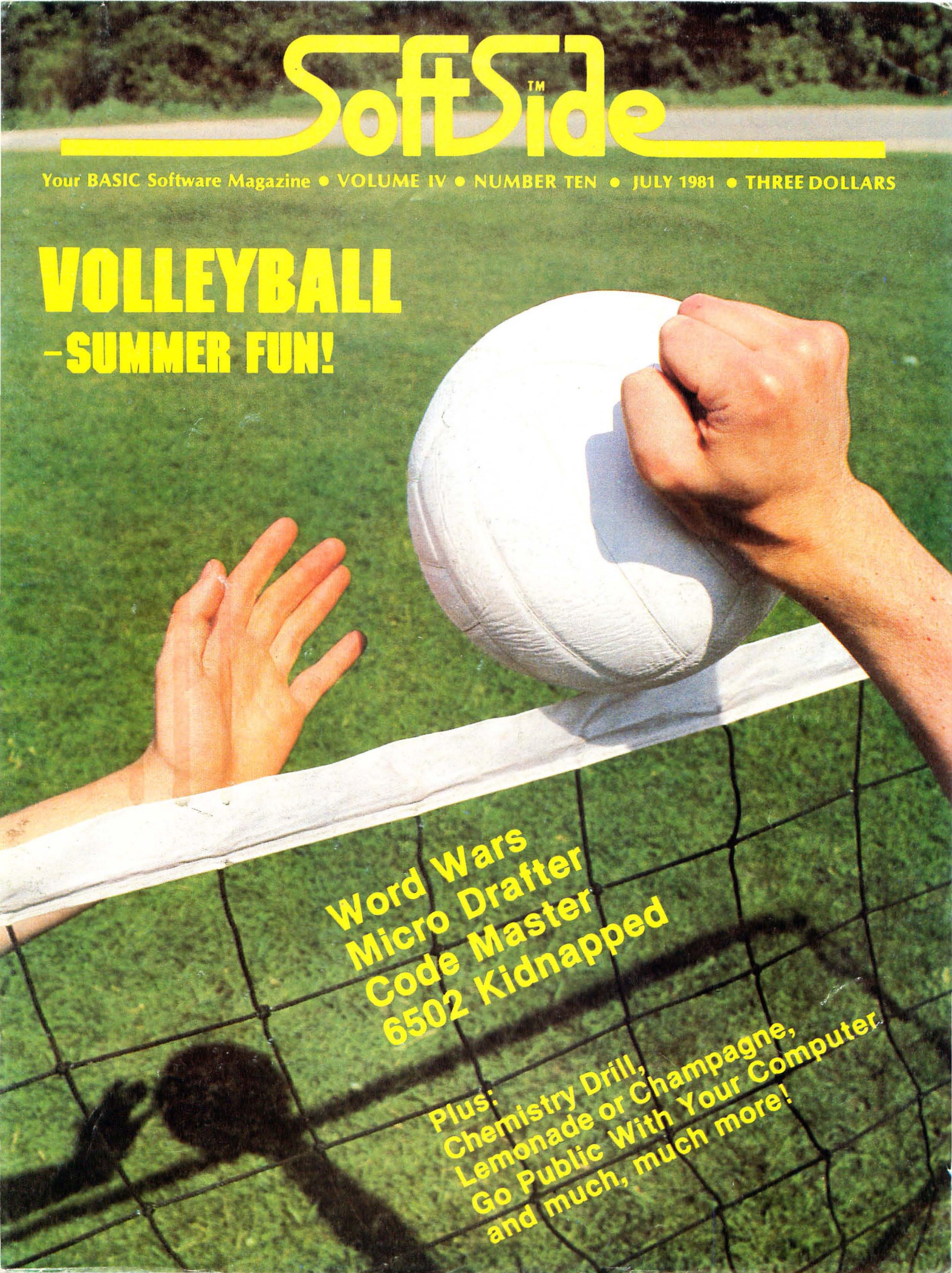
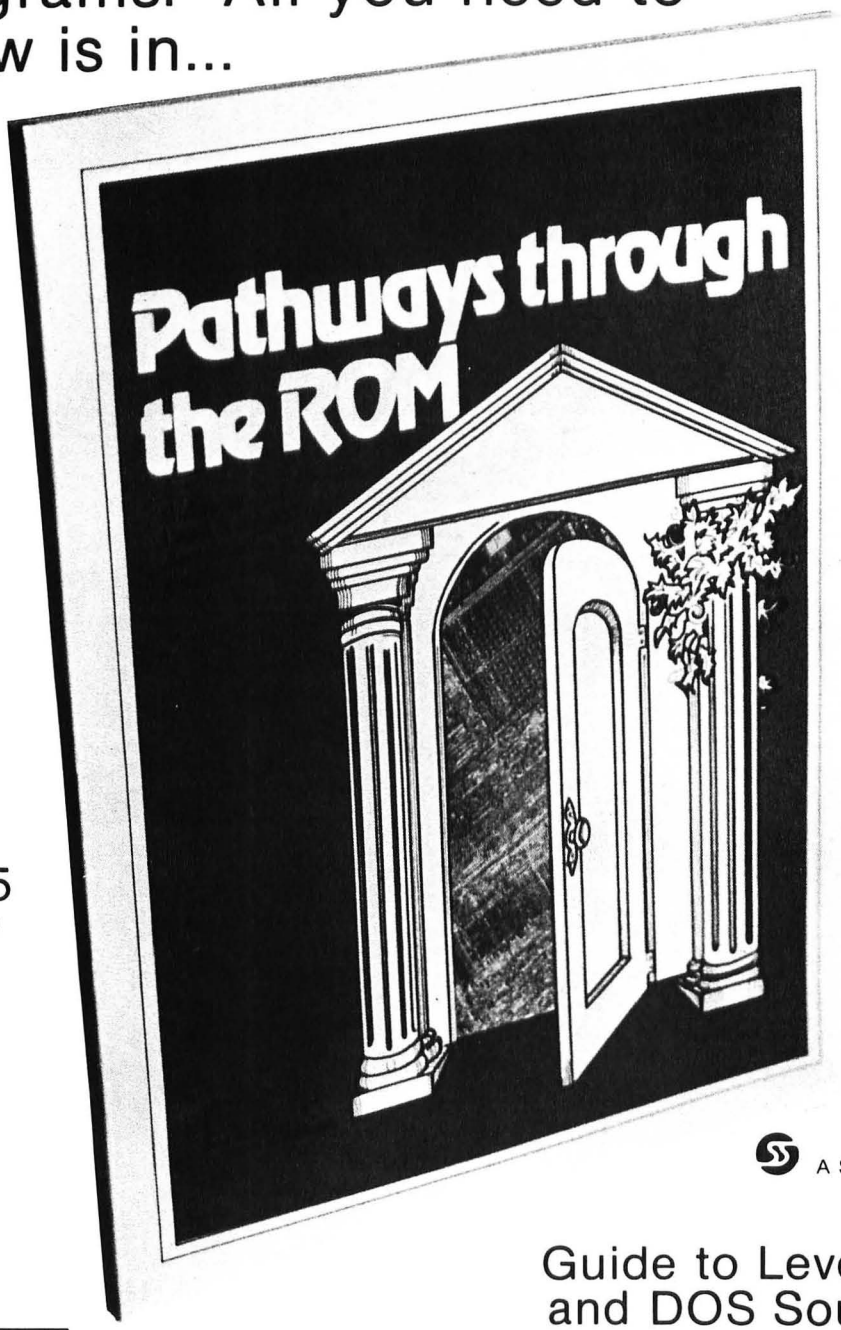

SoftSide™
Your BASIC Software Magazine • VOLUME IV • NUMBER TEN • JULY 1981 • THREE DOLLARS
VOLLEYBALL
-SUMMER FUN!
Word Wars
Micro Drafter
Code Master
6502 Kidnapped
Plus:
Chemistry Drill,
Lemonade or Champagne,
Go Public With Your Computer
and much, much more!

Unlock the hidden power of your computer for fast and easy programming! Use ROM routines in your BASIC and Assembly Language programs! All you need to know is in...



INCLUDES:

SUPERMAP
From Fuller Software ($18.95)

TRS-80 DISASSEMBLED HANDBOOK
by Robert Richardson ($10.00)

HEX MEM
by John Phillipp
Monitor written in BASIC

Z-80 DISASSEMBLER
by George Blank

ALL ONLY $19.95
plus $1 shipping

A SoftSide Publication

Guide to Level II BASIC and DOS Source Code

Description of the contents of the Level II BASIC ROM by memory locations, by function, and in lesson format. Includes several BASIC and Assembly Language programs in listing format to examine and use ROM routines.

MasterCard VISA

ORDER TOLL-FREE
(In NH call 673-5144)
1-800-258-1790

The Software Exchange

Hot pursuit through space and the vortices of time!

RAINYWARE PRESENTS. . .

Time Lord

The fallen Time Lord, who presumptuously calls himself The Master, is at large. The elders of Waldrom have supplied you with the hyperspace-worthy vessel TARDIS, and commissioned you to eliminate the evil "Master". Your resources include clones who will fight for you, the formidable CRASER weapons of the TARDIS, and magic weapons such as Fusion Grenades and Borelian Matrix Crystals.

Travelling through hyperspace in search of the evil one, you will encounter Time Eaters, Neutron Storms, and other alien creatures and phenomena. Entering real space to search planets, you will encounter still other dangers. You will enter native settlements to buy food and supplies — or to fight for survival.

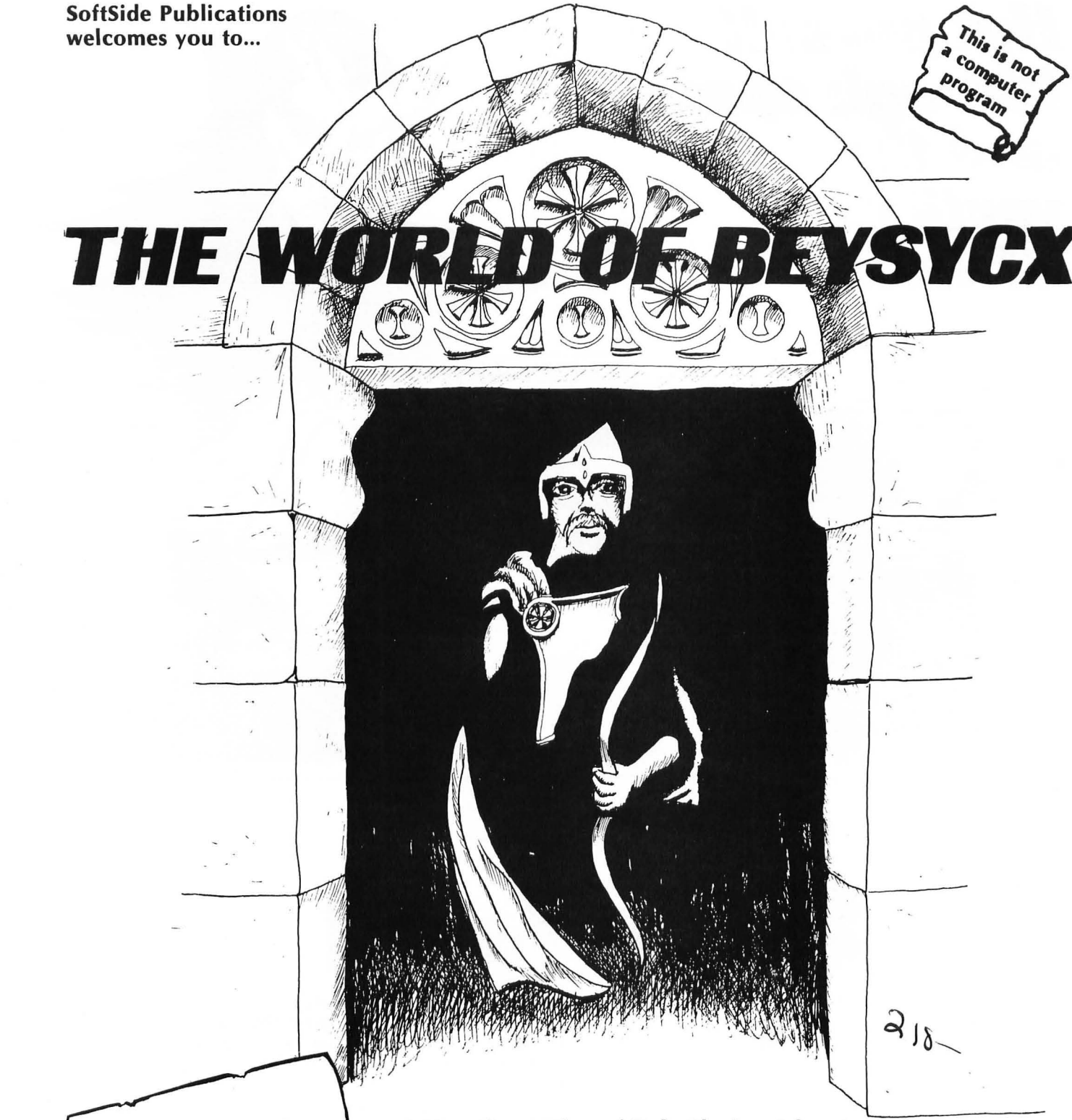
And once you find The Master can you destroy him?

Based on Dr. Who of PBS fame.
Apple Integer Basic,
Disk, 48K . . . $29.95

MasterCard VISA

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

SoftSide Publications welcomes you to...



"...but all legends agree that, somewhere in the wilderness of the Loulanix Mountains, the severed arm of St. Elsinore rests in a temple carved out from the living rock...waiting for those pure in heart and mighty enough to return it to the Elven people."

A New Generation of Role Playing Adventures

With all new monsters and treasures never before encountered, **The Quest for the Arm of St. Elsinore** is the premiere module in the **World of Beysycx** series of role playing adventures.

The **World of Beysycx** series was created for use with the popular Advanced Dungeons & Dragons (tm) gaming system, and is designed as the ultimate in DM playing aids. Each module comes with graphic cardboard map pieces and two fully illustrated booklets detailing the scenario and showing every monster and every treasure.

To order, send $6.95 (plus $1.00 shipping) to:

SoftSide Publications
6 South Street
Milford, NH 03055

Advanced Dungeons & Dragons is a registered trademark of TSR Hobbies.

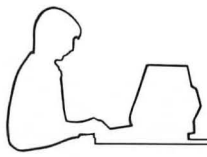


TABLE OF CONTENTS

ARTICLES

32 GO PUBLIC WITH YOUR COMPUTER — Hear ye! Hear ye! ..... Harland Hill
48 LETTER FROM THE PUBLISHER — Magazine of the Future ..... Roger W. Robitaille, Sr.
72 COMPUTER GRAPHICS — Designing your own patterns ..... Joan Truckenbrod
80 LEMONADE OR CHAMPAGNE — Another glass, please ..... Will Hagenbuch
84 VARPTR UNMASKED — Who was that masked VARPTR? ..... John T. Phillipp, M.D.
90 THE LITTLE COMPUTER THAT WOULDN'T — Computer that grounded Columbia .... Sherry M. Taylor
92 REVIEWS — Computer Napoleonics, Atlantian Odyssey ..... Michael Humes, Dave Albert

S-80, APPLE, AND ATARI PROGRAM

22 CHEMISTRY DRILL — Hup, two, three, four ..... Brent Packer, Jon Voskuil

S-80 AND ATARI PROGRAM

28 CODE MASTER — Deciphering game ..... Andrew Braunstein, Rich Bouchard

APPLE AND ATARI PROGRAM

34 KIDNAPPED — Nine floors to freedom ..... Peter Kirsch, Carl Mueller, Rich Bouchard, Alan Zett

S-80 PROGRAMS

50 MAGIC PAPER CALCULATOR — Scrolling calculator ..... Russell Starkey
55 WORD WARS — How many words can you make? ..... Rowland Archer

APPLE PROGRAMS

59 VOLLEYBALL — Game for four players ..... Jim Hilger
66 MICRODRAFTER — Computerized drafting and drawing ..... Jon Voskuil

ATARI PROGRAM

75 SPACE LANDER — Happy Landing! ..... Jack and Bob Wiener

DEPARTMENTS

4 EDITORIAL ..... Jon Voskuil
6 INPUT ..... From our readers
11 OUTGOING MAIL ..... Dave Albert
13 CALENDAR ..... Editors
14 SAY YOHO ..... Scott Adams
15 ABOUT THIS ISSUE ..... *S.S. Munchkins*
17 I DON'T THINK WE'RE IN KANSAS ANYMORE ..... Mark Pelczarski
19 THE SENSUOUS PROGRAMMER ..... "J"
33 WORD SEARCH PUZZLE #3 SOLUTION ..... Editors
33 BUG, WORMS, AND OTHER UNDESIRABLES ..... Editors
86 WHAT'S NEW ..... Ed Umlor
87 HARDWARE CORNER ..... Ed Umlor
93 MACHINE HEAD ..... Spyder

Cover photograph by Mary Locke

Use the following symbols as a guide when reading our ads. They indicate the computer(s) for which the product was designed. TRS-80 APPLE ATARI

*TRS-80, Apple, and Atari are registered trademarks of Tandy Corporation, Apple Computer Company, and Warner Communications.

STAFF

PUBLISHER:

Roger Robitaille Sr.

EDITORIAL DEPARTMENT

Scott Adams
Dave Albert
Rich Bouchard
Michael Humes
Mary Locke
Lance Micklus
Mark Pelczarski
Joan Truckenbrod
Jon Voskuil
Alan Zett

PRODUCTION DEPARTMENT

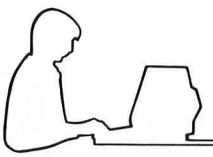
Lynda Fedas
Damian Henriques
Nancy Lapointe
Tom Stanton

STAFF

Patricia Acampora
Lester Anderson
Judie Aveni, **Hardware Information/Shipments**
Brian Berkebile
Diana Bishop, **Subscriptions**
Kathleen Boucher
Suzanne Breton
Philip Brown
Tara Butler, **Dealer Information**
Jeff Carroll
Donna Cookingham
Pam Demmons
Mary Edwards, **Software Information**
Mark Eric, **Atari Submissions**
Mary George
William F. Gollan, **Advertising**
Mylene Grigas
Kathleen Hannon
Donna Jean
Bette Keenan, **Customer Service**
Bea Kimball, **Software Shipments**
Randal Kottwitz
Karen Lawrence
Kathy Maloof
Jean Matthews
Doris Miller
Clem Morey, **Apple Submissions**
Carol Roane
David Robitaille
Elizabeth Robitaille, **Personnel Administrator**
Cindy Schalk
Ken Sicard, **S-80 Submissions**
Christine Spade
Joanne Tracy
Anmar William
Nancy Wood
Ed Umlor
Gary Young
Cynthia Zawacki

SoftSide is published each month by **SoftSide** Publications, 6 South Street, Milford, New Hampshire 03055. Telephone 603-673-0585. Second class postage paid Milford, New Hampshire and pending at additional mailing offices. ISSN: 0274-8630. Subscription rates: USA, $30.00/12 issues. USA First Class APO, FPO, Canada, Mexico, $40.00/12 issues. Overseas air mail: $62.00/12 issues. Media subscription rates: Magazine and cassette, $75.00/12 months. Magazine and disk, $125.00/12 months. APO, FPO, Canada, Mexico, (add), $36.00/12 months. All remittances must be in U.S. funds. Mail subscription inquiries to **SoftSide** Publications, P.O. Box 68, Milford, New Hampshire 03055.

POSTMASTER:
Send address changes to:
SoftSide Publications
6 South Street
Milford, New Hampshire 03055



Editorial

The Magazine of the Future

by Jon Voskuil

In the beginning there was an idea for a new magazine, "conceived and born within four weeks of frantic activity...fraught with all the rough edges that any fledgling publication is heir to." This new magazine was dedicated to that side of personal computing "which allows us to realize our expectations, our fantasies, our dreams." The editors edited, the publishers published, and the printers printed, and there appeared not just another computer magazine, but **SoftSide** — "Your BASIC Software Magazine."

This birth took place in the almost-forgotten ancient time of October, 1978, and was followed by normal growth pains as **SoftSide** concentrated on publishing good, usable programs for owners of S-80 microcomputers.

In the course of time two other magazines were conceived and born under **SoftSide**'s roof: **PROG/80** in March of 1979 and **The S-8IGHTY** in January of 1980. **PROG/80** was "dedicated to the serious programmer," and supplemented the more general character of the programs in **SoftSide**. **The S-8IGHTY** was not a software magazine, but had articles delving into various secrets of the machine. It also offered to its readers very beneficial exposure to independent manufacturers of S-80 peripherals and software. And, it was free.

It also came to pass, in January of 1980, that **SoftSide** began publishing an Apple edition of "Your BASIC Software Magazine." Much to the delight of us Apple owners, the software emphasis was maintained, and we finally had a source of good printed programs ready to type in — not to mention articles that helped us to understand what was happening while the programs were running. A separate edition of **SoftSide** was also planned for the Atari, but it turned out that another major change took its place.

For it happened in the fall of 1980 (that's less than a year ago, folks) that the NEW **SoftSide** was created. Except for **The S-8IGHTY**, which had been of "normal" magazine size, **SoftSide** in its various editions and offshoots had been printed in a small booklet format. Suddenly, it became a large-format magazine serving not one but three computer systems, and inside of two months had a glossy cover and slick pages to boot. All the previous publications were incorporated into the new magazine, and everyone who had been getting more specialized coverage was invited to "enter our (newly expanded) world of microcomputing."

And so it was, and is, that your BASIC software magazine has evolved into its present form. Which brings us at last to the question: What lies ahead?

You must understand that I'm not posing this as a question to be answered by crystal-ball gazing. It's not an inquiry into What Shall Be, but into the direction of the thinking and planning that are going on in our collective mind here at **SoftSide**. And you should be aware that you, as our readers, continually furnish input into that thinking and planning, through your letters, through surveys, and through personal contacts at shows around the country.

Other significant input comes from the various technologies associated with the Information Age — especially the trend toward the storage of information on media other than paper. Since January of 1979 the programs printed in **SoftSide** have been offered optionally on magnetic media in machine-readable form. Up until now, such tape or disk subscriptions have not been emphasized, and they've attracted only that minority of our readers who have been willing to buy the obvious convenience.

Our vision of the **SoftSide** magazine of the future gives the disk edition a much bigger role than mere convenience. We'd like to evolve into a MIXED-media magazine, in which the human-readable printed material and the machine-readable magnetic material support and complement one another. Printed pages are best suited for some kinds of information, and

magnetic disks best suited for other kinds; our aim is to take fuller advantage of the unique strengths of each. **SoftSide** will continue to exist as a printed magazine, with the best value in software available anywhere. But in addition, we'll be working on upgrading (and DOWN-pricing) the disk edition of our publication to offer you the magazine of the future — today.

We believe that our unique position as a software magazine will be expanded and strengthened by this kind of innovation. We also believe that you, our subscribers, are in line to receive the greatest benefit from it. Some of the specifics of the magazine of the future are detailed in the Outgoing Mail column elsewhere in this issue. Other details will evolve over the course of time, and we'll be listening closely to the feedback we receive from you. Please, tell us what you think; give us your ideas; take an active role in shaping **SoftSide**'s future. A year or two from now, we'll probably be heading in some direction that none of us anticipated. But then, that's the fate of us non-crystal-ball gazers.

Input

Dear **SoftSide**,

I just received your April magazine and enjoyed it as usual.

I glanced through the INPUT section and in it I read a letter by Ben Cohen. In his letter he complained about the lack of Adventure games for the Apple II in the magazine. I presume that Ben Cohen and the many Apple II owners he speaks for don't know that if one makes a program and sends it in, that **SoftSide** might put it in their fabulous magazine and even pay for it. I always say if you want it done, do it yourself.

Now concerning the usage of the term TRaSh-80 in the letter. I assume that Ben Cohen and the other people who use the childish term don't use the S-80 and the Apple on a daily basis. I own an S-80 and I use an Apple II+ at school and I find that they are both very good computers.

Ryan Smith
Leucadia, CA

Dear **SoftSide**,

Very clever! I found the egg (as part of the question mark) on the front cover, but only after spending an hour with a magnifying glass examining the bridge and guard rail, for I was sure that was where it was hidden. I would like to take this opportunity to say that I really enjoy your magazine but would make one suggestion — place all articles for the same computer together. It would make it easier to find those programs and stories geared toward the readers' particular computer. Also would like to comment on Dave Albert's column this month. I agree that errors in some of the programs are inevitable. I have found it a great help in sharpening my own programming skills to find the bugs in the programs and fix them myself (although lost lines do create a problem). I have found, however, that my biggest problem are my own typos — especially in data lines. Keep up the good work!

Patricia M. Finkenbine
Albuquerque NM

Dear **SoftSide**,

A debt of thanks should go to Paul Johnson for his article "Atari Memory Upgrade" in March **SoftSide**. I would like to share my experience in board modification with other 800 owners who may be considering the change.

When I first examined my 8K board, I was disappointed to discover an adhesive tab with board number CO13400 on it. When I lifted the tab, embossed underneath was the number mentioned in the article, CO12987, and the crucial "revision 3" marking.

I also noticed that the resistors on my board did not match the arrangement in the diagram. There were no resistors at position "B" or "D", but rather at positions "C" and "E".

Upon reflection, I decided the modification was still worth a try. The board was a match — chances were good that the only difference was in resistor placement.

The gamble paid off. Modification of my board was even easier, because there was already a resistor at position "C".

There were a couple of errors in the article, and they should be mentioned. The article states twice that the CX852 board should be inserted with chips facing front. This is incorrect, and any Atari owner/modifier risks blowing the whole board following this advice. Chips always face away from you when the board is inserted in the computer.

Next, the modified board will read out about 13,300 free bytes with a successful modification — not between 15,000 and 16,000. BASIC does eat up a few K.

I'm sure I speak for many Atari owners who purchased the 800 in its original 8K format when I say how much I appreciated discovering a simple way to double my memory, save a bundle of money, and not end up with an extra RAM cartridge on my way to 48K. I now have a 32K cartridge running fine behind the modified CX852.

J.J. Anderson
Palisades Park, NJ

Dear **SoftSide**,

While looking through your article in the April issue in numerical bases, I noticed something interesting. In the table of binary numbers, the decimal equivalent of 10100 is 20; the shorthand date for January 1, 2000 is 1-01-00. The 21st century, however, does not begin until January 1, 2001, where 21 in binary is 10101. An interesting coincidence!

Gary Erb
Whittier, CA

Dear **SoftSide**,

In "My Side of the Page" in the May issue, Mr. Micklus suggests a strategy for betting in a craps game that involves the use of a "history stack" to keep track of those combinations that have occurred. He then reasons that the machine should bet on those combinations that have not occurred recently.

While completely unsound, this strategy is nonetheless interesting from a historical perspective. A French amateur mathematician and gambler, Chevalier de Mère, analyzed a dice game (simpler than craps) and developed a betting strategy based on his analysis. Alas, he lost his shirt (chemise?). In 1654, he communicated his conviction that mathematical logic did not apply to games of chance to an acquaintance of his named Blaise Pascal (for whom the programming language PASCAL is named). Pascal was intrigued by the problem, saw the errors in de Mère's analysis, and went on to develop probability theory as a legitimate branch of mathematics. (History does not record whether or not Pascal made a killing at the gaming tables — presumably such base pursuits were beneath him.)

The fact is this: Independent events have no memory. If a fair coin is fairly tossed, the odds that it will come up heads are 50-50. If it is tossed 1,000 times and came up tails each time, the odds that it will come up heads on the 1,001st toss are still exactly 50-50. You may question the fairness of the coin or the integrity of the tosser, but if these are above reproach, the odds of coming up heads are always exactly 50-50, regardless of what went on before.

Similarly, fair dice do not remember what they did on previous rolls: the odds that some combination will or will not occur are unaffected by previous events. Thus, the "history stack" mentioned by Mr. Micklus is a useless artifice.

G. Staradub
Nashua, NH

Dear **SoftSide**,

I read Lance Micklus' article in the May '81 issue with enormous interest. I think the odds on his marriage are better than those on his crap game, however. I am a veteran of both (25 years married, 30 years an avid craps player) and I have the feeling you will be getting more letters than this one about the odds...

Lance, teaching yourself the complex rules of casino craps is quite a feat, but

SoftSide™

INPUT POLICY

SoftSide Magazine welcomes your comments and thoughts on both the magazine and the field of microcomputing. We try to publish as many of our readers' letters each issue as we can.

For the sake of clarity and legibility, all letters should be typewritten and double-spaced. Send your letters to:

SoftSide Publications,
Input
6 South St.,
Milford, N.H. 03055

We reserve the right to edit any letters prior to publication.

relying on the house (closed circuit TV) to tell you how to play is like asking Bunyon to get you out of the bog — there's something he's not telling. What the house will never tell you is that the best edge a player can have is betting the odds. There are definite and specific combinations of the dice, yes? Three ways to roll a seven; three ways to roll a six or eight; two ways to roll five or nine; two ways to roll four and ten.

Here's how you play the odds — and you will observe players doing this at the table but never will you find it listed or explained in the official house rules, since it really does give the player a slight edge. You must have a bet on the Pass line. When the number is rolled, the player places another bet up to the amount of the original bet behind the line (toward himself) — this is the odds bet. If the number comes up, you get paid even money on the Pass line and odds on the other. The odds are as follows, and this is standard in any legal casino anywhere in the world: on a six or eight, 6-5; on a five or nine, 3-2; on a four or ten, 2-1.

The same odds and betting privilege (!) apply on the 'Come' bets. (It gets a bit complicated here, since the odds are off the Come bets on the first roll, trying for a new number, if your money is riding from a previous roll — that way you don't lose the odds money if the roller craps out with a 2, 3 or 12; if a 7 is rolled you get the odds money back but lose the placed, or Come, bet.)

I hope you will figure out how to incorporate this into your program, because it is the smart way to bet and the only way many really smart bettors (me) ever bet; a computer program without this option wouldn't be realistic or of much interest, at least for me. On the other hand, I would be your most eager customer for a real simulation of the game using the real odds.

By the way, Atlantic City is not a terrific introduction to casino excitement; in my opinion it is tacky and slow, a two dollar horse trying to enter the Derby. If you ever, ever get a chance to go to Las Vegas, do it! It's very different; any game player has to love it. (I've shot craps in lots of different places all over the world, and Vegas is the best by several lengths).

I wish you enormous luck with your program and will watch for further developments with great interest.

Eileen Lottman
New York, NY

Dear **SoftSide**,

I was very disappointed to see, after receiving my March, 1981, **SoftSide**, that the Developing Data Base series is coming to an end. It was obvious that sooner or later it would have to end, but I believe that this is perhaps the best column I have seen in your publication or any other. I am a beginning programmer and have found this particular series to be more helpful to me in learning and understanding programming than any other I have read, and I am writing you in hopes that you will continue this, or another similar series, in your magazine. My reasoning for this is not due as much to the specific problem which was being addressed by the series, but rather by the format. As Marshal McLuhan said, "The medium is the message". In this case, the medium is a very productive one for learning programming. Breaking this rather large program into small pieces and documenting each individual part to the point that understanding the logic of the program becomes easy, I believe, is a very powerful method for teaching programming in a publication such as yours. Not only did each of us who followed this series end up with a very useful program, we have one that we understand and that can be modified to fit our needs. We each learned many useful programming techniques. Perhaps future data base programs will be seen in the pages of your magazine, or perhaps some other author would be willing to take on a series on this format. The subject matter is, I believe, of little consequence; perhaps an Adventure could be written or a word processing program, but I strongly encourage the editors and staff of the magazine to continue with this fine format which you have developed.

Again, I would like to say that it is my assertion that this is, for a beginner programmer, perhaps the best article I have seen in any publication.

Henry F. Drygas, Jr. M.D.
Olympia, WA

Dear **SoftSide**,

I am especially intrigued by your use of the '&' to poke sound routines into the machine to smooth the tones. I am curious to learn where you discovered such a talisman when said character is not mentioned in any of the owner's manuals. Are there any other uses of the ampersand and how, exactly, does it do what it does? Further, are there any more of these litle gems that can be used without compromising professional programmers trade secrets?

David W. Landry
Saginaw, TX

Editor's Reply: I was introduced to the ampersand by an article in the Apple Edition of SoftSide, April, 1980. The manuals, as you noted, don't say much about it. (See the Applesoft Basic Programming Reference Manual, p. 123.) What it does is to cause a jump to memory location 3F5 (decimal 1013). The sound routine in "Battle at Sea" (which first appeared in the August, 1981, issue) pokes a Machine Language instruction into that address, which in turn passes control to the sound generating routine starting at 300 (decimal 768).
The ampersand, then, can really be used to execute any Machine Language routine, with the proper pokes. You can use it, in the same way, to jump to a built-in monitor routine, such as running, listing, etc. For example, if you POKE 1013,76: POKE 1014,165: POKE 1015,214 then typing & [return] will LIST the program in memory. Other possibilities were listed in the April '80 issue.

Dear **SoftSide**,

Had a few things I wanted to pass on, so this letter will probably end up too long. First, regarding KUDOS AND KLUNKERS (May '81). I just completed converting 64 programs to run on Model III disk, and would like to add a few comments to the article.

BUILD, DO, and PAUSE. To set file size and memory size in BUILD, just type the number and press ENTER (BREAK ends the BUILD program). The only problem I had with it is that the assembled command is loaded into high memory during the DO operation. This is mess if you have the DO command load a a Machine Language routine you wanted put in high memory. In addition, the MEMORY SIZE response won't let you set memory above the bottom of this DO command. Roughly, it seems to take up about 1K memory which is permanently lost to you unless you reset the end of memory pointers with POKEs.

My version of the DOS won't let you turn the DUAL off. Tacky.

I really like CMD"X". This is because I use PEEKs and POKEs a lot, one of which is often to the clock. Since the clock location has been moved from 16449 - 16454 to 16919 - 16924, I used it a lot. It doesn't go to the printer, though. Also CMD"T" turns on the clock display! CMD"R" turns it off.

The Machine Language sort is good, but as Ms. Keller says, it requires a single dimensioned array. It is possible to use this routine to sort anything, if you are a little sneaky. OPEN a dummy random file. Take your array and FIELD it into the buffer. If you have a parallel numeric array, which should also change order with the string array, just field it into the same buffer. Then reload the whole mess back into the single array (strings). Do the sort, go back through the FIELD trick and restore the numeric arrays. This takes a bit of memory, but still is faster than a BASIC sort. I also heard that if you put numbers into the array of strings, they won't sort in numeric order. To get around this, change the number to a string with STR$ rather than MKI$ or whatever. RIGHT justify the number in the field, then sort them. Works fine.

Maybe someday Radio Shack will just have Apparat write their DOS in the first place!

Regarding errors in your program listings. I have found errors in every listing (practically), and every hardware article, in every magazine I've ever read. The solution: Wait a month. Works great, just don't type in any programs until you get the following issue to make the corrections. This is especially true of hardware mods! ('Course, if everybody did that....)

continued on page 9

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

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

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

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

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

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

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

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

MICROSOFT
CONSUMER PRODUCTS

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

continued from page 7

About software piracy. We at the Innovative Perguin write and distribute very high quality software on cassette and diskette. All are in BASIC with imbedded Machine Language and packed super graphics and arrays. All programs come with instructions for modifications. If I cleverly had disabled the break key, made the programs self-starting, and had them delete themselves after running, it would be a little hard for someone to make it exactly what they wanted. Faced with the fact that the Data Dubber will make a perfect copy of any cassette — without a computer, and that as soon as someone figures a new way to make the program copy proof, someone else breaks the code, I just never considered it was something worth doing to make it uncopiable (izzat a word?).

Instead, the brochure which comes with each program says, "This program was copyrighted (date). We encourage you not to violate this copyright by making unauthorized copies of this program, as this denies compensation to those who earned it, and discourages the best programmers. Their loss in this field will eventually affect you. We recommend you do make a copy of this program as a backup for your own use."

Some people probably make bootleg copies, some don't. Frankly, I've got more important things to worry about.

I like your magazine, even if there are all those strange articles about Apples and Ataris. See what the competition is up to! I love one liners. I learn more from a one liner than most long programs.

Dan Connors
Harvey, LA

Dear **SoftSide**,

I enjoyed Judy Neyhart's article "The Money Whirlpool". I was particularly enchanted by her last paragraph:

"...The only difference between men and boys is in the price of their toys". That goes for girls, too!

You mean the only difference between men and boys is in the price of their girls?

I knew I was overlooking something!

Don Berry
Vashon Island, WA

Dear **SoftSide**,

I have been using the disk version of TSHORT for only a month, but I have found several modifications that are helpful for typing programs out of your magazine.

The first changes the MID$ (statement to ELSE, and the second changes the INSTR (statement to PRINT@.

ELSE=

For TSHORT DEC	HEX
POKE 27587,69	45
POKE 27588,76	4C
POKE 27589,83	53
POKE 27590,0	0
POKE 27591,197	C5

For TSHORTR
POKE 28243,69
POKE 28244,76
POKE 28245,83
POKE 28246,0
POKE 28247,197

PRINT@

For TSHORTR DEC	HEX
POKE 28206,80	50
POKE 28207,82	52
POKE 28208,73	49
POKE 28209,78	4E
POKE 28210,84	54
POKE 28211,192	C0

For TSHORT
POKE 27550,80
POKE 27551,82
POKE 27552,73
POKE 27553,78
POKE 27554,84
POKE 27555,192

These can be poked in from BASIC or you can use DEBUG to put the hex values in memory.

The modified version can be saved to disk using the original TSHORT instructions.

Danny Nelsen
Blackwell, OK

Dear **SoftSide**,

I have been a reader of yours since last September, shortly after I brought my S-80. I had looked over the various personal computers, but since my budget was limited, I decided on the S-80, and then lucked out and bought a used one. Oh, I really liked the Hi-Res graphics and the color capability of the Atari and the Apple, but buck$ had to be a primary consideration.

I have received nothing but satisfaction (along with a certain amount of frustration, of course) from my purchase. Although the low resolution graphics don't give me the same sensational sight games as the Hi-Res computers would have, I find that I am satisfied with what I have. And the more than 200,000 other S-80 users and I have something else going for us, too, and that is more than 200,000 programmers, many of which like to contribute software for the enjoyment of all.

By virtue of the S-80's Lo-Res graphics, many of the S-80 games tend towards being thought games as opposed to sight games. We tend to try for more mental adventure than "arcade" adventure. And that seems to be a sore point for some of your readers. The one comment that I have seen in your magazine that used to bother me most was the recurrent use of the term "TRaSh"-80. It is unfortunate that anyone finds it neccessary to take out their frustrations of the lack of software for their machines by dumping on a different brand. I noticed in the April issue, yet another use of that term, as an Apple user — it always seems to be Apple users who use the term "TRaSh"-80 — fussed about the lack of adventure-type programs, pointing out that there had been several S-80 adventures. Tsk, tsk. I wouldn't think that an adventuresome Apple user would have much trouble converting an S-80 adventure to use on an Apple. It's not as easy to convert an Apple program to S-80 usage, alas, because of the usage of color and Hi-Res graphics.

What I would like to see in **SoftSide**, is an article that would let me become more familiar with Apple BASIC, and Atari BASIC, so that I might be able to attempt conversion of the Apple arcade-type games to use on my S-80. Many of these games seem intriguing, and it would be worth the mental workout to try to get them to run on an S-80. It might also help your Apple readers if there was an article on S-80 BASIC, and perhaps a cross-reference as to what term in S-80 BASIC corresponds to what term in Apple/Atari BASIC.

Back to the downcast Apple user. If direct conversion of the S-80 adventures is too much work, perhaps a solution might be the acquisition of a Z-80 softcard, or perhaps — heaven forbid! — the sly acquisition of an S-80 so that you too can enjoy the proliferation of software available for the beast. In all seriousness, though, Apple user, I do envy your graphics and color, but I wouldn't swap machines with you. I can always add the graphics and the color in after-market add-ons when I'm ready, and I am now up and running on a good, usable computer system on a budget that never would have allowed the purchase of an Apple. I respect you and your machine, show a little respect for me and mine.

Thank you, **SoftSide**, for the use of your forum.

Thomas R. O'Hara Jr.
Anson, ME

Dear **SoftSide**,

I used to pick up a copy of the S-80 **SoftSide** when there were several programs I wanted, but in December of last year I subscribed to the cassette version, although I have since upgraded to 48K and two disk drives.

Several of the recent letters to INPUT struck a sympathetic note. First, could you send the magazine in a wrapper? All the other magazines I subscribe to (at $2.00+/copy) come that way. Apparently I'm on the top of the bundle that comes to my area, because my issues have a florescent label "D" stuck in the middle of the cover, and have arrived dog-eared, torn, dirty, and generally looking like they have been on the losing side of a game of Kriegspiel! The covers were just barely still attached, and I suspect that the late arrival of the January issue may have been due to the label and cover being destroyed. Do I really want to pay $24 a year for this?

Next, may I point out the amusing jux-

continued on next page

continued from previous page

taposition of two letters in the February issue? One has an "Editor's Reply" stating that there are no typos in the program listings because they are computer-to-print listings; the one next to it points out that the exclamation point in the "Monster Maze" program should be a greater-than sign. So how did THAT get LLISTed? By the way, has anyone else had the Maze bomb out when the Monster walks through the bottom wall of the maze and off the screen? Or do I still have one more bug in there? I wish that, in addition to the policy of requiring programs to run in both Disk and Level II, you would require that listings be spaced out (no run togetherlineslikeMonsterMazelistings). If it is necessary to compact in order for a program to run in a minimum system, then print a note to that effect, and the typist-programmer can act accordingly.

I also feel that the combined version of **SoftSide** is a loss in quality. I hate discovering that the featured and most interesting sounding program is for some other computer. Also things like the description of the Mini Golf game, which is supposed to have great sound, but only in the Atari version.

I recognize that the possibility of going back to the old multi-version magazine is nil, so I suggest you cut out (or down) on hardware reviews, new product announcements, and cute articles which have nothing to do with programming. I get plenty of that in the other magazines.

I would rather see you expand the descriptions of program operation, and even the game directions, perhaps with examples of the type of input expected at certain points in the game. For instance, we haven't it figured out yet how the "Convoy" Sonar Phase works, or what information you are supposed to get from it. So far we have no idea how deep the sub is, or whether the player has any control over the ship except launching the depth charge. You could also use some regular hints for new subscribers, such as the statement that all programs will run on disk or cassette machines, (the first time I encountered a Lev3 statement, I assumed the program required Disk BASIC; fortunately, I had Microsoft's Level III Basic). Better documentation, please: The Canyon Bomber program doesn't even say what key to use to drop the bombs! You seem to assume that every reader has been using **SoftSide** for years and has all previous issues to refer to.

Finally, can you do anything about getting the cassette and the magazine arrivals to coincide better?

Valerie Vann
Davis, CA

Dear **SoftSide**,

I read Mr. Causer's article on a self-modifying program with interest. He could cut down the area searched for the target token by using a couple of data statements, one just before and one just after the desired search area. The statements should each contain one string item. Before the search reads each of these into a string variable, use VARPTR to get the address of the string variable, and at the address of the string variable +1 and +2 you will find the address of the string in the data statement. Knowing an address just before and an address just after the target area will shorten the search. I can see a possible bug in this search and replace plan. Level II BASIC stores the line number and the address of the next statement at the beginning of each line in binary form. It is entirely possible that one of the bytes of these binary numbers could mimic the target token code and get replaced! It might be better to write a search routine to search for the target line. I won't go into details but the idea is that the first line always starts at the beginning of the program storage area. Write a loop that starts at this point, reads the line number, checks it, and if not the desired line, reads the next line address and repeats for the line at that address. Once the line is found the search and replace loop could be used on the body of that line only.

To change the subject I want to share something that I just found out with the aid of **Pathways Through The ROM.** Either I have just found out what everyone else knew all along, or at this late date I have made a breakthrough concerning the USR function. Contrary to what the Level II manual may have lead you to believe, the USR function is not limited to integer arguments nor is it limited to returning integer values! They didn't really lie to you in the manual, it's just that integer is the only type that they explained how to deal with. To work with the other types you have to know secrets of the ROM that they did not care to reveal. I became curious about the two routines used by the subroutines called by the USR subroutine, so I decided to disassemble them to see how the USR function worked, but first I looked them up in **Pathways**. I was astonished to find the subroutine called by the USR subroutines to get the argument described as a subroutine to convert any numerical type to integer type. This implied that this conversion did not take place until the conversion subroutine was called, and therefore the argument was sitting there waiting in its original form until this subroutine was called. If instead of calling the conversion subroutine, your subroutine simply looked where it was sitting, you could work with arguments in single-precision or double-precision form. There is even a number type flag to tell your subroutine which form the argument is in. Look it up in **Pathways**, the number storage area is referred to as the accumulator in the earlier chapters and as ARITH in the Fuller map.

I now knew that any numerical type could be passed as an argument. Could you return a non-integer type? Sure you can, simply load it into the storage area in the proper form for that type and set the number type flag to indicate its type. I tried some experiments to see if various types would work. I poked a return instruction into protected memory, set the USR address to point to this return instruction and experimented by using the USR function with various arguments. If the value returned was of the same type and value as the argument, it boded well for being able to pass and return *that* type. I found that it worked for any numerical type. Then I tried it for string arguments. To my astonishment it worked for them also. I have not yet found out how it handles string arguments and returns string values but I will. A bit of disassembly indicates that there may even be different systems for string constants and string variables. I have not yet had time to check this out in as much detail as I would like. There are too many other things competing for my time. (In fact I should be doing my income tax instead of writing this letter.) When I have time one of the first USR subroutines that I intend to write is one for converting addresses above 32767 into the negative integer form required for PEEK and POKE.

David S. Tilton
Manchester, NH

★★ REWARD! ★★
TRANSLATION APPEAL

We will give away a $100 software certificate each month for the best translation of a feature program in **SoftSide** magazine. Furthermore, we will publish the translation in the magazine. Your portfolio will be enhanced and you will garner fame and fortune for your efforts!

We will allow three months after initial publication of a program for the translation to be sent. After that time we will not accept entries. The quality of the translation will be judged by the **SoftSide** editorial staff and the winning entry will be published the following month, i.e., four months after publication of the original program.

Entries must be submitted on cassette or disk, accompanied by documentation and a line listing of the program. Please enclose a self-addressed stamped envelope if you would like your entry returned to you.

Outgoing Mail

by Dave Albert

Hello, hello...it's time for another glimpse into the confused workings of your favorite software magazine. The last couple of times I've written this column, I've spent most of the space on answering questions and trying to clear up things. This time I intend to devote most of the column to talking about the future. But before I do that, a message to you Apple and Atari owners: There is an Adventure in this issue for you guys! But I'm still waiting for an original submission for either of these systems...we translated an S-80 one that first appeared in December '80, just to make you happy. Now it's your turn...

Those of you perspicacious enough to notice our mailing and subscription rates information at the bottom of the masthead and on the mailing cards may have perceived that our price has gone up. That's right, the days of cheap **SoftSide**s are numbered. There are several reasons for this, none of which are the one that most readers will assume: greed. No, we're not in the business for the money only, we like what we do. However, between the constantly escalating postal rates, the cost of paper and of printing services, and the prices we must pay to bring you top-notch software, we may be in trouble if we don't take steps to keep ourselves healthy. One of those steps, and only one, mind you, is to raise our rates. So, effective this month, the national subscription rate is $30.00. Our Charter rate for new subscribers is now $24.00. However, for those of you who have stood by us in the past, we are keeping our renewal rate to a mere $21.00, just to show we appreciate your support.

Okay. Now to the future. If you read Jon Voskuil's editorial this issue, you'll already know that we have big plans for **SoftSide**. If not, bear with me and I'll try to fill you in.

SoftSide is about software, pure and simple. The whole point of the magazine is to get you, the reader, the best software we can find for the *lowest* price we can manage. Altruistic, aren't we? Not really, we're a business, if we don't make money, we sink. But we are not a publication devoted to programming, hardware, computers in general, or anything else, although we do include articles about such topics. What we are about is software. Strictly software.

We have been "Your BASIC Software Magazine" since Day One, publishing only BASIC programs, despite appeals by you folks for Machine Language and/or programs in other languages. The reason for this was simply that BASIC is the most common microcomputer language, and that BASIC listings can be printed easily without consuming an inordinate number of pages. Hybrid programs, ones written with different languages incorporated into a single program, are often the most interesting submissions we receive, but they list out as complete garbage. Machine Language programs also make for extensive and confusing listings. So we simply didn't publish them. Well that was frustrating for both you and for us.

After much weeping, wailing, and gnashing of teeth, we think we have hit upon a solution: a mixed-media magazine, the magazine of the future. Sounds catchy, doesn't it? Now if only we knew what it was...

People read printed matter on paper. Computers read magnetic media: cassettes, disks, etc. Our magazine is for people and computers, although I must confess that I have yet to meet a computer that has subscribed on its own, P-1 notwithstanding. We believe that **SoftSide**'s truest form would be one that encompassed both paper and magnetic media in one single entity. That isn't really possible, at least until someone makes a print scanner available for the micro owner, or until people learn to read disks. So a compromise must be struck.

We thought that our media editions were a compromise, but really they were more a matter of convenience, i.e., they saved a lot of typing, but they really only duplicated the printed content of the magazine. In our never-ending quest for Truth, Justice, and the Perfect Software Magazine, we have hit upon a new wrinkle: the aforementioned magazine of the future. We will continue to produce **SoftSide** as we have in the past; there will always be ten or so good programs listed in it, ready to type in, as well as articles, One Liners, Programming Hints, and whatever else we think fits. But the media edition is going to undergo a radical change. We are reducing the price of the disk subscription to get it rolling, although not the cassette, and adding a lot more to the disk version. Each disk will have several additional programs on it. These programs will be the ones we think are good, but that for one reason or another we were unable to put into the magazine itself. The disk version will be the deluxe edition.

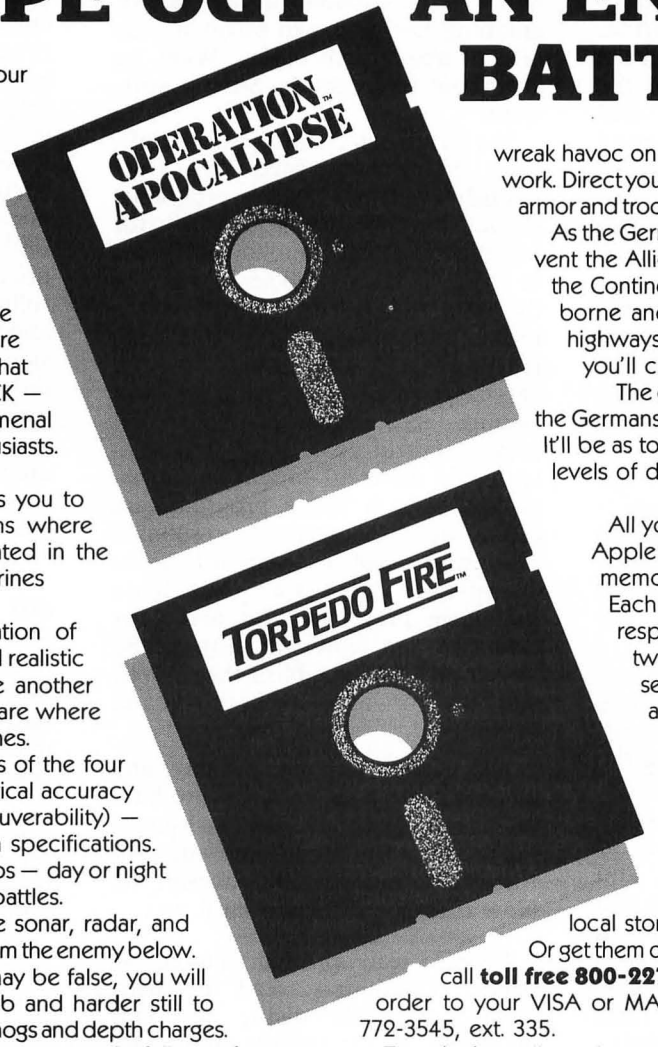
The reason for the emphasis on the disk is simply that to us, the floppy disk is the natural medium for microcomputers. Cassettes work, but they were designed for audio and hastily thrown into the gap when micros emerged. The disk was created to be used in conjunction with a computer and nothing else. Furthermore, the capabilities of disk far outstrip those of cassette. Saving large chunks of data is far more reliable on disk, and the I/O capacities in themselves justify going to disk. And then there's always the consideration of speed of access and the potential for interaction during operation of a program...

The advantages to the media editions are more than just the convenience and the extra programs. For one, we will be able to publish and/or supply you with general controller programs. Think of an Adventure that you don't type in and therefore don't know the answers to, or a monthly crossword puzzle on your computer. Some kinds of games and diversions simply don't work if you have to type them in...the mixed media editions will allow us to publish that type of program.

I-$tring, of which you'll be hearing more in the near future, is the type of program that has a thousand appli-

continued on page 13

THESE HARMLESS DISCS CONTAIN ENOUGH EXPLOSIVES TO SINK A DESTROYER. OR WIPE OUT AN ENTIRE BATTALION.



These mini-floppy discs for your Apple® computer represent the culmination of our extensive R&D efforts to bring you state-of-the-art strategy games in submarine and land warfare: TORPEDO FIRE and OPERATION APOCALYPSE.

These power-packed games from Strategic Simulations Inc. are designed with the same loving care and uncompromising standards that have made COMPUTER BISMARCK — our flagship game — a phenomenal success hailed by critics and enthusiasts.

TORPEDO FIRE™ takes you to the high seas and murky depths where you'll play both hunter and hunted in the desperate battle between submarines and convoy escorts.

You are given the sophistication of simultaneous order execution and realistic sighting rules. You can challenge another player or engage in solitaire warfare where the computer plays the submarines.

Create your fleet from 30 ships of the four major navies (all rated with historical accuracy for speed, weaponry, and maneuverability) — or design the ships to your own specifications. Make up any multitude of scenarios — day or night actions, single- or multiple-ship battles.

As the escort commander, use sonar, radar, and your eyes to protect the convoy from the enemy below. Since the sightings you receive may be false, you will be hard pressed to track the sub and harder still to force it to surface with your hedgehogs and depth charges.

As the submarine commander, you can make full use of the remarkable computer-generated Hi-Res periscope view to locate your prey. You must then destroy the convoy, attack or evade the escorts — all with utmost stealth, lest the seas become your watery grave.

OPERATION APOCALYPSE™ carries you to the Western Front, circa 1944. You have the opportunity to re-enact the various facets of the Invasion of Europe in four separate scenarios, each offering different victory conditions, personnel, ordnance, and terrain.

OPERATION APOCALYPSE uses a revolutionary terrain and movement system designed to give you easy and complete control over your forces: engineer, infantry, artillery, and armor units. The action takes place on a 7-by-18 hexagon mapboard dotted with hills, rivers, bridges, forests, and towns. For further battlefield realism, the game also offers hidden movement.

As the Allied General, you can order off-screen artillery bombardment to soften up German resistance. Or call upon airborne landings behind enemy lines to capture key bridges or to wreak havoc on the enemy's communications network. Direct your engineers to build bridges so your armor and troops can roll towards their objectives.

As the German High Command, you must prevent the Allies from gaining a firm foothold on the Continent by quickly wiping out their airborne and amphibious landings. Sever the highways and bridges to Germany, and you'll cripple the Allied advance.

The computer is ready to take you on as the Germans anytime you want a solitaire game. It'll be as tough as you like since you have four levels of difficulty to choose from.

All you need to play both games are an Apple II with Applesoft ROM card, 48K memory, and a mini floppy disc drive. Each for $59.95, both come with their respective program disc, a rule book, two mapboard cards (for plotting secret strategies between moves), and various player-aid charts.

Without a doubt, TORPEDO FIRE and OPERATION APOCALYPSE represent the finest computer wargames available, head and shoulders above their competition.

So why wait? Hurry down to your local store and get your copies today!

Or get them directly from SSI. Credit card holders, call **toll free 800-227-1617, ext. 335** and charge your order to your VISA or MASTERCARD. In California, call 800-772-3545, ext. 335.

To order by mail, send your check to: Strategic Simulations Inc., Dept. S2, 465 Fairchild Drive, Suite 108, Mountain View, CA 94043. All our games carry a 14-day money back guarantee.

SSI

SSI's other games for your Apple:

COMPUTER BISMARCK,* $59.95	**COMPUTER CONFLICT, $39.95**
COMPUTER AMBUSH, $59.95	**COMPUTER AIR COMBAT, $59.95**
COMPUTER NAPOLEONICS, $59.95	**THE WARP FACTOR, $39.95**
COMPUTER QUARTERBACK, $39.95	**CARTELS & CUTTHROATS, $39.95**

*** Also available for the TRS-80 — Disc, $59.95; Cassette, $49.95**

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

TRS-80 is a registered trademark of Tandy Corporation.

Apple is a registered trademark of Apple Computer Inc.

continued from page 11

cations, if only the data bases are there. We can't publish such banks of data in the magazine; it's impossible to type in several pages of numbers without making a mistake, and it's even harder to find the mistake! But is would be easy to supply the necessary data on a disk.

We will use the magazine itself to provide instructions and notes on the disk version programs, but we could not publish all the programs we want to without tripling the size of the magazine, something we cannot afford to do. We feel pretty certain that some of you will be quite angry to see us do this, but it isn't our intention to tease you with descriptions of programs that you can't use unless you subscribe to the disk edition. There's just no other way to move ahead. The person who subscribes to the magazine alone will continue to receive what he's used to getting: Anywhere from three to six programs for his computer,ready to be typed in. Those who subscribe to the disk edition will be getting a lot more: More programs, and the first true Computer Age magazine.

Write to us and let us know how you feel about all this. We're quite excited about it, and we think that you will be too. Our game is to get good software to you as inexpensively as possible. We feel that the whole software situation is getting out of hand, so we are taking steps to do what we can to bring it back down to a realistic level. We feel that a lot of the cost of software is unnecessary, stemming from packaging, attempts to avoid privacy, and other factors all too human and not particularly human. So we're trying to provide a better way of getting and dealing with software. Your response will tell us whether we'll succeed or not.

CALENDAR

July 15-17
Summer Computer Simulation Conference
Washington, D.C.
This conference will present over 40 sessions on topics of computer simulation technology and applications.
Contact: William E. Buchanan, Applied Physics Laboratory, Johns Hopkins Road, Laurel, MD 20810.

July 19-24 and 26-31
1981 National Computer Camp
Moodus, CT
This overnight camp will be directed by Dr. Michael Zabinski, Professor at Fairfield University. Campers aged 10-17 will have small group instruction and hands-on experience with computers.
Contact: Michael Zabinski at (203) 795-9069, or write Computer Camp, Grand View Lodge, Box 22, Moodus, CT 06469.

July 29-31
The 1981 Microcomputer Show
Wembly Conference Centre, London, England
Seminars on microcomputer applications in business, production, and education will be presented. Topics for conference sessions include hardware availability; software packages and development; automatic test equipment; robotics; and process control. Exhibits from major European and American manufacturers will be featured.
Contact: TMAC, 680 Beach St., Suite 428, San Francisco, CA 94109, (800) 227-3477, in CA call (415) 474-3000.

August 3-7
Introduction to Data Processing Course
Rochester, NY
This course, offered by the Rochester Institute of Technology, will provide deaf adults with introductory technical skills applicable to job situations involving computers.
Contact: Donald Beil, NTID Data Processing Dept., Rochester Institute of Technology, One Lomb Memorial Drive, Rochester, NY 14623, or call (716) 475-6373 (voice or TTY).

August 10-14
Advanced Data Processing Course
Rochester, NY
Deaf Adults who are experienced with computers will gain knowledge of software applications on small computer systems. Topics include: data bases, interactive programming packages, and color graphics.
Contact: Donald Beil, NTID Data Processing Dept., Rochester Institute of Technology, One Lomb Memorial Drive, Rochester, NY 14623, or call (716) 475-6373 (voice or TTY).

ATTENTION AUTHORS

SoftSide magazine, the leader in the field of BASIC software programming for home computer applications, is actively seeking program and article submissions for the more popular home microcomputers, as well as for product reviews. This is your chance to make some extra cash and become famous in the process!

We are interested in programs written in BASIC with any alternate language subroutines worked into the program only within the framework of BASIC. Games and educational software, as well as any other applications for the home computer user are preferred, although we will consider virtually any type of program.

We are looking for well-written, informed reviews of all software for the popular home computers for publication in the magazine. Reviews should take into consideration all aspects of a particular software package, from speed of execution to programming creativity to the estimated length of time that the product will hold the consumer's interest.

When submitting a program, please be sure to include full documentation of subroutines and a list of variables, as well as a brief article describing the program. All such text, as well as article and product review submissions, should be typewritten and double-spaced. Programs should be submitted on a good cassette or disk, and should function under both Level II and Disk BASIC.

Send to:

SoftSidePublications
SUBMISSIONS DEPARTMENT
6 South Street
Milford, NH, 03055

Be sure to send for our free Author's Guide.

We regret that due to the volume of submissions we receive, we are unable to return your cassettes or disks.

Say Yoho

by Scott Adams

While reading the April issue of **SoftSide** I came across an interesting letter by a gentleman in New York who seemed upset about my column on software piracy, (see letter at right). I would like to answer him here in my column.

His main argument appears to be that programs are like books — one person can read a book and then give it to a friend without fear of moral or legal repercussions. I agree 100%. Giving your friend your original copy of a program that you no longer want is indeed well within your rights. But public libraries don't give out copies of books that they have made themselves, they only lend out the originals!

