SoftSide™

The Magazine For You & Your Computer

#43
THREE DOLLARS

FREE Software Printed In This Issue:
Relaaax. . .
Relieve Tension With Your Computer

Computers in Education
New Tools For Thought

Micros And Problem Solving
More Than A Drillmaster

Space Age Schools
Saul Bernstein On Global Village Education

Consumer Electronics Show
Tomorrow's Technology Has Arrived

Revolution On Campus
Students And Faculty Discover Computers

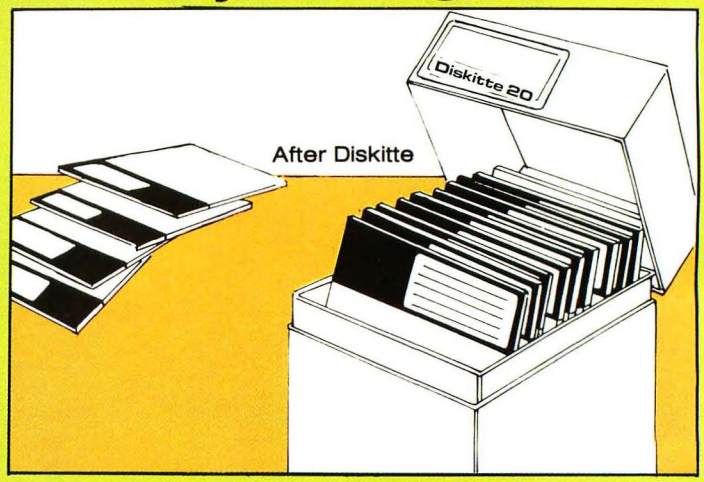
Software Reviews:
APPLE® — Learning Music Theory
ATARI® — ABC Compiler, Speed Reading
IBM® PC — Facts In Five
TRS-80® — Jabbertalky, Weerd

0 74470 12550 43

Is Your Software Library

OUT OF CONTROL?

Diskitte™ will get you together.



Diskitte™ 50

Includes All of the Software in Diskitte 20 **PLUS:**

- Personal Finance Manager Disk
  - ☆ to keep your money affairs in order
- 45 Blank Disks (50 disks in all)
  - ☆ certified for life
- Handsome Storage Case
- 50 Matching Sleeves and Labels
- 100 Index Cards
- Instruction Booklet

$149.95
IBM PC Version
$189.95

Available for: Apple® II Family, Atari®, IBM® PC, and TRS-80®.

Diskitte™ 10

Includes:

- 2 Disks of Software
  - ☆ Disk Managers
  - ☆ Games (see Diskitte 20 for descriptions)
- 8 Blank Disks (10 disks in all)
- Handsome Storage Case
- 10 Matching Sleeves and Labels
- 20 Index Cards
- Instruction Booklet

$39.95
IBM PC Version
$49.95

Diskitte™ 20

Includes:

- 4 Disks of Software
  - ☆ MicroText — a "family" word processor
  - ☆ SoftSide Database Manager — to keep your data at your command
  - ☆ Games — Hopper, Minigolf, and others to entertain you
- 16 Blank Disks (20 disks in all)
  - ☆ certified for life
- Handsome Storage Case
  - ☆ to protect your investment
- 20 Matching Sleeves and Labels
  - ☆ color coded for indexing
- 40 Index Cards
  - ☆ for hard copy of your disk directories
- Instruction Booklet
  - ☆ complete documentation for the software
  - ☆ tips on good disk care

$79.95
IBM PC Version
$99.95

All the Software, disks and organizers to get your computer affairs in order, at the cost of blank disks!

Diskitte
SoftSide Publications, Inc.
10 Northern Boulevard
Amherst, NH 03031

Computer Type:
☐ Atari
☐ IBM PC
☐ Apple II Family
☐ TRS-80 Mod I
☐ TRS-80 Mod III

I am enclosing $ _____
Form of Payment:
☐ Check/M.O. ☐ VISA ☐ MasterCard
Name of Cardholder _____
MC# and Interbank#/VISA# _____
Expiration Date _____
Signature _____

Order Diskitte today! Enclose the coupon in an envelope with your payment or call Toll Free 1-800-345-8122

☐ Diskitte 20 $79.95 (IBM $99.95)
☐ Diskitte 50 $149.95 (IBM $189.95)
☐ Diskitte 10 $39.95 (IBM $49.95)

Name _____
Address _____
City/State/Zip _____

(Please allow 8-10 weeks for delivery.)

Most computer owners neglect their software library until it is out of control. Diskitte offers you a system to put your software in order and provides you with additional valuable programs as well. It's the perfect gift for the new computer owner, too. Place several orders today — where else can you find such value at the price of the blank disks alone?

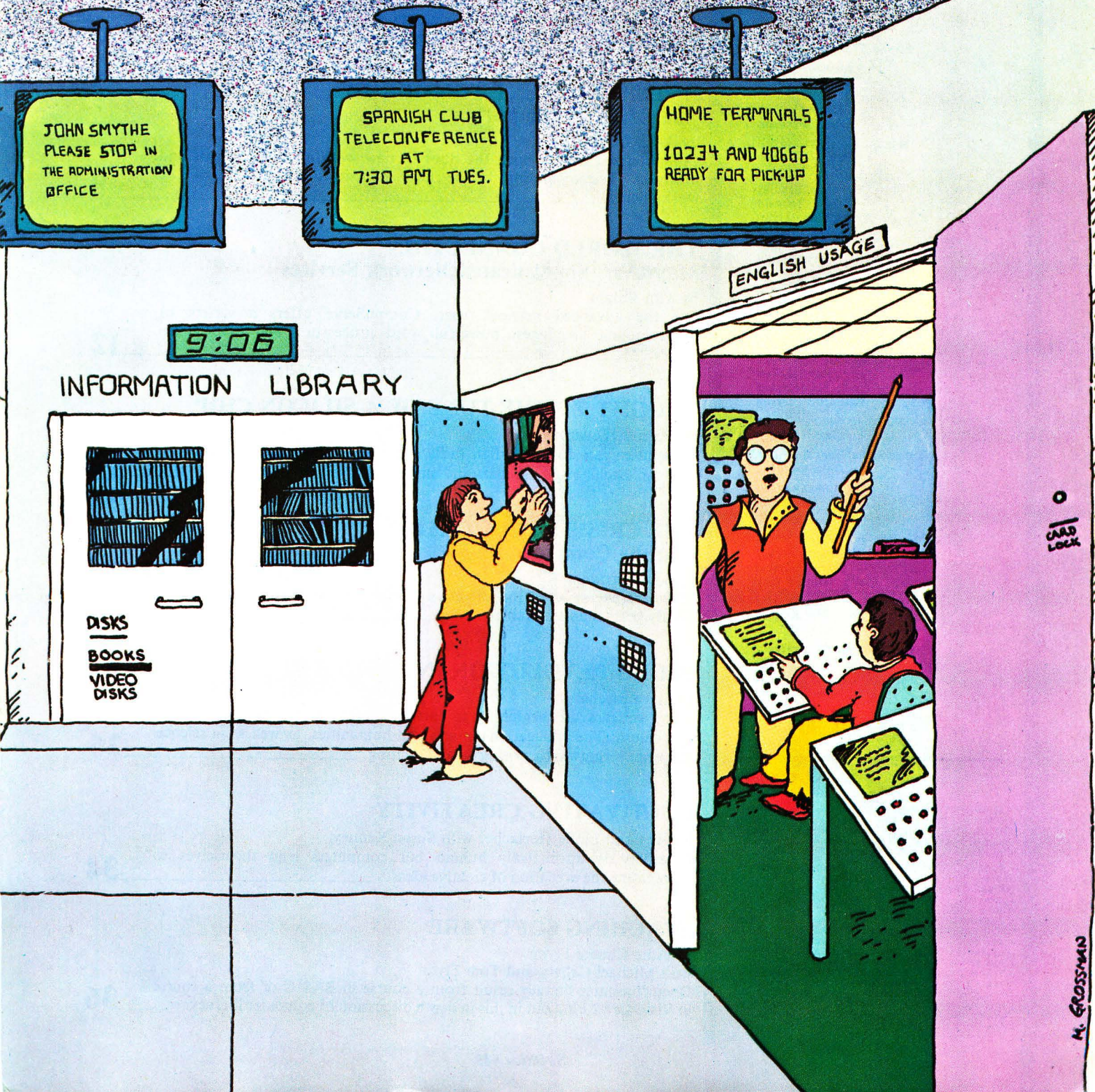
*Refill Kits Available

Diskitte is a product of SoftSide Publications, Inc.

SoftSide #43

Computers in Education

New Tools For Thought



SoftSide™

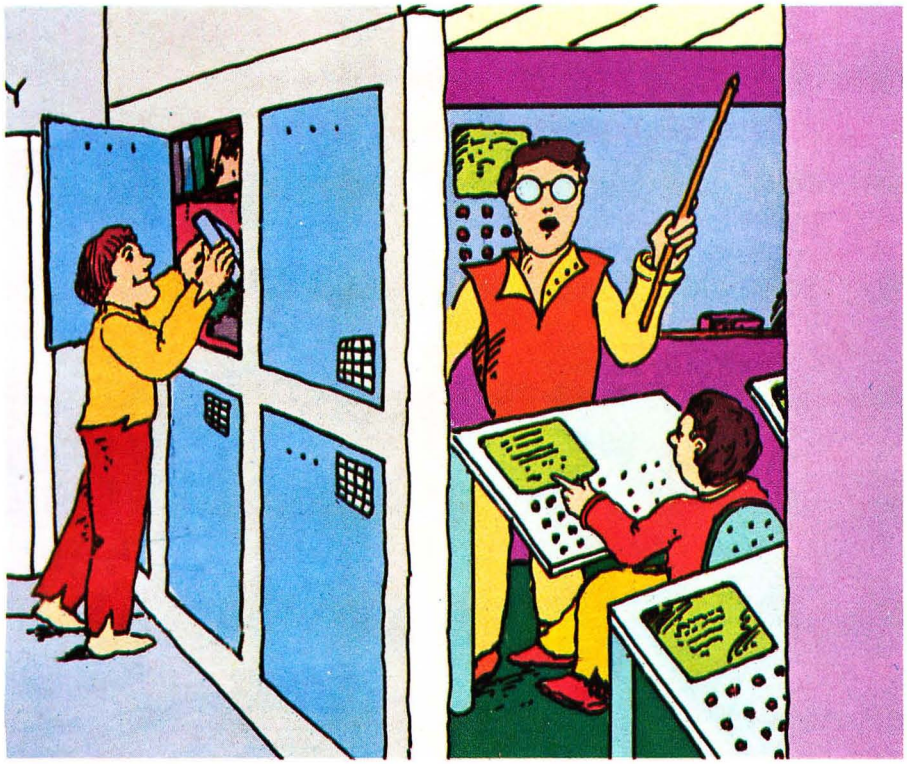
SoftSide Vol. 6, No. 10

SUBSCRIPTION INQUIRIES should be sent to *SoftSide* Publications, Inc., 100 Pine Street, Holmes, PA 19043. EDITORIAL AND ADVERTISING CORRESPONDENCE should be sent to *SoftSide* Publications, Inc., 10 Northern Blvd., Amherst, NH 03031. Telephone (603) 673-0585.

SoftSide (ISSN 0274-8630) is published monthly by *SoftSide* Publications, Inc., 10 Northern Blvd., Amherst, NH 03031. Printed at Volkmuth Printers, St. Cloud, MN. Second class postage paid at Milford, NH, and additional mailing offices. Subscription rates: US, APO/FPO and Canada, $30/12 issues. First Class US, First Class Canada, and Mexico, $40/12 issues. Other foreign countries, $62/12 issues. Media subscription rates: US and APO/FPO, — Magazine and Cassette, $99/12 issues. US and APO/FPO — Magazine and Disk $149/12 issues. Canada and Mexico, — add $20/12 issues. Other foreign countries — add $50/12 issues. All remittances must be in US funds. POSTMASTER: Please send form 3579 to *SoftSide* Publications, Inc., 100 Pine Street, Holmes, PA 19043.

FEATURES

Table of Contents



Cover Illustration: Myron Grossman's vision of the future school shows such fascinating possibilities as video bulletin boards, teaching languages by teleconferencing, and word processors built into desk tops. But notice the human teacher, the smiles, and the activity in the classroom. Human students need human teachers, no matter what technology we employ. Computers take care of all the drudgery, and leave people free to ask — and answer the right questions.

Are you STILL Typing?

If you type in your programs from ***SoftSide*** every month, you're spending a lot of time at the keyboard before you ever get to see the software we produce. By the time you've typed in one program, our DV and CV subscribers have played several games, integrated the ***SoftSide*** utilities into their software library, and probably had the time to read the articles and reviews in their issue of ***SoftSide*** Magazine.

Wouldn't you rather spend your valuable time **using** your software instead of **typing** it? Then there's no time like the present to subscribe or convert to ***SoftSide DV*** or ***CV!***

SoftSide CV

SoftSide's Cassette Version (CV) offers you an inexpensive way to enjoy your programs without hours of typing or hunting for errors. PLUS you'll receive a BONUS program every month, usually an adventure. All programs are tested and ready to RUN!

CV gives you the programs offered for your system each month in ***SoftSide*** plus the BONUS program on a tape, plus a copy of ***SoftSide*** Magazine — 12 tapes and 12 magazines for only $99.

SoftSide DV

SoftSide's Disk Version (DV) offers you not only the programs listed in each month's ***SoftSide,*** but **two** BONUS programs as well. Only the documentation for the bonus programs will appear in ***SoftSide*** magazine, **not** the code. The bonus programs will be of every conceivable type — multiple and Machine Language programs, modified languages, ongoing modular programs, adventures, and software so extensive, it would take an entire issue of ***SoftSide*** just to print the code. You'll receive 12 disks and 12 magazines — all for only $149.

Feel Like You're Missing Something? You Are!

Don't wait. Order your subscription to ***SoftSide DV*** or ***CV*** today. Use the convenient order card located elsewhere in this issue. For the convenience of MasterCard or VISA holders, we offer an installment payment plan for ***DV*** subscriptions: Pay just $38.50 per month for four months (a total of $154, which includes a $5 billing charge). If you already subscribe to ***SoftSide*** magazine, you can convert the remainder of your subscription for only $9.92 per disk or $5.75 per tape. Just check the convert box on the order card and we'll bill you for your conversion cost. For orders outside the USA, see the "SoftSide Ordering Information" located elsewhere in this issue.

Computerists are offered the rare opportunity of marching into a new frontier. Advance to the front of the parade by subscribing to *SoftSide DV* or *CV,* the magazines of the future — available today.

Editorial

You Learn Something New Every Day

Digital portrait by Tom Flynn

My father has taught secondary science and math courses for many years. His courses are filled with facts, figures and laws. I once asked him what he thought was the most important thing he taught his students. He told me that he felt his primary responsibility was to teach them *how to learn* because the facts and figures he could teach them were subject to continual change. If he didn't inspire his students to search out the new facts and figures, then his influence on their lives would be of little value.

Education is a non-stop process. I learned several new things today and plan to do the same every day until I die. In actuality, it's very hard to survive in this world without learning several new things every day. Life constantly presents us with new information to assimilate. This continuous learning process is the essence of education. Ergo, to survive in this world, your education must be non-stop. Does this mean you have to go to school all your life? Of course not, but it brings to focus an important point — When we speak of computers in education, we must be sure to address how computers apply to our life-long continuing education as well as our formal schooling.

Much of the press attention given to computers in education has centered on elementary level applications. This is not surprising, as the infant software industry can apply its new skills most easily to the basics of elementary education. However, the computer is an equally valid tool at all levels of learning. We must not discount its usefulness due to the current lack of appropriate software for higher level application. If my father's principle is to hold true, it is at the highest level of education (after the formal teaching stops and the living begins) that the computer has its greatest validity as a learning tool. The responsibility remains with the formal educational system to teach future generations how to use this tool, but we must address how those of us who don't plan to return to formal schooling can utilize this learning tool without such guidance.

The process of learning how to use this tool is in itself, continuing education. My mail recently has been filled with offers ranging from mail order courses on how to use my IBM® PC to courses on VisiCalc® and word processing at the local community college. Night classes in BASIC, computer programming and computer usage are common at even the smallest high schools. Companies are investing millions of dollars in retraining programs to familiarize their staff with computers. We may be seeing the start of the most far-reaching education program our society has ever attempted. There is, of course, the question of quality in these programs. However, the computer is such a powerful tool that even the least of courses can make students aware of the avenues the computer opens to them.

When we speak of computers in education, we must be sure to address how computers apply to our life-long continuing education as well as our formal schooling.

Once those avenues are opened to a larger percentage of our society, we will see the massive effect computers can have on continuing education in every field of study. The world of telecommunications will make investigation and research a natural part of our habitude. It will be as easy to explore and exploit the information in the great libraries of the world as it currently is to consult the books we keep in our homes. Videodiscs and other interactive teaching systems related to the computer will allow us to explore practically any topic we desire — step-by-step and in the privacy of our own homes. I'm currenty refreshing my knowledge of 35mm photography with an interactive videodisc. I can skip over the parts I know very well, listen to the advanced soundtrack with the parts I know fairly well and listen to the novice soundtrack for parts where I'm still a beginner. It's a custom tailored course for my learning needs. Add a microcomputer to my videodisc player and the course could quiz me, keep track of my progress and let me know what areas I need to review. This will not negate the need for "live" teachers. Some subject matter won't lend itself to the computer. However, the teaching profession will change considerably. Imagine a teacher in a highly specialized field offering a course on one of the large database networks. He could make the material available for a fee and then be accessible on-line or by electronic mail to answer questions and coach anyone having trouble. We may have legions of free-lance teachers working on the networks.

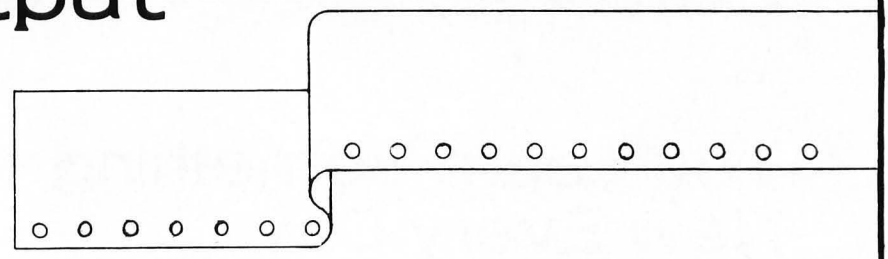
Education is entering a new world. At the recent Harvard conference on video games in education, a teacher of long standing was asked if she was upset by the changes computers were bringing to education. Her reply was that she was upset only because she wouldn't be teaching in twenty years to see what happened.

Harriette and Cecil Kottwitz, my mother and father, retired this year after long and successful careers teaching in the public schools of Nebraska. In recent years, they learned about computers and introduced them into their courses. I am dedicating this Education Issue of ***SoftSide*** to them and to their fellow teachers still struggling to bring this revolutionary technology to our education system during these tumultuous times. Their skill in teaching future generations how to learn will help our society to reach its zenith.

Randal L. Kottwitz

Randal L. Kottwitz
Publisher/Editor-in-Chief

Input/Output



Hammers and Nails

Dear *SoftSide,*

I really appreciated your editorial in #41 ("The Hammer Didn't Hit Your Thumb, You Did"). It's one of the clearest statements on the subject that I've seen, and I almost agree with it. The title is wonderful, and your discussion of how we attribute responsibility to the computer is exactly on target. Cheers for your statement that "We must be willing to take ... responsibility for the computer's effect on our culture." However, your comments on implied use of a tool, I believe, are beside the point. Computer, camera and gun are all general purpose tools in a sense, but they tend towards certain classes of usage. When speech recognition required huge and expensive computers it could still be used for voice input or for automated wiretapping, but the financial realities made the second much more likely. Given how we obtain food today, the handgun tends to be a tool of lawbreaking or supposed policing. It is helpful to assume some generalizations, even though they will be false in certain situations (as your examples clearly demonstrate). We need such generalization to formulate general policy. I believe that real harm can come to children through video games. If I am correct, it should be our responsibility — not just the child's — to curtail video game play.

Finally, I don't think Weizenbaum is asking us to keep computers out of schools until we understand their impact. Rather, I think he is asking that they be put into schools carefully, and with care for the impacts that are already predictable. I agree, but would not be as strident or negative as Joe has become. With all those criticisms, I still feel that your editorial is close to the mark, and I'm delighted to see these issues raised on the editorial page of *SoftSide.*

Arthur Fink
Wilton, NH

Speaking Easy

Dear *SoftSide,*

The article "Speaking Easy" by Peter Favaro was very enjoyable. However, there is one point he failed to make about S.A.M. He states it has one major disappointment. I feel there are two. The second is that no one else can run your programs, unless they have S.A.M.

I am not knocking the Don't Ask people, but I write programs for others to enjoy or use (not for sale). Their ads state, "Add speech to your program," however, they fail to tell the entire story. Your readers should be made well aware of this before they put down that hard earned money.

I was excited about using S.A.M. in my graduate project, an education program on drafting, only to find out the money spent went down the drain. Fifty dollars — zapped! I could have bought more disks with that money.

I would appreciate it if your readers are made aware of the second problem.

Jim Watson
Corpus Christi, TX

Bar Code News

Dear *SoftSide,*

The article titled "Entertainment Tomorrow: The Ubiquitous Bar Code" by Allen J. Wold in *SoftSide* #41 provided a good introduction to the world of bar code data entry. It's a technology that I believe has a lot to offer the personal computer user. Within the past six months, several companies introduced bar code inventory systems (hardware and software) for personal computers.

As Allen mentioned, Carl Helmers and Walter Banks were the creators of the PAPERBYTE system of program listing printer in bar code. Both Carl Helmers and Walter Banks were the founders of *Bar Code News,* the journal of bar code system applications. We are committed to bringing our readers the latest information about bar code applications. We invite anyone interested in bar code technology to apply for a free subscription to Bar Code News by sending their name and address to:

Bar Code News
North American
Technology
174 Concord St.
Peterbourgh, NH 03458

Russ Adams, Editor
Peterborough, NH

SWAT from the Past

Dear *SoftSide:*

I've been a subscriber for several years and plan to continue for many more. Recently though, something has come to my attention which bugs me. Visiting my local computer store I chanced to thumb through *Best Of SoftSide* — Apple edition. I noticed SWAT tables for various games including Quest, Leyte, etc. Now I've never been able to get these games working. Even with a printed listing. The presence of SWAT listings in the book and not printed as a convenience for long term subscribers in your magazine is a disappointment. Why should I have to blow $20 for a book when I have every issue? How about running a special issue with new SWAT listings for us long-timers?

Rich Ferri
Jamestown, NY

Editor's reply: All the programs in *The Best of SoftSide* have been modified and debugged since we published them in *SoftSide.* We have no way to know each subscriber's customized version with their specific debugging and enhancing, so there is no way to provide individualized SWAT tables for each version. One altered letter in a listing is enough to throw off SWAT. We wish we could be of more help but the nature of the beast prevents us.

CB Memories

Dear *SoftSide,*

I would like to thank you for publishing the article on Compuserve's CB simulation. It certainly brought back many familiar and fond memories of a time when I was new to the home computer world. At that time I had already purchased my second home computer (Radio Shack's Color Computer), and, since then, have changed over to an Atari 800. Unfortunately, now I do not have the capability of telecommunications with my computer. That was perhaps my biggest mistake, because I have lost contact with many of the people I met on CB.

The sample conversation included in the article had many handles unfamiliar to me, since I have not been on CB for quite some

SOFTSIDE PROGRAMMERS WIN COMPETITION

by Carolyn Nolan

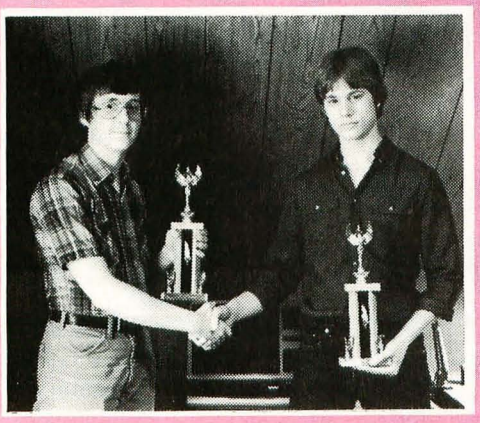
SoftSide is celebrating. Rich Bouchard and Peter Johnson, two members of the ***SoftSide*** programming staff, are also members of the MASH (Milford Area Senior High School) computer science team. They placed second in the United States All-Star Computer competiton held in May on Long Island.

You will all recognize the name Rich Bouchard. He started programming for ***SoftSide four*** years ago. When he was thirteen his father bought a TRS-80® Model I for the family and Rich, who has always been a mathematical whiz kid, began to explore its possibilities. He has been programming ever since.

When Rich got to high school he knew he would concentrate in computer science. He joined a class of twelve students on the 67th DEC PDP-11 ever made, and began mastering the "big" systems. A fellow student, Norman Lastovica, was an employee of ***SoftSide*** at the time, and he suggested Rich fill out an application. He did. His first task at ***SoftSide*** was typing in the names of every Apple® dealer in America." When he finished the job, we not only had a database on Apple dealers; we also had a brand new mail list program which Rich had developed. Anyone who has worked with Rich will tell you he is quite fastidious about his work, and he is *fast.*

Peter Johnson is a junior at MASH this year. He came to work for ***SoftSide*** this spring to develop custom accounting and sales programs — no small task. The TRS-80 DV in issue 40, "IO Miner," is a product of his programming expertise. The ***SoftSide*** tradition of culling talent from the local high school continues.

Peter is versed in machine language as well as BASIC. He took his first computer course in an after-school program when he was in the sixth grade. Then he talked his father, a local architect, into buying an "office computer." Mr. Johnson bought a TRS-80 Model I and Peter bought many of the peripherals himself. The computer now resides in his room. Peter plans to continue his education in computer science and to acquire additional practical experience at ***SoftSide.*** He also hopes to buy his ***second*** computer — probably a Commodore 64 — and develop graphics applications for the magazine. Peter certainly is in the right place at the right time because ***SoftSide*** will run its first Commodore 64® program in issue #45.



The competition these young men participated in was the culmination of a year's worth of work and preparation. During that time the computer team completed a series of five programming projects, including a two-day practical problem which the school sent to the American Computer Science League for judging. Based on that segment of the competition, the League selected the MASH team to compete in the final phase of the "games" on Long Island on May 14, 1983. They won second place in North America.

Rich will be leaving ***SoftSide*** in August to continue his studies at Worcester Polytechnic Institute, but he will continue to develop programs for us. Rich and Peter seem well on their way to accomplishing one of the primary purposes of education: becoming self-supporting and contributing members of society. They certainly are making a contribution to ***SoftSide.*** We congratulate them on work well done. And we thank them.

time. Aunt Nettie was fairly new at that time. My handle on CB was "West Aragorn," dedicated to the famed Tolkien character. The "West" was to distinguish me from another Aragorn who was on the east coast. I eventually changed that handle for a handle some of the old timers on CB may remember — "Cal Dreamer," short for California Dreamer. It was a handle suggested by a good friend of mine whom I met on CB, "Bright Eyes."

My youngest sister (her handle was "Red Rose") met her husband on CB. They now have a daughter, and both she and her husband live near San Francisco.

I had the chance to go to several CB parties up and down the coast, and I met several of the people who were on CB at the time, people I would never have met otherwise. I regret not having kept in touch with them. I would appreciate any of the old timers who remember me (Cal Dreamer) dropping a line saying hello. I am at present saving up for another modem, and telecommunications system for my Atari. Once I have done that, I definitely plan to resume my subscription with Compuserve.

So, ***SoftSide,*** as the song says, "Thanks for the memories." Appreciation is an understatement.

By the way, CB has 40 channels, so there is enough room for everyone.

John L. Urban
(Cal Dreamer)
1117 25th St., Apt. 206
San Diego, CA 92154

Roses & Thorns

Dear ***SoftSide,***

Just when I questioned the value of your magazine to me, I received this month's DV for my Atari with an Assembly Language game in it! Upon booting the disk, I was delighted to see a game that is a lot of fun to play, with a clever idea behind it. I find it very challenging and justifying the cost of your magazine. I only hope that it is a sign of things to come. Let's face it, most games in BASIC are somewhat dull. Thanks again and let's see more!

Bob Albright
York, PA

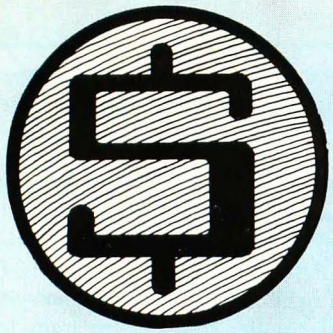
Dear ***SoftSide,***

I would like to start off with a general complaint about the new format of your magazine. I would like to know why you have decided to go with the individual program inserts for each type of computer. I find this extremely annoying. If I wanted a magazine with only Apple programs, I would have subscribed to an Apple only magazine. I also find it difficult to understand how you can offer the "Translation of the Month Contest" and only provide program listings for one computer. If you cannot examine the original program in its original language, maybe even type it in and run it on another machine, then how can you expect people to develop a translation?

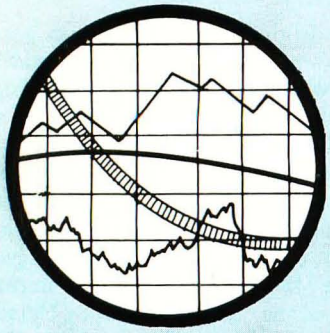
Aside from that aspect, keep up the good work. The articles and reviews are excellent and very helpful.

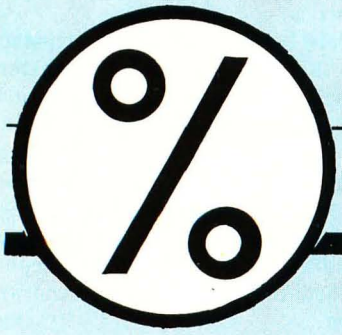
Brian K. Chinn
Seattle, WA

Editor's Reply: The inserts for the Apple®, Atari®, TRS-80® and IBM® PC systems are available at $2.95 each from our offices. Although the "translation of the month" contest is still in effect, we are not advertising it every month because we realize the effect the new format has had. We encourage Mr. Chinn to submit translations and will help him to whatever extent we can.



CALC SIDE





by David Peters

The *VisiCalc*® Spreadsheet Comes Home

Last time, I demonstrated a method of overlaying different variables into a model, to study the resulting changes. To do this I used separate models, overlaying them one at a time on a kind of "master model," each time taking the basic data that came in and multiplying it, dividing it and so on. In this issue you will see how to develop a system to review the effect of variable information on a model in a quicker and more effective way.

One of VisiCalc's most useful capabilities is answering the question "What if?" Because recalculation is fast, and you can chain calculations from one location to another by calling on earlier locations, you can look at all kinds of alternatives. Comparing these possibilities allows you to choose the best solution.

To provide an example situation, imagine that you are going to build some book shelves. You have several plans, which include a single, wide, freestanding unit, a double sized but otherwise similar unit, a built-in system in the corner of a room, and a simple hang-on-the-wall shelving system using standard hardware. You have done your measuring and know how much material each type needs. Figure 1 shows a simplified list: shelving lumber, wooden uprights, lighting units, supporting brackets and paint. The MISC category covers nails, screws and things (including medical supplies if you are a butterfingers).

Building the Model

After consulting a local lumberyard's catalog, plug in the cost of each of the items in each COST column (after figuring the total price for each). This gives you the total cost for each shelving system. Since the purpose of the shelves is to hold books, a calculation at the bottom shows a cost per book for each. Figuring an average of

Figure 1

	SINGLE UNIT		DOUBLE UNIT		BUILT-IN UNIT		ON THE WALL UNIT	
	AMOUNT	COST	AMOUNT	COST	AMOUNT	COST	AMOUNT	COST
MATERIALS								
SHELF FT	48	42.72	96	85.44	128	113.92	96	85.44
SUPPT FT	16	16.80	24	25.20	24	25.20	0	0.00
LIGHTS	3	68.55	6	137.10	8	182.80	4	91.40
BRACKETS	10	35.00	18	63.00	10	35.00	18	63.00
PAINT QT	1	6.89	2	13.78	4	27.56	1	6.89
MISC %	.05	8.50	.05	16.23	.05	19.22	.05	12.34
TOTAL		178.46		340.75		403.70		259.07
$/BOOK		0.31		0.30		0.26		0.22

Figure 2

MATERIALS	MATERIALS COST	SINGLE UNIT AMOUNT	SINGLE UNIT COST	DOUBLE UNIT AMOUNT	DOUBLE UNIT COST	BUILT-IN UNIT AMOUNT	BUILT-IN UNIT COST	ON THE WALL UNIT AMOUNT	ON THE WALL UNIT COST
SHELF FT	0.89	48	42.72	96	85.44	128	113.92	96	85.44
SUPPT FT	1.05	16	16.80	24	25.20	24	25.20	0	0.00
LIGHTS	22.85	3	68.55	6	137.10	8	182.80	4	91.40
BRACKETS	3.50	10	35.00	18	63.00	10	35.00	18	63.00
PAINT QT	6.89	1	6.89	2	13.78	4	27.56	1	6.89
MISC %	.05		8.50		16.23	.05	19.22		12.34
TOTAL			178.46		340.75		403.70		259.07
$/BOOK			0.31		0.30		0.26		0.22

twelve books per foot, divide the SHELF FT by twelve, and then divide the total dollars for each by that number.

In this matrix you can already do some "what-iffing." By changing the quantities in the materials list (if you get a good idea for changing the design) or by changing a price, you can see the implications on the total cost. But now you get smart. You realize that changing the price of an item if you get a good buy is difficult. If cheaper lights are available, you will have to change the contents of each location under the cost columns where you multiply the number of lights by the cost.

Variations on a Shelf

Modify the model, as in Figure 2, bringing the variables out to the left in a column of their own called MATERIALS COST. Change the formulae under each cost column to multiply this new variable by the material required. That is the first simple lesson for this issue. If you intend to use a model for any kind of "what-iffing," make sure that every variable used more than once is brought out to a single point, so that you can access it easily.

If you must change many formulae by editing and re-replicating, and wish to avoid all these keystrokes, use the concept of External Variables — locations that contain only the variables, so that one modification changes them everywhere. Using this method you will be able to switch back and forth between sets of values for the variables, and compare the results quickly and easily.

To go a step further — you want to do a little shopping around for good prices on materials and since several different suppliers are in the area you go on a shopping trip to check their offerings. Rarely is one place cheaper for everything, so change the model again, as in Figure 3. In the top section, you have a column for each supplier, and you enter each list. Now comes a new (and very useful) principle: The Multiple Choice Variable Selector (Ellen Martin, a member of the InterCalc Spreadsheet Users Group, sent this in and it was originally published in their monthly newsletter).

Choice Variables

Note that the figures beneath LUMBERYARD are repeated in the rightmost column, CHOICE. What happens is simple, but effective. The CHOICE column has (surprise!) an @CHOOSE formula. The VisiCalc function @CHOOSE works similarly to the ON GOTO in some versions of BASIC. Given an index value, VisiCalc selects by counting down a list of values. The format of an @CHOOSE command is

@CHOOSE (value, list) or
@CHOOSE (A1,100,200,300,400,
500,600,700,800,900)

Given a value of, for instance, 3 in A1, this formula will count down the list three places and return, of course, 300.

What you have done in the model is provide a location (beneath the word CHOICE) to insert the number of the Variable Group you want to use in the model. The formula which carries out the CHOICE is simply: @CHOOSE (CHOICE, SUPPLIER 1 VALUE LOCATION, SUPPLIER 2 VALUE LOCATION, SUPPLIER 3 VALUE LOCATION...) and so on. This formula is adjusted during replication to use the same supplier number but carry over the relative row value.

To examine a particular supplier, enter his number in the location beneath CHOICE (#2, the LUMBERYARD, is illustrated in Figure 3). VisiCalc brings over the figures from that column and calculates the resultant cost. The model beneath always calls on this CHOICE column, thus every re-calc allows you to review a different set of figures.

If you decide to go for economy, and pick up the cheapest items from each supplier, the last column COMBO uses @MIN. This function looks along each row and pulls over the cheapest price it finds. You can plug in those figures by using a six in CHOICE, and when you are ready to buy, just glance along and see where you have to go to get each item.

Automatic Comparisons

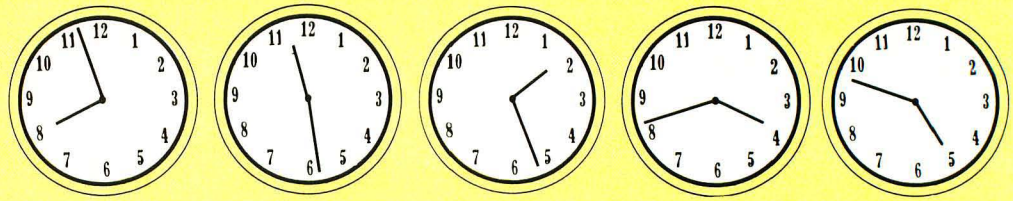
The Multiple Choice Variable Selector method is good if you want to examine lots of different options. If you set the recalculation to automatic, just making one keystroke changes the whole model in a flash. One small point — your order of recalculation should be (R)OWS — otherwise VisiCalc has not moved the group to the pickup point in CHOICE when it is doing the earlier columns and will need a second re-calc. Good luck and good calc-ing!

Figure 3

	SMITHS	LUMBER YARD	HARDWARE STORE	SAWMILL PRODUCTS	PLOTKINS	CHEAPEST COMBO	CHOICE 2<-PICK ONE
	1	2	3	4	5	6	
$ SHELF FT	0.89	0.94	1.05	0.66	0.88	0.66	0.94
$ SUPP FT	1.05	1.00	1.21	1.01	0.88	0.88	1.00
LIGHTS EA	22.85	28.95	31.66	31.66	25.00	22.85	28.95
BRACKET EA	3.50	2.99	1.99	1.99	2.44	1.99	2.99
PAINT QT	6.89	4.44	3.45	3.45	4.10	3.45	4.44
MISC %	.05	.05	.05	.05	.05	.05	0.05

MATERIALS	SINGLE UNIT AMOUNT	SINGLE UNIT COST	DOUBLE UNIT AMOUNT	DOUBLE UNIT COST	BUILT-IN UNIT AMOUNT	BUILT-IN UNIT COST	ON THE WALL UNIT AMOUNT	ON THE WALL UNIT COST
SHELF FT	48	45.12	96	90.24	128	120.32	96	90.24
SUPPT FT	16	16.00	24	24.00	24	24.00	0	0.00
LIGHTS	3	86.85	6	173.70	8	231.60	4	115.80
BRACKETS	10	29.90	18	53.82	10	29.90	18	53.82
PAINT QT	1	4.44	2	8.88	4	17.76	1	4.44
MISC %		9.12		17.53		21.18		13.22
TOTAL		191.43		368.17		444.76		277.52
$/BOOK		0.33		0.32		0.29		0.24

Entertainment Tomorrow



by Allen L. Wold

THE OFFICE OF THE FUTURE

The computerized office will become a reality because *the profit motive will demand it.* Much has been published lately on the computerized office, and how it will increase efficiency, productivity and so on. Many articles discuss some of the problems the electronic office might cause, the resistance to it from certain quarters and implementation difficulties. A possible outcome is the "home office," or the decentralization of normal office functions. Why should a secretary or clerk spend time and money to travel to an office? Typing, data entry and file checking can be done at a home terminal.

The advent of computerized office work brings with it improved working conditions not specifically tied to your place of employment. Since your work place is strictly under your control, it can be precisely the kind of environment you like best. You will have much more control over just how you perform your job, as long as the job gets done correctly and on time, since no managers will be looking over your shoulder. Trivial tedious, and repetitious tasks will be automated, leaving you with work that is necessary and useful.

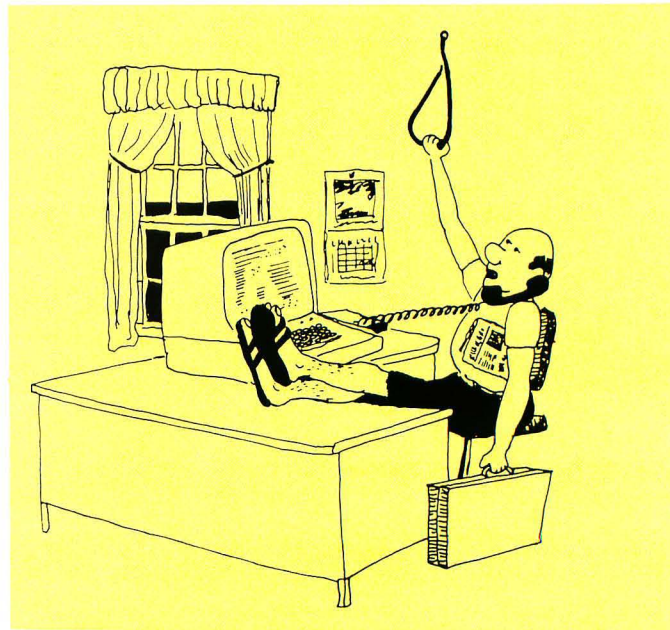
Most or all blue-collar jobs eventually could be performed by robots. The workers thus displaced will find better things to do, as have the farmers and millworkers of the past. Society as a whole must address the problems of displaced workers, and help them find those better things to do, during the transition period. Most jobs, in the sense of required work, can be done by a computer or robot. We are learning that machines can be effective even in such areas of human endeavor as medical care and psychological counseling. Problems, objections and resistance always arise, but reduced operating expenses and higher profits will overcome them.

No Job, No Identity

For many people, this is a frightening prospect. They see themselves as part of an idle society, with nothing to do, no significant task to give meaning to their lives. As I see it, we should not fear that the future will result in such an empty existence, for two major reasons. First, the transition will take place over considerable time (in today's terms), which will give us the chance to adjust. This will enable us to realize the second reason, which is that more meaningful tasks will replace most of the jobs we perform today. These activities will be more personally fulfilling and will occupy the expanding leisure time. The history of technology and civilization shows that new machines ultimately lead to more (and usually more interesting) work for the expanding human population. Except under unusual circumstances, society resists sudden change. The evolution to a more leisured society will come about in part because the computerized office eliminates the need to spend time commuting, and allows people to work whenever they wish.

