DISCOVERY BEGINS WITH ATARI COMPUTERS

1-800-824-7506

DISK DRIVES

RANA 1000	CALL
TRAK AT1 (Single/Double Density)	CALL
TRAK ATD1 (Single Density/Parallel Interface)	CALL
TRAK ATD2 (Single/Double Density/Parallel Interface)	CALL
ASTRA 1620	CALL
INDUS GT	CALL

RAM (MEMORY) BOARDS

ATARI 16K	$ 29.00
Intec 64K Board (400)	109.00
Intec 48K Board (400)	86.00
Intec 32K Board (400)	59.00
Microbits 64K (600)	129.00

MC
VISA
C.O.D. (Add $2.50)

COMPUTER CREATIONS, Inc.
P.O. Box 292467
Dayton, Ohio 45429
For Information Call:
(513) 294-2002
(Or to order in Ohio)

All Orders Add $3.00 Shipping and Handling. Ohio Residents Add 6% for Sales Tax.
ASK FOR OUR FREE CATALOG

PRINTERS

Gemini STX-80 (80 Column Thermal)	$169.00
Gemini 10X Printer (80 Column)	279.00
Gemini 15 Printer (136 Column)	399.00
Delta 10 Printer (80 Column)	499.00
Powertype	379.00
Radix 10	649.00
Radix 15	769.00
Epson RX-80 (80 Column) Dot Matrix	379.00
Epson RX-80FT (80 Column)	469.00
Epson FX-80 (80 Column) Dot Matrix	579.00
Epson FX-100 (136 Column) Dot Matrix)	749.00
Axiom AT-100 (80 Column) Inc. Int. Cable	229.00
Silver-Reed EXP 500 Daisy Wheel Printer (80 Column)	429.00
Silver-Reed EXP 550 Daisy Wheel Printer (136 Column)	679.00

PRINTER INTERFACE CABLES

APE FACE	$ 65.00
MPP-1150	79.00
Star Universal	69.00

MONITORS

Gorilla™ Hi Res 12" Non-Glare Screen	$ 85.00
Gorilla™ Hi Res 12" Non-Glare Amber Screen	99.00
BMC Green Screen	79.00
Sakata SC100 Color Screen	249.00
Monitor Cable	12.00

GENERIC DISKS

Generic 100% Defect-Free/Guaranteed MINI-FLOPPY DISKS

Diskettes (1 Box Min.) - 10 per box			Bulk Diskettes with Sleeves - Price per Disk		
	SS/DD	DD/DD		SS/DD	DD/DD
1 or 2 Boxes	17.49/box	20.99/box	10-29	1.59	1.99
3 - 9 Boxes	15.99/box	19.99/box	30 - 99	1.49	1.89
10+ Boxes	14.99/box	18.99/box	100+	1.45	1.79

ATARI HOME COMPUTERS

ATARI 600XL™ Home Computer (16K RAM)	CALL
ATARI 800XL™ Home Computer (64K RAM)	CALL
ATARI 1200XL™ Home Computer (64K RAM)	CALL
ATARI 1400XL™ Home Computer (64K RAM)	CALL
ATARI 1450XLD™ Home Computer (64K RAM)	CALL

ATARI ADDITIONAL EQUIPMENT ..... CALL

MODEMS

MPP 1000C	$129.00
Signalman Mark II Modem	79.00
Hayes Stack Smartmodem (300 BAUD)	239.00
Hayes Stack Smartmodem (1200 BAUD)	549.00
Novation J CAT	119.00
Novation 103 Smart CAT	189.00
Novation Auto CAT	199.00

CIRCLE #152 ON READER SERVICE CARD.

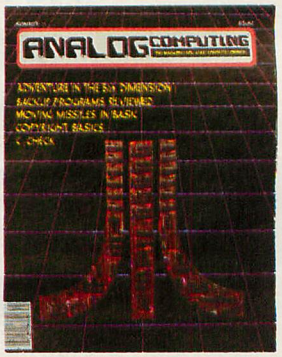
BACK ISSUES

Catch up on what you've missed!