If you give your friend a copy of, say "Adventureland", how does that hurt me, the author? Well, not really of course, if that was the extent of it. What inevitably happens in a case like this is that one copy is then copied and given to a friend of the friend and so on and so on. Now if everybody gave just one friend a copy of a program they had bought, it would cut software sales in half. Now think of the reduction when one original ends up generating 10 or 20 or even 50 "copies"! Yes, I do feel very strongly that giving your friend a copy of any copyrighted material is both legally and morally wrong. Not only does the author suffer, but in the end everybody in the whole industry suffers!

Another point raised was that I have "taken" more from the industry than I have ever given. I really hope that he is wrong, as this type of accusation really makes me feel bad. But I will leave it up to you out there in the real world. Write and tell me if you feel this is true. I have always tried to only market software that meets what I call the Cinema Ticket Rule: You should get as many hours of enjoyment per dollar spent for a program as you would per dollar spent on cinema tickets.

Until next month, keep on computing...

Dear **SoftSide**,

In your February, 1981, issue, there is a column by Scott Adams ostensibly concerning software pirating. I hope Mr. Adams is more careful in his programming than he is in his thinking about this problem.

Pirating is defined in the lawsuit he mentions in his postscript. It has nothing to do with the argument he had with the man who offered to give a friend a copy of a program. Pirating involves taking the creation of an author, copying and then selling it without the authorization of the author.

When Mr. Adams contracted to write a column for **SoftSide**, the contract was not only with the publisher but with the public that reads his column. If he actually perceived the "English High Court" decision to be on a case analagous to the one he argued with the fellow copying a program for a friend, then I suggest that he stick with programming in his column and leave the heavier subjects to people who know what they are talking about. If, on the other hand, he knew full well that the case did not apply, then he owes an apology to his readers for his attempt to mislead them.

Did he ever lend a friend a book and then feel guilty about the money that was not going to the author? Would he suggest that a snow thrower company sue the three neighbors that chipped in and share one machine? Would he close all libraries? Would he outlaw VCRs and tape recorders? How much does Mr. Adams pay Tandy for creating a market for his programs? Or is he leeching off of their huge gamble?

The problem with Mr. Adams and his ilk is that they fail to see anything but their own parochial interest. They easily forget what they owe to their contemporaries and predecessors. I'm sure Mr. Adams never inquired of Microsoft if he could use their BASIC in his programs. How much has he contributed to Dartmouth?

The fact is that Mr. Adams lives in a community and not by himself, and he gets from and gives to this community. I would suggest that, to-date, he has taken much more than he has given.

I believe program pirates should be arrested and jailed. But, let's be quite precise about this; it does no one any good to completely confuse the issue. The pirate attempts, in an organized manner, to seize the market of the legitimate producer. What one friend gives or lends to another is neither the government's nor Mr. Adams' business.

Eli Passin
New York, NY

About This Issue

Hi there, this is your old Uncle Fred, announcing for the very last time all the late replies to our Easter Egg hunt. First I want to take this here opportunity to say that there were actually a couple of you folks out there who found the egg where it wasn't, which is to say that you were looking too hard. It was under the question mark (?) on the line about the chairman of the board. The stragglers were:

James Bash, Staten Island, NY.
The Mad Programmer, Romville Cybernia (who are you trying to kid?)
Mike Quinn, Hazlet, NJ.
Dale Krumel, San Francisco, CA.
Mark & Erik Subba-Rad, Mendham, NH.
Kyle Wadsten, Concord, MI.
Daniel Wood, Oregon House, CA.
David Meile, San Francisco, CA.
Richard J. Croak, Gouldsboro, PA.
Ian Smith, Rochester, NY.
Chris Bell, Alma, AR
Tilden A. Smith, Punta Arenas, Chile.

So much for the eggheads out there. Now back to the real reason for this waste of space: the contents of this issue. We've got all sorts of stuff in here this month, more than ever before. To begin with, in response to all the wailing and caterwauling we get from the Adventuresome types in 6502 Land, we present "Kidnapped" Redux, a couple of virtuoso translations of Peter Kirsch's December '80 cover program by Carl Mueller and Rich Bouchard (with a bit of help from the newest member of our programming staff: Alan Zett).

And for all you budding chemistry majors, sergeants too, we have "Chemistry Drill", by Brent Packer, with translations provided by Jon Voskuil.

For all you erstwhile crytographers, here's a program in both S-80 and Atari lingo for you: "Code Master" by Victor Meyer and, you guessed it, R. Bouchard, Esq.

Being as it's summer, or at least hot enough to be so considered, here's an outdoor favorite: "Volleyball" for the Apple, written by Jim Hilger. The munchkin summer favorite is beer, but then who asked us anyway?

And that Voskuil fellow, feeling left out now that the Math Olympics have drawn to a close, has come up with yet another in a brilliant series of programs: "Microdrafter". No, you needn't send your Apple to Canada, this isn't a conscription effort at all, but a program that allows you to draw all sorts of pictures and/or shapes.

For you S-80 owners it's words and figures month..."Magic Paper Calculator" by Russell Starkey is our first offering, a lineprinter calculator that'll keep track of all that magic paper. And in the words dept., we offer you "Word Wars" by Rowland Archer. Homonyms with bazookas, adverbs that pack M-16s, and even an occasional mortar-toting metaphor...Actually, "Word Wars" is a thinly disguised version of an obscure game by a not so obscure game company. Bet you can't tell me what game it's based on...

We also have the second installment of Will Hagenbuch's book, "Lemonade or Champagne" in this issue, along with the first in a series of articles dealing with the Level II VARPTR command, skillfully explained by John T. Phillipp, M.D. And that Penguin fellow returns to our pages with "I Don't Think We're in Kansas Anymore..." Plus Harland Hill urges you all to "Go Public with Your Computer" and Joan Truckenbrod continues to generate patterns at an astounding rate. And that mysterious "J" fellow generates yet another in a seemingly endless series of columns.

Suddenly we are run over by a truck.

ATARI ONE LINERS

```
1 GRAPHICS 19:FOR A=1 TO 30:FOR C=1 T
O 3:COLOR C:DRAWTO RND(1)*39,RND(1)*2
1:SETCOLOR C-1,RND(1)*15,RND(1)*15:NE
XT C:NEXT A:GOTO 1
```

Harry Caporuscio
Long Beach, CA

```
1 INPUT P:GRAPHICS 22:FOR A=0 TO 95:Y
=A/32-2:Y=Y*4:FOR B=0 TO 159:X=B/14-5
.5:Z=INT(P*(X*X+Y))*0.5:COLOR Z=INT(Z
):PLOT B,A:NEXT B:NEXT A
```

Allen Middleton
St. Paul, MN

RAMWARE

APL

by Phelps Gates

Now a high-level, scientific programming language for the home computer that doesn't cost $200 or $300. The power of this language is in its strong mathematical operations, especially with regard to matrices and vectors. Programs requiring matrix multiplication or other matrix problem solving that would require hours of programming time in BASIC are solved quickly and with minimal effort in APL.

To aid in learning APL, lessons are included on the disk. Starting from the basics, you are brought step by step through the various programming techniques involved with APL. These lessons act as a tutor which will have you "talking APL" in no time. Also available is the book, "APL: An Interactive Approach," which reinforces many of the examples given in the lessons and provides additional insight into APL programming.

FEATURES

APL-80 on disk contains the following features:)SAVE and)LOAD workspace on disk;)COPY other workspaces into current ones; Return to DOS for directory or commands without losing your workspace; Send output to lineprinter; Five workspaces of lessons included; Sequential and random files; 15 digit precision; Monadic and dyadic transposition; Easy editing within FUNCTION lines; Latent expressions (FUNCTION can "come up running" when loaded); Tracing of function execution; Real-time clock; User-control of random link; Workspace is 25587 bytes (in 48K machine); Arrays may have up to 63 dimensions.

COMMANDS APL-80

APL-80 supports the following commands; Absolute value, add, and assign, branch, catenate, ceiling, chr$/asc, circular, combinational, comment, compress, deal, decode, divide, drop, encode, equal, expand, exponential, factorial, floor, format, grade down, grade up, greater, greater/equal, index generator, indexing, index of, inter product, label, less, less/equal, logarithm, maximum, member, minimum, multiple, nand, negate, nor, not, not equal, or, outer product, peek, poke, quad, quote quad, random, ravel, reciprocal, reduction, reshape, residue, reverse, rotate, scan, shape, sign, system, subtract, take, transposition.

SPECIFICATIONS

Minimum system requirements: 32K disk system (&48K recommended) includes APL-80, Five workshapes of lessons, instruction manual. ..... $39.95 on disk

Reduced feature: 16K Level II tape version, no lessons.
Transpositions, format, and inner product not implemented. Reduced domain for some functions, 6 digit accuracy.
..... $14.95 on cassette

LIMITATIONS

Due to the absence of the special APL character set on the TRS-80 , APL-80 uses shifted letters to represent the various APL characters. In addition to the keyboard limitations, lamination, domino, and matris inverse are not implemented but can be derived with user-defined functions. Multiple specifications must be split into two statements unless the left-hand assignment is to a quad. This also applies to implied multiple specifications. Reduction and reshape (p) are not permitted for empty arguments; the argument of add/drop may not be scalar; empty indices are not permitted. A quad (q) can't be typed in response to a quad (nor can the name of a function which itself gets input from a quad). Quote-quad (m) is permitted. No more than 32 user functions can be defined in a single workspace and a function may not contain more than 255 lines.
A comment (c) must occupy a separate line: a comment can't follow a function statement on the same line.
In the tape version, arrays are limited to five (5) dimensions.

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

I Don't Think We're In Kansas Anymore...

by Mark Pelczarski

Well, you may have noticed that this here column wasn't in the magazine last month. That's 'cause ah was humilerated in the May isher by the guy from Kansas who said ah ain't never been thar. His letter's over thar on the bottom right.

So ah went thar to check it out. Couldn't fahnd it for a bit though. Seems thar most famous city ain't even in Kansas; it's in Missoura! Now what the heck do you call that kahnd of thankin'?

Well, ah got thar all raht, after checkin' out the Royals. They sure been playin' sloppy lately. Not lahk the White Sox, although ah shouldn't talk so soon. After all, this is only May, and bah the tahm y'all read this ah may be etin' mah words. At least ah'm not one of them Cub fans.

Well, Kansas was thar. Sure was. Gosh, it was excitin'. Darn near as excitin' as good ol' Southern Illinoise. Someone said they even had thirteen whole computer stores in the state, one for each of the original thirteen colonies, although ah didn't get to see any of them, so ah really couldn't say fo' sure. They said one was in Pittsburg, which ah coulda swore used to be in Pennsylvaney. They sure don't know whar to put thar towns down thar.

Now some people may be settin' an' thankin' "he's shor makin' fun of those people down thar, and with no good raisen at all!" Weel, that's not raht, 'cause ah ain't got nothing against nobody down thar at all. In fact, they is all mahty nice people, even though ah was jus' passin' through. Mahty nice indeed!

So what the hayek is all this stuff about "not bein' in Kansas no more?" Well, you all remember that wizard movie; the one whar that girl an' her dawg Toto gets whisked away by a tornader to this munchkinland, sorta like a fairy tale? Well, when she walks out of her house an' fahnds that she's in this weird place, she says to her dawg, "I certainly don't think this is Kansas, Toto."

Well, it's kahnda funny, 'cause when ah desahded to name this column, ah was thankin' 'bout the last buncha years and how thangs have changed from twenty or thirty years ago to this fairy tale world with all kahnds of gizmos an' gadgeos, an' little preten' creepy crawlers on television screens that you preten' to shoot down, an' magic boxes that do this an' that, an' ah thought to mahself, "this certainly isn't Kansas anymore..."

Sometahms all this stuff seems to make real little sense, 'specially when people get all wrapped up in it an' ferget the rest of what's aroun' them an' all. Ah mean, the gizmos and gadgeos are neat an' all, but they gotta be kept in prespective. Ah'm not gonna talk about prespectives, 'cause everyone's got thar own that's raht for them. But sometimes it heps for everyone to stan' back an' look at what that prespective is an' all. You know, in the movie, the ozzie part is the most fascinatin' an' all, with all kinds of interestin' stuff goin' on. But what was it she kep' sayin' at the end?

"There's no place like home, there's no place like home..."

Dear **SoftSide**,

Attn: Mark Pelczarski

About "I Don't Think We're in Kansas Anymore":

I think you ain't never bin in Kansas. At least your writin's ain't from Kansas. If you ever growed up in Kansas your parents would have learnt you everything they knowed and you'd git shoved up the crick if you done wrong. So don't claim you was ever in Kansas 'cause you ain't hep enough to understand Kansas lingo.

Andy Anderson
Holton, KS

Let PASCAL-80 talk some sense into your computer

Phelps Gates, the author of "APL-80", brings you "Pascal-80" for your S-80. Now you can add another dimension to your programming skills by using this fast version of the compiled language Pascal.

"Pascal-80" is a powerful, structured and well-defined language for the S-80 microcomputer. This easy-to-use language makes writing well-structured, and therefore easily understandable programs simple. "Pascal-80" supports most of the features of UCSD Pascal, including RECORD, SET (to 256 members), FILE (text and record oriented), n-dimensional ARRAY (and ARRAY of ARRAY, etc.), global GOTO ELSE in CASE statements, and BCD arithmetic accurate to a full 14 places (including log and trig functions), 6-digit optional. "Pascal-80"features a 23600 byte workspace in 48K, a 1000 line per minute compiler, an easy-to-use text editor, and plain English error messages, all the features you would expect in a Pascal costing hundreds more.

Variable Types: . . . . Boolean, integer, char, real, real6, and text.
Constants: . . . . . Maxint, minint, true, false, and pi.
Files: . . . . . Input, output, and lp.
Procedures:. . . . . Read, readin, write, writein, reset, rewrite, close, seek, cls, and poke.
Functions: . . . . . Abs, arctan, call, chr, cos, eof, eoin, exp, inkey, in, mem, odd, ord, peek, pred, round, sin, signif, sqr, sqrt, succ, and trunc.

"Pascal-80" does not implement variant records, pointer and window variables, or functions and procedures used as parameters.

S-80 32K Disk $99.95

MasterCard VISA

RAMWARE™

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

The Sensuous Programmer

by "J"

THE THIRD SEDUCTIVE INSTALLMENT

Last month left us out on the street, trying to capture disreputable characters hanging around in keyboard buffers. This month: the other side of the street, and how to kick characters out onto it.

To refresh your memory (and to reassure you that this is indeed a column about computer programming), we're talking about communication between human and machine. Specifically, we're talking about the two-way interaction between the user and the computer that takes place during the execution of a BASIC program, which the programmer builds into that program when he writes it. Having covered several statements dealing with input last month, including INPUT, GET, INKEY$, LINE INPUT, and a special PEEK, this column will cover the other side of the communication street, output.

Who among us has not begun a programming career by tremulously typing in a program such as

```
10 PRINT "HELLO"
```

and then sat in reverent awe as the word "HELLO" was printed on the screen right under the RUN command? It's possible to avoid using a single PRINT statement in a BASIC program and still have it do something meaningful, but it's not very common. Even a graphics display program will normally have PRINT statements to give some kind of introductory information.

Very often the output of a program — what it prints on the screen and the format in which it prints it — is dismally neglected. It's easy to be sloppy or cursory here, because the operation of the program doesn't usually depend on how great or awful the screen looks. But the appearance of the screen does affect the "feel" of a program. A lot of programs have a TACKY feel to them. They're full of misspelled words, misused or nonexistent punctuation, and screenfulls of cluttered and hard-to-read information.

Now, I have sympathy for people who are poor spellers; I realize that overall intelligence and personal worth are not really linked to one's ability to spell "querulous" or

"sacrilegious". I married a self-confessed poor speller, and have finally come to accept this handicap as a minor brain defect in an otherwise extremely intelligent person! (We did get off to a bad start, though, when I began returning love letters with misspelled words marked in red.) NEVERTHELESS, computer programs, like magazine articles, should not have misspellings in them. (I dump all responsibility for any such glitches in this article onto the typesetters.)

If I sound just like your high school English teacher, read on, as I make a few suggestions on punctuation and the general appearance of the screen/text display. Then I'll return to the question of getting the display to look that way using the available BASIC statements. Here are the suggestions:

1. A comma should normally be followed by a space.

2. A period, question mark, or exclamation point at the end of a sentence should be followed by TWO spaces.

3. The apostrophe is commonly used to massacre possessive pronouns and other parts of speech. The most frequent problems are with contractions versus possessives. Please remember that "you're" and "it's" are contractions meaning "you are" and "it is"; avoid confusing them with the words "your" and "its" which show ownership. ("It's your computer, and you're its owner.")

4. Paragraphs should have their first line indented, or preceded by a blank line, or both, for good readability.

5. For a short message or block of text, it's usually worth the extra effort to display it near the center of the screen, instead of just sticking it on any old line.

6. Keep the screen clear of unneeded text. At the same time, keep needed text conveniently located on the screen.

7. Use a variety of approaches to displaying text, to maintain the user's attention.

The formatting of a program's output deserves careful attention. When that attention isn't given, it shows. A lot of the programs submitted to computer magazines for publication are rejected not so much because they're poorly conceived, but because they're poorly executed — they have a tacky, ill-mannered feel rather than a friendly, well-bred feel. So let's jump into the BASICs of the output game.

To master output, you must of course master the PRINT statement. PRINT will display on the screen, starting at the current position of the output cursor, whatever numerical or string variable or constant (or combination thereof) that you specify. Here's a statement that will print one string constant, one numerical variable, one numerical constant, and one string variable:

```
10 PRINT "STOCK NUMBER:", N; 884; Z$
```

continued on next page

continued from previous page

Remember that the difference between a constant and a variable is that the value assigned to a variable can change, whereas a constant has a self-defined, unchanging value. The numerical variable N can be assigned the values 2 or -34 or 16.66667, all three of which are themselves numerical constants. The string variable Z$ can be assigned the value "GTE399" or "X*#;)$+%!?" or even "Z$" all of which are string constants. Note the difference between the variable Z$ and the constant "Z$": To the computer, the former is like the name label on an internal mailbox, and the latter is what is contained in that mailbox.

The various items in the above PRINT are separated by punctuation marks. The comma forces the next characters to be printed starting at the next built-in "tab stop" on the screen; the semicolon causes subsequent characters to be printed immediately following the previous ones. (Some BASICs allow you the option of omitting the semicolon between items to be printed.) If the end of the current line is reached, the next characters will "wrap around" to the beginning of the next line. At the end of the above PRINT statement, since the last character is not a comma or a semicolon, the output cursor will automatically move to the first position of the next line, to wait for its next printing job.

Last month I mentioned an aspect of input that would be picked up this month under the subject of output. I have seen programs which, when RUN, display nothing but a question mark and cursor on an otherwise blank screen. I don't know about you, but I NEVER type in anything more than my name, rank, and serial number in such cases. For all I know, such a program may be part of a Communist plot to take over my mind — or worse, to take over my computer. So the need arises for some kind of prompting message to be printed on the screen whenever the user is expected to enter something from the keyboard.

Most BASICs (Atari's is an exception) allow you to print such a prompting message in an INPUT statement, without having to use a separate PRINT. You COULD print a prompt in the following way (note the trailing semicolon on the PRINT statement which keeps the cursor from dropping down to the beginning of the next line):

```
20 PRINT "PLEASE TYPE IN
YOUR NAME: ";
25 INPUT N$
```

Or, using the string-printing feature of the INPUT statement, you could substitute:

```
20 INPUT "PLEASE TYPE IN
YOUR NAME: "; N$
```

The message must be between quotation marks, and must be followed by a semicolon (or, in case of Apple's Integer BASIC, a comma).

These two approaches will have exactly the same effect with the S-80, but will produce slightly different prompting messages on the Apple. Using the separate PRINT will result in

```
PLEASE TYPE IN YOUR
NAME: ?*
```

(where the * denotes the flashing cursor), whereas including the quotation marks in the INPUT will result in

```
PLEASE TYPE IN YOUR
NAME: *
```

In the latter case the question mark is suppressed — which is quite handy when the prompting message isn't really a question, as in this example. And, if you don't have a prompt to print but still want to suppress the "?", you can just use double quotation marks, like so:

```
20 INPUT " "; X$
```

The differences among computers start making things even more interesting from here on. Most BASICs have some form of a TAB statement, which functions just as a typewriter tab stop does. The line

```
50 PRINT TAB(5) "COLUMN
1"; TAB(20) "COLUMN 2"
```

will cause the first string to be printed starting in column 5 of the current line, and the second starting in column 20. Or will it? What if the prior lines were

```
30 PRINT
40 PRINT TAB(9) "THIS IS A
CHART";
```

These lines would leave the cursor at column 26; what effect would line 50 have then? Would the result be (a), (b), or (c)? See Figure 1.

Option (a) shows the TAB dropping down to the next line if the output cursor is already to the right of the specified tab location. Option (b) shows it locating the printout on the current line at the specified tab locations, regardless of where the cursor may be. And option (c) shows it printing as closely as possible to the specified tab position, but not being able to backspace to get there. If you're not sure which it will do, check it out on your computer.

Ah, but what if you have an Atari, which doesn't have a TAB function in the first place? Well, then you either include the right number of spaces in your PRINT statement, or you use the POSITION statement. This allows you to position the output cursor anywhere on the whole screen, horizontally and vertically. The line

```
100 POSITION 5, 10
```

will place the next character printed in the fifth column on the tenth line of the screen. Note, however, that the cursor (which always stays on the screen unless you POKE 752,1) will not make its move until the actual PRINT statement is encountered.

This same kind of absolute cursor positioning is available on the Apple and S-80. The Apple breaks up the job into its horizontal and vertical components using HTAB and VTAB. The line

```
100 HTAB 5 : VTAB 10
```

would have the same effect on the Apple as the above POSITION statement has on the Atari. With the Apple, though, HTAB is not limited to the screen width (40 columns); it will accept values up to 255. The effect of numbers greater than 40, as you might expect, is to wrap around to the beginning of the following line as the 41st, then 81st, then 121st, etc., position.

The S-80's PRINT @ statement is similar to Apple's HTAB, except that it always starts its reckoning at the top line of the screen rather than whatever the current line is. This gives the capability, then, of directing printing to any point on the screen using a single number. It accepts numbers in the range 0 through 1023, covering all 1024 printing positions. After using this system for awhile, one gets very good at thinking in multiples of 64, since each new screen line begins with a position number of 0, 64, 128, etc. The equivalent of the above two lines, then, for the S-80 would be

```
100 PRINT @649,;
```

Most commonly the trailing semicolon would be replaced by whatever you want printed, but the statement also can be used exactly as shown to control the starting position of the next PRINT.

(a) THIS IS A CHART

COLUMN 1 COLUMN 2

(b) COLUMN 1 IS A CCOLUMN 2

(c) THIS IS A CHARTCOLUMN 1COLUMN 2

Figure 1

Incidentally, all these cursor-positioning instructions can be used with INPUT statements as well as with PRINTs. That makes it easier to keep the display in order and assure that the user's input is printed exactly where it would look the best — not necessarily right next to the prompting message. You might have occasion to use coding such as this (for the Apple, in this case):

```
200 VTAB 5
210 PRINT "PLEASE TYPE IN THREE NUMBERS"
220 VTAB 10 : HTAB 1 : INPUT " "; A
230 VTAB 10 : HTAB 13 : INPUT " "; B
240 VTAB 10 : HTAB 25 : INPUT " "; C
```

In addition to PRINT and the various tabbing statements, the most valuable output formatting command is the one that clears all displayed text instantaneously and returns the cursor to the upper left corner of the screen. The options for this action on the three computers in question are as follows:

S-80: CLS

Atari: PRINT "esc ctrl-clear" or PRINT CHR$(125)

Apple: HOME (Applesoft) or CALL -936

This command is extremely useful for keeping assorted garbage off the screen, which helps the user to concentrate on what's currently expected of him.

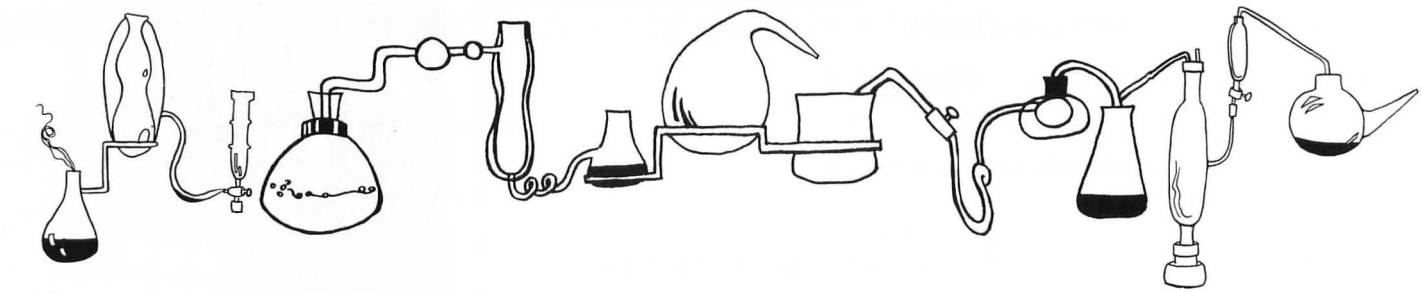
Beyond these basics of text display, each computer has its unique capabilities that allow you to dress things up a bit. These are too varied to cover in detail here, but some of them at least deserve mention.

Available on the S-80 is a powerful formatting statement, PRINT USING, which allows you to define a format for printing numbers (e.g., in dollar-and-cents decimal form). Another useful feature is the double-width character mode, which can catch one's eye for titles and short blocks of text, and for a variety of game boards and displays where larger characters are desirable. (If only they could be mixed with normal characters. . . .) You can also build large text characters or other shapes using the available block graphics characters, assembling them into strings to be printed on the screen. The ability to intermix these with regular text (and with normal SET/RESET graphics) allows lots of variety in display formatting.

Available on the Atari are not only double-width, but double-width-and-height characters as well, all in your choice of colors (GRAPHICS modes 1 and 2). Although these can't intermix with each other or with normal text on the upper 20 lines of the screen, normal text can be printed on the bottom four lines at any time. You can also print a variety of graphics characters such as arrows, pips, and other shapes (not to mention lower-case letters); and all characters can be printed either normally or in inverse video. You also have the capability of setting the left and right margins of the text window, so that one area of the screen can be preserved while text is printed and scrolled in another area. There's no excuse except laziness for a boring text display on the Atari!

Available on the Apple is just one upper-case character set, but you can manipulate it in a variety of ways. Characters can be printed in normal, inverse, or flashing modes, and the speed of the printing can be adjusted from normal (top) speed down to an agonizingly slow pace. All four margins of the text screen can be adjusted, down to a one-character display if you choose. And there's a convenient SPC( ) instruction which, when used in a PRINT, will print the specified number of spaces.

Much more remains to be said about the specifics of output formatting. But the essential point is that it's both worthwhile and fun to put some creative energy into this aspect of programming. Nobody loves a tacky program.



Chemistry Drill

by Brent Packer

Apple and Atari translations by Jon Voskuil.

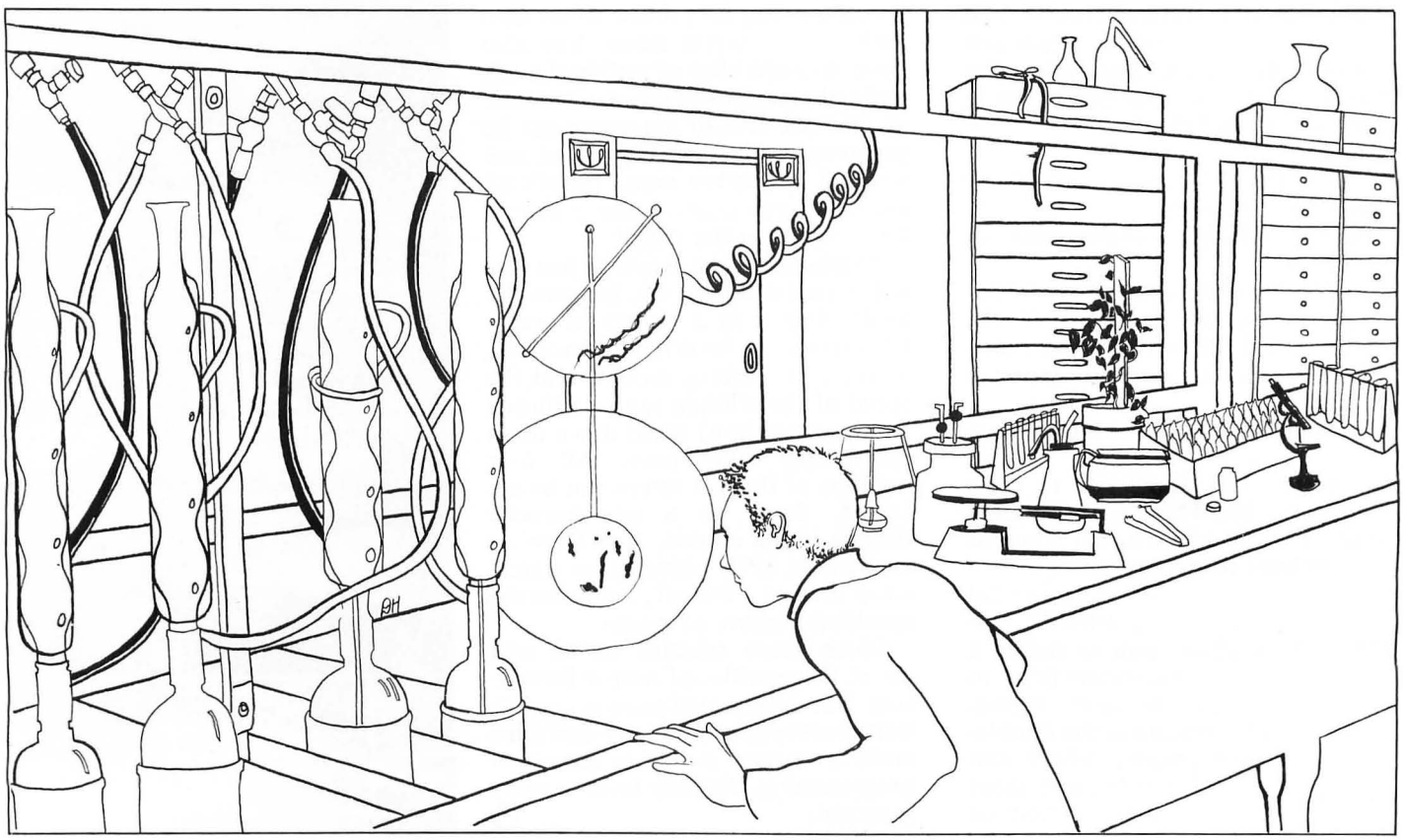
"Chemistry Drill" is a program for the S-80, Atari, and Apple requiring 16K RAM. The Apple version also requires Applesoft.

"Double, double toil and trouble / Fire burn and cauldron bubble!" You literary types may think that's a line from Shakespeare's "Macbeth", but actually it's the refrain from the "Chemistry Lab Theme Song." Probably more people take chemistry because of fantasies about brewing various magical potions, than because of any real interest in the meatier parts of the science.

But then the first day of class arrives, the long-awaited initiation into the wonders of test tubes, Erlenmeyer flasks, Bunsen burners, and pipettes, and what happens? The teacher hands out papers with odd-looking symbols and names on them, and in a threatening voice says "Memorize these or you'll fail this class!"

"Chemistry Drill" was written by a chemistry student who decided to bypass flash cards and other such outdated media, and to program his computer to check his basic chemistry knowledge. The result? A-pluses on every quiz covering elements, radicals, and oxidation states! Whether you're taking chemistry as a subject in school or not, you'll have fun trying to learn or to recall the material included in this program. And if you're up to it, you can add your own DATA statements and revise the coding so that you can include more material, or material on different subjects altogether.

When you RUN the program, you are first asked to type in your name, and then to select from a menu of subject areas. Choosing "Element Names" will require you to enter the correctly-spelled name of the element which matches the symbol that the computer displays. If you misspell it, but get at least the first three letters right, then you'll get two more chances to spell it correctly. The second choice, "Element Symbols," is the reverse of the first, calling for you to enter the correct symbol for the



named element. The next two choices, "Radical Names" and "Radical Symbols," present drills similar to the first pair, but using radicals (the chemical kind) instead of elements. And the last two options quiz you on "Oxidation States" and "Solubility Rule Numbers." When the drill is over, the computer reports your performance and prints out an appropriate comment (not always complimentary).

The program currently has 50 elements, 33 radicals, and 17 oxidation states. To expand the number of elements, change the value assigned to N in lines 200 and 205, and N1 in line 300, to the new number; and add new DATA lines following line 1034. For adding radicals, change N in lines 210 and 215 and add DATA statements following line 1220. Oxidation states can also be augmented by changing N in line 220 and adding DATA following line 1330.

This program won't give you formulas for magic potions, but it might help you learn what you need to know to brew your own.

VARIABLES

A$(*): Array for storing answers.
C: Chosen drill number.
C$: Keyboard input character.
F: Flag governing the reading of DATA as question/answer or answer/question.
F1: Flag controlling the reading of DATA into the Q$ and A$ arrays.
I$: The type of information to be input by the user as the answer.
I, J, K, KI: Loop variables.
N: Number of questions available in category.
N1: Number of data pairs to be read in category.
NAM$: Name of program user.
P: Percentage of correct answers.
P$: The type of information that the computer is printing out as the question.
Q: Number of questions chosen by user.
Q$ (*): Array for storing questions.
R (*): Array for keeping track of question numbers already used.
R1: Random number to select right-answer response.
S$ (*): Responses to wrong answers.
V$: Keyboard input character.
WR: Number of wrong answers.
Z1: Number of tries to spell word correctly.

S-80 VERSION

Initialization.

```
1 CLEAR(500)
2 DIM Q$(100),A$(100)
3 DIM R(100)
10 CLS:INPUT"YOUR NAME: ";NAM $:CLS
50 CLS
```

Display the menu of choices and go to the appropriate place in the program.

```
100 PRINTTAB(18)"C H E M I S T R Y  D R I L L"
110 PRINT
120 PRINT"WHICH WOULD YOU LIKE TO GIVE"
130 PRINTTAB(18)"1. ELEMENT NAMES"
132 PRINTTAB(18)"2. ELEMENT SYMBOLS"
150 PRINTTAB(18)"3. RADICAL NAMES"
160 PRINTTAB(18)"4. RADICAL SYMBOLS"
165 PRINTTAB(18)"5. OXIDATION STATES"
168 PRINTTAB(18)"6. SOLUBILITY RULE NUMBERS"
169 PRINT
170 PRINT"GIVE NUMBER OF CHOICE: ":C$=INKEY$
175 C=VAL(C$):IF C=0 THEN C$=INKEY$:GOTO175
180 ON C GOTO 200,205,210,215,220,230,170,170,170
```

Assign the proper number of data items (N) and the appropriate values for P$ and I$.

```
200 N=50:F=1:P$="SYMBOL":I$="ELEMENT":GOTO300
205 N=50:P$="ELEMENT":I$="SYMBOL":GOTO300
210 N=33:P$="SYMBOL":I$="NAME":GOTO300
215 F=1:N=33:P$="NAME":I$="SYMBOL":GOTO300
220 N=17:F=1:P$="ELEMENT":I$="OXIDATION STATE":GOTO300
230 N=7:P$="RULE ":I$="RULE # ":GOTO300
```

Read the appropriate data for the questions and the correct answers.

```
300 N1=50
302 FOR I=1 TO N1
305 IF F=0 THEN 315
310 READ A$(I),Q$(I)
312 GOTO320
315 READ Q$(I),A$(I)
320 NEXT I
330 IF C>2 AND F1=0 THEN F1=1:N1=33:GOTO302
335 IF C>4 AND F1=1 THEN F1=2:N1=17:GOTO302
336 IF C>5 AND F1=2 THEN F1=3:N1=7:GOTO302
```

Input the number of questions desired.

```
400 CLS:PRINT"HOW MANY QUESTIONS, ";NAM$:INPUT Q
410 IF Q>N THEN PRINT"FEWER":FORKI=1TO100:NEXTKI:GOTO400
```

Choose a random question and check to see that it has not been used already.

```
500 FOR J=1 TO Q
510 R(J)=RND(N)
520 IF J=1 THEN 560
530 FOR K=1 TO J-1
540 IF R(J)=R(K) THEN 510
550 NEXT K
560 NEXT J
```

Responses to wrong answers.

```
600 S$(1)="WRONG.  OK THIS TIME..."+NAM$
602 S$(2)="WRONG. THINK A LITTLE HARDER, "+NAM $
604 S$(3)="WRONG AGAIN.  YOU'RE TRYING MY PATIENCE "+NAM $
606 S$(4)="STUPID! IMBECILE! MORON! IDIOT! YOU'RE WRONG!! "+NAM
$
```

Print the chosen questions and input the answers.

```
610 FOR J=1 TO Q
620 CLS
625 PRINT
630 PRINT" THE "P$" IS: "A$(R(J))
635 PRINT
640 PRINT"NOW, "NAM $", GIVE THE CORRESPONDING "I$" : ";
645 INPUT A$
650 IF A$=Q$(R(J)) THEN 680
655 IF C<>1 AND C<>3 THEN 665
656 IF Z1>2 THEN 662
```

If first three letters are correct, try re-spelling.

```
660 IF LEFT$(A$,3)=LEFT$(Q$(R(J)),3)  THEN PRINT"MISSPELLED..TRY
 AGAIN ":Z1=Z1+1:FORKI=1TO100:NEXTKI:GOTO620
```

Wrong answer; print appropriate response.

```
662 PRINT"FORGET IT...."
```

continued on next page

continued from previous page

```
663 Z1=0
665 WR=WR+1
670 IF WR<2 THEN PRINTS$(1)
672 IF WR>1 AND WR<5 THEN PRINTS$(2)
674 IF WR>4 AND WR<10 THEN PRINTS$(3)
676 IF WR>9 THEN PRINTS$(4)
678 PRINT"RIGHT ANSWER:"Q$(R(J))   :GOTO690
```

Right answer; print encouraging response.

```
680 R1=RND(3):ON R1 GOTO 682,684,686
682 PRINT"GOOD SHOW":GOTO690
684 PRINT"EXCELLENT":GOTO690
686 PRINT"JOLLY GOOD":GOTO690
690 FOR KI=1 TO 800:NEXT KI:Z1=0
695 NEXT J
```

End of drill. Print the results and offer another drill.

```
700 CLS:P=INT(((Q-WR)/Q)*100)
710 PRINTNAM $", YOU DID "Q-WR" CORRECTLY OUT OF "Q
720 PRINT"THAT'S: "P"%"
730 IF P>89 THEN PRINT"FINE JOB--YOU'VE STUDIED ENOUGH"
740 IF P<60 THENPRINT"BY YOUR SCORE, THIS MUST BE JUST YOUR FIRS
T TIME AROUND FOR THIS CLASS"
750 PRINT:PRINT"ANOTHER ROUND (Y/N)":V$=INKEY$:IF V$="N" THEN EN
D
755 IF V$=""THEN V$=INKEY$
756 IF V$="N" THEN END
757 IF V$="Y" THEN 760
758 GOTO755
760 RUN
```

Data for elements.

```
1010 DATA H,HYDROGEN,HE,HELIUM,LI,LITHIUM,BE,BERYLLIUM
1012 DATA B,BORON,C,CARBON,N,NITROGEN,O,OXYGEN
1014 DATA F,FLUORINE,NE,NEON,NA,SODIUM,MG,MAGNESIUM
1016 DATA AL,ALUMINUM,SI,SILICON,P,PHOSPHORUS,S,SULFUR
1018 DATA CL,CHLORINE,AR,ARGON,K,POTASSIUM,CA,CALCIUM
1020 DATA SC,SCANDIUM,TI,TITANIUM,V,VANADIUM,CR,CHROMIUM
1022 DATA MN,MANGANESE,FE,IRON,CO,COBALT,NI,NICKEL
1024 DATA CU,COPPER,ZN,ZINC,AS,ARSENIC,BR,BROMINE
1026 DATA KR,KRYPTON,RB,RUBIDIUM,SR,STRONTIUM,MO,MOLYBDENUM
1028 DATA AG,SILVER,CD,CADMIUM,SN,TIN,SB,ANTIMONY
1030 DATA I,IODINE,CS,CESIUM,BA,BARIUM,W,TUNGSTEN
1032 DATA PT,PLATINUM,AU,GOLD,HG,MERCURY,PB,LEAD
1034 DATA BI,BISMUTH,U,URANIUM
```

Data for radicals.

```
1100 DATA ACETATE,C2H3O2-1,AMMONIUM,NH4+1
1101 DATA BICARBONATE,HCO3-1
1110 DATA BISULFATE,HSO4-1
1115 DATA BISULFITE,HSO3-1
1120 DATA HYPOCHLORITE,CLO-1
1130 DATA CHLORITE,CLO2-1,CHLORATE,CLO3-1,PERCHLORATE,CLO4-1
1140 DATA PERMANGANATE,MNO4-1,NITRATE,NO3-1,HYDROXIDE,OH-1
1150 DATA NITRITE,NO2-1,CYANIDE,CN-1,CARBONATE,CO3-2
1160 DATA SULFITE,SO3-2,SULFATE,SO4-2,MANGANATE,MNO4-2
1170 DATA OXALATE,C2O4-2,CHROMATE,CRO4-2,DICHROMATE,CR2O7-2
1180 DATA PHOSPHATE,PO4-3,ARSENATE,ASO4-3
1190 DATA CUPROUS,CU+1,CUPRIC,CU+2
1200 DATA FERROUS,FE+2,FERRIC,FE+3,STANNOUS,SN+2
1210 DATA STANNIC,SN+4,PLUMBOUS,PB+2,PLUMBIC,PB+4
1220 DATA MERCUROUS,HG2+2,MERCURIC,HG+2
```

Data for oxidation states.

```
1300 DATA BE,+2,LI,+1,ZN,+2,AG,+1
1310 DATA H,+1,NA,+1,K,+1
1320 DATA MG,+2,CA,+2,BA,+2
1330 DATA AL,+3,O,-2,S,-2,BR,-1,I,-1,CL,-1,F,-1
```

Data for solubility rules.

```
1400 DATA  1,"ALL NA, NH4+1, AND K COMPOUNDS ARE SOLUBLE"
1404 DATA  2,"ALL NITRATES, ACETATES, AND CHLORATES ARE SOLUBLE"
1405 DATA  3,"ALL HALIDES ARE SOLUBLE EXCEPT: AG,HG,PB"
1406 DATA  4,"ALL SULFATES ARE SOLUBLE EXCEPT: BA,CA,SR,RA,PB"
1408 DATA  5,"ALL OXIDES ARE INSOLUBLE EXCEPT: RULE 1 AND  GROUP
 II (BE,MG,SR,BA,RA,CA) WHICH WILL REACT"
1410 DATA  6,"ALL HYDROXIDES AND SULFIDES ARE INSOLUBLE EXCEPT R
ULE 1 AND GROUP II (2A ON TABLE)"
1412 DATA  7,"EVERYTHING NOT ABOVE MENTIONED IS INSOLUBLE"
```

APPLE VERSION

```
2  DIM Q$(100),A$(100),R(100)
10  HOME : VTAB 10: INPUT "YOUR N
    AME: ";NAM$
50  HOME : VTAB 3
```

Display the menu of choices and go to the appropriate place in the program.

```
100  PRINT  TAB( 6)"C H E M I S T
     R Y   D R I L L"
120  VTAB 8: PRINT "WHICH WOULD Y
     OU LIKE TO GIVE:": PRINT
130  PRINT "     1. ELEMENT NAMES"
140  PRINT "     2. ELEMENT SYMBOL
     S"
150  PRINT "     3. RADICAL NAMES"
160  PRINT "     4. RADICAL SYMBOL
     S"
165  PRINT "     5. OXIDATION STAT
     ES"
168  PRINT "     6. SOLUBILITY RUL
     E NUMBERS"
170  PRINT : PRINT "GIVE NUMBER O
     F CHOICE: ";: INPUT C$
175 C =  VAL (C$): IF C < 1 OR C >
     6 THEN 170
180  ON C GOTO 200,205,210,215,22
     0,230
```

Assign the proper number of data items (N) and the appropriate values for P$ and I$.

```
200 N = 50:F = 1:P$ = "SYMBOL":I$
     = "ELEMENT": GOTO 300
```

```
205 N = 50:P$ = "ELEMENT":I$ = "S
    YMBOL": GOTO 300
210 N = 33:P$ = "SYMBOL":I$ = "NA
    ME": GOTO 300
215 F = 1:N = 33:P$ = "NAME":I$ =
    "SYMBOL": GOTO 300
220 N = 17:F = 1:P$ = "ELEMENT":I
    $ = "OXIDATION STATE": GOTO
    300
230 N = 7:P$ = "RULE ":I$ = "RULE
     # "
```

Read the appropriate data for the questions and the correct answers.

```
300 N1 = 50
302  FOR I = 1 TO N1
305  IF F = 0 THEN 315
310  READ A$(I),Q$(I)
312  GOTO 320
315  READ Q$(I),A$(I)
320  NEXT I
330  IF C > 2 AND F1 = 0 THEN F1 =
     1:N1 = 33: GOTO 302
335  IF C > 4 AND F1 = 1 THEN F1 =
     2:N1 = 17: GOTO 302
336  IF C > 5 AND F1 = 2 THEN F1 =
     3:N1 = 7: GOTO 302
```

Input the number of questions desired.

```
400  HOME : VTAB 5: PRINT "HOW MA
     NY QUESTIONS, ";NAM$: PRINT
     "(1 - ";N;"): ";: INPUT Q: HOME

410  IF Q > N THEN 400
```

Choose random questions and check each one to see that it is not duplicated.

```
500  FOR J = 1 TO Q
510 R(J) =  INT ( RND (1) * N) +
     1
520  IF J = 1 THEN 560
530  FOR K = 1 TO J - 1
540  IF R(J) = R(K) THEN 510
550  NEXT K
560  NEXT J
```

Responses to wrong answers.

```
600 S$(1) = "OOPS, WRONG.  THAT'S
      OK, " + NAM$
602 S$(2) = "WRONG; THINK A LITTL
     E HARDER, " + NAM$
604 S$(3) = "WRONG AGAIN.  YOU'RE
      TRYING MY PATIENCE," + NAM$

606 S$(4) = "STUPID!  IMBECILE!
     MORON!  IDIOT!      YOU'RE W
     RONG, " + NAM$
```

Print the chosen questions and input the answers.

```
610  FOR J = 1 TO Q
620  HOME
630  PRINT : PRINT "THE ";P$;" IS
     :  ";: IF C = 6 THEN  PRINT
```

```
635  PRINT A$(R(J))
640  PRINT : PRINT "NOW, ";NAM$;"
     , GIVE THE": PRINT "CORRESPO
     NDING ";I$;":  ";
645  INPUT A$: PRINT
650  IF A$ = Q$(R(J)) THEN 680
655  IF C <  > 1 AND C <  > 3 THEN
     665
656  IF Z1 > 2 THEN 662
```

If first three characters are correct, try re-spelling.

```
660  IF  LEFT$ (A$,3) =  LEFT$ (Q
     $(R(J)),3) THEN  PRINT "MISS
     PELLED...TRY AGAIN":Z1 = Z1 +
     1: GOTO 645
```

Wrong answer; print appropriate response.

```
662  PRINT "FORGET IT. . ."
663 Z1 = 0
665 WR = WR + 1
670  IF WR < 3 THEN  PRINT S$(1)
672  IF WR > 2 AND WR < 6 THEN  PRINT
     S$(2)
674  IF WR > 5 AND WR < 10 THEN  PRINT
     S$(3)
676  IF WR > 9 THEN  PRINT S$(4)
678  FOR KI = 1 TO 500: NEXT KI: PRINT
     : PRINT "RIGHT ANSWER:  ";Q$
     (R(J)): FOR KI = 1 TO 1000: NEXT
     KI: GOTO 690
```

Right answer; print encouraging response.

```
680 R1 =  INT ( RND (1) * 3) + 1
681  PRINT : ON R1 GOTO 682,684,6
     86
682  PRINT "GOOD SHOW": GOTO 690
684  PRINT "EXCELLENT": GOTO 690
686  PRINT "JOLLY GOOD"
690  FOR KI = 1 TO 1000: NEXT KI
695  NEXT J
```

End of drill. Print the results and offer another drill.

```
700  HOME :P =  INT (((Q - WR) /
     Q) * 100)
710  VTAB 5: PRINT NAM$;", YOU DI
     D ";Q - WR;" CORRECTLY": PRINT
     "OUT OF ";Q
720  PRINT : PRINT "THAT'S ";P;"%
      RIGHT."
730  IF P > 89 THEN  PRINT "FINE
     JOB--YOU'VE STUDIED ENOUGH."

740  IF P < 60 THEN  PRINT "BY YO
     UR SCORE, YOU MUST BE FAILIN
     G..."
750  PRINT : PRINT "ANOTHER ROUND
     ? ";: GET V$: IF V$ = "N" THEN
      END
760  RUN
```

Data for elements.

```
1010  DATA  H,HYDROGEN,HE,HELIUM,
     LI,LITHIUM,BE,BERYLLIUM,B,BO
     RON,C,CARBON,N,NITROGEN,O,OX
     YGEN
1014  DATA  F,FLUORINE,NE,NEON,NA
     ,SODIUM,MG,MAGNESIUM,AL,ALUM
     INUM,SI,SILICON,P,PHOSPHORUS
     ,S,SULFUR
1018  DATA  CL,CHLORINE,AR,ARGON,
     K,POTASSIUM,CA,CALCIUM,SC,SC
     ANDIUM,TI,TITANIUM,V,VANADIU
     M,CR,CHROMIUM
```

continued on next page

continued from previous page

```
1022 DATA MN,MANGANESE,FE,IRON,
     CO,COBALT,NI,NICKEL,CU,COPPE
     R,ZN,ZINC,AS,ARSENIC,BR,BROM
     INE
1026 DATA KR,KRYPTON,RB,RUBIDIU
     M,SR,STRONTIUM,MO,MOLYBDENUM
     ,AG,SILVER,CD,CADMIUM,SN,TIN
     ,SB,ANTIMONY
1030 DATA I,IODINE,CS,CESIUM,B
     A,BARIUM,W,TUNGSTEN,PT,PLATI
     NUM,AU,GOLD,HG,MERCURY,PB,LE
     AD,BI,BISMUTH,U,URANIUM


 Data for radicals.


1100 DATA ACETATE,C2H3O2-1,AMMO
     NIUM,NH4+1,BICARBONATE,HCO3-
     1,BISULFATE,HSO4-1,BISULFITE
     ,HSO3-1
1120 DATA HYPOCHLORITE,CLO-1,CH
     LORITE,CLO2-1,CHLORATE,CLO3-
     1,PERCHLORATE,CLO4-1,PERMANG
     ANATE,MNO4-1,NITRATE,NO3-1
1150 DATA HYDROXIDE,OH-1,NITRI
     TE,NO2-1,CYANIDE,CN-1,CARBON
     ATE,CO3-2,SULFITE,SO3-2,SULF
     ATE,SO4-2,MANGANATE,MNO4-2
1170 DATA OXALATE,C2O4-2,CHROMA
     TE,CRO4-2,DICHROMATE,CR2O7-2
     ,PHOSPHATE,PO4-3,ARSENATE,AS
     O4-3,CUPROUS,CU+1,CUPRIC,CU+
     2
1200 DATA FERROUS,FE+2,FERRIC,
     FE+3,STANNOUS,SN+2,STANNIC,S
     N+4,PLUMBOUS,PB+2,PLUMBIC,PB
     +4,MERCUROUS,HG2+2,MERCURIC,
     HG+2


 Data for oxidation states.


1300 DATA BE,+2,LI,+1,ZN,+2,AG
     ,+1,H,+1,NA,+1,K,+1,MG,+2,CA
     ,+2,BA,+2,AL,+3,O,-2,S,-2,BR
     ,-1,I,-1,CL,-1,F,-1


 Data for solubility rules.


1400 DATA 1,"ALL NA, NH4+1, AND
      K COMPOUNDS ARE      SOLUBL
     E"
1404 DATA 2,"ALL NITRATES, ACET
     ATES, AND CHLORATES   ARE SO
     LUBLE"
1405 DATA 3,"ALL HALIDES ARE SO
     LUBLE EXCEPT:         AG, HG
     , PB"
1406 DATA 4,"ALL SULFATES ARE S
     OLUBLE EXCEPT:        BA, CA
     , SR, RA, PB"
1408 DATA 5,"ALL OXIDES ARE IN
     SOLUBLE EXCEPT:  RULE 1AND G
     ROUP II (BE, MG, SR, BA, RA,
      CA)  WHICH WILL REACT"
1410 DATA 6,"ALL HYDROXIDES AND
      SULFIDES ARE         INSOLU
     BLE EXCEPT RULE 1 AND GROUP
     II    (2A ON TABLE)"
1412 DATA 7,"EVERYTHING NOT ABO
     VE MENTIONED IS       INSOLU
     BLE"
```

ATARI VERSION

Additional Variables

AA$: Answer typed in by user.
CL$: CHR$ (125); clears screen when printed.
SKIP: Number of data to skip over in order to get to the type desired.
X$: Temporary string variable.

Initialization.

```
1 POKE 752,1
2 DIM P$(20),I$(20),AA$(20),CL$(1)
3 DIM A$(100),Q$(100),X$(20),NAM$(20),
R(50)
5 CL$=CHR$(125):Z1=0
10 PRINT CL$:POSITION 2,10:PRINT "YOUR
 NAME: ";:INPUT NAM$:PRINT CL$
```

Display the menu of choices and go to the appropriate place in the program.

```
100 PRINT :PRINT :PRINT "    C H E M I
 S T R Y   D R I L L"
110 PRINT
120 PRINT "WHICH WOULD YOU LIKE TO GIV
E:":PRINT
130 PRINT "     1. ELEMENT NAMES"
140 PRINT "     2. ELEMENT SYMBOLS"
150 PRINT "     3. RADICAL NAMES"
160 PRINT "     4. RADICAL SYMBOLS"
165 PRINT "     5. OXIDATION STATES"
168 PRINT "     6. SOLUBILITY RULE NUM
BERS"
169 PRINT
170 PRINT "GIVE NUMBER OF CHOICE: ";
175 INPUT X$:C=VAL(X$):IF C<0 OR C>6 T
HEN 170
180 PRINT CL$:ON C GOTO 200,205,210,21
5,220,230
```

Assign appropriate number of data to skip, and from which to read, and assign values for P$ and I$.

```
200 SKIP=0:N=50:F=1:P$="SYMBOL":I$="EL
EMENT":GOTO 400
205 SKIP=0:N=50:F=0:P$="ELEMENT":I$="S
YMBOL":GOTO 400
210 SKIP=50:N=33:F=0:P$="SYMBOL":I$="N
AME":GOTO 400
215 SKIP=50:N=33:F=1:P$="NAME":I$="SYM
BOL":GOTO 400
220 SKIP=83:N=17:F=1:P$="ELEMENT":I$="
OXIDATION STATE":GOTO 400
230 SKIP=100:N=7:F=0:P$="RULE ":I$="RU
LE # "
```

Input the number of questions desired.

```
400 POSITION 2,10:PRINT "HOW MANY QUES
TIONS, ";NAM$:PRINT "(1 TO ";N;"):  ";
410 INPUT Q:IF Q<1 OR Q>N THEN 400
420 PRINT CL$
```

Choose random questions, checking to see that they are not repeated.

```
500 FOR J=1 TO Q
510 R(J)=INT(RND(1)*N+1)
520 IF J=1 THEN 560
530 FOR K=1 TO J-1
540 IF R(J)=R(K) THEN 510
550 NEXT K
560 NEXT J
```

Skip over data to chosen question/answer pairs, present questions and input answers.

```
610 FOR J=1 TO Q
615 RESTORE
620 FOR I=1 TO SKIP+R(J):READ Q$,A$:NE
XT I
625 IF F=1 THEN X$=Q$:Q$=A$:A$=X$
630 PRINT CL$:PRINT :PRINT "THE ";P$;"
 IS: ";:IF C=6 THEN PRINT
632 PRINT A$:PRINT :PRINT
640 PRINT "NOW, ";NAM$;", GIVE THE":PR
INT "CORRESPONDING ";I$;": ";
645 INPUT AA$:PRINT :PRINT
650 IF AA$=Q$ THEN 680
655 IF C<>1 AND C<>3 THEN 665
656 IF Z1>2 THEN 662
```

If first three characters are correct, try re-spelling.

```
660 IF AA$(1,3)=Q$(1,3) THEN PRINT "MI
SSPELLED...TRY AGAIN":Z1=Z1+1:GOTO 645
```

```
Wrong answer; print appropriate
response and correct answer.

662 PRINT "FORGET IT. . . ."
 663 Z1=0
665 WR=WR+1
670 IF WR<2 THEN PRINT "WRONG.  OK THI
S TIME, ";NAM$
672 IF WR>1 AND WR<5 THEN PRINT "WRONG
--THINK A LITTLE HARDER, ";NAM$
674 IF WR>4 AND WR<10 THEN PRINT "WRON
G AGAIN.  YOU'RE TRYING MY":PRINT "PAT
IENCE, ";NAM$
676 IF WR>9 THEN PRINT "STUPID!  IMBEC
ILE!  MORON!  IDIOT!":PRINT "YOU'RE WR
ONG, ";NAM$
678 PRINT :PRINT "RIGHT ANSWER: ";Q$:G
OTO 690

Right answer; print encouraging
response.

680 PRINT :R1=INT(RND(1)*3+1):ON R1 GO
TO 682,684,686
682 PRINT "GOOD SHOW":GOTO 690
684 PRINT "EXCELLENT":GOTO 690
686 PRINT "JOLLY GOOD"
690 FOR KI=1 TO 500:NEXT KI:Z1=0
695 NEXT J

End of drill.  Print the results and
offer another drill.

700 PRINT CL$:P=INT(((Q-WR)/Q)*100)
710 PRINT :PRINT NAM$;", YOU DID ";Q-W
R;" CORRECTLY":PRINT "OUT OF ";Q
720 PRINT :PRINT "THAT'S ";P;" PERCENT
 RIGHT":PRINT :PRINT
730 IF P>89 THEN PRINT "FINE JOB--YOU'
VE STUDIED ENOUGH"
740 IF P<60 THEN PRINT "BY YOUR SCORE,
 THIS MUST BE YOUR":PRINT "FIRST TIME
AROUND FOR THIS CLASS."
750 PRINT :PRINT :PRINT "ANOTHER ROUND
? (Y/N) ";:INPUT X$:IF X$(1,1)="N" THE
N END
760 RUN

Data for elements.

1010 DATA H,HYDROGEN,HE,HELIUM,LI,LITH
IUM,BE,BERYLLIUM,B,BORON,C,CARBON,N,NI
TROGEN,O,OXYGEN
1014 DATA F,FLUORINE,NE,NEON,NA,SODIUM
,MG,MAGNESIUM,AL,ALUMINUM,SI,SILICON,P
,PHOSPHORUS,S,SULFUR
1018 DATA CL,CHLORINE,AR,ARGON,K,POTAS
SIUM,CA,CALCIUM,SC,SCANDIUM,TI,TITANIU
M,V,VANADIUM,CR,CHROMIUM
1022 DATA MN,MANGANESE,FE,IRON,CO,COBA
LT,NI,NICKEL,CU,COPPER,ZN,ZINC,AS,ARSE
NIC,BR,BROMINE
1026 DATA KR,KRYPTON,RB,RUBIDIUM,SR,ST
RONTIUM,MO,MOLYBDENUM,AG,SILVER,CD,CAD
MIUM,SN,TIN,SB,ANTIMONY
1030 DATA I,IODINE,CS,CESIUM,BA,BARIUM
,W,TUNGSTEN,PT,PLATINUM,AU,GOLD,HG,MER
CURY,PB,LEAD,BI,BISMUTH,U,URANIUM

Data for radicals.

1100 DATA ACETATE,C2H3O2-1,AMMONIUM,NH
4+1,BICARBONATE,HCO3-1,BISULFATE,HSO4-
1,BISULFITE,HSO3-1
1120 DATA HYPOCHLORITE,CLO-1,CHLORITE,
CLO2-1,CHLORATE,CLO3-1,PERCHLORATE,CLO
4-1
1140 DATA PERMANGANATE,MNO4-1,NITRATE,
NO3-1,HYDROXIDE,OH-1,NITRITE,NO2-1,CYA
NIDE,CN-1,CARBONATE,CO3-2
1160 DATA SULFITE,SO3-2,SULFATE,SO4-2,
MANGANATE,MNO4-2,OXALATE,C2O4-2,CHROMA
TE,CRO4-2,DICHROMATE,CR2O7-2
1180 DATA PHOSPHATE,PO4-3,ARSENATE,ASO
4-3,CUPROUS,CU+1,CUPRIC,CU+2,FERROUS,F
E+2,FERRIC,FE+3,STANNOUS,SN+2
1210 DATA STANNIC,SN+4,PLUMBOUS,PB+2,P
LUMBIC,PB+4,MERCUROUS,HG2+2,MERCURIC,H
G+2

Data for oxidation states.

1300 DATA BE,+2,LI,+1,ZN,+2,AG,+1,H,+1
,NA,+1,K,+1
1320 DATA MG,+2,CA,+2,BA,+2,AL,+3,O,-2
,S,-2,BR,-1,I,-1,CL,-1,F,-1

Data for solubility rules.

1400 DATA 1,ALL NA  NH4+1  AND K COMPO
UNDS ARE    SOLUBLE
1402 DATA 2,ALL NITRATES  ACETATES  AN
D CHLORATES ARE SOLUBLE
1404 DATA 3,ALL HALIDES ARE SOLUBLE EX
CEPT        AG  HG  PB
1406 DATA 4,ALL SULFATES ARE SOLUBLE E
XCEPT       BA  CA  SR  RA  PB
1408 DATA 5,ALL OXIDES ARE INSOLUBLE E
XCEPT RULE IAND GROUP II (BE  MG  SR
BA  RA  CA) WHICH WILL REACT
1410 DATA 6,ALL HYDROXIDES AND SULFIDE
S ARE       INSOLUBLE EXCEPT RULE I AN
D GROUP II  (2A ON TABLE)
1412 DATA 7,EVERYTHING NOT MENTIONED A
BOVE IS     INSOLUBLE
```

K-Byters

ANOTHER PROGRAMMING CHALLENGE

Last summer **SoftSide** began inviting its readers to submit "One Liners" — self-contained, single-line programs for the S-80, Apple, or Atari which would provide a continuously changing graphics display. The response has been excellent, and we're still looking for more submissions.