The eight-hour workday is a myth. Most of us spend up to two hours each day travelling to and from our jobs. If you had a home office, you could spend these hours pursuing other things — a second job, artistic or creative activities, sleep, recreation or family interaction. Although one benefit to society is reduced reliance on automobiles and fossil fuels, the important factor is the time saved. No longer would you be isolated from your family for nine to twelve hours a day (unless you shut your personal office door). An interesting implication is that "commuting distance" becomes an electronic rather than a physical concept. You could live in the North Carolina woods while working for a chemical firm in New Jersey. Weather conditions or a broken down car will no longer prevent you from working.

continued on page 10



"The ONE thing I really miss is the subway ride into work everyday!"

Some Very Good Reasons to Buy an Echo Speech Synthesizer.

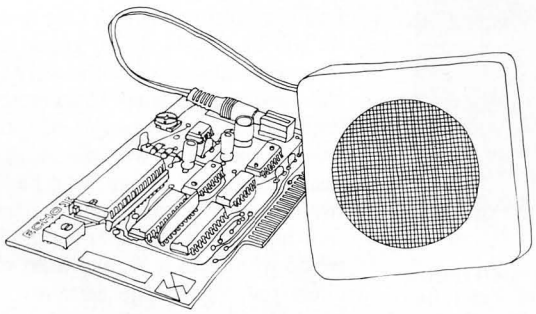
We're confident we have the most intelligible, versatile, and economical speech synthesizer on the market. Once you hear it, we're sure you'll be convinced too. All ECHOs are capable of speaking in four different voice modes which range from a robotic machine voice to natural female speech. (The fixed speech and custom modes are optional.)

It's Easy To Use

Unlike other speech systems, the ECHOs are very simple to use. It only takes a minute or two to get the ECHO talking. Any text which can be printed to the screen can be spoken. If you've written a BASIC program you can add speech with simple modifications.

Software Compatibility

Not only can you add speech to the programs you write yourself, over 25 top educational and adventure software manufacturers are currently designing programs to be compatible with the ECHO][. Be watching for details.



Value

Each ECHO comes ready to use with a speaker and tutorial-style manual. The ECHO][, priced at $149.95, also comes with a variety of demonstration and utility programs. The new ECHO GP (General Purpose), priced at $199.95 is a stand-alone unit with its own on-board microprocessor; it will interface with any computer through the serial port. All ECHOs have a two year warranty.

Bells And Whistles

If you want bells, whistles, music and sound effects, plus state-of-the-art sprite graphics capabilities, the ECHO technology is available on Synetix Systems' new Supersprite Board.

Check It Out

Most Apple dealers have the ECHO][available for demonstration as part of Street Electronics/ Orange Micro's unique Grappler+™, Bufferboard™, ECHO][™ Talking Demonstration.

Street Electronics Corporation
1140 Mark Avenue Carpinteria, CA 93013
Telephone (805) 684-4593

Work When You Feel Like It

Another logical consequence is flexible working time. Already many companies and the federal government use it, primarily to solve traffic problems. Employees may arrive within a half hour to two hour time slot, take off during the day for personal business, and leave an appropriate number of hours later. At home, you could work as quickly or as slowly as you wanted to, within limits. Faster workers could finish the assigned tasks sooner, giving them either more off-time, or the opportunity to do more work for more pay. Hence one person could limit himself to doing just enough to ensure a minimum living, while another could put in extra time, and profit from better working methods with a larger paycheck.

However, you will not be able to talk with fellow employees, to try out ideas conversationally before writing them down. As a writer who works at home alone all week long, I miss the opportunity to interact with other people. My wife has her office companions, even if they're not especially close friends, and that gives her the chance to socialize, even in the office environment. Often, valuable information passes from one person to another during these encounters — insights, points of view, problems and solutions which can make your own work easier. Teleconferencing at a level of intra-office interaction still will be possible and desirable, and I'll talk more about this in a later column.

This physical isolation is not a trivial consequence of the home office. Before my wife got her Ph.D. and a regular job, our life style was similar to what it might be like when the decentralized home office will be the rule rather than the exception. We spent most of our time together. We didn't interact intensively, but we were always there. She could talk about her studies, I about my writing, whenever we wished. We found a way to keep out of each other's hair while at the same time being available to each other. Part of the pleasure of that kind of free time was that we could take trips or do special projects whenever we wanted, instead of having to wait for weekends, evenings, holidays or vacations.

A less structured life was very pleasant. Structure existed, because I had my writing to do every day, she had her studies, but we could arrange our time to our satisfaction. Having the free time to be together was good. Some people might not adapt, but I think most people would like to see more of their families, watch their children grow and help their parents grow.

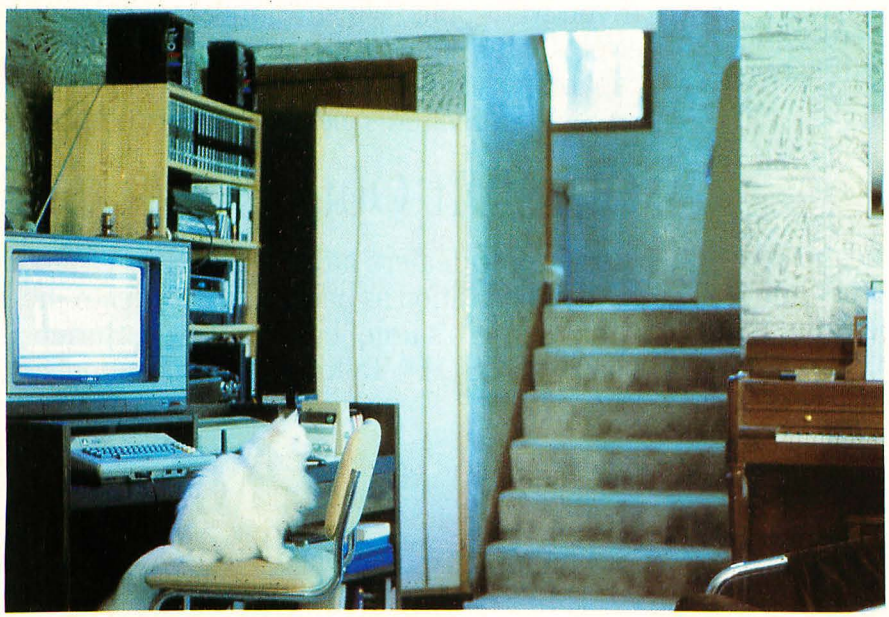
With more free time, more flexible time, and without the necessity to travel to and from work, other social interactions would improve. If everybody in my neighborhood worked as I did, we'd visit more often, and become better friends. The problem now is that we're too jealous of our spare time, and don't want to "waste" it on potentially unrewarding social activities. If any time is "spare" time, it can't be wasted, and interactions between neighbors become more desirable and possible.

Work as Play

Even "work" is recreation, if it provides something to do with your "spare time" and *provides personal satisfaction.* As jobs become automated the boring tasks disappear. Those that remain require human intervention of the most interesting — and rewarding kind. A number of people still will feel the need to work, to contribute to their society in some way. The awareness of this need makes many people dislike the idea of a work-free society.

Machines do not take away jobs. The manager who "hires" the machine instead of a person makes that decision. We are becoming more aware that quality of life is not totally dependent upon income, but also involves satisfaction with work. We can ensure that kind of satisfaction simply by deciding that for any job that exists, humans should have the first option of performing it. If a robot were developed to do brain surgery, a human surgeon would have the chance to do it instead. Street sweeping might not seem a desirable job, but if someone wants to do it and can handle the tools, he or she would be allowed to do it instead of letting a computer control the equipment. This kind of social consciousness implies two things: no one would have a job he or she doesn't like, and no one who wants to work would be without a satisfying job. Let the robot stand in the unemployment line. The whole concept of work and leisure will change — no distinction will exist.

Assuming that, sometime in the not too distant future, all this will come to pass, how will it affect rest, recreation, and family and social interaction? I do not think we will have a society of idle people, passing their time in trivial pursuits. As we find ourselves with more time and money, we will be able to devote ourselves to our other activities. Even at the early stages we will see ourselves adapt to increased leisure time. Compare the people of today, who have an eight-hour workday, with those of just a century or so ago, who had a ten-to twelve-hour workday. Those extra *two to* four hours are not empty for us. What do we do with them now, and what will we be likely to do with the extra two to four hours more within the next few decades?



The home office of the future will integrate the computer with entertainment hardware such as videotape recorders, and digital audio and video disk players. You'll be able to work or play as the spirit moves you. User-friendly devices, such as the mouse, will extend the computer's capabilities to all family members.

Arts Renaissance

Presently, people spend time with families and friends, being entertained, exploring hobbies and crafts, learning at all levels (from reading news magazines to taking college courses), playing games, pursuing fitness and health — the list is as varied as human interests. All these activities will increase as the amount of working time (even more importantly, nonworking but related and necessary time) decreases.

The most common recreational activity at present is TV (see "Television — a Changing World" in Issue #37). With the advent of interactive cable TV and the video disk, the viewing audience will demand more and better programming. After TV, games occupy considerable screen time, as a look through any toy store or computer magazine will confirm. Whether for solitaire or multiplayer use, the number and quality of games will increase. While the traditional board and card games will not disappear entirely, electronic gaming certainly will predominate in the near and distant future. Interactive games on networks will replace some of the not always pleasant social interaction we now experience on the job.

More important are the contributions we will make in the arts and humanities given more leisure time. By nature, people like to make things, to express themselves artistically. Given time to spend, and greater access to materials and instruction, more people will develop their artistic skills and talents in painting, poetry, music, quilting, wood-turning, or theatre. With this participation comes greater understanding and sophistication, which increases the quality. Art and craft (which are two sides of the same coin) will become much more important to society as a whole. Already we see the shift from the production of material goods to the management of ideas and information.

Shared databases, and results sifted by computer, will enable more of us to indulge in primary research. Technology will progress further and faster than ever before. The guy with the crazy idea will have the time — and the encouragement — to develop the idea, to see if it works. Scientific, philosophical and psychological exploration will flourish, because society will value them.

Free To Be Human

As we acquire more leisure, we will make more meaningful contributions to society — not just to the economic and material quality of life, but also to the cultural and aesthetic quality of life. I've said before that what we do with our spare time is what makes culture. If we devote all our time to work, we can't paint. The arts define a culture even more than hunting, farming, commerce, or political systems. These changes will give us the time to devote to the philosophical, social, religious, and aesthetic questions which have been neglected. While not abandoning our desire for technological improvement, finally we will have the time to answer questions about life itself.

All these things are *because* of the computer (more importantly, the home computer), either as a toy, or as a home office workstation. Simple exposure to computers makes us more computer literate — but the more we understand computers, their potential, and how to use them, the quicker this idyllic society will come to pass.

MicroNet: Sophisticated Network Services

by Tim Knight

The large networks which I have discussed in the last two articles have many different facets. The *CompuServe Information Service,* for example, is divided into two distinct divisions. One of these is the information service itself. The other is *MicroNet,* which also costs $5 per hour, is just as useful and varied, and offers additional features.

For Experienced Users

MicroNet is the more advanced area of *CompuServe.* You won't need a Ph.D. in Computer Science to use it, but *MicroNet* does require some experience with networking and a knowledge of computers. *MicroNet* is not quite so user-friendly as *CompuServe.* Instead of a hierarchy of menus, *MicroNet* has commands which may look rather foreign. For instance, R DISPLAY would get you back to *CompuServe.* Even though many *MicroNet* commands seem cryptic, using *MicroNet* makes you feel more efficient at using a database network since you don't have to work through it menu-by-menu, as in *CompuServe. MicroNet* has several "extras," including many languages other than BASIC, large areas of memory storage, and special services. This brief overview will show the power *MicroNet* can give you as a computer user.

The Big Index

One of *MicroNet's* great advantages is access to large mainframe computers through your microcomputer. With the large amount of available memory and a variety of languages at your disposal, *MicroNet* opens up a number of special applications which are not possible on your micro. Naturally, this is going to cost you money, but if you have the inclination to work with exciting new languages or large amounts of memory, the cost might seem reasonable.

I know my computers don't have all the languages I want to try, but I can experiment with them anyway through *MicroNet.* If I had a great interest in using FORTRAN, or had an application which required it, I could jump into *MicroNet,* request FORTRAN, and begin writing in that language. What if I didn't know the first thing about FORTRAN? I could go to another area of *MicroNet,* request a FORTRAN manual, and it would be shipped to me shortly. When I received the manual, I could go ahead and learn the language. In this way, *MicroNet* not only expands the power of your computer, but also serves as a computer language teacher.

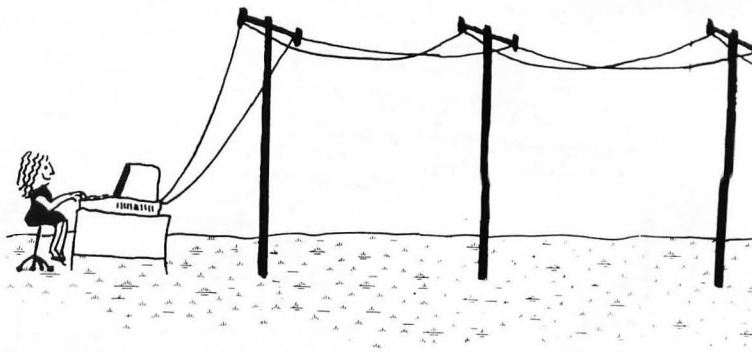
Other languages such as SNOBAL, PASCAL and an extended BASIC are available on *MicroNet.* Where will I store my programs? On *MicroNet,* where else? Since I am a *CompuServe* subscriber, I have a "free" 128K byte area reserved especially for me. If I am prolific enough to use all this space, I may rent out more memory for a small fee every month, depending on how much I need.

The advantages available with this network of languages and memory are apparent. You, as a computer user, have access to a much greater amount of memory and a much wider variety of languages than you might have thought possible on your microcomputer. You can use the service as a powerful development system for your own programs. More importantly, this is an excellent way to learn about more computer languages than you thought existed.

MicroNet's Index is one of the longest lists of available information I've seen on a network. It includes dozens upon dozens of topics, ranging from the latest fashion hints to tips on getting better SAT scores. *MicroNet* has considerably more available subject areas than *CompuServe,* and I can't begin to describe all of them, since they cover just about any subject you could think of. I will, however, describe two specific services *(MicroQuote* and *Bank-At-Home*) to show how useful and powerful *MicroNet* may be to you.

Bull and Bear Network

MicroQuote, as you might guess from the name, provides fast and accurate information on securities traded at stock exchanges and over-the-counter. If you are interested in investing, you very likely will find this a powerful tool. *MicroQuote* gives prices (both



current and historical), volume of sales, dividends, ratings, shares outstanding, earnings per share and so on. It also provides information on bonds including yields, maturity dates and ratings. You can find quickly any data you need on the 32,000 stocks, bonds and options in the *MicroQuote* database.

Using it costs more, depending on how many stocks you check and how long you use the service. For the investor, though, what does this matter? Write the cost off as a business expenditure, and reap the profits of more informed buying and selling. *MicroQuote's* powerful features can help you regain the cost many times over. Programs in the database develop statistics for evaluating investments, present descriptive data on any security, and even transfer data from *MicroQuote* to your personal computer.

Home Banking

One relatively new addition is the *Bank-at-Home* service, which allows you to make transactions without ever leaving home. You can pay bills, get the current balance on your checking account, develop your own bookkeeping system, apply for loans, obtain financial information of every kind, and send and receive messages to your bank — all from the comfort of your favorite chair. For people who cannot go to the bank every time they need to make a transaction (such as the elderly, handicapped, night workers, or auto-less), this is a very big "plus." Banking at home is one of *MicroNet's* most exciting features, and as it expands, more people will use the service to handle most, if not all of their banking business — right from their personal computer, and in the privacy of their own home. For the banks, the system saves the time and expense of processing all the checks, which will result in lower cost banking.

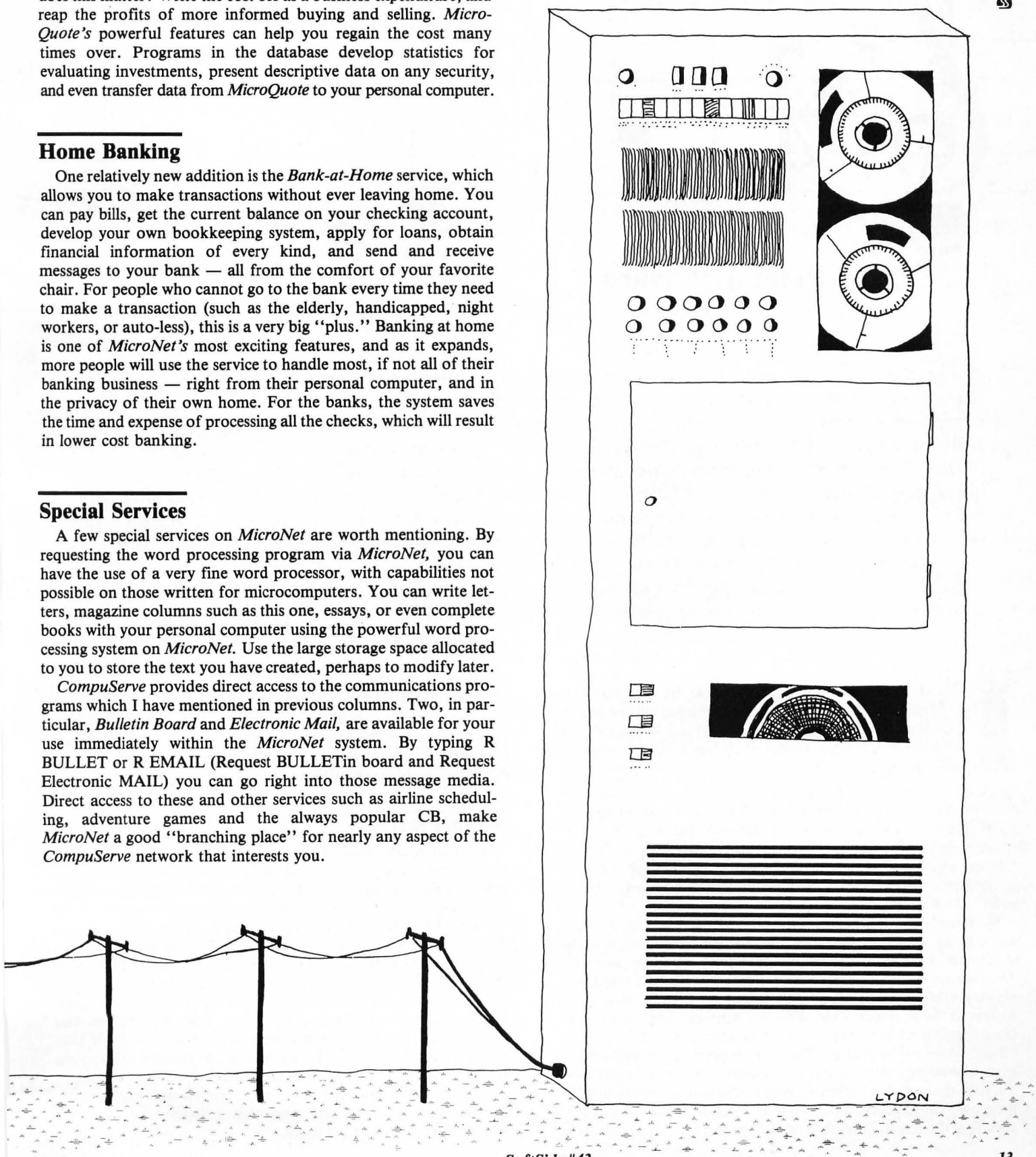
Special Services

A few special services on *MicroNet* are worth mentioning. By requesting the word processing program via *MicroNet,* you can have the use of a very fine word processor, with capabilities not possible on those written for microcomputers. You can write letters, magazine columns such as this one, essays, or even complete books with your personal computer using the powerful word processing system on *MicroNet.* Use the large storage space allocated to you to store the text you have created, perhaps to modify later.

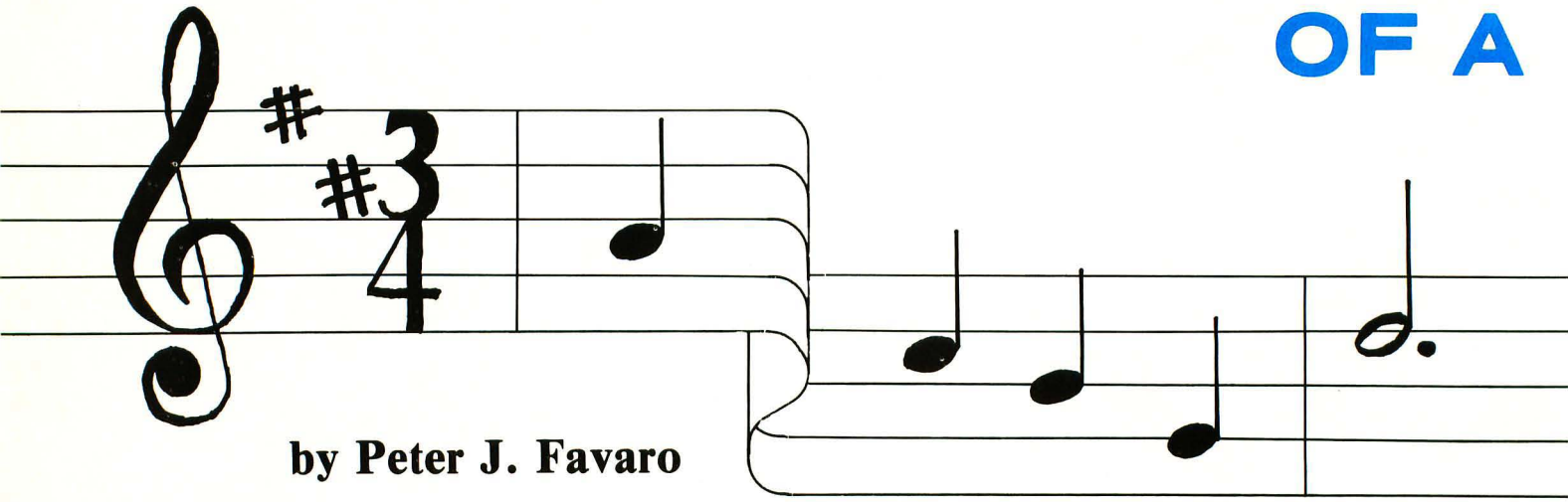
CompuServe provides direct access to the communications programs which I have mentioned in previous columns. Two, in particular, *Bulletin Board* and *Electronic Mail,* are available for your use immediately within the *MicroNet* system. By typing R BULLET or R EMAIL (Request BULLETin board and Request Electronic MAIL) you can go right into those message media. Direct access to these and other services such as airline scheduling, adventure games and the always popular CB, make *MicroNet* a good "branching place" for nearly any aspect of the *CompuServe* network that interests you.

Micronet's Virtues

These are only some of *MicroNet's* great features, so I suggest you grasp the opportunity to see the others. *MicroNet* is an important reason for signing up with *CompuServe* in the first place, since it offers some of the most powerful programming languages and services available on any network. Give *MicroNet* a try and consider yourself a part of the "sophisticated" world connection.



TAUGHT TO THE TUNE OF A



by Peter J. Favaro

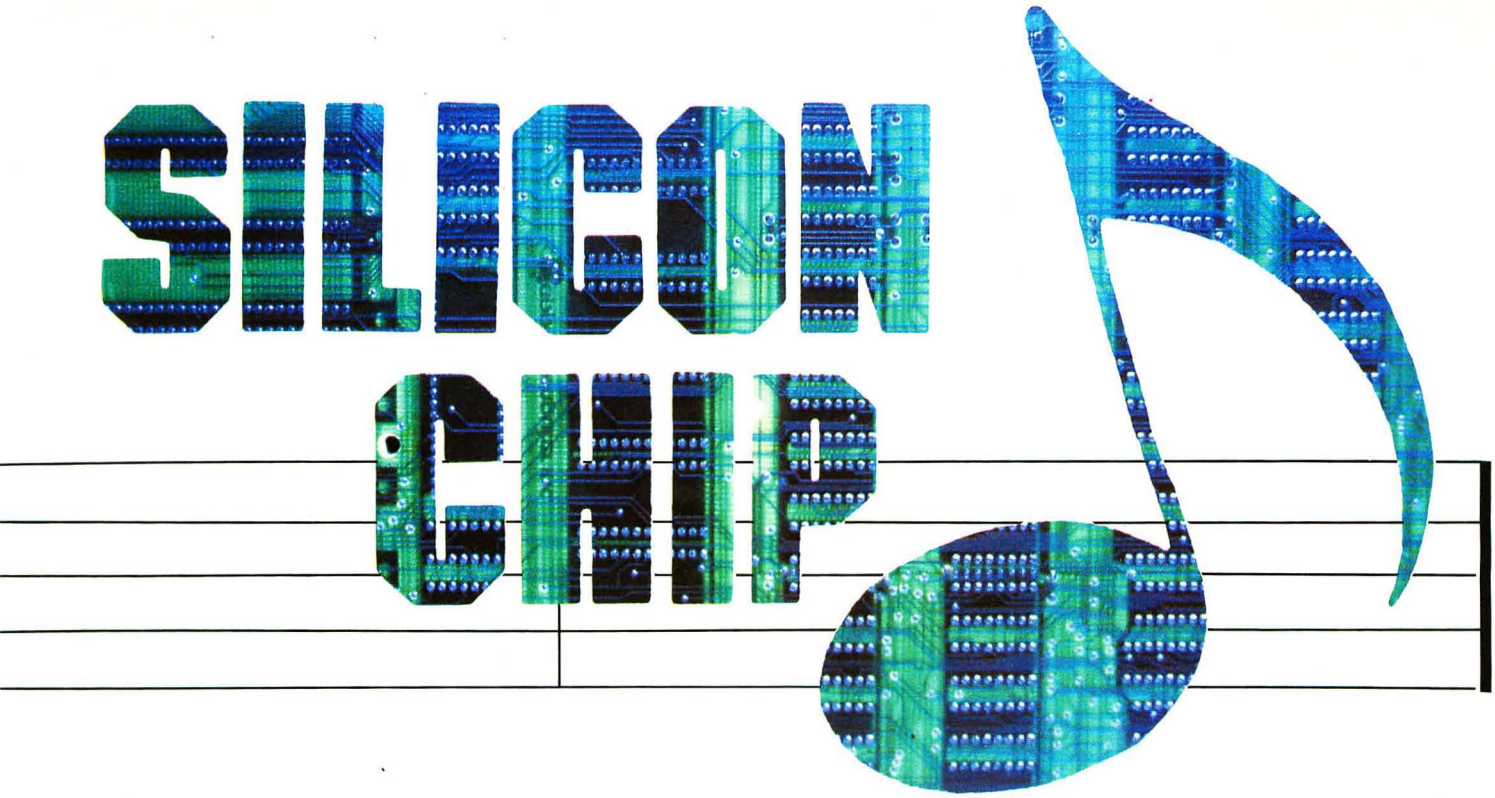
After deciding you want computers in the classroom, how do you make them do the things you want them to do? Most computer assisted instruction so far has been designed in the style of Skinnerian conditioning — an elaborate electronic flashcard. However, educators can use computers in far more imaginative ways to cultivate the higher powers of the human mind, such as reasoning, organizing information, and forming subjective judgements. All they need is the right concept for a lesson and the software to carry it out. User Referenced Interfacing and Nested Interpreters provide easy ways for teachers and students to work with large, nebulous thoughts and to call them by concrete, familiar names. In the process of manipulating and combining these larger thought structures, students can see interrelationships, and draw informed conclusions. The computer becomes a sketch pad for the mind, rather than just a drillmaster.

Educators are the easiest people in the world to criticize. As a matter of fact criticism of our educational system is a national pastime. When children have difficulty learning to read, the immediate response is that the educators are no longer dedicated, that they are inept and incompetent. When we hear that Japanese children are proficient in integral calculus at nine years of age, we point fingers at the educators and demand that they bring our children up to the same level of superiority, regardless of the differences in culture or the possible negative psychological effects of forcing children to learn complex concepts at such a stressful pace.

We have become equally comfortable criticizing educators for not making immediate progress toward using computers in the classroom. I have, on numerous occasions expressed my own disappointment at the current status of educational computing, offering criticisms and comments. Rarely have I provided alternatives to the problems I see. This is a mistake. The people who are struggling to find ways of utilizing computers effectively in education should be praised. They are pioneers. They stay up late at night reading technical manuals that make little sense to them. They do most of their learning about educational computing on their own time, getting little support from their administrators, and from the public at large. I think the time for this to change is now. In this article I will present new strategies for implementing computers in the classroom — not to criticize the current efforts, but to enhance them. I hope to start a new trend toward treating these people, who devote a significant portion of their lives to helping children learn, with more understanding and support.

Skinner Boxes

Most computer-assisted teaching today uses Skinnerian conditioning techniques. This approach is named after Harvard psychologist B.F. Skinner, who invented the "teaching machine" or "Skinner Box" in the Fifties. "Operant conditioning" is the term Skinner used to describe a process of rewarding desired behavior (a correct answer) until the trainee responds correctly every time. Rewards can "shape" even random responses until the subject gives the correct response predictably and consistently. The rewards, or reinforcements, can be tangible, or a simple confirmation that the answer is correct. Skinner's theories had great appeal to the educational establishment in the Fifties because they seemed to offer a scientific, or rational *explanation* for learning. For believers in cause and effect, operant conditioning was an ideal concept. The "operant framework" in computer-assisted instruction is an interaction between the computer (teacher) and the user (student).



The best-known computer implementation of operant conditioning is the "electronic flashcard." This approach has been condemned and praised by educators. On the positive side, the electronic flashcard model of educational computing has proved that computers are efficient drillmasters (even more so than another expression of Skinnerian conditioning, the programmed learning workbooks popular in the Sixties). They provide repeated practice and give immediate feedback. They reinforce and strengthen word recognition and computational skills. Best of all, by providing this service, computers allow teachers to do what they do best: teach new skills. This application alone is enough to justify the presence of several terminals in each classroom, but in my opinion it is only the tip of the iceberg in terms of the computer's overall usefulness in the classroom.

When using the computer strictly as an electronic flashcard you must be aware of its limitations. I do not imply that using the computer as an electronic flashcard is wrong, I simply believe that this approach has limited uses. One limitation is that it breaks the learning process into distinct one step units. At best, it works hierarchically, presenting the easiest "chunk" of information first, followed by a more difficult chunk which depends on the first one, and so on. While this may be useful in teaching some concepts (those susceptible to the fragmentation technique), *it does not simulate the natural way that humans think, behave or solve problems and is therefore limited.*

Consider the typical electronic flashcard lesson, most of which use CLOZE techniques (see Issue #32 for an explanation of the technique). Usually, the lesson takes the form of a fill-in-the-blank exercise. The student's first task is to read the text displayed on the screen. The computer then poses a series of questions which are relatively simple and straightforward (a definite advantage of some CLOZE-type exercises). The student either answers directly, or goes back to the text to find the answer. After the student responds, the computer indicates whether or not the answer is correct, and keeps track of the student's score.

A number of questions arise at all points during this interaction. First, what is being tested? Is it reading comprehension, the specifics of the lesson, visual memory, guessing ability, or all of these? Of course you could control some of these factors so that content is more heavily tested than any of the others. You could remove the passage from the screen before asking the question. This would prevent the student from looking back for the answer. You could eliminate the choices and have the computer accept responses typed in from the keyboard. Both of these strategies have other implications. To conquer the problem of the computer rejecting correct answers, you must program it to accept many different variations in concept and spelling. After all, a lesson should not be overly concerned with spelling as long as the student retains the concepts — or should it? You see, the issues get somewhat confusing and complex with regard to the way information is presented. The problem is to create a structured environment that is flexible enough to allow for the trial and error aspect of human learning, but structured enough to teach something. Achieving this delicate balance is the key to developing effective computer aided educational experiences.

The problem is to create a structured environment that is flexible enough for the trial and error aspect of human learning, but structured enough to teach something.

Human Engineering for the Classroom

How the student operates the computer is a crucial aspect of the educational environment. In most educational software I have reviewed, the author takes great care to *restrict* the user's input. Many keys are disabled. Joysticks, mice or other devices are used to keep children away from the keyboard, and other types of error handling are strongly emphasized. Usually, the younger the child, the more restricted are the input possibilities. ↪

Jean Piaget (a Swiss cognitive psychologist whose work is familiar to anyone who has studied education and learning), has made extensive observations on the way children learn throughout their developmental cycle. While his writings are extensive and somewhat difficult to understand, much of what he said about learning focused on the child's ability to manipulate his environment. As the child learns to do this, he learns about his environment, and therefore is able to control it. Piaget describes four major stages — from the infant's Sensori-Motor stage, through Pre-Operational Thinking, and Concrete Operational Thinking, to the young adult's Formal Operational Thinking. These stages describe the development of thinking as moving from concrete to abstract, simple to complex.

By limiting input and using the Skinnerian approach of breaking knowledge into tiny particles, you make 'manipulating the environment' a very difficult task. Nothing is left to manipulate! What alternatives are available to let you present environments that *can be manipulated,* while retaining the error checking necessary to guide the student? The answer is to develop activities that present problem solving situations to children — environments that allow errors, perhaps even foster them, thus moving from an *operant* to a *cognitive* framework. Learning the typical wrong solutions gives a broader perspective and greater mastery of the subject than simply learning the correct solution. The object is to provide feedback which leads children to manipulate the problem and find a solution.

Logo and Problem Solving

By now anyone familiar with Logo is saying that most of the ideas I have been developing so far are old hat to them. For those who do not know about Logo let me explain it briefly. Logo is a programming language developed at the MIT artificial intelligence laboratory. One of its developers, Dr. Seymour Papert worked with Jean Piaget and long ago understood the importance of a problem solving approach. He kept this in mind when developing Logo.

A small part of Logo involves an interesting, and famous little creature called a "turtle." Actually it is a cursor which moves around the screen, drawing, coloring and creating shapes as it goes. Turtle drawing is a marvelously educational experience because it allows flexibility, trial and error learning and the kind of mastery that Piaget and Papert talk about. It creates a microworld where children develop the tools for manipulating the environment (in this case the turtle and its environment) to solve problems. It is primarily visual in nature, which also makes it a stimulating educational experience for children. Logo has other capabilities besides turtle graphics, which are also worth knowing about. Rather than turning this into a Logo article I will refer you to Papert's book *Mindstorms* (Basic Books, New York, $6.95) for a more detailed explanation.

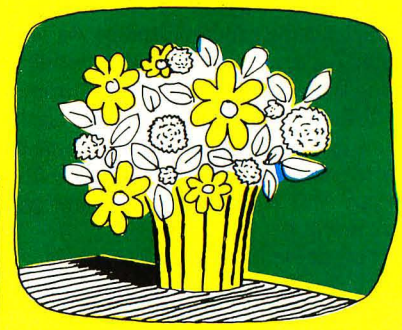
The reason I bring Logo into this discussion of alternatives in educational computing is twofold:

- First because of its unique approach, its flexibility, and its problem solving orientation.
- Second because it draws upon a completely untapped area of computing that has important implications in education, namely artificial intelligence. Techniques to make student-computer interfacing more intelligent are among the ideas I hope to introduce. Unfortunately AI occupies a mysterious place in contemporary thinking about computers. Usually we associate AI with robotics, the eventual domination of computer thought over human thought, and other spooky things.

HOW A NESTED INTERPRETER WORKS

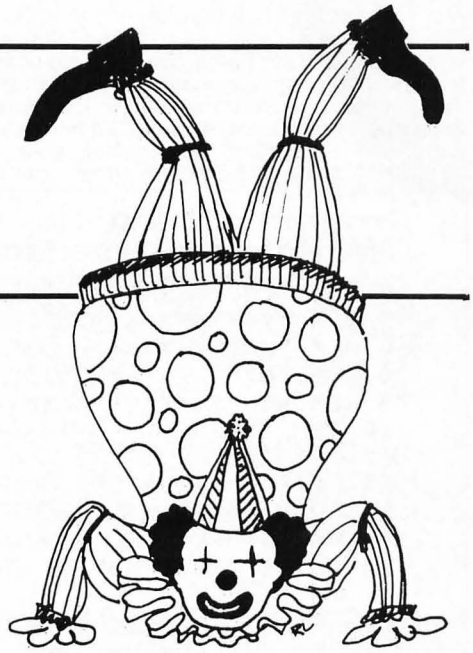
To prepare the lesson, the teacher writes the master program. In this case, it draws flowers and positions them on the screen.

The student explores all the relationships of sizes, shapes and colors for the flowers, and selects the best examples. As she does so, she gives an easily recognizable name to each design. In the next step, she will be able to call these complex drawing routines simply by typing "Small Red Carnation" or "Large Yellow Tulip."



Finally, by stringing together these names, the student assembles them into a larger structure, such as a floral arrangement in a vase. By changing her assembly program, the student can create variations on this larger design. Repetition of elements, color changes, density of elements and so on, are all under the student's control as she improves the arrangement's design.

Each program can create a different environment through two-way communication with the computer, and by asking the student to take apart the different problems presented, to understand them in his own terms and manipulate them in his own language.



AI for microcomputers has not been discussed seriously because many people believe the limited memory makes most AI applications on micros unrealistic. I disagree, simply because AI is nothing more than a philosophy. The basic principle is that humans should be able to interact with computers in a more human way. To do this, computers need to be as similar as possible to humans in their reasoning behavior — and this has proved to be a difficult task. I think the AI philosophy can be applied to BASIC in ways similar to Logo and some AI applications, but without the memory requirements and mystery. I will present two techniques, User Referenced Interfacing, and Nested Interpretation, which have many uses in education and address many of the limitations of contemporary educational computing.

Communicating With Your Computer

Think for a moment about the most important elements in a successful interpersonal relationship. The one which comes to mind almost immediately is communication. A reciprocal understanding of each person's experience, or point of view must occur. The primary vehicle through which we achieve such an understanding is language. Through language we learn — about others, about ourselves and about our environment. Recent studies have shown that language is perhaps the most important feature of the learning experience.

The language of educational computing and of computing in general does not make for effective communication. The computer as a communications partner demands absolute perfection: correct spelling and proper syntax (which may be entirely foreign to the user). If these conditions are not met the user gets the "silent treatment." Could you imagine a friend who was as demanding as your microcomputer? Never! If the computer could somehow meet the user half way, things would be much easier. This is, at least on the surface, a big problem. Computers already need "interpreters" to allow you to interact in words rather than numbers. If you are willing to give up a few machine cycles and some of the computer's capabilities, you can create a system of two-way communication between computer and user — your own programming language. You can do this with a technique I have developed called User Referenced Interfacing .

Listing One is an example of URI for the Atari 400/800 which uses about 3K of RAM. The application is a simple one with no real educational implications other than to demonstrate the technique, using a series of five graphics and sound routines. After each demonstration of what that routine can do, the computer asks the user to name that activity. The computer remembers the name, thus creating a user-defined command set. At the end, it prints out the command set and asks you to write "programs" which perform the activities in any order, and as many times as needed.

What are the educational implications for a User Referenced Interfacing System? Each lesson a teacher creates has its own intended learning outcomes. These goals are defined in terms of the learning task. The teacher generates URI lessons, and provides a "library" of related information chunks necessary to the lessons. Each program can create a different environment through two-way communication with the computer, and by asking the student to take apart the different problems presented, understand them in his own terms and manipulate them in his own language. The key to this approach is for the teacher to identify the salient aspects of the lesson. The fact that the student must write a program using his own commands eliminates the step-by-step process of the usual educational interfacing. In doing so, the URI lesson develops planning ability, problem solving techniques, sequencing and other cognitive skills as well.

"Send In the Clowns"

I recently developed a URI lesson for a group of learning disabled second graders. The program displayed a series of clowns performing various movements. In reality each clown represented one frame of an animation sequence. The problems presented to the children were phrased "Make the clown do a flip," "Make the clown do a jumping jack," etc. Through this exercise the children developed their own plan of attack using their strengths instead of their weaknesses. They used their own words, which almost always made sense and were correct spellings of typical words like "turn," "right," and "left."

Soon I will test to see if their work with this program has had any transfer to other academic tasks. Even if it doesn't show an immediate effect, an exercise like this helps to build organizing and planning skills at the very least.

Any system has inherent limitations and URI is no exception. To utilize these concepts, you must be able to write the main program, which in turn develops the commands and also runs the user's program. This requires a sophisticated knowledge of BASIC, so to make this task simpler, I will present briefly the necessary techniques.

Nested Interpretation

Most languages use an interpreter. BASIC, Logo, Forth and Pascal are all examples of interpreted languages. Machine language does not require interpretation. It acts directly on the CPU, but is much harder to program than most of the interpreted languages. In BASIC, instructions (called "tokens") represent

→

Instructions: Type in this program and RUN it in the usual fashion. The computer will ask you to "name" the operations that you see on the screen. Each name must be eight characters long, no more, no less. If you want a command with less than eight letters type in the command you want and fill in the remainder of spaces with asterisks. For instance, if you would like to use the word "FLASH" (only five characters) as one of your commands, type in "FLASH***" (eight characters). When your computer asks you to type in a "program," type in your "commands" in a linear sequence. DO NOT TYPE "RETURN" AFTER EVERY COMMAND LIKE YOU DO IN BASIC. Example:
FLASH***STARS***SOUNDS**ARROWS**LASERS**FLASH***MUSIC***, then type the RETURN key. In this demo, you are limited to "programs" that are three lines or seventy-two characters long.