ISSUE 2
Wasting Arrays
Atari's CPU
Download Terminal
Converting BASIC
Disk Files

ISSUE 8
GTIA Graphics
Audio in Your Programs
NOREM
Graphic Violence
Color Slot Machine

ISSUE 9
Build Your Own 400
Keyboard
Harvey Wallbanger
Forth-Dos
Letter Writer

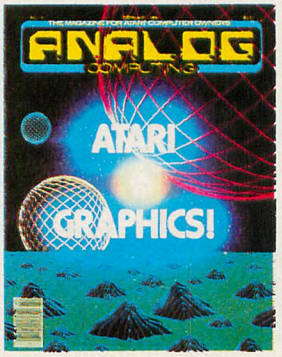


ISSUE 11
Strings in BASIC
C:CHECK
Disk Cataloging Utility
Adventure in the
Fifth Dimension
Moving Missiles in BASIC

ISSUE 13
Fine Scrolling Part 1
Roundup
Space Assault
Observational Astronomy
CIO Routines

ISSUE 14
Fine Scrolling Part 2
Disassembler in BASIC
Hexpad
Lumberjack
Retrofire!

ISSUE 15
Fine Scrolling Part 3
Knights and Chalices
Music Synthesizer
Bricklayer's Nightmare
Alternative Keyboard
Handler



ISSUE 16
Fine Scrolling Part 4
Create-A-Font
Bar Chart Subroutine
Shooting Stars
3-D Object Rotation

All back issues $4.00 each

Send check or money order to:
ANALOG Back Issues
P.O. Box 615
Holmes, PA 19043

MasterCard and Visa orders call:
1-800-345-8112
in PA. 1-800-662-2444

PMI CELEBRATES THE OLYMPICS!!!

PURCHASE *ANY* PMI PROGRAM AND WE WILL GIVE YOU, ABSOLUTELY *FREE*, *NORDIC SKI*, A 100% MACHINE LANGUAGE ARCADE-STYLE GAME BY LEN DORFMAN, IN HONOR OF THE 1984 OLYMPICS *AND* PMI WILL DONATE $1.00 TO THE U.S. OLYMPIC TEAM FOR EACH PROGRAM SOLD!! HURRY! THIS OFFER IS FOR A LIMITED TIME ONLY!

WAMPYR'S TOMB

WARNING!! This adventure may be addictive. WAMPYR'S TOMB, by James Glines, will take you on a journey from the village of Amberpool through a 'mid-evil' countryside filled with magical spells, demons and dragons. Can you stay alive? To prevail in WAMPYR'S TOMB you must do more! You must equip yourself to do battle with the LICH and win. (Of course there is the option of dying!) We will supply you with a little money and the opportunity to go where you've never been; the courage and the desire to conquer the unknown must come from within. WAMPYR'S TOMB will not only challenge your bravery, your intelligence and your cunning, but will take you beyond your imagination into a realm of new and exciting forces where only legends survive. Requires 48K RAM and one disk drive.

$29.95

SWORD POINT

SWORD POINT, by William A. Austin, is a very unique and exciting adventure. You will assume the role of a young man seeking to climb to the pinnacle of success in the mythical monarchy of Simafranconia, a nation not unlike feudal 17th Century Europe. It is a country of totally masculine values, where style supersedes substance, and social level is not the best thing, it is the only thing worth acquiring. Dueling, gambling, wenching and high living are among a few of the things that you will do to gain the King's attention. So tankards up ...a toast to bravery...with a little luck and by dancing to the daredevil's tune on the Swords Point...you may just win! Requires 48K RAM and one disk drive.

$31.95

PROGRAMMERS!

If you have written a program that is technically sophisticated and also "user-friendly", PMI would be interested in publishing your work. For information on how to submit your material write to: PMI, P.O. Box 2895, Winter Park, FL 32790-2895, Or phone (305) 644-3822.

COM-CON

Have you ever wondered what your computer would have to say about that party you had last weekend? Or the shape the world is in today? Well now you can find out. *COM-CON* has arrived! *COM-CON* by William Austin is an unique, exciting and entertaining program for everyone. Your computer can answer your questions on virtually any topic. While carrying on a discussion with *COM-CON* you will notice that your computer actually has a "personality" of its own. And like any human being the answers will sometimes be as vague and nebulous as a political candidate. And at other times quite philosophical. If you're 4 to 104 *COM-CON* is for you. *COM-CON* is available for the Votrax Type 'n Talk, $31.95, the Alien Group Voice Box $31.95, *SAM* $31.95, and in a text only version $26.95. Requires *48K RAM, BASIC* cartridge, one disk drive and optional voice synthesizer.