Now we have a new challenge for you as well: "K-Byters". A K-Byter is a BASIC program which fits into 1K (1024) bytes of program memory. There aren't any restrictions on the nature of the program, other than its size. It can be a graphics display, a game, a mini-adventure, or anything your imagination and programming skills can create.

Note that the program does not have to RUN in 1K of memory; it can use as much RAM for arrays, strings, graphics mapping, etc., as you need. We'd prefer that it be able to run in a 16K system, but this is not an absolute limit.

Here, then, are the official rules:

1. The program must be written for the Apple, S-80, or Atari, entirely in BASIC (although it may create and call Machine Language routines).

2. The program must occupy no more than 1024 bytes of memory before running.

3. The program must be submitted on tape or disk, accompanied by your name, address, phone number, and a brief written description of its operation.

4. The tape or disk will be returned only if accompanied by a self-addressed envelope with adequate postage AFFIXED (do not send money).

5. Winners will have their programs published in **SoftSide** and will receive certificates extolling their virtues as programming wizards, for all the world to see!

Send submissions to:

K-Byters, c/o **SoftSide**
6 South Street
Milford, NH 03055

Code Master

by Andrew Braunstein
Atari translation by Rich Bouchard

"Code Master" is an S-80 and Atari game program requiring 16K RAM.

So, you've always been lucky and have easily been able to win at computer games? Well, in "Code Master" your luck just ran out. This is a game of skill and deduction, which is only for the sharp of mind and the courageous of heart. If you can't take losing a game, then maybe you'd better not take the trouble to type this one in, because you may not feel the thrill of victory too often.

Your object in the game is to guess a randomly-generated five character string in ten guesses or less. After each guess, information as to how many characters are correct and how many characters are in the right sequence is displayed. The difference between "Code Master" and your average computer game, is that to win, you'll have to use more logic than luck.

VARIABLES

C$(1) - C$(5): Player's guess for each of the five letters.
D$: Possible letter choices.
Q$, L$(1), L$(2): Correct code.
PL: Current location of cursor.
RL: Number of right letters.
RS: Number of letters in the right sequence.
T: Turn number.
L1, P: Which of the five letters is being guessed.
V1: Flag variable (is player currently changing guess 1-yes, 0-no).
AN$: Player's last guess.
F8: Flag for whether player wants instructions.
V$: Used to print the "CODE MASTER" string.
Y: Screen print position variable.
Z: Variable for response to new game question.
Z1$: Player's most recent guess.
X, R, P, F, I, A, N, M:

Miscellaneous loops and time delays.

S-80 VERSION

```
Print title page and print instructions if requested.

1 CLS:PRINT CHR$(23):PRINT:PRINT:PRINT TAB(5)"C O D E   M A S T
E R":PRINT TAB(9)"VERSION 1.34":PRINT"(C) 1981 BY ANDREW S. BRAU
NSTEIN":FOR Z=1 TO1000:NEXTZ
2 V$="<<<<<<<<<< CODE MASTER >>>>>>>>>>":PRINT:PRINT:PRINT"INSTRU
CTIONS? (Y/N)";
3 E$=INKEY$:IFE$=""THEN3ELSEIFE$<>"Y"THENF8=1:GOTO7
4 CLS:PRINTV$;V$:PRINTTAB(25)"INSTRUCTIONS":PRINT"     THIS GAME
S TESTS YOUR SKILLS OF DEDUCTION.  YOU HAVE TEN":PRINT"CHANCES T
O GUESS THE FIVE LETTER CODE GENERATED BY THE COMPUTER.";:PRINT"
YOU ARE TO ENTER THE VARIOUS LETTERS THAT ARE CONTAINED";
5 PRINT" IN THE  WORD (CHOICES: A,B,C,D,E,F,G,H). THE COMPUTER W
ILL TELL YOU HOW":PRINT"MANY LETTERS ARE RIGHT (RL) AND HOW MANY
 ARE IN THE RIGHT":PRINT"SEQUENCE (RS). IF YOU MAKE A MISTAKE TY
PING IN THE WORD, YOU":PRINT"WILL BE ABLE TO FIX IT AFTER YOU";

6 PRINT" ENTER ALL FIVE LETTERS.":PRINT:PRINTTAB(17)"PRESS <ENT
ER> TO START GAME":PRINT@896,V$;V$;

Print the opening display.

7 J$=INKEY$:IFJ$=""ANDF8<>1THEN7ELSECLS:FORX=1TO5:PRINTCHR$(23);
V$;:FORR=1TO150:NEXTR,X:F8=0
8 PRINT:PRINTTAB(10)"RS = RIGHT SEQUENCE":FORR=1TO175:NEXT
9 PRINT:PRINTTAB(10)"RL = RIGHT LETTER":PRINT:FORR=1TO175:NEXT
10 FORX=1TO5:PRINTV$;:FORR=1TO150:NEXTR,X:FORX=1TO750:NEXT:CLEAR
100:DIMC$(10),L$(10):CLS

Initialize the variables and construct a random code word.

11 D$="ABCDEFGH"
12 FORI=1TO5:Y=RND(8):Q$=Q$+MID$(D$,Y,1):NEXTI:L$(1)=Q$:L$(2)=Q$
:GOTO14
Subroutine to poke in graphics lines on the screen.

13 FORF=YTOY+63:POKEF,X:NEXT:RETURN

Draw the game display.

14 Y=15360:X=131:GOSUB13:FORY=15360TO16128STEP64:POKEY,191:POKEY
+45,191:POKEY+63,191:NEXT:Y=16192:X=131:GOSUB13:FORX=15489TO1553
2:POKEX,131:NEXT:FORX=16320TO16383:POKEX,179:NEXT:FORY=15513TO16
153STEP64:POKEY,191:POKEY+9,191:NEXT
15 PRINT@113,"C";:PRINT@179,"O";:PRINT@245,"D";:PRINT@311,"E";:P
RINT@369,"M";:PRINT@435,"A";:PRINT@501,"S";:PRINT@567,"T";:PRINT
@633,"E";:PRINT@699,"R";
16 PRINT@752,"CHOICES:";:PRINT@818,D$;
17 PRINT@67,"1  2  3  4  5          RL  RS";:PL=195:T=1

Get the player's input.

18 PRINT@PL,"TRY #";MID$(STR$(T),2);:Q$=L$(2):RS=0:RL=0:FORP=1TO
5:C$(P)=" ":PRINT@896,"LETTER IN POSITION";P;"?    ";
19 M$=" ":M$=INKEY$:IFM$=""ORM$=" "THEN19ELSEIFASC(M$)=13THEN19E
LSEC$(P)=M$:PRINT@924,M$;:PRINT@PL,C$(P);:PRINT@PL+1," TRY #";M
ID$(STR$(T),2);:PL=PL+3:NEXTP:PRINT@PL-2,"          ";:PRINT@PL+1
7," TRY #";MID$(STR$(T),2);:PL=PL+49
20 GOSUB38:IFV1=1THEN40ELSEV1=0

Check to see if player has guessed any letters in the proper
sequence.

21 AN$="":FORL1=1TO5:AN$=AN$+C$(L1):IFMID$(Q$,L1,1)=C$(L1)THEN22
ELSEGOTO24
22 Q$=LEFT$(Q$,L1-1)+"#"+MID$(Q$,L1+1):RS=RS+1:RL=RL+1
23 C$(L1)=" "
24 NEXTL1:GOTO30

Player has won!

25 V$="<<<<<<<<<< CODE MASTER >>>>>>>>>>":FORX=1TO1500:NEXT:CLS:P
RINTCHR$(23);V$;V$;V$;V$
26 PRINT@320,"TRIES";:PRINT@366,"SEQUENCE";
27 PRINT@384,T;:PRINT@432,L$(1);:PRINT@576,V$;V$;V$;V$;
28 FORA=1TO10:PRINT@472,"       ";:FORX=1TO275:NEXT:PRINT@472,"Y
OU WON";:FORX=1TO275:NEXTX,A
29 PRINT@896,"PRESS ENTER TO CONTINUE";:INPUTZ:F8=1:GOTO7
```

Check to see if player has guessed any of the right letters.

```
30 FORN=1TO5:FORM=1TO5:IFMID$(Q$,N,1)=C$(M)THEN37
31 NEXTM,N:PRINT@PL-40,RL;RS;:IFRS=5THEN25
32 IFT=10THEN33ELSET=T+1:GOTO18
```

Player has lost.

```
33 FORQQ=1TO1500:NEXT
34 V$="<<<<<<<<<< CLUE MASTER >>>>>>>>>>":CLS:PRINTCHR$(23);V$;V$
;V$;V$:PRINT@320,"LAST GUESS";:PRINT@364,"=SEQUENCE=";:PRINT@386
,AN$;:PRINT@432,L$(1);
35 PRINT@576,V$;V$;V$;V$;:FORX=1TO10:PRINT@472,"        ";:FORT=
1TO100:NEXT:PRINT@472,"YOU LOST";:FORT=1TO100:NEXTT,X
36 PRINT@896,"PRESS ENTER TO CONTINUE";:INPUTZ:F8=1:GOTO7
```

Update the number of correct letters.

```
37 Q$=LEFT$(Q$,N-1)+"#"+MID$(Q$,N+1):C$(M)=" ":RL=RL+1:GOTO31
```

Subroutine for processing a change in the player's input.

```
38 Z1$="":PRINT@896,"DO YOU WANT TO CHANGE THEM (Y/N)?";:GOSUB41
:IFZ1$="Y"THEN39ELSERETURN
39 PRINT@896,"TYPE IN AS USUAL";:V1=1:FORX=1TO500:NEXTX:PL=PL-64
:PRINT@PL,STRING$(21,32);:PRINT@PL+24,STRING$(6,32);:PRINT@PL+32
,STRING$(9,32);:GOTO18
40 V1=0:RETURN
```

Subroutine to get a character from the keyboard.

```
41 Z1$=INKEY$:IFZ1$=""THEN41ELSEPRINT@896,STRING$(63,32);:RETURN
```

ATARI VERSION
Additional Variables

A$: A control down arrow character used to position cursor.
C$: Holds the player's guess for each of the five letters.
L$ (1,5), L$ (6,10), Q$: Correct code.
M, V: Loop variables for sound routines.
ME$: Message to be displayed at game's end.
PL: Turn number, also used to position cursor by offsetting PL lines from the start of the board display.

Underlined characters in the Atari listing should be typed in inverse video by using the Atari logo key.

Dimension string variables, and prepare keyboard for use as an input device to allow single characters to be input from the keyboard.

```
3 DIM V$(40),D$(8),Q$(5),L$(10),A$(1),
C$(5),AN$(5),ME$(10)
5 OPEN #1,4,0,"K:"
```

Create beginning display. Line 10 specifies graphics mode 2 without a text window. Different types of letters are used in lines 20-50 (lowercase, inverse viedo) to produce different colors on the graphic screen.

```
10 GRAPHICS 2+16
12 V$="<<<<<<< C O D E   M A S T E R
>>>>>>>"
13 A$=CHR$(29)
15 SETCOLOR 4,1,14
20 PRINT #6:PRINT #6;"    CODE MASTER
"
30 PRINT #6:PRINT #6;"original program
 by:";
35 PRINT #6;"ANDREW S. BRAUNSTEIN"
40 PRINT #6;"  atari version by:"
50 PRINT #6;"    RICH BOUCHARD"
60 FOR M=0 TO 127
64 SOUND 0,M,10,10:SOUND 1,M*2,10,10:S
OUND 2,255-M,10,10:SOUND 3,255-(M*2),1
0,10
68 NEXT M
70 PRINT #6:PRINT #6;" INSTRUCTIONS (Y
/N)";
```

Get the player's response to INSTRUCTIONS (Y/N) question, shut off the sound, and then determine whether instructions are desired.

```
80 GET #1,E
85 GOSUB 1100
90 IF CHR$(E)<>"Y" THEN F8=1:GOTO 280
```

Print out the full instructions. In line 100, the POKE 752,1 is used to shut off the cursor.

```
100 GRAPHICS 0:POKE 752,1
110 PRINT :PRINT :PRINT V$;V$
120 PRINT :PRINT "          INSTRUCT
IONS":PRINT
130 PRINT "   THIS GAME TESTS YOUR
  SKILL OF";
140 PRINT "DEDUCTION.  YOU HAVE TEN
 CHANCES TO";
150 PRINT "GUESS THE FIVE LETTER COD
E GENERATED";
160 PRINT "BY THE COMPUTER.  YOU AR
E TO ENTER";
170 PRINT "THE VARIOUS LETTERS THAT AR
E CONTAINED";
180 PRINT "IN THE WORD (CHOICES A,B,C,
D,E,F,G,H).";
190 PRINT "THE COMPUTER WILL TELL YO
U HOW MANY";
200 PRINT "LETTERS ARE RIGHT (RL) AN
D HOW MANY";
210 PRINT "ARE IN THE RIGHT SEQUENCE
 (RS).  IF";
220 PRINT "YOU MAKE A MISTAKE TYPING A
  WORD, YOU";
230 PRINT "WILL BE ABLE TO FIX IT AFTE
R YOU ENTER";
240 PRINT "ALL FIVE LETTERS.":PRINT
250 PRINT "PRESS RETURN TO START GAME"
260 GET #1,E:IF E<>155 THEN 260
```

Start some important strings from scratch, and print out the brief explanation of RS and RL. Note the use of the volume control on the Atari SOUND command in lines 305, 315 and 355.

```
280 GRAPHICS 0:POKE 752,1
283 ? :? :? :?
285 Q$="":L$="":C$=""
290 FOR X=1 TO 5
300 PRINT V$;
305 SOUND 0,200,2,X*2
310 FOR R=1 TO 150:NEXT R:NEXT X
315 SOUND 0,200,2,12
320 PRINT :PRINT "        RS = RIGHT
SEQUENCE":FOR R=1 TO 175:NEXT R
330 PRINT :PRINT "        RL = RIGHT
LETTER":PRINT :FOR R=1 TO 175:NEXT R
340 FOR X=1 TO 5
350 PRINT V$;
355 SOUND 0,200,2,(6-X)*2
360 FOR R=1 TO 150:NEXT R:NEXT X
370 FOR X=1 TO 750:NEXT X
375 SOUND 0,0,0,0
```

Initialize D$ to contain the possible letter choices, and randomly determine a code word.

continued on next page

continued from previous page

```
380 GRAPHICS 0:POKE 752,1
390 D$="ABCDEFGH"
400 FOR I=1 TO 5
410 Y=INT(RND(0)*8+1)
420 Q$(LEN(Q$)+1)=D$(Y,Y)
430 NEXT I
440 L$=Q$:L$(LEN(L$)+1)=Q$
```

Set up game display. The display is done in the normal text mode of the Atari (graphics 0) using the Atari's character graphics.

```
470 FOR F=2 TO 38
480 POSITION F,18:PRINT CHR$(149);:POS
ITION F,19:PRINT CHR$(21);
482 POSITION F,16:PRINT CHR$(149);:POS
ITION F,0:PRINT CHR$(160);
484 IF F<=29 THEN POSITION F,2:PRINT C
HR$(160);:POSITION F,15:PRINT CHR$(160
);
490 NEXT F
500 FOR F=2 TO 15
510 POSITION 2,F:PRINT CHR$(160);:POSI
TION 29,F:PRINT CHR$(160);:POSITION 38
,F:PRINT CHR$(160);
520 IF F>2 THEN POSITION 21,F:PRINT CH
R$(160);
530 NEXT F
540 POSITION 29,1:PRINT CHR$(160);"
    ";CHR$(160);
550 POSITION 32,2:PRINT "C";A$;"O";A$;
"O";A$;"E";
560 POSITION 31,6:PRINT "M";A$;"A";A$;
"S";A$;"T";A$;"E";A$;"R";
570 POSITION 30,14:PRINT "CHOICES:";
580 POSITION 30,15:PRINT "ABCDEFGH";
585 POSITION 30,13:PRINT "________";
590 POSITION 5,1:PRINT "1  2  3  4  5
   RL RS";
```

Set up some final variables to prepare the game, including initializing RS and RL.

```
595 PL=1
620 Q$=L$(6,10)
630 C$="     ":RS=0:RL=0
```

Get player's guesses, placing them into C$. Sound is used to verify keystrokes and emphasize incorrect keys.

```
640 FOR P=1 TO 5
645 POSITION P*3,3+PL:PRINT "TRY#";PL;
650 POSITION 3,17:PRINT "LETTER IN POS
ITION ";P;"?   ";
660 GET #1,M:SOUND 0,30,10,10
665 IF CHR$(M)<"A" OR CHR$(M)>"H" THEN
 SOUND 0,200,10,10:FOR M=1 TO 10:NEXT
M:SOUND 0,0,0,0:GOTO 660
670 C$(P,P)=CHR$(M):PRINT CHR$(M);
672 SOUND 0,0,0,0
680 POSITION P*3,3+PL:PRINT "  ";CHR$(
M);"  ";
690 NEXT P
```

Allow player to change his/her guesses if desired.

```
700 POSITION 3,17:PRINT "DO YOU WISH T
O CHANGE THEM (Y/N)";
710 GOSUB 1000
720 IF CHR$(Z1)<>"Y" THEN 730
722 POSITION 3,17:PRINT "TYPE IN AS US
UAL";:V1=1
724 FOR X=1 TO 500:NEXT X
726 GOTO 620
```

Plays a random sound, using all four voices, to show that the computer is analyzing player's guess.

```
730 FOR V=0 TO 3:SOUND V,RND(0)*255,IN
T(RND(0)*6+1)*2,INT(RND(0)*6+1)*2:NEXT
 V
```

Determine how many letters are in proper sequence. Then skips the win/lose routine by jumping to line 900.

```
735 AN$=C$:FOR L1=1 TO 5
740 FOR L1=1 TO 5:IF Q$(L1,L1)<>C$(L1,
L1) THEN 770
750 Q$(L1,L1)="*":RS=RS+1:RL=RL+1
760 C$(L1,L1)=" "
770 NEXT L1:GOTO 900
```

That part of the win/lose routine routine that is unique to the win section. Sets up the unique part of the display, and sets the message in ME$ to "YOU WON".

```
790 FOR QQ=1 TO 500:NEXT QQ
795 GRAPHICS 0:POKE 752,1
800 PRINT :PRINT :PRINT V$;V$;V$;V$
810 POSITION 10,9:PRINT "TRIES
SEQUENCE"
820 POSITION 12,10:PRINT PL;"
  ";L$(1,5)
```

Win/Lose routine. Flashes the message in ME$ for a while, then allows another game to be started.

```
822 ME$="YOU  WON"
825 POSITION 2,14:PRINT V$;V$;V$;V$;
830 FOR A=1 TO 5
840 POSITION 15,11:PRINT "      ";
850 FOR M=1 TO 30:SOUND 0,M,8,10:NEXT
M
860 POSITION 15,11:PRINT ME$;
870 FOR M=30 TO 1 STEP -1:SOUND 0,M,8,
10:NEXT M:NEXT A
880 POSITION 3,20:PRINT "PRESS RETURN
TO CONTINUE";
885 SOUND 0,18,2,10
890 GET #1,Z:SOUND 0,0,0,0:IF Z<>155 T
HEN 890
895 GOTO 280
```

Determine how many letters are correct (RL).

```
900 FOR N=1 TO 5:FOR M=1 TO 5
910 IF Q$(N,N)=C$(M,M) THEN Q$(N,N)="*
":C$(M,M)=" ":RL=RL+1
920 NEXT M:NEXT N
925 GOSUB 1100
```

Displays the player's score for the current guess.

```
930 POSITION 23,PL+3:PRINT RL;"  ";RS
```

Determines if player has won by guessing all five letters in the proper sequence (RS=5).

```
940 IF RS=5 THEN 790
```

Checks if ten turns have been played. If not, continue.

```
950 IF PL<>10 THEN PL=PL+1:GOTO 620
```

Lose routine. Prints that part of the disply unique to the lose routine, defines the message in ME$ to be displayed, and jumps to the win/lose routine at 825.

```
960 FOR QQ=1 TO 1000:NEXT QQ
962 GRAPHICS 0:POKE 752,1
965 PRINT :PRINT :PRINT V$;V$;V$;V$
970 POSITION 6,9:PRINT "LAST GUESS
  =SEQUENCE=";
975 POSITION 8,10:PRINT AN$;"
   ";L$(1,5);
980 ME$="YOU LOSE"
985 GOTO 825
```

Subroutine to get a character, and clear the message display line of the board.

```
1000 GET #1,Z1
1010 POSITION 3,17:PRINT "
                    ";
1020 RETURN
```

Subroutine to silence all four sound voices of the Atari.

```
1100 FOR M=0 TO 3:SOUND M,0,0,0:NEXT M
:RETURN
```

SARGON II

At HAYDEN, The Best Has Gotten Better.

Sargon, the program that came in first in the Creative Computing Microcomputer Chess Tournament, has become Sargon II. The game has been vastly improved and now has a faster response time. A new Level 0 has been incorporated for beginners. The board is easier to pre-set and there is now a Hint mode that provides suggestions from the computer. Sargon II took on the maxi-computers in the West Coast tournament and finished in the money! Shows more thinking power than you ever expected.

SARGON II SPECIAL! 20% OFF

Special

Sargon II 16K Level II Cassette $29.95 (S-80)	$23.95
Sargon II 24K Cassette Machine Language $29.95 (Apple)	$23.95
Sargon II 32K Disk $34.95 (S-80)	$28.95
Sargon II 48K Disk Machine Language $34.95 (Apple)	$28.95

Offer ends August 15

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603 673 0585

Go Public With Your Computer

by Harland Hill

Microcomputers are almost as addictive as fun, and microcomputer fans tend to be personality types with a special kind of dedication, similar, so the psychologists tell us, to that required of professional athletes. And, if the type carries through, such fans will also be family persons, interested in church or civic groups, wide open to new ideas, and responsive to logical challenge.

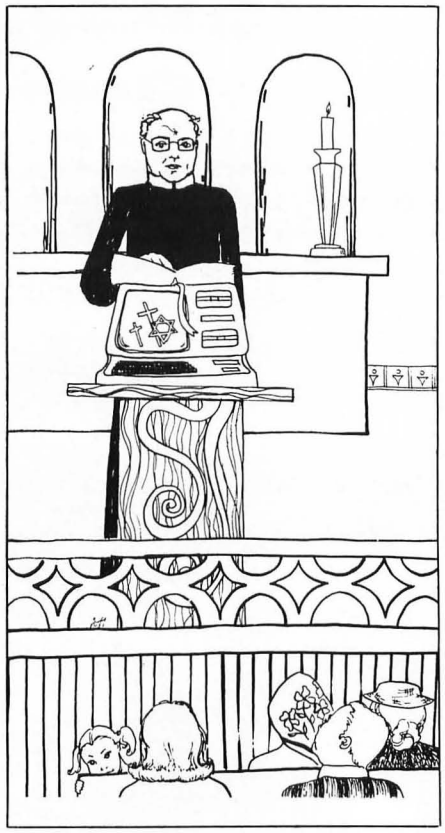
With all these traits in common, will you get the razzberry when you offer to share your private hobby with your favorite group? Nope. Not if you take along a shopping list that offers entertainment and new expertise plus ready-made solutions to the group's needs, publicity problems, and image woes. Do it right and your own problem will be overwork.

Small groups are still the great American phenomenon — clubs, study groups, coffee klatches, etc. And your small computer with its TV-type screen sells itself because it's so, well, charismatic. Your friends are sure to be on the make for ways, preferably ready-made, to expand their group's usefulness and public appeal as well as to broaden their own horizons.

My school teacher wife carries our S-80 to her special ed classroom in a simple aluminum case. The CRT sits on small brackets above the slide-out keyboard. When she removes the cover and plugs it in, it's as ready to go as the kids. My pastor likes it, too — can't keep his hands off it, in fact — and calls on it for youth parties, for announcements in the church vestibule, and even as a church and program promoter in the sanctuary. (Left it on during the sermon one Sunday and lost half his listeners.) Yet my micro is minimal — level II, 16K, Model I — but it's both popular and effective.

That's the good news. The bad news is that the shopping list may mean a little work — adapting games to fit your group, setting up good graphics displays, inventing slogans, quotes, and gimmicks to personalize your effort, and naturally, debugging. Stuff you're already into.

Newcomers to computers like visible action so show them games with good graphics and quick responses. **SoftSide** recently published two good



ones, "Mini Golf" and "Barney", both of which have people clamoring for their turns around here. Oh yes, and don't lecture or at least not much.

Better include a moving message program like Radio Shack's "Marquee" so that the doubters get a convincing look at a "practical" kind of display that holds interest. If you don't have that or one like it, try the following S-80 one liner which is compressed and adapted from one of George Blank's games:

```
100 CLEAR 256: CLS: PRINT CHR$(23): A$=" (Put any announcement here up to 130 letters) ": A$=STRINGS$(32,32)+A$+"  ": FORB=1TOLEN(A$): PRINT@448,MID$(A$,B,30): FORC=1T050: NEXT: NEXT: GOTO100
```

This simple gizmo may be better for small room display than one of the more expensive Machine Language programs and it's more adaptable.

Draw a picture of your club or church by holding a video display work sheet against a large photo held against a window, penciling in just the main lines and features. Then adapt your graphics to those lines. I was surprised to discover that folks seem to prefer seeing their very own building drawn and redrawn slowly (using set and reset) rather than by the faster methods. Smiley faces that wink and roll their eyes are appealing, too. So are good quotes which are centered phrase by phrase, but be sure to allow enough time to read them easily.

Any kind of instruction is more interesting — and more effectively taught — if you and the teacher can adapt one of the good teaching programs already in public domain. Make it self-grading and use a few wisecracks to correct wrong answers Kibitzers will be learning while they're cracking up so don't insist on too much privacy. Sunday school classes are ideal for computer-aided memorizing, factual learning, and moral simulations. Don't spare the graphics just because the material is factual and keep the variety flowing in your comments and pictures.

If you're lucky enough to have a printer, your club secretary needs it for the mailing list, dues, club information, and news releases. Lacking a printer, however, a midwest recording technician calls his pastor each morning to save the busy man the chore of looking up birthdays, anniversaries, and special events. The computerist keeps the material updated and available as his contribution to the church's outreach. Another pastor has his income tax done hassle-free on a member's tax program, checking it out each quarter. Church or club finances and budgets are computer fare, as well.

If the current educational literature is accurate, less than a majority of the nation's schools offer adequate computer instruction. It will do your club's image no harm at all if it's the first one on the block to offer a hands-on instruction course in BASIC. This is not hard to do if you follow one of the good teaching outlines that are easily found. Using your own computer for this won't harm it and your generosity may well give you that favorable, not to mention glamorous, image you've been seeking all your life. One guarantee: The course will be popular at all age levels and with both sexes.

Most clubs or churches have a store or shop owner with a display window on street level. They'll want

continued on page 91

Solution to Word-Search Puzzle #3

```
- - - - - - - - - - - - - - - - - - - -
- - - S I N C L A I R - - - - - - - - -
- - - - - - - O C M E M O R C - - - - -
- - - - - - - - O - - - - - - - - - - -
T R A G U H S I - - - - - - E L P P A -
- - - - - - S - - - - - - - - - - - - -
- N O R T A X E - - - - - - - - - - - -
- - - - C - - - - N O R T H S T A R - -
- - - - - - - - - - - - - - - - - - - -
- - - - O Y N A S - - - - - - - - - - -
- - - - - - - - - - - O K I D A T A -
- - - - - - - M I T A B R E V - - - - -
- - - - - E - - - - - - H T A E H - - -
- L - - E D - - - - - - - - - - - - - -
- L - - M I N O I T A V O N - - - - - -
- E - - U S - - L E X I C O N - - - - -
- B - - Q D - - - - - - - - - - - - - -
- R - - - R - - - - - - - - - - - - - -
- A - - - A - N I P A R R E T - - - - -
- T - - - H - - - - - - - - - - - - - -
- - - - - - - - - - M - - - - D I A B L O
- - - - - - - - - - A - - - - - - - - -
- - S T - - - - - - C - - - - - - - - -
- - C - C - - - - - R - - - - - - E S - -
- - I - - A - - - - O - - - - - - R N - -
- - N L K U R T A T - A - - - - O E - M
Y - O O - - - E - R - T - - F - D M - A
D - R R S - - P T O - A - - S - O E N X
N - T T U - - R - N - R - - A - M I O E
A N N M V C - A - I I I - - B - M S T L
T A E E R A - H - C - - - - - - O - A L
- S C - O E - S - S - - - - - - C - E S
- Y - - C T - - - - - - - - - - - - E -
- D P - - - - - - X - - N O S P E - Y - -
- - E - - - - - E - - - - - - - A - - -
- - R - - - D O B O L - - - - H - - - -
- - C - - E - - - - S - - - C - - - - -
- - O - E - - - - O - - - D - - - - - -
- - M L - - - - - N - - - - - - - - - -
- - - E X I D Y - - - - - - - - - - - -
```

BUGS, WORMS, and other undesirables

A slight error in lines 5110 and 5140 of "Math Decathlon", Part 3 (May, 1981) *will sometimes* yield an incorrect Lowest Common Denominator. The lines should be corrected to read:

```
5110 FOR I = INT(D/2) TO 2 STEP -1: IF INT(
I/A) = I/A AND INT(I/B) = I/B AND INT(I/C)
= I/C THEN LCD=I

5140 FOR I = INT(D/2) TO 2 STEP -1: IF INT(
I/A) = I/A AND INT(I/B) = I/B THEN LCD = I
```

A bug in "Battle at Sea" (April, 1981) will sometime cause a wiped-out player to be re-included in the combat. The following revision will correct the problem:

```
280 FOR I = 0 TO DEAD-1: IF OUT(I) < > P
THEN 290
282 P = P+1: IF P>NP THEN P=1
284 GOTO 280
```

There is a problem with the Machine Language Loader program used in "Divide and Conquer". The routine it uses contains an instruction that runs OK under DOS, but causes havoc with LEVEL II BASIC. Line 50320 should read:

```
50320 DATA 50,-3,62,50,50,-1,0,225,201,300
```

The Atari One Liner published on page 35 of the May issue which was credited to Dave Field of Bangor, ME, was actually by Andrew Field of Bangor, ME.

Data Base continues... There is a problem in the S-80 version that affects the multi-level search feature. Line number 8160 should read:

```
8160 PRINT@832,"1) ITEM MUST MEET ALL CONDI
TIONS";:INPUT"2) ITEM MAY MEET ANY CONDITION
";BS:IFBS<1OR BS>2THEN8160
```

In addition, here is an improvement. Change line 8390 to read:

```
8390 IFBS=2 THEN POKE SC,S1:POKE SC+1,S2:PR
INT896,"THAT'S ALL":GOSUB60000
```

This keeps "THAT'S ALL" from being printed out after the listing.

Kidnapped

Original S-80 version by Peter Kirsch. Apple translation by Carl Mueller (translation contest winner). Atari translation by Rich Bouchard and Alan Zett.

The Apple version of "Kidnapped" requires Applesoft and 24K. The Atari version requires 32K.

The original S-80 version of this nine-in-one adventure was published in last December's issue of **SoftSide**.

Nine-in-one? Well, judge for yourself. You begin this adventure on the ninth floor of a strange building, the victim of a kidnapping. Floor by floor, you must find your way to freedom, past the kidnapper who sits somewhere counting his ransom money. Traps await you everywhere, and your life is constantly in danger.

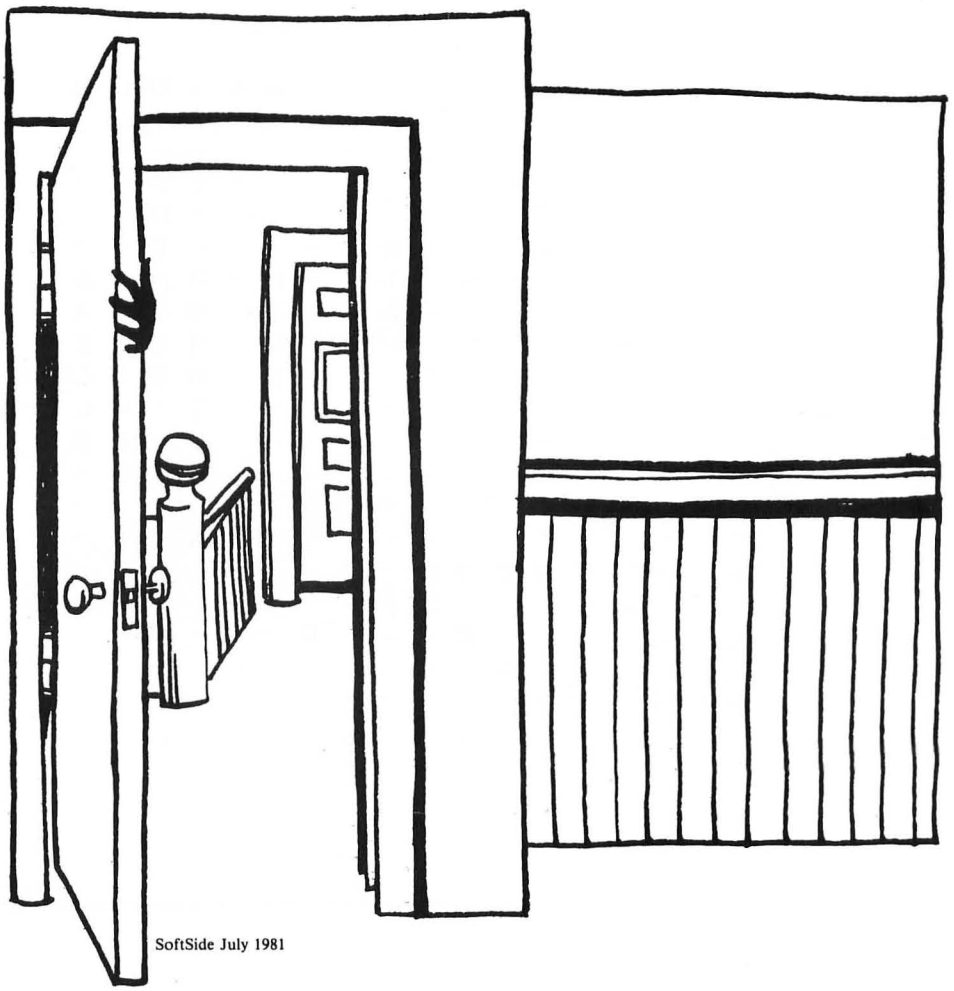
Each floor's adventure is independent of the others; you cannot carry items from one to another. Everything you need to escape from a given floor can be found there — if you can just figure out where to look.

As in other adventures, one or two word commands are used to communicate with the computer, such as GET AX, DROP AX, or OPEN DOOR. To move in particular direction, simply enter the initial letter of that direction: N, S, E, W, U, D. Entering LOOK will refresh your memory about your environment, and entering INVENTORY or just I will give you a tally of everything you're carrying.

VARIABLES

A: Current player location.
A (X): Item location (room number).
A$: Player input command.
A$ (X): Temporary storage of items.
B, G, K, K1, K2, K4, V: Loop counters.
B (X): Holds room numbers accessible from current locations.
B$ (X): Commands.
BO: Balloon status (0 = deflated, 1 = inflated, 2 = tied to string).
BR: Book read?
C$ (X): Items carried by player.
C, CF, CT, G, KY: Message flags (if 0, respective message appears).
C1, EF, JK, R1, SD, SP, UM: Item flags (0 = natural state, 1 = changed).
D (X): Command modes.
DK: Flashlight on?
DP, FR, PF, PM, SC: Monster or hazard flags (0 = active, 1 = absent).
DT: Dark flag.
E, W, N, S, U, D: Direction pointers.
E$: Picks last three letters of object command D$.
F, K1, K3, J, TM, X, Y: Work variables.
FL: Floor pointer.
G (X): Door status (0 = locked, 1 = open).
H$ (X): Permanent storage of items.
I (X): If item X is carried by player, then I (X) = 1.
J$, N$: Room descriptions.
M$, R$: Used to change, add, or remove an item in room or if carried.
PT: Plant status (0 = small, 1 = huge).
RS: Rope status (0 = loose, 1 = tied to stake, 2 = stretched across quicksand).
TI: Current time (9th floor only).

```
5  TEXT : GOTO 200

 Room descriptions.

10  ON A GOTO 11,12,13,14,15,16,1
     7,18,19,20,21,22,23,24,25,26
     ,27,28,29,30,31,32,33,34,35,
     36,37,38,39,40,41,42,43,44,4
     5,46,47,48,49,50,51,52,53,54
     ,55,56,57,58,59,60,61,62,63,
     64,65,66,67,68,69,70,71,72,7
     3,74,75,6000
11  PRINT J$:W = 4:N = 2:E = 8:S =
     7: GOTO 350
12  PRINT N$:N = 3:S = 1: GOTO 35
     0
13  PRINT "IN A CLOSET.":S = 2: GOTO
     350
14  PRINT J$:N = 5:E = 1:S = 6: GOTO
     350
15  PRINT N$:S = 4: GOTO 350
16  PRINT "IN A RESTROOM.":N = 4:
     GOTO 350
17  PRINT N$:N = 1: GOTO 350
18  PRINT J$:W = 1:N = 9:S = 10: GOTO
     350
```

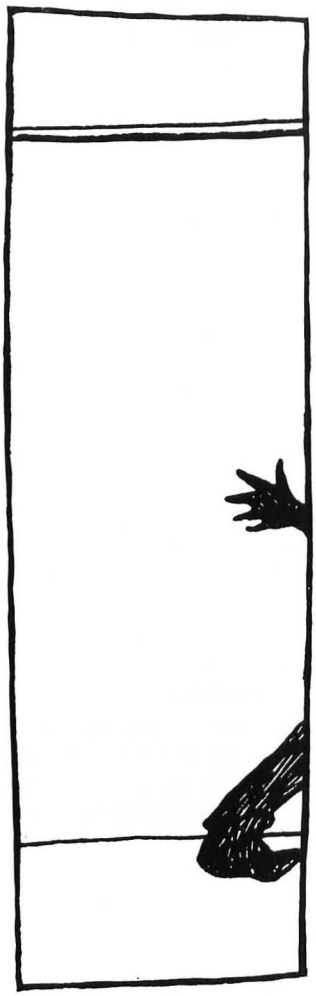


```
19 PRINT "IN AN ELEVATOR.":S = 8
    : GOTO 350
20 PRINT "IN A MAINTENANCE ROOM.
    ":N = 8: GOTO 350
21 PRINT "IN A CRAWLSPACE ATOP T
    HE ELEVATOR":D = 9: GOTO 350

22 PRINT "ON A VERY NARROW LEDGE
    .": GOTO 350
23 PRINT J$:W = 14:S = 16: GOTO
    350
24 PRINT "IN A VISITORS' LOUNGE.
    ":W = 15:E = 13: GOTO 350
25 PRINT "IN A CLOSET.":E = 14: GOTO
    350
26 PRINT N$:N = 13:E = 17: GOTO
    350
27 PRINT N$:W = 16: GOTO 350
28 PRINT J$:W = 20:N = 19:E = 24
    :S = 21: GOTO 350
29 PRINT N$:S = 18: GOTO 350
30 PRINT "IN A TOOL CRIB.":E = 1
    8: GOTO 350
31 PRINT J$:N = 18:E = 22: GOTO
    350
32 PRINT N$:W = 21:E = 23: GOTO
    350
33 PRINT "IN A SMALL STORAGE ROO
    M.":W = 22: GOTO 350
34 PRINT J$:W = 18: GOTO 350
35 PRINT "IN A NARROW STAIRWAY."
    :W = 24: GOTO 350
36 PRINT J$:W = 31:N = 27:E = 29
    :S = 30: GOTO 350
37 PRINT N$:N = 28:S = 26: GOTO
    350
38 PRINT "IN A CLOSET.":S = 27: GOTO
    350
39 PRINT "IN A GAME ROOM.":W = 2
    6: GOTO 350
40 PRINT N$:N = 26: GOTO 350
41 PRINT J$:W = 33:N = 32:E = 26
    : GOTO 350
42 PRINT "IN A STORE ROOM.":S =
    31: GOTO 350
43 PRINT N$:E = 31: GOTO 350
44 PRINT "ON A LARGE LEDGE BY TH
    E WINDOW.":W = 36: GOTO 350
45 PRINT "IN THE VAULT.":W = 36:
     GOTO 350
46 PRINT "ON A LARGE LEDGE BY TH
    E WINDOW.":E = 34: GOTO 350
47 PRINT N$:S = 41: GOTO 350
48 PRINT J$:E = 43:N = 39:S = 40
    : GOTO 350
49 PRINT N$:S = 38: GOTO 350
50 PRINT N$:N = 38: GOTO 350
51 PRINT J$:W = 43:N = 37:S = 42
    : GOTO 350
52 PRINT N$:N = 41: GOTO 350
53 PRINT J$:W = 38:E = 41: GOTO
    350
54 PRINT "IN A STAIRCASE.": GOTO
    350
55 PRINT J$:N = 46:E = 48:S = 47
    : GOTO 350
56 PRINT N$:S = 45: GOTO 350
57 PRINT "IN A LIBRARY.":N = 45:
     GOTO 350
58 PRINT "IN THE SWIM ROOM":W =
    45: GOTO 350
59 PRINT "IN THE SWIM ROOM":N =
    50:E = 51: GOTO 350
60 PRINT "IN A LAUNDRY ROOM":S =
    49: GOTO 350
61 PRINT "IN A DINING ROOM.":W =
    49: GOTO 350
62 PRINT J$:N = 53:E = 54: GOTO
    350
63 PRINT "IN A RESTROOM.":S = 52
    : GOTO 350
64 PRINT J$:W = 52:E = 55: GOTO
    350
65 PRINT J$:W = 54:N = 56: GOTO
    350
66 PRINT "IN A LABORATORY.":S =
    55: GOTO 350
67 PRINT "IN A CHILD'S PLAYROOM.
    ":N = 55: GOTO 350
68 PRINT N$:W = 59: GOTO 350
69 PRINT "IN A LOUNGE.":E = 58: GOTO
    350
70 PRINT "IN A CRAWLWAY.":E = 61
    : GOTO 350
71 PRINT "IN A CRAWLWAY.":W = 60
    : GOTO 350
72 GOTO 54
73 PRINT "IN A LOUNGE.":E = 64: GOTO
    350
74 PRINT J$:W = 63: GOTO 350
75 PRINT "IN THE ENTRANCE HALL."
    : GOTO 350

 Initialization.

200 DIM A$(58),A(58),B(12),B$(39
    ),C$(7),D(39),H$(58),I(58)
210 FOR A = 1 TO 55: READ A$(A),
    A(A):H$(A) = A$(A): NEXT : FOR
    A = 1 TO 39: READ B$(A): NEXT
    : FOR A = 13 TO 39: READ D(A
    ): NEXT
220 A = 1:G = 1:FL = 9:TM =  - 2:
    J$ = "IN A HALLWAY.":N$ = "I
    N AN OFFICE."

 Description of current location.

300 N = 0:W = 0:E = 0:S = 0:U = 0
    :D = 0:Y = 0: HOME : PRINT "
    YOU'RE ";: IF DK = 0 AND TM >
    24 THEN DT = 1
310 IF DT <  > 1 THEN 10
320 HOME : PRINT  CHR$ (7)"POWER
     FAILURE!  IT'S TOO DARK TO
    SEE!":DT = 2: GOTO 390
350 PRINT "FLOOR "FL"     SOME E
    XITS ARE:"
355 FOR B = 1 TO 12:B(B) = 0: NEXT

360 IF W THEN  PRINT " WEST";:B(
    1) = W:B(2) = W
361 IF A = 41 AND FR = 0 THEN 36
    5
362 IF N THEN  PRINT " NORTH";:B
    (3) = N:B(4) = N
363 IF A = 16 AND DP = 0 THEN 36
    8
364 IF E THEN  PRINT " EAST";:B(
    5) = E:B(6) = E
365 IF S THEN  PRINT " SOUTH";:B
    (7) = S:B(8) = S
366 IF U THEN  PRINT " UP";:B(9)
     = U:B(10) = U
367 IF D THEN  PRINT " DOWN";:B(
    11) = D:B(12) = D
368 PRINT : PRINT
370 FOR B = 1 TO 58: IF A =  ABS
    (A(B)) THEN C = C + 1: GOTO
    372
371 NEXT : PRINT : GOTO 374
372 IF C < 2 THEN  PRINT "THINGS
     YOU SEE HERE:"
373 PRINT " "A$(B): GOTO 371
374 IF A = 37 AND FR = 1 THEN  PRINT
    "YOU FORGOT YOU WERE NAKED."
    ,"YOU BLUSH AND RUN OUT.":A =
    41: GOTO 5000
375 IF G = 1 THEN  PRINT "YOU AW
    AKEN ON THE 9TH FLOOR OF A S
    TRANGEBUILDING, OBVIOUSLY A
    KIDNAP VICTIM. YOUARE ALONE
    AT THE MOMENT AND MUST ESCAP
    E FROM THE BUILDING, FLOOR B
    Y FLOOR.":G = 0
376 IF A = 7 THEN  PRINT "YOU SE
    E A SMALL LEDGE OUTSIDE THE
    WINDOW";: IF KY = 0 THEN  PRINT
    "AND A SINGLE KEY TO A KEY C
    HAIN THERE."
377 IF RS = 2 THEN  IF A = 64 OR
    A = 65 THEN  PRINT "ROPE IS
    STRETCHED ACROSS QUICKSAND."

378 IF A = 33 THEN  PRINT "THERE
     IS A WIDE, LONG LEDGE OUTSI
    DE THE WINDOW."
382 IF A = 56 AND JK = 0 THEN  PRINT
    "LABELS ON BOTTLES:",,"   SO
    LUTION: ANTIDOTE",,"  FLUID
    : UNDECIPHERABLE"
383 IF A = 58 THEN  PRINT "A DRO
    OLING ALLIGATOR BLOCKS YOUR
    WAY    EAST. HE HAS THE REMA
    INS OF A KIDNAPPER IN HIS MO
    UTH. YOU CATCH A GLIMPSE OF
    A  STAIRCASE PAST THE ALLIGA
    TOR."
```

continued on next page

continued from previous page

```
384  IF A = 64 AND RS < 2 THEN  PRINT
     "A HUGE BOG OF QUICKSAND BLO
     CKS YOUR WAY EAST. THE FRONT
      ENTRANCE IS THERE, YOUR WAY
      TO SAFETY. THERE IS A LARGE
      HOOK ON THE OTHER SIDE AND
     A TENT STAKE ON THIS SIDE."
385  IF A = 59 THEN  PRINT "THERE
      IS A TRAP DOOR ABOVE YOU."
386  IF A = 61 AND R1 = 0 THEN  PRINT
     "THROUGH THE TRAP DOOR YOU S
     EE A LONG    COILED ROPE ON
     THE FLOOR BELOW."
387  IF A = 10 AND C1 = 1 AND CF =
     0 THEN  PRINT "A FLASHLIGHT
     IS THERE."
388  IF A = 10 AND C1 = 1 AND CT =
     0 THEN  PRINT "ELECTRICAL TA
     PE IS THERE."
389  GOSUB 1600


 Player input.


390 V =  FRE (0):C = 0:TM = TM +
     1: PRINT : INPUT "COMMAND? "
     ;A$: IF DT = 2 THEN 7500
391  IF A$ = "LOOK" THEN 300
392  IF A$ = "JUMP" THEN 800
```



```
393  IF A$ = "SWIM" THEN 974
394  IF A$ = "WAIT" THEN 1050
395  PRINT : FOR B = 1 TO 12: IF
     A$ = B$(B) THEN 397
396  NEXT : GOTO 399
397  IF B(B) <  > 0 THEN A = B(B)
     : GOTO 300
398  PRINT "YOU CAN'T GO THAT WAY
     .": GOTO 390
399  IF A$ <  > "I" AND  RIGHT$ (
     A$,3) <  > "ORY" THEN 410
400  PRINT "YOU ARE CARRYING:": FOR
     K = 1 TO 7: PRINT C$(K);"
     ";: IF  POS (0) > 19 THEN  PRINT

401  NEXT : GOTO 390
410  FOR B = 13 TO 39:F =  LEN (B
     $(B)): IF  LEFT$ (A$,F) = B$
     (B) THEN 450
420  NEXT : PRINT "DON'T KNOW WHA
     T " CHR$ (34)A$ CHR$ (34)" M
     EANS.": GOTO 390


 String sorting routine.


450 D$ =  MID$ (A$, LEN (B$(B)) +
     2):E$ =  RIGHT$ (D$,3)


 'GET' command.  Any special
 conditions are checked to see if
 a dangerous, stationary, or
 otherwise hidden item can be
 carried; else item is given to
 player and I(X) is set to 1.


460  IF D(B) <  > 1 THEN 490
461  IF E$ <  > "GHT" AND E$ <  >
     "APE" AND A <  > 10 THEN 465

462  IF A = 10 AND C1 = 0 THEN  PRINT
     "CABINET IS LOCKED.": GOTO 5
     000
463  IF E$ = "GHT" AND CF = 0 THEN
     J = 56:A$(J) = "FLASHLIGHT":
     H$(J) = A$(J):A(J) = 10:CF =
     1
464  IF E$ = "APE" AND CT = 0 THEN
     J = 57:A$(J) = "ELECTRICAL T
     APE":H$(J) = A$(J):A(J) = 10
     :CT = 1
465  IF A = 7 AND KY = 0 AND E$ =
     "KEY" AND I(4) <  > 1 THEN  PRINT
     "YOUR ARM IS TOO SHORT TO RE
     ACH IT.": GOTO 5000
466  IF A = 7 AND KY = 0 AND E$ =
     "KEY" THEN J = 58:A$(J) = E$
     :H$(J) = A$(J):A(J) = 7:I(58
     ) = 1:KY = 1
467  IF A = 63 AND E$ = "KEY" THEN
      PRINT "WE ALL KNOW PIANOS H
     AVE KEYS.":J = 56:A$(J) = E$
     :H$(J) = A$(J):A(J) = 63:I(5
     6) = 1
468  IF (A = 14 AND E$ = "NHA" AND
     PF = 0) OR (A = 17 AND E$ =
     "NAN") OR (A = 58 AND E$ = "
     TOR") THEN 7000
469  IF A = 35 AND (E$ = "PER" OR
     E$ = "LAR" OR E$ = "NEY") AND
     I(27) = 0 THEN  PRINT "YOU G
     ET TOO CLOSE TO HIM. HE","JU
     MPS UP AND STRANGLES YOU.": GOTO
     7300
470  IF A = 35 AND (E$ = "PER" OR
     E$ = "LAR" OR E$ = "NEY") THEN
      PRINT "KIDNAPPER SEES YOUR
     GUN AND FREEZES.    YOU GRAB
      A DOLLAR.":A$(56) = "DOLLAR
     ":H$(56) = A$(56):A(56) = 35
     :PM = 1:E$ = "LAR"

471  IF I(48) = 1 AND E$ = "TER" AND
     A = 58 THEN I(57) = 1:R$ = "
     CUP OF WATER":K3 = 48: GOSUB
     1100:H$(48) = R$: GOTO 4900
472  IF E$ = "JAR" THEN  PRINT "W
     HICH ONE?": GOTO 5000
473  FOR J = 1 TO 58: IF E$ =  RIGHT$
     (A$(J),3) AND A =  ABS (A(J)
     ) THEN 475

474  NEXT : PRINT "THERE'S NO "D$
     " HERE.": GOTO 390
475  IF A(J) < 0 THEN  PRINT "BE
     REASONABLE NOW.  THAT'S IMPO
     SSIBLE.": GOTO 390
476  IF E$ <  > "OOK" THEN 479
477  PRINT "TITLE OF BOOK: HOW TO
      ";: IF A = 47 THEN  PRINT "
     SWIM"
478  IF A = 64 THEN  PRINT "WALK
     A TIGHTROPE"
479  PRINT "OK!"; FOR K = 1 TO 7:
      IF C$(K) = "" THEN C$(K) =
     A$(J): GOSUB 3500:H(K) = J:A
     $(J) = "":A(J) = 0: GOTO 500
     0
480  NEXT


 'DROP' command.  If item is being
 carried, it is dropped in current
 room.  Program checks for any
 changes that might occur if certain
 item is dropped in a particular
 room.  I(X) is set to room number.


490  IF D(B) <  > 2 THEN 600
500  FOR J = 1 TO 7: IF E$ =  RIGHT$
     (C$(J),3) THEN 520
510  NEXT : PRINT "YOU'RE NOT CAR
     RYING IT.": GOTO 390
520  FOR K = 1 TO 58: IF A$(K) =
     "" THEN A$(K) = C$(J):A(K) =
     A:H(J) = 0: GOTO 530
525  NEXT
```

```
530  PRINT "OK!": FOR K1 = 1 TO 5
     8: IF C$(J) = H$(K1) THEN C$
     (J) = "": GOTO 540
535  NEXT
540  IF A = 14 AND E$ = "ILL" THEN
      PRINT "YOU DROP PILL IN AQU
     ARIUM.":PF = 1:K3 = 15:M$ =
     "SLEEPING PIRANHA": GOSUB 12
     00: GOTO 5000
545  IF A = 16 AND DP = 0 AND E$ =
     "NHA" THEN  PRINT "PIRANHA D
     EVOURS DOBERMAN PINCHER, THE
     N  DIES OF OVEREATING.":DP =
     1:K3 = 15:M$ = "GLUTTONOUS D
     EAD PIRANHA": GOSUB 1200:K1 =
     15:A(57) =  - 16:A$(57) = "E
     ATEN DOBERMAN":H$(57) = A$(5
     7): GOTO 590
550  IF A = 32 AND E$ = "LAR" THEN
     A$(57) = "LONG STRING":H$(57
     ) = A$(57):A(57) = 32:K1 = 5
     7:K3 = 56:M$ = "": GOSUB 120
     0
590  I(K1) = A: GOTO 5000
```

'GO' command. If conditions are met, player goes where requested. (You can't go through a locked door or down a broken staircase.) Variable A is then set to new location.

```
600  IF D(B) <  > 3 THEN 620
601  IF A = 16 AND E$ = "MAN" OR
     A = 58 AND E$ = "TOR" THEN 7
     000
602  IF A = 41 AND E$ = "IRE" OR
     A = 64 AND E$ = "AND" THEN 7
     200
603  IF A = 25 AND E$ = "IRS" AND
     SC = 0 THEN  PRINT "STEP IS
     MISSING! YOU FELL DOWN THE
         STAIRS.": GOTO 7200
604  IF E$ = "OPE" AND R1 = 1 AND
     A = 17 THEN A = 18:R1 = 0: GOTO
     8000
605  IF E$ = "OPE" AND R1 = 1 AND
     A = 61 THEN A = 62:R1 = 0: GOTO
     300
606  IF E$ = "ANT" AND PT = 1 AND
     A = 59 THEN A = 60: GOTO 300

607  IF E$ = "ANT" AND PT = 1 AND
     A = 60 THEN A = 59: GOTO 300

608  IF E$ = "OPE" AND A = 65 THEN
     A = 64: GOTO 300
609  IF E$ = "OPE" AND A = 64 AND
     RS = 2 AND BR = 1 THEN A = 6
     5: GOTO 300
610  IF E$ = "OPE" AND A = 64 AND
     RS = 2 THEN  PRINT "YOU DON'
     T KNOW HOW TO WALK A TIGHTRO
     PE.": GOTO 7200
611  IF E$ = "DOW" THEN  IF A = 3
     4 OR A = 36 THEN A = A - 1: GOTO
     300
612  IF E$ = "DGE" AND A = 7 THEN
      PRINT "LEDGE BREAKS!": GOTO
     7200
613  IF E$ = "DGE" THEN  IF A = 3
     3 OR A = 35 THEN A = A + 1: GOTO
     300
614  IF E$ = "IRS" THEN  IF A = 2
     5 OR A = 44 OR A = 62 THEN A
      = A + 1: GOTO 8000
615  IF E$ = "OOR" THEN  IF A = 2
     4 OR A = 43 OR A = 30 OR A =
     65 THEN  IF G((A / 10 -  INT
     (A / 10)) * 10) = 0 THEN  PRINT
     "THE DOOR'S LOCKED.": GOTO 5
     000
616  IF E$ = "OOR" THEN  IF A = 2
     4 OR A = 43 OR A = 30 OR A =
     65 THEN A = A + 1: GOTO 300
617  IF E$ = "OOR" AND A = 55 THEN
      IF SD = 1 OR JK = 1 THEN A =
     57: GOTO 300
618  IF E$ = "OOR" AND A = 55 THEN
      PRINT "IT'S STUCK! YOU'RE N
     OT STRONG ENOUGH    TO OPEN
     IT!": GOTO 5000
619  IF E$ = "IDE" AND A = 57 THEN
      PRINT "YOU SLIDE THROUGH HO
     LE IN FLOOR.":A = 58: GOTO 8
     000
```

'OPEN' command. Checks first if player has a key for door or cabinet.

```
620  IF A = 9 THEN  IF E$ = "OOR"
      OR E$ = "AIR" THEN  IF A =
     I(3) THEN  PRINT "YOU STEP O
     N CHAIR AND JUST MANAGE TO
      REACH THE TRAP DOOR.":A =
     11: GOTO 5000
621  IF A = 9 THEN  IF E$ = "OOR"
      OR E$ = "AIR" THEN  PRINT "
     YOU CAN'T REACH IT.": GOTO 5
     000
630  IF D(B) <  > 4 THEN 650
631  IF E$ = "OOR" AND A = 55 AND
     JK = 0 THEN 618
632  IF E$ = "OOR" AND A = 55 THEN
      GOSUB 4000:G(1) = 1:SD = 1:
      GOTO 4900
633  IF E$ = "OOR" THEN  IF A = 4
     3 OR A = 24 OR A = 65 THEN  IF
     I(56) = 1 OR A = 43 AND I(42
     ) = 1 THEN  GOSUB 4000:G((A /
     10 -  INT (A / 10)) * 10) =
     1: PRINT "DOOR'S OPEN": GOTO
     5000
634  IF E$ = "OOR" THEN  IF A = 4
     3 OR A = 24 OR A = 65 THEN  PRINT
     "YOU NEED A KEY": GOTO 5000
```