Program Listing 1:
User Reference Interfacing Demo for the Atari® :

```
50 REM DEMO OF***USER REFERENCED*****
55 REM **********INTERFACING*******
65 REM *********BY PETER FAVARO******
70 REM ***** © SOFTSIDE PUBLICATIONS*
75 REM ***************1983**********
110 DIM A$(100),R1$(12),R2$(12),R3$(12),R4$(12),R5$(12)
115 PRINT CHR$(125)
120 PRINT "I PERFORM 5 OPERATIONS."
130 PRINT "HERE IS THE FIRST."
140 FOR PAUSE=1 TO 900:NEXT PAUSE:GOSUB 370
150 PRINT "WHAT WOULD YOU LIKE TO CALL THIS      OPERATION?":INPUT R1$:? CHR$(12
5)
160 PRINT "THANKS,NOW FOR THE SECOND OPERATION...":FOR PAUSE=1 TO 900:NEXT PAUSE
:? CHR$(125):GOSUB 390
170 PRINT "WHAT WOULD YOU LIKE TO CALL THIS?":INPUT R2$:? CHR$(125)
180 ? "AND NOW, MY THIRD OPERATION...":FOR PSE=1 TO 900:NEXT PSE:GOSUB 400
190 GRAPHICS 0:PRINT "WHAT WOULD YOU LIKE TO CALL THIS?":INPUT R3$:? CHR$(125)
200 ? "THIS IS MY FOURTH OPERATION...":FOR PSE=1 TO 900:NEXT PSE:GOSUB 420
210 GRAPHICS 0:PRINT "WHAT WOULD YOU LIKE TO CALL THIS?":INPUT R4$:? CHR$(125)
220 ? "THIS IS MY LAST OPERATION...":FOR PSE=1 TO 900:NEXT PSE:GOSUB 450
230 GRAPHICS 0:PRINT "WHAT WOULD YOU LIKE TO CALL THIS?":INPUT R5$
240 GRAPHICS 0
245 REM PUT COMMANDS ON SCREEN
250 PRINT R1$:? R2$:? R3$:? R4$:? R5$
260 ? "THESE ARE THE COMMANDS THAT ";? "YOU CREATED."
270 ? "TO WRITE A PROGRAM";? "SIMPLY TYPE IN THE COMMANDS"; "THAT YOU CREATED T
HEN RETURN":INPUT A$
280 X=1:Y=8
290 IF A$(X,Y)=R1$ THEN GOSUB 370
300 IF A$(X,Y)=R2$ THEN GOSUB 390
310 IF A$(X,Y)=R3$ THEN GOSUB 400
320 IF A$(X,Y)=R4$ THEN GOSUB 420
330 IF A$(X,Y)=R5$ THEN GOSUB 450
340 X=X+8:Y=Y+8:IF X>LEN(A$) THEN 240
350 GOTO 290
360 END
365 REM SUBROUTINE LIBRARY IS HERE
370 GRAPHICS 2:FOR HUE=1 TO 15:SETCOLOR 4,7,HUE:FOR PAUSE=1 TO 100:NEXT PAUSE:NE
XT HUE:SETCOLOR 4,0,0
380 GRAPHICS 0:RETURN
390 FOR TIMES=1 TO 30:P=INT(RND(0)*200):SOUND 0,P,10,10:FOR PSE=1 TO 20:NEXT PSE
:NEXT TIMES:SOUND 0,0,0,0:RETURN
400 GRAPHICS 2:FOR TIMES=1 TO 19:POSITION TIMES,5:? #6;"*":SOUND 0,121,2,10:FOR
PSE=1 TO 40:NEXT PSE:SOUND 0,0,0,0
410 NEXT TIMES:RETURN
420 GRAPHICS 2:MM=19:NN=1:POSITION MM,5:? #6;"<":POSITION NN,5:? #6;">"
430 FOR T=0 TO 17:MM=MM-1:NN=NN+1:POSITION MM,5:? #6;"<":POSITION NN,5:? #6;">":
FOR PS=1 TO 50
440 NEXT PS:NEXT T:RETURN
450 GRAPHICS 7:COLOR 1
460 FOR D=1 TO 8
470 I=INT(RND(0)*150):J=INT(RND(0)*70):K=INT(RND(0)*79)
480 PLOT I,J:DRAWTO J,K
490 FOR DEC=10 TO 0 STEP -2:FOR TONE=85 TO 0 STEP -5:SOUND 0,TONE,10,DEC:NEXT TO
NE:NEXT DEC
500 NEXT D
510 RETURN
```

Instructions: The "O" that you see on the screen is actually a rudimentary "turtle." You can create simple shapes by programming the turtle to move up, down, left and right. The arrow keys correspond to the movements (i.e., up arrow means move up one space). Each "move" must be preceded by the letter "O" (this helps the turtle get his bearing), so a move of one space up would be "O–" (the minus will appear instead of the up arrow sign). As with Program listing 1, you are limited to an input string of 72 characters; and characters should be typed in one after another. RETURN will cause the "program" to execute. Example of what to type in to create a 3 unit square: "O – O – O – O*O*O*O = O = O = O + O +", then RETURN.

Program Listing 2:
Nested Interpreter Demo for the Atari® :

```
100 REM DEMO OF A NESTED INTERPRETER USING STRINGS
105 REM THIS PROGRAM SIMULATES A SIMPLE 'TURTLE'
110 GRAPHICS 2
120 DIM A$(100),B$(2):B$="O"
130 DX=5:DY=5:POSITION DX,DY:? #6;B$
140 INPUT A$
150 X=2
160 IF A$(X-1,X-1)="O" AND A$(X,X)="*" THEN GOSUB 220
170 IF A$(X-1,X-1)="O" AND A$(X,X)="-" THEN GOSUB 230
180 IF A$(X-1,X-1)="O" AND A$(X,X)="=" THEN GOSUB 240
190 IF A$(X-1,X-1)="O" AND A$(X,X)="+" THEN GOSUB 250
200 X=X+1:IF X<=LEN(A$) THEN GOTO 160
210 END
220 DX=DX+1:DY=DY:POSITION DX,DY:PRINT #6;B$:RETURN
230 DY=DY-1:POSITION DX,DY:PRINT #6;B$:RETURN
240 DY=DY+1:POSITION DX,DY:PRINT #6;B$:RETURN
250 DX=DX-1:POSITION DX,DY:PRINT #6;B$:RETURN
```

the operations BASIC can execute. GOTO, GOSUB, FORNEXT are all examples of tokenized instructions. Before the CPU can act on any of the BASIC instructions they must be translated into tokens.

You can create a similar system using strings inside a BASIC program and, in effect, create an interpreter within an interpreter (a Nested Interpreter). First, decide what tasks you want your program to accomplish and write a series of subroutines to carry them out. Give each subroutine a name. To assemble a larger program, string the modules together, by name. What the larger program accomplishes is determined by the order you call the modules, and which ones you use. The important point is that writing the functioning program is extremely easy — just list the commands.

Listing One uses NI to generate a user defined language. Listing Two uses NI to program a simple turtle-like object to move up, down, left and right. The difference between the two Listings is that the first asks for the commands, while the second already defines them. Here is a step by step breakdown for writing a "Nested Interpreter:"

- 1. Develop your subroutine modules and place them at the beginning of your program (BASIC always goes to the top of the program to look for GOSUBs).
- 2. Name the commands that will call the modules, or
- 2a. (optional) Have the user define them.
- 3. Ask the user to select the modules and place them in a specific order, thus forming the input string which becomes the program.
- 4. Instruct the computer to point to the first character or group of characters in the string.
- 5. Use IF (string place value = name of subroutine) THEN (to carry out the procedure...) GOSUB (name of module), to tell the computer to carry out the sequence of instructions.
- 6. Increment string place counter by one, do the next operation, and so on.

URI and NI Summary

These methods are not complex. They emphasize problem solving, reasoning and planning and can be made flexible enough to teach almost any subject area. The drawbacks of using a system like this, as I mentioned before, are a loss of computing power and the requirement that teachers who develop such programs must have relatively sophisticated programming knowledge. I hope to present some skeleton programs in the future which will simplify this.

Evaluating Existing Software

If NI and URI seem too abstract and difficult for most teachers, what can we do for teachers who do not know how to develop educational software, but have computers in their classrooms, and want to make sure the software they use is in fact, worth using?

One of the problems in the educational software market today is the fact that programs which are advertised as "*educational*" often do not teach what they intend to teach. I mentioned this problem before and suggested that educational software should become more flexible. Designers should move away from formats like the CLOZE technique and emphasize more cognitive, problem solving approaches.

A long-standing problem for educational computing has been how to create activities developmentally appropriate for each age group. Obviously you would not sit a toddler down in front of a terminal and ask him or her to start punching in answers to a computer-assisted physics course. But what is the correct age (if there can be such a thing) to present a mathematics lesson that involves addition with carrying, or subtraction with regrouping? The answer is not so clear, particularly when you look at what is being offered as educational software. This problem stems from the fact that most educational software is being written by programmers who have little knowledge of elementary education or curriculum development. To them, anything that hints at reading, writing or arithmetic is "educational." I believe this is the single most dangerous trend in educational computing today. If allowed to continue, the computer in the classroom will become another one of those nice ideas that didn't work out, just like overhead projectors and filmstrip machines.

More of the larger companies like Atari and Mattel are investing in education. From what I have seen in both of these companies, a few organizational problems remain — such as coordinating the talents of computer programmers, educators and creative people. I believe that one of the essential members of a software development team in large companies is a person who has knowledge of programming, education and learning principles. Such a person would interface with all the phases of educational software design and translate concepts from one set of jargon to another. He would not only speed up the production of educational software, but more importantly, improve the quality

↪

of what is currently being developed.

Suppose however, that you do not want to wait for good educational software to arrive. Most of the latest surveys identify education as the area consumers are most concerned with. How can you make sure you are not throwing money away on inappropriate software? One way is to develop a more standardized approach to educational software evaluation. In psychometrics, the science of test development and construction, psychologists and educators must be able to show that their measurements are both reliable and valid. "Validity" refers to whether the test measures what it's supposed to measure. "Reliability" refers to a test's consistency over time. Take a more psychometric approach to educational software evaluation. Assess its appropriateness and rate it relative to whether it actually can teach the user something. You can do this informally simply by asking the right questions as you look over a program.

Of course, to do this thoroughly, involves research, time and money — which the major corporations have in abundance. If half the time spent researching titles for game cartridges were spent on developing educational software, the market would have plenty of good educational software. I think that consumers have a right to see evidence the software they are buying is effective educationally. An excellent source for educational reviews is the Courseware Report Card (150 West Carob Street, Compton, CA 90220). The approach taken in reviewing software, educational computer publications and even magazines is professional and well done. I suggest that any teacher who is serious about educational computing subscribe.

Another way to evaluate quality in educational software is to have a group of experts review it. This is the approach I took in evaluating the speech synthesizing hardware and software in Issue #41 (*Speaking Easy*) and I will use the method in other reviews. I found the input from the speech experts I consulted to be helpful and used it to teach readers from a perspective that I do not have. Starting with this issue you will see occasionally some educational reviews done with the help of students who have used the software. Feedback from the users is one of the most important but neglected aspects of educational software evaluation.

Lifelong Learning

The last point I want to make is about adults and educational computing. Much of the effort in educational computing has been directed toward reinforcing what is done in the classroom. This again limits the use of computers in an educational framework. "Education" is a big term. As a school psychologist, I see that much of what I do when I practice therapy is educative in nature. We are presenting the Relaxation Training System in this issue because we want to let people know that our commitment to education extends beyond the 3Rs and includes self-help, and adult education. In the future you will see cooperative games which help teach positive social behavior, along with diet and health management programs, and much more. As personal computing becomes a more common activity for people to engage in, the computer will become a knowledge resource. Shoot 'em up games are a thing of the past. After we have run out of alien planets, swords and sorcererers, and movie titles for arcade games, we will use computers to explore our minds more deeply. Recreational games will be more stimulating because you will be more involved cognitively. You will want to play games which emphasize strategy and reasoning, such as the group games which are fast becoming popular on The Source and CompuServe. The nature of humans is to seek stimulation. We are becoming less stimulated by arcade games. They are too much alike. We are getting used to them. To predict what will replace them is difficult — but stay tuned. Chances are, you may see it here first.

"Good" Educational Software for the Apple®?

COMPress has it!

Many educators bemoan the fact that there are hundreds of educational packages coming on the market — often lacking author names or credentials. Nearly every COMPress author is a teacher or professor involved daily with students using Computer Assisted Instruction on the Apple computer. The programs they produce use animations, graphics — often with color, and the latest instructional techniques.

Whether it's an educational game, an authoring language, classroom demonstration material, or a more traditional student "hands-on" CAI package, you can count on one thing from COMPress — it works!

And that's "good".

Interactive Computer Assisted Instruction Materials for High School and College Levels

INTRODUCTION TO GENERAL CHEMISTRY
INTRODUCTION TO ORGANIC CHEMISTRY
ORGANIC QUALITATIVE ANALYSIS
IMPROVING YOUR WRITING SERIES: A Rhetoric Program
STATISTICS AND PROBABILITY DEMONSTRATIONS AND TUTORIALS
ENGLISH AS A SECOND LANGUAGE

Demonstration Diskettes Available

EnBASIC™ AUTHORING SYSTEM
MOLECULAR ANIMATOR
INTRODUCTION TO CRYSTALLOGRAPHY
ENGLISH GRAMMAR DRILLS
WORLD CLOCK
INDEXOR
TEST GENERATOR
PACOR: Multiple Choice Tester
CASE HISTORY: Multidiscipline Problem-Solving
Our package for the TRS-80* Model I and III: INDEX COMPILER

Call us now and request our catalog and demonstration diskettes, or return the coupon below.

COMPress A Division of Van Nostrand Reinhold Company, Inc.

P.O. Box 102, Wentworth, NH 03282
(603) 764-5225/5831

☐ **Yes, send me your catalog.**

Name _____

Company _____

Address _____

City, State, Zip _____

Phone _____

☐ Send me the following demonstration diskettes:

_____

_____

S8/83

® Registered trademarks of Apple Computer Company
TM A trademark of Computer Teaching Corporation
* Registered trademark of Tandy Corporation

The typical all-American-kid's room still is filled with "stuff" but now it includes a computer. Atari's 600XL, for instance, can teach music theory his school usually doesn't include in its curriculum, enabling him to perform better and enjoy music at a deeper level. Computerized instruction in the home is already blurring the distinction between homework and schoolwork.

CRISIS IN EDUCATION

—Can the Computer Help?

by Saul Bernstein

At COMDEX in Atlanta last May, Norman Vincent Peale remarked that all living beings have one thing in common — problems! The quality of the solutions to these problems, he said, is a measure of the person or the civilization. Education is a product of civilization. It is the cornerstone for the future because that is how human society transmits its accumulated wisdom to the next generation.

An examination of our university system clearly shows that today only three universities are graduating students who, while capable in their special art or science, are also liberally oriented to western values. These schools seem to be the only ones left that require a knowledge of Chaucer, Shakespeare, John Locke or Spinoza. I speak, of course, about West Point, Annapolis and the Air Force Academy. *(Ed. — Though this trend is lamentable and endemic to higher education, some schools without the military orientation continue to nurture the liberal arts tradition. DePauw University, Harvard, Yale, Chicago, UC Berkeley and Brown are names that come to mind quickly.)*

In the past, we had an agrarian civilization with small communities scattered across the countryside. In the center of these communities we erected little red schoolhouses. The few people who could read and write became teachers or administrators in these schools. As we changed from an agrarian to an industrial civilization, the little red schoolhouse tradition continued without skipping a beat. Yes, the teachers and administrators became more adept at their jobs, but the schools have not changed in any appreciable way. The school day still ends at 3:00 PM and the academic year ends before summer. Relatively few students in the late twentieth century live and work on family farms. Why, then, are we still letting the children out of school early to tend the farms? We should be teaching them until 5:00 PM, five days a week, for the entire year.

The business of a classroom instructor is to infuse his students with the skills and knowledge of the given subject. Education circles are always full of speculation on novel methods of learning, and many systems are tested. Unfortunately, most of the current methods don't work particularly well. They produce armies of students who can talk your ear off, but can't write a sentence.

As an instructor with 23 years of experience, I believe I know how to teach and what a student must do to learn. The body of subject matter a student must learn today is much greater than in my own student days, but this does not alter the teaching method. Since the days of Socrates and Aristotle, the most essential aspects of teaching have remained unchanged. In more recent times, beginning with the development of the British Royal societies, teaching methods have been well codified. Computers have altered technique and method radically, by taking care of the necessary but tedious drill routines — leaving human teachers free to interact with students at a higher level.

Relatively few students in the late twentieth century live and work on family farms. Why, then, are we still letting the children out of school early to tend the farms? We should be teaching them until 5:00 PM, five days a week, for the entire year.

Memorizing and Reasoning

Demonstration and discussion are important, but memorization and rote imitation are fundamental. Five thousand years of experience have revealed to mankind certain verities in art and science which are not subject to interpretation by either the instructor or the student. We must teach, and students must learn, the essential rules of the game.

Human intelligence should be directed toward learning how to ask the right question in the right way. The biggest problem in education has not been how to teach any given subject. The problem has been not teaching logical thinking — how to use knowledge to find answers to problems.

To acquire facts, students sit with their instructional material before their eyes and memorize it. When the instructor calls upon the student for the information, the student recites it verbally or in writing. The student gives back the essential truth, mimicking the author. No deviation can be tolerated. For example, Euclidean geometry is not subject to debate. Each theorem, axiom and postulate must be stated exactly as given. In this way, we pass the benefits of our civilization from one generation to the next.

Computers: Tools For Thought

The advent of the computer has changed what it is we teach a student. The computer software provides answers and procedural *methods*. For example, is it really necessary for a student to memorize the multiplication tables? Under current practice, the student spends an entire school year learning multiplication alone. If you consider addition, subtraction and division, then we are talking about a minimum of three school years. Memorizing arithmetic "facts" actually slows down the process of education. What a student needs to know is how to *manipulate* numbers to *derive* the *information* he needs. Memorizing the tables is only a means to that end. Why use up three years of the education process when a commonly available tool (a $3 calculator) provides the answer? Human intelligence should be directed toward learning how to ask the right question in the right way. The biggest problem in education has not been how to teach any given subject. The problem has been teaching logical thinking — how to use knowledge to find answers to problems.

We should teach adversary logic and the rules of evidence as a separate subject. Socratic dialogue and the computer were made for each other. The student of today must learn to use the on-line database as a tool of operation, just as students before learned to use books and libraries (or Socrates' memory) as a database. This is not being done presently. Logic is not taught as a subject. If touched upon at all, it is given as an arcane abstract of speculative thinking. Logic is a tool which must be used and taught every day — starting at the grade school level. The three years now spent on rote memorization of arithmetic facts are no longer necessary. With computers to do all the number crunching, students need to know logic, reasoning and fact-finding more than ever before. In the late twentieth century, the question is not what is two plus two, but what can you do with the answer?

In the elementary and secondary schools, we should establish a curriculum based on subject value. We must decide which subjects are truly essential, and grade students by subject. Put the schools on a four-quarter system, and teach the first, second, third and fourth quarter of each subject in each calendar quarter. Under the present system, a student who fails must repeat the entire school year. The new system would require repeating only the subject failed in the next immediate calendar quarter, just as we do at the university level. Conceivably a student could be in the fourth grade, third quarter of history and the fifth grade, second quarter of math. The exceptional student could simply test through each quarter until he found his level in each subject. A student might finish post-graduate studies by age 25.

The proper use of computers makes this system possible. Three main-frame computers strategically placed across our nation could track each student's entire school record from kindergarten through the university. This would go a long way toward insuring the cohesive educational system we all want.

National Priorities

Education is the keystone of western civilization. America's success has always been based upon excellence in education. The fact is that the school system is basically obsolete. The present system was designed for a society of farmers. It does not serve the needs of a space-oriented, high-tech industrial society.

Three times in my life our nation has made a special effort to pursue educational excellence. The first was the G.I. Bill after the Second World War, the second was the G.I. Bill after the Korean War, and the third was our program to land a man on the moon. Whatever the reasons were — fear of the Russians or reward to those who served in our armed forces — our country has derived great benefits economically, socially and scientifically. The growth of the American computer industry specifically in California's Silicon Valley and Boston's Route 128 is recent proof of the important role strong educational institutions can play in stimulating new industry.

Some time ago, Steven Jobs of Apple® Computer offered to equip the California schools with Apples at no cost to the schools. The deal was refused because Apple would make a profit. To place at least one computer in each school required a special act of the California legislature. Accepting Apple's offer would have been far wiser. One day soon, special tax concessions (probably on the Federal level) will be given to manufacturers to place computers in the schools. Furthermore, I want every one of the donor companies to make a profit and employ large numbers of people. Personally, I find large-scale unemployment an anathema.

American industry must become more involved. Today, this usually takes place at the post-graduate level only, and with very specific projects. To insure its own survival and future, industry will be forced to consider lending us people capable of teaching the sciences. If we could borrow some of their experts, even one day per week, we could start producing the graduates in math, computer and general science they will need in the near future. Tax relief seems a likely vehicle to start this process and would provide the right capitalist incentive.

Education In The Global Village

Today, we must demonstrate the wit to understand what the Founding Fathers knew by instinct. In the years since World War II, American foreign aid has been responsible for building more hospitals, schools and universities in foreign nations than any other country in all human history. We have trained more doctors, lawyers, artists and scientists than any other nation in all of human history combined. We need to do this for ourselves as well — a Marshall Plan for the United States. The American people must not lose their way; the American teacher must not stray from this path. We will adapt; we will change, and present the world with a generation properly equipped to cope with the demands of a highly technological world.

☐ APPLE ☐ EAGLE ☐ IBM, PC ☐ NORTHSTAR ☐ TRS-80 II 8" ☐ OSBORNE ☐ FRANKLIN ☐ STANDARD CP/M 8" ☐

☐ ACCOUNTS PAYABLE ☐ MAILING ☐ SPELLING ☐ COMMUNICATIONS ☐ REAL ESTATE ☐ TAXES ☐ STOCK MARKET ☐ INVENTORIES ☐ HEATH/ZENITH 89 8"

☐ XEROX 820 ☐ GAMES ☐ COMPILERS ☐ UTILITIES ☐ BUSINESS ☐ ACCOUNTING ☐ SPREAD SHEETS ☐ MEDICAL ☐ DENTAL ☐ PAYROLL ☐ GENERAL LEDGER

RENT SOFTWARE BEFORE YOU BUY!

from our

SOFTWARE RENTAL LIBRARY

You can now RENT the most popular software available for just 20-25%* of Manufacturers' Retail Price

- Eliminate the risk—rent first!
- 100% of rental fee applies toward purchase
- All purchases are 20% Off of Manufacturer's Suggested List
- Rentals are for 7-days (plus 3 days grace for return shipping)

SPECIAL INTRODUCTORY OFFER

There are now 2 different plans to choose from:

Join the **Game Group,** $25.00 per year and rent as many games as you like for only 20-25% of Mfrs. Sugg. Retail Price.* Minimum order, 3 game rentals	Join the **Business Group,** $50.00 per year and rent as many business application programs as you like for only 20-25% of Mfrs. Sugg. Retail Price.*

REMEMBER, THESE ARE NOT DEMOS, BUT ORIGINAL UNRESTRICTED SOFTWARE PROGRAMS

(complete with manuals in original manufacturers' packages)

To Immediately Order, or for more information:

UNITED COMPUTER CORP.
Software Rental Library
Culver City, California

Toll Free CALL 1-800 992-7777
In California CALL 1-800 992-8888
In L.A. County CALL 1-213 823-4400

Canadian Orders Welcome

MasterCard AMERICAN EXPRESS VISA

*Plus postage and handling.

☐ EDUCATIONAL ☐ ACCOUNTS RECEIVABLE ☐ WORD PROCESSORS ☐ DATA BASES ☐ LANGUAGES ☐ GRAPHICS ☐

micro REVOLUTION in higher education

by Steve Birchall

"All entering students at the Monadnock College of Arts and Sciences are required to buy a computer." Statements similar to this soon may appear in every college catalog. Other schools may include the computer in the tuition and give it to the student with his diploma. Higher education is beginning to wake up to the microcomputer revolution in powerful ways. Many schools have operated extensive campus networks with workstations and time sharing, using a minicomputer or mainframe. What they have discovered is that the terminals create long lines of students and faculty waiting to use them. Worse, once a scholar starts to use a terminal, he is confronted with delays from the electronic traffic jam in the controlling computer.

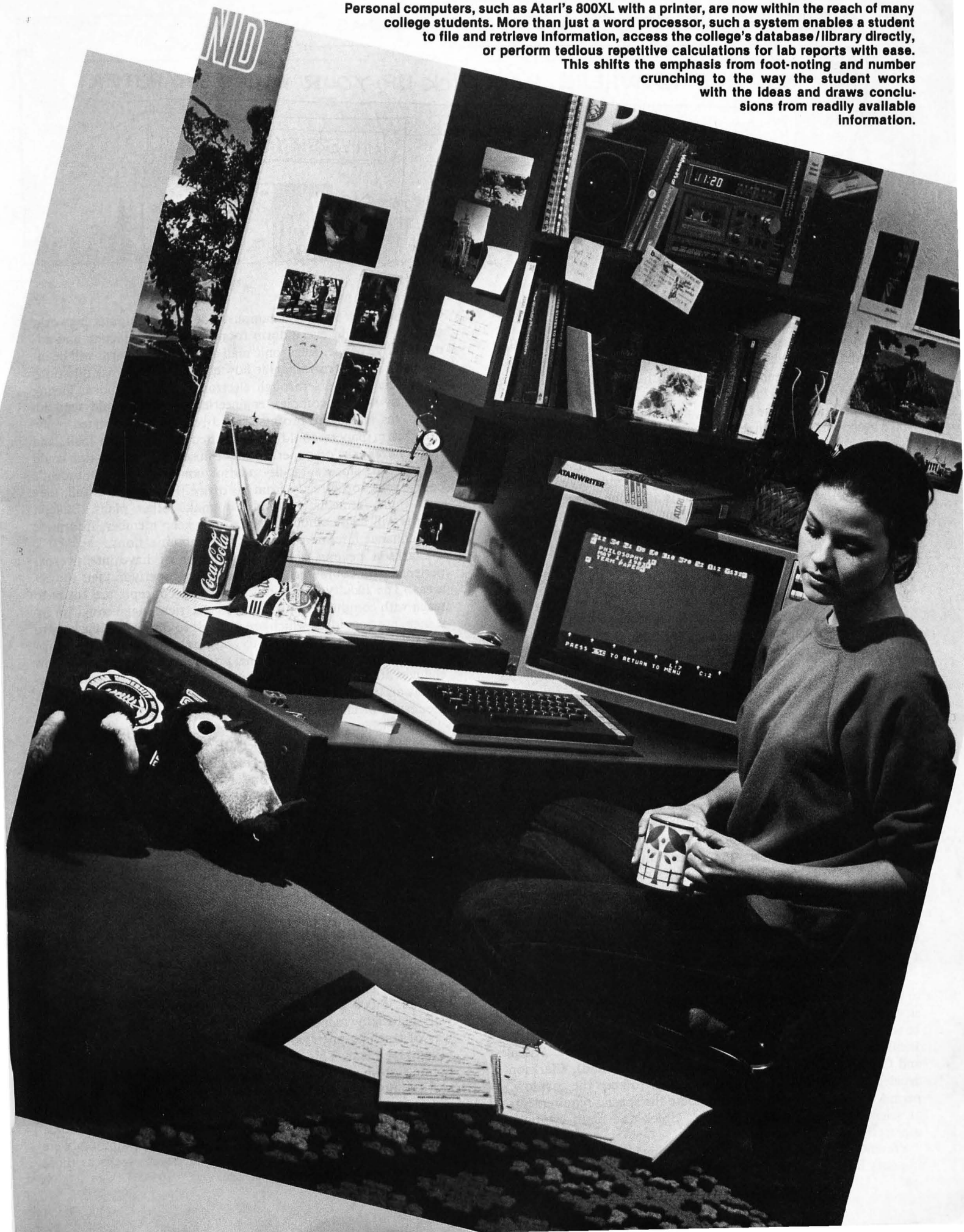
Personal computers solve these problems beautifully. Users can plug them into the network when necessary, or use them as freestanding units. Among the schools which now provide, or require the purchase of a microcomputer are Brown, Drexel and Carnegie-Mellon Universities, Stevens Institute, and Clarkson College. Hamline College in St. Paul, and Baltimore's Goucher College now require entering freshmen to pass a computer literacy test prior to graduation. Many other schools, such as the Rochester Institute of Technology sell microcomputers at a discount. In addition, DEC and IBM are donating large numbers of micros to the Massachusetts Institute of Technology for a far-reaching curriculum development program.

Micros are changing the campus through voluntary ownership as well. A few students have been able to afford them and have used them principally for word processing and computation. Two new developments have made micros much more desirable. The portable, battery operated computers (Radio Shack's Model 100, NEC's PC-8200, Casio's FP-200, Epson's QX-20, and Texas Instruments' CC 40) are ideal for taking notes, working on term papers over coffee in the student union between classes, and so on. In stationary equipment, Coleco's Adam, and Atari's 600XL Word Processing System are, at $600, inexpensive enough that substantial numbers of students will be taking them back to campus this fall — and will use them to great advantage.

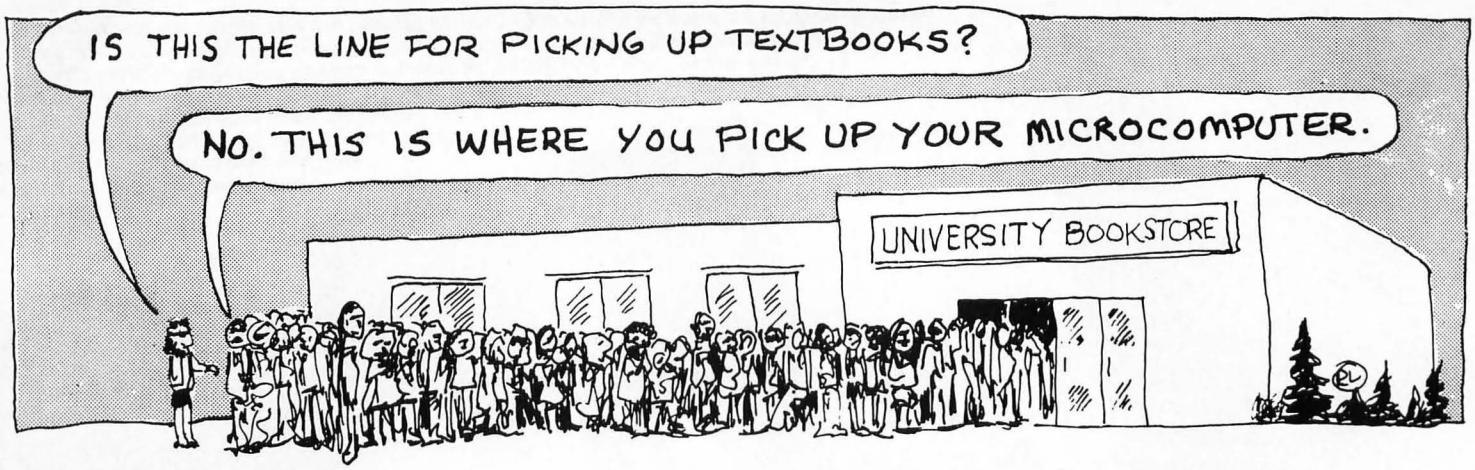
Personal Tool Kits

At Stevens Institute of Technology in Hoboken, NJ, the 80 freshmen in Science and in Systems Planning entering last September were required to purchase an Atari 800. Because of the program's immediately obvious success, the faculty quickly extended the requirement, and upgraded the hardware. In September, 1983, all 500 entering students must purchase a DEC 325. Through special arrangements with DEC, Stevens students will pay only $1800 for the $4500 computer systems, which include DEC's operating system, a word processor, FORTRAN and BASIC.

The Stevens program began in 1978 with a grant from the Sloan Foundation to study ways of integrating computers into the school's curriculum. The faculty found that using computers in Mathematical Analysis I and II, the basic freshman math course, they were able to cover more topics, and treat them in greater depth, than previously. The approach is to lead the student *to* develop his own utilities ("tool kit") for solving the problems he will encounter, such as differential equations and algebraic



Personal computers, such as Atari's 800XL with a printer, are now within the reach of many college students. More than just a word processor, such a system enables a student to file and retrieve information, access the college's database/library directly, or perform tedious repetitive calculations for lab reports with ease. This shifts the emphasis from foot-noting and number crunching to the way the student works with the ideas and draws conclusions from readily available information.



Micro Revolution, ***continued***

matrices. Previously, differential equations were not discussed until the sophomore year, but now they can be introduced in the first semester. As students move through the curriculum, they write and use hundreds of these personal utilities. Each acquires a unique collection specifically related to his own needs, interests and area(s) of specialization.

In preparation for the fall term, a team of faculty and students have been working through the summer to create course materials for the freshman class. According to Dr. Joseph Moeller, Associate Dean for Educational Development, this has led to closer relations between faculty and students, and they expect it to continue in the fall. Human contact is at a higher level, because the computer does the work of grinding out answers. It can provide hundreds of solutions to a problem, using different inputs. What is left is the real problem of choosing the best solution for a *given set* of circumstances — and that problem requires human insight and judgement. This is what leads students to go to the faculty and ask questions. Making choices, reasoning with facts, understanding how a more experienced person deals with the situation — those have been the goals of education for centuries. Micros are giving students and professors more and better reasons to discuss ideas together. "We think [this contribution of microcomputers] is important and positive," Moeller said.

Familiarity with computers and what they can do "improves the student's preparation for professional practice, because the computer plays a very important role in the daily activities they will be involved in," according to Moeller. Stevens has a work/study program, and students take their computers with them during their work terms. The advantages to both employer and trainee are obvious. After graduating, the student has a backlog of experience few competitors for jobs can claim. Perhaps in the near future, corporate interviewers will ask college seniors to include their software libraries in their applications, as evidence of proficiency. Starting salaries might be a function of the potential usefulness of a student/computer/software system to the particular company.

Obtain Advisor's Signature and Proceed to Computer Pick Up Line

Clarkson College, in Potsdam, NY, will present a Zenith Z100 to each freshman at Registration on August 25. These computers, along with the software package, are worth about $4000, and will be sold to the students for $1800, at the rate of $200 per semester. Included are CP/M-88, ZDOS, ZBASIC, FORTRAN, Pascal and Clarkson's own word processor (called Galahad). Clarkson decided to develop its own word processor to keep the cost low, permit expansion, and to accommodate the special requirements of science and engineering, such as Greek and math symbols, superscripts and subscripts.

Presently, they have over 200 terminals on campus, and students have to stand in line to use them. Within two years, Clarkson plans to have a campus-wide microcomputer network, with an outlet in every dorm room, classroom, faculty and administrative office. Electronic mail and bulletin boards will be included in the system. Imagine how easy it will be to ask questions and receive answers through electronic mail. A student having difficulty with a complex engineering problem could send his work "as is" to the professor, and receive an annotated reply. Most term papers could be submitted by electronic mail, and might never exist on paper. Even the miseries of registration could be avoided, since a faculty advisor could transmit a student's course selection directly from his office. If a section is full, they will know immediately, and can make other plans. Bulletin boards will help campus organizations keep members informed, and enhance internal departmental communications.

Clarkson believes that its students must have intensive experience with computers to be prepared properly in their subject areas. The faculty has found that many concepts are easier to teach with computer simulations. In electrical engineering, for instance, abstract ideas which are difficult to draw on a blackboard or explain in words, are easy to grasp because the computer provides a dynamic model of the principle, and can show how changes affect its operation. The advantage is that the student acquires an intuitive feeling for how a process works when he sees the results of changes.

For teaching stress factors in cantilever design, the faculty wrote a teaching program called BEAM. Traditionally, students spend hours calculating and recalculating stress on a cantilever for different weights and locations. Now, with an interactive computer simulation, they can try different loads and positions, and see the beam bend and break on the screen. What was formerly a boring, time-consuming, tedious process, has become a fascinating design problem. The time needed to teach it is shorter, and motivation to learn is greater. David Bray, Dean of Educational Computing, has had so many inquiries about the Clarkson College program that he now publishes a newsletter.

Micros in the Liberal Arts

Perhaps the most highly developed program to apply personal computers to higher education is at Brown University in Providence, RI. They have been at work for several years on their plan to have 10,000 microcomputers on campus. In 1975, under a National Endowment for the Humanities grant, they developed a way of teaching poetry on a large, multiple-user word processor called Hypertext. This enabled dozens of people to edit and annotate a poem *simultaneously.* One benefit was engendering more participation — particularly from students too shy to speak up in class. In Engineering, they have used seventeen Apollo computers, and Apollo is donating an additional 30 workstations, worth more than $600,000. The students using computers were able to cover the same course material in seven weeks as those without Apollos did in fourteen weeks.

Most recently, Brown and Apple have announced a plan to provide over $2,000,000 worth of equipment to students and faculty during the next three years. By 1989, 10,000 personal computers — one for every student, faculty member and administrator — will be in operation, at a cost of over $70 million. Each department will have its own network, with a Lisa in control. The departmental networks will be linked to BRUNET, the campus-wide coaxial cable network, which is already installed and in operation. IBM's newest and latest mainframe, the 3081, which operates the network, will be obsolete within two years.

Apple is donating 50 Lisas to Brown. In preparation, the University is taking a Lisa to each department to show its capabilities and explore that department's potential uses. Theatre Arts will use an interactive videodisk system to teach many aspects of theatrical production such as directing, set, costume, and makeup design, and lighting. By seeing how changes in stage directions can affect the impact of a scene, students can learn without needing live actors to demonstrate it for them. They can select a specific combination of stage directions, lighting, and set design, see the results, and try it all over again, and again. With an annotated script from previous productions, or productions from other campuses, the system will grow and become more useful over the years. In the languages, Lisa's word processing and character generation capabilities will enable scholars to work in the original language and alphabet, on screen, even in Sanskrit, Hebrew, or Chinese.

In music, they will go beyond the usual music theory application, and use the analog of a word processor to analyze compositions for style (chord choice, note-strings, rhythmic patterns), thematic development, and form. Lejaren Hiller's pioneering work in the Fifties on the application of Information Theory to musical analysis should be particularly useful. Not only does it find statistically significant note-strings, but it points out their frequency of usage and locates passages where they are either common or rare. If Brown chose to use digital recordings, they could teach performance and interpretation in the same way the theatre department plans to teach staging. With digital audio editing software (such as Soundstream has developed for their production facilities in Los Angeles and Salt Lake City), they could demonstrate the effect of changing the phrasing, articulation, tempo or dynamics by inserting various alternative passages into the playback of a recording. Such a system would be especially useful in the study of early music and its performance practices.

The science departments like the idea of using the personal computer as a lab notebook for gathering data and comments, rather than just as a number cruncher. Another use is computer simulation of processes, which can demonstrate to the student the full range of possibilities in a shorter period of time. Lab space and competent lab instruction are expensive, so demonstrating physical and chemical processes by computer means better instruction for non-majors. The Medical School is located in eight different hospitals, so communication sometimes is difficult. Electronic mail and bulletin boards will take care of most of those problems. Simulations of illnesses will enable medical students to practice diagnoses and study the effects of treatments.

Simulations in the Humanities

Drexel University in Philadelphia will purchase 3,000 new model Apple computers. About half will be sold to freshmen entering in January, 1984, at a price of $1,000. The rest will go to upperclassmen, faculty, administration and departments. Although neither Apple nor Drexel will confirm it, these very likely will be the long-awaited MacIntosh, a less expensive version of the Lisa. The published description includes a sixteen bit *micro*processor, 64K RAM, built in software with user friendly Lisa-like technology, and integral video screen, keyboard and disk drive.

Micro Revolution, ***continued***

Students in all departments (science and technology, business administration, and the Nesbitt School of Design, Nutrition, and Human Behavior and Development) will use the computers. Business and science applications are fairly evident, but consider what a timesaver computer graphics are for a student in Fashion Design. The History Department will use simulations to show how changes in events might affect the outcome of history. What might have happened if Lincoln had lived? What if the attack on Pearl Harbor had been detected in time? Hari Seldon and the Second Foundationers would love this application. In the English Department, several faculty members have been working on a concordance for the works of Milton.

Printing and Photography

The Rochester Institute of Technology has been using computers actively throughout the school for several years. Although they sell DEC personal computers to students and faculty at a substantial discount through the bookstore, they do not require them. One problem has been the lack of appropriate software for the DECs. On the other hand, RIT has a DEC GIGI system, with over 300 color graphics terminals on campus. Several courses in computer graphics as related to design, printing and photography already are in the curriculum. Like most institutions, RIT's interest is not so much in computer assisted instruction (CAI), as in using the computer as a problem solver and a design aid (CAD). Right now, RIT is placing micros in the hands of faculty who wish to develop specific software for their courses.

Peanuts in Pittsburgh

Carnegie-Mellon University in Pittsburgh is working with IBM to develop a special computer for their campus. The program will begin in September, 1983 and eventually will become a campus-wide network of personal computers, all capable of talking to each other. By 1985, every student will be required to buy one. The one hundred machines delivered for the 1983 fall term will be transitional models based on the Motorola 68000 sixteen bit chip. As IBM learns about the specific needs of the school, it will design the final product.

Human Factors at RPI

At the Renssalear Polytechnic Institute in Troy, NY, a decidedly high tech school, with an elaborate and well-used campus network, they began a five year study on personal computers. The first year, twenty students had Atari 800s; the second year, twenty more had IBM PCs. Control groups were established, bringing the total to 120. Because they have acquired so much data, RPI will not continue the project. Conducted by Dr. Linnda Caporael of the Psychology Department, the study is especially important because it delved into many of the human and social factors of placing personal computers on campus. For starters, possessing a micro "most definitely did not have any impact on grades," according to Dr. James Moss, Director of Computing. Part of the reason for this may be that RPI has an extensive network of terminals on campus, so any student who wanted to work on a computer had an opportunity.