KARMIC CAVERNS

KARMIC CAVERNS, by Len Dorfman, is a 100% machine language arcade-style game that will entertain you hour after hour! You must find your way through the maze-like caverns avoiding the deadly plasmatic guards (not to mention the electrified walls) and acquire as many energy pods as you can. You must climb ever upwards striving to acquire enough energy points to reach the next level before time runs out for you. KARMIC CAVERNS utilizes the full sound and graphics capabilities of the Atari computer! Fun for all ages. Requires 48K RAM and one disk drive.

$34.95

ERG

ERG, another 100% machine language arcade-style game by Len Dorfman, is the ultimate maze escape challenge!! Multiple skill levels, multiple mazes, roving energy fields which will drain your energy and proton barriers which cause disintegration upon contact are the obstacles you will face. If you are an expert strategist and your reflexes are honed to a razor's edge (i.e., you can pull a gnat's whisker out as he flies by) the odds are about even. You might just win! Requires *48K RAM* and one disk drive.

$34.95

RESTORE

List the unlistable! *RESTORE* by William Fletcher will allow you to make a *LISTABLE, EXECUTABLE* copy of any *BASIC* program. *RESTORE* is extremely user-friendly and the simplicity of operation is astounding. *RESTORE* is a compiled program and executes at machine language speed. Utilizing the graphics and sound capability of the Atari, *RESTORE* is not only an indispensible programmer's tool, but is fun to use. If you have two disk drives *RESTORE* will automatically seek out relocated directories and give you the option to either restore all *BASIC* programs on the disk or make a copy of the disk putting the directory in its normal location. Requires *48K RAM* and one or more disk drive.

$27.95

PROGRAMMIN' STUFF — $26.95
VARICOMMANDER — $29.95
PROGRAMMIN' STUFF/VARICOMMANDER PACKAGE — $45.95
WORDPOWER — $27.95
BIOGRAPH — $24.95

PMI, P.O. Box 2895, 2500 Lee Road, Suite 210, Winter Park, FL 32790-2895

☐ *SWORD POINT* ☐ *KARMIC CAVERNS* ☐ *WAMPYR'S TOMB*
☐ *PROGRAMMIN' STUFF* ☐ *T.I.P.S.* ☐ *BIOGRAPH* ☐ *VARICOMMANDER* ☐ *RESTORE* ☐ *ERG*
☐ *COM-CON* ☐ *VARICOMMANDER and PROGRAMMIN' STUFF PACKAGE* ☐ *WORDPOWER*

Total $ _____, plus $3.00 postage & handling (Florida residents add 5% sales tax)
☐ CHECK (allow 3 weeks for personal checks to clear) ☐ MONEY ORDER
☐ C.O.D. (add an additional $2.00 for C.O.D. orders)

NAME (print) _____
Address _____
City _____ State _____ Zip _____

S.A.M. is a trademark of DON T ASK

DEALER INQUIRIES INVITED

ATARI is a trademark of ATARI INC

CIRCLE #153 ON READER SERVICE CARD.

ATARI®

ATARI is a trademark of ATARI, INC.

NEW LOWER PRICES!!

ATARI 800XL ..... Call

Atari 1050 Disk Drive ..... Call
Atari 1010 Recorder 77.00
Atari 1027 Printer ..... Call
Atari 850 Interface ..... Call

MOSAIC

64K Ram/400 ..... 149.00
64K Ram + Cable Kit/400/800 ..... 169.00
48K Ram Kit 94.00
16/32 Expander 64.95
32K Ram 77.95
Mosaic Adaptor 49.95

Memory Expansion for Atari 600XL .. Call

DISK DRIVES

Percom .....
Trak ..... Call
Rana ..... For
Indus Gt .. Prices

NEW

RS232 Modem Adaptor (Through serial port) 39.95

Koala Touch Tablet Cart or Disk ... 69.95

MONITORS

We carry a full line of AMDEK and USI Monitors.

LET YOUR ATARI IMPROVE YOUR LIFE AND YOUR MIND

NEW - THE STIMUTECH SUBLIMINAL PROGRAM

With the **Expando-Vision Interface** the Atari Computer user can now work on improving his or her personal environment during normal television viewing with a proven psychological technique that was banned by the F.C.C.