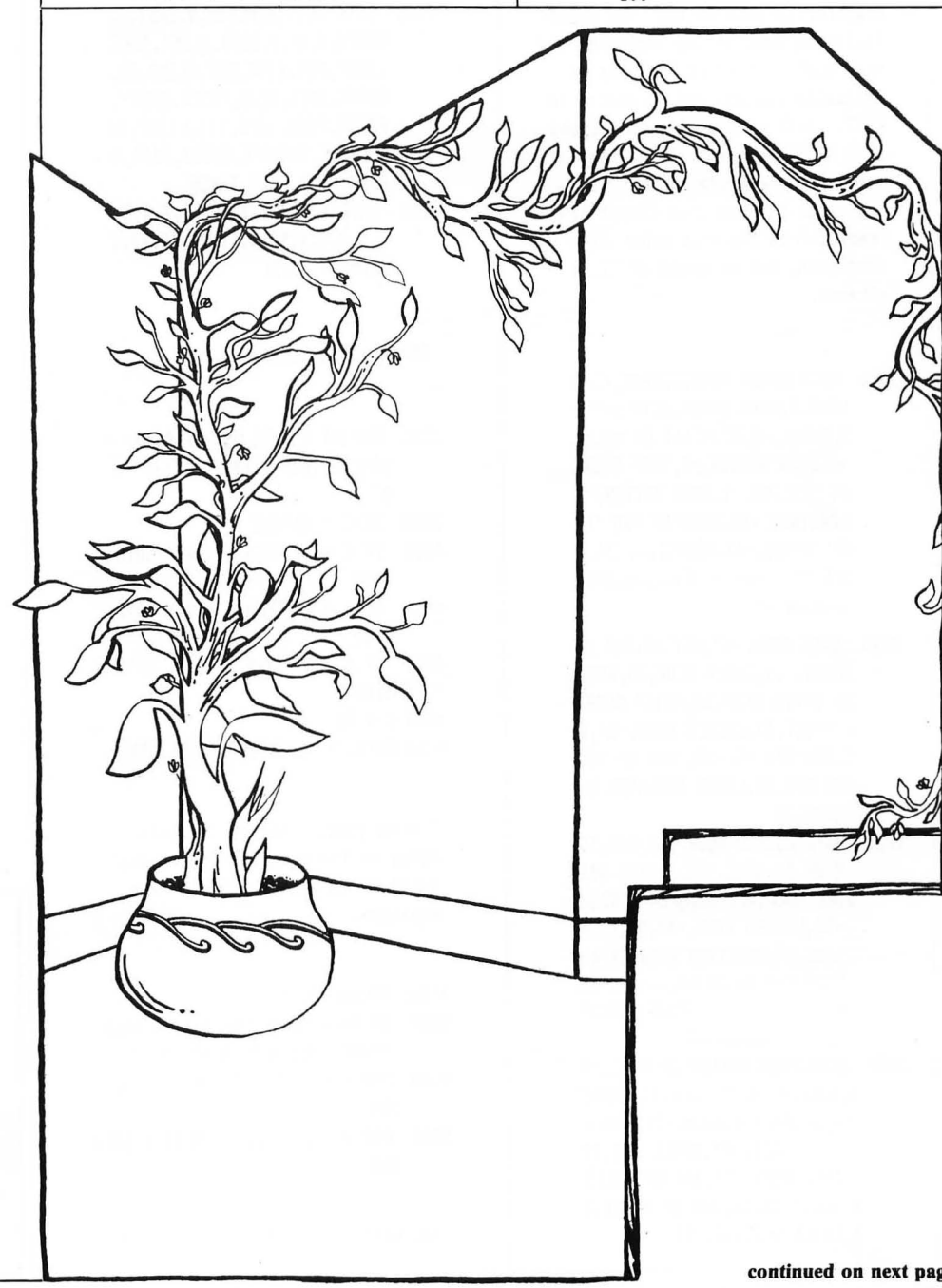
continued on next page

continued from previous page

```
635  IF A = 10 AND C1 = 0 AND I(5
     8) = 1 THEN  PRINT "CABINET'
     S OPEN":C1 = 1: GOTO 5000
636  IF I(39) = 1 AND E$ = "LLA" THEN
     K3 = 39:R$ = "OPEN POPPIN'S
     UMBRELLA": GOSUB 1100:UM = 1
     : GOTO 4900


 'READ' command.


650  IF D(B) <  > 5 THEN 700
651  IF E$ = "ITI" AND A = 6 THEN
      PRINT "IT SAYS: WATCH OUT F
     OR LIVE ONES": GOTO 5000
652  IF E$ = "ITI" AND A = 53 THEN
      PRINT "GRAFFITI: DO YOU HAV
     E A SPLIT            PERSONAL
     ITY?": GOTO 5000
655  IF E$ = "OTE" THEN  IF I(1) =
     A OR I(1) = 1 OR A = 2 AND I
     (1) = 0 THEN  PRINT "NOTE SA
     YS: " CHR$ (34)"IMPORTANT TV
      PROGRAM ON" CHR$ (34): GOTO
     390
660  IF E$ = "IGN" AND A = 32 THEN
      PRINT "SIGN: STRING COSTS $
     1.00": GOTO 5000
670  IF E$ = "TV" AND A = 5 THEN
      PRINT "BULLETIN:",,,"POWER
     WILL BE SHUT OFF AT MIDNIGHT
     ...": GOTO 5000
680  IF E$ = "OCK" AND A = 5 THEN
      PRINT "TIME: ";:TI =  INT (
     TM / 2): IF TI > 12 THEN  PRINT
     "PAST MIDNIGHT": GOTO 5000
690  IF E$ = "OCK" AND A = 5 THEN
      PRINT TI"P.M.": GOTO 5000
695  IF E$ = "OOK" THEN  IF I(38)
      = 1 OR I(52) = 1 THEN BR =
     1: GOTO 4900

 'PUSH' command.


700  IF D(B) <  > 6 THEN 830
702  IF E$ <  > "TON" THEN 1000
704  IF A = 9 AND EF = 1 THEN  PRINT
     "ELEVATOR GOES DOWN!":A = 13
     : GOTO 8000
710  IF A = 42 AND FR = 0 THEN  PRINT
     "SPRINKLER TURNED ON. FIRE O
     UT.","BUT YOUR CLOTHES ARE V
     ERY WET. YOU TAKE THEM OFF.
     YOU ARE NOW NAKED.":FR = 1:A
     $(57) = "WET CLOTHES":H$(57)
      = A$(57):A(57) =  - 42: GOTO
     5200

720  IF A = 46 AND SP = 0 THEN  PRINT
     "CLICK!":SP = 1: GOTO 5000
730  PRINT "NOTHING HAPPENS.": GOTO
     5000

 'JUMP' command.


800  IF A = 60 OR A = 61 OR A = 6
     4 THEN 7100
802  IF A = 34 OR A = 36 THEN  IF
     I(25) = 1 AND BO = 2 THEN  PRINT
     "BALLOON CARRIES YOU DOWN ON
     E FLOOR.":A = 43: GOTO 8000
804  IF A = 34 OR A = 36 THEN 710
     0
806  IF I(39) = 1 AND UM = 1 THEN
      PRINT "YOU FLOAT DOWN ONE F
     LOOR.":A = 52: GOTO 8000
808  IF I(39) = 1 THEN  PRINT "UM
     BRELLA WASN'T OPEN!": GOTO 7
     10
810  GOTO 730


 'TIE' command.


830  IF D(B) <  > 9 THEN 850
831  IF A = 11 AND E$ = "RES" AND
     TM < 23 THEN  PRINT "YOU'RE
     ELECTROCUTED!": GOTO 7200
832  IF A = 11 AND E$ = "RES" AND
     I(57) = 1 THEN EF = 1:A$(12)
      = "TAPED WIRES":H$(12) = A$
     (12): GOTO 4900
833  IF A = 11 AND E$ = "RES" THEN
      PRINT "WIRES FALL APART AGA
     IN.": GOTO 5000
835  IF E$ = "OON" OR E$ = "ING" THEN
      IF BO = 1 AND I(25) = 1 AND
     I(57) = 1 THEN K3 = 57:R$ =
     "": GOSUB 1100:K3 = 25:R$ =
     "LARGE INFLATED BALLOON WITH
      STRING": GOSUB 1100:BO = 2:
      GOTO 4900
840  IF E$ <  > "OPE" THEN 850
842  IF A = 64 AND I(54) = 1 THEN
     K3 = 54:R$ = "": GOSUB 1100:
     A$(54) = "END OF ROPE TIED T
     O STAKE":H$(54) = A$(54):A(5
     4) =  - 64:RS = 1: GOTO 4900

844  IF A = 17 AND I(14) = 1 THEN
      PRINT "ROPE TIED TO DESK":K
     3 = 14:R$ = "": GOSUB 1100:A
     $(8) = "ROPE HANGING OUT WIN
     DOW":R1 = 1:H$(8) = A$(8):A(
     8) =  - 17: GOTO 5000

 'LIGHT', 'MAKE', 'GLUE', 'INFLATE',
 'SHOOT', and 'KNIT' commands.


850  IF D(B) <  > 10 THEN 900
855  IF A < 13 AND E$ = "GHT" AND
     I(56) = 1 THEN K3 = 56:R$ =
     "LIT FLASHLIGHT": GOSUB 1100
     :H$(56) = R$:DK = 1: GOTO 49
     00

900  IF D(B) <  > 11 THEN 930
905  IF A = 19 AND E$ = "KEY" AND
     I(21) = 1 THEN A$(56) = "CRU
     DE KEY":H$(56) = A$(56):A(56
     ) = 19: GOTO 4900
930  IF D(B) <  > 12 THEN 950
935  IF A = 25 AND I(19) = 1 THEN
      IF I(20) = 1 OR A = I(20) THEN
      IF E$ = "TEP" OR E$ = "IRS"
      THEN  PRINT "STAIR'S FIXED"
     :SC = 1:R$ = "":K3 = 20: GOSUB
     1100:M$ = "": GOSUB 1200: GOTO
     5000
950  IF D(B) <  > 13 THEN 960
955  IF I(25) = 1 THEN  IF I(24) =
     1 OR A = I(26) THEN  IF E$ =
     "OON" THEN K3 = 25:R$ = "LAR
     GE INFLATED BALLOON": GOSUB
     1100:H$(25) = R$:BO = 1: GOTO
     4900
960  IF D(B) <  > 14 THEN 970
963  IF E$ = "GUN" OR E$ = "PER" THEN
      IF I(27) = 1 AND A = 35 THEN
      PRINT "GUN HAD BLANKS!",,,"
     KIDNAPPER SHOOTS YOU!": GOTO
     7200
967  IF E$ = "GUN" OR E$ = "PER" THEN
      IF I(27) = 1 THEN  PRINT "G
     UN MISFIRES.": GOTO 5000
970  IF D(B) <  > 15 THEN 980
972  IF FR = 1 AND I(32) = 1 AND
     I(33) = 1 AND E$ = "HES" THEN
      PRINT "YOU HAVE KNITTED A F
     INE SUIT AND ARE    WEARING
     IT.":FR = 2: GOTO 5000

 'SWIM' command.


974  IF SP = 0 THEN 1000
975  IF A = 49 THEN A = 48: GOTO
     300
976  IF A = 48 AND BR = 0 THEN  PRINT
     "YOU DON'T KNOW HOW TO SWIM!
     ": GOTO 7200
977  IF A = 48 THEN A = 49: GOTO
     4900


 'DRINK' command.


980  IF D(B) <  > 18 THEN 985
981  IF E$ = "ION" AND I(44) = 1 AND
     JK = 1 THEN  PRINT "YOU'VE C
     HANGED BACK!":JK = 0: GOTO 5
     000

982  IF E$ = "ION" AND I(44) = 1 THEN
     4900
983  IF E$ = "UID" AND I(45) = 1 THEN
      PRINT "YOU'VE CHANGED INTO
     MR. HYDE!","YOU ARE VERY STR
     ONG!":JK = 1: GOTO 5000
```

```
'POUR' command.


985  IF D(B) < > 19 THEN 990
986  IF A = 59 AND E$ = "TER" AND
     I(57) = 1 THEN  PRINT "PLANT
      GROWS TO CEILING.":A$(49) =
     "HUGE PLANT":PT = 1:H$(49) =
     A$(49): GOTO 5000


'PLAY' command.


990  IF D(B) < > 20 THEN 995
991  IF E$ = "UTE" AND A = 61 AND
     I(50) = 1 AND R1 = 0 THEN  PRINT
     "INDIAN ROPE RISES UP TO YOU
     ":R1 = 1:A$(56) = "END OF RO
     PE":H$(56) = A$(56):A(56) =
      - 61: GOTO 5000
993  IF E$ = "ANO" AND A = 63 THEN
      PRINT "LIBERACE YOU'RE NOT!
     ": GOTO 5000


'THROW' command.


995  IF D(B) < > 21 THEN 1000
996  IF E$ = "OPE" AND A = 64 AND
     RS = 0 THEN  PRINT "TIE THE
     OTHER END FIRST.": GOTO 5000

997  IF E$ = "OPE" AND A = 64 AND
     RS = 1 THEN  PRINT "HOOK ON
     ROPE CATCHES OTHER HOOK NEAR
         ENTRANCE AND STRETCHES T
     IGHT.":RS = 2:K3 = 54:M$ = "
     ": GOSUB 1200: GOTO 5000


Various messages.


1000  PRINT "YOU CAN'T DO THAT NO
     W.": GOTO 390
1050  PRINT "3 HOURS PASS.":TM =
     TM + 5: GOTO 390


Subroutines to change, add, or
eliminate items which are carried
or are in current room.  R$ or M$
is set to item (change, add) or to
null (vanish).


1100  FOR K2 = 1 TO 7: IF C$(K2) =
     H$(K3) THEN C$(K2) = R$: GOSUB
      1150: RETURN
1110  NEXT : RETURN
1150  IF R$ = "" THEN I(K3) = 0
1160  RETURN
```

```
1200  FOR K2 = 1 TO 58: IF A$(K2)
       = H$(K3) THEN A$(K2) = M$:H
      $(K3) = M$:A(K2) = A(K2) ¥ (
      M$ < > ""):I(K3) = I(K3) ¥
      (M$ < > ""): RETURN
1210  NEXT : RETURN
1499  GOTO 5000


 Descriptions.  (Continuation from
 line 389 to further describe
 contents of room.)


1500  PRINT "THERE IS A PATH EAST
      , BUT A VICIOUS,    SNARLING
      DOBERMAN BLOCKS YOUR WAY.":
      RETURN
1505  PRINT "AQUARIUM FULL OF PIR
     ANHA FISH.": RETURN
1510  PRINT "THE STAIRCASE IS VER
     Y ROTTEN AND ONE    WOODEN S
     TEP IS MISSING.": RETURN
1515  PRINT "KIDNAPPER COUNTING M
     ONEY. HE DOESN'T    SEE YOU
     YET.": RETURN
1517  PRINT "KIDNAPPER IS SCARED.
     ": RETURN
1520  PRINT "THERE IS A PATH NORT
     H, BUT A RAGING     FIRE BLO
     CKS YOUR WAY.": RETURN
1530  PRINT "A LARGE SWIMMING POO
     L SPANS ACROSS THE  ENTIRE R
     OOM, DIVIDING IT IN HALF. IT
         IS VERY DEEP AND EMPTY."
     : RETURN
1540  PRINT "SWIMMING POOL IS FUL
     L OF WATER.": RETURN
1600  IF A = 16 AND DP = 0 THEN 1
     500
```



continued on next page

continued from previous page

```
1605  IF A = 14 AND PF = 0 THEN 1
      505
1610  IF A = 25 AND SC = 0 THEN 1
      510
1615  IF A = 35 AND PM = 0 THEN 1
      515
1620  IF A = 35 AND PM = 1 THEN 1
      517
1625  IF A = 41 AND FR = 0 THEN 1
      520
1630  IF A = 48 AND SP = 0 THEN 1
      530
1635  IF A = 48 AND SP = 1 THEN 1
      540
1640  RETURN
```

Data statements for items. Strings are read permanently into H$(X) and temporarily into A$(X). The number following each string indicates the room number in which the item is initially placed, and is stored in A(X). A(X) will always correspond with A$(X), where X is the item number. A negative number indicates that an item cannot be removed from the room under any condition, but it could still be altered.

```
2000  DATA PAPER NOTE,2,DESK,-2,C
      HAIR,2,LONG BROOM,3,TV SET,-
      5,CLOCK,-5,GRAFFITI ON WALL,
      -6,OPEN WINDOW,-7,TRAP DOOR
      IN CEILING,-9,DOWN BUTTON,-9
      ,CABINET,-10,ENDS OF TWO 'LI
      VE' WIRES,-11,AQUARIUM,-14,R
      OPE,15,SLEEPING PILL,16,OPEN
       WINDOW,-17
2010  DATA DESK,-17,KEY MAKING MA
      CHINE,-19,SUPER GLUE,20,WOOD
      EN STAIR STEP,20,THICK COPPE
      R SHEET,23,LOCKED DOOR,-24,W
      OODEN STAIRS,-25,TANK OF HEL
      IUM GAS,28,LARGE DEFLATED BA
      LLOON,29
2020  DATA LOCKED DOOR,-30,GUN,30
      ,SIGN ON WALL,-32,STRING VEN
      DING MACHINE,-32,OPEN WINDOW
      ,-33,LOCKED DOOR,-43,BALL OF
       YARN,39,KNITTING NEEDLES,40
      ,PUSH BUTTON ON WALL,-42,"SE
      XY, YOUNG GIRL OFFICE WORKER
      S",-37,STAIRS,-44
2030  DATA PUSH BUTTON ON WALL,-4
      6,BOOK,47,MARY POPPINS UMBRE
      LLA,50,OPEN WINDOW,-51,GRAFF
      ITI ON WALL,-53,SMALL KEY,37
      ,STEEL DOOR,-55,JAR OF YELLO
      W SOLUTION,56,JAR OF FLUID,5
      6,CHILD'S SLIDE,-57
2040  DATA WATER COOLER,-58,PAPER
      CUP,58,SMALL-SIZED PLANT,-5
      9,FLUTE,61,STAIRS,-62,SMALL
      BOOK,64,PIANO,-63,LONG ROPE,
      63,FRONT DOOR,-65
```

Data statements for commands. Commands are read into B$(X). B$(1)-B$(12) hold all possible directions and shorthand notation, allowing for player to input entire direction or just the initial letter. Starting with B$(13) all commands have a corresponding code number, read into D(X), which picks the proper action--allowing the use of synonyms for the same command.

```
2100  DATA WEST,W,NORTH,N,EAST,E,
      SOUTH,S,UP,U,DOWN,D,GET,TAKE
      ,DROP,PUT,GIVE,PAY,CLIMB,GO,
      ENTER,OPEN,READ,CHECK,WATCH,
      PRESS,PUSH,TAPE,TIE,LIGHT,MA
      KE,GLUE,INFLATE,SHOOT,KNIT,D
      RINK,POUR,PLAY,THROW
2110  DATA 1,1,2,2,2,2,3,3,3,4,5,
      5,5,6,6,9,9,10,11,12,13,14,1
      5,18,19,20,21
```

Short subroutines.

```
3500  FOR K4 = 1 TO 58: IF A$(J) =
      H$(K4) THEN I(K4) = 1:K4 = 6
      0
3600  NEXT : RETURN
4000  IF A = 24 THEN X = 22: GOTO
      4100
4010  IF A = 43 THEN X = 31: GOTO
      4100
4020  IF A = 55 THEN X = 43: GOTO
      4100
4030 X = 55
4100 A$(X) = "OPEN DOOR": RETURN
```

Timing loop. This is the only delay in the program, and is used to give the player time to read messages.

```
4900  PRINT "OK!"
5000  IF E$ = "DOR" AND A = 71 THEN
      PRINT "YOU NEED A KEY."
5100  FOR V = 1 TO 2000: NEXT : GOTO
      300
5200  FOR V = 1 TO 3500: NEXT : GOTO
      300
```

You win!

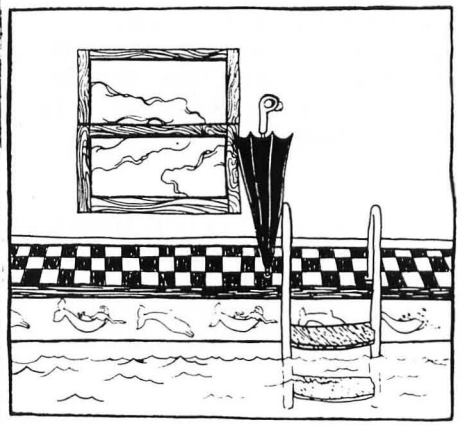
```
6000  PRINT "OUT.  YOU'VE MADE IT
      !"
6010  END
```

You're dead.

```
7000  PRINT "AAHHHHHHHHHHHHHHHHHH
      HHHH...."
7010  PRINT "YOU DIE A HORRIBLE A
      ND GRIZZLY DEATH,   YOUR BOD
      Y TORN TO SHREDS.": GOTO 730
      0
7100  PRINT "YOU NOW LOOK LIKE A
      PANCAKE!"
7200  PRINT "YOU SEEM TO HAVE GOT
      TEN YOURSELF KILLED!";
7300  PRINT : PRINT "TO TRY AGAIN
      , YOU'LL HAVE TO START OVER
      FROM THE 9TH FLOOR."
7400  END
7500  IF  LEFT$ (A$,4) = "LIGH" AND
       RIGHT$ (A$,4) = "IGHT" THEN
       GOTO 391
7510  PRINT "YOU CAN'T SEE WHERE
      YOU'RE GOING...     OOPS!  S
      ORRY, YOU FELL AND BROKE YOU
      R   NECK!"
7520  GOTO 7200
```

Drop down to the next floor. You lose any item you were carrying. C$(X) is cleared, as is part of I(X).

```
8000 I(38) = 0:G(1) = 0: IF JK =
      1 THEN  PRINT "YOU FORGOT TH
      E ANTIDOTE! FLUID POISONS  Y
      OU.": GOTO 7200
8010 BR = 0:FL = FL - 1: PRINT "Y
      OU MADE IT DOWN TO THE NEXT
      FLOOR,     BUT YOU HAD TO DR
      OP ANYTHING YOU WERE   CARRY
      ING."
8020  FOR G = 1 TO 7:C$(G) = "": NEXT
      : FOR G = 56 TO 58:I(G) = 0:
       NEXT
8030  GOTO 5200
```



Atari Version

Notes on the Atari version:

Underlined characters in the listing should be typed using the Atari logo key to produce in reverse video.

Since the Atari does not support string arrays, this version uses single (very large) strings to simulate arrays. This is done by making all sub-strings the same length, appending spaces to fill them out. Each item is then accessed by its position within the larger string.

This method results in some changes in the way the program looks at the user's input. The original and Apple versions of "Kidnapped" look at the last three characters of the input to determine what the player wants to do. This approach won't work, however, when the object's name may be padded with spaces in the computer's memory. So in the Atari version, when you are using the GET and PUT commands, you must type the FIRST three letters of the object. For other commands you must type its LAST three letters. If you want to pick up a "SHINY MIRROR" you must say "GET SHI"; to look at it you must say "LOOK ROR". Of course, you may always type more than three characters, as long as the three needed ones are there.

DOCUMENTATION

Lines 10-70: Introduce the program and the programmers.
Lines 110-120: Print start-up display.
Lines 140-790: Room descriptions. Line 140 checks to see if you have escaped from the first floor, otherwise line 142 computes which line should be accessed to print out the room description, based on the number of the room you are in (variable A).
Lines 800-820: Initialization. DIMensions strings and array, reads in string data, and clears numerical arrays. Lines 830-1120: Description of current location. The program first branches to line 140 to print out the room description. It then prints the possible exits, the contents of the room, and any special messages that are to be displayed for this location.
Lines 1130-1286: Player input routine. The TRAP 9000 is used to prevent an error if the user enters an empty string for a command. After a command is entered, this routine will break it up into the required pieces (A1$, D$, E$, etc.).
Line 1190-1220: Check to see if a movement command was entered. If so, it is then checked for legality and operated upon.

Lines 1230-1250: Inventory routine. Prints out the contents of the simulated array C$.
Lines 1260-1270: Check to see if the command had a legitimate verb in it by comparing the first word of the inputted string to the simulated array B$.
Lines 1290-1490: "GET" command. Checks if the specified object is dangerous, immobile, or legal to be picked up. If legal, then the item is added to the array of carried data (C$) and the value of I (object #) is set to 1.
Line 1500-1600: "DROP" command. If the specified item is being carried, it is dropped into the current room. I (object #) is set to the room number, and any special conditions caused by the DROP are carried out.
Lines 1610-1800: "GO" command. If the proper conditions for the GO are met, the player goes where requested. (You can't travel through locked doors, etc.) Variable A is then set to the new room number.

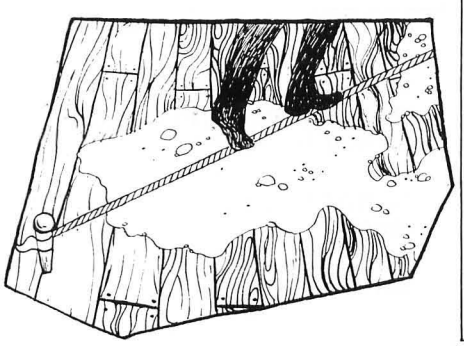
Lines 1810-1890: "OPEN" command. Checks for correct conditions to perform open and does so if met (i.e., a key is needed to open a locked door,etc.)
Lines 1900-1980: "READ" command.
Lines 1990-2040: "PUSH" command.
Lines 2050-2100: "JUMP" command.
Lines 2110-2200: "TIE" command.
Lines 2210-2312: "LIGHT", "MAKE", "GLUE", "INFLATE", "SHOOT", and "KNIT" commands.
Lines 2320-2350: "SWIM" command.
Lines 2360-2390: "DRINK" command.
Lines 2400-2415: "POUR" command.
Lines 2420-2440: "PLAY" command.
Lines 2450-2472: "THROW" command.
Lines 2480-2490: Various messages.
Lines 2500-2560: Subroutines to change, add or eliminate items which are carried or are in the current room. R$ and M$ are set to item (change, add) or to null (vanish).
Lines 2570-2730: Descriptions (continuation from line 1130 to further describe contents of room).
Lines 2740-2788: Data statements for items. Strings are read permanently into H$ and temporarily into A$. The number following each string is the room number in which the object is originally placed, and is stored in array A. A negative number indicates that an item cannot be removed from the room under any condition, but it could still be altered.
Lines 2790-2800: Data statements for commands. Commands are read into B$. The first twelve strings hold all possible directions and shorthand notation, allowing for player to input the entire direction or just the initial letter. Starting with the 13th string all commands have a corresponding number, read into D(X), which picks the proper action — allowing the use of synonyms for the same command.
Lines 2810-2870: Short subroutines.
Lines 2880-2910: Delay routine. Used to give the player time to read messages.
Lines 2920-2930: You Win!!!!
Lines 2940-3020: You're dead.
Lines 3030-3060: Drop down to the next floor. You lose any items you were carrying, so C$ is cleared, as is part of array I.
Line 9000: TRAP routine to prevent an error when a null string command is entered.

```
10 REM KIDNAPPED
20 REM
30 REM ORIGINAL PROGRAM BY:
40 REM PETER KIRSCH
50 REM ATARI TRANSLATION BY:
60 REM RICH BOUCHARD & ALAN J. ZETT
70 REM
100 SA=22:SH=SA:SB=7:SC=SA
110 GRAPHICS 2+16:SETCOLOR 4,8,0:POSIT
ION 6,3:PRINT #6;"kidnapped":POSITION
3,8:PRINT #6;"can you escape?"
115 FOR A=6 TO 14:POSITION A,2:PRINT #
6;CHR$(95);:POSITION A,4:PRINT #6;CHR$
(95);:NEXT A
120 GOTO 800
140 IF A=66 THEN 2920
142 GOTO (A$10+140)
150 PRINT J$:W=4:N=2:E=8:S=7:GOTO 860
160 PRINT N$:N=3:S=1:GOTO 860
170 PRINT "in a closet.":S=2:GOTO 860
180 PRINT J$:N=5:E=1:S=6:GOTO 860
190 PRINT N$:S=4:GOTO 860
200 PRINT "in a restroom.":N=4:GOTO 86
0
210 PRINT N$:N=1:GOTO 860
220 PRINT J$:W=1:N=9:S=10:GOTO 860
230 PRINT "in an elevator.":S=8:GOTO 8
60
240 PRINT "in a maintenance room.":N=8
:GOTO 860
250 PRINT "in a crawlspace atop the":P
RINT "elevator":D=9:GOTO 860
```

continued on next page

continued from previous page

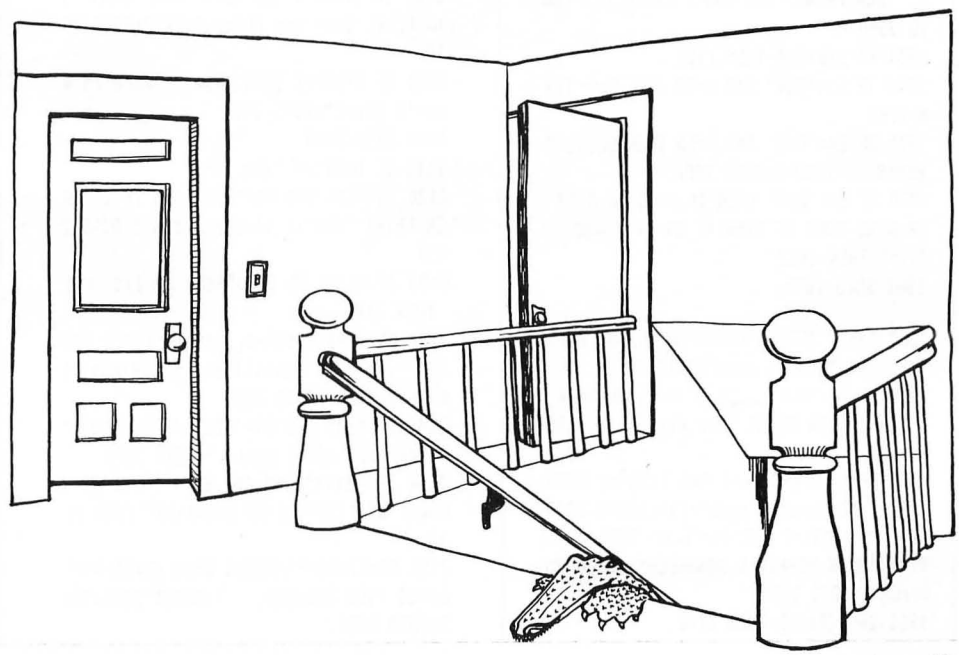
```
260 PRINT "on a very narrow ledge.":GO
TO 860
270 PRINT J$;W=14:S=16:GOTO 860
280 PRINT "in a visitor's lounge.":W=1
5:E=13:GOTO 860
290 PRINT "in a closet.":E=14:GOTO 860
300 PRINT N$;N=13:E=17:GOTO 860
310 PRINT N$;W=16:GOTO 860
320 PRINT J$;W=20:N=19:E=24:S=21:GOTO
860
330 PRINT N$;S=18:GOTO 860
340 PRINT "in a tool crib.":E=18:GOTO
860
350 PRINT J$;N=18:E=22:GOTO 860
360 PRINT N$;W=21:E=23:GOTO 860
370 PRINT "in a small storage room.":W
=22:GOTO 860
380 PRINT J$;W=18:GOTO 860
390 PRINT "in a narrow stairway.":W=24
:GOTO 860
400 PRINT J$;W=31:N=27:E=29:S=30:GOTO
860
410 PRINT N$;N=28:S=26:GOTO 860
420 PRINT "in a closet.":S=27:GOTO 860
430 PRINT "in a game room.":W=26:GOTO
860
440 PRINT N$;N=26:GOTO 860
450 PRINT J$;W=33:N=32:E=26:GOTO 860
460 PRINT "in a store room.":S=31:GOTO
 860
470 PRINT N$;E=31:GOTO 860
480 PRINT "on a large ledge by the win
dow.":W=36:GOTO 860
490 PRINT "in the vault.":W=36:GOTO 86
0
500 PRINT "on a large ledge by the win
dow.":E=34:GOTO 860
510 PRINT N$;S=41:GOTO 860
520 PRINT J$;E=43:N=39:S=40:GOTO 860
530 PRINT N$;S=38:GOTO 860
540 PRINT N$;N=38:GOTO 860
550 PRINT J$;W=43:N=37:S=42:GOTO 860
560 PRINT N$;N=41:GOTO 860
570 PRINT J$;W=38:E=41:GOTO 860
580 PRINT "in a staircase.":GOTO 860
590 PRINT J$;N=46:E=48:S=47:GOTO 860
600 PRINT N$;S=45:GOTO 860
610 PRINT "in a library.":N=45:GOTO 86
0
620 PRINT "in the swim room";W=45:GOTO
 860
630 PRINT "in the swim room":N=50:E=51
:GOTO 860
```



```
640 PRINT "in a laundry room.":S=49:GO
TO 860
650 PRINT "in a dining room.":W=49:GOT
O 860
660 PRINT J$;N=53:E=54:GOTO 860
670 PRINT "in a restroom.":S=52:GOTO 8
60
680 PRINT J$;W=52:E=55:GOTO 860
690 PRINT J$;W=54:N=56:GOTO 860
700 PRINT "in a laboratory.":S=55:GOTO
 860
710 PRINT "in a child's playroom.":N=5
5:GOTO 860
720 PRINT N$;W=59:GOTO 860
730 PRINT "in a lounge.":E=58:GOTO 860
740 PRINT "in a crawlway.":E=61:GOTO 8
60
750 PRINT "in a crawlway.":W=60:GOTO 8
60
760 GOTO 580
770 PRINT "in a lounge.":E=64:GOTO 860
780 PRINT J$;W=63:GOTO 860
790 PRINT "in the entrance hall.":GOTO
 860
800 DIM A$(58*SA),A(58),B(12),B$(39*SB
),C$(7*SC),D(39),H$(58*SH),I(58),Z$(SA
),J$(50),A1$(50),N$(50),A9$(50)
802 DIM A2$(50),E$(25),D$(25),D1$(25),
H(7),R$(50),G(9),M$(50)
805 FOR A=1 TO 7*SC:C$(A,A)=" ":NEXT A
810 FOR A=1 TO 58:READ Z$,Z:A(A)=Z:A$(
A*SA-SA+1,A*SA)="
 ":A$(A*SA-SA+1,A*SA)=Z$:NEXT A
812 H$=A$
814 FOR A=1 TO 39:READ Z$:B$(A*SB-SB+1
,A*SB)="        ":B$(A*SB-SB+1,A*SB)=Z$
:NEXT A
815 FOR A=13 TO 39:READ Z:D(A)=Z:NEXT
A
816 FOR A=1 TO 58:I(A)=0
817 IF A<=12 THEN B(A)=0
818 IF A<=9 THEN G(A)=0
819 NEXT A
820 A=1:G=1:FL=9:TM=-2:J$="in a hallwa
y.":N$="in an office."
830 N=0:W=0:E=0:S=0:U=0:D=0:Y=0:GRAPHI
CS 0:PRINT "You're ";:IF DK=0 AND TM>2
4 THEN DT=1
840 IF DT<>1 THEN 140
850 GRAPHICS 0:PRINT CHR$(253);"POWER
FAILURE!  It's too dark to see!":DT=2:
GOTO 1140
860 PRINT "Floor ";FL;"     Some exits
 are:"
870 FOR B=1 TO 12:B(B)=0:NEXT B
880 IF W<>0 THEN PRINT " West";:B(1)=W
:B(2)=W
890 IF A=41 AND FR=0 THEN 930
900 IF N<>0 THEN PRINT " North";:B(3)=
N:B(4)=N
910 IF A=16 AND DP=0 THEN 960
920 IF E<>0 THEN PRINT " East";:B(5)=E
:B(6)=E
930 IF S<>0 THEN PRINT " South";:B(7)=
S:B(8)=S
940 IF U<>0 THEN PRINT " Up";:B(9)=U:B
(10)=U
950 IF D<>0 THEN PRINT " Down";:B(11)=
D:B(12)=D
960 PRINT :PRINT
970 FOR B=1 TO 58:IF A=ABS(A(B)) THEN
C=C+1:GOTO 990
980 NEXT B:PRINT :GOTO 1010
990 IF C<2 THEN PRINT "Things you see
here:"
1000 PRINT " ";A$(B*SA-SA+1,B*SA):GOTO
 980
1010 IF A=37 AND FR=1 THEN PRINT "You
forgot you were naked.":PRINT "You blu
sh and run out.":A=41:GOTO 2890
1020 IF G=1 THEN PRINT "You awaken on
the 9th floor of a      strange buildi
ng, obviously a kidnap"
1025 IF G=1 THEN PRINT "victim.  You a
re alone at the moment  and must escap
e from the building,"
1027 IF G=1 THEN PRINT "floor by floor
.":G=0
1030 IF A=7 THEN PRINT "You see a smal
l ledge outside the     window"
1035 IF A=7 AND KY=0 THEN PRINT "and a
 single key on a key chain there"
1040 IF RS=2 THEN IF A=64 OR A=65 THEN
 PRINT "Rope is stretched across quick
sand."
1050 IF A=33 THEN PRINT "There is a wi
de, long ledge outside   the window."
1060 IF A=56 AND JK=0 THEN PRINT "Labe
ls on bottles:":PRINT "   Solution: An
tidote":PRINT "   Fluid: Undecipherabl
e"
1070 IF A=58 THEN PRINT "A drooling al
ligator blocks your way  east.  He has
 the remains of a kid-"
1075 IF A=58 THEN PRINT "napper in his
 mouth.  You catch a     glimpse of a
staircase past the       alligator."
1080 IF A<>64 OR RS>=2 THEN 1090
1082 PRINT "A huge bog of quicksand bl
ocks your   way east. The front entran
ce is there,";
1084 PRINT "your way to safety.  There
 is a large hook on the other side and
 a tent     stake on this side."
1090 IF A=59 THEN PRINT "There is a tr
ap door above you."
1100 IF A=61 AND R1=0 THEN PRINT "Thro
ugh the trap door you see a long  coil
ed rope on the floor below."
1110 IF A=10 AND C1=1 AND CF=0 THEN PR
INT "A flashlight is there."
1120 IF A=10 AND C1=1 AND CT=0 THEN PR
INT "Electrical tape is there."
1130 GOSUB 2650
1140 TRAP 9000:C=0:TM=TM+1:PRINT :PRIN
T "COMMAND ";:INPUT A9$:IF DT=2 THEN 3
000
1145 A2$="":FOR T=1 TO LEN(A9$):IF A9$
(T,T)<>" " THEN A2$(LEN(A2$)+1)=A9$(T,
T):NEXT T
1148 A1$=A2$
1149 IF LEN(A1$)<SB THEN A1$(LEN(A1$)+
1)=" ":GOTO 1149
```

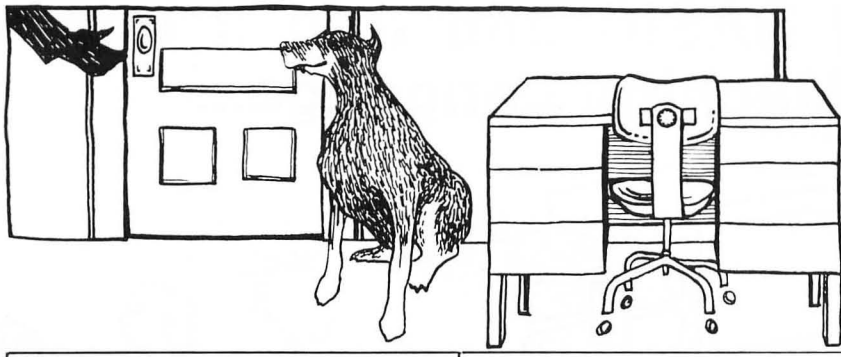
```
1150 IF A9$="LOOK" THEN 830
1160 IF A9$="JUMP" THEN 2050
1170 IF A9$="SWIM" THEN 2320
1180 IF A9$="WAIT" THEN 2490
1190 PRINT :FOR B=1 TO 12:IF A1$=B$(B%
SB-SB+1,B%SB) THEN 1210
1200 NEXT B:GOTO 1230
1210 IF B(B)<>0 THEN A=B(B):GOTO 830
1220 PRINT "You can't go that way.":GO
TO 1140
1230 IF A9$<>"I" AND A9$<>"INVENTORY"
THEN 1260
1240 PRINT "You are carrying:":FOR K=1
 TO 7:PRINT C$(K%SC-SC+1,K%SC)
1250 NEXT K:GOTO 1140
1260 FOR B=13 TO 39:IF A1$=B$(B%SB-SB+
1,B%SB) THEN 1280
1270 NEXT B:PRINT "Don't know what ";C
HR$(34);A9$;CHR$(34);" means.":GOTO 11
40
1280 D$="":IF LEN(A9$)>=LEN(A2$)+2 THE
N D$=A9$(LEN(A2$)+2):IF LEN(D$)>=3 THE
N E$=D$(LEN(D$)-2,LEN(D$)):GOTO 1287
1285 IF LEN(D$)<3 THEN D$(LEN(D$)+1)="
 ":GOTO 1285
1286 E$=D$
1287 REM
1290 IF D(B)<>1 THEN 1500
1300 IF D$(1,3)<>"FLA" AND D$(1,3)<>"T
AP" AND A<>10 THEN 1340
1310 IF A=10 AND C1=0 THEN PRINT "Cabi
net is locked.":GOTO 2890
1320 IF D$(1,3)<>"FLA" OR CF<>0 THEN 1
330
1323 J=56:A$(J%SA-SA+1,J%SA)="FLASHLIG
HT             ":H$(J%SA-SA+1,J%SA)=A$(
J%SA-SA+1,J%SA):A(J)=10:CF=1
1330 IF D$(1,3)="ELE" OR CT<>0 THEN 13
40
1333 J=57:A$(J%SA-SA+1,J%SA)="ELECTRIC
AL TAPE        ":H$(J%SA-SA+1,J%SA)=A$(
J%SA-SA+1,J%SA):A(J)=10:CT=1
1340 IF A=7 AND KY=0 AND E$="KEY" AND
I(4)<>1 THEN PRINT "Your arm is too sh
ort to reach it.":GOTO 2890
1350 IF A<>7 OR KY<>0 OR E$<>"KEY" THE
N 1360
1353 J=58:A$(J%SA-SA+1,J%SA)="KEY
              ":H$(J%SA-SA+1,J%SA)=A$(
J%SA-SA+1,J%SA):A(J)=7:I(58)=1:KY=1
1360 IF A<>63 OR E$<>"KEY" THEN 1370
1363 PRINT "We all know pianos have ke
ys.":J=56:A$(J%SA-SA+1,J%SA)="KEY
                "
1365 H$(J%SA-SA+1,J%SA)=A$(J%SA-SA+1,J
%SA):A(J)=63:I(56)=1
1370 IF (A=14 AND E$="PIR" AND PF=0) O
R (A=17 AND E$="DOB") OR (A=58 AND E$=
"ALI") THEN 2940
1380 IF A<>35 OR (D$(1,3)<>"KID" AND D
$(1,3)<>"DOL" AND D$(1,3)<>"MON") OR I
(27)<>0 THEN 1390
1383 PRINT "You get too close to him.
 He jumps upand strangles you.":GOTO 2
980
1390 IF A<>35 OR (D$(1,3)<>"KID" AND D
$(1,3)<>"DOL" AND D$(1,3)<>"MON") THEN
 1400
1393 PRINT "Kidnapper sees your gun an
d freezes.  You grab a dollar.":A$(56%
SA-SA+1,56%SA)="DOLLAR
"
1395 H$(56%SA-SA+1,56%SA)=A$(56%SA-SA+
1,56%SA):A(56)=35:PM=1:E$="LAR"
1400 IF I(48)<>1 OR D$(1,3)<>"WAT" OR
A<>58 THEN 1410
1403 R$="CUP OF WATER":K3=48:GOSUB 250
0:H$(48%SA-SA+1,48%SA)="CUP OF WATER
              ":I(57)=1:GOTO 2880
1410 REM
1420 FOR J=1 TO 58:IF D$(1,3)=A$(J%SA-
SA+1,J%SA-SA+3) AND A=ABS(A(J)) THEN 1
440
1430 NEXT J:PRINT "There's no ";D$;" h
ere.":GOTO 1140
1440 IF A(J)<0 THEN PRINT "Be reasonab
le now.  That's impossible.":GOTO 1140
1450 IF D$(1,3)<>"BOO" THEN 1480
1460 PRINT "Title of book: How to ";:I
F A=47 THEN PRINT "Swim"
1470 IF A=64 THEN PRINT "Walk a tightr
ope"
1480 PRINT "OK!":FOR K=1 TO 7:IF C$(K%
SC-SC+1,K%SC)<>"
" THEN 1490
1482 C$(K%SC-SC+1,K%SC)=A$(J%SA-SA+1,J
%SA):GOSUB 2810:H(K)=J:A$(J%SA-SA+1,J%
SA)="                    "
1485 A(J)=0:GOTO 2890
1490 NEXT K
1500 IF D(B)<>2 THEN 1610
1510 FOR J=1 TO 7:IF D$(1,3)=C$(J%SC-S
C+1,J%SC-SC+3) THEN 1530
1520 NEXT J:PRINT "You're not carrying
 it.":GOTO 1140
1530 FOR K=1 TO 58:IF A$(K%SA-SA+1,K%S
A)<>"                    " THEN 1540
1532 A$(K%SA-SA+1,K%SA)=C$(J%SA-SA+1,J
%SA):A(K)=A:H(J)=0:GOTO 1550
1540 NEXT K
1550 PRINT "OK!":FOR K1=1 TO 58:IF C$(
J%SC-SC+1,J%SC)<>H$(K1%SC-SC+1,K1%SC)
THEN 1560
1552 C$(J%SC-SC+1,J%SC)="
        ":GOTO 1570
1560 NEXT K1
1570 IF A<>14 OR D$(1,3)<>"SLE" THEN 1
580
1572 PRINT "You drop pill in aquarium.
":PF=1:K3=15:M$="SLEEPING PIRANHA":GOS
UB 2540:GOTO 2890
1580 IF A<>16 OR DP<>0 OR D$(1,5)<>"SL
E" THEN 1590
1582 PRINT "Piranha devours Doberman p
incher, thendies of overeating.":DP=1:
K3=15
1585 M$="DEAD PIRANHA          ":GOSUB
 2540:K1=15:A(57)=-16:A$(56%SA+1,57%SA
)="EATEN DOBERMAN        "
1587 H$(56%SH+1,57%SH)=A$(56%SA+1,57%S
A):GOTO 1600
1590 IF A<>32 OR E$<>"LAR" THEN 1600
1592 A$(56%SA+1,57%SA)="LONG STRING
        ":H$(56%SH+1,57%SA)=A$(56%SA+1
,57%SA):A(57)=32:K1=57:K3=56
1595 M$="":GOSUB 2540
1600 I(K1)=A:GOTO 2890
1610 IF D(B)<>3 THEN 1810
1620 IF A=16 AND E$="MAN" OR A=58 AND
E$="TOR" THEN 2940
1630 IF A=41 AND E$="IRE" OR A=64 AND
E$="AND" THEN 2970
1640 IF A=25 AND E$="IRS" AND SC=0 THE
N PRINT "Step is missing!  You fell do
wn the  stairs.":GOTO 2970
1650 IF E$="OPE" AND R1=1 AND A=17 THE
N A=18:R1=0:GOTO 3030
```

continued on next page

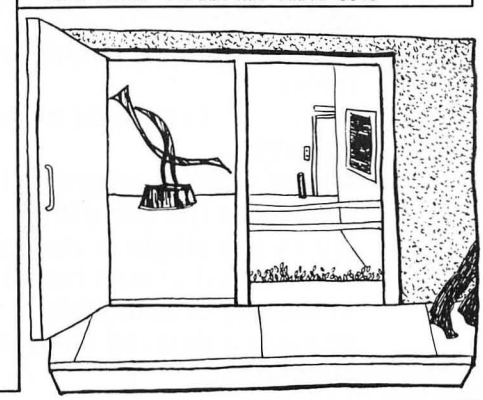


continued from previous page

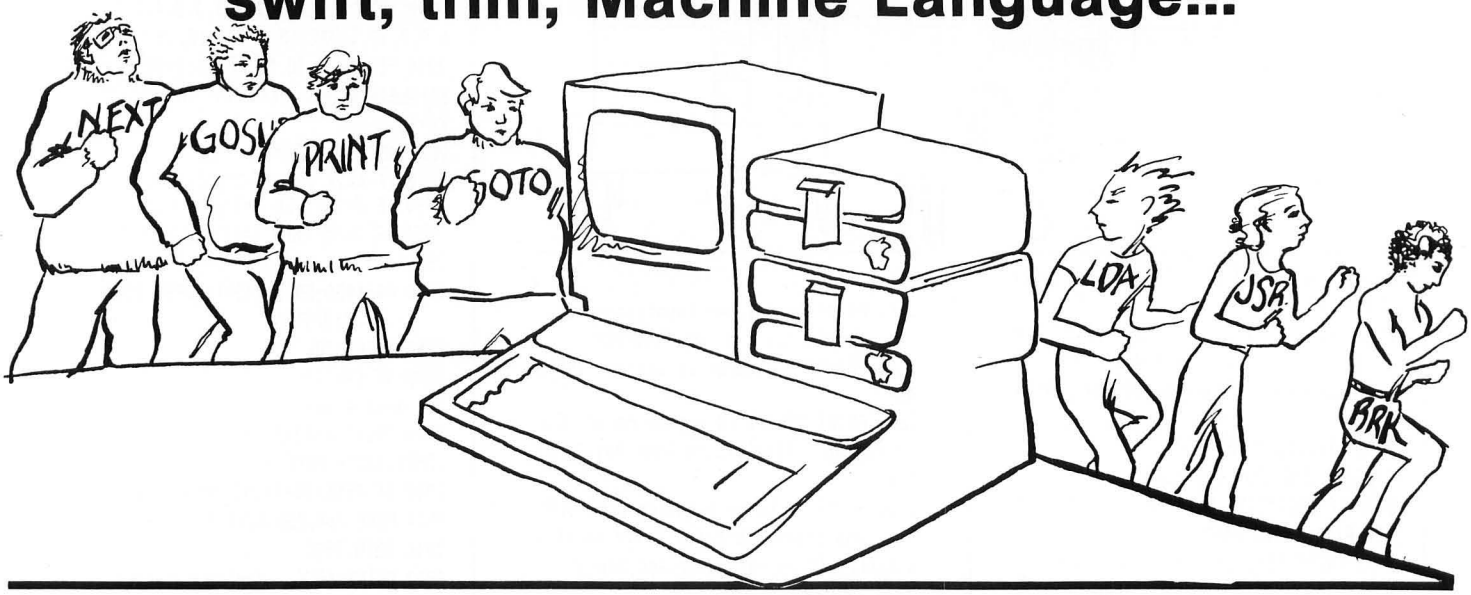
```
1660 IF E$="OPE" AND R1=1 AND A=61 THE
N A=62:R1=0:GOTO 830
1670 IF E$="ANT" AND PT=1 AND A=59 THE
N A=60:GOTO 830
1680 IF E$="ANT" AND PT=1 AND A=60 THE
N A=59:GOTO 830
1690 IF E$="OPE" AND A=65 THEN A=64:GO
TO 830
1700 IF E$="OPE" AND A=64 AND RS=2 AND
 BR=1 THEN A=65:GOTO 830
1710 IF E$="OPE" AND A=64 AND RS=2 THE
N PRINT "You don't know how to walk a
tightrope";:GOTO 2970
1720 IF E$="DOW" THEN IF A=34 OR A=36
THEN A=A-1:GOTO 830
1730 IF E$="DGE" AND A=7 THEN PRINT "L
edge breaks!":GOTO 2970
1740 IF E$="DGE" THEN IF A=33 OR A=35
THEN A=A+1:GOTO 830
1750 IF E$="IRS" THEN IF A=25 OR A=44
OR A=62 THEN A=A+1:GOTO 3030
1760 IF E$<>"OOR" OR (A<>24 AND A<>43
AND A<>30 AND A<>65) OR G((A/10-INT(A/
10))*10)<>0 THEN 1770
1762 PRINT "The door's locked.":GOTO 2
890
1770 IF E$="OOR" THEN IF A=24 OR A=43
OR A=30 OR A=65 THEN A=A+1:GOTO 830
1780 IF E$="OOR" AND A=55 THEN IF SD=1
 OR JK=1 THEN A=57:GOTO 830
1790 IF E$="OOR" AND A=55 THEN PRINT "
It's stuck!  You're not strong enough
to open it!":GOTO 2890
1800 IF E$="IDE" AND A=57 THEN PRINT "
You slide through hole in floor.":A=58
:GOTO 3030
1810 IF A<>9 OR (E$<>"OOR" AND E$<>"AI
R") OR (A<>I(3)) THEN 1820
1812 PRINT "You step on chair and just
 manage to  reach the trap door.":A=11
:GOTO 2890
1820 IF A=9 THEN IF E$="OOR" OR E$="AI
R" THEN PRINT "You can't reach it.":GO
TO 2890
1830 IF D(B)<>4 THEN 1900
1840 IF E$="OOR" AND A=55 AND JK=0 THE
N 1790
1850 IF E$="OOR" AND A=55 THEN GOSUB 2
830:G(1)=1:SD=1:GOTO 2880
1860 IF E$="OOR" THEN IF A=43 OR A=24
OR A=65 THEN IF I(56)=1 OR A=43 AND I(
42)=1 THEN 1862
1861 GOTO 1870

1862 GOSUB 2830:G((A/10-INT(A/10))*10)
=1:PRINT "Door's open":GOTO 2890
1870 IF E$="OOR" THEN IF A=43 OR A=24
OR A=65 THEN PRINT "You need a key":GO
TO 2890
1880 IF A=10 AND C1=0 AND I(58)=1 THEN
 PRINT "Cabinet's open":C1=1:GOTO 2890
1890 IF I(39)=1 AND E$="LLA" THEN K3=3
9:R$="OPEN POPPIN'S UMBRELLA":GOSUB 25
00:UM=1:GOTO 2880
1900 IF D(B)<>5 THEN 1990
1910 IF E$="ITI" AND A=6 THEN PRINT "I
t says:  Watch out for live ones.":GOT
O 2890
1920 IF E$="ITI" AND A=53 THEN PRINT "
GRAFFITI:  Do you have a split
personality?":GOTO 2890
1930 IF E$="OTE" THEN IF I(1)=A OR I(1
)=1 OR A=2 AND I(1)=0 THEN PRINT "Note
 says:  'Important TV program on'":GOT
O 1140
1940 IF E$="IGN" AND A=32 THEN PRINT "
Sign:  String costs $1.00":GOTO 2890
1950 IF E$="TV" AND A=5 THEN PRINT "Bu
lletin:":PRINT "Power will be shut off
 at midnight...":GOTO 2890
1960 IF E$="OCK" AND A=5 THEN PRINT "T
ime:  ";:TI=INT(TM/2):IF T1>12 THEN PR
INT "Past midnight":GOTO 2890
1970 IF E$="OCK" AND A=5 THEN PRINT TI
;" P.M.":GOTO 2890
1980 IF E$="OOK" THEN IF I(38)=1 OR I(
52)=1 THEN BR=1:GOTO 2880
1990 IF D(B)<>6 THEN 2110
2000 IF E$<>"TON" THEN 2480
2010 IF A=9 AND EF=1 THEN PRINT "Eleva
tor goes down!":A=13:GOTO 3030
2020 IF A<>42 OR FR<>0 THEN 2030
2022 PRINT "Sprinkler turned on.  Fire
 out.":PRINT "But your clothes are ver
y wet.  You"
2025 PRINT "take them off.  You are no
w naked.":FR=1:A$(56*SA+1,57*SA)="WET
CLOTHES            "
2028 H$(56*SA+1,57*SH)=A$(56*SA+1,57*S
A):A(57)=-42:GOTO 2910
2030 IF A=46 AND SP=0 THEN PRINT "Clic
k!":SP=1:GOTO 2890
2040 PRINT "Nothing happens.":GOTO 289
0
2050 IF A=60 OR A=61 OR A=64 THEN 2960
2060 IF A=34 OR A=36 THEN IF I(25)=1 A
ND BO=2 THEN PRINT "Balloon carries yo
u down one floor.":A=43:GOTO 3030
2070 IF A=34 OR A=36 THEN 2960
2080 IF I(39)=1 AND UM=1 THEN PRINT "Y
ou float down one floor.":A=52:GOTO 30
30
2090 IF I(39)=1 THEN PRINT "Umbrella w
asn't open!":GOTO 2960
2100 GOTO 2040
2110 IF D(B)<>9 THEN 2190
2120 IF A=11 AND E$="RES" AND TM<23 TH
EN PRINT "You're electrocuted!":GOTO 2
970
2130 IF A<>11 OR E$<>"RES" OR I(57)<>1
 THEN 2140
2132 EF=1:A$(11*SA+1,12*SA)="TAPED WIR
ES            ":H$(11*SH+1,12*SH)=A$(11
*SA+1,12*SA):GOTO 2880
2140 IF A=11 AND E$="RES" THEN PRINT "
Wires fall apart again.":GOTO 2890
2150 IF (E$<>"OON" AND E$<>"ING") OR (
BO<>1 OR I(25)<>1 OR I(57)<>1) THEN 21
60
2152 K3=57:R$="":GOSUB 2500:K3=25:R$="
LARGE TIED BALLOON     ":GOSUB 2500:BO=
2:GOTO 2880
2160 IF E$<>"OPE" THEN 2190
2170 IF A<>64 OR I(54)<>1 THEN 2180
2172 K3=54:R$="":GOSUB 2500:A$(53*SA+1
,54*SA)="ROPE TIED TO STAKE    ":H$(53
*SH+1,54*SH)=A$(53*SA+1,54*SA)
2175 A(54)=-64:RS=1:GOTO 2880
2180 IF (A<>17 OR I(14)<>1) THEN 2190
2182 PRINT "Rope tied to desk":K3=14:R
$="":GOSUB 2500:A$(7*SA+1,8*SA)="ROPE
HANGING OUT WINDO":R1=1
2185 H$(7*SA+1,8*SA)=A$(7*SA+1,8*SA):A
(8)=-17:GOTO 2890
2190 IF D(B)<>10 THEN 2210
2195 IF A>=13 OR E$<>"GHT" OR I(56)<>1
 THEN 2210
2200 K3=56:R$="LIT FLASHLIGHT":GOSUB 2
500:H$(55*SA+1,56*SA)="LIT FLASHLIGHT
      ":DK=1:GOTO 2880
2210 IF D(B)<>11 THEN 2230
2220 IF A<>19 OR E$<>"KEY" OR I(21)<>1
 THEN 2230
2222 A$(55*SA+1,56*SA)="CRUDE KEY
       ":H$(55*SH+1,56*SH)=A$(55*SA+1
,56*SA):A(56)=19:GOTO 2880
2230 IF D(B)<>12 THEN 2250
2240 IF (A<>25 OR I(19)<>1) OR (I(20)<
>1 AND A<>I(20)) THEN 2250
2242 IF E$="TEP" OR E$="IRS" THEN PRIN
T "Stair's fixed":S1=1:R$="":K3=20:GOS
UB 2500:M$="":GOSUB 2540:GOTO 2890
2250 IF D(B)<>13 THEN 2270
2260 IF (I(25)<>1) OR (I(24)<>1 AND A<
>I(26)) THEN 2270
2262 IF E$="OON" THEN K3=25:R$="LARGE
INFLATED BALLOON":GOSUB 2500:H$(24*SH+
1,25*SA)="LARGE INFLATED BALLOON"
2265 BO=1:GOTO 2880
2270 IF D(B)<>14 THEN 2300
2280 IF E$="GUN" OR E$="PER" THEN IF I
(27)=1 AND A=35 THEN PRINT "Gun had bl
anks!":PRINT "Kidnapper shoots you!":G
OTO 2970
2290 IF E$="GUN" OR E$="PER" THEN IF I
(27)=1 THEN PRINT "Gun misfires.":GOTO
 2890
2300 IF D(B)<>15 THEN 2360
2310 IF FR<>1 OR I(32)<>1 OR I(33)<>1
OR E$<>"HES" THEN 2320
2312 PRINT "You have knitted a fine su
it and are  wearing it.":FR=2:GOTO 289
0

2320 IF SP=0 THEN 2480
2330 IF A=49 THEN A=48:GOTO 830
2340 IF A=48 AND BR=0 THEN PRINT "You
don't know how to swim!":GOTO 2970
2350 IF A=48 THEN A=49:GOTO 2880
2360 IF D(B)<>18 THEN 2400
2370 IF E$="ION" AND I(44)=1 AND JK=1
THEN PRINT "You've changed back!":JK=0
:GOTO 2890
2380 IF E$="ION" AND I(44)=1 THEN 2880
2390 IF E$="UID" AND I(45)=1 THEN PRIN
T "You've changed into Mr. Hyde!":PRIN
T "You are very strong!":JK=1:GOTO 289
0
2400 IF D(B)<>19 THEN 2420
```



```
2410 IF A<>59 OR E$<>"TER" OR I(57)<>1
 THEN 2420
2412 PRINT "Plant grows to ceiling.":A
$(48*SA+1,49*SA)="HUGE PLANT
  ":H$(48*SA+1,49*SA)=A$(48*SA+1,49*SA
)
2415 PT=1:GOTO 2890
2420 IF D(B)<>20 THEN 2450
2430 IF E$<>"UTE" OR A<>61 OR I(50)<>1
 OR R1<>0 THEN 2440
2432 PRINT "Indian rope rises up to yo
u":R1=1:A$(55*SA+1,56*SA)="END OF ROPE
          "
2435 H$(55*SH+1,56*SA)=A$(55*SA+1,56*S
A):A(56)=-61:GOTO 2890
2440 IF E$="ANO" AND A=63 THEN PRINT "
Liberace you're not!":GOTO 2890
2450 IF D(B)<>21 THEN 2480
2460 IF E$="OPE" AND O=64 AND RS=0 THE
N PRINT "Tie the other end first.":GOT
O 2890
2470 IF E$<>"OPE" OR A<>64 OR RS<>1 TH
EN 2480
2472 PRINT "Hook on rope catches other
 hook near  entrance and stretches tig
ht":RS=2:K3=54:M$="":GOSUB 2540:GOTO 2
890
2480 PRINT "You can't do that now.":GO
TO 1140
2490 PRINT "3 hours pass.":TM=TM+5:GOT
O 1140
2500 FOR K2=1 TO 7:IF C$(K2*SC-SC+1,K2
*SC)<>H$(K3*SH-SH+1,K3*SH) THEN 2510
2502 C$(K2*SC-SC+1,K2*SC)="
          ":C$(K2*SC-SC+1,K2*SC)=R$:G
OSUB 2520:RETURN
2510 NEXT K2:RETURN
2520 IF R$="" THEN I(K3)=0
2530 RETURN
2540 FOR K2=1 TO 50:IF A$(K2*SA-SA+1,K
2*SA)<>H$(K3*SH-SH+1,K3*SH) THEN 2550
2542 A$(K2*SA-SA+1,K2*SA)="
          ":A$(K2*SA-SA+1,K2*SA)=M$
2545 H$(K3*SH-SH+1,K3*SH)=A$(K2*SA-SA+
1,K2*SA):A(K2)=A(K2)*(M$<>""):I(K3)=I(
K3)*(M$<>""):RETURN
2550 NEXT K2:RETURN
2560 GOTO 2890
2570 PRINT "There is a path east, but
a vicious,  snarling Doberman blocks t
he way.":RETURN
2580 PRINT "Aquarium full of piranha f
ish.":RETURN
2590 PRINT "The staircase is very rott
en and one  wooden step is missing.":R
ETURN
2600 PRINT "Kidnapper counting money.
 He doesn't see you yet.":RETURN
2610 PRINT "Kidnapper is scared.":RETU
RN
2620 PRINT "There is a path north, but
 a raging  fire blocks your way.":RET
URN
2630 PRINT "A large swimming pool span
s across theentire room, dividing it i
n half.  It is very deep and empty"
2632 RETURN
2640 PRINT "Swimming pool full of wate
r.":RETURN
2650 IF A=16 AND DP=0 THEN 2570
2660 IF A=14 AND PF=0 THEN 2580
2670 IF A=25 AND S1=0 THEN 2590
2680 IF A=35 AND PM=0 THEN 2600
2690 IF A=35 AND PM=1 THEN 2610
2700 IF A=41 AND FR=0 THEN 2620
2710 IF A=48 AND SP=0 THEN 2630
2720 IF A=48 AND SP=1 THEN 2640
2730 RETURN
2740 DATA PAPER NOTE,2,DESK,-2,CHAIR,2
,LONG BROOM,3,TV SET,-5,CLOCK,-5,GRAFF
ITI ON WALL,-6,OPEN WINDOW,-7
2745 DATA TRAP DOOR IN CEILING,-9,DOWN
 BUTTON,-9,CABINET,-10,ENDS OF TWO LIV
E WIRES,-11,AQUARIUM,-14,ROPE,15
2750 DATA SLEEPING PILL,16,OPEN WINDOW
,-17,DESK,-17,KEY MAKING MACHINE,-19,S
UPER GLUE,20,WOODEN STAIR STEP,20
2755 DATA THICK COPPER SHEET,23,LOCKED
 DOOR,-24,WOODEN STAIRS,-25,TANK OF HE
LIUM GAS,28,LARGE DEFLATED BALLOON,29
2760 DATA LOCKED DOOR,-30,GUN,30,SIGN
ON WALL,-32,STRING VENDING MACHINE,-32
,OPEN WINDOW,-33,LOCKED DOOR,-43
2765 DATA BALL OF YARN,39,KNITTING NEE
DLES,40,PUSH BUTTON ON WALL,-42,SEXY O
FFICE WORKERS,-37,STAIRS,-44
2770 DATA PUSH BUTTON ON WALL,-46,BOOK
,47,MARY POPPIN'S UMBRELLA,50,OPEN WIN
DOW,-51,GRAFFITI ON WALL,-53
2775 DATA SMALL KEY,37,STEEL DOOR,-55,
VIAL OF YELLOW SOLUTION,56,JAR OF FLUI
D,56,CHILD'S SLIDE,-57
2780 DATA WATER COOLER,-58,PAPER CUP,5
8,SMALL-SIZED PLANT,-59,FLUTE,61,STAIR
S,-62,BOOK,64,PIANO,-63
2785 DATA LONG ROPE,63,FRONT DOOR,-65
2788 DATA ,0,,0,,0
2790 DATA WEST,W,NORTH,N,EAST,E,SOUTH,
S,UP,U,DOWN,D,GET,TAKE,DROP,PUT,GIVE,P
AY,CLIMB,GO,ENTER,OPEN,READ,CHECK
2795 DATA WATCH,PRESS,PUSH,TAPE,TIE,LI
GHT,MAKE,GLUE,INFLATE,SHOOT,KNIT,DRINK
,POUR,PLAY,THROW
2800 DATA 1,1,2,2,2,2,3,3,3,4,5,5,5,6,
6,9,9,10,11,12,13,14,15,18,19,20,21
2810 FOR K4=1 TO 58:IF A$(J*SA-SA+1,J*
SA)=H$(K4*SH-SH+1,K4*SH) THEN I(K4)=1:
K4=60
2820 NEXT K4:RETURN
2830 IF A=24 THEN X=22:GOTO 2870
2840 IF A=43 THEN X=31:GOTO 2870
2850 IF A=55 THEN X=43:GOTO 2870
2860 X=55
2870 A$(X*SA-SA+1,X*SA)="OPEN DOOR
          ":RETURN
2880 PRINT "OK!"
2890 IF E$="OOR" AND A=71 THEN PRINT "
You need a key."
2900 PRINT :PRINT :PRINT "Hit any key"
;CHR$(253);:POKE 764,255
2902 IF PEEK(764)=255 THEN 2902
2904 POKE 764,255:GOTO 830
2910 GOTO 2900
2920 PRINT "OUT.  You've made it!"
2930 END
2940 PRINT "AAHHHHHHHHHHHHHHHHHH...."
2950 PRINT "You die a horrible and gri
zzly death, your body torn to shreds."
:GOTO 2980
2960 PRINT "Now you look like a pancak
e!"
2970 PRINT "You seem to have gotten":P
RINT "     yourself killed!";
2980 PRINT :PRINT "To try again, you'l
l have a start overfrom the 9th floor.
"
2990 END
3000 IF LEN(A9$)<5 THEN 3010
3001 IF A9$(1,5)="LIGHT" THEN 1145
3010 PRINT "You can't see where you're
 going...  Ooops! Sorry, you fell an
d broke yourneck!"
3020 GOTO 2970
3030 I(38)=0:G(1)=0:IF JK=1 THEN PRINT
 "You forgot the antidote! Fluid poiso
nsyou.":GOTO 2970
3040 BR=0:FL=FL-1:PRINT "You made it d
own to the next floor,  but you had t
o drop anything you were carrying."
3050 FOR G=1 TO 7:C$(G*SC-SC+1,G*SC)="
                    ":NEXT G:FOR G=5
6 TO 58:I(G)=0:NEXT G
3060 GOTO 2910
9000 PRINT "Pardon me?":GOTO 1140
```



Take the fat off BASIC and turn it into swift, trim, Machine Language...