Obviously, students spent a lot of time playing games on the micros (sophomores even more than freshmen), but with no negative effects on grades. The dorm rooms with personal computers became popular places, at times almost turning into arcades, which was a problem for the students living in those rooms. Another problem was space — two systems in one room was too much, and many students compromised by *sharing the* use of one computer, and putting the other one under a bed. To their surprise, the RPI researchers found that even with access to the campus network through the phone lines in the rooms, many

continued on page 32

GREAT PRODUCTS for GREAT COMPUTERS

BOOKS

ATARI BASIC, Learning by Using. This is an action book. You program with it more than you read it. You use it, you discover with it. Learn ATARI BASIC easily through the short programs provided. A great source of work problems for teacher or student. 73 pages.
ISBN 3-92-1682-86-X $5.95

Games For The ATARI. Provides ideas on how to create your own computer games. Contained primarily BASIC examples but, for very advanced programmers, a machine language example is included at the end of the book. 115 pages.
ISBN 3-911682-84-3 $7.95

How to Program Your ATARI In 6502 Machine Language. To teach the novice computer user machine language, the use of an assembler, and how to call subroutines from the BASIC interpreter. 106 pages.
ISBN 3-921682-97-5 $9.95

FORTH on the ATARI. Explore this versatile programming language with numerous graphics and sound examples. Designed for both the novice and experienced programmer. 118 pages.
ISBN 3-88963-170-3 $7.95

ATARI BASIC Faster and Better. Programming tricks and techniques. Three companion software diskettes available (sold separately).
280 pages, ISBN 0-936200-29-4
$29.95

SECRETS OF ATARI I/O. Theory of operation and application programs for input/output to disk, screen, cassette, and RS232 serial port. Machine language with POKE tables for use with BASIC programs. Companion software available on disk (sold separately). 285 pages.
ISBN 0-936200-33-2
$29.95 retail

SOFTWARE

CASDUP 1.0 & 2.0. To copy most BOOT tapes and cassette data files. 1.0 is a file copier. 2.0 is a sector copier. Cassette only $24.95

CASDIS. To transfer most BOOT tapes and cassette data files to disk. Disk only $24.95

FULMAP. BASIC Utility Package. VMAP-variable cross-reference, CMAP-constant cross-reference (includes indirect address references), LMAP-line number cross-reference, FMAP-all of the above. Will list "unlistable" programs. Also works with Editor/Assembler cartridge to allow editing of string packed machine language subroutines. All outputs may be dumped to printer. Cassette or disk $39.95

DISASM. To disassemble machine language programs. Works with or without Editor/Assembler cartridge. May be used to up or down load single boot files. All output can be dumped to printer. Cassette or Disk $24.95

DISDUP. For disk sector information copying. May specify single sector, range of sectors, or all. Copies may be made without read verify. Disk $24.95

V-COS Cassette Operating System. Control baud rate, leader time, screen width, background and letter color, cassette motor (on/off); provides cassette file verification. Cassette $24.95

DOWNLD Diskette Download Utility. Allows single BOOT files and Binary DOS files to be transferred from disk to cassette. Fast, easy, menu-driven. NOT FOR PROTECTED SOFTWARE. Disk $19.95

DISKPAK. A program that frees the unused sectors on a boot disk for storage of normal DOS files without disturbing the boot file. May be used on all boot files including multi-stage files. NOT FOR PROTECTED SOFTWARE. Disk $24.95

ABFAB Assembly Disk Companion software to ATARI BASIC Faster and Better. Ten assembly language source programs and ten object programs. Disk $19.95

ABFAB Library Disk. 81 subroutines that can be included in your BASIC programs. Includes BASIC and machine language (some programs POKEd into memory). Disk $19.95

ABFAB Demo/Applications Disk. Eleven application programs and fourteen demonstration programs from the ATARI BASIC Faster and Better book. Disk $19.95

SECRET Library Disk for the ATARI. More than a dozen I/O routines that exemplify material in SECRETS OF ATARI I/O (sold separately). Includes Super Menu, Screen Dump, BASIC AutoRUN, Binary Loader, String Search, Disk Copier, Cassette Copier and much, much more. Disk $19.95

IJG products are available at computer stores, B. Dalton Booksellers and independent dealers around the world.

If IJG products are not available from your local dealer, order direct. Include $4.00 for shipping and handling per item. Foreign residents add $11.00 plus purchase price per item. U.S. funds only, please.

IJG, Inc. 1953 W. 11th Street
Upland, California 91786
Phone: 714/946-5805

IJG

ATARI TM Warner Communications, Inc.

SOFTSIDE ORDERING INFORMATION

For Payment In The USA
VISA, MasterCard, certified checks, money orders and personal checks are accepted.

For Payment In Canada/Mexico
The preferred method of payment is by VISA or MasterCard. A bank check is acceptable if it has been preprinted for payment in U.S. dollars. No personal or company checks accepted.

For Payment On 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 MasterCard.

GUARANTEE
All software is guaranteed to load and run. If you experience difficulties with the product within 30 days, it may be returned for replacement. Send your properly protected tape or disk to the attention of the Customer Service Representative and include your name, address, and the reason it is being returned.

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 or operation of such software.

PRICES
Prices are subject to change without notice. We are not responsible for typographical errors.
Unless otherwise noted in a published advertisement, the following prices are in effect as of this issue:

	USA/Canada APO/FPO	USA/Canada FIRST CLASS Mexico	Other Foreign
SoftSide Magazine* (yr)	$30	$40	$62
SoftSide Magazine* (6 mo.)	$18	$23	$34

	USA APO/FPO	Mexico Canada	Other Foreign
CV (year) & magazine	$99	$119	$149
(6 mo.)	$55	$65	$80
DV (year) & magazine	$149	$169	$199
(6 mo.)	$79	$89	$104

***Includes SoftSide Selections.**

BACK ISSUES
Minimum order — $10. Price includes shipping to the 48 states only. Alaska, Hawaii, Puerto Rico, APO/FPO, and ALL foreign orders — postage is additional.
All Foreign ordersand all magazine/media combination orders — Order directly: **SoftSide, 10 Northern Blvd., Amherst, NH 03031.**

Micro Revolution, ***continued***

students still preferred to use the regular terminals. The reasons given were mainly technical — convenience in operating the system on the terminal intended for it, and so on. But the real reason turned out to be that room mates became upset when the phone line was tied up for long periods of time. The answer to that is to wire in an extra set of data lines for every dorm room, which is what RPI is doing now.

As a result of the study, RPI's faculty has "decided unequivocally that it is wrong to require all students to buy personal computers at this time," according to Dr. Moss. The students agree, but add that they should have access to one. About 300-400 students now have their own, and the school has installed clusters of microcomputers connected to the network in all the dorms. Unlike many similar schools, RPI does not at present sell computers through its bookstore, but is considering doing so. The most reliable predictor as to whether a student will bring his own micro to campus is whether his parents work with a computer professionally. Personality types seem to be irrelevant, although those who own computers also tend to be high energy people who are active in all phases of campus politics and activities, rather than introverts who spend all their time at the keyboard.

Revolution in Thinking

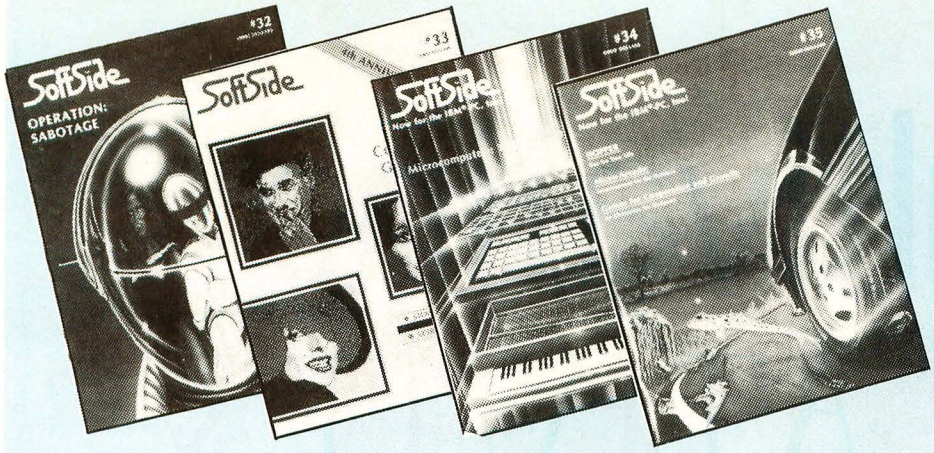
What a student has accumulated after four or five years of study (and off-campus work experience) is a substantial personal library of applications programs for solving problems and doing routine work in his field. If the student also owns the computer all this runs on, he takes with him a powerful array of skills and abilities when he goes looking for his first job. Those students graduating from schools with mainframes and minicomputers, and large networks, may have the same experience and expertise, but they can't take their personalized software and computers with them. The combination of a student who owns a micro, which he thoroughly understands and knows how to operate, along with a specialized repertoire of software he has developed for his own use is a powerfully synergistic man/machine system. We surely will see large numbers of new companies with innovative products and services started by graduates from these programs — particularly in the areas surrounding these colleges. This outpouring of creative energy will change our lives in ways we cannot now predict.

After going through a program which emphasizes problem solving through computer simulation techniques, a student has a far different outlook on things. The impact this new generation of graduates will have on existing American business and technology will be significant — and no less important than the new companies these people will spawn. For example, what new levels of design maturity will they bring to consumer products developed on CAD/CAM systems? They will not be preoccupied with the *process* of getting an answer. To them, that will be boring — drudge-work suitable for a computer. They will be concerned with *evaluating* all the possible outcomes through a computer model of the problem, and *optimizing* the variables to get the *most desirable results.* The difference in point of view is as great as the difference between learning to read and write, and knowing how to write a novel.

This new generation of "novelists" (i.e. problem solvers) will challenge and even threaten all the old hands who still worry about how to solve problems, because they will not regard that level of thinking as productive, and will want to try many different solutions before selecting the best one — a difficult situation for someone who has spent years simply finding one solution that works.

In three or four years, an army of people trained to use micros in their major subjects will enter the work force. At first, most of them will be scientists and engineers, but more and more will be from the liberal arts, fine arts, and professions. What the results will be is nearly impossible to predict, but we can be certain they will affect our lives significantly. Moreover, large numbers of young people graduating from college with a computer and extensive knowledge about its use will become avid consumers of much more computer-related technology, such as videotex services, networks, interactive videodisks, and of course increasingly sophisticated computer hardware and software. As the economic effects domino their way through society, great numbers of new jobs, and large amounts of new capital, will be created.

Educators have always maintained that the schools must be responsive to the needs of society, and must change their curricula to meet changing needs. As Saul Bernstein points out in *A Crisis In Education* (page 22), the little red school house, short days, long summer vacations, and locally elected school boards met the needs of an agrarian society in the Eighteenth and Nineteenth centuries. If the schools are to survive, they must change their programs significantly to adapt to the rapid changes we are now experiencing. The rate of change is much swifter. We don't have 50 to 100 years to think about it this time around. If we take as long as five years to begin to accommodate these changes, other countries may overtake us. They will have the healthy, growing economies, not us. The Micro-Revolution on campus is just beginning.



SoftSide Back Issues...

Once They're Gone...Are They Gone Forever?

If you like the programs, reviews, and programming information in this issue of *SoftSide*... think of what's waiting for you in past issues!

Exciting Games Like:

- **Defense**
- **Micro Man**
- **Hopper**

Great Graphics Like

- **Shape Wizard**
- **Titan**
- **Flight of the Bumblebee**

PLUS

- **Databases**
- **Utility Programs**
- **Educational Programs and more!**

It's all here in the Back Issues of *SoftSide* magazine. And many of these issues are still available for your enjoyment. But not all. Several issues are SOLD OUT...others are available but supply is limited.

So check out the issues and features listed below and order today!

Back Issues ..... $5.00 ea.*
Back Issue on cassette ..... $12.95 ea.*
Back Issue w/ Enhanced Disk Version (contains an additional program) ..... $19.95 ea.*

To Order

Use the bind-in card on the facing page. Send order card and payment to:
***SoftSide* Magazine, 10 Northern Blvd., Northwood Executive Park, Amherst, NH 03031 (minimum order - $10.00.)**

*Prices good for USA orders only — for foreign pricing see page 32.

February 1982 · Space Rescue · Rubicube · Defense · Maze Sweep ·
Enhanced Disk Versions
Andorra · Kismet II · Help Package ·

March 1982 · Hexapawn · Magical Shape Machine · Outer Space Attack · Killer Cars ·
Enhanced Disk Versions
PEEKER/POKER · Curse of the Pharaoh · Warpath ·

April 1982 · Microtext · Poster Maker · Atari' Banner Machine · Database ·
Enhanced Disk Versions
Semaphore · Renumbering for the Atari' · Screen Print ·

May 1982 · Solitaire · Micro-Man · Cross Reference · Ladders ·
Enhanced Disk Versions
List Formatter · Robot Battle · Breakthru ·

#30 · Escape from the Dungeons of the Gods · SWAT · Code Breaker · Sabotage · Piazza Hotel ·
Enhanced Disk Versions
Dr. Livingston · Random Access Database · Kriegspiel II ·

#31 · C.A.T.S. · Puzzle Jumble · Tunein · Dots ·
Enhanced Disk Versions
Applesoft Extensions · Menu Plus · Starbase Gunner ·

#32 · Operation Sabotage · C.A.T.S. Computer Assisted Testing System Part II & III ·
Enhanced Disk Versions
Cloze Test · Neat List · Up Periscope ·

#33 · Shape Wizard · SpiralGraphics · Graphic Writer ·
Enhanced Disk Versions
Hi-Res Character Generator · Paranoia · ILIST ·

#34 · Sabotage · Pokey Player · PC Blues Box · Tunein ·
Enhanced Disk Versions
Auto Menu · The Rotberg Synthesizer · Cavern Quest ·

#35 · Hopper · Blockade · Saucer Formation · Fireman ·
Enhanced Disk Versions
Sub Hunt · Deadstick Landing · Ping Pong/Hockey ·

#36 · Atlantis · Pokey Player II · Space Fire ·
Enhanced Disk Versions
Fortress · Atari Fig-FORTH · APL-80 ·

#37 · Deluxe Personal Finance · Blackjack · Car Race · Garage Sale Records ·
Enhanced Disk Versions
Applesoft Extensions · Disk Peeker/Poker · Turret and Track ·

#38 · Deluxe Personal Finance · Concentration · Pokey Player III · Word Search Puzzle Generator · Squares A Deux ·
Enhanced Disk Versions
Super Quest · Human Errors · ST80-DUC'M ·

#39 Flip-It II · Kangarilla · Trapped · Sort Routines for Atari Database · Page Up, Page Down · Savo Island ·
Enhanced Disk Versions
Space Fungus · Sucess · Force of Arms ·

#40 Mini-Golf · AppleMaze · Smokey · Escape From The Dungeon Of The Gods · Defense ·
Enhanced Disk Versions
Micro-Monopoly · Arachnid · Io Miner ·

#41 Poker Squares · Moving Marble Maze · Solitaire · The Memory Game ·
Enhanced Disk Versions
Apple Blaster · Five Card Stud Poker · Micro-Monopoly ·

#42 Microtext 2.0 · King II · Hopper · Football · Screamin' Demon ·
Enhanced Disk Versions
Life II · Photon Flyers · Family Tree Organizer ·

· Apple® · Atari® · IBM® PC · TRS-80®

Cultivating CREATIVITY

— How Computers Can Help

by Wes and Leslie Horlacher
In Collaboration with Susan Kenny

Few educators (or computer programmers) realize that they can design computer programs to have a capacity for simple creativity — to perform common creative tasks. *Originating* creative ideas still is a function of the human spirit, because the software can do only what the programmer designed it to do, in conjunction with the user's input. But computers can *automate* the *execution* of creative ideas (within the limitations of the software) and thus stimulate the user's creative inspiration by showing a range of possibilities an artist might not care to (or have the time) to carry out. Creative inspiration generates new and original ideas, while common creativity generates interesting new forms of existing ideas. Nearly everyone seeks true creative inspiration while actually enjoying only a limited sense of common creativity.

Ball State University music professor George Wolfe, in his article "Creative Computers — Do They 'Think?' " (*Music Educators Journal,* January 1983) makes two interesting points:

- Although human beings are the only source of true creative inspiration, computers are better suited for common creativity, since the required transformation skills involve high speed processing of complex forms and images.
- Modern society encourages people to develop common creativity while discouraging true creative inspiration.

Effective implementation of "creative programs" is a matter of devising complete and unique experiences for the user under program control. Complete experiences involve the user in a wide range of physical, intellectual, emotional and social experiences. Minimally, such programs should transform a given image into a new, logically structured, audio-visual form, which should have a unique interest to the user. To do this, the program must learn the user's particular interests by giving him subjective control over one or more aspects of the algorithm.

Melody Processing

In Wes Horlacher's program, *The Magic Melody Box,* published by the ATARI Program Exchange (APX), the user chooses a rhythm and draws a melody line. The computer uses these two simple inputs to create one of the thousands of possible tunes. The computer has not "composed" a melody, but has realized a specific example, based on the user's rhythmic and melodic specifications. After the user sees and hears the tune, he may either replay it for the enjoyment of everyone in the room, alter the specifications to improve it, or choose a new rhythm and melody line for a new song. In this way, he acquires practice in making aesthetic judgements. He can evaluate a series of slightly different versions of a melody in terms of their suitability for his specific expressive intent, and can reprogram until he is satisfied with the results. The real creativity and artistic sensibility reside in the human spirit, but the computer does all the tedious working out of ideas.

Experienced musicians who use this simple program are fascinated and noticeably entertained by it. But how do musical novices react? At first they just choose a rhythm and draw a melody line randomly. They soon notice that certain rhythms and melodic relationships are more appealing to them than others. The program becomes a game to find the tune they like best. The reward comes as a natural consequence of playing the game, and people describe successful results as emotionally satisfying. Clearly, the user is the driving creative force, as evidenced by an emotional attachment to his musical creations. People will even defend the merits of their work against the derogatory remarks of bystanders.

In this example, the user enjoys a truly unique experience of his own making. His experience stimulates him physically with desirable sights and sounds, challenges him mentally with interesting patterns and images, charges him emotionally with creative power and engages him socially by giving him title to a new and unique tune.

Bi-Level Creativity

Dividing creativity into two levels (a working-out process for computers, and an inspired or higher level for people) opens up an exhilarating view of learning and doing creative work. By programming computers with the basic rules and principles of common creativity, we human creators can enjoy the results of creative inspiration — even if we lack the specific training to realize our ideas. The quiet hope of applying computers to arts courses is to teach students about higher-level creative processes, and enable them to experience the joy and fulfillment of spontaneous creative expression.

Large companies, such as United Technologies in Hartford, CT, now sponsor in-house courses on using microcomputer software. Spreadsheets, business graphics, databases, and electronic mail are all covered in the workshops. The benefits to the company include increased productivity from managers, and a strong competitive edge. Over 1,100 UT executives will participate, and each will have an IBM PC at his desk. The program costs $5 million and will run through mid-1984.

Teaching Software

by Ame Choate Flynn with Michael Callery and Tom Flynn

This article represents several years of software consulting and teaching by the authors. Ame Flynn and Michael Callery designed the computer Art and Graphics courses at the New School for Social Research and have been teaching them for over two years. In addition, both operate their own businesses (TechniGraphics and Learningware, respectively). Tom Flynn has experienced courses similar to those described in this article as a student and has contributed his reactions and technical expertise.

In the beginning...

there was hardware, and people saw that it was good. But the hardware was too hard to use. The only people who could *use* personal computers were programmers and hobbyists. As recently as 1978 few application programs were available, except for some games, and Electric Pencil (a word processor). If you wanted to create a character set, you had to start from scratch. Most early applications software was poor. People who bought a microcomputer literally were stuck with the basics — BASIC if they were lucky.

When computer owners built their computers from scratch, they knew how their machines worked. They programmed the source code themselves, and took it from there. It was a very exclusive club — everyone was discovering new tricks to teach their computer and new ways to use this fancy new machine.

Things have changed. Many software packages are on the shelves, each performing a specific function. As application software got more complicated, it created a paradox of its own. Its availability is what attracted non-hobbyist computer users. After taking their computer and software home, they realized the enormity of the learning task ahead. Inadequate or inappropriate documentation compounded the problems of inflexible and user-grouchy programs. They needed help not only to *use* the computer, but also to *apply* it to their needs. This created a new industry — the Micro Consulting business. Not everyone who owns a computer can afford a consulting firm to help them wade through miles and miles of abstruse text. Users in the Eighties are not computer builders, and don't want to spend months learning about the innards of their computer.

Software for Micros

VisiCalc® was the first software complex enough to require instruction. To use it in any way requires extensive reading and preparation. In 1981 the new generation of less expensive home computers appeared. The high-end application software, such as the VisiCalc family, word processors and data base packages have become more expensive. VisiCalc is not only more expensive now, but much more complex. It has even spawned a family of related products, such as VisiSched®, VisiFile®, and VisiPlot®— all of which can interchange data in complex ways. The new spreadsheet programs such as Multiplan® are almost a programming language in themselves. People need these complex functions, but in the process of creating the software, the designers made it much more difficult to use.

In response to this, the philosophy implemented in the Xerox Star® and Apple® Lisa is to make computers and software so friendly that a novice can sit down and just use them. Whether a novice can use their full power in twenty minutes is debatable. Learning computers and software involves more than just knowing the commands. "User-friendly" doesn't mean *easy* to use, it just means *friendly* to use. You still have the difficult problems of data structure and formulas to solve and apply to your purpose. If you are doing statistical analysis or accounting, you need to know the subject area, whether you use a computer or do it by hand. Don't confuse your goals — if you want to learn to write, take a writing class, not a word processing class. If you want to learn accounting, don't take a VisiCalc course, study accounting.

Computer Literacy Through Software

A new topic in education is "computer literacy." A body of opinion holds that one way of achieving it is to teach people to use a word processor. Most students type their papers, so a word processor (an extended typewriter) becomes a point of tangency between a familiar activity and the world of computers. The process of learning a word processor teaches many basic computer skills, particularly utilizing the computer's operating system. You become literate on one particular system and, properly instructed, gain a generalized understanding of how computers work.

Teaching about computers by starting with the software works as well as a course in BASIC. Motivation often is higher because the practical application is more obvious and results more immediate. For instance, knowing how to use VisiCalc is more satisfying to a beginner than learning how to manipulate arrays in abstract ways. Beginners who have just bought a computer and software may encounter manuals, documentation, tutorials and sample problems that raise more questions than they answer. The frustrating search through mysterious mountains of manuals to find those answers costs time and money. If you are in business, you may find that doing a year's worth of home budgets in the VisiCalc tutorial does not hold your interest.

In our classes we give students as many prepared notes as possible. We may rewrite or reformat the documentation in more easily understood terms, or start completely from scratch. However, documentation is no substitute for working with the program itself. Students find they must "turn their minds around" to grasp some of the concepts. Precepts they have taken for granted for years now have different meanings. Teaching word processing to secretaries is an example. Since the invention of the typewriter, a carriage return signified the end of a line. Now, in most word processing programs it means the end of a paragraph. The software takes care of the sentence. That is a big conceptual difference, and only experimentation leads to learning it.

In graphics, working on a black screen instead of a white sheet of paper is a fundamental difference. Working in one area, with paddles or joystick, and having your work appear on a monitor may also take a while to get used to — not to mention minor items that loom larger (Is this the button on paddle one or two?). In teaching graphics, I find that the students can put a pixel on the screen easily. Acquiring a concept of the screen memory in which the pixel is placed is more difficult, but it is an important part of learning to operate the software.

The same kind of conceptual switch occurs when working with the spreadsheet and investment packages. The student must understand their dynamic, interactive nature. Changing one item may cause changes all through the model. This is what makes computers powerful tools — but the concept still has to to sink in. The working medium is different.

Individual Differences

In our programming classes, finding a level on which to begin is difficult. Every student has a different level of expertise. The more technical the programming language or software package, the more difficulty we experience finding the proper level. Even accomplished computer users find that switching between many packages or between different areas of programming creates confusion. If you can't keep the commands straight, the timesaving features of a software package may get lost in the mist of time. Many computer buyers have very short-term goals. Their aim is not to get a computer and then proceed to write a program that will make them a lot of money. They buy the computer to help them handle their business right now, not at some vague time in the future. They want to know how to operate the computer and use the software. Refresher courses eventually will be as common as typing refresher courses are now.

Computer literacy courses vary from no hands-on, to part hands-on. You may have to learn esoteric (and ultimately irrelevant) protocols if the course is taught on a multi-terminal main frame system. Before enrolling, ask how the course is structured. How many class meetings are scheduled, and how much computer time will you get? Is the material covered in a realistic number of classes?

When we started out with our Apples and graphics packages, everything was trial and error. Some peripherals, such as the

↪

The MICRO TRENDS Computer Festival

by Carolyn Nolan

The MICRO TRENDS Computer Festival and Exposition for the Future will take place October 20-23, 1983, at Fort Mason Center in San Francisco, CA. The concept grew out of extensive surveys of the business, professional, educational and technical communities which expressed an urgent need to access the mass of information about the microcomputer industry without the inevitable barrier of high-tech jargon.

Planned exhibits include a hands-on center where visitors will explore a microcosm of the microcomputer world at a variety of terminals in a casual setting. In Issue 42 we described one of the Festival modules — the Micro Cottage. Another module, The Micro Classroom, will serve as a model to educators who attend the Festival in search of ways to implement computer programs in their schools. A Corvus OmniNet system will connect networks of Apple IIs in a simulated classroom scene where rotating groups of selected students will show visitors how technology can enhance elementary and secondary education.

The Micro Classroom will feature a raft of educational software aimed at the stereotype of the computer as an electronic drill instructor. The PLATO software series (Spanish, German, French, computer literacy, physics-elementary mechanics) emphasizes life-coping skills: building relationships, forming positive behavior patterns and increasing self esteem and communications skills.

The Festival will offer other interesting software such as "Raise the Flag," an alternative to "Hangman" by Joyce Hakansson. Her programs are marketed by Milton Bradley, by Apple (under their Discovery System) and by Spinnaker. Her educational software philosophy stresses cooperation and growth and reflects the entire Festival approach to computers in contemporary life. Students will write their assignments with "BankStreet Writer" (Broderbund), study the stars with the "Tell Star" (Information Unlimited SoftWare) astronomy program and do historical simulations in which they cooperatively build towns and set up the social and political systems.

The Micro Classroom will present educators with an impressive array of software demonstrations, and allow them to observe students of every age level working with computer languages (BASIC, Logo and Pilot). They also can participate in seminars where educators and industry spokesman will discuss computers in programs for the gifted, and for the developmentally disabled student.

The Festival will offer introductory and advanced seminars on word processing, the Visi-Series, database management and selecting hardware and software. A special Hands-on Center with 60 terminals and a capacity of 200 people will serve as a laboratory where visitors can investigate the technology and ask questions at their leisure. The general atmosphere of cooperation and understanding together with the agenda of speakers which includes, among others, Alvin Toffler (*The Third Wave*) and Barbara Marx Hubbard (*The Electronic Cottage*) should provide excitement and exhilaration for everyone.

Teaching Software, ***continued***

graphics tablet, were simple to learn. However, we had to learn and understand all the tablet's protocols. In the process, we spent a lot of money on phone calls to find out why a particular package would not fill colors on graphics created with the tablet.

In classes on the IBM PC, we presented a Graphics Sketch program. A student later tried to run it on one of the systems at work, and got "Illegal Function Call." The answer to the problem was that the work system had a monochrome adaptor and the software expected a color adaptor. She spent her time learning an IBM idiosyncrasy, instead of basic computer concepts.

In a Microprocessor Assembly Language course, the prerequisite was knowledge of at least one programming language. Even though the attendees fulfilled the requirements, at least 50 per cent had no idea what was going on. Their background was playing games and writing simple functions, but with no understanding of how computers work with those programs. "Hex" was a mystery to them, and they fell behind. Most were taking the course to find out the *one* thing they didn't understand about the processor. Even in books that explain machine language commands in detail, some concepts don't get through.

Computer Graphics at the New School

"Art and Graphics with a Small Computer" is an introductory, prerequisite course in which we survey all aspects of computer graphics in 36 hours. Other courses cover programming for graphics, advanced applications and animation. We teach the "Complete Graphics System" at the entry level in the first course, but later explore the concepts in greater depth and show the students how to adapt it for their own applications. We have also developed many of our own tools and utilities — timesavers that have taken us months or years to develop — and pass them on to our students. Everyone enrolled in the graphics classes leaves with tangible products. We require a project from all students taking the course for credit, and strongly suggest that *all* students produce one.

For those who require only a brief introduction to computer graphics, we offer the "Graphics MicroMarathon." In this one-day, eight-hour course, we talk about computers and graphics screens in the morning seminar. The afternoon is all "hands" on," using software written specifically for this class. The participants gain a feeling for the power of the machine when they use new ideas which were theory only a few hours ago. If more than one teacher is available for the class, one may be at the board explaining a function while the others stand in the back or middle of the class, watching the students and helping those having difficulty.

Practical Problems

In the regular computer graphics classes, we use several software packages rather than one all-encompassing piece of software. A persistent problem is that each piece of software and the various peripherals (keyboard, tablet, light pen, paddles and joysticks) use different commands. The commands vary from one to another. Worse, the *results* from the *same command* may vary. One package expects a PIC. before the picture title, another a .PIC after the title, and a third doesn't care. Graphics are not as portable as business software because the graphics capabilities differ from system to system. *Micro-Painter* on the Atari, for example, takes advantage of the luminance controls. Atari's Player Missile Graphics are similar to block moves on the Apple, but not the same. If manufacturers expect sales to increase, they must establish standards of compatability.

Protected software is another problem. Our students are infamous for destroying system masters when initializing disks. If we were not able to make copies of disks, we would have none left. Some software we would like to teach is impractical because

it is protected. We keep the software under control to prevent pirating, but we must be able to keep the master separate from the disks that are in day to day use. A nice fat fingerprint on a recorded surface destroys a disk. If we had to wait two to three weeks for another copy, the class would be over before we could cover that subject.

"Help"

Frequently students comment, "When I come in to work alone and something unusual happens, I don't know what to do. When the instructor is here, he hits a few keys and suddenly I'm back where I wanted to be. Do you know any books that can tell me these things?" The answer is, "No. Experience has shown me how to recover like that. The only way you are going to learn is to practice and listen to the instructor." When you take a class you are paying for the teacher's expertise and ability to cut through the verbiage contained in the documentation. In a class, you have the advantages of the teacher's experience with the software, and knowing what to use in particular situations — better and cheaper than trial and error buying.

One of the smallest and most important things taught in the graphics class is how to listen to your disk drive — is it loading information, saving information or just spinning its wheels? Although the image of a classroom full of people with their ears pressed to their disk drives seems a little silly, understanding what is going on inside is beneficial. Trying to describe these sounds in a book is extremely difficult.

We cover some aspects of BASIC to help students with their projects, along with Logo and CEEMAC, the language of Fire Organ. These languages are not the focal point of the courses, so we attempt nothing more than an introduction. Even so, a student recently created a project using Logo. What we emphasize is equipping the students with the tools they will use most often during the beginning of the course, such as the drawing/filling and text packages. As the course proceeds we examine animation and 3D packages. Students choose those tools with which they feel comfortable to create their projects while they are learning the more esoteric subjects.

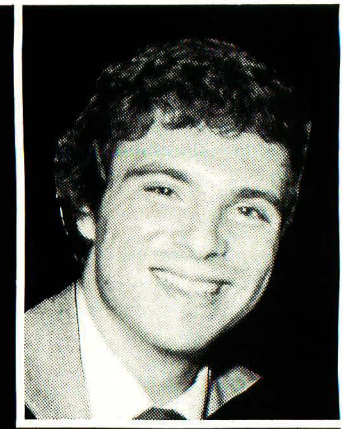
The only way you really learn to save a design in intermediate stages is by losing a picture in progress. This lesson is better learned in a classroom setting, than after you advance to a $200,000 system or are working under a deadline. You can read a thousand books, but until it actually happens, it won't sink in. In class the instructors can make comforting noises, hold your hand and reminisce about the time it happened to them.

Computer graphics teachers do not teach art: they teach computer usage to artists. If we were to teach art, we would need a four-year course instead of a twelve-session course. The students' backgrounds vary, but what we perceived as a basic liability (too many students, not enough computers), has turned into an asset. Doubling and tripling up is common. By this means, the non-artists benefit from the expertise of the artists. This also reinforces earlier study of the software — someone around the keyboard will remember the proper command.

Micros For Everyone

Software at this point in the development of personal computers is much more important than the particular computer (in the future, integrated hardware/software systems, such as Lisa, will change that situation). If an applications program proves to be useful, you can be sure it will be translated to run on other computers. Software is becoming more powerful, but with this power comes complexity. Software classes can make this power accessible to those who need such tools. They also can introduce more people to the uses of computers, and start them on the road to reaping the benefits of this technology.

Schooling with Software



by Peter J. Favaro

As *SoftSide's* new Education Editor, I would like to outline my goals for the coming year and encourage anyone interested in educational computing to communicate your needs and experiences to us so that we know what we should cover in this area.

If you have read the articles I have written for *SoftSide,* you know that I am interested in the design of both educational and recreational software. My emphasis has been on combining curriculum development, psychology and human factors in software design. In the past few months, I have written articles and software on cooperative games, the use of cognitive and behavioral principles in programming, and self-help programs (see Relaaax... in *SoftSide Selections,* page 48).

As I "settle into" *SoftSide,* I have two important goals for future issues. They are:

- Develop and publish creative educational software — software that goes beyond the three R's; software that teaches and is fun to use at the same time. Let's put our heads together. The results could be profitable both to you and the educational community.
- Develop a communications network for people interested in educational computing. To do this, I need to know what your educational computing activities are. Are you developing software or curricula for computer education? Have you used any particularly effective educational software? Are there educational materials that you would like to see reviewed? Would you like some advice on how to use computers in your classroom or district? Drop us a line and maybe we can help. We are still in the earliest stages of determining how computers help people learn. The only way to develop techniques that work is through communication, and that communication can start right here.

It's not often that I get the opportunity to review strictly educational materials, so I'm pleased to devote this article to reviewing an array of educational aids, including software, publications and materials designed to help teachers teach, evaluate and learn about educational computing. The materials chosen for review were selected because they illustrate important issues in educational computing today.

I Love America Series

This series consists of four programs, *Small Town U.S.A, U.S. Cities, History Star* and *U.S. Map.* A teacher's manual comes with each of these programs.

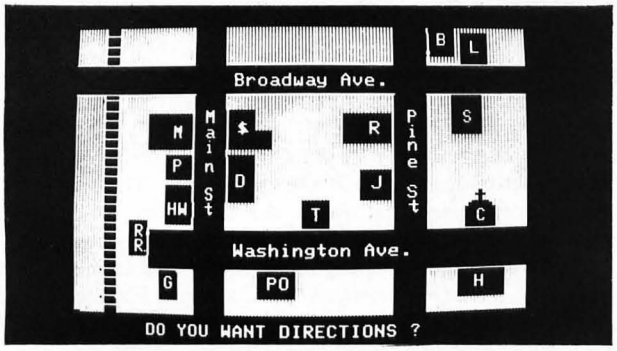
On the positive side, the series' subject areas do not reflect the overemphasized reading, reading comprehension and spelling areas so common in educational computing today. I also liked the fact that some of the activities seemed well-suited to very young children. One such program is *Small Town U.S.A.,* a map reading task designed to help teach direction finding, street naming and skills for everyday living. Children are presented with a map of a small town and asked questions about how to get from one area of town to the next, what they might do once they get there, and so on. When I tested the software, I thought it would be ideally suited for children in the five- to eight-year-old range, because these children are just exploring and starting to understand the environment beyond their own house and block. However, the five- and seven-year-olds I tested couldn't follow the directions for the tasks because the reading level was beyond their ability. I questioned the people at Island Software about this and they said that the program was designed for third and fourth graders (nine- through eleven-year-olds). When I tried the program out on some older children, I found that they weren't challenged. In other words, it seems that if the children were old enough to understand the task, they already had the skill to do it perfectly.

The problem inherent in this piece of software and in so much current educational software is the lack of appropriateness of the task to its intended population. The program does have its good points, however, and is worth considering for very young children and perhaps some special education populations under the appropriate supervision.

U.S. Cities and *U.S. Map* give practice at reading the coordinates of a grid map and help teach identification of the states. These programs complement each other nicely but do not justify the $40 or $50 necessary to purchase them together. They should be offered either on the same disk or cassette, or as a package for a lower price.

History Star is a unique approach to teaching history facts. It asks questions in a TV game show format. Two students compete to answer a wide variety of questions in areas including Indians, pioneers, wars, black Americans and famous women. Overall, this program is the best value in this particular series. If you are a teacher considering this program, check with the people at Island Software to make sure that the facts covered in the programs reflect what you are teaching in class.

From Island Software, Lake Grove, NY 11755. System requirements: the Apple® II family with 32K and disk drive. Suggested retail price: $25 per disk; $100 for the set.



Screen display from Island Software's "Small Town U.S.A."

Educational Activities Series

The programs I reviewed from the Educational Activities line were *Math Invaders* ($16.95 cassette; $19.95 disk. Also available for the TRS-80 Model I/III on cassette and disk), *Math for Everyday Living* (five cassettes, activity masters and program guide, $79; two disks, activity masters and program guide, $85. Also available for the TRS-80 Model I/III on cassette and disk.) and the *Dragon Game Series* for Language Arts (three cassettes, $49; one disk-six programs, $49. Also available for the TRS-80 Model I/III on cassette and disk and the Atari 800 on disk only). In addition, Andy Kassinove, age fifteen, reviews EA's *Cells* program in this issue (page 42).

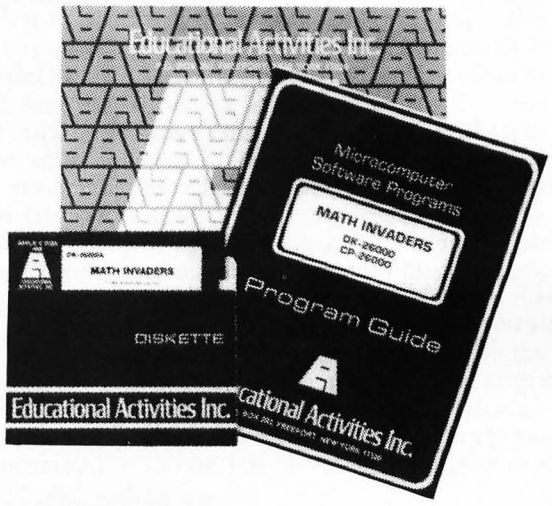
These programs were all of good to very good quality and good value. All had good visual effects, used character fonts that were large and easy to see and were fun to play. An element of surprise and humor kept the attention of the students I tested these on. Many of the lessons in these programs are taught in the the context of "real life experiences."

Take *Math for Everyday Living,* for example. This series of ten lessons puts the child in various settings where money must be exchanged (in stores, restaurants, etc.) and numbers must be manipulated. One task is to order from a menu and then compute the bill. These exercises are practical, useful and transfer nicely to real life situations. The only unrealistic event I encountered in *Math for Everyday Living* was when I was told that my smallest bill was a fifty and I had to figure out the change for a bill which totaled $4.37. I don't know too many fourth graders who get that kind of allowance!

The Dragon Games from the Language Arts series gives practice identifying nouns, verbs, adjectives, synonyms, antonyms and contractions. Each lesson provides a review example of the appropriate part of speech, followed by a board game in which the goal is to keep away from the fire-breathing dragon by scoring points with the correct answer. The computer's banter, during the exercises, is very engaging.

Math Invaders is an interesting math drill which asks the student to punch in the numbers to various problems presented by invading alien aircraft. Although the task is a drill exercise, it is done in an amusing way and at a reasonable price. All of these programs would make a fine addition to any classroom software library.

From Educational Activities, Inc., P.O. Box 392, Freeport, NY 11520. System requirements: the Apple II family with 48K RAM. Certain packages in this series are available for the TRS-80® and Atari® computers. See text for details.



CELLS

Reviewed by Andy Kassinove

An often neglected component in reviewing educational software has been obtaining user feedback. Starting with this issue, therefore, we'll allow students to have a crack at reviewing some educational software. If you are between the ages of ten and fifteen and would like to write a review, either alone or with a group of friends, (your parents can help) tell us who you are, what grade you're in and what you would like to review. Address your correspondence to: Review Editor, SoftSide Publications, Inc., 10 Northern Boulevard, Amherst, NH 03031. This month Andy Kassinove, age fifteen, from Dix Hills, New York reviews Cells from Educational Activities.

From Educational Activities, Inc., P.O. Box 392, Freeport, NY 11520. System requirements: the Apple II family with 48K, disk drive and DOS 3.3; TRS-80 Model I/III with 32K disk, 16K cassette; Atari 800 with 32K and disk drive. Suggested retail price: $39.

If you have a young child who is interested in science, or if you are taking high school Biology, this program is for you! It is packed with facts about cells and presents them in an eye-catching and sometimes humorous way. The entire program is done on the hi-res screen and uses large, easy-to-see letters. The animation could be smoother and show less flicker but, all in all, it does a good job of illustrating the concepts involved with cells. The program is divided into two parts, each selectable from the menu which boots up following the logo.

Part one deals with the structures and functions of the various parts of a cell. It covers the three main structures: the cell membrane, the cytoplasm and the nucleus. A short, easy quiz follows on the material discussed. The program positively reinforces correct answers, and gives encouragement when you answer incorrectly. This procedure is in keeping with modern educational methods, and encourages rapid learning.

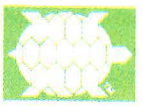
Next, the program illustrates the amoeba's amorphous shape with color graphics and animation. A similar process illustrates the formation of pseudopodia and how they are used in feeding. It illustrates different kinds of organelles and their functions, along with enzymes and vacuoles and their respective functions. A short quiz accompanies this section.

The second part of the program gives a good, graphically animated presentation of cell reproduction. It discusses both the binary fission and mitosis forms. This, too is followed by a short quiz, which ends the program.

Cells handles errors well. The keyboard is childproof while the program is running, and if you hit RESET at any time the program reboots.

In conclusion, *Cells* is a useful learning tool both during a biology course and as a review at the end. It is a good starting point for young scientists. The graphics could be smoother, with less flicker, but they get the point across. Except for an occasional beep, the program lacks music and sound effects which would make it somewhat more attractive. The material, however, is presented in a logical fashion and the text is easy on your eyes. All in all, it is a highly informative program, with good graphics and animation. I recommend it highly as a teaching aid, and at $39 it is an excellent value.

Krell's MIT LOGO

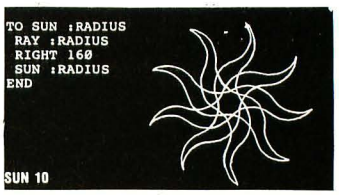


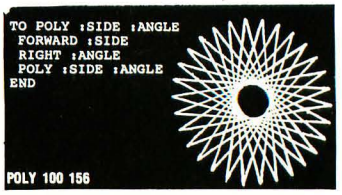
From Krell Software Corporation, 1320 Stony Brook Road, Stony Brook, NY 11790. System requirements: the Apple II family with 64K RAM and disk drive. Suggested retail price: $89.95.

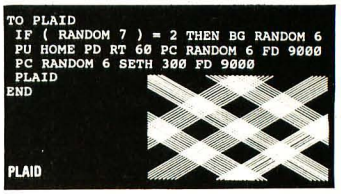
If you are thinking of buying a *LOGO* interpreter for your Apple computer, give serious thought to *Krell's LOGO.* It's versatile, instructive, easy-to-learn and a great buy for the money. For $89.95, you get Krell's *MIT LOGO,* plus a backup disk, a utility disk (with several demo programs, shape tables for changing the shape of the turtle and a full 6502 assembler), a tutorial disk called *Alice in LOGOLAND,* a wall chart of the *LOGO* primitives (commands), and a sample copy of *LOGO* and *Educational Computing Journal* (written primarily by Krell staffers and very good). This kind of program support is rare these days.