Buy the Expando-Vision Interface for $99 and get the Rom Cart of your choice for **Free!!!**

Rom Carts are available for:

Weight Control	Drinking Control
Study Habits	Smoking Control
Stress Control	Career Success

Sexual Confidence

Additional Rom Carts $29.95 each

3RD Party Printer Interfaces

Interfast I 139.00
Apeface w/cable included 69.95

PRINTERS

SUPER SPECIAL - Prowriter Package with Apeface or Interfast I ..... Call
Axiom AT-100 ..... 229.00
Axiom AT-550 ..... 329.00

ATARI

Programming 2 & 3 - T 23.95
Conversational Lang. - T ... 44.95
Music Composer - Cart ... 32.95
My First Alphabet - D 26.95
Touch Typing - T 19.95
Home Filing Mgr. - D 37.95
Mailing List - T 19.95
Caverns of Mars - D 28.95
Computer Chess - Cart ... 26.95
Missile Command - Cart ... 26.95
Super Breakout - Cart 26.95
Star Raiders - Cart 32.95
Assembly Editor - Cart 46.95
Basic - Cart 41.95
Macroassembler - D 67.95
Pilot (Home Pkg.) - Cart ... 58.95
Invitation to Program I T ... 19.95
Speed Reading - T 55.95
Basketball - Cart 26.95
Graph-it - T 15.95
Juggles House - D/T 22.95
Pilot (Educator) - Cart 97.95
Video Easel - Cart 26.95
Defender - Cart 32.95
Galaxian - Cart 32.95
Qix - Cart 32.95
Dig Dug - Cart 32.95
ET - Cart 37.95
Timewise - D 23.95
Atariwriter - Cart 74.95
Donkey Kong - Cart 37.95
Ms. Pac-Man - Cart 39.95
Tennis - Cart 35.95
Eastern Front - Cart 32.95
Donkey Kong Jr. - Cart ... 39.95
Pengo - Cart 35.95
Logo - Cart 79.95
Robitron - Cart 35.95
Pole Position - Cart 39.95
Microsoft Basic II - C 67.95
Paint - D 33.95
Caverns of Mars - Cart 32.95
Joust - Cart 39.95
Visicalc - D 159.95

THORN

War Games - Cart 27.95
Hockey - Cart 27.95
Soccer - Cart 30.95
Submarine Commander - Cart ... 30.95

SIERRA ON-LINE

Homeword - D ..... Call
Dark Crystal - D 27.95
Frogger - D/T 23.95
Quest For Tires - D 23.95
Ultima I - D 23.95
Ultima II - D 41.94
Wizard/Princess - D 22.95

SPINNAKER

Up For Grabs - Cart 27.95
Cosmic Life - Cart 23.95
Facemaker - Cart/Disk 23.95
Alphabet Zoo - Cart 23.95
Fraction Fever - Cart 23.95
Kids on Keys - Cart 23.95
Delta Drawing - Cart 27.95
Story Machine - Disk 23.95
Rhymes & Riddles - D 20.95
Amazing Things - D 27.95
Hey Diddle Diddle - D 20.95
Snooper Troops 1 or 2 - D .. 30.95
Trains - D 27.95
Aerobics - D 30.95

SIRIUS

Bandits - D 23.95
Way Out - D 27.95
Repton - D 27.95
Twerps - D 23.95
Blade of Blackpoole - D .. 27.95
Type Attack - D 27.95
Beer Run - D 23.95
Critical Mass - D 27.95
Wavy Navy - D 23.95

GAMESTAR

Baja Buggies -D/T 22.95
Starleague Baseball - D/T 22.95
Starbowl Football D/T ... 22.95

SEGA

Buck Rogers - Cart 27.95
Star Trek- Cart 27.95
Congo Bongo- Cart 27.95
Tac Scan- Cart 27.95

EPYX

Gateway To Apshai - Cart 27.95
Pitstop - Cart 27.95
Lunar Outpost - Cart 27.95
Swat Rescue - Cart 27.95
Silicon Warrior - Cart 27.95
Fun With Music - Cart 27.95
Jumpman Jr. - Cart 27.95
Dragonriders of Pern - D/C 27.95
Temple of Apshai - D/C ... 27.95
Hellfire Warrior - D/C 20.95
Fun With Art - Cart 27.95

INFOCOM

Deadline - D 34.95
Witness - D 34.95
Zork I, II, III - D 27.95
Starcross - D 27.95
Planetfall - D 34.95
Suspended - D 34.95
Enchanter - D 34.95
Infidel - D 34.95

Hundreds of items available for the ATARI ... please call.