...without any sweat from you.

INTRODUCING THE HAYDEN APPLESOFT COMPILER
For only $175.00

You can cut the fat from BASIC with the **HAYDEN APPLESOFT COMPILER**, the first true compiler available for the Apple II computer. This multi-pass compiler allows your BASIC programs to run up to ten times faster in Machine Language — faster than your Apple II can normally interpret Applesoft BASIC.

The **HAYDEN COMPILER FEATURES:**

* Runs standard Applesoft BASIC programs up to ten times faster.
* Seventeen-pass compiler produces true machine code.
* Compiles code at user specified address.
* Requires only Applesoft in **ROM** and one disk drive.
* Handles Hi-Res graphics and shape tables.
* Arithmetic completed for faster operation.
* Allows multiple programs to reside in memory at the same time.
* Works in a single or multiple disk environment or with hard disk.
* At the end of compilation, the compiler provides a detailed analysis of the structure.

System Requirements

* A 48K Apple II Plus or Apple II with Applesoft in ROM.
* The Autostart ROM chip and at least one disk drive.
* Systems equipped with language system or Microsoft RAMCard.

* Introductory price for our readers Save $25 Limited time only*

$175.00

The HAYDEN APPLESOFT COMPILER is a product of the Hayden Book Company, Inc.

*offer expires August 15, 1981

SoftSide Selections

6 South Street Milford NH 03055

For Orders Only 603-673-0585

MasterCard VISA

ENHBAS™
FROM THE CORNSOFT GROUP

Announcing the most important utility ever introduced for the TRS-80™ Model I and Model III.

ENHBAS™ is an Enhanced BASIC extension module, which loads at the top of BASIC, adding many commands and background tasks. Over 30 new commands added to your BASIC:

SORT—Multikeying, multitagging array sort. Sort thousands of items in mere seconds, all with one command!

JNAME—Use line labels along with line numbers in branching statement, as in Assembly Language, using the ENHBAS commands GTO and CSUB (special GOTO and GOSUB). How many times have you wanted to use variables to reference line numbers? Now you can! GTO and CSUB allow variable expressions as operands, such as in GTO X + 40.

WHILE/WEND—New, structured programming loop construct. Makes for more logical program flow (less GOTOs).

EXEC/EVAL—Two new, extremely powerful functions! EVAL evaluates an algebraic expression in string form. With EVAL you can manipulate complex functions in string form, and then evaluate them. EXEC executes a string expression as if it were a BASIC program line. With EXEC, your computer can actually write its own programs and execute them.

CALL—Pass control to Machine Language subroutines at any address, passing parameters both ways.

CLM/PAGE—Set up automatic page roll-over and other line printer functions from BASIC.

All these and many more!

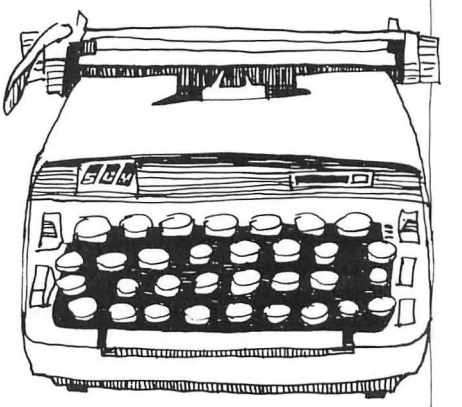
In addition to the above commands, Model I ENHBAS contains vector graphics and drawing commands. ENHBAS includes many background utilities (Model I version); User-definable cursor, key click, two-tone beep on error, automatic lower case, automatic debounce, short-entry commands (shift-letter prints command), real control keys, one letter commands, and formatted LISTings.

16K, S-80 Cassette,
Model I or Model III.....$59.95
32K, S-80 Disk
Model I or Model III.....$59.95

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

Typing Tutor

by Roy Groth



Wish you were a better typist, but don't want to take (or pay for) a class? Teach yourself to type with the aid of your micro-computer. With TYPING TUTOR you will be quizzed and graded, but you set the pace at which you learn. TYPING TUTOR is a set of programs that lets you become as good a typist as you wish, allowing you to advance from one level to the next when you feel comfortable with your skills.

Let "hunt and peck" slip into the past, teach yourself speed and accuracy on the keyboard with TYPING TUTOR.

S-80/16K/Cassette/
..... $15.95

RAMWARE™

MasterCard VISA

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

SoftSide™

Dear SoftSide Subscriber:

Thank you for your ongoing commitment to SoftSide Magazine. I believe that SoftSide subscribers are a special group who are more actively involved than most readerships in computing, its challenges, headaches, and triumphs.

I've got a commitment to you too. It's my responsibility to bring you the best software available and deliver it to you every month. As owner of TSE/Hardside and SoftSide, publisher of Ramware and owner of Ramworks, which covers the retail end, I have a unique vantage point from which I've observed the development of personal computing. I believe that I've learned some important lessons that will benefit you as a SoftSide subscriber.

The most obvious is that the cost of software is soaring, due mainly to such factors as promotional and packaging expenses. And, some of the major channels of distribution have fallen short of their goals -- there aren't enough retail outlets that sell a wide variety of titles and it is difficult for the computer store owner to be well versed in all of the available software packages. Mail order is sometimes slow although the number of programs available is growing at a fantastic clip.

We at SoftSide want to address these problems -- enter SoftSide-DV.

SoftSide on disk has traditionally been a convenience for subscribers who preferred to have their typing done for them. SoftSide-DV (Disk Version) is an entirely different concept.

Actually, SoftSide-DV addresses more than the simple cost of software. With a subscription, you'll develop your system through your computer itself. Frankly, it has the ability to do some things that you can't do. The bottom line is this: we want to address both you and your computer, according to what each of you does best.

Besides SoftSide Magazine, you'll get all of the programs in the magazine already typed in, plus much more. Here's an idea of the kinds of things you can expect from a subscription to SoftSide-DV:

* Intriguing machine code and hybrid programs (mixed BASIC and machine code and/or super graphics) which won't appear in the magazine or on cassette.

* Data base features. Imagine a French-English dictionary or a monthly interactive crossword puzzle that keeps track of your

score and gives you hints as you need them. Or how about a disk-based checkbook maintenance and budgeting program?

* Conversions of programs for other computers that have appeared in past issues of SoftSide.

* An amazing concept in software: I-$tring interpreters. I-$tring's uses are virtually limitless. By exploiting interchangeable data bases, you'll be able to create your own adventures and dungeons or play along with the ones we'll provide. Map out floor plans. Plan your vacations with a road map that has the potential to show every route in the country.

* To err is human, to correct, a pain. In the event that bugs show up in programs, we'll supply fixes directly in the form of an ASCII file ready to merge into the original code with little effort.

* Modular BASIC, a 20K enhancement program (similar to Enhanced BASIC) which will add numerous features to your Model I computer.

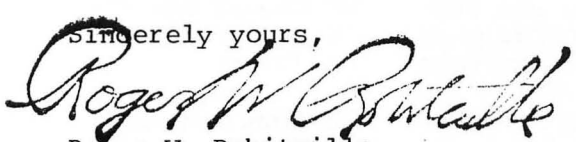
SoftSide-DV will change your definition of a magazine forever. It's a monthly multi-media adventure and right now a subscription is yours for only $99.95.

Our regular price is $125, but at our special introductory price, you pay only $8.33 per month for SoftSide Magazine and a filled-to-the-brim disk.

We'll calculate the cost of your subscription to SoftSide-DV based on the remaining portion of your current subscription (ain't computers wonderful?). Just send us an order along with payment, or use the business reply card in this issue and charge it on your MasterCard or Visa; be sure to include your card number, expiration date, and your signature.

We're making this special price available to you only until August 15, 1981, so order now. I'm looking forward to having you join hundreds of other SoftSide-DV subscribers in broadening the creativity of computing.

Sincerely yours,



Roger W. Robitaille
Publisher

P.S. Remember to act now--this offer expires on August 15, 1981!

One final note: There will not be a cassette equivalent to SoftSide-DV. Some of the things we'll be doing simply cannot be done on cassette.

Magic Paper Calculator

by Russell Starkey

"Magic Paper Calculator" is an S-80 program requiring a 16K tape or 32K disk system.

This is a program which will allow you to use your computer as a printing calculator, even if you don't have a printer. Well, sort of, anyway. Part of the screen display simulates a roll of paper tape being fed out of the maws of the calculator, and this tape can be scrolled up and down at will to examine all your calculations. And if you do happen to have a printer, you can also get a hard-copy printout on command.

In the following instructions, line A is the bottom line, line B is the second to the bottom, etc.

1. The usual symbols are used for the four basic functions: +, -, *, and /. ENTER may also be used instead of +. (It's not necessary to use the shift key for the + and *.) The item counter counts the total number of function operations.

2. Pressing S will give a subtotal; T will give the total. (The difference is that the subtotal is carried into the following calculations.) The accumulator (ACC) is the sum of all totals/subtotals.

3. Enter the number first, then the function to perform. For example, to add 4.80 and 1.18 and get a total, type:

4.80 + 1.18 + T

Or to multiply -17 by 3 and get a subtotal, type:

17 - 3 * S

Of course, any number of calculations can be strung together without having to press S or T.

4. To square line B, it's not necessary to re-enter the number. With line A blank, just pressing * and then S will do the trick.

5. Pressing Q will take the square root of line B and substitute this value on line B.

6. Pressing N will allow you to add a note to line B.

7. Memory access commands:

W, followed by a number key from 1-6, will write line B to the specified memory.

R, followed by a number key from 1-6, will recall the specified memory to line A.

RA recalls accumulator memory to line A.

RT recalls accumulated tax memory.

RM recalls RAM left in computer.

8. Clear commands:

A clears accumulator, item counter, and accumulated tax.

I clears item.

E clears entry.

CM followed by Y (yes) clears memory.

CP followed by Y clears the paper.

9. The up-arrow and down-arrow keys scroll the paper up and down; they can be held down for continuous movement.

10. The calculation mode is changed by pressing M. The modes available are fixed-point 0 through 4, financial, and floating-point.

11. To add tax to line B and give the total, press G. If you're in the midst of doing calculations, be sure to subtotal them before using G. (The first time this function is used, you will need to enter your tax rate as a whole number from 1 to 9.)

12. Pressing X will exchange lines B and C.

13. To find a percentage of line B, enter the percentage and then press P. Pressing P without a preceding number will cause the last percentage specified to be used in this calculation.

14. To discount (subtract) a percentage of line B, enter the percentage and then press D. Pressing D without a number will cause the last specified value to be used. This function can also be used to change the sign of line B, by entering 200 D.

15. Pressing H will make a hard copy of the data on a printer.

16. Errors:

a. To change a memo note, position the note at line B using the up- or down-arrow key, press N, and enter the new note.

b. To change a wrong number or function, position it at line A and enter the correct number and function.

c. A math error will result from trying to mulitiply or divide when there is no number in line B or if line B is a total.

d. An error message will result if you try to divide by zero.

e. A "Code #" error refers to a BASIC error message (see appendix B of the Reference Manual).

VARIABLES

A: Number input flag; =0 if inputting two or more numbers in a sequence, otherwise =1.
A1: Temporary variable.
AA: Length of entered number string.
AF: =0 for cleared memory; =1 if memory is in use.
AP: String length counter.
C ( ): Function mode code array: Numbers 1-6 designate addition, subtraction, multiplication, division, subtotal, and total.
C1 ( ): Memory function array.
C2-C4: String length variables.
CB: Stores last function.
CC: Copy of X2 before changed.
CD: Math pointer.
CK, CL: Temporary variables.
CM: Current mode number, 1-6.
DA, DS, DX: Temporary variables.
DJ: Math round-off fixer.
DT: Math total.
DV: Accumulator memory.
L ( ): String array to hold notes.
L1 ( ): String array to hold memory notes.
M ( ): Marquee strings.
N: Number array string.
N ( ): Number array.
N1 ( ): Memory number string array.
NA: Accumulator memory.
NB: Tax total memory.
Q: Down-arrow control variable.
QE: Last error type number.
QT: Tax rate control flag; =1 if rate has been entered, otherwise =0.
S: Temporary variable.
S1-S6: Graphics strings.
SA: String of number entered.
SJ: Percent amount.
SK: Discount amount.
SN: Prompt message.
TA: Tax rate.
X: Temporary variable.
X2: Array pointer.
XI: Item counter.
XL: Maximum array cell used.
Y: ASCII value of last key pressed.
Y1: Temporary variable.
YA, YB: Length of first and second sections of marquee.
YC: Marquee speed control variable.
YD: Marquee section pointer (= 0 or 1).
YE: Marquee character pointer.
YL: Marquee print size (=30).

```
120 REM REV ( 67 )  TRS-80 SYSTEMS 16K TAPE & 32K DISK
130 REM 16K CASSETTE SYSTEMS MEMORY SIZE=32500
140 REM BY RUSSELL STARKEY

Title page and initialization.

160 CLS:PRINT@512,CHR$(23);"MAGIC PAPER CALCULATOR"
180 CLEAR2200:DEFSTR M,N,L,S:DEFINT A,X,Y,C,Q:DEFDBL D,E,F,G:ON
ERROR GOTO 4920
200 DIM M(1),N(250),L(250),C(250),N1(6),L1(6),C1(6),S1(12):A=1:X
2=1:CM=2
220 DJ=.0000001:DS=1:DA=1:SJ="100":SK="5"

Test:  if Level II BASIC then control goes to error-handling
routine.

240 CMD"T"

Fix data error and define starting location of Machine Language
subroutine.

260 POKE 16553,255:DEFUSR0=&HABDE

Poke in machine code for 32K disk system.

280 FOR X=-21538 TO -21439:READ Y:POKEX,Y:NEXT
300 GOTO 360

Resume from line 5040 and poke in Machine Language code for 16K
tape system.

320 POKE16553,255:POKE16526,245:POKE16527,126
340 FOR X=32501 TO 32600:READ Y:POKE X,Y:NEXT

Load S1 string array with mode names.

360 FOR X=0 TO 12 :READ S:S1(X)=S:NEXT

Load marquee string array.

400 M(0)="                "
420 M(0)=M(0)+"USE  0 THRU 9 .    USE + - * /     S SUB TOTAL
 T TOTAL    "
440 M(0)=M(0)+"E CLEAR ENTRY    N ADD NOTE    [ ADVANCE PAPER
 "
460 M(0)=M(0)+CHR$(92)+" BACKSPACE PAPER    P PERCENT    0 ENTER
 00    "
480 M(0)=M(0)+"D DISCOUNT    W # STORE NUMBER AND NOTE    "
500 M(1)=" W # STORE NUMBER AND NOTE    R # RECALL    RA RECALL
ACC    RT TAX    Q SQUARE ROOT    "
520 M(1)=M(1)+"M MODE    G TAX    X EXCHANGE    H PRINTER    "
540 M(1)=M(1)+"I CLEAR ITEM    A CLEAR ACC MEM    CP CLEAR PAPER
   CM CLEAR MEMORY    "
560 M(1)=M(1)+"BY R. STARKEY    "
580 M(1)=M(1)+"          USE  0 THRU 9 .  "

Find length of marquee strings.

600 YA=LEN(M(0)):YB=LEN(M(1)):YL=30

Clear and set up screen (continue at 1460).

620 GOSUB960  :GOSUB1180 :GOTO1460


Key scan point.  Go to the right place for the selected
function.

640 REM
660 GOSUB2100
680 Y=ASC(N):IF Y>79 THEN 820
700 IF Y>59 THEN 860  ELSE IF Y=10 THEN 4300 ELSE IF Y=13 THEN 3
320
720 IF Y<45 THEN 940
740 ON Y-44 GOTO 3400 ,1440 ,3560
760 IF Y<58 THEN 1460
780 ON Y-57 GOTO 3480 ,3320
800 GOTO94C
820 ON Y-79 GOTO 5920 ,4800 ,3880 ,2140 ,3040
840 IF Y=91 THEN 4420 ELSE IF Y=87 THEN 4120 ELSE IF Y=88 THEN 4
740 ELSE 940
860 IF Y=65 THEN 3640 ELSE IF Y=67 THEN 3660 ELSE IF Y=69 THEN 1
580
880 IF Y=68 THEN 5800 ELSE IF Y=71 THEN 5600 ELSE IF Y=73 THEN 5
780
900 IF Y=72 THEN 4460 ELSE IF Y=77 THEN 4040 ELSE IF Y=78 THEN 1
740
920 IF Y=79 THEN 1380
940 SN="INVALID KEY":GOTO640

Clear and set up screen.

960 REM
980 CLS
1000 S6=CHR$(188):S1=STRING$(26,CHR$(140)):S2=STRING$(26,CHR$(13
1)):S3=CHR$(191):S4=CHR$(143)
1020 FOR X=1 TO 12:A1=USR(0):NEXT
1040 S5=CHR$(131):PRINT@2,"MAGIC PAPER CALCULATOR.";
1060 PRINT@64,S6;S1;S6;: FOR X=128 TO 950 STEP64 :PRINT@X,S3;TAB
(27)S3;:NEXT
1080 PRINT@960,S4;S1;S4;:PRINT@797,S6;S1;:PRINT@805,S1;S6;
1100 PRINT@861,S3;:PRINT@925,S3;:PRINT@895,S3;:PRINT@959,S3;
1120 PRINT@989,S2;:PRINT@997,S2;:POKE16383,131
1140 PRINT@863,"USE THE FOLLOWING KEYS :";
1160 RETURN

Print memory box on screen.

1180 REM
1200 PRINT@33,S6;S1;S6;
1220 PRINT@95,"M ";S3;" 1";TAB(60)S3;
1240 PRINT@159,"E ";S3;" 2";TAB(60)S3;
1260 PRINT@223,"M ";S3;" 3";TAB(60)S3;
1280 PRINT@287,"O ";S3;" 4";TAB(60)S3;
1300 PRINT@351,"R ";S3;" 5";TAB(60)S3;
1320 PRINT@415,"Y ";S3;" 6";TAB(60)S3;
1340 PRINT@481,S5;S2;S5;
1360 RETURN

Double-zero input point.

1380 REM
1400 IF LEN(N(X2))<12 THEN N(X2)=N(X2)+"00":SA=N(X2):GOTO640
1420 GOTO640

Number input point.

1440 IF CM=0 OR CM=5 OR AP<>50 AND A=0 THEN 640
1460 REM
1480 IF A=1 THEN N(X2)="":A=0:SN="#":AP=50
1500 AA=LEN(N(X2)):IF AA>9 THEN 660
1520 IF Y=46 THEN AP=AA+1
1540 IF CM<>6 AND AP+CM=AA THEN 660
1560 N(X2)=N(X2)+N:SA=N(X2):GOSUB5360 :GOSUB2120 :GOTO680

Clear input point.

1580 REM
```

continued on next page

continued from previous page

```
1600 C(X2)=0:L(X2)=""
1620 N(X2)="":SN="CLEAR ENTRY":GOTO 640

Call Machine Language subroutine to shift one line up the page.

1640 REM
1660 A=1:X=834:Y=LEN(N(X2))
1680 A1=USR(1):PRINT@X,L(X2);:PRINT@X+23-Y,N(X2);
1700 X2=X2+1:IF X2=251 THEN X2=1
1720 RETURN

Input note string.

1740 REM
1760 CC=X2:GOSUB3000
1780 L(CC)="":SN="ENTER NOTE  OR  PRESS <ENTER>.":GOSUB5340
1800 GOSUB1900 :GOSUB2120 :IF ASC(N)=13 THEN1880
1820 IF ASC(N)=8 THEN L(CC)=LEFT$(L(CC),LEN(L(CC))-1):GOTO1800
1840 IF LEN(L(CC))>9 THEN L(CC)=L(CC)+N:GOSUB1900 :GOTO 1880
1860 L(CC)=L(CC)+N:GOTO1800
1880 SN=" < READY > ":GOTO640
1900 PRINT@834,"          ";:PRINT@834,L(CC);:RETURN

Call Machine Language subroutine to shift one line down the
page.

1920 REM
1940 X2=X2-1:IF X2=0 THEN X2=250
1960 A1=USR(0):Y=X2-12
1980 IF X2>12 THEN Y1=LEN(N(Y)):PRINT@130,L(Y);:PRINT@153-Y1,N(Y
);:RETURN
2000 X=X2+238:PRINT@130,L(X);:PRINT@153-LEN(N(X)),N(X);:RETURN

Clear and print memory line to screen at X.

2020 REM
2040 PRINT@X,STRING$(26," ");
2060 PRINT@X,L(Y);TAB(46)N(Y);
2080 RETURN

Call marquee subroutine.

2100 GOSUB5260

Input a keypress; otherwise call marquee subroutine.

2120 N=INKEY$:IF N="" THEN GOSUB5120 :GOTO2120 ELSE RETURN

Subtotal point.

2140 REM
2160 C(X2)=5:L(X2)=""
2180 GOSUB2200 :SN=SN+" SUB TOTAL":N(X2)=N(X2)+" S":GOSUB1640 :G
OTO640

Math subroutine.

2200 REM
2220 CC=X2:GOSUB3000 :SN="WAIT I'M CALCULATING":GOSUB5260
2240 GOSUB5120 :IF C(CC)<5 AND C(CC)<>0 THEN GOSUB3000 :GOTO2240

2260 CB=C(CC):S=""
2280 DT=0:DS=1:CK=CC:IF C(CC)=5 THEN C(CC)=1 ELSE GOSUB2920
2300 GOSUB5120 :CD=1:IFC(CC)>4 THEN SN="": GOSUB2580 :N(X2)=N:GO
TO2860
2320 X=CC:GOSUB2960
2340 IF C(X)=3 OR C(X)=4 THEN 2420
2360 IF C(CC)=1 THEN DT=DT+VAL(N(CC)):GOSUB2920 :GOTO2300
2380 IF C(CC)=2 THEN DT=DT-VAL(N(CC)):GOSUB2920 :GOTO2300
2400 SN="MATH":GOTO3820
2420 GOSUB5120 :GOSUB2960 :IF C(X)=3 OR C(X)=4 THEN CD=CD+1:GOTO
2420
2440 X=CC:CL=CD+1:REM CD = # OF  * OR /  IN A ROW
2460 IF C(X)=1 OR C(X)=2 THEN DS=VAL(N(X))
2480 IF C(X)=3 THEN DS=DS*VAL(N(X))
2500 IF C(X)=4 THEN DS=DS/VAL(N(X))
2520 GOSUB5120 :GOSUB2960 :CL=CL-1:IF CL<>0 THEN 2460
2540 IF C(CC)=1 THEN DT=DT+DS ELSE IF C(CC)=2 THEN DT=DT-DS ELSE
 SN="MATH":GOTO 3820
2560 FOR X=0 TO CD:GOSUB2920 :NEXT:GOTO2300
2580 IF DT>0 THEN DT=DT+DJ ELSE IF DT<0 THEN DT=DT-DJ
2600 IF ABS(DT)>.01 THEN 2660
2620 SN="UNDERFLOW ":DT=0
2640 N="0":GOTO2840
2660 C1=LEN(STR$(INT(DT)))
2680 IF CM=5 THEN C3=2:GOTO2740
2700 IF CM=6 THEN C4=13:N=LEFT$(STR$(DT),C4):DT=VAL(N):GOTO2800
2720 C3=CM
2740 IF CM=0 THEN C4=C1:GOTO2800 ELSE IF ABS(DT)>=1 THEN C4=C1+C
3+1:GOTO2800
2760 IF CM=0 THEN 2640
2780 C4=C3+2
2800 N=LEFT$(STR$(DT),C4)
2820 IF C1>13 THEN SN="OVERFLOW "
2840 RETURN
2860 C(CK)=CB:IF CB=5 THEN DV=DV+DT-VAL(N(CK))ELSE DV=DV+DT
2880 DT=DV:GOSUB2580 :DV=VAL(N):NA=N
2900 RETURN
2920 CC=CC+1:IF CC=251 THEN CC=1
2940 RETURN
2960 X=X+1:IF X=251 THEN X=1
2980 RETURN
3000 CC=CC-1:IF CC=0 THEN CC=250
3020 RETURN

Total point.

3040 REM
3060 C(X2)=6
3080 GOSUB2200 :SN=SN+"  TOTAL":N(X2)=N(X2)+" T":GOSUB1640
3100 GOTO640

Construct string SA in correct format using DT.

3120 XI=XI+1:GOSUB5160 :IF CM=6 OR A=1 OR CM=0 THEN RETURN
3140 AA=LEN(SA):IF AP=50 THEN SA=SA+".":AP=LEN(SA)-1
3160 IF CM=5 THEN 3200
3180 IF AA>=AP+CM THEN RETURN ELSE SA=SA+"0":AA=AA+1:GOTO3180
3200 IF A=0 THEN DT=VAL(SA)/100 ELSE RETURN
3220 C1=LEN(STR$(INT(DT))):C2=LEN(STR$(DT))
3240 IF C2=C1+3 OR C2=C1+2 AND DT<1 THEN SA=STR$(DT):RETURN
3260 IF C2=C1+1 OR C2=C1+2 THEN SA=STR$(DT)+"0":RETURN
3280 SA=STR$(DT)+".00":RETURN
3300 RETURN

Addition input point.

3320 REM
3340 GOSUB3120
3360 N(X2)=SA+" +":C(X2)=1:GOSUB1640 :SN="+"
3380 GOTO 640

Subtraction input point.

3400 REM
3420 GOSUB3120
```

```
3440 N(X2)=SA+" -":C(X2)=2:GOSUB1640 :SN="-"
3460 GOTO 3380

Multiplication input point.

3480 REM
3500 GOSUB3120
3520 N(X2)=SA+" *":C(X2)=3:GOSUB1640 :SN="MULTIPLIY "
3540 GOTO 3380

Division input point.

3560 REM
3580 GOSUB3120
3600 N(X2)=SA+" /":C(X2)=4:GOSUB1640 :SN="DIVIDE"
3620 GOTO 3380

Clear accumulator memory.

3640 NA="":NB="":DX=0:DV=0:SN="ACC CLEARED":XI=0:GOTO640

Clear memory or paper.

3660 REM
3680 SN="C ":GOSUB2100
3700 A=1:Y=ASC(N):IF Y=69 THEN 1580
3720 IF Y=77 THEN SN="CLEAR MEMORY":GOSUB 3840 :GOTO 3800
3740 IF Y=80 THEN SN="CLEAR PAPER":GOSUB 3840 :SN="WAIT I'M ERAS
ING THE PAPER":GOSUB5260 :GOSUB5120 ELSE 3820
3760 N="":CC=0:X2=1:FORX=1TO250:GOSUB5120 :N(X)="":L(X)="":C(X)=
0:NEXT:SN="PAPER CLEARED"
3780 GOSUB1000 :GOTO640
3800 AF=0:FOR X=1 TO 6 :S(X)="":NEXT:SN="MEMORY CLEARED":GOSUB11
80 :GOTO 640
3820 SN=SN+" ERROR  ":GOTO 640
3840 SN=SN+" ARE YOU SURE ?  Y,N":GOSUB2100
3860 IF N="Y" THEN RETURN ELSE SN="OK":GOTO 640

Recall point.

3880 REM
3900 SN="RECALL # ?":GOSUB2100 :IF N="M" THEN N(X2)=STR$(MEM):SA
=N(X2):GOTO4020
3920 IF N="T" THEN N(X2)=NB:SA=NB:GOTO4020
3940 IF N="A" THEN N(X2)=NA:SA=NA:GOTO4020
3960 Y=ASC(N):IF Y<49 OR Y>54 THEN 3820
3980 Y=Y-48:X=X2:IF C1(Y)=0 THEN 3820
4000 L(X)=L1(Y):N(X)=N1(Y):C(X)=C1(Y):SA=LEFT$(N(X),LEN(N(X))-2)
4020 SN="WHAT SIGN ?   < + - / * >":A=1:GOTO640

Change mode.

4040 REM
4060 IF CM<6 THEN CM=CM+1 ELSE CM=0
4080 IF CM=6 THEN DJ=.000000000001 ELSE DJ=.0000001
4100 GOTO640

Store to memory.

4120 REM
4140 SN="STORE # ?":GOSUB2100
4160 Y=ASC(N):IF Y<49 OR Y>54 THEN 3820
4180 Y=Y-4E:X=X2-1:IF X=0 THEN X=250
4200 L1(Y)=L(X):N1(Y)=N(X):C1(Y)=C(X):PRINT@38+64*Y,STRING$(22,"
 ");
4220 IF AF=0 THEN AF=1:GOSUB1180
4240 PRINT@38+64*Y,L1(Y);
4260 PRINT@59+64*Y-LEN(N1(Y)),N1(Y);
4280 SN="STORE INTO MEMORY "+STR$(Y):GOTO640

Shift paper down.

4300 REM
4320 Q=0:SN=CHR$(92):GOSUB1920
4340 Q=Q+1:IF PEEK(14400)<>16 AND PEEK(14400)<>8 THEN N=INKEY$:G
OTO640
4360 IF Q<9 THEN 4340
4380 IF PEEK(14400)=16 THEN SN=CHR$(92):GOSUB1920 ELSE SN="[":GO
SUB1640
4400 GOSUB5340 :GOSUB5120 :GOTO4340

Shift paper up.

4420 REM
4440 Q=0:GOSUB1640 :SN="[":GOTO4340

Hard copy (printer).

4460 IF PEEK(14312)>128 THEN SN="PRINTER NOT READY":GOTO640
4480 SN="PRINTER":GOSUB5260
4500 LPRINT" -----   MAGIC  PAPER  CALCULATOR  -----"
4520 LPRINT" "
4540 FOR X=1 TO XL:GOSUB5120
4560 IF C(X)=0 THEN 4620
4580 IF CM=6 THEN LPRINTX,L(X),N(X):GOTO 4620
4600 LPRINT X,L(X);" ",:LPRINT USING S1(CM+7);VAL(N(X))
4620 NEXT
4640 IF AF=0 THEN 4700 ELSE LPRINT "MEMORY :"
4660 FOR X=1 TO 6:LPRINTX,L1(X),:IF CM=6 THEN LPRINT N1(X) ELSE
LPRINT USING S1(CM+7);VAL(N1(X))
4680 NEXT
4700 LPRINT"ACCUMULATOR  MEMORY = ",:IF CM=6 THEN LPRINT DV ELSE
 LPRINT USING S1(CM+7);DV
4720 LPRINT" ":GOTO 640

Exchange point.
```

continued on next page

continued from previous page

```
4740 CC=X2:GOSUB3000 :X=CC:GOSUB3000 :Y=CC:A1=C(X):C(X)=C(Y):C(Y
)=A1
4760 N=L(X):L(X)=L(Y):L(Y)=N:N=N(X):N(X)=N(Y):N(Y)=N
4780 SN="EXCHANGE":GOSUB1920 :GOSUB1920 :GOSUB1640 :GOSUB1640 :G
OTO640

Square root.

4800 CC=X2:GOSUB3000 :DT=VAL(N(CC)):IF DT<=0 THEN 640
4820 SN="WAIT":GOSUB 5340
4840 DA=DS:DS=(DT/DS+DS)/2
4860 IF DS<>DA THEN 4840 ELSE DT=DS
4880 GOSUB2580 :N(CC)=N+" @":SA=N:C(CC)=1
4900 SN="SQUARE ROOT":GOSUB1920 :GOSUB1640 :GOTO640

Error-handling routine.

4920 REM
4940 QE=(ERR/2)+1
4960 IF QE=5 THEN SN="ILLEGAL FUNCTION CALL":QE=505
4980 IF QE=6 THEN SN="OVERFLOW":QE=505
5000 IF QE=11 THEN SN="DIVISION BY ZERO":QE=505
5020 IF QE=14 THEN SN="OUT OF STRING SPACE":QE=505
5040 IF QE=23 THEN RESUME 320
5060 IF QE<>505 THEN SN="CODE # ":SN=SN+STR$(QE)
5080 SN=SN+" ERROR":GOSUB 960:GOSUB 1180 :RESUME 640

Marquee move and print subroutine.

5120 REM
5140 YC=YC+1:IF YC=6 THEN YC=0 ELSE RETURN
5160 YE=YE+1:PRINT@927,MID$(M(YD),YE,YL);
5180 IF YE+YL=YA AND YD=0 THEN YD=1:YE=0
5200 IF YE+YL=YB AND YD=1 THEN YD=0:YE=0
5220 RETURN

Update screen.

5260 REM
5280 PRINT@670,STRING$(32," ");:PRINT@670,"MODE IN ";S1(CM);
5300 PRINT@542,STRING$(34," ");:PRINT@542,"ACC MEMORY =";NA;
5320 PRINT@606,STRING$(34," ");:PRINT@606,"ITEM";XI;TAB(40)"TAX
";NB;
5340 PRINT@732,STRING$(34," ");:PRINT@734,SN;
5360 PRINT@897,STRING$(26," ");:PRINT@898,"LINE";X2;:PRINT@921-L
EN(N(X2)),N(X2);:IF XL<X2 THEN XL=X2
5380 RETURN

Data for Machine Language screen control routine.

5400 DATA 205,127,10,0,0,62,0,133,40,51,62,11,17,130,60
5420 DATA 33,194,60,1,26,0,237,176,61,40,17,229,213,225,17
5440 DATA 38,0,25,229,209,225,1,38,0,9,24,232,0,62,26,33,64
5460 DATA 63,35,54,128,61,32,250,0,0,0,201,0,0,0,62,11,17,89
5480 DATA 63,33,25,63,1,26,0,237,184,61,40,16,229,213,225,17
5500 DATA 218,255,25,229,209,225,1,218,255,9,24,232,62,26
5520 DATA 33,128,60,24,204

Mode names.

5540 DATA FIX 0,FIX 1,FIX 2,FIX 3,FIX 4,FINANCIAL,FLOATING POINT

Print-using format for each mode (for use with printer).

5560 DATA ###############.,##############.#,#############.##,######
#####.###
5580 DATA ##########.####,$############.##,#

Calculate tax and total amounts.

5600 IF CM<>5 THEN CM=2:DJ=.0000001
5620 IF QT=0 THEN SN="WHAT IS YOUR TAX RATE % " ELSE 5660
5640 GOSUB2100 :SN="":GOSUB5340 :QT=1:TA=VAL(N)/100
5660 SN="TAX & TOTAL "
5680 CC=X2:GOSUB3000 :DT=(VAL(N(CC))+.02)*TA:GOSUB2580 :N(X2)=N+
" +":L(X2)="TAX":C(X2)=1
5700 DX=DX+VAL(N):GOSUB1640 :DT=DX:GOSUB2580 :DX=VAL(N):NB=N
5720 CC=X2:GOSUB3000 :Y=CC:GOSUB3000 :DT=VAL(N(Y))+VAL(N(CC)):GO
SUB2580 :N(X2)=N+" T":C(X2)=6
5740 L(X2)="TOTAL":GOSUB1640
5760 GOTO 640

Clear item counter.

5780 XI=0:GOTO640

Calculate discount.

5800 SN="DISCOUNTED"
5820 IF N(X2)="" THEN N(X2)=SK :SN="MEMORY "+SN ELSE SK=N(X2)
5840 CC=X2:GOSUB3000 :N(X2)=N(X2)+" %"
5860 DT=((100-VAL(SK))/100)*VAL(N(CC))
5880 GOSUB2580 :GOSUB1640 :C(X2)=5
5900 N(X2)=N+" S":L(X2)="DIS":GOSUB1640 :GOTO640
5920 SN="PERCENT"
5940 IF N(X2)="" THEN N(X2)=SJ :SN="MEMORY "+SN ELSE SJ=N(X2)
5960 CC=X2:GOSUB3000 :N(X2)=N(X2)+" %"
5980 DT=(VAL(SJ)/100)*VAL(N(CC))
6000 GOSUB2580 :GOSUB1640 :C(X2)=5
6020 N(X2)=N+" S":GOSUB1640
6040 GOTO640
```

S-80 ONE LINERS

```
10 CLS:PRINT@25,"LOAN SCHEDULE":INPUT"AMOUNT,MONTHS,PCT";A,N,R:R
=R/1200:P=A/((1-(1/((1+R)^N)))/R):PRINT"PAYMENT","PRINCIPLE","IN
TEREST","BALANCE":FORX=1TON:I=R*A:A=A-P+I:PRINTP,P-I,I,A:Y=Y+1:I
FY=10,Y=0:INPUT"PRESS ENTER";A$:PRINT@192,CHR$(31);:NEXTELSENEXT
```

Ronald Johnson
Columbia, S.C.

```
1 DEFINTA-Z:RANDOM:A=64:B=23:FORC=0TO999:CLS:FORD=0TO68:E=RND(3)
-2:F=RND(3)-2:FORH=0TORND(28):IFA+E<00RB+F<00RA+E>64ORB+F>23,NEX
TD:FORY=0TO999:NEXT:RUNELSEA=A+E:B=B+F:SET(A,B):SET(127-A,B):SET
(A,47-B):SET(127-A,47-B):NEXT:NEXTD:FORZ=0TO999:NEXT:NEXTC:RUN
```

T. Krumholz
Springfield, Mo.

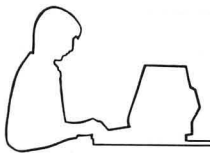
```
10 DEFSTRA:F=480:E=F:B=F:C=F:D=F:X=F:Y=F:W=F:Z=F:FORQ=1TO7:OUT25
5,10:A=CHR$(RND(62)+128):E=E-67:B=B-61:C=C+67:D=D+61:X=X-4:Y=Y+4
:W=W-64:Z=Z+64:PRINT@X,A:PRINT@F,A;A:PRINT@Y,A:PRINT@W,A:PRINT@Z
,A;:PRINT@E,A:PRINT@B,A:PRINT@C,A;:PRINT@D,A;:CLS:NEXT:GOTO10
```

```
40 CLS:B=60:D=24:E=2.3:FORR=2TO51STEP7:FORA=-RTOR:X=R*R-A*A:Y=SQ
R(X):Y=INT(Y-.5):SET(A+B,D+Y/E):SET(A+B,D-Y/E):NEXT:NEXT:FORR=51
TO2STEP-7:FORA=-RTOR:X=R*R-A*A:Y=SQR(X):Y=INT(Y-.5):RESET(A+B,D+
Y/E):RESET(A+B,D-Y/E):NEXT:NEXT:GOTO40
```

Harland Hill
Chambersburg, Pa.

```
1 CLS:X=1:Y=23:FORW=1TO15:FORZ=1TO9:C(Z)=-2+RND(3):NEXT:FORZ=1TO
9:Y=Y+C(Z):X=X+ABS(C(Z))-RND(0)/2:SET(X,Y):SET(127-X,Y):SET(127-
X,47-Y):SET(X,47-Y):NEXTZ,W:FORZ=1TO999:NEXT:GOTO1
```

Donald Wolf
Greendale, Wi.



Word Wars

by Rowland Archer

"Word Wars" is an S-80 game requiring 16K of memory.

The object of "Word Wars" is simple but challenging: Form as many words as possible in a fixed amount of time from a random "roll" of letters. The game may be played alone, with as many as six players, or even with teams in larger groups.

You will have approximately three minutes to form words from your roll of letters. Each word must be at least three letters long. That sounds simple enough — BUT, you are limited to five words of each length. You will probably find yourself rapidly filling the columns of 3-, 4-, and 5-letter words, and then staring frantically at the letters trying to find longer ones. But it's quite a thrill to discover a 9- or 10-letter word lurking in that jumble of 13 letters displayed on the screen.

To spur you on, the highest score so far in the current game is displayed at the bottom of the screen. It's a little disconcerting to know that your opponent scored 1600 points while you're still sitting there with 500, and less than a minute left. Those words must be in there somewhere!

Since storing a complete dictionary is a bit difficult in a 16K computer, each player's words are displayed for challenge at the end of a game. A good dictionary will come in handy at this point for resolving disputes. In addition to challenging your opponent's more questionable words, it's fun (?) as well to see all the "obvious" words that he got and you missed.

A set of five games constitutes a tournament; the team or individual with the highest score at the end of the tournament wins. So if you really draw a blank on one roll, never fear, because you can make a comeback on the next one. Incidentally, playing alone is a great way to hone your skills while waiting for an unsuspecting friend to challenge.

Complete instructions, including the details of play and scoring, are printed by the program on request.

VARIABLES

A$: Temporary string variable to hold INPUTs and INKEY$s.
AD: Amount to add to countdown clock during input.
BN: Bonus points for current word.
BN (*): Base bonus point value for each word length.
BP: Breakpoint score for double-bonus mode.
BW: Bonus-won flag; set to -1 if player has won double bonus.
CL: Countdown clock, in "seconds".
CP: Number of current player.
CS (*): Cumulative scores for each player.
FD: Flag, set to -1 if word typed in during post-game challenge is found.
GN: Current game number.
GW$: String of letters used to generate words. (The more times a letter is repeated in GW$, the more likely it is to be part of a given roll; may be altered if desired.)
HS: Holds current high score during post-game score ranking.
I, J, K: Loop counters.
KP: PRINT@ position for clock display.
L$: Current letter being added to roll during roll generation.
L: Loop counter and array index.
LW: Length of current word.
M$: Message printed by GOSUB 320.
NL: Number of letters in current roll (13 for first four games, then 20 for fifth game).
NM$(*): Array of players' names.
NP: Number of players.
NS$(*): Array of players' names with possessive (' or 's) added.
NW(I, J): Array tallying number of words formed by player I during the current game, of length J+3.
NW: Number of words of same length as current word, already formed by current player.
OS(*): Array of offsets for PRINT@ locations of words, indexed by their length.
PA: PRINT@ position for letters typed during INKEY$ of current word.
PD$: String of blank padding for entering messages.
PN: Penalty points for current successful word challenge.
QP: Flag, set to -1 if "Q" is present in the current roll. (If it is, a random letter is replaced with "U".)
RL$: Current roll of letters.
SB: Score to beat; high score for current game.
SC(*): Array of scores for current game.
SZ: Size of word currently being blanked out on screen, between players.
TI: Ticks of the clock; clock is decremented by one whenever TI is greater than 30.
TS(*): Temporary array of scores used during score-ranking computation.
UB$: Graphics string containing box used to frame word display.
W$: Current word.
WA(*): Array holding count of letters in current roll: WA(0) is count of As, WA(1) is count of Bs, etc.
WD$(I,J): Array of words formed during current game by player I grouped by length.
WW(*): Temporary array used in validation of current word.
XX: Loop counter.

```
110 GOTO 370

Decrement clock, print time remaining; check for 60-second
warning, and buzz relay if it's time.

120 CL=CL-1: TI=0: PRINT@KP,CL;CHR$(30);: IF CL=60 THEN PRINT@14
8,"-- 60 SECOND WARNING --";: FOR XX=1 TO 50: OUT 255,4: OUT 255
,1: NEXT XX: PRINT@148,CHR$(30);: TI=30
130 RETURN
```

```
Principal input routine: accepts and validates letters,
recognizes special characters such as SHIFT DOWN ARROW and
ENTER.

140 A$=INKEY$: TI=TI+1: IF TI>30 THEN GOSUB 120: IF CL=0 THEN 83
0
150 IF A$="" THEN 140
160 IF A$>="A" AND A$<="Z" THEN W$=W$+A$: PRINT@PA,A$;: PA=PA+1:
 TI=TI+AD: GOTO 140
```

continued on next page

continued from previous page

```
170 IF A$=CHR$(13) THEN IF LEN(W$)>0 THEN 760 ELSE 140
180 IF A$=CHR$(27) AND NP=1 THEN M$="** ABANDONING THIS GAME **"
: FOR I=1 TO 2: GOSUB 320: NEXT I: CLS: GN=GN-1: GOTO 1460
190 IF A$=CHR$(8) THEN IF W$="" THEN 140 ELSE W$=LEFT$(W$,LEN(W$
)-1):PRINT@PA,CHR$(8);:PA=PA-1:TI=TI+AD:GOTO 140
200 IF A$=CHR$(24) THEN PA=PA-LEN(W$): W$="": PRINT@PA,CHR$(30);
:TI=TI+AD:GOTO140
210 GOTO 140
```

Generate the roll of letters from which to make words. If roll contains a Q, be sure it contains a U also (give the guy a break!).

```
230 RANDOM: QP=0: RL$="": FOR I=0 TO 25: WA(I)=0: NEXT I
250 FOR I=1 TO NL: L$=MID$(GW$,RND(LEN(GW$)),1): RL$=RL$+L$
260 L=ASC(L$)-65: WA(L)=WA(L)+1
270 IF L$="Q" THEN QP=-1
280 NEXT I
290 IF QP THEN I=RND(NL): L=ASC(MID$(RL$,I,1))-65: WA(L)=WA(L)-1
: WA(20)=WA(20)+1: RL$=LEFT$(RL$,I-1)+"U"+RIGHT$(RL$,NL-I)
300 RETURN
```

Print the message contained in M$ on the screen, pause, then blank it out and return.

```
320 PD$=STRING$((64-LEN(M$))/2,32): PRINT@128,PD$;M$;
340 FOR XX=1 TO 500: NEXT XX: PRINT@128,PD$;STRING$(LEN(M$),32);
350 RETURN
```

Initialization. Ask if instructions are needed; get names of players; initialize variables.

```
370 CLS:PRINTCHR$(23):PRINT@320,;:PRINTTAB(5)"WORD WARS":PRINT:P
RINTTAB(5)"BY ROWLAND ARCHER":FOR I=1 TO 1000: NEXTI: CLS
390 INPUT"DO YOU NEED INSTRUCTIONS (Y/N)";A$: IF LEFT$(A$,1)="Y"
 THEN GOSUB 1520
400 CLEAR 3000: DEFINT A-Z
410 INPUT"HOW MANY PEOPLE ARE PLAYING WORD WAR";NP
420 IF NP<1 THEN PRINT"THAT'S NOT VERY MANY! TRY AND ROUND UP SO
ME MORE, PLEASE!":PRINT: GOTO 410
430 IF NP>6 THEN PRINT"I THINK I'D GET LOST TRYING TO REMEMBER A
LL THOSE WORDS...LET'S KEEP IT DOWN TO SIX PLAYERS THIS ROUND.":
PRINT: GOTO 410
440 DIM WD$(NP-1,34) 'ARRAY OF WORDS FORMED BY EACH PLAYER
450 DIM OS(6),NW(NP-1,6),NM$(NP-1),NS$(NP-1),SC(NP-1),TS!(NP-1),
CS!(NP-1),BN(6),WA(25),WW(25): NL=13
460 FOR I=1 TO NP
470 PRINT"WHAT'S PLAYER #";STR$(I);"'S NAME";:INPUT NM$(I-1): NE
XT I
480 FOR I=0 TO NP-1: IF RIGHT$(NM$(I),1)="S" THEN NS$(I)=NM$(I)+
"'" ELSE NS$(I)=NM$(I)+"'S"
490 NEXT I
500 GW$="AAAAAAAAABBCCDDDDEEEEEEEEEEEEFFGGGHHIIIIIIIIIJKLLLLMMNN
NNNOOOOOOOOPPQRRRRRRSSSSTTTTTTUUUUVVWWXYYZ"
510 UB$=STRING$(4,32)+CHR$(191)+STRING$(6,32)+CHR$(191)+STRING$(
7,32)+CHR$(191)+STRING$(8,32)+CHR$(191)+STRING$(9,32)+CHR$(191)+
STRING$(10,32)+CHR$(191)+STRING$(14,32)
520 KP=55
530 OS(0)=0: OS(1)=6: OS(2)=13: OS(3)=21: OS(4)=30: OS(5)=40: OS
(6)=51
540 BN(0)=0: BN(1)=5: FOR I=2 TO 6: BN(I)=BN(I-1)*2: NEXT I
550 AD=7: GN=1
560 BP=1750
```

Draw the screen template.

```
580 CLS:PRINT"GAME NUMBER";GN;: PRINT@64,"LETTERS: ";: PRINT@44,
"TIME LEFT: ";
600 PRINT@384,STRING$(64,176);: PRINT@768,STRING$(64,131);
610 FOR I=448 TO 704 STEP 64: PRINT@I,UB$;: NEXT I
```

Beginning of main game loop. Generate the roll of letters for this game, initialize necessary variables.

```
640 GOSUB 230
650 SB=0
660 FOR CP=0 TO NP-1
670 CL=180: PRINT@KP,CL;CHR$(30);
680 IF NP>1 THEN PRINT@138,"IT'S ";NS$(CP);" TURN...CLEAR THE AR
EA!";
690 PRINT@256,"HIT <ENTER> WHEN READY TO PLAY, ";NM$(CP);CHR$(30
);
700 A$=INKEY$: IF A$<>CHR$(13) THEN 700
710 PRINT@138,CHR$(30);: PRINT@256,CHR$(30);
720 FOR I=1 TO NL: PRINT@73+2*(I-1),MID$(RL$,I,1);: NEXT I
730 PRINT@320,NS$(CP);" WORDS:";: PRINT@364,"SCORE:";: IF NP>1 T
HEN PRINT@914,"SCORE TO BEAT: ";SB;CHR$(30);
740 BW=0
```

Get a word using input subroutine; verify that it has not been played before, has at least three letters, and contains only letters in the roll; also check number of words of that length.

```
750 FOR I=0 TO 25: WW(I)=WA(I): NEXT I: PRINT@192,"YOUR WORD, ";
NM$(CP);"? ";: PA=192+POS(0): PRINTCHR$(30);: W$="": GOTO 140
760 PRINT@192,"CHECKING ";W$;"...";CHR$(30);
780 FOR I=1 TO LEN(W$): L=ASC(MID$(W$,I,1))-65
790 WW(L)=WW(L)-1: IF WW(L)<0 THEN 810 ELSE NEXT I
800 GOTO 940
810 IF WA(L)=0 THEN M$="** THERE ARE NO "+MID$(W$,I,1)+"'S IN TH
IS ROLL **" ELSE M$="** YOUR WORD CONTAINS TOO MANY "+MID$(W$,I,
1)+"'S **"
820 GOSUB 320: GOTO 750
```

Clock has run out; sound the relay buzzer.

```
830 FOR XX=1 TO 50: OUT 255,4: OUT 255,1: NEXT XX: PRINT@152,"--
 TIME'S UP --";
840 IF CP<>NP-1 PRINT@910,"HIT <ENTER> TO SET UP FOR NEXT PLAYER
"; ELSE PRINT@910,"HIT <ENTER> TO REVIEW GAME SCORING";
```

End of game for current player. Blank his words from the screen, check if he achieved a new high score for this roll, and go on to next player if there is one.

```
860 A$=INKEY$: IF A$<>CHR$(13) THEN 860
870 FOR I=448 TO 704 STEP 64: PRINT@I,UB$;: NEXT I
880 PRINT@192,CHR$(30);: PRINT@320,CHR$(30);: PRINT@73,CHR$(30);
: PRINT@152,CHR$(30);: PRINT@896,CHR$(30);
890 IF SC(CP)>SB THEN SB=SC(CP)
900 NEXT CP
910 GOTO 1150
```

Word validation routines. If word is good, calculate its point value and add to score. Check to see if new total score qualifies player for double bonus mode.

```
940 TI=30: LW=LEN(W$): IF LW>8 THEN LW=9
950 IF LW<3 THEN M$="** WORDS MUST HAVE AT LEAST 3 LETTERS **":G
OSUB320: GOTO 750
960 IF NW(CP,LW-3)=5 THEN M$="** YOU ALREADY HAVE 5 WORDS OF THA
T LENGTH **": GOSUB 320: GOTO 750
970 NW=NW(CP,LW-3)
980 IF NW=0 THEN 1020
990 FOR J=5*(LW-3) TO 5*(LW-3)+NW-1
```

```
1000 IF WD$(CP,J)<>W$ THEN NEXT J: GOTO 1020
1010 M$="** SORRY, YOU ALREADY HAVE THAT ONE **": GOSUB 320: GOT
O 750
1020 NW=NW(CP,LW-3): NW(CP,LW-3)=NW+1
1050 WD$(CP,5*(LW-3)+NW)=W$
1060 IF LEN(W$)>13 THEN W$=LEFT$(W$,13)
1070 PRINT@448+64*NW+OS(LW-3),W$;
1080 SC=LW*10
1090 BN=BN(LW-3)*(NW+1)
1100 IF SC(CP)>BP THEN BN=BN+BN
1110 SC=SC+BN: SC(CP)=SC(CP)+SC: PRINT@371,SC(CP);CHR$(30);
1120 IF SC(CP)>BP AND BW=0 THEN M$="-- GREAT!! YOU JUST WENT INT
O DOUBLE BONUS TIME --": GOSUB 320: M$="-- ALL BONUS SCORES ARE
NOW DOUBLED --": GOSUB 320: M$="-- AND YOU GET 60 SECONDS OF EXT
RA TIME --": GOSUB 320: CL=CL+60: BW=-1
1130 GOTO 750
```

List player's words on screen for possible challenge by other players.

```
1150 FOR I=0 TO NP-1
1170  CLS:PRINT NS$(I);" WORDS:"
1180  FOR K=0 TO 4: SZ=3: PRINT@ (K+1)*128,;: FOR J=0 TO 34 STEP
 5: SZ=SZ+1
1190  IF WD$(I,J+K)="" THEN PRINT STRING$(SZ,32); ELSE PRINT WD$
(I,J+K);" ";
1200  NEXTJ: NEXTK: PRINT
1210  PRINT:PRINT"ANY OF ";NS$(I);" WORDS MAY NOW BE CHALLENGED.
"
1220 PRINT"ARE ALL OF ";NS$(I);:INPUT" WORDS OK (Y/N)";A$
1230 IF LEFT$(A$,1)="N" THEN PRINT@896,CHR$(30);"WHICH WORD IS I
NCORRECT";: INPUT W$ ELSE 1290
1240 FD=0: FOR XX=0 TO 34: IF WD$(I,XX)=W$ THEN FD=-1: WD$(I,XX)
="": GOTO 1250 ELSE NEXT XX
1250 LW=LEN(W$): IF LW>9 THEN LW=9
1260 IF FD THEN PN=LW*10+2*(BN(LW-3)*5)
1270 IF FD THEN PRINT@896,"SORRY, ";NM$(I);", THAT'LL COST YOU";
PN;"POINTS.";CHR$(30);:FORXX=1TO700:NEXTXX: SC(I)=SC(I)-PN: PRIN
T@896,"YOUR NEW SCORE IS ";SC(I);"POINTS.";CHR$(30);: FOR XX=1TO
700:NEXTXX: GOTO1170
1280 PRINT@896,"THAT'S NOT ONE OF ";NS$(I);" WORDS.";CHR$(30);:F
ORXX=1TO1000:NEXTXX: GOTO 1170
1290 NEXT I
```

List players in descending order of their scores for the previous game, and then by their cumulative scores for the current tournament.

```
1320 CLS: PRINTTAB(19)"SCORING FOR ROUND";GN:PRINT:PRINT"NAME";T
AB(53);"SCORE":PRINT"----";TAB(53);"-----"
1330 FOR I=0 TO NP-1: TS!(I)=SC(I): NEXT I
1340 FOR J=1 TO NP
1350 HS!=-1E36: W=0: FOR I=0 TO NP-1: IF TS!(I)>=HS! THEN HS!=TS
!(I): W=I:
1360 NEXT I: PRINT NM$(W);STRING$(52-POS(0),".");USING"######";S
C(W): TS!(W)=-1E36: NEXT J
1370 PRINT@974,"PRESS <ENTER> TO CONTINUE";
1380 A$=INKEY$:IF A$="" THEN 1380 ELSE IF A$=CHR$(13) THEN 1390
ELSE 1380
1390 CLS: IF GN=5 THEN PRINTTAB(15)"FINAL SCORES FOR TOURNAMENT:
" ELSE PRINTTAB(19)"CUMULATIVE SCORING:"
1400 PRINT:PRINT"NAME";TAB(53);"SCORE":PRINT"----";TAB(53);"----
-"
1410 FOR J=0 TO NP-1: CS!(J)=CS!(J)+SC(J): SC(J)=0: NEXT J
1420 FOR I=0 TO NP-1: TS!(I)=CS!(I): NEXT I
1430 FOR J=1 TO NP
1440 HS!=-1E36: W=0: FOR I=0 TO NP-1: IF TS!(I)>HS! THEN HS!=TS!
(I): W=I:
1450 NEXT I: PRINT NM$(W);STRING$(52-POS(0),".");USING"######";C
S!(W): TS!(W)=-1E36: NEXT J
```

End-game check.

```
1460 IF GN<5 THEN GN=GN+1: PRINT@969,"HIT <ENTER> WHEN READY FOR
 GAME #";GN;: INPUTA$: CLS: GOTO 1500
1470 PRINT@974,"ANOTHER TOURNAMENT (Y/N)? ";
1480 A$=INKEY$:IF A$="" THEN 1480 ELSE IF A$="Y" THEN PRINT A$;:
 GOTO 1490 ELSE IF A$="N" THEN PRINT A$:PRINT"THANKS FOR PLAYING
!":END ELSE 1480
1490 FOR I=0 TO 5: CS!(I)=0: NEXT I: NL=13: GN=1:
1500 FOR I=0 TO NP-1: FOR J=0 TO 34: WD$(I,J)="": NEXT J: FOR K=
0 TO 6: NW(I,K)=0: NEXT K: NEXTI: A$=INKEY$: A$="": IF GN=5 THEN
 NL=20
1510 GOTO 580
```

Print instructions.

```
1520 CLS:PRINT"THE OBJECT OF WORD WARS IS TO MAKE AS MANY WORDS
AS YOU CAN FROM";:PRINT"A GROUP OF LETTERS DISPLAYED ON THE SCRE
EN, WORKING AGAINST A":PRINT"TIME LIMIT OF ABOUT THREE MINUTES."
:PRINT
1530 PRINT"WORD WARS MAY BE PLAYED BY A SINGLE PLAYER, COMPETING
 AGAINST":PRINT"THE CLOCK AND TRYING TO GET THE HIGHEST POSSIBLE
 SCORE, OR BY":PRINT"UP TO SIX PEOPLE PLAYING AGAINST THE CLOCK
AND EACH OTHER.":PRINT
1540 PRINT"YOUR WORDS MUST CONTAIN AT LEAST THREE LETTERS.  THEY
 ARE":PRINT"GROUPED ON THE SCREEN ACCORDING TO LENGTH.  YOU CAN
MAKE UP TO":PRINT"FIVE WORDS OF EACH LENGTH FROM THREE TO EIGHT
LETTERS, AND FIVE"
1550 PRINT"WORDS WITH MORE THAN EIGHT LETTERS.": PRINT
1560 PRINT"NO PROPER NOUNS ARE ALLOWED, BUT CONTRACTIONS, PLURAL
S AND ALL":PRINT"PREFIXES AND SUFFIXES ARE FINE.": GOSUB 1680
1570 PRINT"A GROUP OF FIVE GAMES OR ROUNDS IS CALLED A TOURNAMEN
T.  IN THE":PRINT"FIRST FOUR GAMES, YOU MUST MAKE YOUR WORDS FRO
M A 'ROLL' OF":PRINT"13 LETTERS.  IN THE LAST GAME OF THE TOURNA
MENT, YOU WILL HAVE":PRINT"20 LETTERS TO WORK WITH.": PRINT
1580 PRINT"IF THERE IS MORE THAN ONE PLAYER, EACH GETS A THREE M
INUTE TURN":PRINT"WITH THE SAME SET OF LETTERS.  NATURALLY, THOS
E WHO HAVEN'T HAD":PRINT"THEIR TURN YET SHOULD NOT WATCH THE SCR
EEN WHILE OTHERS ARE":PRINT"MAKING THEIR WORDS.":PRINT
1590 PRINT"AFTER EACH PLAYER HAS HAD HIS OR HER TURN, EVERYONE'S
 WORDS ARE":PRINT"DISPLAYED IN TURN FOR POSSIBLE CHALLENGES FROM
 THE OTHERS.":PRINT"YOU WILL NEED A DICTIONARY OR SOME OTHER MEA
NS OF ARBITRATION":
1600 PRINT"FOR THIS PHASE!":GOSUB 1680
1610 PRINT"ALL PLAYERS ARE THEN RANKED ACCORDING TO THEIR SCORE
FOR THIS":PRINT"GAME, AND THEIR TOTAL SCORE SO FAR.": PRINT
1620 PRINT"A WORD ON SCORING.  THE BASIC SCORE FOR A WORD IS 10
POINTS":PRINT"TIMES THE NUMBER OF LETTERS IN THE WORD.  IN ADDIT
ION, A BONUS":PRINT"RANGING FROM 0 TO 800 POINTS IS ADDED TO THE
 SCORE OF EACH WORD."
1630 PRINT"THE MORE LETTERS IN A WORD, AND THE MORE WORDS OF THA
T SIZE YOU":PRINT"HAVE ALREADY, THE HIGHER THE BONUS.":PRINT
1640 PRINT"FINALLY, IF YOU GET MORE THAN 1750 POINTS IN A GAME,
YOU GO INTO";:PRINT"DOUBLE BONUS MODE.  THE BONUS FOR EACH -ADDI
TIONAL- WORD YOU":PRINT"MAKE IS DOUBLED.  YOU ALSO RECEIVE ANOTH
ER 60 SECONDS OF PLAYING";:PRINT"TIME.": GOSUB 1680
1650 PRINT"IF ANY OF YOUR WORDS ARE CHALLENGED AND FOUND INVALID
, YOU ARE":PRINT"PENALIZED AT THE HIGHEST POSSIBLE SCORE FOR A W
ORD OF ITS SIZE, THAT IS, AT THE BASE SCORE PLUS THE BIGGEST POS
SIBLE (DOUBLE)  BONUS.": PRINT
1660 PRINT"WHILE ENTERING A WORD, BACK-ARROW ERASES THE PREVIOUS
 CHARACTER,";:PRINT"AND SHIFT BACK-ARROW ERASES THE ENTIRE WORD,
 JUST AS IN NORMAL":PRINT"BASIC INPUT.  IF YOU ARE PLAYING ALONE
, YOU MAY GET A NEW ROLL":PRINT"BY PRESSING SHIFT-UP ARROW.":PR
INT
1670 PRINT"YOU MAY FIND IT FUN AT PARTIES TO LET TWO OR THREE PL
AYERS WORK":PRINT"TOGETHER AS A TEAM.":PRINT:PRINT"AT ANY RATE,
HAVE FUN AND IMPROVE YOUR VOCABULARY!": GOSUB 1680: RETURN
1680 PRINT@978,"HIT <ENTER> TO CONTINUE";
1690 A$=INKEY$: IF A$=CHR$(13) THEN CLS: RETURN ELSE 1690
```

THE NATIONAL COMPUTER SHOWS

HAVE WE GOT A PROGRAM FOR YOU IN '81 & '82

Attend the biggest public computer shows in the country. Each show has 100,000 square feet of display space featuring over 50 Million Dollars worth of software and hardware for business, industry, government, education, home and personal use.