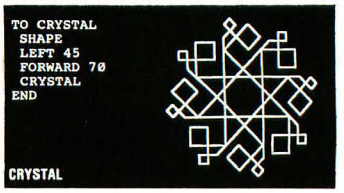
Krell's *MIT LOGO* manual is brief but fairly concise. I had to re-read most things twice, because the language in the manual is somewhat technical. Procedures and commands soon become clear, however. One of the manual's positive features is that it provides examples of almost everything. The first paragraph of each chapter neatly summarizes the chapter, which also lends clarity.

This version of *LOGO* has a full screen editor that can be used as a text editor out of the *LOGO* environment. In the "Draw" mode there is a full screen option with no text, or a split screen which allows four lines of program text on the bottom of the graphics screen. You may save *LOGO* drawings with up to six pen colors to disk using the SAVE and SAVEPICT commands and retrieve them from disk with the READ and READPICT commands. There are provisions in the software for using Apple peripherals, such as printers and paddles, as well.









The Krell *LOGO* primitives support all the major *LOGO* drawing, number manipulation, and word and list operations. One of the most impressive features of this package is its accessibility to people of all programming levels. For the absolute beginner, or young child, there is the *Alice in LOGOLAND* tutorial disk which helps the user understand *LOGO's* capabilities. The disk presents games and exercises that teach the user how to create and manipulate shapes, work with lists and define procedures. This disk allows even the most beginning level programmer to work immediately in a *LOGO-centered* environment. For the advanced programmer, ample documentation is provided on modifying *LOGO* with Machine Language routines, hints on how to generate sound and music, and a 6502 assembler to help you get the job done.

Bruce Vassilakos-Long is an elementary school teacher in Bishop, California who uses Krell *LOGO*. He relates his success with Krell *LOGO* to the documentation and the *Alice in LOGOLAND* disk. His comments: "From the first time I attempted to move the turtle to create geometric patterns, I felt that I had come upon a natural gift. When I first learned programming (in BASIC), I was disappointed by the tedium involved, and the great difficulty I had in getting the computer to re-create my conceptions. On the contrary, with *LOGO's* accessibility, I found my thoughts flowing smoothly into the computer and taking form." Bruce would like to get in touch with other teachers who are working with *LOGO*. You can reach him at 2284 Longview Drive, Bishop, CA 93514

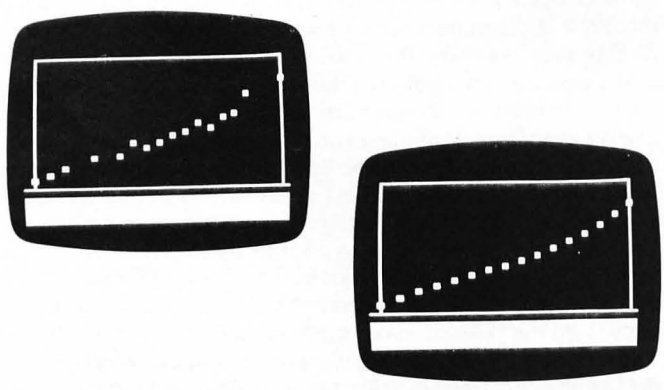
Statistics Software

Much of what is done in education requires evaluation. Unfortunately, statistical analysis goes hand in hand with evaluation and aggravation goes hand in hand with statistical analysis. I occasionally teach statistics for the behavioral and social sciences and can attest to the resistance people have to the laborious work of adding up numbers, minusing their squares and so on. The people at Dynacomp have put an end to all this. Their statistics packages (*Stattest* ($22.95), *ANOVA* ($43.95), *REGRESSION1* ($23.95), *REGRESSION2* ($23) and *MULITILINEAR REGRESSION* ($28.95)), are excellent, and solve what used to be a three hour statistics problem right before your eyes in a matter of seconds. All of the programs offer the option of storing your data on disk for later loading and editing, and most of the programs have options to send output to your printer. Programs are all menu driven and autoboot format.

Stattest includes t-test, Chi-square, F-tests, simple regression and a random number generator. Descriptive statistics are provided by the t and F test programs. This package, along with the *ANOVA* package, which performs one-way through five-way analyses, are most of what the educator with an eye toward program evaluation would need to get some good measures on the program's effectiveness. For more exotic applications, the *REGRESSION* packages are helpful in making predictions from large groups of variables.

The manuals for these programs are superb, each one giving background, references and examples for each type of statistic. Procedures are laid out in a clear and orderly fashion. I was able to complete my first statistic problem, a two by two factorial *ANOVA,* literally seconds after I sat down at the keyboard. Programs like these should find their way into many educational computing environments. They are important tools in helping us discover what works and what does not.

From Dynacomp, Inc., 1427 Monroe Avenue, Rochester, N.Y. 14618. System requirements: Atari 400/800/1200 with 24K RAM; the Apple II family with 16K RAM and the TRS-80 Model I/III with 16K RAM. All are available on both disk and cassette.



Screen displays from Dynacomp's REGRESSION1 package.

Educational Publications and Aids

I thought it might be helpful to mention a few publications of interest to those involved in educational computing. *Computers in Curriculum and Instruction* (Association for Supervision and Curriculum Development, 225 North Washington Street, Alexandria, Virginia 22314, 1983) is a compendium of articles focusing on important issues in educational computing. Primarily for those just getting their feet wet, article titles include "Some Basic Information About Computers," "Resources for Instructional Computing," and "Help! What Computer Should I Buy?". The articles are informative, well organized and clearly written. Also included are samples of statewide guidelines for computer curricula.

The Computing Teacher: The Journal of The International Council for Computers in Education offers information on a variety of topics in educational microcomputing nine times a year. A recent issue focused on using microcomputers with the handicapped (Feb., 1983). Regular departments include software and book reviews, "The LOGO Center," and "Computers in Science, English, and the Media Center." The subscription price is $16.50 for nine issues. Send subscription orders to: The Computing Teacher, 135 Education, University of Oregon, Eugene, Oregon 97403.

Anyone interested in seeing the product of a rather ambitious school district publishing its own *Computer News* should contact Molly Weeks. She edits a newsletter about classroom computing for the Dade County Public Schools, Dade County, Florida. *Computer News* provides product listings and brief reviews and condenses selected educational articles from several microcomputer magazines. Other school districts interested in undertaking similar projects probably can learn much from this newsletter. Write to Molly Weeks, Computer Education Specialist, Dade County Public Schools, 1410 Northeast Second Avenue, Miami, Florida 33132.

Finally, one educational computing aid that exemplifies the "no frills" approach to learning about computers is Bell Laboratories *The Information Machine,* a paper computer and manual designed to be a guide to "understanding how computers work, how to program them and the logical structure that forms the basis of all programming." The manual is a cartoon-illustrated guide and can be used effectively as an introduction to computers, or as a one or two week curriculum on computer awareness. For more information, contact Bell Laboratories, Holmdel, NJ.

In Conclusion

That wraps up our brief overview of some of the exciting things going on in educational computing. As we gradually move away from the "teaching machine" approach, this area will develop more rapidly than any other aspect of our lives influenced by microcomputers. As the interest in microcomputers in education increases, so will our coverage of technologies in education, educational games and teaching aids. Please keep us informed of what you would like to see in this area. To everyone involved in educational computing, best wishes for a successful academic year.

COMPUTERS CAPTURE THE

Consumer Electronics Show

by Steve Birchall

Computers have taken over the Consumer Electronics Show. For years, this show has been primarily an audio show — even video (an extremely popular consumer electronics item) took a back seat to King Audio. All the rest, from calculators and watches to smoke and radar detectors was, and still is, banished to the McCormick Place basement.

The second piece of CES news is the staggering enormity of this show. Well over 80,000 people attended the four-day affair. As recently as five years ago, McCormick Place was spacious enough to hold most of the exhibits. This time, it completely filled both floors of the main hall, plus the lobby, function rooms, and two floors of rooms in the McCormick Inn, two floors of a nearby warehouse (McCormick West, where most of the computer hardware and software was displayed), and the Conrad Hilton Hotel (where the "esoteric" high-end audio manufacturers held forth). To someone like me, who had abstained from this madhouse for several years, the impact of all this was numbing. You have to revise your expectations completely and accept the fact that to see everyone you hoped to see, check out the exhibits you have some interest in — much less get around to see everything — is impossible.

What exactly is the CES, you may ask, and why is it held? The show is open only to the industry, not the general public. In theory, it is the marketplace where manufacturers (or importers) display their wares to the retailers. Buying and selling, making deals, opening new dealers, finding new lines to carry, introducing new products — all are the order of the day in this orgy of free market capitalism. Anything you will buy at Christmas was likely displayed at the CES, where deals between manufacturer and retailer were clinched. Needless to say, much wining and dining usually accompanies this high-level wheeling and dealing, and Chicago is the perfect place to do it in style.

Along with this, everyone looks up friends and exchanges all the latest gossip on which company is doing what, which company is on the skids or making money so fast they can't count it. The CES is a "slave market" for salesmen at every level — from the lowliest car stereo installer to the loftiest corporate marketing expert. If you want to be in a place where human energy is being expended at a breathtaking rate, the CES is for you.

The RB5X robot from Androbot obediently delivers your morning paper, but can it chase a Frisbee? The Atari booth drew large crowds of game fanatics eager to try out the newest releases. A portent of things to come on computer displays is Casio's pocket-size flat screen LCD television receiver. A keyboard instrument from Yamaha, the MP-1, not only remembers any melody you play, but can print it out in correct musical notation.

Adam Bites the Apple

In computer circles, the talk of the show was a new machine from Coleco®, called Adam. Imagine a home computer with 80K RAM, which can run "most" Apple® software (SmartBasic™ accepts all AppleSoft™ commands except PEEK and POKE), and also is compatible with CP/M™. Adam has a word processor in ROM, complementing a nicely laid out full ASCII keyboard (with six function keys, separate cursor control key cluster, Command Keys for common operations, and an "Undo" key). They claim their new tape storage system is nearly as fast as a disk. The two game controllers each have a ten-key numeric keypad. The letter quality printer uses standard, interchangeable daisy wheels. All of this (including printer but no monitor) has a suggested retail price of $600. And Coleco hasn't forgotten all those who bought their game machine. For $400, you can upgrade it to an Adam.

Software for Adam seems to be no problem, since it runs the majority of AppleSoft and CP/M programs. The biggest hitch is that no disk system is available at present (although it has provisions for two drives, and Coleco says they will introduce one soon). Thus, for the present, you must buy software on their DataPack tapes, or type it in from printed listings (such as found in ***SoftSide Selections).*** Among Coleco's planned early releases

Coleco had the hottest new product of the show. Adam (above) is a complete word processor with daisy wheel printer, which also happens to work as a full-function home computer and game machine — all for under $600. Atari had similar thoughts with their Writing System (above left) which features a new letter quality printer and 600XL computer with AtariWriter, for the same price as Adam. New Atari peripherals include a Trak Ball Controller and a graphics tablet.

Consumer Electronics Show, ***continued***

are a self-teaching course in BASIC, a version of Logo developed under the guidance of Seymour Papert himself, the Dr. Seuss Early Learning Series, ColorForms® (electronic crayons), the Homework Helper Series (quizzes), and the SmartFiler® database manager. All the familiar Coleco games of course run on Adam, with sixteen colors and 32 sprites.

Atari's XL Series

Atari® had a large number of new products to announce, including four new computers, and many intriguing peripherals. The XL family starts with the 16K (expandable to 64K) 600XL for $200. Features include a keyboard with 62 keys, a Help key, and an international character set. The 800XL is similar, with 64K built in, and a monitor output. On the upper two models, a built in modem (300 baud) and a speech synthesizer are standard, along with 64K, and four programmable function keys. In addition, the 1450XLD has an integral disk drive (double-sided, double density), with space for a second drive. Atari's new disk drive stores up to 254K bytes, and does it two and a half times faster than the old drives.

Three package deals are available. The Programming System, for $380, includes a 600XL, 1010 tape drive, and assorted software. The Writing System has the 600XL, the new Letter Quality Printer, AtariWriter™, and other accessories, for $600, placing it in direct competition with Coleco's Adam. Atari's Entertainment System includes the 600XL, MS PAC-MAN, Donkey Kong, two joysticks, and other goodies, for $300.

Atari Peripherals

The Letter Quality Printer is an interesting device — smaller and quieter than most printers. Its print head looks something like an adjustable date stamp, with several rotating bands of characters side by side. The design is ingeniously simple and relatively quiet in operation. Occasionally a letter will be slightly out of line or crooked, but no worse than on an inexpensive typewriter. For the typical home user, the printer's output is quite acceptable.

Steve Gibson was at the Atari booth demonstrating his new Light Pen for the Atari. It seems to have the same capabilities as the one he designed for the Apple, and should stimulate some excellent graphics designs from Atari owners.

The new Touch Tablet® has a 4½ × 6 inch active drawing area (228 × 228 points). Two buttons on the tablet and one switch on the stylus can be used as fire buttons, or to select menu items, and to start or stop line drawing. Other Atari new products include the 1050 disk drive, 1030 direct connect Modem, CP/M® Module, and several joysticks and trak ball controllers.

Future Home

GE's Video Products division demonstrated a home network system which uses a central computer to control major appliances in the home or office. Logically enough, the computer is built into a TV set, and secondary microprocessors are included in each appliance (other manufacturers are putting in controllers also). Voice synthesizers can announce the end or start of an operation — "Beep. Beep. Time to wake up. Shall I start the coffee now?" The system transmits appropriate control signals through the existing household wiring to each appliance, and can control such things as temperatures of cooking elements as well as turn them on and off. Anything electrical — interior and exterior lights, audio and video components, telephone answering machines, home security systems, smoke detectors, automatic emergency notification devices — is capable of being operated by the Homenet. By letting your computer control your heating and air conditioning, you could save money and fuel. A telephone interface lets you call up and alter the programming when your plans change. GE's system uses a simple programming language and interfaces, which they hope will become an industry standard.

Just in case you're overly impressed by all this, I received a catalog from JS&A, showing a *wristwatch* which turns appliances on and off with an infrared beam and control boxes. The watch is not as versatile as the GE system, but it *is* on your wrist.

American Bell displayed the Genesis Telesystem, a microprocessor controlled phone with smoke detectors and home security (i.e. closed circuit TV) capabilities. For them, this was only a hurried beginning, because eventually they will bring out their own computer/telephone/home control system, which certainly will compete with GE. A struggle over standards may develop between these corporate giants, but let's hope they can agree on common protocols so consumers can buy the systems with some degree of confidence that the controllers will be compatible with the appliances they buy.

Personal portable stereos let you listen to music wherever you are. Yamaha's lap-sized keyboard instruments, such as the PS-400 (above), take the idea a step further — now you can have the pleasure of making music anywhere you want to, without disturbing others.

Control all the appliances in your home through GE's Homenet (below). The system, incorporated into a TV set with a keyboard, uses a simple programming language. If your plans change, you can call your computer and reprogram it to start supper an hour later.

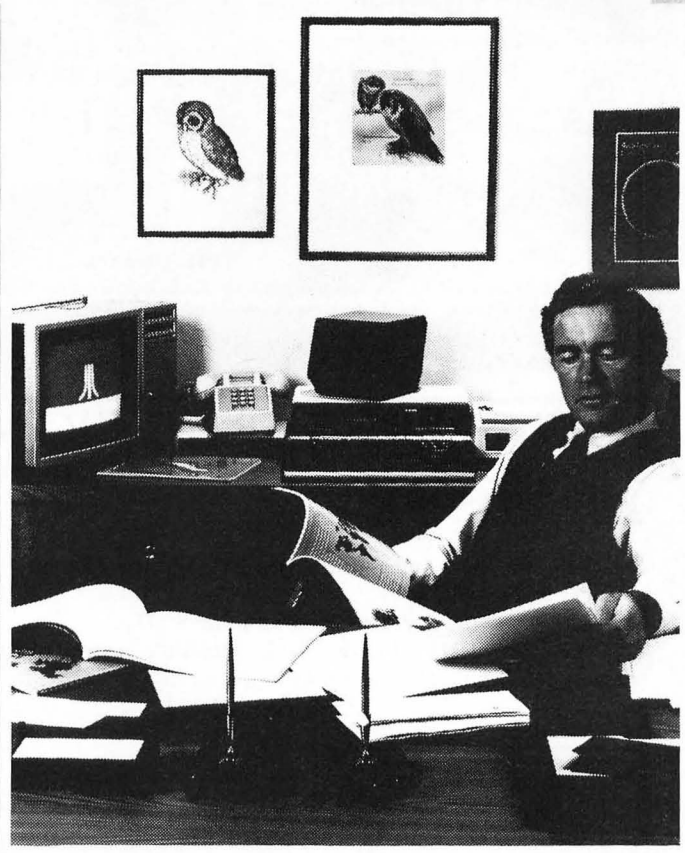
Some Videotex equipment was on display at various booths, but no one was available to talk about or demonstrate it. This will change when the major networks and their local affiliates finish gearing up to transmit it. CBS and NBC currently are broadcasting Videotex and ABC plans to begin soon. Next year, manufacturers should be willing to give more serious attention to this equipment.

Pocket TV and Color LCDs

Some of the more frivolous items have far-reaching implications for the future of personal computers. You all know about the Sony Watchman TV, but now Casio has a true flat screen LCD TV set about the size of two half-decks of cards placed beside each other, with a 2¾ inch screen. Seiko announced a Dick Tracy style wrist TV, which they claim could be two-way if desired (however it has a large auxiliary unit worn on your belt). The American importer suggested military applications — a tiny camera on a soldier's wrist could give a frontline view to a field commander. The Japanese were quite surprised when they heard about this kind of application. The difference in point of view is symptomatic of why American manufacturers no longer do well with consumer electronics.

Some hand held video games appeared with *color* LCD displays. When will Casio, Seiko and Sony have color personal TV sets? More to the point, when will larger color LCD displays be applied to portable computers? Can you imagine, for instance, a Radio Shack Model 100 with a 24 × 80 color graphics display? Such things are certain to appear within the next six to twelve months. Zaxxon, Coleco's game featuring 3D graphics (using two superimposed images), also points toward the future of display devices for computers.

Yamaha showed some keyboard instruments which looked like fun — the equivalent of a "Walkbeing" in a performing instrument. Some had little musical games incorporated into them. The MP-1 has a plotter, which will print out the melody line with the correct rhythm, and chord names of any tune you play into the device. The ball point pen plotter handles up to three sharps and flats, and prints on a 2½ inch roll of paper. With the unit's memory, you can record and playback two voices, and play live on the keyboard along with the stored information.



Atari envisions the office-of-the-present as a comfortable working environment. Their top of the line 1450XLD (above), is simply another device integrated into the executive's work-style, rather than a large monster with white-coated attendants. It sports a built-in modem and a voice synthesizer for ease of operation and friendly human interface.

RELAAAX...

SoftSide has brought you programs that will keep you up at night — this one may just help you get to sleep. Peter Favaro, a school psychologist and educational and recreational video game design consultant, and Richard M. O'Brien, a clinical psychologist and professor of psychology at Hofstra University, have developed this system to help you relieve tension, and **Relaaax...**

SOFTSIDE SELECTIONS

FRONT RUNNER

Relaaax... for the Apple, Atari, IBM PC, and the TRS-80
by Peter J. Favaro
Translations by the ***SoftSide*** Programming Staff
This program offers an easy, highly accessible system for relieving tension that builds up after a long day.

APPLE® VERSION

Gladiator ..... Disk Version Bonus
by David Calaprice

TV Art ..... Program
by Jerry Cole

IBM® PC VERSION

User-Friendly Input ..... Program
by Kerry Shetline.

ATARI® VERSION

Chemistry Assistant..... Disk Version Bonus
by Art V. Cestaro III

Math Quiz ..... Program
by Jerry P. Waid

TRS-80® VERSION

Nine Games for Preschool Children .... Disk Version Bonus
by George Blank

Fall Constellations ..... Program
by Kirk Darbe

SoftSide CV/DV ADVENTURE SERIES

High School for the Apple, Atari, IBM PC and TRS-80.
By David Pleacher
Translations by Peter Kirsch
High school is one of life's great adventures. Now you can play all the peer-group games and grade games again with this month's adventure and graduate with honors.

Home Bar Codes

Bar codes have been around for awhile, but readers compatible with home computers have been scarce and expensive. Databar showed the Oscar™ Model 1, which will work with Atari, Commodore, TRS-80, Texas Instruments, and Timex/Sinclair computers, and costs less than $80 retail. Software printed in bar code form is much less expensive to publish and distribute than in disk form, so the personal computer potential for Oscar is considerable. Many dot printers can print bar codes, with suitable software, so even individuals could send programs and data to each other via bar code. If you could address mail with the bar codes used by the Postal Service, mail service might be faster. For a stamp or 35mm slide collection, you could index each item with bar code labels, and read out detailed information on a particular item from a data file simply by scanning the label.

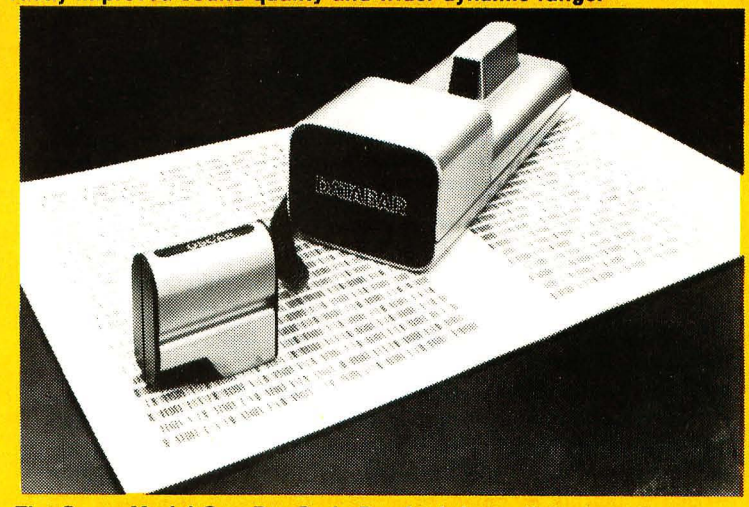
Video: Stereo Sound and Smaller

In the field of video equipment, every manufacturer had videodisk players on display. Significant numbers of new disk titles were announced also. Pioneer had a prototype interface to enable an Apple to control a videodisk interactively. Video is becoming less bulky, too. You will see many new portable systems using the down-sized cameras and recorders. This light-weight, convenient-to-use equipment may turn us all into "home movie" buffs, and you will see lots of TV advertising promoting the portables. One tiny Toshiba camera uses a CCD (charge coupled device) "tube" that looks more like a chip. Many other cameras boasted of their low-light capabilities. Sony has a complete one piece system — a camera with a Beta recorder built into the handle.

Sony also showed their Beta HiFi unit, which encodes (via an FM technique) a stereo audio signal into the video signal (using the unused space, or blanking interval, between frames). The 80dB signal-to-noise ratio is close to the 90dB of digital audio disks, and is of much higher quality than normal VCR audio. For compatibility, Sony uses the regular narrow audio track as well, but a circuit on the Beta HiFi player switches to the stereo signal when it detects the carrier. Since this format does not alter broadcast parameters, I wonder what would happen if such tapes were transmitted over the air. Just run the signal through your Beta HiFi player and you'll have stereo sound, without needing FCC action (or the new, improved inaction/deregulation) on standards. Needless to say, the VHS format has this potential also, and those manufacturers are ready to go ahead if demand seems likely.

Television receivers in general now are moving in the direction

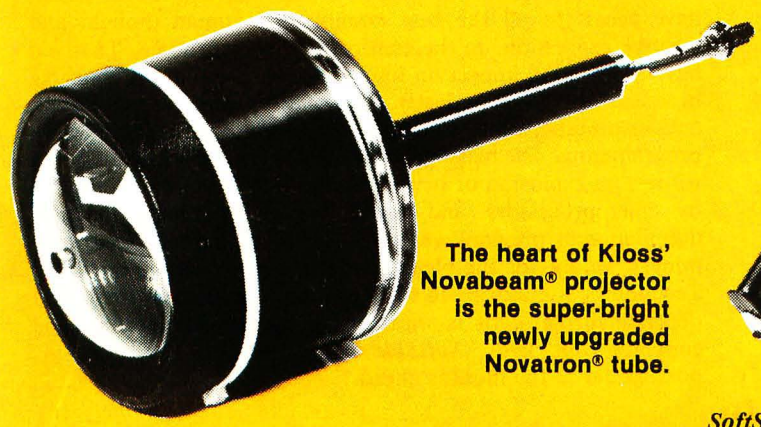
NEC's VC-739E Beta HI-FI video recorder offers stereo audio, with vastly improved sound quality and wider dynamic range.



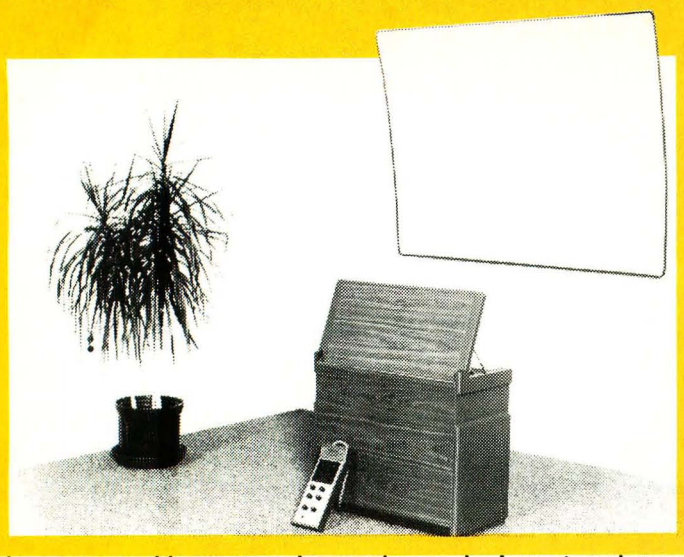
The Oscar Model One Bar Code Reader brings all the benefits of bar code technology to the home computer at a reasonable price. Databar offers printed software in bar code format — and it's less expensive to manufacture than disk-based software.

Kloss' video projector produces large images with outstanding clarity and brightness. Some models now use flat screens.

Canon's portable VHS video recorder and camera system is lightweight and compact.



The heart of Kloss' Novabeam® projector is the super-bright newly upgraded Novatron® tube.



Large screen video systems have an impressive impact on viewers, especially when the signal source is extremely clear, such as from a videodisk or a good cable company.

One of the many digital audio disk (compact disk) players now available is NEC's CD-803E. They represent a quantum leap in sound quality and user convenience. You can hear voices and instruments reproduced with astonishing clarity and detail. Acoustic images have spacious ambience, a sense of depth and convincing realism.

Consumer Electronics Show, ***continued***

of separate components, like audio equipment. Soon you'll buy a tuner, a monitor, a Direct Broadcast Satellite decoder, a videodisk player, a tape deck, and maybe a large screen projector, separately, based on the equipment's merits and cost, and your own needs. The all in one TV set will disappear. Picture quality is moving upward to accommodate better signal sources (such as videodisk), and even the sound is improving. Stereo is fairly common (again because of videodisk and Beta HiFi). Unfortunately, the "new improved" speakers typically are of car stereo quality — better, but not by much, than the El Cheapo four-inchers TV manufacturers used to stick in. You'll hear a lot of advertising hype about better audio quality on new sets, but be sure to *listen* before you buy.

Digital Audio Disk

The hottest item in audio is of course the digital audio disk, and nearly every manufacturer had one. Six months ago, only one or two were available. The ultra-clear sound quality possible with this technology is breathtaking. With a 90dB signal-to-noise ratio, the music rises out of virtually complete silence — no tape hiss, no ticks and pops, nothing to disturb the music. You can put fingerprints on the records, and even scratch them, without causing any audible effects. As you might expect, now that a large number of companies are making the compact disk players, the sound quality varies from one to another, depending on how well the design is executed. Some sounded positively muddy, while others, notably Denon's, produced the cleanest sound I have ever heard on home equipment. Once again, listen before you buy.

Cultural Winds in Chicago

No trip to Chicago would be complete without partaking of some of the city's cultural opportunities. As it happens, the Publisher has a background in musical theatre, and the Technical Editor in electronic music, so the first thing we did was to look through the paper for possibilities. Within minutes of arriving at our hotel, we found ourselves rushing off to Ravinia for a performance of Marc Blitzstein's *The Cradle Will Rock,* given by The Acting Company, and directed by no less than John Houseman, who directed the first performance in the Thirties. Originally written as a WPA project, this show depicts the struggle of a working class hero against the Establishment, a topic which seems newly relevant in the Reagan era. The original production had considerable problems and almost didn't open. Only Blitzstein's determination to go ahead, even if he had to do it as a one-man show, saved it from that fate.

The staging is austere, as befits a Depression era opera: a bare stage, with an upright piano in the center (Blitzstein himself played in the original), and rows of chairs on either side. Blitzstein's score makes up for the austerity with rich inventiveness. He continually alludes to a wide variety of styles from both popular and serious music, and displays a wonderful imagination in his harmonic language and rhythmic textures. The Acting Company gave it an authentic, spirited, and well-polished performance. Perhaps they were a bit too careful to recreate the performance practice of the period, which took away some of the spontaneity and gave it a more scholarly flavor. Nonetheless, it's an excellent production.

On another evening, we discovered that Carol Channing was doing *Hello Dolly* at McCormick Place's Arie Crown Theatre, so of course we went, and enjoyed ourselves thoroughly as she romped through her showpiece with style and humor. Gower Champion's choreography, a tradition with this show, was executed with lots of energy.

A trip to the Loop resulted in a detour through the Chicago Art Institute. We couldn't take the time to see much of the collection, but the translucent, transcendent beauty of Chagall's stained glass windows will remain in memory for a long time. That alone was worth a trip to Chicago.

Computers and the Human Experience

Returning to the primary subject, you can now buy from museums, such as the National Gallery, a videodisk containing 10,000 paintings for $65. Pop that on your computer-controlled interactive videodisk player. Program it to do things such as display in succession ten random paintings by Bosch, El Greco, and Picasso, and you'll have a thoroughly enjoyable evening at home exploring the minds of these great artists. The point is that the micro-electronic technologies are merging and re-forming in new ways to make our lives richer, more satisfying, and more fun. For the first time in human history, anyone who wants to can have access to all the best examples of human thought and creative expression, in the comfort of his own home. That will have a profound impact on future society. It already is changing broadcast TV (at long last!). Now that viewers have real alternatives on tape, cable, and disk, they are not watching network programming. The message from the CES is that we are about to witness an explosion of new consumer electronics made possible by microprocessors and digital storage methods. Audio's dominance in previous years was really an expression of how hungry we are for the pleasure that the arts bring to our lives. Technology is maturing to the point that it no longer is cold and depersonalizing — it is warming up human relationships and enriching our lives. ***SoftSide*** will look into these exciting possibilities in the months ahead.

Financial Operating System

by J. M. Keynes

"I'LL BUY THE EMPIRE STATE BUILDING WITHIN 48 HOURS WITH NO CASH"

You have, no doubt, seen the multitudes of ads by so-called multi-millionaires who, for a small consideration, will teach you how to become wealthy overnight. Usually the method is purchasing single- or multi-family real estate for no money down. They even suggest that you can do so even though your credit rating may be worse than International Harvester's. They hold free (?) seminars, conducted by nice-looking young men who are well-dressed, well-scrubbed, and who wouldn't *think* of putting any con on you. However, the super secret information they are selling will allow you to purchase half the city for no money down. In no time at all, you will be able to fleece all those unsuspecting sellers out of their property, and wind up with bundles of cash in your pocket along with that choice piece of real estate.

I recently dropped in on one of these well-attended seminars. The average age of the potential pigeons was about 50. The pitch was excellent. In fact, the jokes were identical to those I heard while attending a get-rich-quick commodity options seminar last year. The young man knew little of what owning rental property is really like, but he spoke convincingly, and knew enough to sell his product to many eager plutocrats that night. The price to become an instant real estate baron (or baroness) was $75, I believe. The generosity of these multi-millionaires is admirable — they are always willing to share their secrets for a paltry sum. To know that some good people are left is reassuring....

So much for "Rentalscam." Your computer can help you to examine quickly and accurately the real potential of any rental property you may consider. The program in Listing 1 deals with the real world. A few basic rules enhance the program's value.

- KEYNES' LAW: "It's three times the work you think it is." If you are buying less than about ten to fifteen units, plan on being your own plumber, carpenter and manager. If you are unwilling to work up a sweat, look for another investment. You can't afford management if you are to prosper.
- THE VACANCY FACTOR: Even the most desirable rental units experience at least a five percent loss from maximum gross income due to such unavoidable things as repainting, recarpeting and other periodic maintenance. The average property experiences about twelve percent.
- BUYING WITH SMALL DOWN PAYMENTS: The properties which can be purchased for small down payments usually rent to people who are at the low end of the pecking order. Hence, the landlord must face such additional expenses as unpaid damage, getting court orders to evict people who refuse to move, bad checks, *ad nauseum.* Many excellent deals are available from sellers who went in with no real idea of the work and problems involved. Think about it: If these units are such a bed of roses, why is the seller so eager to accommodate you?
- PERSONAL MORTGAGE LIABILITY: Before you sign up for a big mortgage, check the law in your state. You may find that you personally are liable for the mortgage. If you default and the mortgage holder forecloses, he sells the property at public auction. If that does not pay him in full, he may be able to obtain a deficiency judgement to seize and attach your other assets.
- FURNISHED VS UNFURNISHED: A furnished property generally shows a better current return than one unfurnished. This is an illusion, since you must replace the furniture about every three to five years. The unfurnished unit requires no such periodic expenditure. Unfurnished units attract, generally speaking, a more stable class of tenant. You need not worry about this group disappearing in the middle of the night.
- THE UNEXPECTED: Twenty-five years ago I bought five units. I didn't mind spending my weekends painting, fixing and trying to collect the rent from 200 pound gorillas in a surly mood. Then, one night at about 2:00 AM my phone rang. "My toilet won't flush," said Mrs. Brooks. "Oh S...," said I. "That is exactly what I intend to do as soon as you fix it, Mr. Keynes." I do not suggest that rental property is a bad investment, but it involves hard work and being on call twenty-four hours. In fact, I parlayed my five units into 50 units in ten years.

The program in Listing 1 does not weight the expenses based on the age of the building. Older structures are more likely to incur unanticipated expenses. Another factor which helps determine the worth to you is your tax bracket. The higher your tax bracket, the more valuable the depreciation becomes. When you apply the program to the property you are considering, you may find that the chicken pie is, in fact, chicken feathers.

The program is easy to use — simply answer the questions. After you have answered all the questions, the computer will present you with information which will help you evaluate a structure as an

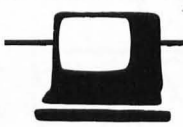
→

Financial Operating System, ***continued***

investment. Note: To exit the program, enter a "0" after a page of information is displayed.

When you invest your money in a small number of rental units you are NOT an investor. It is a business venture wherein you are an active, hard working, partner. Compare this with the returns available from risk-free investments (currently about fourteen percent from AAA rated tax-free bonds) before you decide.

My business is managing other people's money — millions of dollars in the stock, bond and commodity markets *(Ed...and a few million of his own).* I think you will find my Bulletin Board Service of interest. It runs 24 hours a day at 300 baud, seven-bit words, one stop bit, even parity. Call (305) 744-0190.



```
SS SS SS SS SS SS SS SS SS SS SS SS
SS                               SS
SS         TRS-80 BASIC          SS
SS 'Financial Operating System'  SS
SS     Author:  J. M. Keynes     SS
SS      Copyright © 1983         SS
SS  SoftSide Publications, Inc   SS
SS                               SS
SS SS SS SS SS SS SS SS SS SS SS SS
```

```
10 GOTO200
100 C$="":IFA$=""THENRETURNELSEFORJ2=1TOLEN(A$):J1=ASC(MID$(A$,J
2,1)):C$=C$+CHR$(J1+32*(J1)=97ANDJ1(=121)):NEXTJ2:RETURN
110 PRINTMID$(STR$(INT(A+FH)),2);USING".##";A-INT(A)+FH;:RETURN
120 PRINTA$;:RETURN
200 CLEAR100:DEFDBLA-E,G-H,I-W:DEFINTF,J,X,Z
210 DV(1)=0.02:DV(2)=0.01:DV(3)=0.0125:DV(4)=0.015
220 DT$(1)="SL":DT$(2)="DDB":DT$(3)="125%":DT$(4)="150%"
230 D$(1)="BUILDING":D$(2)="PERSONAL PROPERTY"
240 FH=5/1000:Z1=45:Z2=28:Z3=40
250 BU$=CHR$(29)+CHR$(27)+CHR$(30)
260 A1$="###,###.##"
270 A2$="##.##"
280 A3$="##"
290 A4$="##,###,###.##-"
400 CLS:PRINT"NOTE: WHEN YOU ARE ASKED FOR A PERCENTAGE, YOU MAY
 ENTER IT":PRINT"AS EITHER AN INTEGER OR A DECIMAL (24 OR .24, F
OR EXAMPLE).":PRINT
410 GOSUB3050
420 GOSUB3060
430 GOSUB3080
440 GOSUB3120
450 GOSUB3130
460 GOSUB3170
470 GOSUB3350
500 CLS:PRINT@260,"I'M THINKING....."
510 VS=VA*GI/100
520 EV=ER*(GI-VS)/100
530 OI=GI-VS-EV
540 M1=0:M2=0:PT(1)=0:PT(2)=0:IT(1)=0:IT(2)=0
550 IFMN=0THEN710
560 FORX=1TOMN
570 IFT(X)=12000THENIT(X)=AM(X)*(RT(X)/100):GOTO690
580 I(X)=RT(X)/1200
590 T=(I(X)+1)[(-T(X)):T=1-T:MP(X)=I(X)*AM(X)/T
600 MP(X)=INT(MP(X)*100+.5)/100
610 IT(X)=0:PT(X)=0:RP=AM(X)
620 FORZ=1TO12
630 IM=INT(RP*I(X)*100+.5)/100
640 PM=MP(X)-IM
650 IT(X)=IT(X)+IM
660 PT(X)=PT(X)+PM
670 RP=RP-PM
680 NEXTZ
690 NEXTX
700 M1=PT(1)+IT(1):M2=PT(2)+IT(2)
710 CF=OI-M1-M2
720 PP=PT(1)+PT(2)
730 RN=CF+PP
740 EQ=G-MT
750 RT=RN/EQ*100
760 DP=D(1)*G/DY(1)*DV(DP(1))+D(2)*G/DY(2)*DV(DP(2))
770 TI=RN-DP
780 PU=G/UN
790 PF=G/SF
800 CR=OI/G*100
810 MR=G/(GI-VS)
1000 CLS
1010 PRINT"COMPUTED GROSS";TAB(17);USINGA4$;GI;
1020 PRINTTAB(33)"MTG PPM";TAB(49);"#1";TAB(59);"#2"
1030 PRINT"VAC ALLOWANCE";USINGA3$;VA;:PRINT"% ";USINGA4$;VS;
1040 PRINTTAB(33);"PRINCIPAL ";USINGA1$;PT(1);:PRINTUSINGA1$;PT(
2)
1050 PRINT"EXP. RATIO ";USINGA2$;ER;:PRINT"%";USINGA4$;EV;
1060 PRINTTAB(33);"INTEREST  ";USINGA1$;IT(1);:PRINTUSINGA1$;IT(
2)
1070 PRINT"OPERATING INCOME";TAB(17);USINGA4$;OI;:PRINTTAB(33);"
TOTAL";TAB(43);USINGA1$;M1;:PRINTUSINGA1$;M2
1080 PRINTSTRING$(64,131);
1090 PRINT"CASH FLOW";TAB(20);USINGA1$;CF;:PRINTTAB(33);"PURCHAS
E PRICE";TAB(50);USINGA4$;G;
1100 PRINT"PRINCIPLE PAYMENT";TAB(20);USINGA1$;PP;
1110 PRINTTAB(33);"MORTGAGE TOTAL";TAB(50);USINGA4$;MT;
1120 PRINT"EQU RTN = ";USINGA2$;RT;:PRINT"%";TAB(20);USINGA1$;RN
;
1130 PRINTTAB(33);"EQUITY";TAB(50);USINGA4$;EQ;
1140 PRINT"DEPRECIATION";TAB(20);USINGA1$;DP;
1150 PRINTTAB(33);"% RETURN ON INVESTED $";TAB(58);USINGA2$;CR;:
PRINT"%";
1160 PRINT"TAXABLE INCOME";TAB(20);USINGA1$;TI;
1170 PRINTTAB(33);"MULTIPLIER (X EARNINGS)";TAB(58);USINGA2$;MR
1180 PRINTSTRING$(64,131);:FORY=0TO9:PRINT@Y*64+31,CHR$(191);:NE
XTY
1190 PRINT@704,"PRICE PER UNIT =";TAB(17);USINGA4$;PU;
1200 PRINTTAB(33);"PRICE PER SQ FT =";TAB(53);USINGA1$;PF
1210 PRINTSTRING$(64,131);
1220 PRINT"  1 - SHOW SUMMARY";:GOSUB3020
1230 IFF=2THEN2000
1240 CLS:PRINT"PRICE";TAB(30);USINGA4$;G
1250 PRINT"GROSS INCOME";TAB(30);USINGA4$;GI
1260 PRINT:PRINT"DEPRECIATION";TAB(20);"% ALLOC.";TAB(30);"YEARS
";TAB(39);"TYPE"
1270 FORJ=1TO2
1280 PRINTD$(J);TAB(20);D(J);TAB(30);DY(J);TAB(39);DT$(DP(J))
1290 NEXTJ
1300 IFMN=0THENPRINT:PRINT"THERE ARE NO MORTGAGES":GOTO1360
1310 PRINT:FORX1=1TOMN
1320 IFX1=1ANDMN=1THENPRINT"MORTGAGE";ELSEPRINT"MTG. #";MID$(STR
$(X1),2);
1330 PRINT" IS $";:A=AM(X1):GOSUB110:PRINT" AT";RT(X1);CHR$(24);
"% ";
1340 IFT(X1)=12000THENPRINT"(FIRST YEAR INTEREST ONLY.)"ELSEPRIN
T"OVER";T(X1);"MONTHS."
1350 NEXTX1
1360 PRINT:PRINT"THE BUILDING HAS";UN;"UNITS AND";SF;"SQUARE FEE
T."
```

```
1370 PRINT:PRINT:PRINT"1 - DISPLAY FIRST YEAR";:GOSUB3020
1380 IFF=1THEN1000
2000 CLS
2010 PRINT:PRINT"DO YOU WISH TO CHANGE:":PRINT
2020 PRINTTAB(5);"1 - COMPUTED GROSS INCOME"
2030 PRINTTAB(5);"2 - VACANCY ALLOWANCE"
2040 PRINTTAB(5);"3 - EXPENSE RATIO"
2050 PRINTTAB(5);"4 - PURCHASE PRICE"
2060 PRINTTAB(5);"5 - MORTGAGE INFORMATION"
2070 PRINTTAB(5);"6 - DEPRECIATION INFORMATION"
2080 PRINT@640,CHR$(31);"ENTER THE NUMBER OF YOUR CHOICE ";:INPU
TX
2090 X=INT(X):IFX<1ORX>6THEN2080
2100 PRINT:ONXGOSUB3120,3060,3140,3050,3170,3350
2110 GOTO500
3000 INPUTP:IFP<1THENP=P*100
3010 RETURN
3020 PRINT"   2 - CHANGE A VALUE   3 - RERUN PROGRAM"
3030 PRINT@896,CHR$(31);TAB(14);"ENTER THE NUMBER OF YOUR CHOICE
: ";:INPUTA$:F=INT(VAL(A$))
3040 IFF=0THEN32767ELSEIFF<1ORF>3THEN3030ELSEIFF=3THENRUNELSERET
URN
3050 PRINT"WHAT IS THE PURCHASE PRICE OF THE PROPERTY";TAB(Z1);:
INPUTG:IFG<=0THENPRINTBU$;:GOTO3050ELSERETURN
3060 PRINT"WHAT IS THE VACANCY ALLOWANCE, IN %";TAB(Z1);
3070 P=VA:GOSUB3000:IFP<=0THENPRINTBU$;:GOTO3060ELSEVA=P:RETURN
3080 PRINT"HOW MANY UNITS DOES THE PROPERTY HAVE";TAB(Z1);
3090 INPUTUN:IFUN<=0THENPRINTBU$;:GOTO3080
3100 PRINT"HOW MANY SQUARE FEET OF USABLE SPACE";TAB(Z1);
3110 INPUTSF:IFSF<=0THENPRINTBU$;:GOTO3100ELSERETURN
3120 PRINT"YEARLY GROSS INCOME (ASSUMING 100% OCCUPANCY)";TAB(Z1
);:INPUTGI:IFGI<=0THENPRINTBU$;:GOTO3120ELSERETURN
3130 PRINT
3140 PRINT"WHAT IS THE EXPECTED EXPENSE RATIO? THAT IS, THE"
3150 PRINT"YEARLY EXPENSES DIVIDED BY GROSS INCOME, IN %";TAB(Z1
);
3160 P=ER:GOSUB3000:IFP<=0THENPRINTBU$;:GOTO3150ELSEER=P:PRINT:R
ETURN
3170 CLS:PRINT"HOW MANY MORTGAGES WILL THERE BE, 0, 1, OR 2";TAB
(Z1);
3180 INPUTMN:MN=INT(MN):IFMN<0ORMN>2THEN3170
3190 IFMN<>0THENPRINT"FOR INTEREST ONLY OR BALLOON MORTGAGES, TE
RM = 1000 YEARS"
3200 MT=0
3210 FORJ=1TO2
3220 AM(J)=0:T(J)=0:RT(J)=0:PT(J)=0
3230 IFJ>MNTHEN3320
3240 PRINT:PRINT"PLEASE GIVE THE FOLLOWING";:IFMN<>1THENPRINT" F
OR MORTGAGE NUMBER ";MID$("ONETWO",J*3-2,3);
3250 PRINT":"
3260 PRINT"TOTAL AMOUNT OF THE MORTGAGE";TAB(Z2);:INPUTAM(J):IFA
M(J)<=0THENPRINTBU$;:GOTO3260
3270 IFAM(1)>=GTHENPRINT:PRINT"THE VALUE OF THE MORTGAGE":GOTO33
30
3280 IFMN=2ANDJ=2ANDAM(1)+AM(2)>=GTHENPRINT:PRINT"THE SUM OF THE
 MORTGAGE VALUES":GOTO3330
3290 PRINT"THE TERM, IN YEARS";TAB(Z2);:INPUTT(J):IFT(J)<=0THENP
RINTBU$;:GOTO3290ELSET(J)=T(J)*12
3300 PRINT"THE INTEREST RATE, IN %";TAB(Z2);
3310 P=RT(J):GOSUB3000:IFP<=0THENPRINTBU$;:GOTO3300ELSERT(J)=P:M
T=MT+AM(J)
3320 NEXTJ:RETURN
3330 PRINT"MUST BE LESS THEN THE PURCHASE PRICE OF $";:A=G:GOSUB
110:PRINT:PRINT:PRINT"PLEASE PRESS ENTER";:INPUTX$:GOTO3170
3340 RETURN
3350 CLS:PRINT"DESCRIBE DEPRECIATION METHODS"
3360 PRINT:FORJ=1TO2
3370 PRINT"ALLOCATION TO ";D$(J);" IN %";TAB(Z3);
3380 P=D(J):GOSUB3000:IFP<0THENPRINTBU$;:GOTO3370ELSED(J)=P:NEXT
J
3390 IFD(1)+D(2)>100THENPRINT"THE SUM OF THESE ALLOCATIONS CAN'T
 EXCEED 100%.  TRY AGAIN.":GOTO3360
3400 FORJ=1TO2:PRINT
3410 PRINT"WHAT METHOD OF DEPRECIATION IS TO BE USED FOR THE"
3420 PRINTD$(J);"?  INPUT SL, DDB, 125%, OR 150%";TAB(Z3);
3430 INPUTA$:IFA$=""ANDDP(J)<>0THEN3450ELSEGOSUB100:C$=LEFT$(C$,
3)
3440 IFC$="SL"THENDP(J)=1ELSEIFC$="DDB"THENDP(J)=2ELSEIFC$="125"
THENDP(J)=3ELSEIFC$="150"THENDP(J)=4ELSEPRINT"ILLEGAL METHOD.  P
LEASE ENTER SL, DDB, 125%, OR 150% ";:GOTO3430
3450 PRINT"TERM OF ";D$(J);" DEPRECIATION, IN YEARS";TAB(Z3);:IN
PUTDY(J):IFDY(J)<=0THENPRINTBU$;:GOTO3450
3460 NEXTJ:RETURN
32767 CLS:END
```