LJK

Letter Perfect - D 74.95
Data Perfect - D 74.95
Spell Perfect - D 64.95

BRODERBUND

AE - D 23.95
Arcade Machine - D 41.95
Bank St. Writer - D 49.95
Lode Runner - D 23.95
Drol - D 23.95
Spare Change - D 23.95
Choplifter - D 23.95

DATA SOFT

Money Wizard - D 49.95
Letter Wizard - D 49.95
Spell Wizard - D 34.95
Genesis - D/T 20.95
Heathcliff - D/T 20.95
Pooyan - D/T 20.95
Zaxxon - D/T 27.95
Micropainter - D 23.95

SYNAPSE

Blue Max - D/T 23.95
Dimension X - D/T 23.95
Drelbs - D/T 23.95
Encounter - D/T 23.95
Necromancer - D/T 23.95
Quasimoto - D/T 23.95
Rainbow Walker - D/T ... 23.95
Shamus Case II - D/T ... 23.95
Zeppelin - D/T 23.95
N.Y.C. - D/T 23.95

FIRST STAR

Bristles - D/T 20.95
Flip Flop - D/T 20.95
Astro Chase - D 20.95
Boulder Dash - D/T 20.95

STRATEGIC SIMULATIONS

Carrier Force - D 41.95
Combat Leader - D/T ... 27.95
Battle For Normandy -D/T 27.95
Eagles - D/T 27.95

PARKER BROS.

Astrochase - Cart 34.95
Frogger - Cart 34.95
Q*Bert - Cart 34.95
Popeye - Cart 34.95

MISCELLANEOUS

Castle-Wolfenstein - D .. 20.95
Home Accountant - D ... 52.95
Master Type - D/Cart ... 27.95
Spelunker - D 27.95
Flight Simulator II - D ... 37.95
Zombies - D/T 23.95
Mr. Robot - D 23.95
River Raid - Cart 31.95
Diskey - D 34.95
Chatterbee - D 27.95
Rally Speedway - Cart ... 34.95
Ultima III - D 41.95
Miner 2049'er - Cart 34.95
Scraper Caper - Cart ... 34.95
Basic XL - Cart 74.95
Monkey Wrench II - Cart . 49.95
Omnimon 82.95
Scraper Caper - Cart ... 34.95

BOOKS

Compute's First Bk/Games 12.95
Inside Atari DOS 19.95
Mapping the Atari 14.95
Machine Language/ Beginners 14.95
Your Atari Computer ... 16.95
The Atari Assembler 12.95
Visicalc Book-Atari 14.95

ACCESSORIES

WICO Joystick ..... Call
Flip 'n' File Trunks 20.95
Flip 'n' File Cart 20.95
Joysensor 24.95
Elephant Disks (Box of 10) 20.00
WICO Trakball 37.95
KRAFT Joystick 15.95

D - Disk T - Cassette
Cart - Cartridge

ComputAbility™

To Order Call Toll Free

800-558-0003

For Technical Info, Order Inquiries, or for Wisc. Orders -

414-351-2007

NO SURCHARGE FOR MASTERCARD OR VISA

COMPUTABILITY
P.O. Box 17882
Milwaukee, WI 53217

MasterCard VISA

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

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

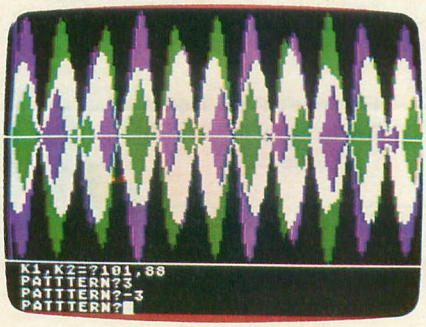
CIRCLE #154 ON READER SERVICE CARD.