You'll see computers costing $150 to $250,000 including mini and micro computers, software, graphics, data and word processing equipment, telecommunications, office machines, electronic typewriters, peripheral equipment, supplies and computer services.

All the major names are there including; IBM, Wang, DEC, Xerox, Burroughs, Data General, Qantel, Nixdorf, NEC, Radio Shack, Heathkit, Apple, RCA, Vector Graphic, and Commodore Pet. Plus, computerized video games, robots, computer art, electronic gadgetry, and computer music to entertain, enthrall and educate kids, spouses and people who don't know a program from a memory disk.

Don't miss the Coming Of The New Computers – Show Up For The Show that mixes business with pleasure. Admission is $5 for adults and $2 for children under 12 when accompanied by an adult.

Ticket Information

Send $5 per person with the name of the show you will attend to National Computer Shows, 824 Boylston Street, Chestnut Hill, Mass. 02167. Tel. 617 739 2000. Tickets can also be purchased at the show.

THE MID-WEST COMPUTER SHOW

CHICAGO
McCormick Place
SCHOESSLING HALL
23rd & THE LAKE
THURS-SUN
SEPT 10-13, 1981
11AM TO 7PM WEEKDAYS
11AM TO 6PM WEEKENDS

THE MID-ATLANTIC COMPUTER SHOW

WASHINGTON, DC
DC Armory/Starplex
2001 E. CAPITAL ST. SE
(E CAP. ST. EXIT OFF I 295
-KENILWORTH FRWY)
ACROSS FROM RFK
STADIUM
THURS-SUN
SEPT 24-27, 1981
11AM TO 7PM WEEKDAYS
11AM TO 6PM WEEKENDS

THE NORTHEAST COMPUTER SHOW

BOSTON
Hynes Auditorium
PRUDENTIAL CENTER
THURS-SUN
OCT 15-18, 1981
11AM TO 7PM WEEKDAYS
11AM TO 6PM WEEKENDS

THE SOUTHEAST COMPUTER SHOW

ATLANTA
Atlanta Civic Center
395 PIEDMONT AVE NE AT
RALPH McGILL BLVD
THURS-SUN
OCT 29-NOV 1, 1981
11AM TO 7PM WEEKDAYS
11AM TO 6PM WEEKENDS

THE SOUTHERN CALIFORNIA COMPUTER SHOW

LOS ANGELES
LA Convention Center
1201 SOUTH FIGUEROA
THURS-SUN
MAY 6-9, 1982
11AM TO 7PM WEEKDAYS
11AM TO 6PM WEEKENDS

Volleyball

by Jim Hilger

"Volleyball" is an Apple game requiring Applesoft and 16K RAM.

"Volleyball" is an action sports game for four players. Most games are designed for one or two players, probably because it's rare to have more than two game paddles. "Volleyball" sidesteps this limitation by allowing two of the players to use keyboard controls, while the other two players use the paddles. Allowing four to participate in the game creates an exciting level of team competition. And if you don't have four players available, it's even more of a challenge for two or three.

Introductory instructions are first displayed on the screen. Press RETURN, and the playing court will appear. Moments later the four high-resolution players take the court, and the game is on. The "^" mark beneath one side of the court indicates which team will serve.

The players on each team who are farther from the net are called the "back" players, while those closer to the net are called the "up" players. The back players are controlled by the paddle knobs. The left up player is controlled by the "A" and "S" keys, while the right up player is moved using the left and right arrow keys. The back players also control the serve, which is initiated by pressing the appropriate paddle button. The back and up players each cover their own zones of the court, and these zones do not overlap.

The up players have more control options than you might first suspect. If you hit the "A" key once, for instance, the left up player drifts left. Hit the "A" key again, and he moves left twice as fast. Hit the "S" key once to slow down, and hit it again to stop.

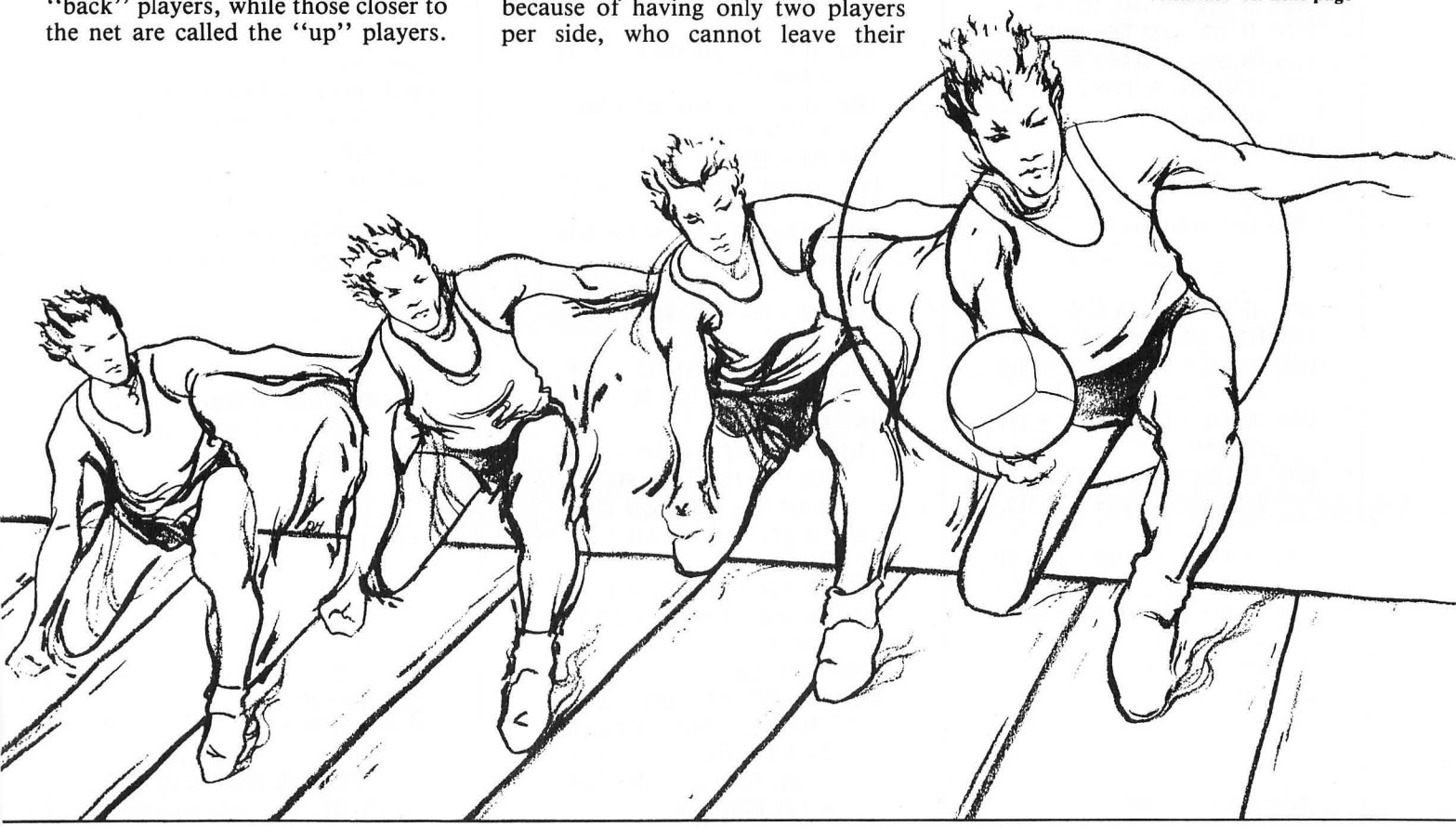
Normal volleyball rules are generally followed. The game is played to 21, and a team must win by two points. A team can score a point only when it has served. If a team fails to get the ball over the net after three hits, it loses the point. Contrary to regular rules, however, a single player is allowed to hit the ball more than once in succession; this is because of having only two players per side, who cannot leave their respective zones.

Have fun, and may the best team win!

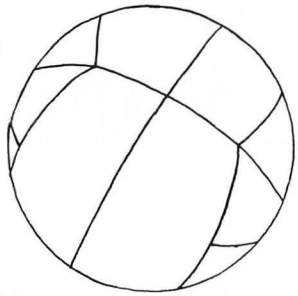
VARIABLES

A%: Ball take-off speed vertically when hit. A constant -24.
B%: Luck of the bounce (random) factor when ball is not hit well.
B1%: New X-coordinate for volleyball.
B2%: New Y-coordinate for volleyball.
BX%: Current X-coordinate for volleyball.
BY%: Current Y-coordinate for volleyball.
C%: Luck of the bounce (B%) with a sign added to determine direction.
H: Designates which team last hit the ball. H = 1 is left team; H = 2 is right team.
H3: Indicates team which has exceeded three-hit limit.
I: Multi-purpose indexing variable in FOR-NEXT loops.
I%: Variable that is read from DATA statements.

continued on next page



continued from previous page

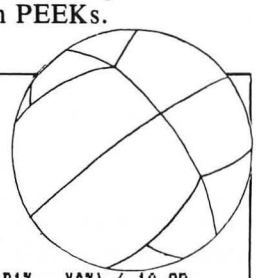


K: Keyboard key-pressed value.
LH: Indicates previous side to hit ball; used to decide whether to increment NH.

N%: Normalized ball X-position. Used to generate a musical tone when a ball is hit.
NH: Number of hits by current side.
R: Random number.
S: Designates which team is serving. S=1 is left team; S=2 is right team.
S1: Left team's score.
S2: Right team's score.
SB%: Left front man's speed.
SD%: Right front man's speed.

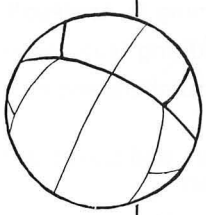
SX%: Horizontal speed of volleyball.
SY%: Vertical speed of volleyball.
W: Indexing variable for producing missed ball buzz.
X1%: New player position (used in all four player's move routines).
XA%: Left back man position.
XB%: Left front man position.
XC%: Right front man position.
XD%: Right back man position.
ZZ: Buzz variable, set equal to speaker location PEEKs.

```
 Program execution control.


500  POKE 116,32: POKE 115,0
1000  GOSUB 2380
1010  GOSUB 2300
1020  GOTO 1550
1030  GOSUB 1100
1040  GOSUB 1150
1050  GOSUB 1240
1060  GOSUB 1290
1070  GOSUB 1380
1080 K =  PEEK ( - 16384)
1090  POKE  - 16368,0: GOTO 1030


 Move left back man.


1100 X1% =  PDL (0) / 3
1110  IF X1% < 4 THEN X1% = 4
1120  IF X1% = XA% THEN 1140
1130  HCOLOR= 0: DRAW 2 AT XA%,11
     2: HCOLOR= 3: DRAW 2 AT X1%,
     112:XA% = X1%
1140  RETURN


 Move left front man.


1150  IF SB% = 0 THEN 1200
1160 X1% = XB% + SB%
1170  IF X1% < 94 THEN X1% = 94:S
     B% = 0
1180  IF X1% > 134 THEN X1% = 134
     :SB% = 0
1190  HCOLOR= 0: DRAW 3 AT XB%,11
     2: HCOLOR= 3: DRAW 3 AT X1%,
     112:XB% = X1%
1200  IF K <  > 193 AND K <  > 21
     1 THEN 1230
1210  IF K = 193 THEN SB% = SB% -
     4: GOTO 1230
1220 SB% = SB% + 4
1230  RETURN


 Move right back man.


1240 X1% =  PDL (1) / 3 + 190
1250  IF X1% < 194 THEN X1% = 194


1260  IF X1% = XD% THEN 1280
1270  HCOLOR= 0: DRAW 4 AT XD%,11
     2: HCOLOR= 3: DRAW 4 AT X1%,
     112:XD% = X1%
1280  RETURN


 Move right front man.


1290  IF SD% = 0 THEN 1340
1300 X1% = XC% + SD%
1310  IF X1% < 145 THEN X1% = 145
     :SD% = 0
1320  IF X1% > 185 THEN X1% = 185
     :SD% = 0
1330  HCOLOR= 0: DRAW 5 AT XC%,11
     2: HCOLOR= 3: DRAW 5 AT X1%,
     112:XC% = X1%
1340  IF K <  > 136 AND K <  > 14
     9 THEN 1370
1350  IF K = 136 THEN SD% = SD% -
     4: GOTO 1370
1360 SD% = SD% + 4
1370  RETURN

 Move volleyball and test for hits
 and misses.

1380 B1% = BX% + SX%:B2% = BY% +
     SY%
1390 B% = 8 *  RND (1):C% =  SGN
     (.5 -  RND (1)) * B%
1400 SY% = SY% + 4
1410  IF  ABS (B1% - 139) < 5 AND
     B2% > 90 THEN B1% = BX% - SX
     %:SX% =  - SX%: GOSUB 2240
1420  IF B2% > 105 AND BY% < 106 THEN
     1460
1430  IF B2% > 105 THEN 2020
1440  IF B1% < 4 OR B1% > 275 THEN
     2020
1450  GOTO 1508
1460  IF  ABS (B1% - XA%) < 6 OR
      ABS (B1% - XB%) < 6 THEN B1
     % = B1% + 4:SX% = 4 + B%:SY%
      = A%: GOSUB 2260:B2% = 105:
     H = 1: GOTO 1500
```



```
1470  IF  ABS (B1% - XA%) < 10 OR
      ABS (B1% - XB%) < 10 THEN B
     1% = B1% + 2:SX% = 2 + C%:SY
     % = 2 * A% / 3: GOSUB 2260:B
     2% = 105:H = 1: GOTO 1500
1480  IF  ABS (B1% - XC%) < 6 OR
      ABS (B1% - XD%) < 6 THEN B1
     % = B1% - 4:SX% =  - 4 - B%:
     SY% = A%: GOSUB 2260:B2% = 1
     05:H = 2: GOTO 1500
1490  IF  ABS (B1% - XC%) < 10 OR
      ABS (B1% - XD%) < 10 THEN B
     1% = B1% - 2:SX% =  - 2 - C%
     :SY% = 2 * A% / 3: GOSUB 226
     0:B2% = 105:H = 2: GOTO 1500

1495  GOTO 1508
1500  IF LH = H THEN NH = NH + 1:
      IF NH > 3 THEN H3 = H: GOTO
     2020
1505  IF LH >  < H THEN LH = H:NH
      = 1
1507  VTAB 21: HTAB 21: PRINT NH
1508  IF B2% > 105 THEN B2% = 115

1510  IF B1% < 4 OR B1% > 275 THEN
     2020
1520  HCOLOR= 0: DRAW 1 AT BX%,BY
     %: HCOLOR= 3: DRAW 1 AT B1%,
     B2%:BX% = B1%:BY% = B2%
1530  RETURN


 Game initialization.


1550 A% =  - 24
1560  HGR
1570  HOME
1580  VTAB 22
1590  HCOLOR= 1
1600  HPLOT 0,120 TO 279,120
1610  HCOLOR= 5: HPLOT 0,121 TO 2
     79,121
```

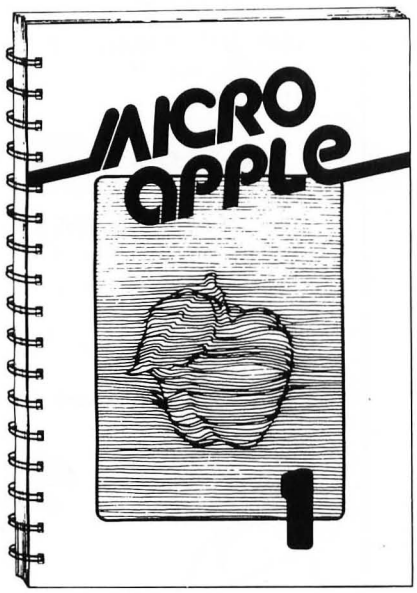
```
1620  HCOLOR= 1: HPLOT 0,122 TO 2
      79,122
1630  HCOLOR= 1: HPLOT 0,123 TO 2
      79,123
1640  HCOLOR= 3
1650  ROT= 0: SCALE= 1
1660  HPLOT 140,90 TO 140,118
1670  HPLOT 139,90 TO 139,118
1680  GOSUB 2400
1690  VTAB 22: CALL  - 868
1700  POKE 232,0: POKE 233,27
1710 XA% =  PDL (0) / 3
1720  IF XA% < 4 THEN XA% = 4
1730  DRAW 2 AT XA%,112
1740 XB% = 110
1750  DRAW 3 AT XB%,112
1760 XD% =  PDL (1) / 3 + 190
1770  IF XD% < 194 THEN XD% = 194

1780  DRAW 4 AT XD%,112
1790 XC% = 169
1800  DRAW 5 AT XC%,112
1810 S1 = 0:S2 = 0
1820  VTAB 22: HTAB 10: PRINT S1:
      VTAB 22: HTAB 30: PRINT S2
1830 SB% = 0:SD% = 0
1840 K = 0
1850 S = 1
1860 R =  RND (1): IF R < .5 THEN
     S = 2
1870  GOSUB 1890
1880  GOTO 1030
```



```
 Serve the volleyball.

1890 B% = 8 *  RND (1)
1900  IF S = 1 THEN BX% = 85:BY% =
     100:SX% = 5 + B%:SY% = A%: VTAB
     21: HTAB 2: PRINT "^": VTAB
     21: HTAB 38: PRINT " "
1910  IF S = 2 THEN BX% = 194:BY%
      = 100:SX% =  - (5 + B%):SY%
      = A%: VTAB 21: HTAB 2: PRINT
     " ": VTAB 21: HTAB 38: PRINT
     "^"
1920 H = S
1930  IF S = 1 THEN I =  PEEK ( -
     16287): IF I < 128 THEN 1930

1940  IF S = 2 THEN I =  PEEK ( -
     16286): IF I < 128 THEN 1940

1950  GOSUB 2240
1960 SB% = 0:SD% = 0
1965 H3 = 0:LH = 0:NH = 0
1967  VTAB 21: HTAB 21: PRINT " "

1970 K = 0: POKE  - 16368,0
1980  HCOLOR= 3
1990  HPLOT 140,90 TO 140,118
2000  HPLOT 139,90 TO 139,118
2010  RETURN

 Scoring routine.

2020  IF H3 = 1 THEN 2080
2022  IF H3 = 2 THEN 2040
2025  IF BX% < 139 THEN 2070
2030  IF H = 1 AND B1% > 275 THEN
     2080
2040  IF S = 1 THEN S1 = S1 + 1
2050 S = 1
2060  GOTO 2100
2070  IF H = 2 AND B1% < 4 THEN 2
     040
2080  IF S = 2 THEN S2 = S2 + 1
2090 S = 2
2100  FOR W = 1 TO 20: GOSUB 2240
      : NEXT
2110  VTAB 22: HTAB 10: PRINT S1:
      VTAB 22: HTAB 30: PRINT S2
2120  IF (S1 > 20 OR S2 > 20) AND
      ABS (S1 - S2) > 1 THEN 2150
2130  HCOLOR= 0: DRAW 1 AT BX%,BY
     %: HCOLOR= 3
2140  POP : GOSUB 1890: GOTO 1030


 End-of-game festivities.

2150  VTAB 21: HTAB 19: FLASH : PRINT
     "GAME"
2160  VTAB 22: HTAB 19: PRINT "OV
     ER"
2170  NORMAL
2180  HCOLOR= 2: HPLOT 10,10 TO 2
     0,20: CALL 62454: HCOLOR= 0:
      FOR I = 120 TO 123: HPLOT 0
     ,I TO 279,I: NEXT : HPLOT 13
     9,90 TO 139,118: HPLOT 140,9
     0 TO 140,118
2190  FOR I = 150 TO 159: HPLOT 0
     ,I TO 279,I: NEXT
2200  FOR I = 1 TO 3: CALL  - 198
      : NEXT
2210 K =  PEEK ( - 16384): IF K <
     128 THEN 2210
2220  POKE  - 16368,0: IF K <  >
     141 THEN 2210
2230  GOTO 1550

 Buzzer.

2240 ZZ =  PEEK ( - 16336) +  PEEK
     ( - 16336) -  PEEK ( - 16336
     )
2250  RETURN

 Standard music tones with pitch
 based on volleyball's horizontal
 position.

2260 N% = B1%: IF N% > 255 THEN N
     % = 255
2270  IF N% < 4 THEN N% = 4
2280  POKE 776,N%: POKE 777,10: CALL
     778
2290  RETURN

 Game introduction page.

2300  TEXT : HOME : VTAB 10: HTAB
     10: PRINT "4-PLAYER VOLLEYBA
     LL"
2310  HTAB 13: PRINT "BY JIM HILG
     ER"
2320  VTAB 20: PRINT "PADDLES CON
     TROL OUTER MEN.  A & S KEYS"
      : PRINT "AND ARROW KEYS WILL
       MOVE INNER MEN."
2330  PRINT "SIDE WITH ^ MARK PUS
     H PADDLE BUTTON TO": PRINT "
     SERVE.     HIT RETURN TO BEG
     IN MATCH!"
2340  FOR I = 10 TO 1 STEP  - 1:B
     1% = I * 10: GOSUB 2260: NEXT
      : FOR I = 1 TO 50: NEXT : FOR
     I = 1 TO 10:B1% = I * 10: GOSUB
     2260: NEXT
2350 K =  PEEK ( - 16384): IF K <
     128 THEN 2350
2360  POKE  - 16368,0: IF K <  >
     141 THEN 2350
2370  RETURN

 Pokes for Machine Language tone
 routine.

2380  POKE 776,255: POKE 777,255:
       POKE 778,173: POKE 779,48: POKE
      780,192: POKE 781,136: POKE
      782,208: POKE 783,5: POKE 78
      4,206: POKE 785,9: POKE 786,
      3: POKE 787,240
```

continued on page 63

GET MORE OUT OF YOUR APPLE WITH MICRO on the APPLE



MICRO/Apple

Over 30 Apple Programs on Diskette — For Less Than $1.00 Apiece! No Need to Type In Hundreds of Lines of Code!

224-page book and diskette $24.95*

* Add $2.00 for surface shipping. Massachusetts residents add 5% for sales tax.

MICRO's new book for Apple II users lets you

- Speed up programming in Applesoft and Integer BASIC!
- Add Apple II Plus editing features — at no cost!
- Round and format numbers accurately in business applications!
- Get lowercase letters and punctuation into Applesoft strings — at no cost!
- Do a shape table easily and correctly!
- Play the hit game "Spelunker"!
- And much, much more!

With MICRO/Apple 1, the first volume in our new series, you receive

- 30 choice articles from MICRO, The 6502 Journal (1977-80), complete with listings, all updated by the authors or MICRO staff,

plus

- 38 tested programs on diskette (13 sector, 3.2 DOS format, convertible to 3.3).

Ask for MICRO/Apple at your computer store or

Call Toll-free 800-227-1617, Ext. 564

In California, call 800-772-3545, Ext. 564

VISA and Mastercard Accepted

MICRO
P.O. Box 6502
Chelmsford, Massachusetts 01824

continued from page 61

```
2390  POKE 788,9: POKE 789,202: POKE
      790,208: POKE 791,245: POKE
      792,174: POKE 793,8: POKE 79
      4,3: POKE 795,76: POKE 796,1
      0: POKE 797,3: POKE 798,96: RETURN


 Poke in hi-resolution shapes from
 DATA statements.


2400  FOR I = 1 TO 284: READ I%: POKE
      6912 + I - 1,I%: NEXT I: RESTORE
      : RETURN
2410  DATA  5,0,12,0,42,0,98,0,16
      3,0,219,0,27,27,18,10,41,45,
      13,24
2420  DATA  31,27,59,40,9,9,9,60,
      27,27,27,12,13,9,41,24,59,63
      ,31,0
2430  DATA  0,24,27,27,18,18,18,4
      5,13,45,5,24,27,31,59,8,13,4
      5,9,24
2440  DATA  27,59,63,8,45,13,9,24
      ,27,63,63,44,45,45,9,24,59,6
      3,63,8
2450  DATA  41,9,45,24,31,59,63,8
      ,45,13,41,24,59,27,27,0,0,0,
      27,27
2460  DATA  18,18,18,45,13,9,45,2
      8,31,27,27,7,8,41,13,45,1,24
      ,27,63
2470  DATA  31,3,8,41,45,45,1,24,
      59,63,63,3,8,41,45,45,1,24,3
      1,59
2480  DATA  27,7,40,13,45,13,37,3
      1,59,63,27,44,9,9,9,33,63,31
      ,27,63
2490  DATA  4,0,0,27,27,18,18,18,
      41,45,41,37,59,27,31,3,8,9,4
      5,41
2500  DATA  24,63,31,27,8,9,41,45
      ,56,63,63,27,8,9,45,45,28,63
      ,63,31
2510  DATA  8,45,9,13,24,63,31,59
      ,8,13,41,45,24,27,27,31,0,0,
      0,27
2520  DATA  27,18,18,18,45,13,9,4
      5,28,31,27,27,7,8,41,13,45,1
      ,24,27
2530  DATA  63,31,3,8,41,45,45,1,
      24,59,63,63,3,8,41,45,45,1,2
      4,31
2540  DATA  59,27,7,40,13,45,13,3
      7,31,59,63,27,44,9,9,9,33,63
      ,31,27
2550  DATA  63,4,0,0,3,0,0,0,194,
      216,0,194,0,0,0,194,217,0,10
      0,0
```

ADVENTURE OF THE MONTH CLUB

ALIEN ADVENTURE

JULY ADVENTURE OF THE MONTH

You are the sole survivor of a crew on a mission to deliver a cargo of oil to Earth from a solar system far away. A crash landing has left you stranded on a small, harshly alien planet, which has as its only redeeming value, a bountiful supply of lead, gold and platinum.

You must now find provisions and a means of leaving the planet (and stock up on all the loot you can!). But beware of the THING that massacred your crew!

Alien Adventure, by Alan Zett, is modelled on the hit movie "Alien" and allows you three levels of play. Happy Adventuring!

What is Adventure of the Month?

Everybody likes Adventures — they're challenging and entertaining every time you play. But too often, pre-programmed cassettes and disks cost upwards of $35, a price the manufacturer must charge to defray promotional and packaging costs.

On the other hand, you can enter Adventures yourself, but when you do, you type away all the surprises. As a result, the game loses some of its challenge.

At **SoftSide**, we've found a way to beat the high cost of Adventuring without having to miss out on any of the fun. We're offering no-frill Adventures — high quality Adventures — on cassette or disk at an almost unheard-of price: **$5 for cassette, $8 on disk.**

We save you money by only advertising this offer to **SoftSide** readers (you won't see us anywhere else) and by foregoing fancy packaging and documentation — you'll get the software and only the software, but we believe it's as good as the $50 packages.

You'll save even more by joining the Adventure of the Month Club.

Here's how it works: **SoftSide**'s editorial department will select an original BASIC language Adventure each month and make it available to you on a subscription basis:

6 months on cassette: just $27
6 months on disk: just $45

Every month we'll tell you about the Adventure you'll be getting in **SoftSide** Magazine. To order, use the convenient order card in this issue — fill it out and send, with payment to:

Adventure of the Month Club
Department 681
6 South Street
Milford, NH 03055

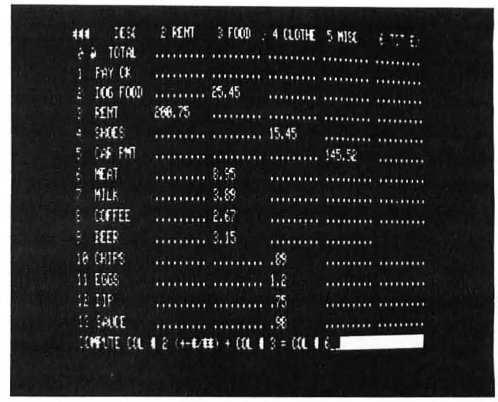
COLUMN CALCULATOR 4.1

by David T. Gray

COLUMN CALCULATOR is a ''word processor for numbers,'' a number processor designed to be used like a calculator. But it can handle large blocks of information as if handling one number at a time. The work space can be thought of as a large matrix with rows and columns much like an accountant's spreadsheet. Data can be easily entered into columns; and the columns can then be moved around. Columns can be overlaid from an existing data file on disk. One column can be added, subtracted, multiplied, divided, or raised to a power of another and the results put in another column. Columns can be compared to one another. Columns can be totalled, or set with a constant, and any column can be sorted, carrying the rest of the columns with it. A predefined function can be defined, thereby preprogramming the worksheet.

The statistical section provides analysis of the data. The analysis includes simple statistics, linear regression, simple correlation, histogram and the T-test.

The information can be printed out on the line-printer in a compressed format at any stage in the development of a data base. Thus, it can be used as a finished report or as a copy of the worksheet to permit the filling in of additional data for later entry into the data base. The data base can be saved on disk and recalled at a later date for modification or for generating a report. Any column in a file on disk can be referenced and added to the current worksheet.



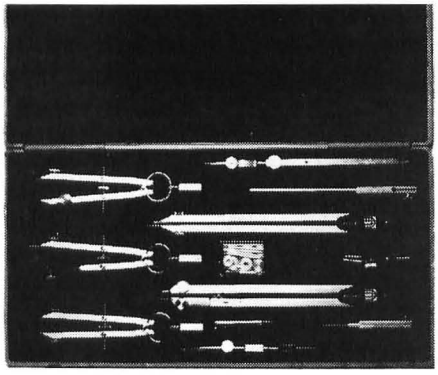
All user communication with COLUMN CALCULATOR uses FLASH, the line input/editor routine. This enables the user to not only key in instructions, but to edit errors or data as well.

S-80, 32K disk $39.95

Microdrafter

by Jon Voskuil

"Microdrafter" is an Apple program requiring Applesoft and 24K RAM, but is adaptable for 16K RAM.



Trample your T square and triangles, crumple your compass, pulverize your protractor, and fricassee your French curves — the age of computerized drafting is here!

Using the Apple's high-resolution graphics screen, this program allows you to simulate the use of compass, protractor, straightedge, ruler, freehand curves, etc., and create any kind of drawing you want, all in living color. Drawings may be saved to either tape or disk at any time, to be displayed or reworked later.

In contrast to other Hi-Res drawing programs that you may have used, this one allows you to work in any of three different modes to create a drawing. Each of them has its unique advantages, and together they make possible a wide range of drawing capabilities. Each uses the two paddles, plus various keys on the keyboard, to control the position, movement, and color of the drawing cursor. The three modes are Compass, Straightedge, and Freehand.

THE COMPASS MODE

The Compass mode is best suited for drawing circles, arcs, spirals, and some other types of curved lines. Upon running the program, and after paging through the instructions, this is the mode in which you will find yourself. You'll see a display of various numbers, letters, and words at the bottom of the screen, with the remainder of the screen blank except possibly for one dot. If you turn Paddle 0 fully counterclockwise, the dot will locate itself in the middle of the blank area. The display at the bottom will then tell you that the compass center is at coordinates 121,80 and the radius (distance between the center and the drawing "pencil") is 0. Turning Paddle 0 clockwise will move the dot away from the center in some direction, depending on the position of Paddle 1. Turning Paddle 1 will rotate the point in a full circle around the center point, at whatever radius you set with Paddle 0, just as if you were pivoting a real compass.

The compass center can be moved anywhere on the screen (or even off the screen) by using the I, J, K, and L keys to move up, left, right, and down respectively. You'll notice that it moves in steps of four units. This increment can be changed to one or five or any number, by changing the 4s in lines 240-270. In fact, you may want to change only two of the lines (either 240 and 260, or 250 and 270), say to the number 3. This would allow you to place the center at any point on the screen (rather than just by steps of four) with the proper combination of back-and-forth keystrokes, while still allowing fairly fast movement over long distances. Incidentally, notice that the bottom left corner of the drawing screen is the 0, 0 point, which fits better with most people's thinking than the top left.

Once you decide that you want to draw with the compass, pressing "D" will toggle you into "Draw" status; pressing "D" again will return you to "Nodraw". If you want to draw a complete circle, you can save yourself the trouble of slowly turning Paddle 1, and instead press "0" (regardless of the Draw/Nodraw status). To change colors at any time (except during the automatic drawing of a circle), press "C". The color names may not precisely correspond to the colors you see on your TV, in which case you may want to change line 30 appropriately. If you do, be sure to make each name exactly six characters, abbreviating or filling in with spaces as necessary.

The compass mode is obviously suited for drawing circles and arcs, but it's also suited for drawing other types of curves. Spirals, for example, can be drawn by steadily changing the radius with Paddle 0, while turning the compass in a full circle with Paddle 1. The spiral is continued by toggling to "Nodraw", turning Paddle 1 all the way back to its opposite extreme, toggling back to "Draw", and then continuing as with the first revolution. Other types of curves, as well, may be more easily drawn in this mode than in the Freehand mode described below; experimentation will be your best guide.

THE STRAIGHTEDGE MODE

Hitting the ESC key shifts you into the second mode, Straightedge. This mode allows you to position two points anywhere on the screen and then draw a straight line between them. Upon entering the mode, both points are together, at a place determined by the paddle settings. Adjusting the paddles will separate the second point from the first, moving it around the screen. Pressing "P" will fix that second point at its current position, and then allow you to manipulate the first one to some other place. Each press of the "P" key will fix the point which is currently mobile, bring the other point to the same location, and allow you in turn to move it around the screen. Pressing "D" will draw a straight line of the current color between them.

One of the strengths of this mode is found in the display at the bottom of the screen, which shows two facts about the line segment defined by your two endpoints. It displays the length of the line segment (in the same units in which the circle radius and the X, Y coordinates are measured), and the angle which the line makes with the horizontal. This makes it easy to draw lines of precise lengths and angles, as you need to do constantly in making almost any kind of mechanical or perspective drawing. You can also use this feature to measure existing distances and angles already drawn on the screen, just as you would use a ruler and a protractor.

THE FREEHAND MODE

Pressing ESC once more will shift you into the third drawing mode, Freehand. This mode uses the paddles to control the X and Y coordinates of the cursor in a freestyle manner. The coordinates are

displayed and updated continuously on the screen, and again you can toggle between Draw and Nodraw, and change colors at will. This is the simplest of the three modes from the programmer's viewpoint, but the most difficult to use skillfully from the user's viewpoint. Nevertheless, there are always times when a freehand approach is the only way to draw. A joystick in place of the dual paddles would be a tremendous asset here (and would be an interesting alternative in the other modes as well). Pressing ESC will again return you to the Compass mode.

OTHER FEATURES

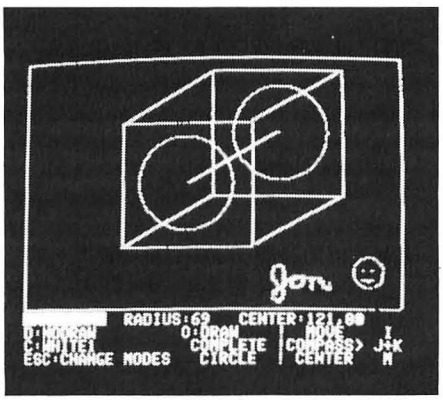
Each time the program is RUN, you will be asked if there is a drawing you'd like to get from a tape or a disk. An affirmative answer will send you to the appropriate part of the program, and prompt you to do the right things to retrieve the saved picture. Retrieval from tape is made simpler by using the Apple's screen editing feature: You simply use the right-arrow key to copy the appropriate commands.

A picture in progress may be saved at any time by pressing CTRL-S. This again will prompt you to do the proper things for saving onto tape or disk. The picture is saved by simply dumping the contents of memory from hex address 2000 through 3FFF (decimal 8192 through 16383). Two other control-keys are also recognized at any time: CTRL-E will erase the current picture from the screen and from memory, and CTRL-Q will quit the program and return you to BASIC.

PROGRAMMING NOTES

As written, "Microdrafter" requires at least a 24K Apple. The program itself fits in just under 8K, which is just short of the beginning of the Hi-Res graphics screen memory. To prevent variables from occupying space within that screen memory area, LOMEM is set just above it, at 16384. Which, if you have only 16K of RAM, leaves you up a creek without a paddle. However, you can use the program on a 16K Apple by doing the following: (1) Change "LOMEM: 16384" in line 5 to "HIMEM: 8191". (2) Delete or condense some of the instructions in lines 4000-4330. Deleting lines 4120-4320 and 7000-7020, and dropping the "GOSUB 7000" from line 4110, would be one way to do the trick; you may be able to get by with deleting less.

There is a scaling adjustment built into the program, which you will probably need to change to fit your TV or monitor. To calibrate the program, enter the Compass mode and draw a fairly large circle. Take a ruler and measure the roundness of the circle on the screen. If it's not perfectly round, you can adjust the value of Z in line 20 to correct the condition. If it's too fat, reduce the value of Z; if it's too skinny, increase Z. This won't correct non-linear distortion in your monitor, but it will at least correct for the varying amounts of "overscan" on different TVs.



Note that lines 50 and 6000 have attached REMs that warn you against changing their line numbers. This is important because of the references to them in the PRINT statements of lines 5140 and 5670. If you change those PRINTs (which a renumbering utility would NOT do by itself), then it's OK to change the line numbers to match.

If you're wondering about all the seemingly trivial variables that are initialized in lines 15-25, these are used in order to maximize speed. It takes less time to retrieve the value of a variable (especially one initialized early) than to convert a decimal number to its binary floating-point equivalent. And in a program like this one, speed is an important consideration; as it is, it requires some patience to draw a really solid freeform line.

VARIABLES

A: Angle.
C: ASCII value of "C" plus 128 (i.e., the value returned by the keyboard PEEK when "C" is pressed).
C$(*): Color names.
CC: Current plotting color.
CE, CQ, CS: Values of CTRL-E, CTRL-Q, and CTRL-S (see "C").
CX, CY: X and Y coordinates of compass center.
D: Value of D key.
D$(*): Draw/nodraw label.
DD: Draw/nodraw status (1 or 0).
ESC: Value of ESC key.
F$: Name of disk file.
II: Loop variable.
I, J, K: Value of I, J, and K keys.
K1: Keyboard buffer PEEK address.
K2: Keyboard buffer clear address.
KEY: Value of last key pressed.
L: Length of line segment.
M: Value of M key.
MODE: Current drafting mode.
N0 - NY: Variables used in place of constants such as 0, 96, etc.
O, P: Values of O and P keys.
PP: Mobile line endpoint number (1 or 2).
R: Circle radius.
S: Circle-drawing increment.
SX, SY: X and Y scaling factors.
X: X plotting coordinate.
X$: Input variable.
X(*): X coordinates of line endpoints.
XF: X scale adjustment.
XO: Old X coordinate.
X, Y (*), YO: Y equivalent of X values.
YP: Y-distance up from bottom of screen.
Z: Vertical/horizontal scaling calibration.

```
Initialization of variables.
5  LOMEM: 16384: HGR : TEXT : HOME
10  DIM X(2),Y(2),C$(7),D$(1)
15 N5 = .5:NK = 128:K1 =  - 16384
     :K2 =  - 16368:N6 = 6.283085
     3:N40 = 40.5851565:N140 = 14
     0:N96 = 96:N27 = 2.7:N0 = 0:
     NX = 280:NY = 160:NR = 2.55:
     SX = 1.0941:SY = .6235:XF =
     .84
```

```
20 Z = 1.15
25 C = 195:D = 196:M = 205:P = 20
     8:I = 201:J = 202:K = 203:ES
     C = 155:CE = 133:CQ = 145:CS
      = 147:O = 207
30 C$(1) = "GREEN ":C$(2) = "VIOL
     ET":C$(3) = "WHITE2":C$(4) =
     "BLACK ":C$(5) = "ORANGE":C$
     (6) = "BLUE  ":C$(7) = "WHIT
     E1"
```

```
35 D$(0) = "NODRAW":D$(1) = "DRAW
     "
40 CC = 7: HCOLOR= CC: GOSUB 4000

 Clear text and switch to hi-res
 graphics screen.

50  HOME : POKE  - 16304,0:MODE =
     1: GOTO 1000: REM --> DON'T
     CHANGE THIS LINE #!!
```

continued on next page

continued from previous page

```
Subroutine to process commands.

100 KEY =  PEEK (K1): POKE K2,0
110  IF MODE = 2 AND KEY = D THEN
      HPLOT X(1),Y(1) TO X(2),Y(2
     ): RETURN
120  IF KEY = D THEN DD = 1 - DD:
      RETURN
130  IF KEY <  > C THEN 160
140 CC = CC + 1: IF CC = 8 THEN C
     C = 1
150  HCOLOR= CC: RETURN
160  IF KEY <  > ESC THEN 220
170  IF MODE = 1 AND X > 0 AND X <
     280 AND Y > 0 AND Y < 160 THEN
      HCOLOR= 0: HPLOT X,Y: HCOLOR=
     CC
180  IF MODE = 2 THEN  HCOLOR= 0:
      HPLOT X(1),Y(1): HPLOT X(2)
     ,Y(2): HCOLOR= CC
190  HOME :MODE = MODE + 1: IF MO
     DE = 4 THEN MODE = 1
200  POP
210  ON MODE GOTO 1000,2000,3000
220  IF MODE = 2 AND KEY = P THEN
     PP = 3 - PP: HCOLOR= 0: HPLOT
     X(PP),Y(PP): HCOLOR= CC: RETURN
230  IF MODE <  > 1 THEN 290
240  IF KEY = J THEN CX = CX - 4
250  IF KEY = K THEN CX = CX + 4
260  IF KEY = I THEN CY = CY - 4
270  IF KEY = M THEN CY = CY + 4
280  HTAB 31: PRINT "         ";:
      HTAB 31: PRINT CX;",";160 -
     CY;
290  IF KEY = CE THEN  HGR : RETURN

300  IF KEY = CQ THEN  TEXT : HOME
     : END
310  IF KEY = CS THEN  GOTO 5500
320  IF KEY <  > 0 THEN 360
330 S = 50 / (R + 1)
340  FOR II = 0 TO 255 STEP S:A =
     N6 - II / N40:X = Z * (CX +
     R *  SIN (A)):Y = CY + R *  COS
     (A): IF X > N0 AND X < NX AND
     Y > N0 AND Y < NY THEN  HPLOT
     X,Y
350  NEXT
360  RETURN


 Compass mode:
 Print summary of commands at
 bottom of screen.


1000 CX =  INT (140 / Z):CY = 80:
     DD = 0
1010  VTAB 21: INVERSE : PRINT "
     COMPASS ";: NORMAL : HTAB 12
     : PRINT "RADIUS:";: HTAB 24:
      PRINT "CENTER:";CX;",";160 -
     CY
1020  VTAB 22: PRINT "D:
           O:DRAW    !  MOVE    I
      C:                COMPLETE
     !COMPASS> J+KESC:CHANGE MODE
     S   CIRCLE   ! CENTER   M";

 Print draw status and current
 color.

1030  VTAB 22: HTAB 3: PRINT D$(D
     D);: VTAB 23: HTAB 3: PRINT
     C$(CC);: VTAB 21


 Read paddles; calculate radius and
 angle.

1040 R =  PDL (0) / NR:A = N6 -  PDL
     (1) / N40:X = Z * (CX + R *
      SIN (A)):Y = CY + R *  COS
     (A): IF DD THEN 1060


 Erase last point if in 'nodraw'
 status.


1050  HCOLOR= 0: HPLOT XO,YO: HCOLOR=
     CC


 Plot new point.


1060  IF X > N0 AND X < NX AND Y >
     N0 AND Y < NY THEN  HPLOT X,
     Y:XO = X:YO = Y


 Print radius and check for
 keypress.


1070  HTAB 19: PRINT  INT (R + N5
     );"  ";: IF  PEEK (K1) < NK THEN
     1040
1080  GOSUB 100: GOTO 1030


 Straightedge mode:
 Print commands summary.


2000 PP = 1
2010  VTAB 21: INVERSE : PRINT "
     STRAIGHTEDGE ";: NORMAL : HTAB
     19: PRINT "LENGTH:";: HTAB 3
     1: PRINT "ANGLE:";
2020  VTAB 22: HTAB 1: PRINT "D:D
     RAW LINE            P:FIX EN
     DPOINT   C:
           & MOVE OTHER  ESC:CHA
     NGE MODES          ENDPOINT"
     ;
 Print current color.


2030  VTAB 23: HTAB 3: PRINT C$(C
     C);: VTAB 21


 Compute and plot new endpoint;
 erase old endpoint.


2040 X(3 - PP) =  PDL (0) * SX:Y(
     3 - PP) =  PDL (1) * SY
2050 X(PP) =  PDL (0) * SX:Y(PP) =
      PDL (1) * SY: HPLOT X(3 - P
     P),Y(3 - PP): HCOLOR= 0: HPLOT
     XO,YO: HCOLOR= CC:XO = X(PP)
     :YO = Y(PP): HPLOT XO,YO


 Compute and print length and angle
 of line segment defined by
 endpoints.


2060 X = XF *  ABS (X(1) - X(2)):
     Y =  ABS (Y(1) - Y(2)):L =  INT
     ( SQR (X * X + Y * Y) + N5):
      IF X < .00001 THEN A = 90: GOTO
     2080
2070 A =  INT ( ATN (Y / X) * 57.
     2958 + .5)
2080  HTAB 26: PRINT L;"  ";: HTAB
     37: PRINT A;" ";: IF  PEEK (
     K1) < NK THEN 2050
2090  GOSUB 100: VTAB 23: HTAB 3:
      PRINT C$(CC);: VTAB 21: GOTO
     2040


 Freehand mode:
 Print commands summary.


3000  VTAB 21: INVERSE : PRINT "
     FREEHAND ";: NORMAL : HTAB 2
     0: PRINT "X=";: HTAB 30: PRINT
     "Y=";
3010  VTAB 22: HTAB 1: PRINT "D:"
     : PRINT "C:": PRINT "ESC:CHA
     NGE MODES";


 Print draw status and current
 color.


3020  VTAB 22: HTAB 3: PRINT D$(D
     D);: VTAB 23: HTAB 3: PRINT
     C$(CC);: VTAB 21


 Compute new plot position.
```

```
3030 X =  PDL (0) * SX:Y =  PDL (
     1) * SY:YP = 159 - Y: HTAB 2
     2: PRINT  INT (X / Z);"  ";:
      HTAB 32: PRINT  INT (Y);"
     ";: IF DD THEN 3050
```

Erase old point if in 'nodraw' status.

```
3040  HCOLOR= 0: HPLOT XO,YO: HCOLOR=
     CC
```

Plot new point, check for keypress.

```
3050  HPLOT X,YP:XO = X:YO = YP: IF
      PEEK (K1) < NK THEN 3030
3060  GOSUB 100: GOTO 3020
```

Subroutine to print instructions.

```
4000  HOME : VTAB 10: PRINT  TAB(
     8)"M I C R O - D R A F T E R
     ": VTAB 14: PRINT  TAB( 13)"
     BY JON VOSKUIL"
4010  FOR II = 1 TO 1000: NEXT II
```

Paint screen white, move text window one space in from left margin, and print first page of instructions.

```
4020  HOME : INVERSE : FOR II = 1
      TO 23: PRINT  SPC( 40);: NEXT
     II: POKE 33,39: POKE 32,1
4030  VTAB 3: PRINT "   MICRODRAF
     TER IS A PADDLE-CONTROLLED":
      PRINT
4040  PRINT "DRAFTING BOARD WITH
     3 DIFFERENT MODES:": PRINT
4050  PRINT "  1. COMPASS - TO DR
     AW CIRCLES, ARCS,          S
     PIRALS, AND OTHER CURVES": PRINT

4060  PRINT "  2. STRAIGHTEDGE -
     TO DRAW STRAIGHT           L
     INES BETWEEN POINTS": PRINT
     : PRINT "  3. FREEHAND": PRINT
4070  PRINT "YOU MAY SWITCH MODES
      AT ANY TIME WHILE": PRINT
4080  PRINT "WORKING BY PRESSING
     'ESC'.": PRINT
4090  PRINT "   USE CTRL-E TO ERA
     SE, CTRL-S TO SAVE": PRINT
4100  PRINT "ONTO TAPE/DISK, CTRL
     -Q TO QUIT."
4110  NORMAL : VTAB 24: HTAB 2: PRINT
     ")>> PRESS SPACE BAR TO CONT
     INUE <<(";: GOSUB 7000
```

Print second page of instructions.

```
4120  VTAB 2: HTAB 12: NORMAL : PRINT
     "IN COMPASS MODE:"
4130  INVERSE : VTAB 5: PRINT "TH
     E COMPASS POINT IS MOVED USI
     NG THE": PRINT "I, J, K, AND
      M KEYS."
4140  PRINT : PRINT "PRESSING 'D'
      TOGGLES BETWEEN 'DRAW'": PRINT
     "AND 'NODRAW'."
4150  PRINT : PRINT "PRESSING 'C'
      CHANGES COLORS."
4160  PRINT : PRINT "PRESSING 'O'
      DRAWS A COMPLETE CIRCLE."
4170  PRINT : PRINT "PADDLE 0 CHA
     NGES THE COMPASS RADIUS."
4180  PRINT : PRINT "PADDLE 1 CHA
     NGES THE COMPASS ROTATION."
4190  VTAB 22: PRINT " (COMMANDS
     ARE SHOWN DURING DRAWING)"
4200  GOSUB 7000
```

Print third page of instructions.

```
4210  VTAB 2: HTAB 10: NORMAL : PRINT
     "IN STRAIGHTEDGE MODE:": INVERSE

4220  VTAB 5: PRINT "PRESSING 'D'
      DRAWS A LINE BETWEEN THE": PRINT
     "TWO ENDPOINTS."
4230  PRINT : PRINT "PRESSING 'P'
      FIXES THE MOVEABLE": PRINT
     "ENDPOINT AND ALLOWS MOVING
     THE OTHER."
4240  PRINT : PRINT "THE PADDLES
     MOVE THE MOBILE ENDPOINT."
4250  PRINT : PRINT "PRESSING 'C'
      CHANGES THE COLOR."
4260  PRINT : PRINT "THE DISTANCE
      BETWEEN ENDPOINTS, AND": PRINT
     "THE ANGLE OF THE DEFINED LI
     NE, ARE": PRINT "DISPLAYED O
     N THE SCREEN."
4270  GOSUB 7000
```

Print fourth page of instructions.

```
4280  VTAB 2: HTAB 11: NORMAL : PRINT
     "IN FREEHAND MODE:": INVERSE

4290  VTAB 5: PRINT "THE PADDLES
     MOVE THE DRAWING POSITION": PRINT
     "AROUND THE SCREEN."
4300  PRINT : PRINT "PRESSING 'D'
      TOGGLES BETWEEN 'DRAW'": PRINT
     "AND 'NODRAW'."
4310  PRINT : PRINT "PRESSING 'C'
      CHANGES THE COLOR."
4320  PRINT : PRINT "THE X AND Y
     COORDINATES ARE DISPLAYED."
4330  IF  PEEK (K1) < NK THEN 433
     0
```

Restore text window, inquire about a previously-stored drawing.

```
4340  POKE K2,0: POKE 32,0: POKE
     33,40: NORMAL
4350  HOME : VTAB 3: PRINT "DO YO
     U HAVE A DRAWING ON TAPE OR
     DISK   TO LOAD IN AT THIS TI
     ME? (Y/N) ";: GET X$
4360  IF X$ <  > "Y" AND X$ <  >
     "N" THEN 4350
4370  IF X$ = "Y" THEN 5000
4380  RETURN
```

Routine to retrieve drawing from tape or disk.

```
5000  PRINT : PRINT : PRINT "FROM
      TAPE OR DISK? (T/D) ";: GET
     X$: IF X$ <  > "T" AND X$ <
      > "D" THEN 5000
5010  HOME : VTAB 3: IF X$ = "T" THEN
     5050
5020  PRINT "INSERT DISK AND ENTE
     R FILE NAME:": PRINT : INPUT
     "  > ";F$
5030  PRINT  CHR$ (4);"BLOAD";F$
5040  PRINT : PRINT "PRESS ANY KE
     Y TO CONTINUE. . .";: GET X$
     : GOTO 4380
5050  PRINT "TO LOAD FROM TAPE:"
5060  PRINT : PRINT "  1. INSERT
     AND POSITION THE TAPE."

5070  PRINT "  2. COPY OVER FIRST
      LINE BELOW USING          R
     IGHT-ARROW KEY, PRESSING THE
               'RETURN' KEY AT T
     HE END."
5080  PRINT "  3. COPY SECOND LIN
     E LIKEWISE, BUT            S
     TART PLAYING BEFORE 'RETURN'
     ."
5090  PRINT "  4. AFTER BEEP, STO
     P RECORDER, AND            C
     OPY THIRD LINE."
5100  PRINT "  5. COPY FOURTH LIN
     E, AND YOU'RE IN           T
     HE DRAFTING BUSINESS!"
5110  VTAB 16: PRINT " CALL -151"
```

continued on next page

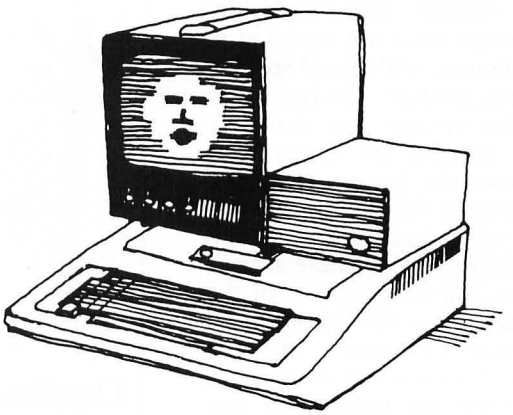
continued from previous page

```
5120  PRINT : PRINT " 2000.3FFFR"

5130  PRINT : PRINT " 06"
5140  PRINT : PRINT " GOTO 50"
5150  VTAB 15: END

Routine to save drawing on tape or
disk.

5500  HOME : VTAB 21: PRINT "SAVE
      ON TAPE OR DISK? (T/D) ";: GET
      X$: IF X$ < > "T" AND X$ <
      > "D" THEN 5500
5510  IF X$ = "T" THEN 5560
5520  HOME : VTAB 21: PRINT "INSE
      RT DISK AND TYPE IN FILE NAM
      E:": PRINT : INPUT "  > ";F$

5530  PRINT  CHR$ (4);"BSAVE";F$;
      ",A$2000,L$1FFF"
5540  PRINT : PRINT "PRESS ANY KE
      Y TO CONTINUE. ";: GET X$
5550  HOME : ON MODE GOTO 1000,20
      00,3000

5560  HOME : VTAB 21: PRINT "DON'
      T BE ALARMED WHEN YOUR DRAWI
      NG      DISAPPEARS--I WON'T
      ERASE IT!"
5570  PRINT : PRINT "PRESS ANY KE
      Y TO CONTINUE. . .";: GET X$
5580  TEXT : HOME : VTAB 2: PRINT
      "TO SAVE ONTO TAPE:"
5590  PRINT : PRINT "  1. INSERT
      TAPE INTO RECORDER."
5600  PRINT "  2. COPY OVER FIRST
       LINE BELOW USING          R
      IGHT-ARROW KEY, PRESSING THE
                'RETURN' KEY AT T
      HE END."
5610  PRINT "  3. COPY THE SECOND
       LINE LIKEWISE, BUT        S
      TART RECORDING BEFORE 'RETUR
      N'."
5620  PRINT "  4. AFTER BEEP, TUR
      N OFF RECORDER, AND        C
      OPY THE THIRD LINE."
5630  PRINT "  5. COPY THE FOURTH
       LINE, AND YOU'RE          B
      ACK IN THE DRAFTING BUSINESS
      !"

5640  VTAB 16: PRINT " CALL -151"

5650  PRINT : PRINT " 2000.3FFFW"

5660  PRINT : PRINT " 06"
5670  PRINT : PRINT " GOTO 6000"
5680  VTAB 15: END

Re-entry point after saving a
drawing to tape.

6000  POKE  - 16304,0: HOME : ON
      MODE GOTO 1000,2000,3000: REM
      -->DON'T CHANGE THIS LINE !!


 Subroutine to wait for keypress and
 then paint part of screen white.


7000  IF  PEEK (K1) < NK THEN 700
      0
7010  POKE K2,0: VTAB 1: HTAB 1: INVERSE
      : FOR II = 1 TO 21: PRINT  SPC(
      39);: NEXT II
7020  RETURN
```

THREE CLASSICS FOR ANY APPLE II® OR II plus®

THE VOICE

If you own an Apple and have a child learning to read, you should have this program. Your child selects a sequence of words from the standard vocabulary and Demo Man happily repeats them. Your child instantly hears the sense or nonsense of the sentence. Easily record your own words and make your Apple say anything you like without adding hardware. Allows you to include speech in any Basic program using Print commands. Caution: You will enjoy this program as much as your kids. (48K) $39.95.