SWAT TABLE

For TRS-80®
FINANCIAL OPERATING SYSTEM

LINES	SWAT CODE	LENGTH
10 - 270	LI	429
280 - 510	ZW	293
520 - 630	ZD	323
640 - 750	RN	192
760 - 1050	RA	338
1060 - 1170	GO	538
1180 - 1290	UR	382
1300 - 2020	JV	436
2030 - 3020	PV	371
3030 - 3120	UR	527
3130 - 3240	LF	488
3250 - 3350	PJ	519
3360 - 3450	EC	581
3460 - 32767	WK	17

```
SS SS SS SS SS SS SS SS SS
                        SS
IBM-PC BASIC            SS
SS 'Financial Operating System'  SS
SS     Author:  J. M. Keynes     SS
SS   Translator: Rich Bouchard   SS
SS      Copyright © 1983         SS
SS  SoftSide Publications, Inc   SS
SS                               SS
SS SS SS SS SS SS SS SS SS SS SS SS

10 GOTO 200
100 C$="":IF A$="" THEN RETURN ELSE FOR
J2=1 TO LEN(A$):J1=ASC(MID$(A$,J2,1)):C$
=C$+CHR$(J1+32*(J1)=97 AND J1(=121)):NEX
T J2:RETURN
110 PRINT MID$(STR$(INT(A+FH)),2);USING"
.##";A-INT(A)+FH;:RETURN
120 LOCATE CL:PRINT SP$:LOCATE CL:RETURN
130 LOCATE CL:PRINT SP$;SP$:LOCATE CL:RE
TURN
200 CLEAR:DEFDBL A-E,G-H,I-W:DEFINT F,J,
X,Z
210 WIDTH 40:KEY OFF:COLOR 7,0
220 DV(1)=.02:DV(2)=.01:DV(3)=.0125:DV(4
)=.015
230 DT$(1)="SL":DT$(2)="DDB":DT$(3)="125
%":DT$(4)="150%"
240 D$(1)="BUILDING":D$(2)="PERSONAL PRO
PERTY"
250 FH=5/1000
260 SP$=STRING$(39,32)
270 A1$="###.##"
280 A2$="##,###,###.##-"
400 CLS:PRINT"When you are asked for a p
ercentage":PRINT"you may enter it as eit
her an integer":PRINT"or a decimal (24 o
r .24, for example).":PRINT
```

↪

Financial Operating System, *continued*

```
410 GOSUB 3050:PRINT
420 GOSUB 3070:PRINT
430 GOSUB 3090:PRINT
440 GOSUB 3140:PRINT
450 GOSUB 3160:PRINT
460 GOSUB 3190:PRINT
470 GOSUB 3400:PRINT
500 CLS
510 VS=VA*GI/100
520 EV=ER*(GI-VS)/100
530 OI=GI-VS-EV
540 M1=0:M2=0:PT(1)=0:PT(2)=0:IT(1)=0:IT
(2)=0
550 IF MN=0 THEN 710
560 FOR X=1 TO MN
570 IF T(X)=12000 THEN IT(X)=AM(X)*(RT(X
)/100):GOTO 690
580 I(X)=RT(X)/1200
590 T=(I(X)+1)^(-T(X)):T=1-T:MP(X)=I(X)*
AM(X)/T
600 MP(X)=INT(MP(X)*100+.5)/100
610 IT(X)=0:PT(X)=0:RP=AM(X)
620 FOR Z=1 TO 12
630 IM=INT(RP*I(X)*100+.5)/100
640 PM=MP(X)-IM
650 IT(X)=IT(X)+IM
660 PT(X)=PT(X)+PM
670 RP=RP-PM
680 NEXT Z
690 NEXT X
700 M1=PT(1)+IT(1):M2=PT(2)+IT(2)
710 CF=OI-M1-M2
720 PP=PT(1)+PT(2)
730 RN=CF+PP
740 EQ=G-MT
750 RT=RN/EQ*100
760 DP=D(1)*G/DY(1)*DV(DP(1))+D(2)*G/DY(
2)*DV(DP(2))
770 TI=RN-DP
780 PU=G/UN
790 PF=G/SF
800 CR=OI/G*100
810 MR=G/(GI-VS)
1000 CLS:PRINT"Computed Gross";TAB(25);U
SING A2$;GI
1010 PRINT"Vac Allowance ";USING A1$;VA;
:PRINT"%";TAB(25);USING A2$;VS
1020 PRINT"Expense Ratio ";USING A1$;ER;
:PRINT"%";TAB(25);USING A2$;EV
1030 PRINT"Operating Income";TAB(25);USI
NG A2$;OI
1040 PRINT STRING$(39,196)
1050 PRINT"Mtg PPM";TAB(19);"#1";TAB(33)
;"#2"
1060 PRINT"Principal";TAB(11);USING A2$;
PT(1);:PRINT USING A2$;PT(2)
1070 PRINT"Interest";TAB(11);USING A2$;I
T(1);:PRINT USING A2$;IT(2)
1080 PRINT"Total";TAB(11);USING A2$;M1;:
PRINT USING A2$;M2
1090 PRINT STRING$(39,196)
1100 PRINT"Cash Flow";TAB(25);USING A2$;
CF
1110 PRINT"Principle Payment";TAB(25);US
ING A2$;PP
1120 PRINT"Equ Rtn = ";USING A1$;RT;:PRI
NT"%";TAB(25);USING A2$;RN
1125 PRINT"Depreciation";TAB(25);USING A
2$;DP
1130 PRINT"Taxable Income";TAB(25);USING
 A2$;TI
1140 PRINT"Purchase Price";TAB(25);USING
 A2$;G
1150 PRINT"Mortgage Total";TAB(25);USING
 A2$;MT
1160 PRINT"Equity";TAB(25);USING A2$;EQ
1170 PRINT"% Return on Invested $";TAB(3
2);USING A1$;CR;:PRINT"%"
1180 PRINT"Multiplier (X earnings)";TAB(
32);USING A1$;MR
1190 PRINT"Price per unit =";TAB(25);USI
NG A2$;PU
1200 PRINT"Price per sq foot =";TAB(25);
USING A2$;PF
1210 COLOR 0,7
1220 PRINT"1-Summary";:GOSUB 3020
1230 IF F=2 THEN 2000
1240 CLS:LOCATE 3:PRINT"Price:";TAB(21);
USING A2$;G
1250 PRINT"Gross Income:";TAB(21);USING
A2$;GI
1260 PRINT:PRINT"Depreciation";TAB(19);"
% Alloc.";TAB(29);"Years";TAB(36);"Type"

1270 PRINT STRING$(39,196)
1280 FOR J=1 TO 2
1290 PRINT D$(J);TAB(20);D(J);TAB(29);DY
(J);TAB(36);DT$(DP(J))
1300 NEXT J
1310 IF MN=0 THEN PRINT:PRINT"There are
no mortgages":PRINT:GOTO 1370
1320 PRINT:FOR X1=1 TO MN
1330 PRINT"Mortgage";:IF MN=2 THEN PRINT
" #";MID$(STR$(X1),2);
1340 PRINT" is $";:A=AM(X1):GOSUB 110:PR
INT" at";RT(X1);CHR$(29);"%"
1350 IF T(X1)=12000 THEN PRINT"(First ye
ar interest only)." ELSE PRINT"over";T(X
1);"months."
1360 PRINT:NEXT X1
1370 PRINT"The building has";UN;"units":
PRINT"and";SF;"square feet."
1380 PRINT:COLOR 0,7:PRINT"1-First year"
;:GOSUB 3020
1390 IF F=1 THEN 1000
2000 CLS
2010 PRINT:PRINT"Do you wish to change:"
:PRINT
2020 PRINT TAB(5);"1 - Computed Gross In
come"
2030 PRINT TAB(5);"2 - Vacancy Allowance
"
2040 PRINT TAB(5);"3 - Expense Ratio"
2050 PRINT TAB(5);"4 - Purchase Price"
2060 PRINT TAB(5);"5 - Mortgage Informat
ion"
2070 PRINT TAB(5);"6 - Depreciation Info
rmation"
2080 PRINT:PRINT"Enter the number of you
r choice ";:INPUT X
2090 X=INT(X):IF X<1 OR X>6 THEN 2000
2100 PRINT:ON X GOSUB 3140,3070,3160,305
0,3190,3400
2110 GOTO 500
3000 INPUT P:IF P<1 THEN P=P*100
3010 RETURN
3020 PRINT"  2-Change value  3-Rerun"
3030 CL=CSRLIN:PRINT"Enter the number of
 your choice: ";:INPUT A$:F=INT(VAL(A$))

3040 IF F<0 OR F>3 THEN GOSUB 120:GOTO 3
030 ELSE COLOR 7,0:IF F=0 THEN 32767 ELS
E IF F=3 THEN RUN ELSE RETURN
3050 CL=CSRLIN:PRINT"What is the purchas
e price of the":INPUT"property";G
3060 IF G<=0 THEN GOSUB 130:GOTO 3050 EL
SE RETURN
3070 CL=CSRLIN:PRINT"What is the vacancy
 allowance,":PRINT"in percent";
3080 P=VA:GOSUB 3000:IF P<=0 THEN GOSUB
130:GOTO 3070 ELSE VA=P:RETURN
3090 CL=CSRLIN:PRINT"How many units does
 the":INPUT"property have";UN
3100 IF UN<=0 THEN GOSUB 130:GOTO 3090
3110 PRINT
3120 CL=CSRLIN:PRINT"How many square fee
t of":INPUT"usable space";SF
3130 IF SF<=0 THEN GOSUB 130:GOTO 3120 E
LSE RETURN
3140 CL=CSRLIN:PRINT"Yearly gross income
 (assuming":INPUT"100% occupancy)";GI
3150 IF GI<=0 THEN GOSUB 130:GOTO 3140 E
LSE RETURN
3160 PRINT"What is the expected expense
ratio?":PRINT"That is, the yearly espens
es divided"
3170 CL=CSRLIN:PRINT"by gross income, in
 percent";
3180 P=ER:GOSUB 3000:IF P<=0 THEN GOSUB
130:GOTO 3170 ELSE ER=P:RETURN
3190 CLS:PRINT:PRINT"How many mortgages
will there be,":INPUT"(0, 1, or 2) ";MN
3200 MN=INT(MN):IF MN<0 OR MN>2 THEN 319
0
3210 IF MN=0 THEN RETURN
3220 PRINT:PRINT"For interest only or ba
lloon":PRINT"mortgages, term = 1000 year
s"
```

```
3230 MT=0
3240 FOR J=1 TO 2
3250 AM(J)=0:T(J)=0:RT(J)=0:PT(J)=0
3260 IF J>MN THEN 3370
3270 PRINT:PRINT"Please give the followi
ng";:IF MN<>1 THEN PRINT:PRINT"for mortg
age number ";MID$("onetwo",J*3-2,3);
3280 PRINT":":PRINT
3290 CL=CSRLIN:INPUT"Total amount of the
 mortgage";AM(J)
3300 IF AM(J)<=0 THEN GOSUB 120:GOTO 329
0
3310 IF AM(1)>G THEN PRINT:PRINT"The va
lue of the mortgage":GOTO 3380
3320 IF J=2 AND AM(1)+AM(2)>=G THEN PRIN
T:PRINT"The sum of the mortgage values":
GOTO 3380
3330 CL=CSRLIN:INPUT"The term, in years"
;T(J)
3340 IF T(J)<=0 THEN GOSUB 120:GOTO 3330
 ELSE T(J)=T(J)*12
3350 CL=CSRLIN:PRINT"The interest rate,
in percent";
3360 P=RT(J):GOSUB 3000:IF P<=0 THEN GOS
UB 120:GOTO 3350 ELSE RT(J)=P:MT=MT+AM(J
)
3370 NEXT J:RETURN
3380 PRINT"must be less than the purchas
e":PRINT"price of $";:A=G:GOSUB 110
3390 PRINT:PRINT:PRINT"Please hit RETURN
";:INPUT X$:GOTO 3190
3400 CLS:PRINT"Describe depreciation met
hods"
3410 PRINT:FOR J=1 TO 2
3420 CL=CSRLIN:PRINT"Allocation to ";D$(
J);:PRINT"in percent";
3430 P=D(J):GOSUB 3000:IF P<0 THEN GOSUB
 120:GOTO 3420 ELSE D(J)=P
3440 PRINT:NEXT J
3450 IF D(1)+D(2)>100 THEN PRINT"The sum
 of these allocations can't":PRINT"excee
d 100%.  Try again.":GOTO 3410
3460 FOR J=1 TO 2
3470 PRINT"What method of depreciation i
s":PRINT"to be used for the ";D$(J);"?"
3480 CL=CSRLIN:PRINT"Input SL, DDB, 125%
, OR 150% ";
3490 INPUT A$:IF A$="" AND DP(J)<>0 THEN
 3510 ELSE GOSUB 100:C$=LEFT$(C$,3)
3500 IF C$="SL" THEN DP(J)=1 ELSE IF C$=
"DDB" THEN DP(J)=2 ELSE IF C$="125" THEN
 DP(J)=3 ELSE IF C$="150" THEN DP(J)=4 E
LSE GOSUB 120:GOTO 3480
3510 CL=CSRLIN:PRINT:PRINT"Term of ";D$(
J);:PRINT"depreciation, in years";
3520 INPUT DY(J):IF DY(J)<=0 THEN GOSUB
130:GOTO 3510
3530 PRINT:NEXT J:RETURN
32767 CLS:END
```

SWAT TABLE

For IBM® PC

FINANCIAL OPERATING SYSTEM

LINES	SWAT CODE	LENGTH
10 - 260	LS	441
270 - 510	DF	278
520 - 630	WB	335
640 - 750	GZ	193
760 - 1050	CC	350
1060 - 1160	TK	437
1170 - 1280	AU	384
1290 - 2000	SC	430
2010 - 3000	YO	392
3010 - 3120	UI	514
3130 - 3240	RW	504
3250 - 3360	OY	544
3370 - 3480	UO	495
3490 - 32767	AL	281

```
10 TRAP 5000:GOTO 200
100 IF Z$="" THEN RETURN
110 FOR J2=1 TO LEN(Z$):J1=ASC(Z$(J2,J
2)):Z$(J2,J2)=CHR$(J1-32*(J1>=97 AND J
1<=121)):NEXT J2:RETURN
120 POSITION 2,CL:PRINT CHR$(156);CHR$
(156);CHR$(156);CHR$(156);:RETURN
200 DIM DT$(16),D$(34),SP$(40),LINE$(3
7),SG$(1),Z$(20)
202 DIM DV(4),AM(2),T(2),RT(2),PT(2),I
T(2),D(2),DP(2),DY(2),I(2),MP(2)
210 DV(1)=0.02:DV(2)=0.01:DV(3)=0.0125
:DV(4)=0.015
220 DT$="SL  DDB 125%150%"
230 D$="BUILDING123456789PERSONAL PROP
ERTY"
231 FOR J1=9 TO 17:D$(J1,J1)=CHR$(254)
:NEXT J1
240 FH=5/1000:TAB=85
250 SP$(1)=" ":SP$(40)=" ":SP$(2)=SP$(
1)
260 LINE$(1)=CHR$(18):LINE$(37)=CHR$(1
8):LINE$(2)=LINE$(1)
400 GRAPHICS 0:PRINT "Note: When you a
re asked for a":PRINT "percentage, you
 may enter it as"
410 PRINT "either an integer or a deci
mal";:PRINT "(24 or .24, for example)."
:PRINT
420 GOSUB 3080:PRINT
```

```
SS SS SS SS SS SS SS SS SS
                        SS
      Atari BASIC       SS
SS 'Financial Operating System'  SS
SS     Author:  J. M. Keynes     SS
SS   Translator: Rich Bouchard   SS
SS      Copyright © 1983         SS
SS  SoftSide Publications, Inc   SS
SS                               SS
SS SS SS SS SS SS SS SS SS SS SS SS
```

Financial Operating System, ***continued***

```
430 GOSUB 3110:PRINT
440 GOSUB 3140:PRINT
450 GOSUB 3200:PRINT
460 GOSUB 3230:PRINT
470 GOSUB 3270:PRINT
480 GOSUB 3490:PRINT
500 GRAPHICS 0
510 VS=VA*GI/100
520 EV=ER*(GI-VS)/100
530 OI=GI-VS-EV
540 M1=0:M2=0:PT(1)=0:PT(2)=0:IT(1)=0:
IT(2)=0
550 IF MN=0 THEN 710
560 FOR X=1 TO MN
570 IF T(X)=12000 THEN IT(X)=AM(X)*(RT
(X)/100):GOTO 690
580 I(X)=RT(X)/1200
590 T=(I(X)+1)^(-T(X)):T=1-T:MP(X)=I(X
)*AM(X)/T
600 MP(X)=INT(MP(X)*100+0.5)/100
610 IT(X)=0:PT(X)=0:RP=AM(X)
620 FOR Z=1 TO 12
630 IM=INT(RP*I(X)*100+0.5)/100
640 PM=MP(X)-IM
650 IT(X)=IT(X)+IM
660 PT(X)=PT(X)+PM
670 RP=RP-PM
680 NEXT Z
690 NEXT X
700 M1=PT(1)+IT(1):M2=PT(2)+IT(2)
710 CF=OI-M1-M2
720 PP=PT(1)+PT(2)
730 RN=CF+PP
740 EQ=G-MT
750 RT=RN/EQ*100
760 DP=D(1)*G/DY(1)*DV(DP(1))+D(2)*G/D
Y(2)*DV(DP(2))
770 TI=RN-DP
780 PU=G/UN
790 PF=G/SF
800 CR=OI/G*100
810 MR=G/(GI-VS)
1000 GRAPHICS 0:NW=8:PRINT "Computed G
ross";:POKE TAB,25:Z1=GI:GOSUB 3700
1010 PRINT :PRINT "Vac Allowance ";:Z1
=VA:NW=3:GOSUB 3700:PRINT "%";:POKE TA
B,25:NW=8:Z1=VS:GOSUB 3700
1020 PRINT :PRINT "Expense Ratio ";:NW
=3:Z1=ER:GOSUB 3700:PRINT "%";:POKE TA
B,25:NW=8:Z1=EV:GOSUB 3700
1030 PRINT :PRINT "Operating Income";:
POKE TAB,25:Z1=OI:GOSUB 3700
1040 PRINT :PRINT LINE$
1050 PRINT "Mtg PPM";:POKE TAB,18:PRIN
T "#1";:POKE TAB,32:PRINT "#2"
1060 PRINT "Principal";:POKE TAB,11:Z1
=PT(1):GOSUB 3700:POKE TAB,25:Z1=PT(2)
:GOSUB 3700
1070 PRINT :PRINT "Interest";:POKE TAB
,11:Z1=IT(1):GOSUB 3700:POKE TAB,25:Z1
=IT(2):GOSUB 3700
1080 PRINT :PRINT "Total";:POKE TAB,11
:Z1=M1:GOSUB 3700:POKE TAB,25:Z1=M2:GO
SUB 3700
1090 PRINT :PRINT LINE$
1100 PRINT "Cash Flow";:POKE TAB,25:Z1
=CF:GOSUB 3700
1110 PRINT :PRINT "Principle Payment";
:POKE TAB,25:Z1=PP:GOSUB 3700
1120 PRINT :PRINT "Equ Rtn = ";:NW=3:Z
1=RT:GOSUB 3700:PRINT "%";:POKE TAB,25
:NW=8:Z1=RN:GOSUB 3700
1125 PRINT :PRINT "Depreciation";:POKE
 TAB,25:Z1=DP:GOSUB 3700
1130 PRINT :PRINT "Taxable Income";:PO
KE TAB,25:Z1=TI:GOSUB 3700
1140 PRINT :PRINT "Purchase Price";:PO
KE TAB,25:Z1=G:GOSUB 3700
1150 PRINT :PRINT "Mortgage Total";:PO
KE TAB,25:Z1=MT:GOSUB 3700
1160 PRINT :PRINT "Equity";:POKE TAB,2
5:Z1=EQ:GOSUB 3700
1170 PRINT :PRINT "% Return on Investe
d $";:POKE TAB,30:NW=3:Z1=CR:GOSUB 370
0:PRINT "%"
1180 PRINT "Multiplier (X earnings)";:
POKE TAB,30:Z1=MR:GOSUB 3700
1190 PRINT :PRINT "Price per unit =";:
POKE TAB,25:NW=8:Z1=PU:GOSUB 3700
1200 PRINT :PRINT "Price per sq foot =
";:POKE TAB,25:Z1=PF:GOSUB 3700
1210 PRINT
1220 PRINT "1-Summary";:GOSUB 3020
1230 IF F=2 THEN 2000
1240 GRAPHICS 0:POSITION 2,3:PRINT "Pr
ice:";:POKE TAB,21:NW=8:Z1=G:GOSUB 370
0
1250 PRINT :PRINT "Gross Income:";:POK
E TAB,21:Z1=GI:GOSUB 3700
1260 PRINT :PRINT :PRINT "Depreciation
";:POKE TAB,18:PRINT "% Alloc.";
1270 POKE TAB,28:PRINT "Years";:POKE T
AB,35:PRINT "Type"
1280 PRINT LINE$
1290 FOR J=1 TO 2
1300 PRINT D$(J*17-16,J*17);:POKE TAB,
20:PRINT D(J);:POKE TAB,29:PRINT DY(J)
;
1310 POKE TAB,35:PRINT DT$(DP(J)*4-3,D
P(J)*4)
1320 NEXT J
1330 IF MN=0 THEN PRINT :PRINT "There
are no mortgages":PRINT :GOTO 1400
1340 PRINT :FOR X1=1 TO MN
1350 PRINT "Mortgage";:IF MN=2 THEN PR
INT " #";X1;
1360 PRINT " is $";:NW=0:Z1=AM(X1):GOS
UB 3700:PRINT " at ";RT(X1);"%"
1370 IF T(X1)=12000 THEN PRINT "(First
 year interest only).":GOTO 1390
1380 PRINT "over ";T(X1);" months."
1390 PRINT :NEXT X1
1400 PRINT "The building has ";UN;" un
its":PRINT "and ";SF;" square feet."
1410 PRINT :PRINT "1-First year";:GOSU
B 3020
1420 IF F=1 THEN 1000
2000 GRAPHICS 0
2010 PRINT :PRINT "Do you wish to chan
ge:";PRINT
2020 POKE TAB,5:PRINT "1 - Computed Gr
oss Income"
2030 POKE TAB,5:PRINT "2 - Vacancy All
owance"
2040 POKE TAB,5:PRINT "3 - Expense Rat
io"
2050 POKE TAB,5:PRINT "4 - Purchase Pr
ice"
2060 POKE TAB,5:PRINT "5 - Mortgage In
formation"
2070 POKE TAB,5:PRINT "6 - Depreciatio
n Information"
2080 PRINT :PRINT "Enter the number of
 your choice ";:GOSUB 2500:X=Z
2090 X=INT(X):IF X<1 OR X>6 THEN 2000
2100 PRINT :ON X GOSUB 3200,3110,3230,
3080,3270,3490
2110 GOTO 500
2500 COLUMN=PEEK(TAB):ROW=PEEK(84)
2510 POSITION COLUMN,ROW:INPUT Z:RETUR
N
3000 GOSUB 2500:P=Z:IF P<1 THEN P=P*10
0
3010 RETURN
3020 PRINT "  2-Change value  3-Rerun"
3030 CL=PEEK(84):PRINT "Enter the numb
er of your choice: ";:GOSUB 2500:F=Z
3040 IF F<0 OR F>3 THEN GOSUB 120:GOTO
 3030
3050 IF F=0 THEN 6000
3060 IF F=3 THEN RUN
3070 RETURN
3080 CL=PEEK(84):PRINT "What is the pu
rchase price of the":PRINT "property";
:GOSUB 2500:G=Z
3090 IF G<=0 THEN GOSUB 120:GOTO 3080
3100 RETURN
3110 CL=PEEK(84):PRINT "What is the va
cancy allowance,":PRINT "in percent";
3120 P=VA:GOSUB 3000:IF P<=0 THEN GOSU
B 120:GOTO 3110
3130 VA=P:RETURN
3140 CL=PEEK(84):PRINT "How many units
 does the":PRINT "property have";:GOSU
B 2500:UN=Z
3150 IF UN<=0 THEN GOSUB 120:GOTO 3140
3160 PRINT
```

```
3170 CL=PEEK(84):PRINT "How many squar
e feet of":PRINT "usable space";:GOSUB
 2500:SF=Z
3180 IF SF<=0 THEN GOSUB 120:GOTO 3170
3190 RETURN
3200 CL=PEEK(84):PRINT "Yearly gross i
ncome (assuming":PRINT "100% occupancy
)";:GOSUB 2500:GI=Z
3210 IF GI<=0 THEN GOSUB 120:GOTO 3200
3220 RETURN
3230 PRINT "What is the expected expen
se ratio?":PRINT "That is, the yearly
espenses divided"
3240 CL=PEEK(84):PRINT "by gross incom
e, in percent";
3250 P=ER:GOSUB 3000:IF P<=0 THEN GOSU
B 120:GOTO 3240
3260 ER=P:RETURN
3270 GRAPHICS 0:PRINT "How many mortga
ges will there be,":PRINT "(0, 1, or 2
) ";:GOSUB 2500:MN=Z
3280 MN=INT(MN):IF MN<0 OR MN>2 THEN 3
270
3290 IF MN<>0 THEN PRINT :PRINT "For i
nterest only or balloon":PRINT "mortga
ges, term = 1000 years"
3300 MT=0
3310 FOR J=1 TO 2
3320 AM(J)=0:T(J)=0:RT(J)=0:PT(J)=0
3330 IF J>MN THEN 3460
3340 PRINT :PRINT "Please give the fol
lowing";:IF MN<>1 THEN PRINT :PRINT "f
or mortgage number ";J;
3350 PRINT ":":PRINT
3360 CL=PEEK(84):PRINT "Total amount o
f the mortgage";:GOSUB 2500:AM(J)=Z
3370 IF AM(J)<=0 THEN GOSUB 120:GOTO 3
360
3380 IF AM(1)>=G THEN PRINT :PRINT "Th
e value of the mortgage":GOTO 3470
3390 IF J=2 AND AM(1)+AM(2)>=G THEN PR
INT :PRINT "The sum of the mortgage va
lues":GOTO 3470
3400 CL=PEEK(84):PRINT "The term, in y
ears";:GOSUB 2500:T(J)=Z
3410 IF T(J)<=0 THEN GOSUB 120:GOTO 34
00
3420 T(J)=T(J)*12
3430 CL=PEEK(84):PRINT "The interest r
ate, in percent";
3440 P=RT(J):GOSUB 3000:IF P<=0 THEN G
OSUB 120:GOTO 3430
3450 RT(J)=P:MT=MT+AM(J)
3460 NEXT J:RETURN
3470 PRINT "must be less than the purc
hase":PRINT "price of $";:NW=0:Z1=G:GO
SUB 3700:PRINT :PRINT
3480 PRINT "Please press RETURN";:INPU
T Z$:GOTO 3270
3490 GRAPHICS 0:PRINT "Describe deprec
iation methods"
3500 PRINT :FOR J=1 TO 2
3510 CL=PEEK(84):PRINT "Allocation to
";D$(J*17-16,J*17):PRINT "in percent";
3520 P=D(J):GOSUB 3000:IF P<0 THEN GOS
UB 120:GOTO 3510
3530 D(J)=P
3540 PRINT :NEXT J
3550 IF D(1)+D(2)<=100 THEN 3570
3560 PRINT "The sum of these allocatio
ns can't":PRINT "exceed 100%.  Try aga
in.":GOTO 3500
3570 FOR J=1 TO 2
3580 PRINT "What method of depreciatio
n is":PRINT "to be used for the ";D$(J
*17-16,J*17);"?"
3590 CL=PEEK(84):PRINT "Input SL, DDB,
 125%, OR 150% ";
3600 INPUT Z$:IF Z$="" AND DP(J)<>0 TH
EN 3670
3610 GOSUB 100:Z$(LEN(Z$)+1)="   ":Z$=
Z$(1,3)
3620 IF Z$="SL " THEN DP(J)=1:GOTO 367
0
3630 IF Z$="DDB" THEN DP(J)=2:GOTO 367
0
3640 IF Z$="125" THEN DP(J)=3:GOTO 367
0
3650 IF Z$="150" THEN DP(J)=4:GOTO 367
0
3660 GOSUB 120:GOTO 3590
3670 CL=PEEK(84):PRINT :PRINT "Term of
 ";D$(J*17-16,J*17):PRINT "depreciatio
n, in years";
3680 GOSUB 2500:DY(J)=Z:IF DY(J)<=0 TH
EN GOSUB 120:GOTO 3670
3690 PRINT :NEXT J:RETURN
3700 SG$="":IF Z1<0 THEN Z1=ABS(Z1):SG
$="-"
3705 Z1=INT(Z1*100+0.5)/100:Z2=INT((Z1
-INT(Z1))*100)
3710 Z$=STR$(INT(ABS(Z1))):Z$(LEN(Z$)+
1)="."
3720 SP=0:IF NW<>0 THEN IF LEN(Z$)<NW+
1 THEN PRINT SP$(1,NW-LEN(Z$)+1);
3730 PRINT Z$;
3740 IF Z2=0 THEN PRINT "00";SG$;:RETU
RN
3750 IF Z2<10 THEN PRINT "0";
3760 PRINT Z2;SG$;:RETURN
5000 ERR=PEEK(195):ERL=PEEK(186)+PEEK(
187)*256
5010 IF ERR=8 THEN TRAP 5000:GOTO ERL
5020 PRINT "ERROR-   ";ERR;" AT LINE "
;ERL:TRAP 33333:END
5030 PRINT " AT LINE ";ERL
5040 TRAP 33333:END
6000 TRAP 33333:GRAPHICS 0:END
```

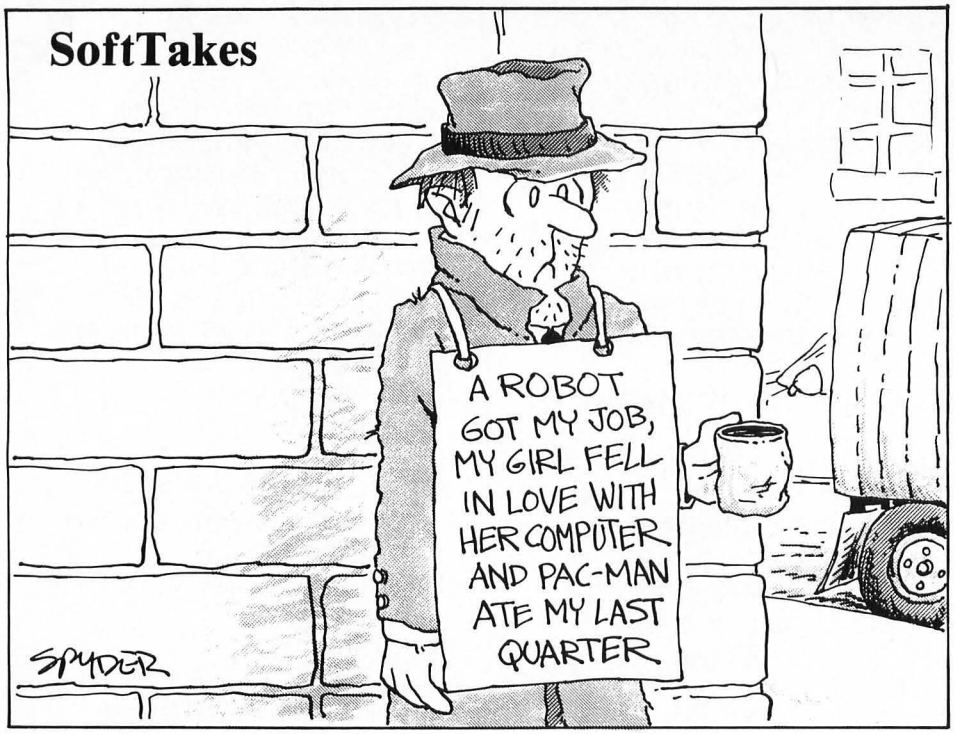
Financial Operating System, ***continued***