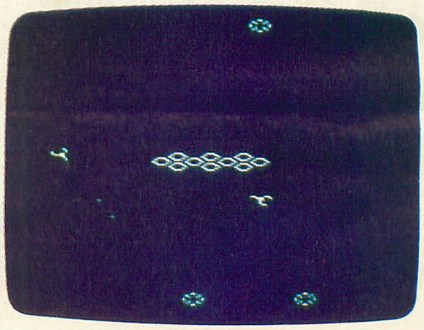
INDEX TO ADVERTISERS

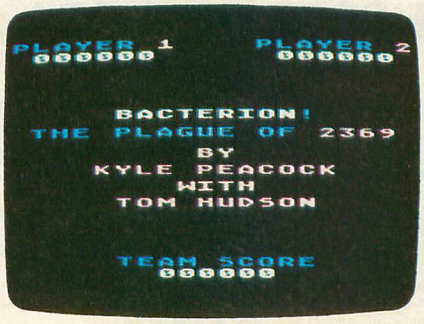
READER SERVICE #	ADVERTISER	PAGE #
156	Adventure International	OBC
116	Alien Group	24
127	Allen Macroware	41
125	Alpha Systems	39
120	ALOG Computing	33
155	ANALOG Publishing	101, IBC
109	Astra Systems	11
154	Computability	103
147	Computer Creations	93
135	Computer Discount Sales	70
133	Computer Palace	64
140	Compu Ware	75
118	Convologic	29
144	Cosmic Computers	88
124	Datamost	37
101	Dennison/Elephant	IFC
114	Dorcett	21
152	Draper Software	100
146	Eastern House	93
130	Eclipse Software	59
141	El Comp Publishing	77
117	Epyx	27
150	Gardner Computing	99
110	Happy Computing	14
111	Hayden Publishing	18
103	Infocom	4
122	Interactive Software	35
121	Lateral Software	35
145	Lyco	90
115	Mach-ina	23
105	Macrotronics	9
113	Micca Enterprises	20
134	Microprose	67
132	Mighty Byte/Spartan	62
128	Miles Computing	43
104	OSS Precision Software	7
143	Overbyte	87
119	Pace	32
151	PC Gallery	99
153	PMI	102
131	Quality Software	61
149	Ram	95
148	RCE	94
129	Rising Sun	56
136	Sar-An	71
102	Scholastic	2,3
138	SIM	73
108	Soft Sector	10
126	Software Plus	40
112	Southern Software	20
106	Super Ware	9
123	Telemetrics	36
142	Wedgewood Rental	87
137	XLent Software	72
139	Ziza Presents	74

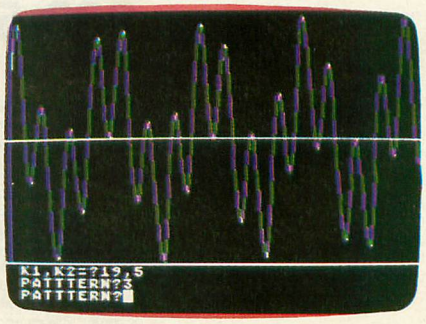
This index is an additional service. While every effort is made to provide a complete and accurate listing, the publisher cannot be responsible for inadvertent errors.

Coming soon in future issues of **ANALOG Computing!**









From the editors of
A.N.A.L.O.G. Computing

$14.95

THE ANALOG COMPENDIUM

The best ATARI® Home Computer Programs from the first ten issues of A.N.A.L.O.G. Computing Magazine.

All Compendium programs are available on DISK.

The **ANALOG Compendium** is available at selected book and computer stores, or you can order it direct. Send a check or money order for $14.95 + $2 shipping and handling to: **ANALOG Compendium, P. O. Box 615, Holmes, PA 19043.**

Or you can order by phone with MasterCard or VISA. Call toll free: 1-800-345-8112 (in PA, call 1-800-662-2444). **For orders outside the U.S., add an additional $5 air mail, $2 surface.**

CIRCLE #155 ON READER SERVICE CARD.

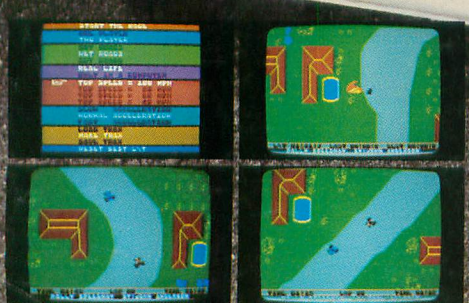
John Anderson's

RALLY SPEEDWAY

SLIP INTO SOMETHING TURBO-CHARGED

like the road-scorching speedster in RALLY SPEEDWAY — the game that turns your ATARI into a full-blown, four-wheeled demon! Take on our demanding course, or use the unique Construct-A-Track option to design your own. Push yourself to the limit for a better lap time, or get a friend in on the action for a one-on-one duel to the finish line — it's your choice!

CALL TOLL FREE 1-800-327-7172



MAKE TRACKS TO YOUR DEALER TODAY!

ATARI 400/800/1200 CARTRIDGE

053-0171 . . . . . $49.95

If your dealer doesn't have Rally Speedway in stock, phone 1-800-327-7172 with your order, or write us at Adventure International, P.O. Box 3435, Longwood, FL 32750 for your FREE catalog.

DEALER ORDERS WELCOME

CIRCLE #156 ON READER SERVICE CARD.

155

COPYRIGHT 1983

Adventure® INTERNATIONAL

A DIVISION OF SCOTT ADAMS, INC.