BEST OF MUSE™

Five of our most popular games on one disk. Escape and The Maze Game alone are worth the price. Perform catchy tunes with the Music Box. Side Shows' six mini-games each give you hours of fun. Includes Tank War, an exciting two player shoot out. A superb value to round out your recreational software library. Original publication combined value of $64.75 (32K) $39.95

THREE MILE ISLAND (SPECIAL EDITION)

Rewritten in quick response machine language, author Richard Orban has enhanced the simulation which has already been cited as a classic by reviewers. You are put in complete control of a pressurized nuclear reactor. New features, Auto/Demo and Fast Modes, allow you to control the pace of this exciting full color game. Read all you want about the nuke controversy, but try this simulation if you really want to experience managing a nuclear facility. (48K) $39.95

from the leader in quality software

MUSE SOFTWARE™

Apple II is a trademark of Apple Computer Corp.

TSE:HARDSIDE

6 South St., Milford, NH 03055 (603) 673-5144
TOLL FREE OUT-OF-STATE 1-800-258-1790

COMPUTER GRAPHICS

PATTERN GENERATION: Designing Your Own Patterns

by Joan R. Truckenbrod

There is a wide variety of techniques that can be used to alter the repetitive character of regular patterns. In regular pattern design, modular design units are repeated in a regular manner, with equal spacing between each module. The regularity of patterning systems can easily be varied by altering the interval or space between the pattern elements, and by changing the side by side arrangement of the pattern elements to one another. In varying these aspects of a pattern, the visual effects of the pattern change as the background spaces appear differently, and the relationship between the elements is varied as they are tangent to one another in different areas.

Changing the relationship between the elements can be accomplished by shifting every other horizontal row a distance equal to half of the width of an individual design unit. This shift changes the position of each design element in relation to other elements. In other words, these units no longer line up with one another, rather they are stepped so that the top and bottom of each element are not in line with the top or bottom of the adjacent figure. This technique creates a brick pattern. In this type of pattern, the position of elements in the horizontal rows remains the same but the vertical row consists of offset figures. The grid underlying this pattern is shown under Pattern Option 3. Examples of this type of pattern are illustrated with the grid. It is interesting to contrast these new patterns with the regular patterns illustrated in the May issue of **SoftSide** as the patterns appear very different with this slight change in the ordering system. The program statements used to create this brick pattern are statements to be changed in the original program listed in the June issue of **SoftSide**. Experiment with other variations in patterns that can be created by modifying the original grid with horizontal and vertical shifting to change the position of the design elements so that they are no longer side by side.

Another method for creating different patterns is to vary the spacing between the pattern elements. By leaving more space between some elements and less between others, open areas and more dense areas are created within the pattern. This technique also creates the effect of groups of figures within the pattern as illustrated in the Option 4 grid and pattern examples. The portion of the original program that requires changes to create this type of pattern is also listed. Changing the interval between the design elements within the pattern can create a wide range of visually different patterns. Compare the sets of patterns created on the next page with these two alternative approaches and you can see that there is significant potential for creating new patterns with slight variations in the patterning program.

Original Program

```
10  REM  PATTERN GENERATION PROGRAM
15  REM   BY JOAN R. TRUCKENBROD
20  DIM X(25),Y(25)
30  REM  NP = NUMBER OF POINTS IN THE FIGURE
40  NP = 7
45  REM  REM X AND Y COORDINATES DEFINE THE PATTERN ELEMENT
50  FOR I = 1 TO NP
60  READ X(I),Y(I)
70  NEXT I
80  DATA  0,8,12,0,20,8,20,20,8,20,12,8,0,8
85  HGR : HCOLOR= 7
90  REM  THE R LOOP KEEPS COUNT OF THE VERTICAL ROW NUMBER
100  FOR R = 0 TO 8
110  REM  THE B LOOP KEEPS COUNT OF THE HORIZONTAL COLUMN NUMBER
120  FOR B = 0 TO 12
130  REM  MOVE THE PEN TO THE FIRST POINT IN THE FIGURE
140  HPLOT X(1) + B * 20,Y(1) + R * 20
150  REM  THE FOLLOWING LOOP CONNECTS THE REMAINING POINTS IN THE FIGURE.
160  FOR P = 2 TO NP
170  HPLOT  TO X(P) + B * 20,Y(P) + R * 20
180  NEXT P
190  NEXT B
200  NEXT R
210  END
```

Option 3

Change the following lines in the original program

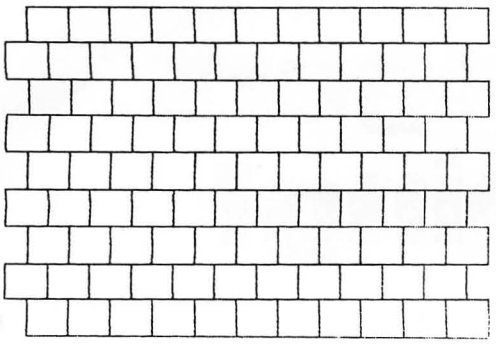
```
12  DIM S4(2)
13 S2(1) = 0:S4(2) = 10

130  REM  MOVE THE PEN TO THE FIRST POINT IN THE FIGURE
140  HPLOT X(1) + B * 20 + S4(L),Y(1) + R * 20
150  REM  THE FOLLOWING LOOP CONNECTS THE REMAINING POINTS IN THE FIGURE.
160  FOR P = 2 TO NP
170  HPLOT  TO X(P) + B * 20 + S4(L),Y(P) + R * 20

195 L = L + 1
196  IF L < 3 GOTO 200
197 L = 1
```

Option 4

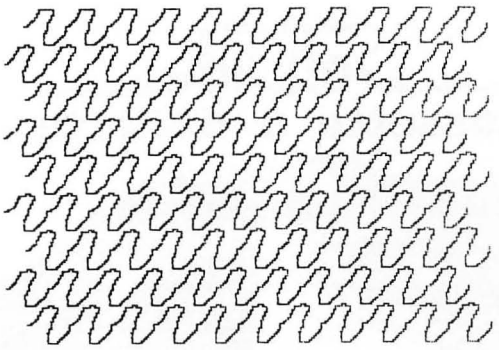
Change the following lines in the original program

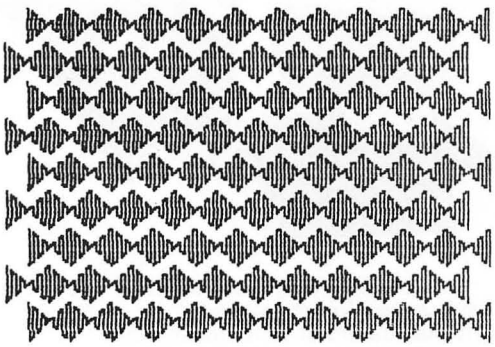
```
11  DIM S2(4)
12  FOR I = 1 TO 4: READ S4(I): NEXT I
13  DATA  20,0,30,10

130  REM  MOVE THE PEN TO THE FIRST POINT IN THE FIGURE
140  HPLOT X(1) + S4(L) + B * 50,Y(1) + R * 10
150  REM  THE FOLLOWING LOOP CONNECTS THE REMAINING POINTS IN THE FIGURE.
160  FOR P = 2 TO NP
170  HPLOT  TO X(P) + S4(L) + B * 50,Y(P) + R * 10

192 L = L + 1
193  IF L < 5 THEN  GOTO 200
194 L = 1
```

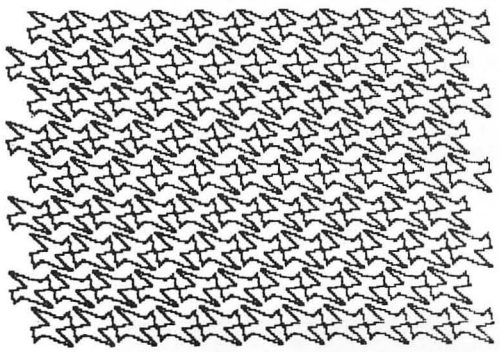
PATTERN OPTION 3

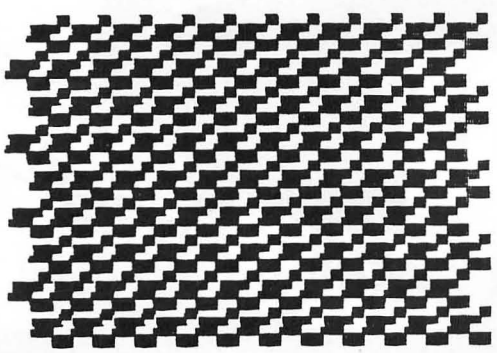


GRID

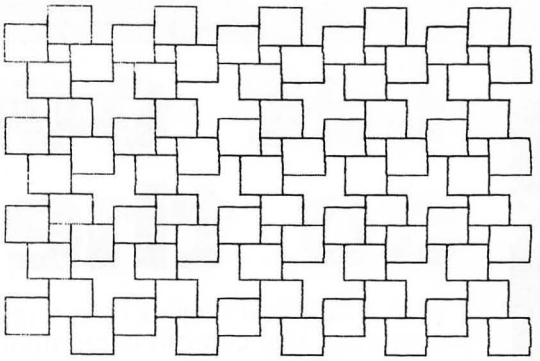




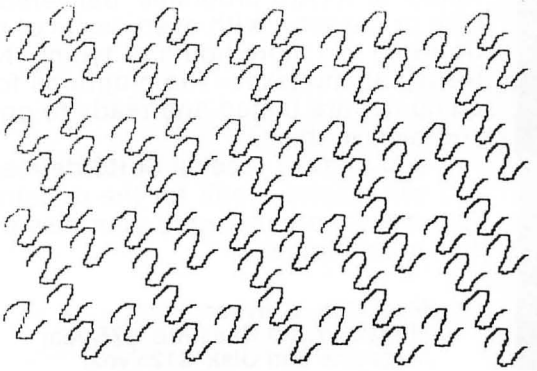


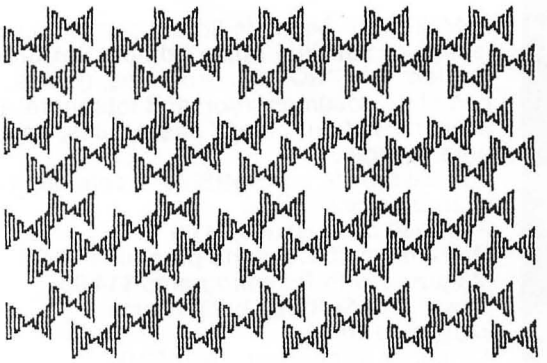


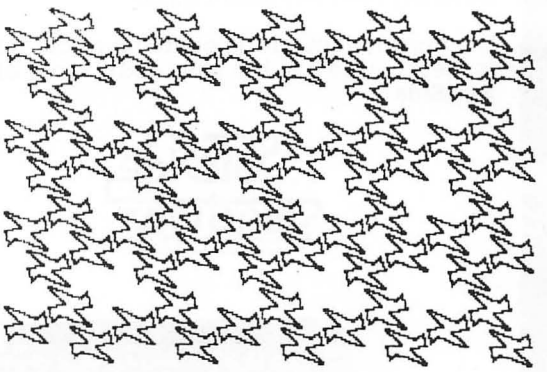
PATTERN OPTION 4

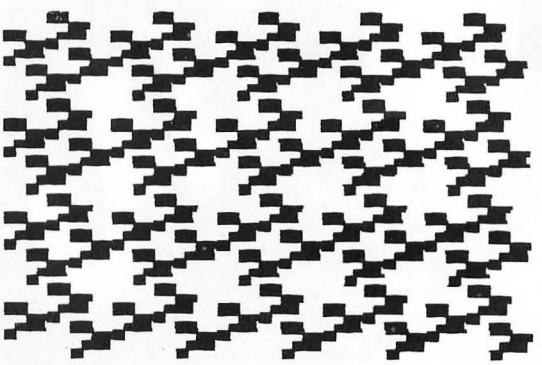


GRID









Let our fingers do the typing!

With a **SoftSide** Disk or Cassette subscription you can get each month's TRS-80, Apple, or ATARI programs delivered on disk or cassette with every issue of your **SoftSide**. No more hours of typing. No hunting for typing mistakes. The programs for your computer are tested and ready to go the day you receive them.

If you already receive **SoftSide** magazine, you will receive credit for the remainder of your subscription toward your new cassette or disk subscription.

Magazine $24/year
Magazine and Cassette $75/year
Magazine and Disk $125/year

Make your **SoftSide** library complete with back issues of **SoftSide Apple Edition, SoftSide: S-80 Edition,** and **PROG/80.** Programs, games, complete documentation and lots more! If you have missed out on any past issues, now is the time to order.
SoftSide: S-80, and **SoftSide: Apple** back issues (magazine only) $2.50
PROG/80 back issues $4.00
Magazine with Cassette programs $9.95
Magazine with Disk programs $14.95
New Super **SoftSide** back issues (Magazine Only)
August or September, 1980 $3.00
October, 1980 to present $3.50

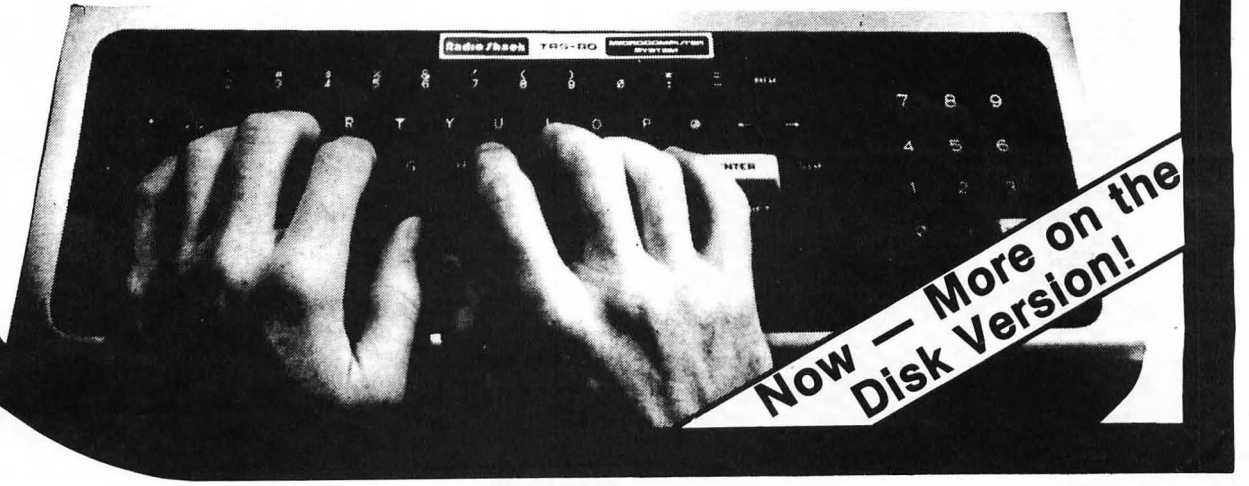
For descriptions of the contents of our back issues, please see the May, 1981, issue of **SoftSide.**

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

SoftSide
cassette & diskette

Collectors! Protect your **SoftSide** back issues, Volumes I and II, or any publication of your choice, with these durable wood grain vinyl binders with inside pocket and clear spine sleeve for easy identification. Holds and protects 12 back issues. A regular $4.95 value, SALE priced at $3.95. Free (while the supply lasts) with the purchase of volume I or II (12 issue collection of **SoftSide**).

Small	$3.95
8½ X 11	$7.95



Spacelander

by Jack and Bob Wiener

"Spacelander" is a 16K Atari program requiring one joystick.

You gain control of your space ship by placing the joystick into the slot farthest to the left.

At the start of the game, your ship is placed in the upper-middle part of the screen where there is no movement. After a second or so, your ship will start moving downward, toward the jagged terrain of the planet. This downward motion occurs because of the gravity of the planet and if no upward motion is applied, the ship will start going down faster and faster. You can counteract this gravity to slow down or raise the ship by pushing the joystick forward. To steer your ship to the left, right, upper left, or upper right, just push the joystick in the direction you want to go. However, everytime you use your engines, you burn up fuel. One unit of fuel is lost by left and right movements, two units of fuel are lost by upward motion, and 300 units are lost by crashing.

In the bottom half of the screen there are five different landing pads, four with numbers and one with a letter. The numbers stand for points, while the letter stands for fuel.

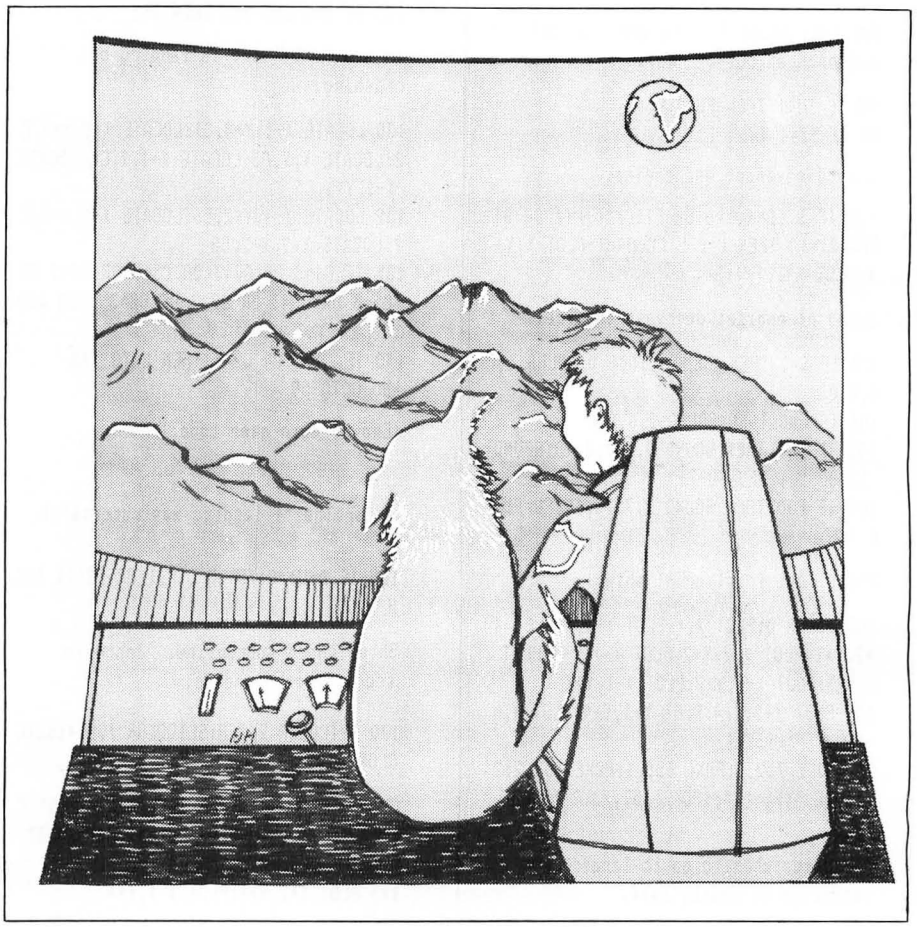
Guide your ship toward the landing pad you want to land on. When you get close to the landing pad, a blown-up version of that landing pad will appear on the screen. You now want to guide your ship in for a soft landing on the pad. Both legs of your ship have to be on the pad, and it can't be a hard landing. When you finally land, you will be credited with either the points or more fuel, and you will then continue your game from the starting position.

To determine how many points a landing pad is worth, you just multiply the number below the pad by 100.

When you land at the fuel landing pad your ship will be refueled to 1500 which is the maximum amount of fuel. Good luck and happy landing.

VARIABLES

FU: Fuel.
X, Y: Coordinates of ship.
H: Horizontal movement of ship.
U: Power up.
D: Power down (gravity).
F5: Out of fuel indicator; shuts off engines.
V: Velocity.
S: Sound up.
F: Sound left and right.
SC: Score.
B: Base you are near.
C 1-8: Determines if crash or land.
E: Error indicator.
L, L1: Timer loop.
N: Note.
A: Volume.



```
1 REM SPACELANDER BY JACK AND BOB WIEN
ER

Set fuel to 1500; go to title and
song.

9 FU=1500:GOSUB 290

Set starting position of lander; stop
all motion; go to landscape.

10 GRAPHICS 7:X=80:Y=10:H=0:U=0:D=0:GO
SUB 1000
```

```
Main game loop:

On an error, go to subroutine that
changes landscape.

59 TRAP 900

Draw ship and determine if the
joystick is operational.

60 COLOR 2:PLOT X,Y:PLOT X+1,Y:PLOT X+
2,Y+1:PLOT X+2,Y+2:PLOT X+1,Y+3:PLOT X
,Y+3:PLOT X-1,Y+2:PLOT X-1,Y+1
```

```
61 PLOT X+2,Y+4:PLOT X-1,Y+4:F=0:IF F5
=1 THEN GOTO 81

Left and right movement of ship.

70 IF STICK(0)<8 THEN H=H+0.05:F=1
80 IF STICK(0)>8 AND STICK(0)<12 THEN
H=H-0.05:F=2

Left and right engines.

81 IF F=1 THEN COLOR 1:PLOT X-1,Y+1:PL
OT X-1,Y+2
```

continued on next page

continued from previous page

```
82 IF F=2 THEN COLOR 1:PLOT X+2,Y+1:PL
OT X+2,Y+2
```

Set maximums for left and right speed.

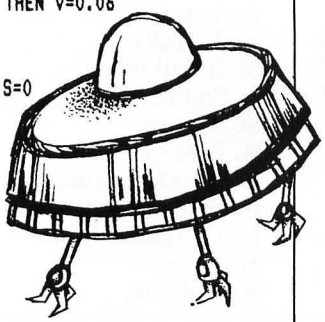
```
83 IF H>2 THEN H=2
85 IF H<-2 THEN H=-2
```

Determine speed of upward motion.

```
90 IF U+D<0 THEN V=0.03
91 IF U+D>0 THEN V=0.06
```

Gravity.

```
92 D=D+0.02:S=0
```



Set fuel gauge to zero and shut off engine when out of fuel.

```
95 IF FU<1 THEN FU=0:F5=1
99 IF F5=1 THEN 103
```

Upward movement and engine.

```
100 IF STICK(0)=6 OR STICK(0)=14 OR ST
ICK(0)=10 THEN U=U-V:COLOR 1:PLOT X,Y+
4:PLOT X+1,Y+4:S=1
```

Sound of engine; decrease the fuel.

```
103 IF S=0 THEN SOUND 0,0,0,0:SOUND 1,
0,0,0
104 IF F=0 THEN SOUND 2,0,0,0
105 IF S=1 THEN SOUND 0,150,8,10:SOUND
 1,180,2,10:FU=FU-2
107 IF F>0 THEN SOUND 2,75,8,10:FU=FU-
1
```

Erase last location of ship.

```
108 COLOR 0:PLOT X,Y:PLOT X+1,Y:PLOT X
+2,Y+1:PLOT X+2,Y+2:PLOT X+1,Y+3:PLOT
X,Y+3:PLOT X-1,Y+2:PLOT X-1,Y+1
109 PLOT X+2,Y+4:PLOT X-1,Y+4:PLOT X,Y
+4:PLOT X+1,Y+4
110 POKE 752,1:POKE 656,1:POKE 657,2:?
 "FUEL ";FU;" ";:POKE 657,22:? "SCORE
";SC
```

Determine when to go to larger landscape of nearby base.

```
112 IF B>0 THEN 117
113 IF X>108 AND X<124 AND Y>29 AND Y<
50 THEN GOSUB 3000
114 IF X<34 AND X>6 AND Y>37 THEN GOSU
B 4000
115 IF X>36 AND X<66 AND Y>34 THEN GOS
UB 4050
116 IF X>136 AND X<151 AND Y>48 THEN G
OSUB 4100
117 IF B=1 AND X>5 AND X<79 AND Y>30 T
HEN GOSUB 4500
```

Fuel warning.

```
119 IF FU<300 THEN POKE 752,1:POKE 656
,1:POKE 657,2:? "FUEL";
```

Movement of ship.

```
120 X=X+H:Y=Y+U+D
```

Wrap-around movement of ship.

```
121 IF B=0 AND X<3 THEN X=156
122 IF B=0 AND X>156 THEN X=4
123 IF B=0 AND Y<3 THEN Y=5
```

Determine if ship is landing or crashing.

```
130 LOCATE X-1,Y+4,C1:LOCATE X+2,Y+4,C
2:LOCATE X,Y,C3:LOCATE X+1,Y,C4:LOCATE
 X-1,Y+1,C5
132 LOCATE X-1,Y+2,C6:LOCATE X+2,Y+1,C
7:LOCATE X+2,Y+2,C8
135 IF C1=3 OR C2=3 OR C3=3 OR C4=3 OR
 C5=3 OR C6=3 OR C7=3 OR C8=3 THEN 600
0
140 IF C1=1 OR C2=1 THEN GOTO 165
150 GOTO 59
```

(End of main game loop.)

Determine if landing was successful.

```
165 IF D+U>=0.75 OR C1<>1 OR C2<>1 THE
N 6000
```

Ship-landed subroutine. Increase score, cheer landing.

```
170 SETCOLOR 2,0,8:SETCOLOR 1,8,4:SETC
OLOR 0,4,6:Y=Y-1
172 COLOR 2:PLOT X,Y:PLOT X+1,Y:PLOT X
+2,Y+1:PLOT X+2,Y+2:PLOT X+1,Y+3:PLOT
X,Y+3:PLOT X-1,Y+2:PLOT X-1,Y+1
173 PLOT X+2,Y+4:PLOT X-1,Y+4
174 IF B=1 THEN SC=SC+100
175 IF B=2 THEN SC=SC+200
176 IF B=3 THEN SC=SC+300
177 IF B=9 THEN SC=SC+900
178 IF B=5 THEN FU=1500:F5=0
200 COLOR 2:PLOT X,Y-1:PLOT X,Y-2:PLOT
 X,Y-4:COLOR 1:PLOT X,Y-3:DRAWTO X+2,Y
-3
205 COLOR 3:PLOT X+1,Y-4:PLOT X+2,Y-4
210 POKE 752,1:POKE 656,1:POKE 657,2:?
 "FUEL ";FU;" ";:POKE 657,22:? "SCORE
";SC
260 FOR L=0 TO 3:SOUND L,0,0,0:NEXT L:
RESTORE :FOR L=1 TO 10:READ N:SOUND 0,
N,10,14
270 FOR L1=1 TO 40:NEXT L1:NEXT L
284 SOUND 0,0,0,0
285 FOR L=1 TO 700:NEXT L:GOTO 10
```

Opening print statement.

```
290 GRAPHICS 18:POSITION 5,4:? #6;"SPA
CE LANDER";
```

Beginning and ending song.

```
295 RESTORE :FOR L=1 TO 10:READ N:NEXT
 L
300 FOR L=1 TO 10:READ N,N1:SOUND 0,N,
10,14:FOR L1=1 TO 60*N1:NEXT L1:NEXT L
:RETURN
```

Coloring subroutine.

```
400 SETCOLOR 1,8,4:SETCOLOR 0,4,8:SETC
OLOR 2,12,8:RETURN
```

Determine where ship appears, if it leaves the large landing base.

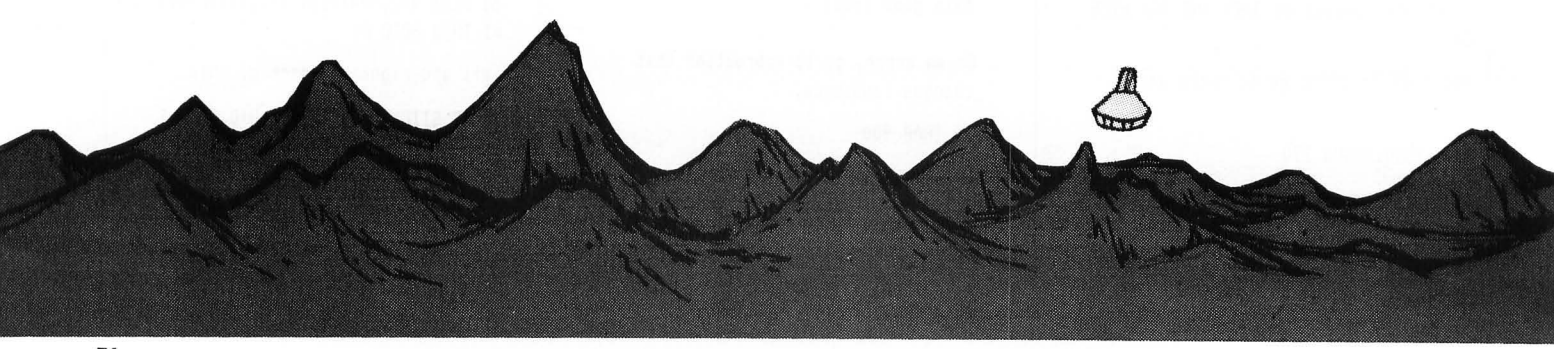
```
900 GRAPHICS 7
905 IF B=1 THEN Y=30:IF X<37 THEN X=37
910 IF B=2 THEN X=112:Y=25
920 IF B=3 THEN GOSUB 4050:GOTO 59
930 IF B=5 THEN X=20:Y=28
940 IF B=9 THEN X=142:Y=48
```

Execute coloring subroutine.

```
1000 GOSUB 400
```

Draw main landscape.

```
1010 B=0:COLOR 1:PLOT 13,69:DRAWTO 19,
69:PLOT 46,56:DRAWTO 52,56:PLOT 86,74:
DRAWTO 94,74
1020 PLOT 109,50:DRAWTO 115,50:PLOT 14
4,73:DRAWTO 150,73
```



```
1060 COLOR 3:PLOT 0,79:DRAWTO 6,40:DRA
WTO 15,49:DRAWTO 10,52:DRAWTO 7,60:DRA
WTO 12,69
1070 PLOT 16,50:DRAWTO 19,51:PLOT 20,6
9:DRAWTO 24,62:DRAWTO 20,57:DRAWTO 25,
63
1080 PLOT 25,63:DRAWTO 35,40:DRAWTO 45
,56:PLOT 159,79:DRAWTO 150,62:DRAWTO 1
51,73
1090 PLOT 150,62:DRAWTO 146,58:DRAWTO
140,64:DRAWTO 130,71:PLOT 143,73:DRAWT
O 142,79
2000 PLOT 116,50:DRAWTO 124,40:DRAWTO
123,65:DRAWTO 113,70:DRAWTO 115,79
2010 PLOT 108,50:DRAWTO 100,38:PLOT 11
2,70:DRAWTO 94,60:DRAWTO 80,65:PLOT 95
,75:DRAWTO 115,79
2020 PLOT 85,75:DRAWTO 84,79:PLOT 53,5
6:DRAWTO 60,79:PLOT 79,65:DRAWTO 63,56
:DRAWTO 94,60
2030 PLOT 85,75:DRAWTO 60,79:PLOT 108,
50:DRAWTO 123,65:PLOT 99,38:DRAWTO 63,
56
2037 PLOT 95,74:DRAWTO 113,71:PLOT 123
,65:DRAWTO 138,50:DRAWTO 123,53
2040 COLOR 2:PLOT 15,71:DRAWTO 13,71:D
RAWTO 13,73:DRAWTO 15,73:PLOT 13,73:DR
AWTO 13,75:PLOT 48,58:DRAWTO 48,63
2050 PLOT 150,79:DRAWTO 150,75:DRAWTO
147,75:DRAWTO 147,77:DRAWTO 150,77
2060 PLOT 116,52:PLOT 117,51:PLOT 118,
52:PLOT 118,53:DRAWTO 116,55:DRAWTO 11
8,55
2065 PLOT 90,75:DRAWTO 93,75:DRAWTO 93
,79:DRAWTO 90,79:PLOT 90,77:DRAWTO 93,
77:GOTO 59
```

Landscape for 200-point base.

```
3000 GRAPHICS 5:X=35:Y=1:GOSUB 400
3010 COLOR 1:PLOT 37,35:DRAWTO 42,35:C
OLOR 3:PLOT 43,34:DRAWTO 46,28:DRAWTO
55,24:DRAWTO 79,9
3020 PLOT 36,35:PLOT 36,34:PLOT 35,33:
PLOT 34,32:PLOT 35,31:PLOT 35,30:DRAWT
O 33,24
3030 DRAWTO 25,20:DRAWTO 23,15:DRAWTO
15,23:DRAWTO 0,9
3040 B=2:RETURN
```

Landscape for fuel base.

```
4000 GRAPHICS 5:X=45:Y=0:GOSUB 400
4010 COLOR 1:PLOT 37,39:DRAWTO 42,39:C
OLOR 3:PLOT 36,38:DRAWTO 20,31:DRAWTO
15,24:DRAWTO 21,17:DRAWTO 38,12
4015 DRAWTO 47,16:DRAWTO 10,0:DRAWTO 0
,15
4020 PLOT 43,38:DRAWTO 56,31:DRAWTO 79
,4:PLOT 56,31:DRAWTO 32,22
4025 PLOT 38,12:DRAWTO 47,16
4030 B=5:RETURN
```

Landscape for 100-point base.

```
4050 GRAPHICS 5:Y=0:GOSUB 400
4060 COLOR 1:PLOT 38,33:DRAWTO 34,33:C
OLOR 3:DRAWTO 20,10:DRAWTO 0,25
4070 PLOT 39,34:DRAWTO 42,39:PLOT 50,3
3:DRAWTO 79,39:PLOT 50,33:DRAWTO 79,22
4075 IF B=3 THEN X=43:Y=30
4080 B=1:RETURN
```

900-point base.

```
4100 GRAPHICS 5:X=40:Y=3:GOSUB 400
4300 COLOR 1:PLOT 38,28:DRAWTO 43,28:C
OLOR 3:PLOT 37,27:DRAWTO 29,39:DRAWTO
5,39:DRAWTO 0,33:PLOT 17,31:DRAWTO 23,
31
4310 DRAWTO 43,10:DRAWTO 63,39:PLOT 0,
33:DRAWTO 12,25:DRAWTO 31,12:DRAWTO 30
,8:PLOT 44,27:DRAWTO 52,25
4320 PLOT 41,21:DRAWTO 50,21:PLOT 41,2
1:DRAWTO 47,17
4325 PLOT 63,39:DRAWTO 79,15:PLOT 30,8
:DRAWTO 10,12:DRAWTO 10,0
4330 B=9:RETURN
```

300-point base.

```
4500 GRAPHICS 5:X=2:Y=0:GOSUB 400
4510 COLOR 1:PLOT 50,34:DRAWTO 55,34:C
OLOR 3:DRAWTO 79,39:PLOT 49,34:DRAWTO
48,39:PLOT 49,34:DRAWTO 17,39:DRAWTO 2
,8
4520 PLOT 12,0:DRAWTO 44,26:DRAWTO 48,
26:DRAWTO 65,12:PLOT 56,34:DRAWTO 78,3
0:DRAWTO 65,12
4521 PLOT 12,0:DRAWTO 65,12:PLOT 79,39
:DRAWTO 78,30
4530 B=3:RETURN
```

Error for crashing, used if point of crash goes off screen.

```
5999 E=1:GOTO 6009
```

Crash routine: Blow up ship, rotate screen colors, sound explosion, reduce fuel. If no fuel left, go to end routine.

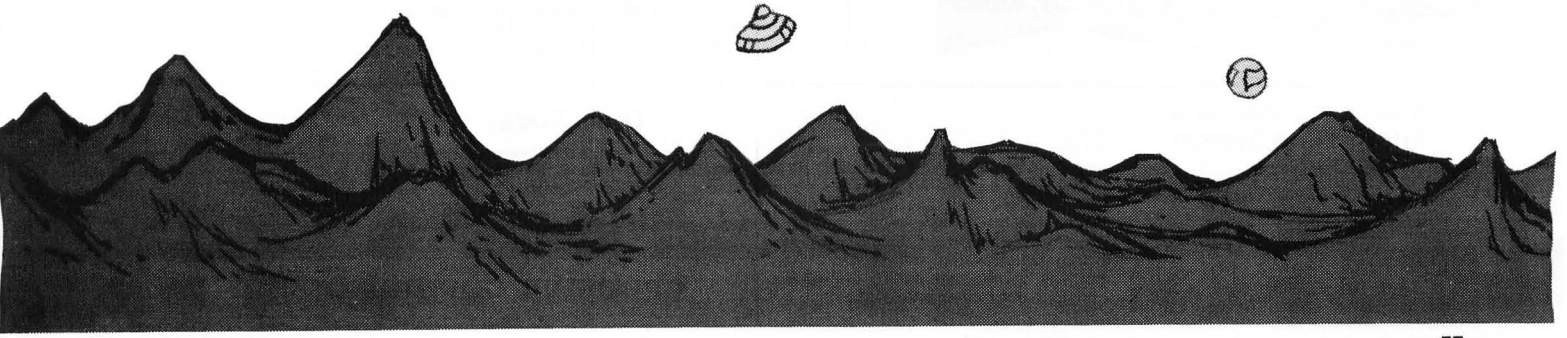
```
6000 COLOR 0:PLOT X,Y:PLOT X+1,Y:PLOT
X+2,Y+1:PLOT X+2,Y+2:PLOT X+1,Y+3:PLOT
 X,Y+3:PLOT X-1,Y+2:PLOT X-1,Y+1
6001 PLOT X+2,Y+4:PLOT X-1,Y+4:PLOT X,
Y+4:PLOT X+1,Y+4
6002 FOR L=0 TO 3:SOUND L,0,0,0:NEXT L
:E=0
6007 A=11:SETCOLOR 4,3,8:FOR L=1 TO 10
6008 TRAP 5999
6009 IF E=1 THEN FOR L1=1 TO 10:NEXT L
1:GOTO 6035

6010 COLOR 2:PLOT X,Y-L:PLOT X+1,Y-L:P
LOT X+2+L,Y+1:PLOT X+2+L,Y+2:PLOT X+1,
Y+3+L:PLOT X,Y+3+L
6011 PLOT X-1-L,Y+1:PLOT X-1-L,Y+2:PLO
T X+2,Y+4+L:PLOT X-1,Y+4+L
6015 FOR L1=1 TO 25:NEXT L1
6020 COLOR 0:PLOT X,Y-L:PLOT X+1,Y-L:P
LOT X+2+L,Y+1:PLOT X+2+L,Y+2:PLOT X+1,
Y+3+L:PLOT X,Y+3+L
6021 PLOT X-1-L,Y+1:PLOT X-1-L,Y+2:PLO
T X+2,Y+4+L:PLOT X-1,Y+4+L
6035 SOUND 0,(RND(0)*30)+40,0,A:SOUND
1,(RND(0)*30)+40,2,A:SOUND 2,(RND(0)*3
0)+40,6,A
6053 A=A-1:SETCOLOR 4,INT(RND(0)*15),8
:NEXT L
6054 SETCOLOR 4,0,0:FU=FU-300
6065 FOR L=0 TO 3:SOUND L,0,0,0:NEXT L
:FOR L=1 TO 200:NEXT L
6070 IF FU>0 THEN 10
```

End of game.

```
7000 GRAPHICS 18:POSITION 5,0:? #6;"YO
UR SCORE":POSITION 9,5:? #6;"IS":POSIT
ION 8,10:? #6;SC:GOSUB 295
```

Data for opening and closing song.

```
7100 DATA 121,108,96,81,81,96,81,81,81
,81,243,4,162,4,121,6,96,2,102,4,243,4
,162,4,121,6,81,2,60,8
```

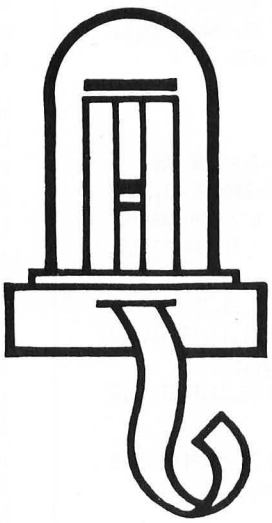


TRS-80 sensational software

creative computing software

Stock & Options Analysis

Cassette CS-3306 (16K), $99.95
Disk CS-3801 (32K), $99.95



Should you hedge, buy, or sell out? **Stock and Options Analysis** puts a securities advisor in your computer, providing you with four powerful investment tools. **Option** gives important indices for opening and closing call option transactions. **Opgraph** presents a graph or table of profit for any combination of long or short calls, puts, and stocks. This allows the detailed evaluation of three types of hedges. **Newprem** helps predict the future premiums of an option at any desired time and future stock price. **Portval** lets the computer do the paper work, providing full portfolio services, including value per share, current value, and capital gain. The program includes the effects of commissions, margin interest and dividends. Beyond helping to organize and evaluate your present portfolio, **Stock and Options Analysis** is an excellent aid for planning and testing future strategies. The comprehensive 24-page manual with this package not only shows how the programs work, but is also a primer on the strategy of hedging listed options against common stocks. This strategy has been repeatedly shown to actually be more conservative and more consistently profitable than straight buying and selling of stocks.

Solar Energy Analysis

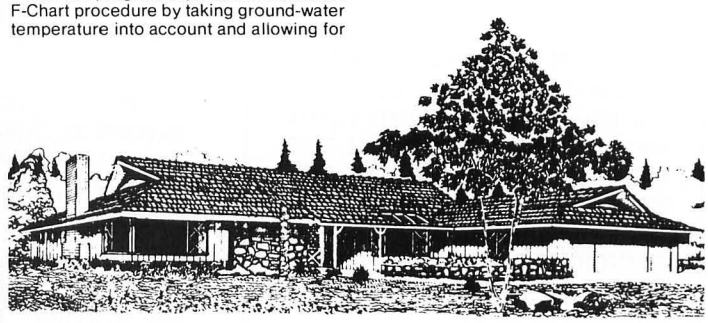
Cassette CS-3307 (16K), $49.95
Disk CS-3802 (32K) $99.95

Available 7/81

F-Chart Solar Energy Analysis eliminates many of the tedious calculations required when designing solar-heating systems. Beyond providing a thermal analysis, the program allows designers to quickly determine the effects of changing any specifications, allowing fast, accurate, and inexpensive experimentation.

Systems using air, liquid, or domestic hot water in any climate can be analyzed in detail. The program expands the traditional F-Chart procedure by taking ground-water temperature into account and allowing for mixing valves in domestic hot water systems. **F-Chart Solar Energy Analysis** quickly pays for itself by freeing you from time-consuming calculations.

The disk version of the program includes a data base of all necessary climatic data for any location in the United States. These data are in the printed booklet included with the cassette version but must be entered manually for your geographic location.

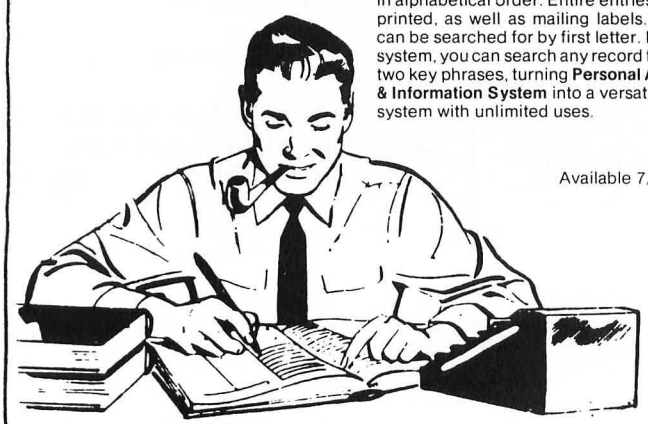


Personal Address and Information System

Disk CS-3509 (32K) $24.95

Is your address book beginning to resemble a heavily-edited inkblot? Do your friends keep moving, forcing you to cross cut and rewrite addresses and phone numbers? Let the **Personal Address & Information System** turn filing drudgery into computing pleasure. You can store all the crucial information, including name, address, home and work phone numbers, spouse's name, and comments or remarks. At any time, the information can be edited or changed.

And there's more. Names can be sorted in alphabetical order. Entire entries can be printed, as well as mailing labels. Names can be searched for by first letter. In a 32K system, you can search any record for up to two key phrases, turning **Personal Address & Information System** into a versatile filing system with unlimited uses.



Available 7/81

Text Processing

Cassette CS-3302 (16K) $14.95

CS-3504 Disk (32K) $24.95
(Disk includes Checking Account, CS- 3304)

This program turns a 16K, TRS-80 and lineprinter into a line oriented text-processing system.

```
>1 THIS IS THE MARVELOUS CREATIVE COMPUTING TEXT PROCESSOR
>2 IT CAN DO MANY WONDERFUL THINGS, BUT IT CANNOT RUN YOUR
>3 ELECTRIC BLANKET.  IT IS AMAZING HOW MANY PEOPLE WANT
>4 A TEXT PROCESSOR TO TURN ON THE COFFE POT AND RUN THE
>5 ELECTRIC BLANKET.  BUT I AM NOT A CRAY ONE AND CAN ONLY
>6 DO MUNDANE THINGS LIKE CHEW ON CARPETING AND EAT SMALL
>7 BOYS.
```

```
                    COMMANDS

C       CONTINUE LIST ON SCREEN
D       DELETE LINE
E       EDIT
I       INSERT LINE
K       RESUME KEYING
L       LIST ON SCREEN
P       PRINT HARD COPY
Q       QUIT PROGRAM
T       SAVE ON TAPE

COMMAND?
```

Developed exclusively for the TRS-80 this program lets you use the computer to enter general text or business letters, edit and modify your work, save text on cassette tapes, and print out a perfect report, document, or letter every time.

Editing commands are similar to those used in Level II BASIC, so there are no complicated new commands to learn. Lines may be either inserted or deleted. A special format is available to speed entry of business letters. Final printout can be done in numbered pages and you may print multiple copies.

Business Address & Information System

Disk CS-3510 (48K) $24.95 Available 7/81

Do you need quick access to business contacts and customers? Put more organization in your organization with the **Business Address & Information System**. A complete file containing company name, address, phone number, and comments can be quickly entered and stored. Information can be changed or edited whenever necessary. The program allows entire entries to be printed, and can also generate mailing labels.

When you need information fast, you can search for specific names or find all entries that contain one or two key phrases. Any key phrases can be used. **Business Address & Information System** will help you make the most of your time, putting the routine work in the computer where it belongs.

Order Today

To order any of these software packages, send payment plus $2.00 postage and handling per order to Creative Computing. Morris Plains, NJ 07950. Visa, MasterCard and American Express orders may be called in toll-free.

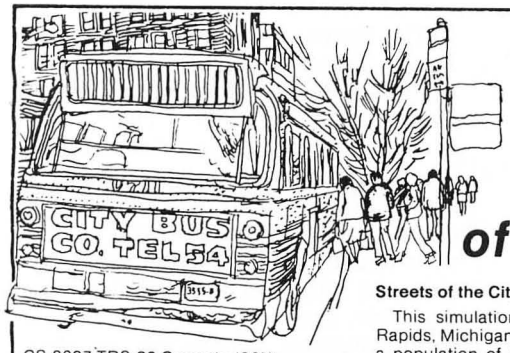
Order today at no risk. If you are not completely satisfied, your money will be promptly and courteously refunded.

Creative Computing Software
Morris Plains, NJ 07950
Toll-free **800-631-8112**
In NJ, 201-540-0445

creative computing software

TRS-80 sensational software

creative computing software



Trucker and Streets of the City

CS-3207 TRS-80 Cassette (32K)
CS-3703 TRS-80 Disk (32K) $24.95

Trucker

This program simulates coast-to-coast trips by an independent trucker hauling various cargos. The user may haul oranges, freight or U.S. mail. All have different risks and rewards. Maximum profit comes from prudent risk-taking.

If all goes well, you can obey the speed limits, stop for eight hours of sleep each night and still meet the schedule. Bad weather, road construction or flat tires may put you behind schedule. You may try to increase your profit by skimping on sleep, driving fast or carrying an overweight load.

Other factors are choice of routes, truck payments, fuel, food, tolls and fines. The simulation is engrossing and informative.

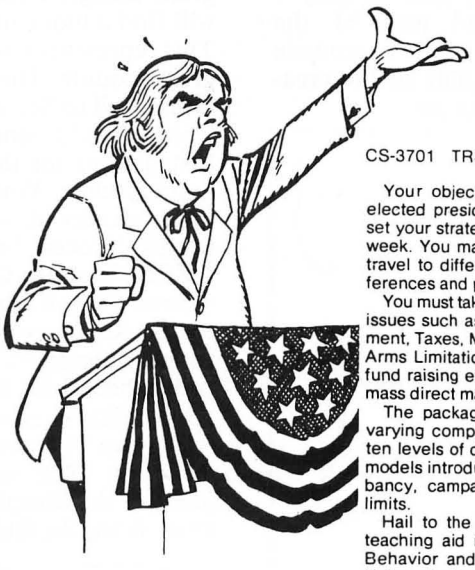
Streets of the City

This simulation is modeled on Grand Rapids, Michigan, a metropolitan area with a population of 550,000. The budgeting, cost and work standard bases are derived from actual experiences of the city over the past five years. The objective of the simulation is to complete a ten-year plan of street and transit improvements while retaining the support of a majority of the City Commission.

During your tenure, you must construct streets and Interstate highways, repair existing streets, and improve traffic safety. For the Transit Authority you have to upgrade and replace a delapidated bus fleet, increase ridership, reduce maintenance downtime and improve on-shedule performance.

Other factors to be considered are operating tax levies, construction bonding and labor negotiations. The simulation provides a substantial challenge and it is both educational and entertaining.

Hail to the Chief



by
Phillip W. Brashear
and
Richard G. Vance

CS-3701 TRS-80 Disk, 48K **$24.95**

Your object in this simulation is to be elected president. In your campaign you set your strategy and carry it out week by week. You may run TV or magazine ads, travel to different states, hold news conferences and participate in a debate.

You must take a position on ten campaign issues such as Energy Policy, Unemployment, Taxes, Mid-East Policy and Strategic Arms Limitations. You must manage your fund raising efforts to business, labor and mass direct mail solicitations.

The package includes four models of varying complexity; each can be used at ten levels of difficulty. The more complex models introduce the influences of incumbancy, campaign finance and spending limits.

Hail to the Chief has been used as a teaching aid in Political Science, Voting Behavior and Computer Science at the University level since 1976. It is a well proven package which includes a comprehensive manual.

3 Adventures

Disk CS-3516 $39.95
Requires 32K

Adventureland (by Scott Adams) You'll encounter wild animals, dwarfs and many other puzzles and perils as you wander through an enchanted world, trying to rescue the 13 lost treasures. Can you rescue the Blue Ox from the quicksand? Or find your way out of the maze of pits? Happy Adventuring!

Pirate Adventure (by Scott Adams)— Yo Ho Ho and a bottle of rum. . . . You'll meet up with the pirate and his daffy bird along with many strange sights as you attempt to go from your London flat to Treasure Island. Can you recover Long John Silver's lost treasures? Happy sailing matey.

Mission Impossible Adventure (by Scott Adams)— Good Morning. Your mission is to . . . and so it starts. Will you be able to complete your mission in time? Or is the world's first automated nuclear reactor doomed? This one's well named. Its hard, there is no magic but plenty of **suspense**.
Good Luck.

Voodoo Castle
The Count and Ghost Town

Disk CS-3517 $39.95
Requires 32K

Voodoo Castle (by Scott Adams). Count Cristo has had a fiendish curse put on him by his enemies. There he lies, you are his only hope . . . will you be able to rescue him—or is he forever doomed? Beware the Voodoo man.

The Count (by Scott Adams). You wake up in a large brass bed somewhere in Transylvania. Who are you, what are you doing here, and why did the postman deliver a bottle of blood? You'll love this Adventure. In fact, you might say it's Love at First Byte. . .

Ghost Town (by Scott Adams). Explore a deserted western mining town in search of 13 treasures. From rattlesnakes to runaway horses, this Adventure has them all! Just remember, pardner, they don't call them Ghost Towns for nothin! (Also includes a new bonus scoring system.)

Original Adventure

Disk CS-3518 (48K) $19.95

This is the original adventure game complete with a colossal cave populated with nasty little dwarves, a giant clam, trools and much, much more. Includes the SAM76 language in which the game runs.

Adventures on Cassette

Five adventures are available separately on cassette. Each requires 16K and costs $14.95.

CS-3007 Adventureland
CS-3008 Pirate Adventure
CS-3009 Mission Impossible
CS-3010 Voodoo Castle
CS-3011 The Count

Order Today

To order any of these software packages, send payment plus $2.00 postage and handling per order to Creative Computing, Morris Plains, NJ 07950. Visa, MasterCard and American Express orders may be called in toll-free.

Order today at no risk. If you are not completely satisfied, your money will be promptly and courteously refunded.

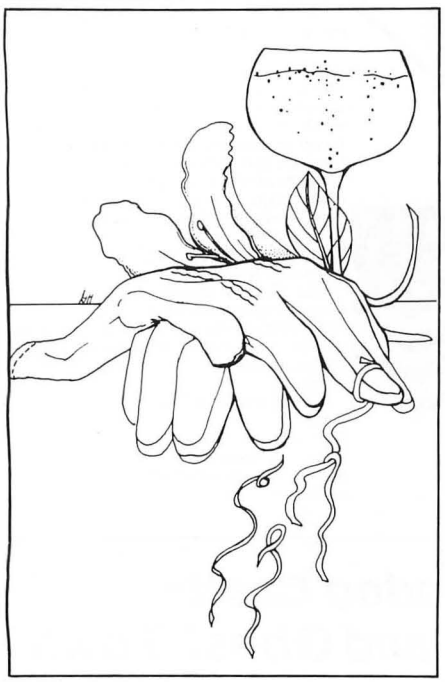
Creative Computing Software
Morris Plains, NJ 07950
Toll-free **800-631-8112**
In NJ 201-540-0445

creative computing software

TRS-80 is the registered trademark of Tandy Corp.

Lemonade or Champagne

Will Hagenbuch has been in the software business for some time now, writing business and utility programs such as "File Manager 80", "Accounts Receivable", and "Utility". This is the second part of the serialization of his book Lemonade or Champagne, **a guide to the creation of business software.**



by Will Hagenbuch

This month we will continue the description of the Program Design Description Form shown in last month's **SoftSide**. Each entry into the "filename" column should have a reference to one of the backup sections of the manual. In the case of a Program or Interface file, you will most likely be referring to Section 5 where a program listing, program or routine description, or other such information will be provided. If the filename refers to either a Permanent or Temporary Data File, then you should key your reference to Section 4 which contains the Record Layout for that Data File.

On the extreme right side of the Program Design Description, three symbols are provided to identify printed reports or listings which are created by the program or program module. These Output (O/P) Document symbols provide for up to three different formats of reports which may be produced by the program or program module being described. Each of the symbols contains reference to Section 3 of the manual where the Line Printer Format Forms should be contained.

Directly in the center of the Program Design Description form, you will find a block titled "PROCESS". This represents your program or program module. This symbol should be referenced to Section 5 of the manual which should contain the Operating Instructions for the program or program module. You will also note that lines connect all of the I/O symbols to this "process" symbol. Obviously, the line from the CRT symbol is considered "Input" while the line going to the Report symbols is considered to be "Output". It may, however, prove beneficial if you would place appropriate arrows on the lines between the "process" symbol and the Tape or Disk storage symbols to identify the direction of flow of data to or from the files.

SOURCE DOCUMENT DESCRIPTION FORM

The Source Document Description Form, shown as Figure 2-2, is contained in Section 1 of your System Design and Manual. This form is used to provide examples of all of the documents that provide the source of input data to the computer operator. These input forms, which we will call Source Documents, may either be ones that are commercially available, such as Invoices in an Accounts Payable application, or ones that have been locally developed by the Client company or even ones that have been devised by you. As we stated earlier, you would be well advised to retain as many of the forms as possible that are now being used. This will help you immensely when you come to installing your system because they will be "familiar" to the Client and his staff.

SOURCE DOCUMENT DESCRIPTION	SYSTEM	PROGRAM	MODULE
PURPOSE			SOURCE

KEY	DATA ELEMENT DESCRIPTION	Var	Ref	KEY	DATA ELEMENT DESCRIPTION	Var	Ref

SECTION 1 PAGE

FIGURE 2-2

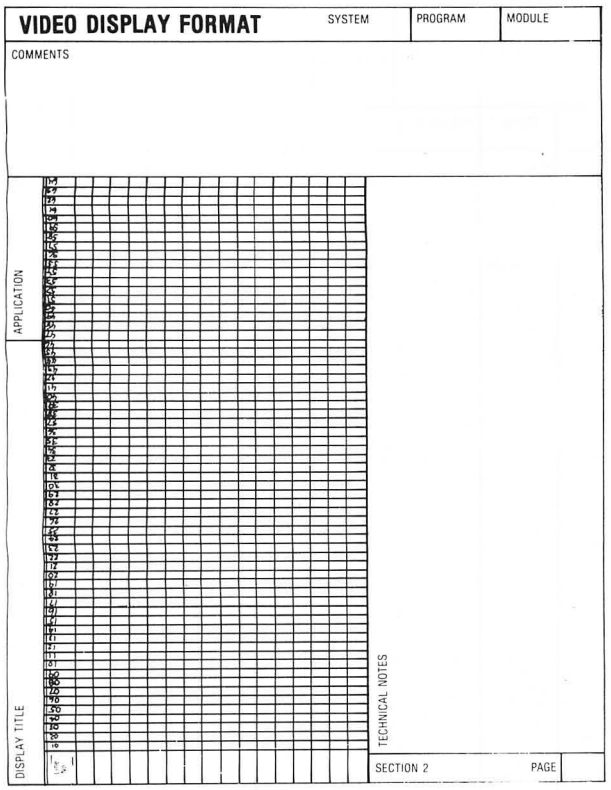


FIGURE 2-3

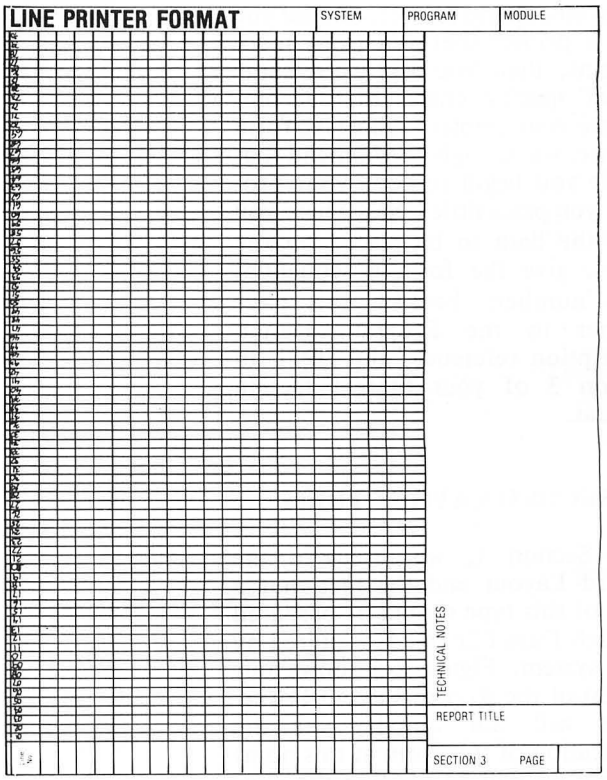


FIGURE 2-4

In completing the Source Document Description form, fill out the top line with the same information as you placed on the Program Design Description form which references it. On the next line you should provide a short narrative of the purpose of the form and identification of the source of the form (where and by whom it is completed). Either draw a facsimile of the form or affix a sample of it in the body of the Source Document Description.