For **ATARI®**
FINANCIAL OPERATING SYSTEM

LINES	SWAT CODE	LENGTH
10 - 230	DC	540
231 - 470	GO	469
480 - 600	MZ	418
610 - 720	UD	276
730 - 1020	DH	466
1030 - 1120	RO	590
1125 - 1200	ZH	527
1210 - 1320	NY	461
1330 - 2010	XZ	448
2020 - 2510	TK	469
3000 - 3110	UK	445
3120 - 3230	DZ	520
3240 - 3340	HN	524
3350 - 3450	YG	521
3460 - 3570	TC	520
3580 - 3670	LT	513
3680 - 5010	KI	434
5020 - 6000	KY	107

```
   SS SS SS SS SS SS SS SS SS SS SS
                                  SS
        Applesoft BASIC           SS
  SS 'Financial Operating System'  SS
  SS     Author:  J. M. Keynes     SS
  SS  Translator:  Kerry Shetline  SS
  SS      Copyright © 1983         SS
  SS  SoftSide Publications, Inc   SS
  SS                               SS
  SS SS SS SS SS SS SS SS SS SS SS SS

10  GOTO 200
100 C$ = "": IF A$ = "" THEN  RETURN

110  FOR J2 = 1 TO  LEN (A$):J1 =
     ASC ( MID$ (A$,J2,1)):C$ =
     C$ +  CHR$ (J1 - 32 * (J1 >
     = 97 AND J1 <  = 121)): NEXT
     J2: RETURN
120  VTAB CL: CALL  - 958: RETURN

200 NW = 8:DS$ = "": FOR X = 1 TO
     39:DS$ = DS$ + "=": NEXT :DV
     (1) = .02:DV(2) = .01:DV(3) =
     .0125:DV(4) = .015
210 DT$(1) = "SL":DT$(2) = "DDB":
     DT$(3) = "125%":DT$(4) = "15
     0%"
220 D$(1) = "BUILDING":D$(2) = "P
     ERSONAL PROPERTY"
400  HOME : PRINT "NOTE: WHEN YOU
      ARE ASKED FOR A": PRINT "PE
     RCENTAGE, YOU MAY ENTER IT A
     S": PRINT "EITHER AN INTEGER
      OR A DECIMAL": PRINT "(24 O
     R .24, FOR EXAMPLE).": PRINT

410  GOSUB 3070: PRINT : GOSUB 31
     00: PRINT : GOSUB 3130: PRINT
     : GOSUB 3190: PRINT : GOSUB
     3220: PRINT : GOSUB 3260: PRINT
     : GOSUB 3470: PRINT
500  HOME : VTAB 4: PRINT  TAB( 1
     0)"I'M THINKING....."
510 VS = VA * GI / 100:EV = ER *
     (GI - VS) / 100:OI = GI - VS
      - EV
520 M1 = 0:M2 = 0:PT(1) = 0:PT(2)
     = 0:IT(1) = 0:IT(2) = 0
530  IF MN = 0 THEN 600
540  FOR X = 1 TO MN: IF T(X) = 1
     2000 THEN IT(X) = AM(X) * (R
     T(X) / 100): GOTO 580
550 I(X) = RT(X) / 1200:T = (I(X)
      + 1) ^ ( - T(X)):T = 1 - T:
     MP(X) = I(X) * AM(X) / T
560 MP(X) =  INT (MP(X) * 100 + .
     5) / 100:IT(X) = 0:PT(X) = 0
     :RP = AM(X)
570  FOR Z = 1 TO 12:IM =  INT (R
     P * I(X) * 100 + .5) / 100:P
     M = MP(X) - IM:IT(X) = IT(X)
      + IM:PT(X) = PT(X) + PM:RP =
     RP - PM: NEXT Z
580  NEXT X
590 M1 = PT(1) + IT(1):M2 = PT(2)
      + IT(2)
600 CF = OI - M1 - M2:PP = PT(1) +
     PT(2):RN = CF + PP:EQ = G -
     MT:RT = RN / EQ * 100

610 DP = D(1) * G / DY(1) * DV(DP
     (1)) + D(2) * G / DY(2) * DV
     (DP(2))
620 TI = RN - DP:PU = G / UN:PF =
     G / SF:CR = OI / G * 100:MR =
     G / (GI - VS)
1000  HOME :X = GI:NW = 8: GOSUB
     4100: PRINT "COMPUTED GROSS"
      TAB( 28);X$
1010 X = VA:NW = 3: GOSUB 4090: PRINT
     "VAC ALLOWANCE "X$;:X = VS:N
     W = 8: GOSUB 4100: PRINT "%"
      TAB( 28);X$
1020 X = ER:NW = 3: GOSUB 4090: PRINT
     "EXPENSE RATIO "X$;:X = EV:N
     W = 8: GOSUB 4100: PRINT "%"
      TAB( 28);X$
```

```
1030 X = OI: GOSUB 4100: PRINT "O
     PERATING INCOME" TAB( 28);X$
1040  PRINT DS$
1050  PRINT "MTG PPM" TAB( 20)"#1
     " TAB( 32)"#2"
1060 X = PT(1): GOSUB 4100: PRINT
     "PRINCIPAL" TAB( 16);X$;:X =
     PT(2): GOSUB 4100: PRINT " "
     X$
1070 X = IT(1): GOSUB 4100: PRINT
     "INTEREST" TAB( 16);X$;:X =
     IT(2): GOSUB 4100: PRINT " "
     ;X$
1080 X = M1: GOSUB 4100: PRINT "T
     OTAL" TAB( 16);X$;:X = M2: GOSUB
     4100: PRINT " ";X$
1090  PRINT DS$
1100 X = CF: GOSUB 4100: PRINT "C
     ASH FLOW" TAB( 28);X$
1110 X = PP: GOSUB 4100: PRINT "P $
     RINCIPLE PAYMENT" TAB( 28);X
1120 X = RT:NW = 3: GOSUB 4090: PRINT
     "EQU RTN = "X$;:X = RN:NW =
     8: GOSUB 4100: PRINT "%" TAB(
     28);X$
1130 X = DP: GOSUB 4100: PRINT "D
     EPRECIATION" TAB( 28);X$
1140 X = TI: GOSUB 4100: PRINT "T
     AXABLE INCOME" TAB( 28);X$
1150 X = G: GOSUB 4100: PRINT "PU
     RCHASE PRICE" TAB( 28);X$
1160 X = MT: GOSUB 4100: PRINT "M
     ORTGAGE TOTAL" TAB( 28);X$
1170 X = EQ: GOSUB 4100: PRINT "E
     QUITY" TAB( 28);X$
1180 X = CR: GOSUB 4100: PRINT "%
      RETURN ON INVESTED $" TAB(
     28);X$"%"
1190 X = MR: GOSUB 4100: PRINT "M
     ULTIPLIER (X EARNINGS)" TAB(
     28);X$
1200 X = PU: GOSUB 4100: PRINT "P
     RICE PER UNIT =" TAB( 28);X$
1210 X = PF: GOSUB 4100: PRINT "P
     RICE PER SQ FT =" TAB( 28);X $
1220  INVERSE : PRINT "1-SUMMARY"
     ;: GOSUB 3020
1230  IF F = 2 THEN 2000
1240  HOME : VTAB 3:X = 6: GOSUB
     4100: PRINT "PRICE"; TAB( 25
     );X$
1250 X = GI: GOSUB 4100: PRINT "G
     ROSS INCOME" TAB( 25);X$
1260  PRINT : PRINT "DEPRECIATION
     " TAB( 19)"% ALLOC." TAB( 29
     )"YEARS" TAB( 36)"TYPE"
1270  FOR J = 1 TO 2
1280  PRINT D$(J); TAB( 20);D(J);
      TAB( 29);DY(J); TAB( 36);DT
     $(DP(J))
1290  NEXT J
1300  IF MN = 0 THEN  PRINT : PRINT
     "THERE ARE NO MORTGAGES": GOTO
     1370
1310  PRINT : FOR Z1 = 1 TO MN
1320  PRINT "MORTGAGE";: IF MN =
     2 THEN  PRINT " #"Z1;
1330 X = AM(Z1): GOSUB 4000: PRINT
     " IS $"X$" AT "RT(Z1)"%"
1340  IF T(Z1) = 12000 THEN  PRINT
     "FIRST YEAR INTEREST ONLY.":
      GOTO 1360
1350  PRINT "OVER "T(Z1)" MONTHS.
     "
1360  NEXT Z1
1370  PRINT : PRINT "THE BUILDING
     HAS "UN" UNITS": PRINT "AND
     "SF" SQUARE FEET."
1380  INVERSE : PRINT : PRINT "1-
     FIRST YEAR";: GOSUB 3020
1390  IF F = 1 THEN 1000
2000  HOME
2010  PRINT : PRINT "DO YOU WISH
     TO CHANGE:": PRINT
2020  PRINT  TAB( 5)"1 - COMPUTED
      GROSS INCOME"
2030  PRINT  TAB( 5)"2 - VACANCY
     ALLOWANCE"
2040  PRINT  TAB( 5)"3 - EXPENSE
     RATIO"
2050  PRINT  TAB( 5)"4 - PURCHASE
      PRICE"
2060  PRINT  TAB( 5)"5 - MORTGAGE
      INFORMATION"
2070  PRINT  TAB( 5)"6 - DEPRECIA
     TION INFORMATION"
2080  INPUT "ENTER THE NUMBER OF
     YOUR CHOICE: ";X
2090 X =  INT (X): IF X < 1 OR X >
     6 THEN 2000
2100  PRINT : ON X GOSUB 3190,310
     0,3220,3070,3260,3470
2110  GOTO 500
3000  INPUT "? ";P: IF P < 1 THEN
     P = P * 100
3010  RETURN
3020  PRINT "  2-CHANGE VALUE  3-
     RERUN"
3030 CL =  PEEK (37) + 1: NORMAL
     : INPUT "ENTER THE NUMBER OF
      YOUR CHOICE: ";A$:F =  INT
     ( VAL (A$)): IF F = 0 THEN  HOME
     : END
3040  IF F < 1 OR F > 3 THEN  GOSUB
     120: GOTO 3030
3050  IF F = 3 THEN  RUN
3060  RETURN
3070 CL =  PEEK (37) + 1: PRINT "
     WHAT IS THE PURCHASE PRICE O
     F THE": INPUT "PROPERTY? ";G
3080  IF G < = 0 THEN  GOSUB 120
     : GOTO 3070
3090  RETURN
3100 CL =  PEEK (37) + 1: PRINT "
     WHAT IS THE VACANCY ALLOWANC
     E,": PRINT "IN PERCENT";
3110 P = VA: GOSUB 3000: IF P < =
     0 THEN  GOSUB 120: GOTO 3100
3120 VA = P: RETURN
3130 CL =  PEEK (37) + 1: PRINT "
     HOW MANY UNITS DOES THE": INPUT
     "PROPERTY HAVE? ";UN
3140  IF UN < = 0 THEN  GOSUB 12
     0: GOTO 3130
3150  PRINT
3160 CL =  PEEK (37) + 1: PRINT "
     HOW MANY SQUARE FEET OF": INPUT
     "USABLE SPACE? ";SF
3170  IF SF < = 0 THEN  GOSUB 12
     0: GOTO 3160
3180  RETURN
3190 CL =  PEEK (37) + 1: PRINT "
     YEARLY GROSS INCOME (ASSUMIN
     G": INPUT "100% OCCUPANCY)?
     ";GI
3200  IF GI < = 0 THEN  GOSUB 12
     0: GOTO 3190
3210  RETURN
3220  PRINT "WHAT IS THE EXPECTED
      EXPENSE RATIO?": PRINT "THA
     T IS, THE YEARLY ESPENSES DI
     VIDED"
3230 CL =  PEEK (37) + 1: PRINT "
     BY GROSS INCOME, IN PERCENT"
     ;
3240 P = ER: GOSUB 3000: IF P < =
     0 THEN  GOSUB 120: GOTO 3230
3250 ER = P: RETURN
3260  HOME : PRINT "HOW MANY MORT
     GAGES WILL THERE BE,": INPUT
     "(0, 1, OR 2)? ";MN
3270 MN =  INT (MN): IF MN < 0 OR
     MN > 2 THEN 3260
3280  IF MN <  > 0 THEN  PRINT "F
     OR INTEREST ONLY OR BALLOON"
     : PRINT "MORTGAGES, YEARS =
     1000"
3290 MT = 0
3300  FOR J = 1 TO 2
3310 AM(J) = 0:T(J) = 0:RT(J) = 0
     :PT(J) = 0
3320  IF J > MN THEN 3450
3330  PRINT : PRINT "PLEASE GIVE
     THE FOLLOWING";: IF MN <  >
     1 THEN  PRINT : PRINT "FOR M
     ORTGAGE NUMBER "; MID$ ("ONE
     TWO",J * 3 - 2,3);
```



Financial Operating System, ***continued***

```
3340  PRINT ":"
3350 CL =  PEEK (37) + 1: INPUT "
     TOTAL AMOUNT OF THE MORTGAGE
     ? ";AM(J)
3360  IF AM(J) <  = 0 THEN  GOSUB
     120: GOTO 3350
3370  IF AM(1) >  = G THEN  PRINT
     : PRINT "THE VALUE OF THE MO
     RTGAGE": GOTO 3460
3380  IF J = 2 AND AM(1) + AM(2) >
      = G THEN  PRINT : PRINT "TH
     E SUM OF THE MORTGAGES": GOTO
     3460
3390 CL =  PEEK (37) + 1: INPUT "
     THE TERM, IN YEARS? ";T(J)
3400  IF T(J) <  = 0 THEN  GOSUB
     120: GOTO 3390
3410 T(J) = T(J) * 12
3420 CL =  PEEK (37) + 1: PRINT "
     THE INTEREST RATE, IN PERCEN
     T";
3430 P = RT(J): GOSUB 3000: IF P <
      = 0 THEN  GOSUB 120: GOTO 3
     420
3440 RT(J) = P:MT = MT + AM(J)
3450  NEXT J: RETURN
3460  PRINT "MUST BE LESS THAN TH
     E PURCHASE":X = G: GOSUB 400
     0: PRINT "PRICE OF $";X$: PRINT
     : PRINT : GOTO 3260
3470  HOME : PRINT "DESCRIBE DEPR
     ECIATION METHODS"
3480  PRINT : FOR J = 1 TO 2
3490 CL =  PEEK (37) + 1: PRINT "
     ALLOCATION TO ";D$(J): PRINT
     "IN PERCENT";
3500 P = D(J): GOSUB 3000: IF P <
     0 THEN  GOSUB 120: GOTO 3490
3510 D(J) = P
3520  PRINT : NEXT J
3530  IF D(1) + D(2) > 100 THEN  PRINT
     "THE SUM OF THESE ALLOCATION
     S CAN'T": PRINT "EXCEED 100%
     .  TRY AGAIN.": GOTO 3480
3540  FOR J = 1 TO 2
3550  PRINT : PRINT "WHAT METHOD
     OF DEPRECIATION IS": PRINT "
     TO BE USED FOR THE ";D$(J);"
     ?"
3560  PRINT "ENTER SL, DDB, 125%,
      OR 150%";
3570  INPUT "? ";A$: IF A$ = "" AND
     DP(J) <  > 0 THEN 3640
3580  GOSUB 100:C$ =  LEFT$ (C$,3
     )
3590 DP(J) = 0: IF C$ = "SL" THEN
     DP(J) = 1
3600  IF C$ = "DDB" THEN DP(J) =
     2
3610  IF C$ = "125" THEN DP(J) =
     3
3620  IF C$ = "150" THEN DP(J) =
     4
3630  IF DP(J) = 0 THEN  PRINT "I
     LLEGAL METHOD.  PLEASE": PRINT
     "ENTER SL, DDB, 125%, OR 150
     % ";: GOTO 3570
3640 CL =  PEEK (37) + 1: PRINT :
      PRINT "TERM OF ";D$(J): PRINT
     "DEPRECIATION, IN YEARS";
3650  INPUT "? ";DY(J): IF DY(J) <
      = 0 THEN  GOSUB 120: GOTO 3
     640
3660  NEXT J: RETURN
4000 X =  INT (X * 100 + .5) / 10
     0:X1 =  INT ((X -  INT (X)) *
     100 + .5)
4010 X$ =  STR$ ( INT ( ABS (X)))
      + "."
4020 SP$ = "": IF  LEN (X$) < NW +
     1 THEN SP$ =  LEFT$ ("
      ",NW -  LEN (X$) + 1)
4030  IF X >  = 0 THEN 4060
4040  IF SP$ = "" OR SP$ = " " THEN
     SP$ = "-": GOTO 4060
4050 SP$ =  LEFT$ (SP$, LEN (SP$)
      - 1) + "-"
4060  IF X1 = 0 THEN X$ = X$ + "0
     0": RETURN
4070  IF X1 < 10 THEN X$ = X$ + "
     0"
4080 X$ = X$ +  STR$ (X1): RETURN

4090  GOSUB 4000:X$ = SP$ + X$: RETURN

4100 SX =  SGN (X):X =  ABS (X): GOSUB
     4000:X$ = SP$ + X$: IF SX <
     0 THEN X$ = X$ + "-"

4110  RETURN
```

For **APPLE®**
FINANCIAL OPERATING SYSTEM

LINES	SWAT CODE	LENGTH
10 - 500	KT	536
510 - 610	WP	525
620 - 1090	VO	501
1100 - 1210	MK	517
1220 - 1330	JQ	373
1340 - 2050	GI	360
2060 - 3050	ST	366
3060 - 3170	ME	385
3180 - 3290	MR	451
3300 - 3410	VE	436
3420 - 3530	LI	442
3540 - 3650	WQ	449
3660 - 4100	IL	339
4110 - 4110	HZ	6

KIDS AND COMPUTERS:

The Parents' Microcomputer Handbook

Reviewed by Judy Neyhart

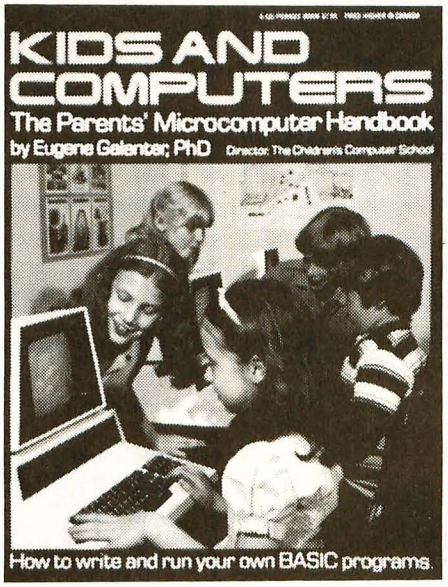
I've observed that there are two types of computer people — the *users* and the *programmers.* The *users* are those who, through curiosity, bravery or innovativeness, have bought a computer and found that this strange new machine can be of great value in their daily lives. They do not write programs, but understand enough to use their own machines. The *programmers* are usually in the data processing field and understand the inner and outer workings of these machines. Unfortunately, they are so far above the rest of us that it is hard to understand them when they talk. They use many acronyms and electronics terms and have a hard time getting down to the level of the beginner.

Author Background

It is rare when these two computer types merge in one personality. Dr. Eugene Galanter, the author of *Kids and Computers — the Parents' Microcomputer Handbook,* is such a person. Through his Children's Computer School in New York City he has taught thousands of children (and their parents) to understand and program the personal computer. In this book he details the trials and tribulations of purchasing his first microcomputer as a professor at Columbia in 1974 (very early in the personal computer history), teaching his three daughters to program — yes, he has very definite opinions on the statement that boys are better programmers than girls — and finally planning and setting up The Children's Programming School in New York City in 1981. As I read of these personal triumphs, I developed an understanding of his background and felt a great sense of "I've been there, too!" Many of us are going through these same steps as we teach our own children about computers.

An Educational Resource

This book is a tremendously valuable resource for the parent and/or teacher. It has value for the beginner as well as for the confirmed personal computer addict. Through his easy-to-read explanations of all aspects of the personal computer, I gained a new understanding of the inner workings of this powerful machine. His chapter, "The Microcomputer's Parts," answered many questions that I didn't even realize I needed to ask: the difference between RAM and ROM, how the many personal computers differ in their memory arrangement, what a pixel is, where the word bit came from, and what the word modem stands for, for example. Dr. Galanter describes these and more in such an easy and down-to-earth way that it almost seems fun. Perhaps some of you are like I am. When I hear computer jargon, I tense up and close my ears since I know ahead of time that it will be unintelligible. Dr. Galanter, however, can make you understand.



Programming and Computer Literacy

He has a firm and well-documented opinion on the value of children knowing how to program. As he says in the book, "There is really nothing like the experience of finding out how to do something that was rumored to be arcane and ends up comprehensible." There are only three chapters — out of the eleven — on programming, but they are enough to whet the beginner's appetite. He sorts out the different versions of BASIC so that no matter what machine you use, the book describes it.

Of particular interest to me were his opinions on the value of computer literacy. As he says, "The computer is making its way into every home and work place. As a consequence, computer literacy becomes a necessary skill." He has some informed and interesting opinions on where we are going with computers and what we should do to keep in step.

Every teacher should read his chapter on "Evaluating Computer Education" — as should every parent in assessing the computer education in his child's school. Because Dr. Galanter has planned and set up his own computer school, he has valuable suggestions on teaching: how large and long should the classes be, the pupil-to-teacher ratio, and at what age to begin computer teaching.

Consumer Advice

In his chapter, "How to Buy a Microcomputer," he tries to sort out all the choices for the layman. His opinions are sometimes different from mine, but then we all have a loyalty to our particular system — ours has actually become a member of the family. He does, however, say that his "educational advice and assertions should be taken for what they are — *informed opinion and reasoned hypothesis.*"

If you've browsed in bookstores from time to time over the past year, you've noticed that the Computer Science section has grown by leaps and bounds in an attempt to meet the demand of the consumer who is eager for advice and help with understanding the Computer Revolution. Many of these books resemble college level textbooks — about as entertaining as stale toast and for that reason, a waste of money. Dr. Galanter's is the first book I have seen that I would recommend without reservation for parents and teachers alike at all levels of ability. Your $7.95 could not be better spent. As a reference on your computer room shelf, *Kids and Computers* is a must.

From Perigee Books, 200 Madison Avenue, New York, NY 10016. Suggested retail price: $7.95.

Sides Of SoftSide

ATARI® /SIDE

Article
Exploring the Atari Frontier
by Alan J. Zett
Continuing the discussion of Player/Missile Graphics, Alan Zett shows how to make your displays come alive through animation. Vertical PM/G movement is difficult, but a simple BASIC approach can do the job efficiently.

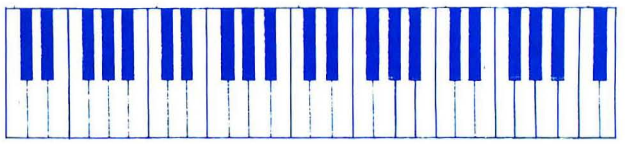
Reviews
Speed Reading
Reviewed by Steven Oliver II
The information explosion requires everyone to read faster than ever, and this program is your ticket to high speed reading skills — maybe several hundred words faster than you have ever been able to read before.

ABC Compiler
Reviewed by Carl Firman
A BASIC compiler can ease your pokey program blues if Assembly Language still eludes you. Compiled programs run ten to twenty times faster when you answer the prompts and let ABC take over.

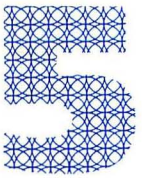
APPLE® /SIDE

Apple Diskourse
by Cary W. Bradley
This installment moves away from a semi-technical, utility orientation to tutor Apple users in how to use DOS and incorporate disk functions within your programs.

Music Theory Software
Reviewed by Steve Birchall
Microcomputers improve musical pedagogy by permitting students to master the ABCs of music theory at their own pace, in relative privacy. This method of teaching how to hear chords, scales and keys will have a substantial impact on college music curricula.



PC/SIDE



Reviews
Facts In Five
Reviewed by Robert C. Gray
Test your trivia knowledge to the limit with this exciting word identification game. You can customize your categories — everything from tractor parts to *Fortune 500* CEOs — and drill yourself and your opponents until one of you hollers, "Uncle!"

TRS-80®/SIDE

Reviews
Weerd
Reviewed by Mark Renne
Not all extraterrestrial beings are as benevolent as E.T. Tackle some of the 25 variations of "weerdness" packed into this software and compete for the highest score in your quadrant of the universe.

Jabbertalky
Reviewed by Mark Renne
Here is a learning tool that teaches while it entertains like an arcade game. Your child's gleeful responses also will signal the occurrence of constructive educational experiences and you can relax as he spends his precious time at the computer productively.

EXPLORING THE ATARI® FRONTIER

The Player/Missile Connection: Part III

by Alan J. Zett

No one likes a dead video game. Inanimate invaders and stationary space ships don't capture your imagination. I can't think of a single game with a display that just sits there and stares back at you. Movement and action are synonymous with fun and excitement. Many games have become famous just because of the motions of meandering monsters. PACMAN™ and DONKEY KONG™ have patterns everybody wanted to learn. The alien ships in GALAGA™ have a variety of interesting flight patterns. CENTIPEDE™ has some truly unique movement routines. Games like VANGUARD™ and DEFENDER™ rely quite heavily on making things move on the screen. Trying to learn the maneuvers contributes to the game's intrigue and to the player's fascination.

In the last installment of Frontier, I discussed Player/Missile Graphics (PMGs) motion, covering horizontal movement and lightly touching on the difficulties of vertical movement. While working on the problems associated with that subject, I discovered a way to control all vertical movement fast, efficiently and easily from BASIC.

Player/Missile Motion

Vertical motion on PMGs is not a difficult concept to understand or implement. To move a Player around the screen from BASIC is quite easy. To move it left or right, only a single POKE is required. To move it up or down, use a simple FOR/NEXT loop. The real problem is that it takes a relatively long time to move a Player vertically. If you want to move a PMG shape up or down, move all the bytes (within PMG RAM) that define the shape. To control how far it moves, count the number of scan lines, and shift the PMG's memory location by the appropriate number of bytes.

Each byte in a PMG shape draws one or two scan lines of graphics depending on the resolution mode. When the shape in Player 0 needs to move from Y coordinate 4 to Y coordinate 18, the shape-

continued on page 65

HAVE YOU FLOWN YOUR ATARI TODAY?

Imagine yourself at the controls of a small, single-engine plane, 10,000 feet in the air, on your approach to the runway and safety. You're running low on fuel, but your instruments show that you're on the glide path, and lined up with the runway. It's a beautiful, sunny day, and you can see the airport in the distance, across the grassy fields. But the crosswind is tricky, and it will take all your skill to land safely. You're coming down now, and the runway is getting closer. A bit left, OK, now lower the power, fine, now put down the flaps. Pull the nose up a bit more, you're a little low. Watch the power! Don't stall. OK. Here comes the runway. You hear the squeal of tires on pavement, your pulse quickens, you're down, but watch it, you're pulling right! Brakes, brakes! Left more! You've stopped safely! Good job. The first real-time flight simulator for ATARI is now available from MMG Micro Software. Written entirely in machine language, there are four levels of difficulty, landings in clear or foggy weather, landings with or without instruments, and with or without the real-time view from the cockpit. **Final Flight!** requires Atari 400/800, 24K, 1 joy stick, and is offered on tape or disk for the same suggested retail price of $29.95.

MMG

Final Flight! is available at your local dealer or direct from **MMG Micro Software.** Just send check or money order to P.O. Box 131, Marlboro, N.J. 07746 or for Mastercard, Visa, and C.O.D. deliveries call **(201)431-3472.** Please add $3.00 for postage and handling. New Jersey residents add 6% sales tax.

Atari Frontier, *continued*

defining bytes must be moved fourteen bytes in PMG RAM. In addition, if you desire smooth motion, you must display all the Y coordinate positions between both locations as well. Then the shape will appear to travel, rather than jump, from one point to another on the screen.

Motion Methods

Traditionally, PMG vertical movement is accomplished in two ways: from BASIC in a FOR/NEXT loop, or from Machine Language with a block move or page rotate routine. Until now, programmers have always opted for the Machine Language method since it was incredibly fast compared to BASIC. But BASIC can do the same thing at almost the same incredible speed. The answer is to fool Atari BASIC into using the text in a string variable for PMG RAM. If you put the data for a PMG into a string, you can make modificatons to the PMG shape using the normal string operations, but which resemble Machine Language capabilities. In reality, a PMG shape is nothing more than a series of consecutive bytes in memory with values ranging from 0 to 255. That description also suits a BASIC string variable. The only difference between the two is that bytes in a string variable are represented on the screen as an ATASCII character. When used as PMG data, the same bytes represent a series of eight bits which correspond to eight graphics pixels on a line of a PMG shape. Even though the bytes are identical, they yield different results, depending on what section of the hardware/software interprets them.

PMG Strings

The problem with this idea is that PMGs must be at the beginning of a memory page boundary, whereas strings are free floating in memory. If this were not so, you could point PMBASE (see Table 1) to a string variable text address (i.e. POKE PMBASE,ADR(PM$)/256). One way to get around this problem is to DIMension so much memory for a string that some portion of it will have to fall on a page boundary. I don't like this solution, because it wastes precious program memory space. The elegant way around this problem is to change the location where a string variable points for its text data. This method also allows you to point a string anywhere in ROM or RAM, such as screen memory, an I/O buffer, or the ROM character set. But what are variables really, and where are they in memory?

Table 1

IMPORTANT PMG SHADOW REGISTER LOCATIONS

Hex	Dec	Title	Register Description
0058	00088	SAVMSC	LSB of screen memory pointer.
0059	00089		MSB of screen memory pointer.
006A	00106	RAMTOP	Top of RAM pointer.
0082	00130	VNTP	LSB of variable name table pointer.
0083	00131		MSB of variable name pointer.
0084	00132	VNTD	LSB of variable name table delimiter.
0085	00133		MSB of variable name table delimiter.
0086	00134	VVTP	LSB of variable value table pointer.
0087	00135		MSB of variable value table pointer.
008C	00140	STARP	LSB of string/array data pointer.
008D	00141		MSB of string/array data pointer.
022F	00559	SDMCTL	Direct Memory Access control.
02C0	00704	PCOLRO	Player zero color register.

IMPORTANT PMG HARDWARE REGISTER LOCATIONS

Hex	Dec	Title	Register Description
D000	53248	HPOSPO	Register Description.
D004	53252	POPF	Player zero to Playfield collision.
D01C	53276	VDELAY	Player/Missile Vertical Delay.
D01D	53277	GRACTL	Player/Missile Graphics control.
D01E	53278	HITCLR	PMG collision register reset strobe.
D407	54279	PMBASE	Memory page of PMG data.

Altering BASIC Variables

Whenever you enter a variable into a line of Atari BASIC code, it is "tokenized." A line of BASIC code *in memory* looks nothing like the *display* you get when you type LIST. Tokenization is the process by which long groups of symbols are replaced with a single or smaller group of symbols that mean the same thing. For example: When you type the command PRINT from BASIC, it goes into the input buffer. The Atari looks up PRINT in a table of tokens understood by Atari BASIC. Then it replaces the ATASCII values for the characters in the word PRINT (80, 82, 73, 78, and 84) with a single byte having, in this case, a value of 32. All Atari BASIC commands, statements and operators have a replacement token value.

When BASIC encounters a variable during the tokenizing process, it places the variable name in a table. From then on, a single byte represents that variable in the stored BASIC code. The value of the byte is the variable's number in the table minus one, plus 128. For example: the first variable in a program always becomes the token value of 128. The second is 129, and so on, up to 255. This is why all variable names take only one byte of memory in a BASIC program regardless of the name's length. Another

↪

Listing 1.

```
30000 VNTP=PEEK(130)+PEEK(131)*256
30010 VNTD=PEEK(132)+PEEK(133)*256
30020 VVTP=PEEK(134)+PEEK(135)*256
30030 STARP=PEEK(140)+PEEK(141)*256
30040 AZ1=-1
30050 AZ1=AZ1+1:FOR AZ=1 TO LEN(VAR$)
30060 AZ2=PEEK(VNTP):IF AZ2>127 AND AZ
<LEN(VAR$) THEN 30080
30070 IF AZ2-128*(AZ2>127)=ASC(VAR$(AZ
)) THEN VNTP=VNTP+1:NEXT AZ:GOTO 30110
30080 IF PEEK(VNTP)<128 THEN VNTP=VNTP
+1:GOTO 30080
30090 VNTP=VNTP+1:IF VNTP<VNTD THEN 30
050
30100 GRAPHICS 0:? VAR$;" ISN'T A LEGA
L VARIABLE":END
30110 AZ=VVTP+AZ1*8+2:IF PEEK(AZ-2)<>1
29 THEN 30090
30120 AZ1=LOC-STARP:GOSUB 30140:AZ1=LE
N:GOSUB 30140:AZ1=LEN:GOSUB 30140
30130 RETURN
30140 AZ2=INT(AZ1/256):AZ1=AZ1-AZ2*256
:POKE AZ,AZ1:POKE AZ+1,AZ2:AZ=AZ+2:RET
URN
```

LINES	SWAT CODE	LENGTH
30000 - 30110	MM	477
30120 - 30140	JG	132

Atari Frontier, ***continued***

table holds the information on what the variable contains. For a string variable, the table has pointers for the current length of the string, its maximum length (the DIM value), and the address where the string's text is located in memory. This value is an offset from the start of the string and array data block in memory. By changing the current length, DIM length, and the offset, you can point a string to any amount of existing memory — all without taking *any* string space. Finding the location of the variable in the name and value tables can be complicated, so I wrote the program STRPUT, which stands for STRing PUTter.

Tell Your Strings Where To Go

The subroutine in Listing 1 can be merged into any program when you need to redirect strings. It adds only six extra sectors to a disk file, or about 700 additional bytes of BASIC program code. To use it, first set up a few variables and then execute a GOSUB to line 30000. Assign variables as follows:

- LEN: The amount of memory occupied by the string.
- LOC: The memory address of the new string text data.
- VAR$: The name of the variable to be reassigned.

The variables LEN and LOC are scalar variables — simply assign the values listed. However, VAR$ adds a string variable entry into the variable table. Since all strings must be preDIMensioned before use, **don't forget to dimension VAR$ at the beginning of your program.** Also, the variable to be reassigned, whose name is contained in VAR$, must be a string variable. For best results, put the string to be reassigned at the beginning of your program and use the LIST/ENTER technique from SWAT to reorder the table of variable names. This places the variable name at the beginning of the name table and speeds up the search time for the subroutine.

STRPUT Step by Step

Lines 30000 through 30030 set up some important memory locations used by Atari BASIC in page zero. Table 1 shows memory locations that bear the same name as the variables in these lines. VNTP is the Variable Name Table Pointer. It points to the memory address for the start of the variable names list. VNTD is the Variable Name Table Delimiter pointer. The Atari can have up to 128 variables in a program. If you use less than 128, this location points to the zero byte following the last variable name. If you use exactly 128 variables, it points to the last character of the last variable name in the table. VVTP is the Variable Value Table Pointer. This points to the variable value entries which are all eight bytes long, regardless of variable type. The last one, STARP, is the STring/ARray Pointer. It points to the first byte of the string and array data block in memory.

Lines 30040 and 30050 set up the variable name search routine in a FOR/NEXT loop. Line 30060 gets a character from the name table and checks for inverse. To signify the end of an entry, the table stores the variables with the last character of the name in inverse. Scalar variables have the last character in inverse, array variables end with an inverse left parenthesis, and string names are followed by an inverse dollar sign.

Line 30070 continues the name matching process and if a match is made, jumps to line 30110. Line 30080 bumps the pointer to the next variable name entry (next character after an inverse character). If the pointer is now past the end of the table (VNTP is equal to or greater than VNTD) then line 30090 prints an error message stating that the variable is invalid.

Line 30110 checks to see if the program currently is using the string to be reassigned, and prints an error message if not. Every variable entered since the last time the computer was powered up or a NEW command was executed, is considered active. The Atari "hangs on" to old variable names even though the program currently in memory does not use them. The SAVE command stores the entire variable name table, including unused variables, along with the program data (which is in the tokenized format). LOADing the file back in restores all the variable names to memory. But when you LIST a program to cassette or disk, it saves the expanded ATASCII characters, exactly as you see on the screen. What the ENTER command does is to cause each line of BASIC code to be "typed in," or reentred, as if by hand. This process creates a new variable table consisting only of variables used in the program, and in the order they occur within the program text.

Lines 30120 through 30140 POKE the new length and memory location of the reassigned string. Note that for ease of use, the subroutine sets the maximum DIM length equal to the current string length in LEN. It adjusts the memory location in LOC by subtracting the starting address of string and array data from the actual memory location. This allows it to store the actual location in the variable value table as an offset, as is expected. This also means that pointing a string to a memory area lower than the start of STARP is a bit tricky. You must add 65536 to the actual memory address so the offset will wrap around to the start of memory. For example, if I wanted a string to point to the last half of page zero, I would set LEN equal to 128, and LOC equal to 65664 (128 plus 65536) instead of 128.

Another Take-A-Part

Listing 2 is a short demo that moves the letters 'SS' around the screen with a joystick. Type it in and try it, then look at the listing. Starting with the first line, let's take it apart:

Line 100 DIMensions all of the program strings. PM$ is for PMG RAM manipulations. Eventually you will reassign it to point at the memory occupied by Player 0. C$ holds the PMG shape data in the form of ATASCII characters. By assigning C$ to a portion of PM$, you can place a shape instantly anywhere in the Player 0 RAM. Line 100 also reserves eight pages of memory (2K RAM) for PMGs with single-line-resolution. The program also bumps back the register location known as RAMTOP (see Table 1 for all register descriptions), thus reducing the amount of free RAM by lowering the value contained in the pointer. The extra RAM, which BASIC thinks is not available, is now free for PMGs. Line 100 also sets the variable LOC equal to the start of PMG RAM for Player 0, and sets LEN to 256, the maximum number of bytes in a single-line-resolution Player shape.

Line 110 POKEs the PMG shape's color into the PCOLR0 register, assigns PM$ to point to the Player 0 PMG RAM, and uses an interesting method for quickly filling a string with any character from BASIC. Because of a quirk in the way Atari BASIC handles strings, a string assignment can copy the first character into every position in the string up to the current length. When you specify only the first position within a string, such as A$(44), Atari BASIC assumes the second position is the

default, which is the end of the string. If you assign the second position in a string equal to the first position, in the form A$(2) = A$(1), the filling effect occurs. The second character will become the same as the first, and then the third will become the same as the second, etc. The computer thinks it's moving an entire string down one character, but in actuality, it's filling consecutive characters with the previous character. In this case, PM$ and C$ will be set to character zeros.

Line 120 reads the PMG shape data into C$ and does the three mandatory POKEs into SDMCTL, GRACTL, and PMBASE that are required to implement PMGs. The values used are for single line resolution PMGs. SDMCTL turns on the PMG DMA control and sets the resolution mode, GRACTL enables the GTIA's PMG processing hardware, and PMBASE tells GTIA where to look for PMG RAM. Line 120 also sets the vertical and horizontal positions for the shape in Player 0 to 40 (PMV = 40) and 60 (PMH = 60).

Lines 130 through 210 convert joystick input into adjustments for the horizontal and vertical posititions of Player 0. Lines 220 through 250 check these new positions to see if they are still within the valid coordinate range. Line 260 POKEs the adjusted horizontal position into the HPOS0 register and assigns the PMG shape string into PMG RAM via PM$. Line 270 branches back for more input and line 280 contains the Player/Missile shape data. Lines 290 through 430 are the STRPUT subroutine which I discussed previously.

Unexpected Uses

While experimenting with PMGs and the STRPUT subroutine, I somewhat accidentally started to write a demo, which eventually evolved into a car race program that used a PMG car shape and a downward scrolling racetrack. I had gone through the trouble of writing a screen scroll routine in 6502 Machine Language to make the game playable, but I wasn't satisfied with it. The whole purpose of writing the demo was to show what could be done with PMGs from Atari BASIC through the use of STRPUT. Adding a Machine Language routine to the demo seemed to defeat the entire purpose. Then a thought came to me: "Wait a minute... a downward scrolling screen is similar to a downward scrolling PMG." That's when I got the idea of using STRPUT to assign a string to the screen memory RAM. By setting a string equal to the entire area of screen memory, a simple string assignment statement could scroll the entire screen downward.

Listing 2.

```
100 DIM VAR$(3),PM$(256),C$(11):POKE 1
06,PEEK(106)-8:GRAPHICS 0:LOC=PEEK(106
)*256+1024:LEN=256:POKE 752,1
110 PRINT :VAR$="PM$":POKE 704,30:GOSU
B 290:PM$=CHR$(0):PM$(256)=CHR$(0):PM$
(2)=PM$(1):C$=PM$
120 FOR X=3 TO 9:READ AZ:C$(X,X)=CHR$(
AZ):NEXT X:POKE 559,62:POKE 53277,3:PO
KE 54279,PEEK(106):PMV=40:PMH=60
130 ST=STICK(0):IF ST=15 THEN 130
140 IF ST=5 THEN PMH=PMH+1:PMV=PMV+2
150 IF ST=6 THEN PMH=PMH+1:PMV=PMV-2
160 IF ST=7 THEN PMH=PMH+1
170 IF ST=9 THEN PMH=PMH-1:PMV=PMV+2
180 IF ST=10 THEN PMH=PMH-1:PMV=PMV-2
190 IF ST=11 THEN PMH=PMH-1
200 IF ST=13 THEN PMV=PMV+2
210 IF ST=14 THEN PMV=PMV-2
220 IF PMV<1 THEN PMV=256
230 IF PMV>256 THEN PMV=1
240 IF PMH<30 THEN PMH=220
250 IF PMH>220 THEN PMH=30
260 POKE 53248,PMH:PM$(PMV)=C$
270 GOTO 130
280 DATA 68,170,136,68,34,170,68
290 VNTP=PEEK(130)+PEEK(131)*256
300 VNTD=PEEK(132)+PEEK(133)*256
310 VVTP=PEEK(134)+PEEK(135)*256
320 STARP=PEEK(140)+PEEK(141)*256
330 AZ1=-1
340 AZ1=AZ1+1:FOR AZ=1 TO LEN(VAR$)
350 AZ2=PEEK(VNTP):IF AZ2>127 AND AZ<L
EN(VAR$) THEN 370
360 IF AZ2-128*(AZ2>127)=ASC(VAR$(AZ))
 THEN VNTP=VNTP+1:NEXT AZ:GOTO 400
370 IF PEEK(VNTP)<128 THEN VNTP=VNTP+1
:GOTO 370
380 VNTP=VNTP+1:IF VNTP<VNTD THEN 340
390 GRAPHICS 0:? VAR$;" ISN'T A LEGAL
VARIABLE":END
400 AZ=VVTP+AZ1*8+2:IF PEEK(AZ-2)<>129
 THEN 380
410 AZ1=LOC-STARP:GOSUB 430:AZ1=LEN:GO
SUB 430:AZ1=LEN:GOSUB 430
420 RETURN
430 AZ2=INT(AZ1/256):AZ1=AZ1-AZ2*256:P
OKE AZ,AZ1:POKE AZ+1,AZ2:AZ=AZ+2:RETUR
N
```

SWAT TABLE

LINES	SWAT CODE	LENGTH
100 - 150	MU	501
160 - 270	ZP	347
280 - 390	QB	448
400 - 430	SM	190

A Free Bonus Program

The final result is Listing 3, which I call RACER. Combining most of the code from Listings 1 and 2, plus a little extra for a racetrack algorithm, I created a complete game written entirely from BASIC. The startling thing about RACER is that it is *fast* — surprisingly fast when you consider what is being done in BASIC. Since I have already discussed most of the code for RACER, the only thing (other than the screen scroll routine) that might not be clear is in lines 150 and 410. In Table 1 you'll find two POKE locations with the names HITCLR and P0PF. They deal with the PMG collison registers which I will cover in a later installment. These registers can tell BASIC when the car touches a racetrack wall.

To play the RACER program just type it in and RUN it. You earn points for staying alive on the racetrack. You get higher points for driving faster, but this cuts down on your response time, making accidents more probable. The racetrack also gets more and more twisting, so the right combination of speed and caution will pay off. To move the car, push your joystick left or right. To accelerate push the joystick forward, to decelerate, pull back. See how long you can survive!

Wait a Micro Second...

Up until now, I have been concerned only with vertical movement for single-line-resolution PMGs. The only real difference between them and double-line-resolution PMGs is that the latter take half as many bytes. That will change some of the calculations for STRPUT and such, but not too seriously. For example in Listing 2, line 100: LOC would change to PEEK(106)*256 + 512, and the LEN of PM$ would change to 128. The rest would stay the same. In fact, the only really objectionable problem with double line resolution PMGs is that when you move them, they also *move*

↪

Atari Frontier, ***continued***

two lines at a time. For really smooth motion, a PMG shape should move one scan line at a time vertically, otherwise a certain amount of jumping will be visible.

To get around this problem, the GTIA processor has a built in hardware register known as VDELAY, as listed in Table 1. The purpose of VDELAY is to cause the video processing hardware to wait for one scan line before "drawing" the PMG shape into the video signal. You can activate VDELAY by POKEing values into it. Look at the bit map for VDELAY in Figure 1. It shows which bits in VDELAY affect which Players and Missiles. Instead of moving the shape in memory one byte at a time, you can alternate between moving one byte and POKEing the shape down one scan line with VDELAY. Note that you can set more than one bit at any time. To calculate what value to POKE into VDELAY, add the decimal values of each of the bits together.

That's it for this month. Next time we get together for a Frontier Fireside we'll continue our chat on PMGs. Until then — how about some letters people?

Figure 1.

VDELAY —
VERTICAL DELAY ($D01C)
A hardware register which controls the vertical position of PMGs.

Bit Map:

Bit:	7	6	5	4	3	2	1	0

B7 — 1 Delays Player 3. Value 128.
B6 — 1 Delays Player 2. Value 64.
B5 — 1 Delays Player 1. Value 32.
B4 — 1 Delays Player 0. Value 16.
B3 — 1 Delays Missile 3. Value 8.
B2 — 1 Delays Missile 2. Value 4.
B1 — 1 Delays Missile 1. Value 2.
B0 — 1 Delays Missile 0. Value 1.