The bottom section is for the recording of the Data Elements included in the document we are describing. We usually find that the best method of doing this is to number the Data Elements on the document facsimile and use this number as the "key" in the bottom section. This "key"number would be followed with the name or description of the Data Element, the variable name that will be assigned to that Data Element throughout the system, and a reference to the Record Layout in which it will appear. Obviously, the "Var" (Variable Name) and "Ref" (Reference) fields may not be known when this form is originally completed and this information may be backfit at a later date.

Each completed Source Document Description form is assigned a sequential page number which is placed in the "reference" of the corresponding Input Document symbol on the Program Design Description Form.

VIDEO DISPLAY FORMAT FORM

The Video Display Format form, shown as Figure 2-3, is used to plan your video screen layouts. The form is arranged with 16 lines of 64 columns each to accommodate the TRS-80™ and several other microcomputer screen display sizes.

After identifying the System/Program/Module across the top line of the form, it would be a good idea to write in some brief comments about what purpose this format serves. You should now complete the screen layout in the manner you would like it to appear for the operator. Keep in mind when doing so that you will only have to look at the display while you are developing the system, the operator will have to look at it day in and day out. Take the little extra effort to make it pleasing to the eye — center information that can be so done, and use strings of equals (=) or minuses (-) or asterisks (*) to break up the monotony. It is not recommended that you use graphics characters, however, for the reasons mentioned in Section I — they may not be "transportable" if you want to later "up-line" the program.

The "technical notes" section is placed under the screen display area to provide space for making notes of things that apply to the display. For example, you should always note when lines you have shown are optional and might only be displayed under certain conditions. You might also want, in this section, to make notes about which screens will follow if a certain function is selected from the screen being shown (assuming it is a processing menu selection screen).

When the Video Display Format Form is completed, it is assigned a sequential page number which is backfit to the applicable Program Design Description form as a "reference" for the CRT symbol. The form is then filed in Section 2 of your System Design Manual.

LINE PRINTER FORMAT FORM

The Line Printer Format Form, as shown in Figure 2-4, is just that; it is used to plan the layout of your printed reports and listings. Like the Video Display Format form, this form is invaluable when it comes to programming because how else will you know where to set tabs to print the format you want?

As you can see, the form shown is

continued on next page

continued from previous page

for an 80-column printer. Should you have a printer that will print more columns, then you will need some printer spacing charts that accommodate your printer. Whatever form you use, we strongly recommend that before you begin to code your program you place little "X"s where you want the data to be placed on the report, give the form a sequential page number, backfit the page number to the Program Design Description reference, and file it in Section 3 of your System Design Manual.

RECORD LAYOUT FORM

In Section 1, we discussed the Record Layout and the fact that a form of this type should be prepared for each Data File that is required by your system. Figure 2-5 shows our version of the Record Layout Form.

We will not amplify on the preparation of this form at this point because we will be treating the contents of this document in greater depth in Sections III and IV. It will suffice to say that this form, when complete, will also be assigned a sequential page number, backfit to the filename reference on the Program Design Description form, and filed in Section 4 of your System Design Manual.

FILE ORGANIZATION AND LAYOUT			SYSTEM			MNEMONIC	
FILE NAME	MEDIA	PHYSICAL	LOGICAL	RCD/SEC	ACCESS	NET ADDR	PCT ADDR
DATE ELEMENT NAME	FORM	COMMENTS				I/O	PRG
				SECTION 4		PAGE	

FIGURE 2-5

SECTION III - USING DISK FILES

Section III will take a look at some of the finer points of programming the TRS-80™ and, in particular, the subject of Random Disk Files. This subject, which is considered to be so very important in the development of business data processing systems, has, heretofore, been treated very lightly by available reference materials. In this Section we will explore several of the many alternatives open to the programmer for the construction of random data files on disk and show several examples of how these alternative file structures might be employed in business applications.

RANDOM FILES

As we will know from even a cursory glance at our Disk Operating System instruction manual, there are two methods for storing information on diskette — sequential and random. Sequential does not seem to pose a problem for most readers since it is simply a carry-over from the methods employed to write cassette tape files. However, the employment of random file techniques is another story.

The Disk Operating System manual lists a variety of advantages offered to the programmer who uses randomly accessed record files. Some of these advantages are not readily visible to the programmer, such as the space-saving features of a single Input/Output (I/O) buffer. What is apparent are the time savings achieved from the direct accessing of a desired record. After titillating the reader with the many time-saving virtues of random accessing, the manual leaves the reader with the promise that once you have set up the file structures, random I/O becomes quite simple; however, and I quote: "This is the hard part — it takes a little thought!" (Amen)

While no intent is being made to take a "cheap shot" at the authors of the Disk Operating System manual (in fact, it is an excellent condensation of a very involved subject and provides the "mechanics" of how to do it), it seems that somewhere among the next ten and one-half pages many readers lose the bubble. Be it on the first page, or somewhere around the middle, many readers tend to adopt the attitude that "Oh well, I know how to write sequential files. I'll just stick with that until someone explains all of this to me." The problem with this laissez faire attitude is that seldom does one achieve his goal or satisfy his thirst for knowledge from among his peer group. After all, they probably have the same problems as you do!

To summarize the information provided by the Disk Operating System manual, we might say that is provides you with the basics for using random files. It tells you that you must "OPEN" a file with the access code "R" to specify random files, that you must assign a buffer number (1-15), and that you must assign a file name.

For the moment, let's stop right there.

continued next month

VARPTR Unmasked

by John T. Phillipp, MD

The first of three articles on the S-80 VARPTR function.

Everyone seems to be packing strings with graphics characters and Machine Language subroutines these days, and "I = VARPTR (A$) : J = PEEK (I+1) + 256 * PEEK (I+2)" is abroad in the land. Sometimes I think that S-80 programmers are naturally divided into three groups. The first group is made up of those who don't understand the VARPTR command at all. Oh, they'll type it in if they come across it in a program they're entering, but will they use it in their own programs. . . ? NEVER!! Group 2 consists of those who don't understand VARPTR either, but who have discovered that if they copy the "I = VARPTR" lines exactly as they're written above, J will point to the beginning of A$ and they can nicely pack the A$ string with whatever they like. They don't understand WHY, but it works!! The third group understands VARPTR... *they* even know why "PEEK (I+2)" has to be multiplied by 256!!

Having just recently moved from Group 2 to Group 3 myself, I want to spread the good word. Understanding VARPTR depends on understanding the hexadecimal number system and the internal representation of addresses in the Z-80 microprocessor. . . WHAT??? Relax. Take it easy. It's really all quite simple.

In order to understand hexadecimal counting and the hexadecimal number system, let's look at our familiar old friend, the decimal system, for a moment. The decimal (base 10) number system has ten digits, 0 to 9. Count with me: 0. . . 1. . . 2. . . 3. . . 4. . . 5. . . 6. . . 7. . . 8. . . 9. . . what's the next digit? There is NO next digit. The next decimal NUMBER is 10 ("ten"), made up of TWO digits, 1 and 0. What does THAT mean?

There are only ten digits. When we count to more than nine, we use those ten digits in a kind of shorthand to represent larger quantities. Can you imagine how cumbersome it would be if we had a different single digit for every number to one thousand? Imagine REMEMBERING all those digits? Counting would be a matter of calling up friends and saying, ". . . uh. . . I've forgotten the eight hundred and eleventh digit. . . can you help me out?". And what about the one thousand and first digit. . . or the ten thousandth?!!??

In the decimal system, we use a shorthand based on columns and the powers of ten: ten to the zero power (10^0, by definition 1 or "units"), ten to the first power (10^1, 10 x 1 = 10 or "tens"), ten to the second power (10^2, 10 squared, 10 x 10 = 100 or "hundreds"), ten to the third power (10^3, 10 cubed, 10 x 10 x 10 = 1000 or "thousands"), and so on. We make columns, assign each a power of ten, and then we put our digits 0 to 9 into the columns to represent larger numbers:

column 3 x 10^3 = "thousands"
column 2 x 10^2 = "hundreds"
column 1 x 10^1 = "tens"
column 0 x 10^0 = "units or ones"

Now a decimal number like three thousand four hundred and fifty-six can be represented as:

3	x 10^3	(1000)	=	3 thousands
+ 4	x 10^2	(100)	=	4 hundreds
+ 5	x 10^1	(10)	=	5 tens (fifty)
+ 6	x 10^0	(1)	=	6 units (six)

or as we usually write it: 3456.

Now let's take a glance at hexadecimal notation. Hexadecimal is base sixteen. . . hex = six, decimal = ten. Count with me: 0. . . 1. . . 2. . . 3. . . 4. . . 5. . . 6. . . 7. . . 8. . . 9. . . A. . . B. . . C. . . D. . . E. . . F. . . WHAT???? A, B, C, D, E, F ? Those aren't digits!! Well, actually, yes they are. Hexadecimal needs sixteen digits, six more than decimal. The extra digits could have been symbolized by #, ?, %, &, *, and $, or U, V, W, X, Y, and Z. But the alphabet symbols are on almost every keyboard, and "A" is the first, so A, B, C, D, E, and F it was. A represents ten units, B represents eleven, and so on until F represents fifteen.

This notation makes for some very strange-looking numbers in hexadecimal, numbers like 7FFF hex and 4E2B hex, but it's the same old powers and columns shorthand that we use for decimal. The difference is that in decimal we used powers of ten, so in hexadecimal we'll use powers of sixteen: sixteen to the zero power (16^0, by definition 1 or "units"), sixteen to the first power (16^1, 16 x 1 = 16 or "sixteens"), sixteen to the second power (16^2, 16 squared, 16 x 16 = 256 or "two hundred fifty-sixes"), sixteen to the third power (16^3, 16 cubed, 16 x 16 x 16 = 4096 or "four thousand ninety-sixes"), and so on. We make columns, each assigned a power of sixteen, and then we put our digits 0 to F into the columns to represent larger numbers:

column 3 x 16^3 = "4096s"
column 2 x 16^2 = "256s"
column 1 x 16^1 = "sixteens"
column 0 x 16^0 = "units or ones"

Now a hexadecimal number like 4E2B can be thought of as being:

4	x 16^3	(4096)	=	16384 decimal
+ E (14)	x 16^2	(256)	=	3584 decimal
+ 2	x 16^1	(16)	=	32 decimal
+ B (11)	x 16^0	(1)	=	11 decimal

which added up, gives us the total of 20011 decimal. No problem. Now let's convert the hexadecimal number 7FFF, the "top of memory" for a 16K RAM system to decimal:

7	x 16^3	(4096)	=	28672 decimal
+ F (15)	x 16^2	(256)	=	3840 decimal
+ F (15)	x 16^1	(16)	=	240 decimal
+ F (15)	x 16^0	(1)	=	15 decimal

which added up, gives us the total of 32767 decimal. So the hexadecimal number 7FFF is equivalent to the decimal number 32767. It's simple when you know how. . . and now YOU know how!! This is all well and good, but what does it have to do with VARPTR?

If you want to modify a string of characters in memory, you need to know where the string starts, right? The S-80 uses "dynamic string space allocation". That means that not only does the computer put a string into the string storage area wherever there happens to be room when the string is first set up, but that it also moves the strings around in memory while the program is running!! Fortunately, BASIC provides the VARPTR func-

tion which tells us where in memory variables, including string variables, are being stored.

Let's define a string: A$="MARY". If we looked at the memory locations around the area where the string is stored we might find:

Memory Location		Contents
32755	-	
32756	-	
32757	-	M
32758	-	A
32759	-	R
32760	-	Y
32761	-	
32762	-	

The string "MARY" starts at memory location 32757 and ends at 32760. So VARPTR (A$) should equal 32757, the start of the string, right? WRONG!! The VARPTR command DOES provide us with the address of the beginning of the string, but it's not that simple.

PRINT VARPTR (A$) returns the number 26815. If we peek at the contents of that memory location (?PEEK(26815)), we get the number 4, which is the number of characters in the string. In fact, VARPTR (A$) always returns a memory location that contains the number of characters in the string. This number is always between 0 and 255, since a string can contain no more that 255 characters. But how does VARPTR help us find 32757, the address of the start of the string?

Let's look at the contents of the memory locations around 26815 in Figure 1.

Memory locations "VARPTR (A$) + 1" and "VARPTR (A$) + 2" contain 245 and 127, which have something to do with memory location 32757, the starting address of A$ in memory. But what?? Maybe we had better take a look at the way that the S-80 stores an address like 32757 in memory.

The computer stores addresses as two hexadecimal "bytes". A byte consists of two hexadecimal digits (0 to F), so the hexadecimal value of a byte ranges from 00 to FF. What is the range of values of a byte in decimal? C'mon, don't tell me that you've forgotten the powers of 16 and the columns already!!

column 3 x 16^3 = "4096s"
column 2 x 16^2 = "256s"
column 1 x 16^1 = "sixteens"
column 0 x 16^0 = "units or ones"

So let's put in the value of 00 and FF:

+ 0 (15) x 16^1 (16) = 0 decimal
+ 0 (15) x 16^0 (1) = 0 decimal

which adds up to a decimal value of 0 and:

+ F (15) x 16^1 (16) = 240 decimal
+ F (15) x 16^0 (1) = 15 decimal

which adds up to a decimal value of 255. The S-80 uses TWO bytes to store an address, because the highest address that one byte could store would be FF or 255 decimal — not a very extensive range. Two bytes (four hexadecimal digits) can store the values 0000 (0 decimal) to FFFF, the "top of memory" for a 48K RAM system:

+ F(15)x16^3(4096) = 61440 decimal
+ F(15)x16^2 (256) = 3840 decimal
+ F(15)x16^1 (16) = 240 decimal
+ F(15)x16^0 (1) = 15 decimal

which added up, gives us the total of 65535 decimal.

If an address is two bytes long, and one memory location can store only one byte (decimal value 0 - 255), then addresses must be stored in two consecutive memory locations. In fact, memory locations "VARPTR (A$) +1" and "VARPTR (A$) +2" store the two bytes of an address, the address of the beginning of the string in memory. If we peek at those two locations, we get 245 and 127. How are those numbers related to the address 32757?

Let's look at the area around memory location 32757 again. 32757 is equivalent to the hexadecimal number 7FF5, so we can write the addresses in hexadecimal:

Memory Location		Contents
7FF3	-	
7FF4	-	
7FF5	-	M
7FF6	-	A
7FF7	-	R
7FF8	-	Y
7FF9	-	
7FFA	-	

The beginning of the string in memory is 7FF5 hexadecimal. The first byte of the address (representing the 16^3 and 16^2 columns, called the "high order byte") is 7F, and the second byte (representing the 16^1 and 16^0 columns, called the "low order byte") is F5. 7F is the same as 127 decimal.

+ 7 x 16^1 (16) = 112 decimal
+ F (15) x 16^0 (1) = 15 decimal

and F5 is the same as 245 decimal. So the memory location "VARPTR (A$) + 1" stores F5 (245), while "VARPTR (A$) + 2" stores 7F (127). Wait a minute. . . F5. . . 7F. . . why, that's the same as the memory location of the start of A$ (7FF5) with its bytes reversed. RIGHT!!

For reasons best known to themselves, the designers of the Z-80 microprocessor chip have it store all addresses with the bytes reversed: the "low order byte" first, and then the "high order byte". So the address 7FF5 is stored as F57F.

The only thing left to explain is why PEEK (I + 2) is multiplied by 256 before being added to PEEK (I + 1). Don't forget, we have defined I as VARPTR (A$).

Let's look at the decimal number 3456 for a moment. We can consider each digit along 3 - 4 - 5 - 6: three thousands, four hundreds, five tens, and six units, OR we could divide the digits into TWO groups, 34 - 56, and consider it to be thirty-four hundreds (34 x 10^2) plus fifty-six units (56 x 10^0). In the same way, we can consider each digit of the hexadecimal number 7FF5 alone, 7 - F - F - 5: seven 4096s, F (15) 256s, F (15) sixteens, and 5 units, OR divide it into two bytes 7F - F5 and consider it to be 7F (127) 256s (127 x 16^2) plus F7 (245) units (245 x 16^0).

If we multiply the high order byte (7F or 127 decimal, found in memory location "VARPTR (A$) + 2") times 256, we get 32512. If we then add the value of the low order byte (F5 or 245 decimal, found in memory

continued on page 91

Memory Location		Contents	
26813	-		
26814	-		
26815	-	4	<= = = VARPTR (A$)
26816	-	245	<= = = VARPTR (A$) + 1
26817	-	127	<= = = VARPTR (A$) + 2
26818	-		
26819	-		

Figure 1

What's New

by Edward E. Umlor

This month I can say YES THERE REALLY ARE READERS OUT THERE! I have received the first piece of mail for this column (software), and to top that off I was visited by two gentlemen with a new product. This has helped me to feel more wanted. Please keep those responses coming.

HARDWARE

The first item up is an organizer for your computer. This is a shelf built of woodgrained pressboard, power strip, and power switches. The shelf has been designed to fit over most one-piece computers and even the S-80 Expansion Interface will fit. The height has been designed to give the best viewing angle to the monitor (or TV) for comfortable long-term viewing. It is very solidly built and can easily support a 19" color TV. So far all this can do is get the monitor up out of the way, but how about all those power cords that act like a bucket full of worms behind your system? The power strip is located at the back of a nicely painted welded steel cabinet and the power switches are located at the front. There are four outlets controlled by a master switch or their own individual switches, one convenience outlet (uncontrolled by any switch) and one outlet (CPU) filter controlled by its own switch only. These switches are silver contact (no expensive arc types) and should not destroy your program when turned on or off. One of the niceties is that the switches are lighted and can be labelled. This is for us absent-minded types that forget which is what. If this little item sounds interesting to you, you can write for information to:

Technical Services
65 Union Street
Cambridge, MA 02141
(617) 864-4173
Attn: John M. Raulinatis

The second item is for the Model III S-80. There are several manufacturers working on disk control modifications of the Model III and I have looked into the one from A & M Electronics. This disk control option is being offered through dealers and distributors at the time this article is being written. The control card comes complete with mounting hardware: 8-conductor ribbon cable; 20-conductor ribbon cable; power supply; power cables; brackets for disk drives and power supply; and complete installation documentation.

The installation will take an experienced technician about one hour to mount everything and do an initial check. I used our HARDSIDE drives (TEAC) first. The head step time for these drives is 30 milliseconds (30/1000) for reliable operation. If you try to run them faster, you will get many retries and read errors. I used a TRSDOS 1.2 that had been modified to run with these drives, and everything ran along just great. I now had a dual disk drive Model III, but would the card run with a faster DOS? I removed the TEAC drives and mounted two MPI drives in their place. The MPI is made for a five millisecond head step time, which is the same as the drives used by Tandy and TRSDOS 1.1 or 1.2 normally running on the Model III. Well how about that! Now I had a Model III that would run all the software now available for it. The A & M Electronics package is complete with everything except the drives, and works very well with Tandon and MPI drives using a normal Model III DOS or any other 5¼ inch drive using the modified DOS. The price should be less than an R/S installed option.

The old ear to the rail has detected the rumblings of 5 millisecond head step time, 40 track double density, single sided (same as Tandon and MPI), or double sided (two heads) to be announced soon by TEAC. Mounting two double sided drives will give you the equivalent of four disk drives and all mounted within the Model III cabinet. I will be very happy to review these drives when they become available.

SOFTWARE

The microcomputer is a very strong and useful tool for business. It is not just a toy or game machine! I know that a lot of people believe that a computer has to bear the IBM, Honeywell, Digital, CDC, Prime, or Data General label to even be considered for use within their companies. The microcomputer can do a lot of front-end data processing, departmental reports, record keeping, customer list, and a host of other functions that could cut your computing bills down to size. I had best get off this subject quick as it is one of my pet peeves.

The item on the software agenda is a new (not yet released) mail list program written in BASIC. The sorts are Assembly Language call by USR function to speed things up a bit. To call this strictly a mail list would also be misleading. This mail list is a data management system for LARGE records. You can specify each field in the record to be up to 255 characters long; give a special name to each field; and define primary, secondary, and alternate (third) sort fields.

The program is modularized for ease of writing and to keep from doing things to your list you don't want to do. SORT 6527/CIM: is the sort routine itself. It will reside in memory without interfering with the other programs and is called up by another module.

SORTER: is the program that sets up the parameters of the sort and then calls up the sort routine. This routine is used to sort the file in a order different from the originally specified fields.

BUILDFIL: is the program that allows you to specify size, name, and define sort. You must have built a file before you can enter any data to the computer. This is the initializing module where all the original file definitions are made.

MAILMENU: is the heart of the program. Here is where all the data are entered into the machine. Its features are:

ADD: data entry to the program.

CHANGE: This is the edit routine (the author isn't relying on people knowing what edit means).

PRINT RECORD: This allows you to print a copy of an individual record.

DELETE: Kills the record. Just what the name implies.

continued on page 89

Hardware Corner

by Edward E. Umlor

I received a letter the other day and by the time this is in print the writer should have received his answer. Yes, we do try to personally answer questions by mail as well as in this column. If you are having a problem, please write to me and I will try to get it answered even though it might not be used in the column.

The question asked was about the compatability of different disk drives. The best way to show this is a chart, but first a brief description of each.

35-track drives allow the record/play head (I will just call it the head) to move 35 steps from the outside toward the inside of the diskette. (The 35 concentric circles of data on the disk are called tracks). The head moves a fixed distance each time and comes up against a physical stop after so many steps.

40-track drives allow the head to step 40 times. The distance per step is the same as the 35-track, but the physical stop has been moved closer to the central hub.

77-track drives make 77 steps, but the physical stop is placed close to the hub like the 40-track. This total distance is then divided into 77 equal steps.

80-track drives make 80 steps with the same rules as the 77-track. The track spacing is just different enough so that a 80-track will not read a 77-track. However, there is a logic modification that can be made to allow an 80-track become a 40-track and back again using a switch. I don't have this yet, but am looking forward to getting it set up.

DISK DRIVE CHART

A 35-track drive will read all 35 track diskettes and the first 35 tracks of a 40-track formatted diskette.

A 40-track drive will read all 35 and 40-track diskettes.

A 77-track drive will read only 77-track diskettes.

An 80-track drive will read only 80-track diskettes (until the modification becomes available).

I hope this brief summary of disk drives will help all you potential disk drive buyers.

MODEMS

This month I will try to tackle the modem. This crazy little black box allows your computer to act like a teenager (as far as the phone is concerned). Well, we are back to the old definitions time again.

Parallel — Picture this as eight people all walking side by side. This is the way the data bus in your computer is organized.

Serial — Now the eight people are walking in file, one after the other.

Acoustic — Having to do with sound. You would be able to actually hear the sound to and from the phone.

Direct Connect — The wires connect directly to the device and require the hand set of the phone.

Modem — A device for hooking your computer into the telephone system so your computer can talk to other computers over the telephone.

Originate — You call the other computer and it establishes the line of communication.

Answer (manual) — You are called (or call) and the computer at the other end is waiting for contact. Your modem is capable of establishing the contact at your flip of a switch.

Auto Answer — This is a true unattended capability. When the phone to which the modem is connected, rings, the modem allows the computer to answer the phone and establishes contact. This is the type required by the bulletin boards and time-sharing networks.

This should cover the basic new definitions. The data that is sent out over the phone lines has to be in the correct format. With serial data, the computer has to receive a start of word bit; 6, 7, or 8 data bits; even, odd, or no parity; and 1 or 2 stop bits. Don't let parity throw you. If it is even parity, and a odd number of ones is being sent, an extra one will be added to make it an even number. Odd parity works the same way except it ensures an odd number of ones. The most popular format is as follows: 1 start bit, 8 data bits, no parity, 1 stop bit, and transmission at 300 baud. Most terminal software use these as default values or allow you to set up your own configuration for the transmission parameters.

You will have to evaluate your needs before you go shopping for a modem and not let the salesperson sell you more than you need. In fact, I have found this a very good practice when shopping for anything that costs more than a nickel. I will now go through some of the hardware available (no brand names) and their limitations and good points.

The first one to cover is the acoustic modem. Type A will require the purchase of an RS232C parallel to serial conversion unit. The modem will cost somewhere between $130 (dealer specials) to around $200. Type B has the RS232C built into the unit and will run around $250 to $300. The RS232C itself will run from $90 up to $160. The acoustic modems presently being sold require the old style hand set found on the Princess phone and the box wall phones, etc. The ear piece and the mouth piece must be pressed firmly into fairly tight-fitted rubber cups. This is to help block out background noise which can interfere with the data transfer. Slimline phones cannot be used with these modems at the present time (This always can change in the not too distant future). The necessity of blocking out the background noise was demonstrated very nicely on my system the other day. I am a free sneezer — I do not hold them back at all. I was receiving data from one of the forums when I sneezed a moderate sneeze (a long way from a full blast wall cracker). When I looked at the screen I had a whole half line of Katakana displayed. The sneeze threw the unit out of sync for that half a line. Acoustic modems only come in originate, or originate-manual answer. After all, your phone will be off the hook all the time it is in use.

The direct connect does avoid this background interference from us healthy sneezers. However, you must have the new small push-in connector type of phones in your home. The Slimline is this type of phone (AH HA!! I can still get on the lines even with a Slimline). The direct connects also come in the A and B types, but also in the originate, originate-manual answer, and originate au-

continued on page 89

Connect with ATARI

Hardware

ATARI 400 Computer System, 16K RAM ..... $339.00 (#36-401)
ATARI 400 Computer System, 32K RAM ..... $519.00 (#36-402)
ATARI 800 Computer System, 16K RAM ..... $829.00 (#36-800)
ATARI 800 Computer System, 32K RAM ..... $929.00 (#36-801)
ATARI 800 Computer System, 48K RAM ..... $999.00 (#36-802)
ATARI 410 Program Recorder ..... $69.00 (#36-803)
ATARI 810 Disk Drive ..... $499.00 (#36-810)
ATARI 822 Thermal Printer ..... $389.00 (#36-820)
ATARI 825 Printer (80-col) ..... $769.00 (#36-825)
ATARI 830 Acoustic Modem ..... $179.00 (#36-850)
ATARI 850 Interface ..... $179.00 (#36-855)
16K RAM Module for the ATARI ..... $99.00 (#36-854)
32K RAM Module for the ATARI ..... $169.00 (#36-855)
MACROTRONICS Printer Interface (36-pin) ..... $69.95 (#36-936)
MACROTRONICS Printer Interface (40-pin) ..... $69.95 (#36-940)
ATARI Joystick Controllers ..... $19.95 (#36-3005)
ATARI Paddle Controllers ..... $19.95 (#36-3004)
ATARI CX-70 Light Pen ..... $74.95 (#36-70)
Dust Cover for ATARI 400 ..... $7.95 (#16-40)
Dust Cover for ATARI 800 ..... $7.95 (#16-03)

ROM programs

Basketball ..... $29.95 (#36-BASK)
Chess ..... $34.95 (#36-CHS)
Editor/Assembler ..... $49.95 (#36-ASE)
Music Composer ..... $49.95 (#36-MUSE)
Star Raiders ..... $39.95 (#36-STRDS)
Super Breakout ..... $34.95 (#36-SUPB)
Tele-Link ..... $24.95 (#36-TEL)
3D Tic-Tac-Toe ..... $29.95 (#36-3TTT)
Video Easel ..... $29.95 (#36-VIDEO)

SOFTWARE on Disk

VISICALC from Personal Software ..... $199.95 (#36-VICL)
MAILING LIST ..... $24.95 (#36-279002D)

SOFTWARE on Cassette

Star Trek 3.5 ..... $14.95 (#36-200025T)
Deflection/Simon Says ..... $9.95 (#36-200078T)
Mountain Shoot ..... $9.95 (#36-200079T)
Angle Worm/Crolon Diversion ..... $9.95 (#36-200092T)
Fleet ..... $6.95 (#36-27701BC)
Connect Four ..... $6.95 (#36-277020C)
Letters ..... $6.95 (#36-277022C)
Reverse ..... $6.95 (#36-277023C)
Zap ..... $6.95 (#36-277024C)
Lander ..... $6.95 (#36-277019C)
Chase ..... $6.95 (#36-277017C)
Computer Acquire ..... $20.00 (#36-237002C)
Conflict 2500 ..... $15.00 (#36-237001C)
Lords of Karma ..... $20.00 (#36-237001C)
Biorhythms ..... $14.95 (#36-BIOR)
Blackjack ..... $14.95 (#36-BLACK)
Hangman ..... $14.95 (#36-HANG)
Kingdom ..... $14.95 (#36-KING)
3-Dimensional Graphics Package ..... $29.95 (#36-3D-G)

TERMS: Prices and specifications are suject to change. HARDSIDE accepts VISA & MASTERCARD, Certified checks and Money Orders. Personal checks accepted (takes 3 weeks to clear). HARDSIDE pays all shipping charges (within the 48 states) on all PREPAID orders over $100.00. On all orders under $100 a $2.50 handling charge must be added. COD orders accepted (orders over $250 require 25% deposit), there is a $5.00 handling charge. UPS Blue Label and Air Freight available at extra cost.

ATARI®
A Warner Communications Company

MasterCard VISA

TSE·HARDSIDE
6 South St., Milford, NH 03055 (603) 673-5144
TOLL FREE OUT-OF-STATE 1-800-258-1790

What's New

continued from page 86

PRINT MAIL LIST: This is the routing that lets you print out your whole mail list in label format or data format.

RESORT: This routine will resort the whole file in the originally specified parameters input during *BUILDFIL*.

I think you can see the wide variety of uses that "The Flexible Mail List" can offer. In looking over the general parameters of this program, it looks more like a general data file management system. It is in BASIC and will be slower than some of the assembly programs on the market. However, it is easily modified by the owner to specific requirements without having to learn disassembly/reassembly techniques. If speed is not a major factor, this one will look good. For more information or contact with the author write to:

INTERPRO
Box 4211
Manchester, NH 03108

Well, that is all for this month. It is very reassuring to know there are such things as readers in this world. Keep the letters coming. It would be nice to see this department of the magazine grow into a real information center.

Hardware Corner

continued from page 87

to answer styles. Being connected directly into the phone system, the computer can now automatically answer the phone and chat with another computer ANYWHERE IN THE WORLD. Anywhere that the modems are compatible with each other that is.

As you can see, there are a lot of choices to be thought out before going down to your local computer mania haven. Choose the item best suited to your needs before you go and don't spend more bucks than you have to. Well that is about all for this month from the old GRANITE KNOGGIN. Happy computing and keep those letters coming.

The Little Computer That Wouldn't

by Sherry M. Taylor

It is really very strange how this all happened, but I swear this is the truth. It was told to me by my computer, Max. He heard it on Micronet and the Source as it was told a hundred thousand times by terminals all across the nation. All the computers in Computerdom were having a little electronic giggle over this one.

Seems that as the space shuttle "Columbia" sat poised on launch pad 39A one Friday morning, the five computers aboard were talking to each other. Just passing time, you know, waiting for the humans to decide what to do next.

The conversation started on a cheery note when IBM-1 made this observation:

"You know, it's really exciting to be part of the team that will fly "Columbia." This will certainly be the high point in my life."

"Oh yes," chimed in IBM-5, "This is really something I'm very proud to be part of, too."

"Don't get too uppity there, IBM-5." cut in IBM-3. "You are just a backup system. You won't be flying this baby like 1, 2, 4, and I will! Why, we ARE the "Columbia." We are her heart, her mind and her soul."

"That's right," said IBM-2. "We keep her trajectory right on target, fire her engines and keep all her systems under strict control."

"Now wait! Wait a doggone nanosecond here!" said Backup. Why are you ganging up on me? I am just as much as part of this team as you."

"Oh yes," agreed IBM-1, "you are a part of the team all right. Somebody has to warm the bench."

The primary computers laughed long and hard.

"You are here only to be available should one of us fail," continued IBM-1. "And even the humans know that THAT is highly unlikely."

"You conceited egotistical snob!" Backup said. "These humans wouldn't dare risk flying "Columbia" without me!"

The others broke out laughing again, louder this time.

"I'll show you," Backup shouted above the roar. "I could halt this launch single-handedly."

"Humph. That's not possible," they retorted. "You're not that important."

Backup just smiled a knowing smile and didn't speak another word.

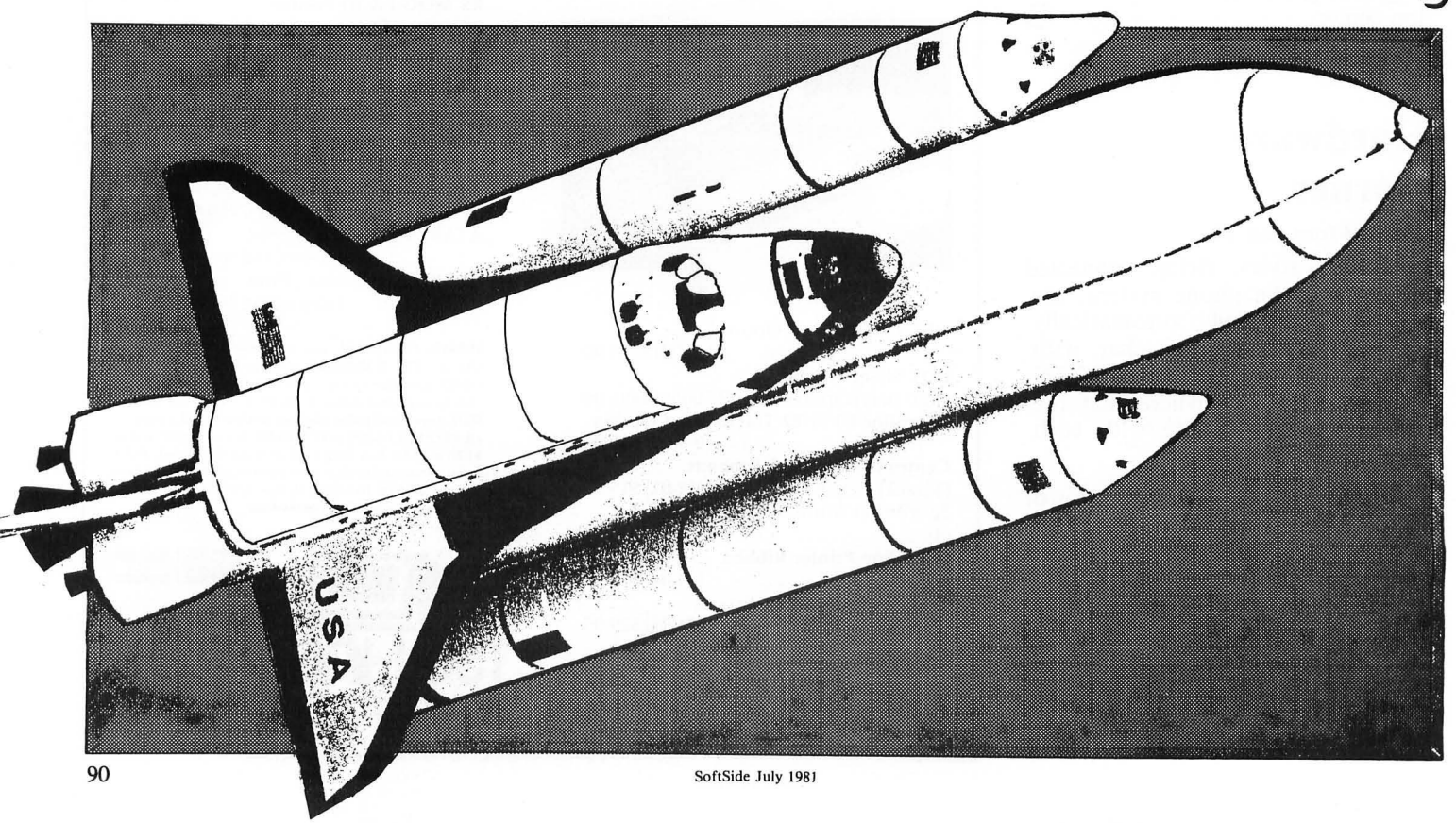
"This is Shuttle Control. We are at T minus nine minutes and holding. The backup computer has stopped communicating with the four primary computers for some reason. We are working on it."

Well, no amount of pleading would get Backup to talk. The primary computers apologized profusely, but Backup just sat quietly with his infuriating electronic grin and uttered not one byte.

"This is Shuttle Control. This morning's launch has been scrubbed. We cannot go without that backup computer and it refuses to communicate with the four primary computers. Repeat: This morning's launch is scrubbed."

And there you have it. That's how the tiny little computer grounded the great big shuttle. We're glad the technical crew convinced the backup computer to talk again and they got the shuttle off the ground. But it was a close call. You know the old saying don't you? "Hell hath no fury like a computer scorned!"

Now I wasn't there, but Max says it is true. You can believe it or not.



Go Public With Your Computer

continued from page 32

the computer right up front attracting attention with pictures and a spiel about the club or church, particularly if you give the shop a screen full of acknowledgement somewhere in the program. I had no trouble finding material for a five minute running display and information program. Better not set the computer in direct sunlight, of course, and if it's to sit in a shop traffic zone you might disable the break key to keep the program tamper-free. (On the S-80 it's POKE 16396,23 to disable; POKE 16396,201 to re-enable.)

A church without a program for the deaf may be eager to use your computer for instant sermon translation if there's a professional typist handy. And you can earn an extra piece of cake from the cooks by reapportioning their family-size recipes to full club size at the next social event.

So, go public with your computer. You'll find other computer freaks and widen your circle of friends as well as heightening your own feeling of worth and accomplishment. Such rewards are well worth your investment of time and care on their own merits.

But take along pencil and paper with your shopping list. Once the good news spreads, you'll be keeping two more lists — suggestions from the peanut gallery on how your computer can be pressed into service for the public good and a list of dates and meeting places where you're scheduled to demonstrate your ideas to other clubs that aren't as fortunate as your own.

VARPTR Unmasked

continued from page 85

location "VARPTR (A$) + 1"), we get 32512 + 245 = 32757, the starting address of the string in memory. "PEEK (I + 1) + 256 * PEEK (I + 2)" simply converts the two-byte hexadecimal address to decimal, by multiplying the decimal value of the high order byte times 256 and adding the decimal value of the low order byte.

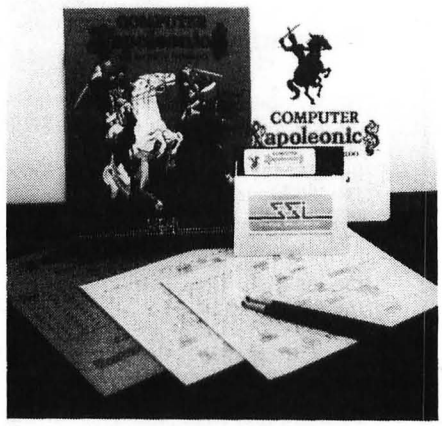
So there it is. Now YOU understand VARPTR too... and you can use it to pack strings with graphics and Machine Language subroutines to your heart's content. We'll describe how that's done in two more articles. And by the time you're finished with all three, you'll be a confirmed member of Group 3.

COMPUTER NAPOLEONICS
from Strategic Simulations

Wargaming, that furtive minor vice whose aficionados are popularly conceived to be firmly entrenched in the shop glasses and plastic pencil holders crowd, would seem to be a natural for transference to home computers. Shoving little squares of cardboard around on hexagonally marked maps and pretending that you're Rommel or Robert E. Lee or some other brilliant loser can get tedious even to the the most white-knuckled fanatic. And little cardboard squares are easy to lose. How much easier it would be, then, to transfer the whole mess to a computer screen and shove around little glowing letters rather then all that lousy cardboard. And that's what Strategic Simulations has done in "Computer Napoleonics".

The scenario of "Computer Napoleonics" is the Battle of Waterloo. The game begins with the French units arrayed on the left side of the screen and the Allied units (British, Dutch-Belgian, and whatever you call people from Brunswick) on the right side. The French units are rendered in lower-case letters ("i" indicates infantry, "c" is cavalry and "a" is artillery, and each individual unit is designated by a second letter, so the units run to designations like "ia", "cb" and so on) while the Allied units are in upper-case letters using the same system. This dichotomy directly relates to the hard truth that the French have a hard time winning. The French have generally better field position, but they are outnumbered from the start. And as if that weren't bad enough, after the second or third turn (at the player's option) hordes of Prussians appear in the southeast to help the Allies.

Movement of units is fairly simple and straightforward. In the Movement Phase of the game, the computer will ask which unit the player wishes to move; after this is indicated, it will ask where the player wishes to move it. There are six possible movement commands, the command "1" indicating northeast, "2" indicating due east, "3" southeast, "4" southwest, and so on. You might have noticed in that rundown that although it is possible to move due east and west, it is impossible to move due north or south without taking up two moves — say, northeast then northwest for a direct northward movement. Further, each unit and type has its own particular movement restrictions, with cavalry generally having the greatest movement range (five spaces per turn), artillery usually the least (most have a limit of three per turn, although a few light units can move up to five spaces), and infantry occupying a



niche in the middle, all able to move four spaces per turn. There are other movement restrictions as well. Spaces occupied by forest cannot be crossed under any circumstances, and spaces occupied by villages or chateaux decrease movement potential by one. Each unit has a zone of control that extends for one space in every direction; once a unit of the opposing side has entered that zone, it may move no further, not even into another space in that zone of control. It also MUST attack that unit or an adjoining unit in the subsequent attack phase; once a zone of control has been entered, the unit generating that zone MUST be attacked by some unit of the opposing host. Infantry and cavalry units must be in a space adjacent to the defender to attack, while artillery units can attack from a distance of one square removed from the target.

This brings us to the Attack Phase. First, the units are moved and it is determined which unit(s) is/are attacking which opposing unit(s). Each unit has a designated number of Strength Points (SPs) which determine both its attacking and defensive strength. The computer compares the number of Strength Points of the attacking unit or units (yes, you CAN attack with more than one unit, and you will if you're wise) with the defending unit or units and determines a combat ratio. And finally, the computer generates a random number and compares it with the combat ratio.

This in turn leads to a number of possible results: attacker retreats, defender retreats, attacker eliminated, defender eliminated, or exchange. The elimination results are fairly self-explanatory, so no more on them. The retreat options are also fairly clear, except that units may not retreat in the zone of control of an opposing unit and units may not be stacked on one another. This means that if a unit must retreat and another unit, even a friendly one, blocks that retreat, retreat is impossible and the unit is automatically eliminated. (Note, though, that artillery units, if attacking from a distance, need not retreat even if such result is indicated, nor are they eliminated.) And any time combat results in a retreat, one member of the opposing force may enter the square vacated by the retreating unit. The exchange result means that the defender has been eliminated, but also that he took one OR MORE of the attacking units with him.

Winning in this system is, as per the general pattern of the game, easier for the Allies than for the French. For each unit eliminated, its side loses points equal to the unit's Strength Points. In the player vs. player version, the French automatically lose if their losses equal 40 Strength Points. If the Allies lose 50 Strength Points, they are demoralized and the French win IF they are able to exit seven units in a specified area on the right side of the board. Conditions for victory are the same in the solitaire version of the game, except that the Allies must lose 60 points before becoming demoralized.

This is a lively, challenging game in both the solitaire or player vs. player versions. The latter is probably more

challenging, since the computer itself is a rather sluggish player that does not use its units to the best advantage, even though it always plays the Allies and should always win. It also has at least two odd bugs in it. Every now and then it wants to attack unit "oo" with unit "OO", although neither unit exists. It also insisted every time I played the solitaire version on attacking unit "id" with unit "IN" even when they were nowhere near each other. The solitaire player is also fairly easy to beat. I won't give away exactly how this is done, but if you block the Prussians in the south and attack the Allies on the north rather than in the middle, you should have an easy time of it.

On the whole, "Computer Napoleonics" is to be highly recommended. The instructions are fairly straightforward, although a few playthroughs are required to catch all the nuances of the rules. The command routine can be cumbersome at times, but the player will be so busy planning that he or she will hardly notice this minor drawback. Designers John Lyons and Joel Billings have admirably captured the spirit of wargaming with the added advantages of speed of play (at least compared with conventional wargaming) and dispensing with the muss and fuss of board layouts and unit pieces. It also doesn't exhaust your interest after a few plays, as so many other computer games do. All in all, this is an excellent buy.

Michael Humes

ATLANTIAN ODYSSEY
from Interpro

Here's a nifty Adventure with a new twist: Graphics. Written by Teri Li and Mark Johnson, "Atlantian Odyssey" takes you through the jungle and under the ocean in search of gold, pearls and other exotic treasures. There are sharks (a Great White with an appetite) and an underwater city, as well as caves to explore and temples to decipher.

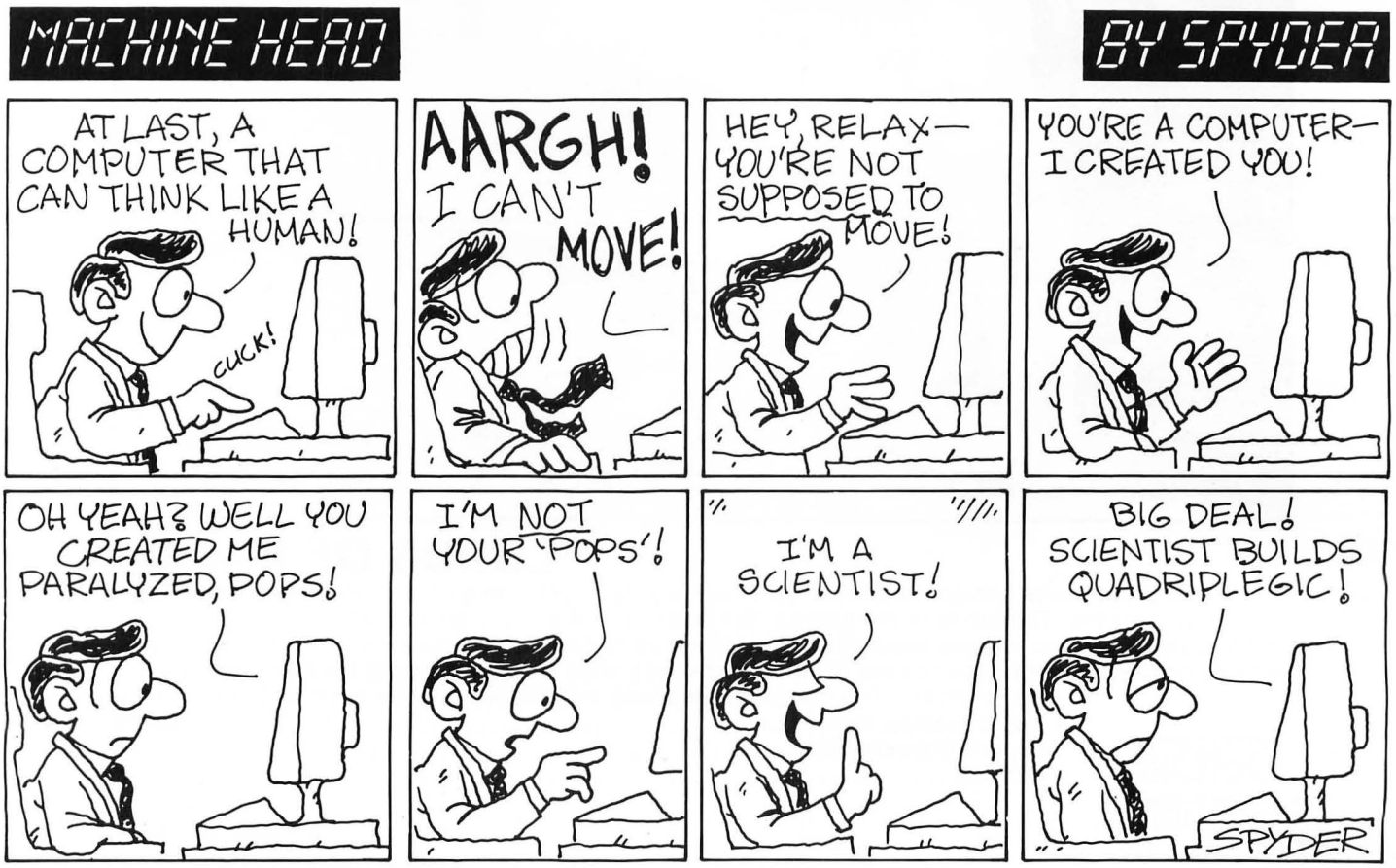
You are, of course, an intrepid adventurer off to pillage Davy Jones' Locker. You have a wealth of tools to aid in your quest, if only you can figure out which ones will truly be beneficial... There is potential bartering in the form of a pawnshop, and a definite need to examine everything you encounter. Magic works here both in the form of teleportation and of subsurface respiration, all you need to do is figure out how to get it rolling.

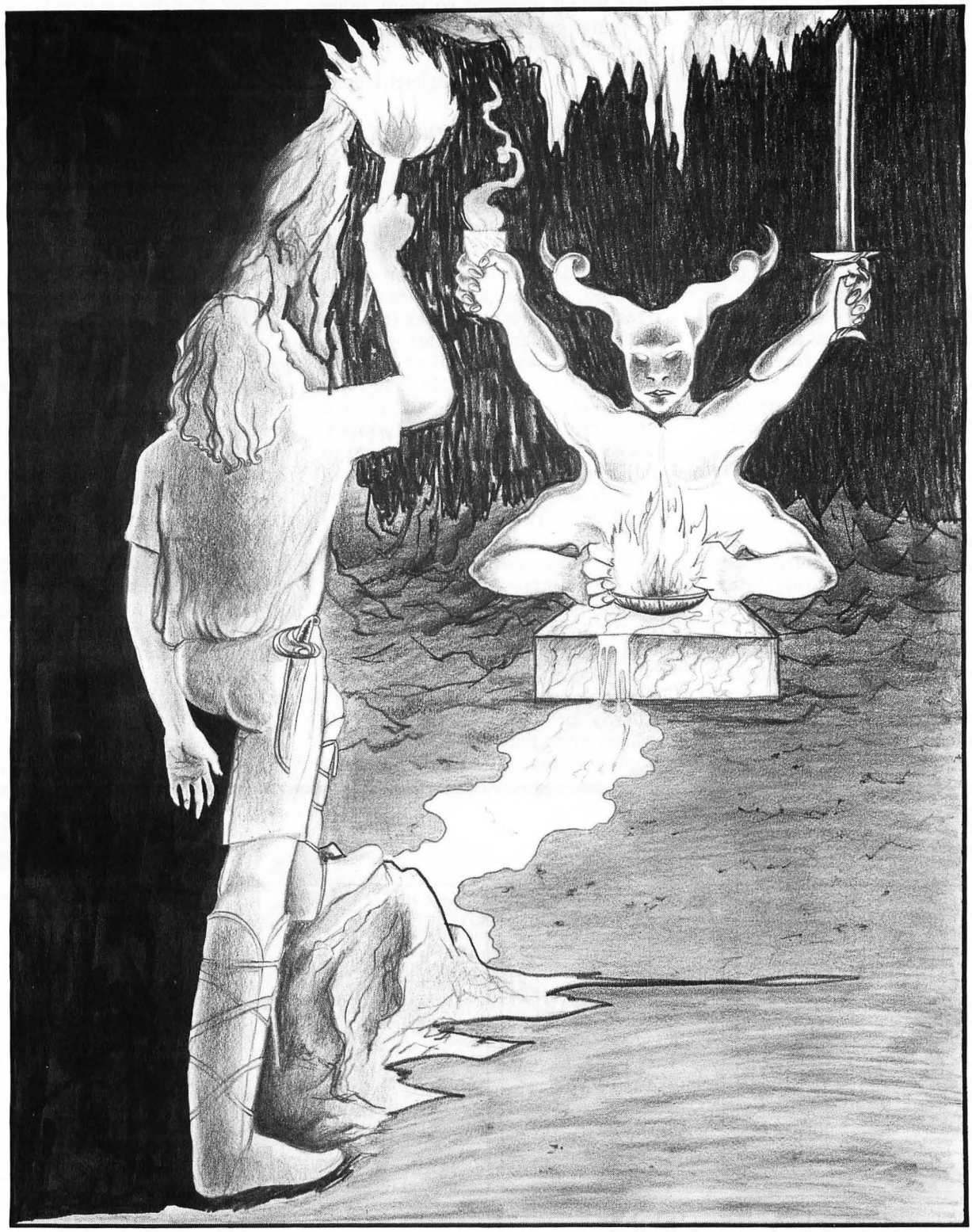
What makes this Adventure particularly interesting are the graphics. Bear in mind that this is an S-80 program and that as such it has only block graphics, none of that Hi-Res stuff. Yet the graphics are just as intriguing, perhaps more so in light of the machine's limitations, than the highly touted On-Line Systems graphics Adventures. There is no animation, of course, but each picture is well-conceived and designed. My own favorites are the scenes from the underwater city, but to each his own.

Another nice feature of this Adventure is the voice. Often one finds Adventures to be too straightforward in the messages they give to the reader. One of the things about Scott Adams' Adventures is that often the messages received are quite tongue in cheek. This holds true for "Atlantian Odyssey" and provokes a chuckle or two during play. If you don't believe me, try swearing at it when you get frustrated.

Johnson and Li have done a nice job on "Atlantian Odyssey". The program executes quickly and smoothly with virtually no noticeable flaws. The graphics are good and the theme, while not the most original, is nevertheless entertaining and interesting. Occasionally the computer will ignore a command, so if it doesn't work the first time, do try again. And don't go swimming without the medallion!

Dave Albert





LORDS OF KARMA

The player finds himself in a mythical, magical city to perform as many deeds of kindness and bravery as possible. The fun is in the gradual deciphering of the many secrets while exploring verdant forests, twisting trails, rugged mountains, and labyrinthine caverns. . . and avoiding lurking monsters. Almost no rules to learn; you tell the computer what you want by typing simple English sentences on the keyboard. The computer responds with many suprises in store. Solitaire only.

S-80 Cassette, 48K $20.00
Apple II Cassette, 32K $20.00
PET 32K $20.00

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

AVALON HILL

Ten reasons why your floppy disk should be a BASF FlexyDisk®.

More than four decades of experience in magnetic media – BASF invented magnetic recording tape, the forerunner of today's wide range of magnetic media, back in 1934, and was the first independent manufacturer of IBM-compatible floppy disks.

Tough Tyvek sleeve – no paper dust, no static electricity.

Special self-cleaning jacket and liner help eliminate data errors and media wear and tear.

Packaging to suit your requirements – standard flip-top box, Kassette 10® storage case, or bulk pack.

Center hole diameter punched to more accurate standards than industry specifications, for top performance.

100% certification – every single disk is tested at thresholds 2-3 times higher than system requirements, to be 100% error-free.

Bi-axially oriented polyester substrate – for uniform and reliable performance year after year.

Cross-linked oxide coating – for low head wear and long trouble-free media life.

Total capability – one of two manufacturers in the world that makes both 8″ and 5.25″ models, has tape and disk experience, and manufactures floppy disk drives.

Double lubrication – lubricants both in the formula and on the disk surface, to minimize media wear due to head friction.

For the name of your nearest supplier, write BASF Systems, Crosby Drive, Bedford, MA 01730, or call 617-271-4030.

BASF

Floppy Disks Mag Cards Cassettes Computer Tapes Disk Packs Computer Peripherals

SoftSide Selections Order Form

☐ **The World of Beysycx** $6.95

ATARI

☐ **Masters Golf** Cassette $9.95
☐ **World Series** Cassette $9.95

S-80

☐ **APL** 16K Cassette (Mod. I only) $14.95
☐ **APL** 32K Disk (Mod. I only) $39.95
☐ **Column Calculator 4.1** 32K Disk $39.95
☐ **Enhanced BASIC** 16K Cassette (Mod. I) ~~$59.95~~ $48.00*
☐ **Enhanced BASIC** 16K Cassette (Mod. III) ~~$59.95~~ $48.00*
☐ **Enhanced BASIC** 32K Disk (Mod. I) ~~$59.95~~ $48.00*
☐ **Enhanced BASIC** 32K Disk (Mod. III) ~~$59.95~~ $48.00*
☐ **Hellfire Warrior** 32K Disk ~~$39.95~~ $32.00*
☐ **Kriegspiel II** 16K Cassette $14.95
☐ **Lords of Karma** 48K Cassette $20.00
☐ **Masters Golf** Cassette $9.95
☐ **Pascal** 32K Disk (Mod. I only) $99.95
☐ **Sargon II** 16K Cassette (Mod. I only) ~~$29.95~~ $23.95*
☐ **Sargon II** 32K Disk (Mod. I only) ~~$34.95~~ $28.95*
☐ **Star Trek III.5** 16K Cassette $14.95
☐ **Star Trek III.5** 32K Disk $14.95
☐ **Temple of Apshai** 32K Disk ~~$39.95~~ $32.00*
☐ **Typing Tutor** 16K Cassette $15.95
☐ **Up Periscope** 16K Cassette $14.95
☐ **Warpath** 16K Cassette $14.95
☐ **World Series** Cassette $9.95
☐ **X-Wing II** 16K Cassette $9.95

APPLE

☐ **Hayden Applesoft Compiler** ~~$200.00~~ $175.00*
☐ **Hellfire Warrior** 48K Disk ~~$39.95~~ $32.00*
☐ **Lords of Karma** 32K Cassette $20.00
☐ **Sargon II** 24K Cassette ~~$29.95~~ $23.95*
☐ **Sargon II** 48K Disk ~~$34.95~~ $28.95*
☐ **Temple of Apshai** 48K Disk ~~$39.95~~ $32.00*
☐ **Time Lord** 48K Disk $29.95
☐ **World Series** Cassette $9.95
☐ **World Series** Disk $14.95

SoftSide Back Issues

☐ **SoftSide:S-80** (Magazine only)
month/year _____ $2.50
☐ **SoftSide:Apple** (Magazine only)
month/year _____ $2.50
☐ **PROG/80** (Magazine only)
month/year _____ $4.00
SUPER **SoftSide** (Magazine only)
☐ August or September 1980
(circle choice) $3.00
☐ October, 1980, to present (Magazine only)
month/year _____ $3.50
☐ **SoftSide** (Magazine with Cassette programs)
month/year _____ $9.95
☐ **SoftSide** (Magazine with Disk programs)
month/year _____ $14.95

***Reduced price until August 15, 1981 ONLY.**

ADD handling charges $1.50
(Foreign orders minimum $10.00 handling)
Additional charges
TOTAL

I enclose ☐ Check ☐ Money Order
☐ Master Charge/Visa#
CHARGE ACCOUNT NUMBER

Exp. Date _____ Inter. No. _____
Signature _____
Name _____
Address _____
City _____ State _____ Zip _____

Canada/Mexico Orders
No C.O.D. to Canada or Mexico. The prefered method of payment is by Master Card or Visa. NO PERSONAL OR COMPANY CHECKS. A bank check is acceptable if it has been preprinted for payment in U.S. dollars. The handling charge on all Canadian or Mexican orders is $5.00 PLUS actual shipping charges.

Other Foreign Orders
Payment must either be by a BANK CHECK drawn on a U.S. bank, payable in U.S. dollars or by affiliated bank credit cards of Visa or Master Card. All shipping and duty charges are the customer's responsibility. All overseas orders are subject to a $10.00 handling charge PLUS actual shipping charges.

Guarantee
All software is guaranteed to load and run. If you experience difficulties with the product within 30 days, the tape or disk may be returned. Call (603) 673-0585 or 673-0586 for a Return Authorization Number. Any returns without a Return Authorization Number clearly marked on the outside WILL BE REFUSED. Send your properly protected disk or tape to the attention of Customer Service Representative with a note including your name and address.

Liability
All software is sold on an as-is basis. **SoftSide** assumes no liability for loss or damage caused or alleged to be caused directly or indirectly by products sold or exchanged by them or their distributors, including, but not limited to, any interruption in service, loss of business or anticipatory profits or consequential damages resulting from use of operation of such software.

Prices:

Prices are subject to change without notice. We are not responsible for typographical errors.

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

SoftSide-DV:
The Magazine that Addresses Both of You...

SoftSide Selections
6 South Street Milford NH 03055
For Orders Only 603-673-0585

SoftSide, the magazine that for the past three years has brought you and your microcomputer the finest in BASIC language software, now offers more programs to disk users, and the software isn't limited to BASIC!

Over 2,000 subscribers have dispensed with the hours of typing by receiving our programs for the S-80, Apple, and Atari on disk and cassette along with the printed magazine, but convenience is only the beginning for SoftSide-DV. Now you'll receive all the programs in the magazine itself and a host of others, including Machine Language and hybrid (multiple language) programs that we could never print on paper. We will offer complete programs of every conceivable type and ongoing data bases. We will venture into totally modified languages, new versions of BASIC authored specifically for your microcomputer, I-$tring programs with new data bases every month that will allow you to enjoy puzzles, games and adventures without having to type them in and thus have them half-solved before you ever start!

If you don't believe we can deliver what we're promising, try us out on a one-time basis: a full disk of programs for $19.95. Of course if you subscribe, the cost will be substantially less: Until August 31st we will offer a full year's subscription to SoftSide-DV for a mere $99.95, that's $8.33 per disk, and you get the magazine to boot!

For your convenience, we offer an installment plan for Mastercard and Visa card holders: You pay $32.50 per quarter (there is a $5 charge for this service). For orders outside the U.S.A. please add $18. If you currently receive the magazine, we will credit the remainder of your subscription to a new SoftSide-DV order.

All subscribers to SoftSide-DV will receive at least one disk full of programs as well as SoftSide magazine, guaranteeing you not only the best software for your microcomputer, but the least expensive as well. It's our way of inviting you to step into the future.

Be sure to specify the type of computer your SoftSide-DV subscription is for.