Listing 3.

```
100 DIM VAR$(3),PM$(256),C$(20),S$(400
),T$(400):POKE 106,PEEK(106)-8:GRAPHIC
S 4:LOC=PEEK(106)*256+1024
110 VAR$="PM$":LEN=256:GOSUB 480:LOC=P
EEK(88)+PEEK(89)*256:VAR$="S$":LEN=400
:GOSUB 480:POKE 752,1
120 POKE 708,68:POKE 709,14:POKE 710,0
:RESTORE 470:FOR AZ=1 TO 18:READ AZ1:C
$(AZ)=CHR$(AZ1):NEXT AZ:POKE 559,62
130 PM$=CHR$(0):PM$(256)=CHR$(0):PM$(2
)=PM$(1):POKE 53277,3:POKE 54279,PEEK(
106):POKE 53248,122:SCORE=0
140 DIS=3:SIDE=1:FAC=0.9:LAP=0:POS=32:
? "RACER BY ALAN J. ZETT":? :? :PM$(17
6)=C$:PMH=122:PMV=176:POKE 704,200
150 COLOR 1:PLOT 32,0:DRAWTO 32,39:PLO
T 45,0:DRAWTO 45,39:POKE 53278,0
160 LAP=LAP+1:? CHR$(28);"SCORE: ";INT
(SCORE):T$=S$:S$(11)=T$:IF RND(0)>FAC
AND D=0 THEN D=INT(RND(0)*DIS)-SIDE
170 COLOR 0:PLOT POS,0:PLOT POS+13,0
180 IF D>0 THEN POS=POS+1
190 IF D<0 THEN POS=POS-1
200 IF POS<10 THEN POS=10:D=0
210 IF POS>64 THEN POS=64:D=0
220 COLOR 1:PLOT POS,0:PLOT POS+13,0
230 SCORE=SCORE+22/PMV:POKE 77,0
240 SOUND 0,52+(PMV/2),2,4
250 IF D<0 THEN D=D+1
260 IF D>0 THEN D=D-1
270 IF LAP=200 THEN DIS=DIS+4:SIDE=SID
E+2:IF DIS>61 THEN DIS=61:SIDE=30
280 IF LAP>400 THEN LAP=0:FAC=FAC-0.07
:IF FAC<0.25 THEN FAC=0.25
290 ST=STICK(0):IF ST=15 THEN 410
300 IF ST=5 THEN PMH=PMH+2:PMV=PMV+2
310 IF ST=6 THEN PMH=PMH+2:PMV=PMV-2
320 IF ST=7 THEN PMH=PMH+2
330 IF ST=9 THEN PMH=PMH-2:PMV=PMV+2
340 IF ST=10 THEN PMH=PMH-2:PMV=PMV-2
350 IF ST=11 THEN PMH=PMH-2
360 IF ST=13 THEN PMV=PMV+2
370 IF ST=14 THEN PMV=PMV-2
380 IF PMV<32 THEN PMV=32
390 IF PMV>176 THEN PMV=176
400 POKE 53248,PMH:PM$(PMV)=C$
410 IF PEEK(53252)=0 THEN 160
420 FOR X=15 TO 200 STEP 5:SOUND 0,X,8
,8:NEXT X:FOR X=-1 TO 18:PM$(PMV+X)=CH
R$(RND(0)*254):NEXT X
430 FOR X=255 TO 0 STEP -1:POKE 704,X:
SOUND 0,X,X,X:NEXT X:SOUND 0,0,0,0:COL
OR 0:PLOT 0,0:DRAWTO 79,0
440 POKE 53248,0:FOR X=0 TO 39:T$=S$:S
$(11)=T$:NEXT X:? :? "GAME OVER (PRESS
 TRIGGER)"
450 IF STRIG(0)=1 THEN 450
460 GOTO 130
470 DATA 0,0,153,153,255,255,189,189,3
6,36,189,189,255,255,153,153,0,0
480 VNTP=PEEK(130)+PEEK(131)*256
490 VNTD=PEEK(132)+PEEK(133)*256
500 VVTP=PEEK(134)+PEEK(135)*256
510 STARP=PEEK(140)+PEEK(141)*256
520 AZ1=-1
530 AZ1=AZ1+1:FOR AZ=1 TO LEN(VAR$)
540 AZ2=PEEK(VNTP):IF AZ2>127 AND AZ<L
EN(VAR$) THEN 560
550 IF AZ2-128*(AZ2>127)=ASC(VAR$(AZ))
 THEN VNTP=VNTP+1:NEXT AZ:GOTO 590
560 IF PEEK(VNTP)<128 THEN VNTP=VNTP+1
:GOTO 560
570 VNTP=VNTP+1:IF VNTP<VNTD THEN 530
580 GRAPHICS 0:? VAR$;" ISN'T A LEGAL
VARIABLE":END
590 AZ=VVTP+AZ1*8+2:IF PEEK(AZ-2)<>129
 THEN 570
600 AZ1=LOC-STARP:GOSUB 620:AZ1=LEN:GO
SUB 620:AZ1=LEN:GOSUB 620
610 RETURN
620 AZ2=INT(AZ1/256):AZ1=AZ1-AZ2*256:P
OKE AZ,AZ1:POKE AZ+1,AZ2:AZ=AZ+2:RETUR
N
```

SWAT TABLE

LINES	SWAT CODE	LENGTH
100 - 130	SU	508
140 - 200	MB	510
210 - 320	BF	525
330 - 430	SU	546
440 - 550	NZ	511
560 - 620	FH	309

GOOD NEWS FOR COMPUTER OWNERS

Save Money On Software!

ATARI®

★TOP SOFTWARE HITS INCLUDE★

Title	Retail	Our Price
Apple Panik Atari 24K-D-	29.95	24.95
Atlantis 400/800 -C-	34.95	29.95
Baja Buggies 16K 400/800 -T-	31.95	24.95
Baja Buggies 16K 400/800 -D-	31.95	24.95
Bank Street Writer (price upon request)		
Canyon Climber 400/800 -T-	29.95	21.95
Canyon Climber 400/800 -D-	29.95	21.95
Chicken 800 -C-	44.95	33.95
Choplifter 48K -D-	34.95	24.95
Claim Jumper 800 -C-	44.95	34.95
Cross Fire 400/800 -T-	29.95	24.95
Cross Fire 400/800 -D-	29.95	24.95
Curse of RA 32K 400/800 -D-	19.95	17.95
Danger in Drindsti 400/800 -D-	19.95	17.95
Darts 400/800 -T-	29.95	24.95
Deluxe Invaders 400/800/1200 -C-	34.95	29.95
Dominoes and Cribbage 400/800 -T-	29.95	24.95
Filemanager 800 -D-	99.95	74.95
Frogger 400/800 -T-	34.95	24.94
Frogger 400/800 -D-	34.95	24.95
Gorf 400/800 1200 -C-	39.95	29.95
Hellfire Warrior 32K -D-	39.95	34.95
Hellfire Warrior 32K -T-	39.95	34.95
Home Financial Management 400/800 -T-	29.95	24.95
Humpty Dumpty/Jack & Jill -T-	29.95	24.95
Jawbreaker 400/800 -T-	29.95	23.95
Jawbreaker 400/800 -D-	29.95	23.95

Title	Retail	Our Price
Jumbo Jet Pilot 400/800 -C-	49.95	36.95
Major League Hockey 400/800 -C-	49.95	39.95
Micro-Painter 400/800 -D-	34.95	29.95
Orc Attack 400/800 -C-	49.95	39.95
Protector 2 800 -C-	44.95	35.95
River Rescue 400/800 -C-	49.95	34.95
Sands of Egypt 400/800 -D-	39.95	34.95
Shamus 800 -C-	44.95	34.95
Slime 800 -C-	44.95	34.95
Snooker & Billiards 400/800 -T-	29.95	25.95
Soccer 400/800 -C-	49.95	39.95
Spell Wizard 32K 400/800 -D-	79.95	59.95
Starbowl Football 24K 400/800 -D-	31.95	22.95
Starbowl Football 24K 400/800 -T-	31.95	22.95
Supercubes and Tilt 400/800 -T-	29.95	24.95
Temple of Apshai 32K -D-	39.95	34.95
Temple of Spshai 32K -T-	39.94	34.95
Text Wizard 400/800 -D-	99.95	79.95
Tumble Bugs 400/800 -D-	29.95	24.95
Ultima 2 400/800 -D-	59.95	44.95
Upper Apshai 32K -D-	19.95	17.95
Wizard of Wor 400/800/1200 -C-	39.95	29.95
Zaxxon 16K 400/800 -T-	39.95	29.95
Zaxxon 16K 400/800 -D-	39.95	29.95

SYMBOL INFORMATION:
DISK = D CASSETTE = T ROM = C

-BOOKS-

HAVE FUN LEARNING HOW TO PROGRAM YOUR COMPUTER.

	Retail	Our Price
Kids & the Atari	— 19.95	16.95
Kids & the T.I.	— 19.95	16.95
Kids & the Vic 20	— 19.95	16.95
Elementary 64	— 14.95	11.95

• We also carry software and accessories for Commodore 64, Vic 20 and T.I. 994-a.

maxell

MDI — BOX OF 10

FLOPPY DISKS ONLY 27.95

TEXAS RESIDENTS ADD 5 % TAX

SPECTRAVISION™

JOYSTICK ONLY 11.95

TEXAS RESIDENTS ADD 5 % TAX

FOR COMPLETE CATALOG SEND $1.00 TO:

FUTURISTICS UNLIMITED

P.O BOX 820636
DALLAS, TEXAS 75382-0636

FOR ORDERS CALL: 214-368-7320

For fast delivery send certified check, cashiers check or money order. We also accept C.O.D.'s. For MasterCard and VISA include card #, expiration date, and add 4 %. Include $3.00 for shipping and handling for orders under $100.00. No shipping charge for orders over $100.00. Personal checks require 14 working days clearance before shipping. We ship same day for most orders. Prices subject to change without notice. No returns without authorization.

MasterCard VISA

ATARI SPEED READING

Reviewed by Steve Oliver II

"If I finish a book a week, I will read only a few thousand books in my lifetime, about a tenth of a percent of the contents of the greatest libraries of our time. The trick is to know which books to read." — Carl Sagan, COSMOS

Another trick is to speed read them. Speed reading allows you to read more material and comprehend it more effectively. You learn how to set goals on the material you are going to cover and achieve them. You also learn that sometimes you only need to scan or skim for the desired facts.

From Atari, Inc., Sunnyvale, CA 94086, and Learning Multi-Systems, Inc. System requirements: 16K Atari 400/800, 410 program recorder, joystick, BASIC. Suggested retail price: $74.95.

When you speed read, you read a *group* of words at a time, absorbing them as one chunk. You actually move your eyes from one *fixation* to the next. These movements between fixations are called *saccadic* movements. In the example below, each X is a fixation and each -- > is a saccadic movement.

X-------->X-------->X-------->X-------->X

As your eyes move over a line of print, they proceed in a series.

As the number of words you can handle per fixation increases, your reading speed and efficiency increase. This is the core to speed reading: when your eyes move more efficiently, you absorb the material better. You *never* subvocalize (pronounce words to yourself), even in your mind. This is the hardest part to master.

With the theory behind you, you can now break your life-long reading habits and form new, efficient ones. Atari, in conjunction with Learning Multi-Systems, has produced an excellent speed reading package. The clever combination of the computer/tutor and an excellent workbook enable you to learn speed reading in your own home, at your own pace.

The Package

The course comes in a binder with eight units on four cassettes, a 200 page workbook (practically error-free), and a cassette with the Pacer/Timer program. This program allows you to measure and improve your speed when reading your own materials after you finish the course.

The Course

It is recommended that you complete two units of the course per week, and each takes about two hours. It is best to finish each unit in one sitting, and also to do a lot of light, outside reading in between units to keep yourself in shape.

● *Warm-Up Exercise.* This computer-run drill sharpens your mind for the following work. In particular, it hones your concentration, the single most important part of effective speed reading.

First, you select the rate at which you want the drill to proceed and then you are run through twenty-five exercises. A word is highlighted on the screen, then a list of words follows, one at a time. When the matching word appears, you press the joystick button and proceed to the next exercise. As you progress through the units, the drills become more difficult. By Unit 8, you are matching synonyms, antonyms, and entire phrases. You can use the joystick to speed up the drills anytime during the exercise.

These exercises prime your mind to work faster and concentrate better. You have the option after each Warm-up and Phrase-reading exercise to try again, and

I did these drills a few times each.

● *Phrase-Reading Exercise.* This exercise puts pages of text on the screen and moves a highlighter over groups of text at a speed you determine. This trains you to take more words per fixation and to develop rhythmic saccadic movements. As with the Warm-ups, you are encouraged to go as fast as you can — to really push yourself. In Unit 1, you start with three fixations per line and by Unit 8 you are reading an entire line at a time.

You use the joystick fire button to start and stop the drill, which is timed. The computer then gives you your first words per minute (WPM) reading of the unit. Reading from a color television screen involves a tremendous amount of eyestrain. I came away from these exercises somewhat dazed. On the positive side, however, this unit really emulates speed reading.

● *Paced and Timed Readings.* Here, you select a pace for the audio metronome, start the timer, read an article from the workbook, stop the timer, and receive your WPM rate. The audio metronome tells you when you should be at a new fixation. This teaches you to read at a constant pace, although you choose the amount you absorb per fixation. Most of the articles in these exercises, and those throughout the entire course, are light, entertaining reading (most are from airplane magazines) and are generally informative. Occasionally, however, you have to read technical or heavily-detailed material. A five or ten question quiz follows each article to test your comprehension.

● *New Techniques.* These workbook exercises are designed to teach valuable ways to improve comprehension. Skills taught are word recognition, enlarging the amount of text taken in per fixation, scanning (locating only a specific fact or key word in the material), skimming (reading only important words to get the gist of or to review the material), notetaking, and outlining material (which helps you find the main ideas).

This section is both the most valuable, because of what is taught, and also the most difficult, because the computer isn't involved at all.

● *Flexible Reading.* This exercise is the same as the Paced and Timed ones except that the audio metronome is not used. You read the material, the computer gives your WPM, and you answer the questions to compute your comprehension score. If you do the entire unit in one sitting, you should be at your optimum concentration and speed and your scores should prove this.

● *Reading Progress Graph.* You enter this unit's WPM rates and comprehension scores and the computer gives your REI (Reading Efficiency Index). This is the average number of words per minute for the Paced, Timed, and Flexible readings multiplied by the average comprehension score for each (comprehension scores are in percentages of number of correct answers). Example: in Unit 1, my WPM's were 349, 522, and 318. My comprehension scores were 80%, 50%, and 80%. So, my REI for Unit 1 was 264, which means I could read AND comprehend 264 words per minute. You then enter the previous REIs and the computer shows you a graph of your progress.

But Does It Work?

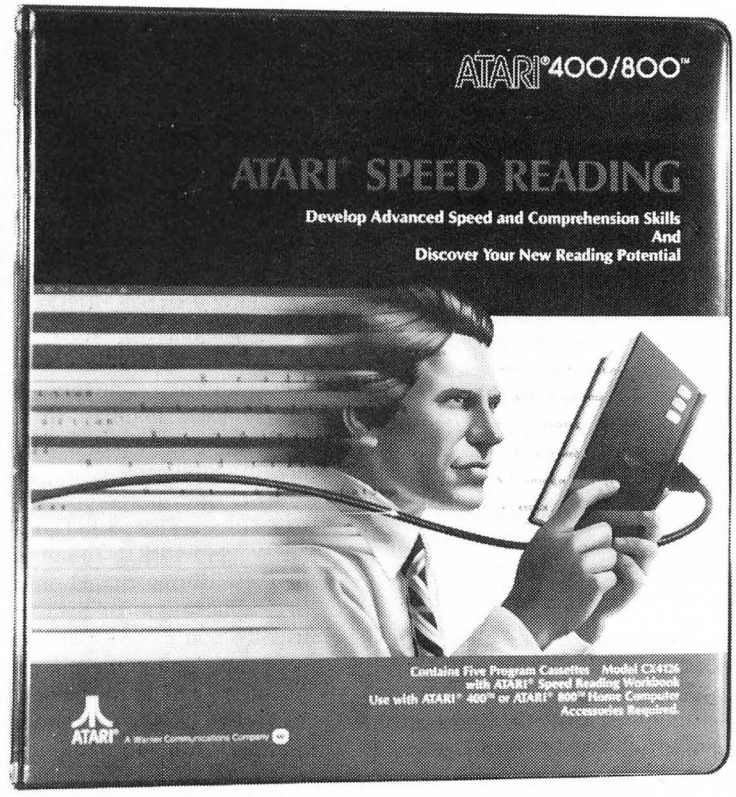
The introduction to *Atari Speed Reading* states that average readers can double their speed and comprehension, and mine was up 3.5 times! My Pretest REI was 163 and after Unit 8 it was 566. As far as I'm concerned, the *Atari Speed Reading* course works.

But is the concept of speed reading itself valid? The Atari method is just like the Evelyn Wood method in that results are high and promising — at first; but then, for most, a point comes where progress falls and you may become lost and disillusioned. Persevere, however, because things will come together. Suddenly you realize that you're reading fast and retaining most of what you read. Besides, I found that what I learned about scanning, skimming, and other reading techiques themselves were worth the investment.

So is it for you? If you want to keep up with the ever-increasing deluge of reading material, get out of it exactly what you intend to, understand it better, and are willing to throw out a life-long habit for a better one, then *yes!* It does take a lot of hard work, however.

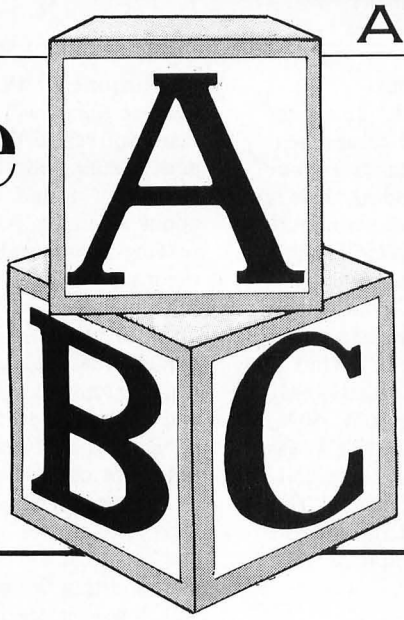
Technically, the program is fine. It lacks sensational graphics and sound, and has no aliens to zap, but it succeeds in supporting the workbook in teaching you how to speed read. As it is on cassettes, loading time is slow. I enjoyed this time however, because a man talked while the programs loaded. This "coach" informed and gave me little pep talks.

I only hope that Atari will produce more of these high-quality learning courses for its mature users. Atari has the money, staff, and the incredible machine that it takes to produce more courses like *Atari Speed Reading,* but I fear it has lost the desire since nothing substantial in the area of adult education has come from Atari recently. Maybe they're not to blame; I'm sure *Centipede* sells better than *Atari Speed Reading.*



ATARI®

The ABC Compiler



Reviewed by Carl M. Firman

Have you noticed that BASIC just seems to poke along? It pokes along because the BASIC cartridge must translate a BASIC program to the Machine Language your computer understands — a very pokey (pun) process. If you've considered Assembly Language as a solution, but find it confusing, don't despair. You don't have to learn Assembly Language to speed up your programs. A BASIC compiler will translate your BASIC programs to a form of Machine Language your computer can understand without the BASIC cartridge, and the compiled program runs about ten to 20 times faster.

The *ABC Compiler,* from Monarch Data Systems, is as easy to use as "ABC." To compile a program, all you do is boot the disk, answer the prompts for filenames and *ABC* takes over.

How *ABC* Compiles

ABC uses a special type of Machine Language called "Pseudo Codes," or P codes. The P code compiler reads your BASIC program and translates it into P codes which are then written to disk along with a P code "run time interpreter." This interpreter is the P machine that runs your P code program. Although you don't have to know anything about P codes to use *ABC,* you might want to know what *ABC's* P codes do and don't do.

Let's deal with *ABC's* limitations first. *ABC* is an integer compiler and as such, does not recognize any of Atari's floating point functions. However, this is the primary reason why programs compiled with *ABC* run faster. (Compilers which support floating point numbers run only two or three times faster than BASIC.) One of these floating point functions is the decimal fraction (i.e., 12.34). This is not a big problem, however, because Atari games, graphics and systems software rarely use decimal fractions. If you must use them, the *ABC* users manual suggests a method of scaling numbers.

Other unrecognized floating point functions are ATN, COS, EXP, LOG, SIN, SQR and RND. However, all of these may be simulated in a manner acceptable to *ABC.* The trigonometric functions, for example, can be simulated with look up tables. RND can be replaced by PEEK (53770), which is the location of Atari's random number generator. These alternative methods are discussed in the users manual. In addition, the floating point functions DEG and RAD are not supported.

ABC does not support BYE, DOS, CONT, ENTER, LIST, LOAD, RUN, NEW, and LPRINT, but you use most of these during the editing, testing and SAVEing of a BASIC program and usually don't need them in a compiled program. You can replace LPRINT by opening a line to the printer and using a PRINT # XX command (XX = the number of the opened line).

One last point — *ABC* is disk oriented and does not support CLOAD or CSAVE.

The Advantages of *ABC*

- While some other compilers can't hack "almost any program," *ABC* handles just about any program written in integer BASIC.
- To my knowledge, *ABC* is the only compiler that is *easy* to use on a single drive system. Other compilers require some knowledge of 6502 architecture or produce standard 6502 source code files which they then assemble into standard 6502 Machine Language. A compiler that requires an understanding of system architecture can be very powerful, but is definitely not for novice programmers. The compilers that produce 6502 source codes also eat up disk space fast; they produce faster code, but work best with two disk drives.
- Another important advantage of a P code compiler is software security. Without intimate knowledge of *ABC's* P machine, disassembly of your compiled program is almost impossible.
- *ABC* uses a fast, one-pass compile technique. Once the compiler disk is loaded you remove it — no additional disk swaps are required. With some other compilers, you must swap disks four times.
- *ABC* compiles an approximately 1000 line program on a 40K machine (about 2300 lines on a 48K machine). These large programs are compiled independent of actual program size in bytes or sectors.

From Monarch Data Systems, Inc., P.O. Box 207, Cochituate, MA 01178. System requirements: Atari 400/800/1200 with 40K RAM and one disk drive. Suggested retail price: $69.95.

The ABC Compiler makes your computer even more versatile and easy to use for a reasonable investment.

- Compiling takes place at the speedy clip of about 100 lines a minute. (Some other compilers take as long as ten minutes to compile a similar program.)
- *ABC* normally produces code starting at hex address 2600, but the compiler disk comes with provisions to produce code starting at hex 3000 and hex 1F00. If you want relocatable code, you compile your program twice, using different load addresses. These two programs are then compared, byte by byte, and a relocatable version is written to disk. If you intend to use your compiled programs only on your machine, however, you don't need this extra step.

Some compilers require that you structure your BASIC program in a certain order, but *ABC* supports DATA statements in any part of the program. Some other compilers require that you place all DATA statements at the beginning of the program. Also, they may not support variables in expressions like GOTO, GOSUB, DIM and RESTORE. For example, *ABC* supports statements like GOTO X, while other compilers require a statement like GOTO 125.

- *ABC* comes with an excellent, easy-to-understand, twenty page instruction manual. It covers everything you need to know to be off and running.

Testing *ABC*

For test purposes I compiled the "Happy Birthday" music player (*SoftSide,* issue #34, p. 70). The BASIC version took eighteen seconds to initialize. The compiled version was playing music in about two seconds — nine times faster.

If you want an inexpensive word processor, you can compile *Microtext (SoftSide,* April, 1982). In fact, I wrote this article on a compiled version of *Microtext.* Like a lot of us, I did not have $150 to spend on a word processor — *ABC Microtext* did the job nicely.

Compiled programs are ready to run and can be loaded with the DOS "L" option or you can rename them to "AUTORUN.SYS" and they will boot automatically.

A Bonus Program

I have included the BASIC code for a program called "ABCLOADER". Type it in exactly as is. Be very careful when typing the DATA statements. Save the BASIC version to disk. Compile the program using a hex load address of 3000. (The instructions for different load addresses are on page 15 of the users manual.) Name the compiled file "AUTORUN.SYS". When you boot a disk with "ABCLOADER", it displays the compiled programs on your disk along with a number; simply type the program number and "ABCLOADER" runs that program. "ABCLOADER" only loads programs compiled with the normal load address of hex 2600. It will not load BASIC programs.

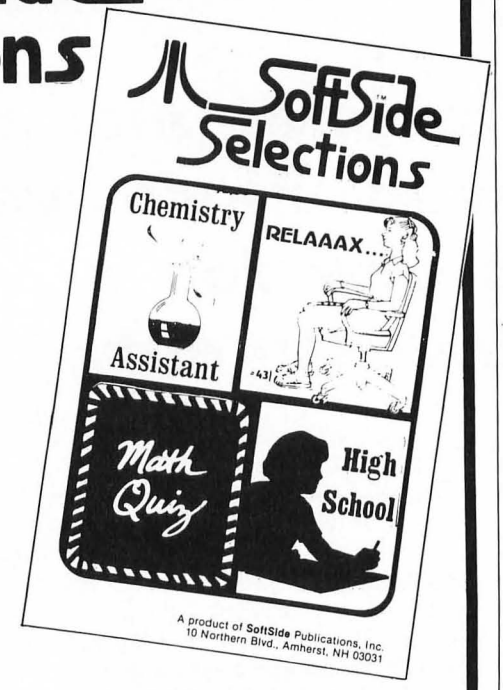
In Conclusion

The *ABC Compiler* makes your computer even more versatile and easy to use for a reasonable $69.95 investment. Now, you can write programs that run as fast as Forth or "C" without learning a new lanquage. *ABC* does just what the advertisement says; it makes BASIC "fly".....Happy landings!

```
 SS SS SS SS SS SS SS SS SS SS SS
 SS                             SS
 SS        Atari BASIC          SS
 SS        'ABC Loader'         SS
 SS     Author: C. Firman       SS
 SS     Copyright © 1983        SS
 SS SoftSide Publications, Inc SS
 SS                             SS
 SS SS SS SS SS SS SS SS SS SS SS
```

```
1 IF PEEK(6)=1 THEN ? "PLEASE REMOVE T
HE CARTRIDGE":END
2 DATA 162,16,32,173,6,134,207,104,104
,157,69,3,104,157,68,3,169,4,157,74,3,
169,3,157,66
3 DATA 3,32,86,228,16,3,76,166,6,169,2
03,157,68,3,169,0,157,69,3,169,2,157,7
2,3,169
4 DATA 0,157,73,3,169,7,157,66,3,32,86
,228,16,6,192,136,240,92,208,96,169,25
5,197,203,208
5 DATA 4,197,204,240,210,169,205,157,6
8,3,169,0,157,69,3,32,86,228,16,2,48,6
9,165,207,240
6 DATA 14,165,203,141,224,2,165,204,14
1,225,2,169,0,133,207,165,203,157,68,3
,165,204,157,69,3
7 DATA 165,205,56,229,203,157,72,3,165
,206,229,204,157,73,3,254,72,3,208,3,2
54,73,3,32,86
8 DATA 228,16,137,192,3,240,133,76,166
,6,32,173,6,108,224,2,152,133,212,169,
0,133,213,169,12
9 DATA 157,66,3,32,86,228,96
20 GRAPHICS 0:POKE 559,0:POKE 752,1:SE
TCOLOR 4,11,4:SETCOLOR 2,11,4:SETCOLOR
 1,0,14:DIM LIN$(15),LIB$(768)
22 FOR A=1536 TO 1717:READ B:POKE A,B:
NEXT A
25 POKE 559,0:? "}";:? "THE COMPILED P
ROGRAMS ON THIS DISK ARE":? :LIN$="D1:
":LIN$(4)="*.*"
30 LIB$(1,1)=" ":LIB$(768)=" ":LIB$(2)
=LIB$:CLOSE #1:OPEN #1,6,0,LIN$:NUM1=1
:FLAG1=1
35 FOR FILE=1 TO 64
```

SoftSide® Selections™



The Magazine *Especially* For Your Atari® Computer

Bound into the center of this issue, you'll find ***SoftSide Selections,*** the handy, pull-out booklet with program listings for your computer. If you bought your copy of ***SoftSide*** at a newsstand, your booklet contains this issue's Front Runner, ***Relaaax...,*** for the Atari®, IBM® PC, Apple®, and TRS-80®.

This issue, ***SoftSide Selections*** for the Atari features:

- ***Relaaax...*** — you always get the current issue's Front Runner!

SoftSide has brought you programs that will keep you up at night — this one may just help you get to sleep. Peter Favaro, a school psychologist and educational and recreational video game design consultant, and Richard M. O'Brien, a clinical psychologist and professor of psychology at Hofstra University, have developed this system to help you relieve tension, and ***Relaaax...***

- ***Math Quiz*** — This colorful educational program can help kids practice their arithmetic.

Enhanced Disk and Cassette Versions

If you don't like typing, you need the ***SoftSide*** Disk or Cassette Version. The Cassette Version (CV) has all of the programs in ***SoftSide Selections*** on one handy cassette tape plus the latest installment of the ***SoftSide*** Adventure Series.

SoftSide Disk Version (DV) has everything that the CV has, plus a *bonus program.* You get all this, and the benefits that only a disk can offer: speed, reliability, and versatility.

- **Atari DV Bonus Program: *Chemistry Assistant***

This educational program performs over ten common chemical calculations.

- **DV and CV Bonus: *The SoftSide Adventure Series***

This issue's Adventure — ***High School,*** by David Pleacher.

You are in high school, and who said that going to high school is not an adventure?

DV — $19.95
CV — $14.95

To order your copy of this issue's Disk or Cassette Version, or to subscribe to either of the ***SoftSide*** media versions, see the bind-in cards opposite page 64.

ATARI®

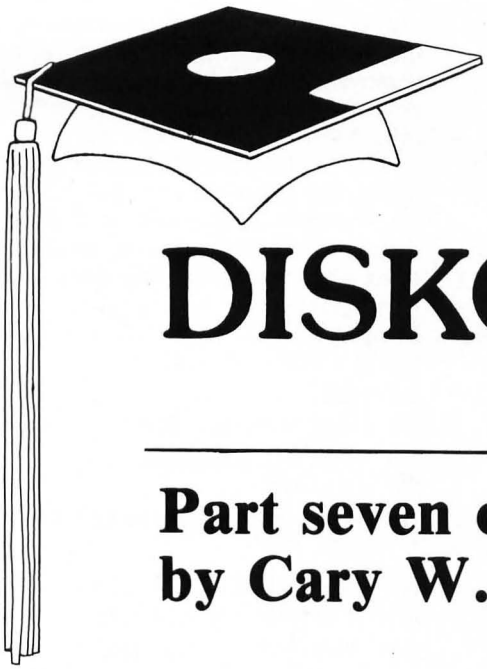
The ABC Compiler, *continued*

```
40 INPUT #1,LIN$:FOR NUM3=4 TO 12:IF L
IN$(NUM3,NUM3+3)=" SYS" THEN POP :GOTO
 40
42 IF LIN$(5,10)="FREE " THEN POP :GOT
O 110
45 NEXT NUM3
50 IF LIN$(2,2)<>" " THEN GOTO 110
55 NUM2=NUM1:FOR NUM3=3 TO 10:IF LIN$(
NUM3,NUM3)=" " THEN 65
60 LIB$(NUM2,NUM2)=LIN$(NUM3,NUM3):NUM
2=NUM2+1:NEXT NUM3
65 IF NUM3<10 THEN POP
70 IF LIN$(11,11)=" " THEN 80
75 LIB$(NUM2,NUM2)=".":NUM2=NUM2+1:FOR
 NUM3=11 TO 13:LIB$(NUM2,NUM2)=LIN$(NU
M3,NUM3):NUM2=NUM2+1:NEXT NUM3
80 IF FLAG1=3 THEN ? :FLAG1=1
85 IF FILE<10 THEN ? " ";
90 ? FILE;" ";LIB$(NUM1,NUM1+11);"  ";
:FLAG1=FLAG1+1
94 REM THE LAST = IN LINE 95 IS A CONT
ROL CHARACTER
95 NUM1=NUM1+12:IF FILE=32 THEN ? "
 =There are more files - press RETURN
  to continue....":POKE 559,34
100 IF FILE=32 AND PEEK(764)<>12 THEN
100
105 POKE 764,255:POKE 559,0:NEXT FILE
110 ? :? :? LIN$(1,4);"SECTORS AVAILAB
LE"
115 ? :? "Enter number of file to load
:";:POKE 559,34:TRAP 115:INPUT N
120 IF N<1 OR N>FILE-1 THEN GOTO 115
125 LIN$="D1:":LIN$(4)=LIB$(N*12-11,N*
12):CLOSE #1:OPEN #1,4,0,LIN$:GET #1,N
:CLOSE #1:IF N<>255 THEN GOTO 135
130 ? "}":POSITION 8,10:? "LOADING  ";
LIN$:X=USR(1536,ADR(LIN$)):END
135 ? "}":POSITION 11,10:? "NOT A BINA
RY FILE":FOR I=1 TO 5000:NEXT I:GOTO 2
5
```

SWAT TABLE

For **ABC LOADER**

LINES	SWAT CODE	LENGTH
1 - 7	TE	584
8 - 30	BV	533
35 - 85	YY	463
90 - 125	MS	594
130 - 135	KJ	149



APPLE DISKOURSE

Part seven of a series by Cary W. Bradley

Using DOS Within a Program

This article moves away from the past year's semi-technical, utility-oriented approach, to some useful techniques for incorporating disk functions into your programs. I won't duplicate what is in the Apple DOS manual, but will demonstrate the practical uses and implications of disk programming methods. The objective is to make your programs as bug-free, fast and efficient as possible. As always, I will refer at times to the Apple manuals, and will assume that you have at least some familiarity with their contents.

The topic of sequential files is a natural place to start, because this type of file functions very much like a series of DATA statements, something most beginners are familiar with. Although the statements used with these files differ slightly from those used with DATA, the principle is much the same. In both cases, the computer reads data from the beginning, one item at a time, until instructed to stop reading, or until it runs out of data. You can go back to the beginning of a set of DATA statements or a sequential disk file at any time. Finally, both situations can have unpleasant results if you are not careful to prevent your program from attempting to read data when the file or the DATA statement ends.

Some Preliminaries

One of the worst habits beginning programmers can develop is sitting down at the keyboard and typing program lines with little or no idea where they're headed. You won't have too much trouble with simple programs, but when you set out to do anything significant, you need to plan ahead before flipping the power switch. Of course, you should outline the structural logic of the program, but the concern here is the data. If you plan to use disk files to store data, you should know what your data will *look like* before you begin pounding the keys. Your skill in structuring data will increase with experience, and you will acquire an understanding of when and why to use a particular method.

Consider, for example, the simple matter of sequential disk files versus DATA statements within a program. DATA statements are handy for giving values to variables, such as in initialization, when the alternative is a truckload of assignment statements. If your program uses a set of data that must be the same each time the program is run, why bother with a disk file? It's usually slower than the READ-DATA combination, and is another detail to worry about when you copy the program to another disk. On the other hand, if your data set changes from one run to the next, or if memory space is a problem, a sequential file could be the best alternative.

Still, with a sequential file you are restricted to moving through the file from beginning to end. Although they have a little more flexibility than DATA statements, sequential files definitely are the wrong choice in some situations (I'll discuss random-access or relative files in a future article). After deciding to use a sequential file, you must create program segments which write to the file and which read it. Several types of routines may be necessary for a given file, and they may or may not be in the same program.

Sequential File Techniques

The following eight program segments are examples of methods commonly used for reading and writing sequential files. The task (creating multiplication tables) accomplished by these programs is rather trivial, but it illustrates the different ways to write data into a file and read it back. You, as a programmer, must decide which method is most appropriate for a given application. In these examples, the user chooses the base for the table, and in all but the simplest case, the range of multipliers to be used for the table. Four methods are shown, each with two parts — one to generate the data file, the other to read it back and display the result.

The simplest situation is most like the ordinary DATA statement. After the user chooses a number, the table consists of that number multiplied by two through twelve. Here is the program that allows the selection of the base and writes the eleven products to a disk file:

Listing 1

```
100 D$ =  CHR$ (4)
200  INPUT "MULTIPLICATION TABLE
     FOR WHAT NUMBER?:";NUM
300  IF NUM <  = 0 THEN  PRINT "E
     NTER A POSITIVE NUMBER, PLEA
     SE...": GOTO 200
400  PRINT D$;"OPEN TABLE1"
500  PRINT D$;"DELETE TABLE1"
600  PRINT D$;"OPEN TABLE1"
700  PRINT D$;"WRITE TABLE1"
800  PRINT NUM
900  FOR M = 2 TO 12
1000  PRINT M * NUM
1100  NEXT M
1200  PRINT D$;"CLOSE TABLE1"
```

The important point is to know that the file always contains exactly twelve numbers — the base and eleven products. An array stores the products easily. The file might be read back like this:

Listing 2

```
100 D$ =  CHR$ (4): DIM MULT(11)
200  HOME : PRINT "READING DATA"
600  PRINT D$;"OPEN TABLE1"
```

↪

Apple Diskourse, ***continued***

```
700  PRINT D$;"READ TABLE1"
750  INPUT NUM
800  FOR M = 1 TO 11
900  INPUT MULT(M)
1000  NEXT M
1100  PRINT D$;"CLOSE TABLE1"
1200  HOME : PRINT "MULTIPLICATIO
     N BY ";NUM
1300  FOR M = 1 TO 11
1400  PRINT NUM;" TIMES ";M + 1;"
      = ";MULT(M)
1500  NEXT M
```

In many cases, however, the file will not have exactly the same number of data items every time the program is run. Suppose you want to give the user the option of determining how far the table will go — not just from two to twelve, but to any number. One way to handle this type of situation is the following:

Listing 3

```
100 D$ =  CHR$ (4)
200  INPUT "MULTIPLICATION TABLE
     FOR WHAT NUMBER?:";NUM
300  IF NUM <  = 0 THEN  PRINT "E
     NTER A POSITIVE NUMBER, PLEA
     SE...": GOTO 200
310  INPUT "TABLE FROM 2 THROUGH:
      ";LIM
320  IF LIM <  = 2 THEN  PRINT "G
     REATER THAN 2, PLEASE...": GOTO
     310
400  PRINT D$;"OPEN TABLE2"
500  PRINT D$;"DELETE TABLE2"
600  PRINT D$;"OPEN TABLE2"
700  PRINT D$;"WRITE TABLE2"
750  PRINT NUM
800  FOR M = 2 TO LIM
900  PRINT M * NUM
1000  NEXT M
1050  PRINT  - 99
1100  PRINT D$;"CLOSE TABLE2"
```

End-of-Data Markers

After writing all the values into the file, the program placed an additional number, −99, at the end of the file. This is an end-of-data marker, often called a sentinel or trailer. Numbers such as −99 or −99999 are good markers, because they usually do not result from the operation being performed. When reading back the file, the computer checks each value to see if it is the end-of-data marker. When it finds the marker, it stops reading and closes the file. In the next listing, looking for a negative number is sufficient, since the program creates only positive data. I could have used any negative number in place of −99 and the result would be the same. The end-of-data marker can be any character appropriate to a given situation, including a special character string when you are reading string data from a file:

Listing 4

```
100 D$ =  CHR$ (4):MULT = 2
200  HOME
600  PRINT D$;"OPEN TABLE2"
700  PRINT D$;"READ TABLE2"
750  INPUT NUM
775  PRINT "MULTIPLICATION BY ";N
     UM
800  INPUT PRODUCT
900  IF PRODUCT < 0 THEN 1100
1000  PRINT NUM;" TIMES ";MULT;"
     = ";PRODUCT
1010 MULT = MULT + 1
1020  GOTO 800
1100  PRINT D$;"CLOSE TABLE2"
```

Sometimes, placing a number in the file is necessary to indicate how many data items follow. You can do this only when you know in advance the number of items the file will contain, as in Listing 5. This is especially useful when the number of data items is a quantity needed for another calculation in the program, prior to reading the data:

Listing 5

```
100 D$ =  CHR$ (4)
200  INPUT "MULTIPLICATION TABLE
     FOR WHAT NUMBER?:";NUM
300  IF NUM <  = 0 THEN  PRINT "E
     NTER A POSITIVE NUMBER, PLEA
     SE...": GOTO 200
310  INPUT "TABLE FROM 2 THROUGH:
      ";LIM
320  IF LIM <  = 2 THEN  PRINT "G
     REATER THAN 2, PLEASE...": GOTO
     310
330 COUNT = LIM - 1
400  PRINT D$;"OPEN TABLE3"
500  PRINT D$;"DELETE TABLE3"
600  PRINT D$;"OPEN TABLE3"
700  PRINT D$;"WRITE TABLE3"
750  PRINT NUM
800  PRINT COUNT
850  FOR M = 2 TO LIM
900  PRINT M * NUM
1000  NEXT M
1100  PRINT D$;"CLOSE TABLE3"
```

Since calculating how many products will be generated and written is possible, place that number before the actual data in the file. You can read back the count and use it as an index for a loop which controls the processing of the data that follows:

Listing 6

```
100 D$ =  CHR$ (4):MULT = 2
200  HOME
600  PRINT D$;"OPEN TABLE3"
700  PRINT D$;"READ TABLE3"
750  INPUT NUM,COUNT
775  PRINT "MULTIPLICATION BY ";N
     UM
800  INPUT PRODUCT
1000  PRINT NUM;" TIMES ";MULT;"
     = ";PRODUCT
1010 COUNT = COUNT - 1:MULT = MUL
     T + 1
1020  IF COUNT >  = 1 THEN 800
1100  PRINT D$;"CLOSE TABLE3"
```

Using Error Messages

Another method for reading an indeterminate amount of data is to let the computer read until it runs out of data. This generates a DOS error, but the ONERR GOTO statement allows you to return to the main program. A nice feature of this method is that it puts just the data in the file, without the housekeeping numbers required by the previous two methods:

Listing 7

```
100 D$ =  CHR$ (4)
200  INPUT "MULTIPLICATION TABLE
     FOR WHAT NUMBER?:";NUM
300  IF NUM <  = 0 THEN  PRINT "E
     NTER A POSITIVE NUMBER, PLEA
     SE...": GOTO 200
310  INPUT "TABLE FROM 2 THROUGH:
      ";LIM
320  IF LIM <  = 2 THEN  PRINT "G
     REATER THAN 2, PLEASE...": GOTO
     310
400  PRINT D$;"OPEN TABLE4"
500  PRINT D$;"DELETE TABLE4"
600  PRINT D$;"OPEN TABLE4"
700  PRINT D$;"WRITE TABLE4"
750  PRINT NUM
850  FOR M = 2 TO LIM
900  PRINT M * NUM
1000  NEXT M
1100  PRINT D$;"CLOSE TABLE4"
```

And to read back the table:

Listing 8

```
100 D$ =  CHR$ (4):MULT = 2
200  HOME
300  ONERR  GOTO 1030
600  PRINT D$;"OPEN TABLE4"
700  PRINT D$;"READ TABLE4"
750  INPUT NUM
775  PRINT "MULTIPLICATION BY ";N
     UM
800  INPUT PRODUCT
1000  PRINT NUM;" TIMES ";MULT;"
     = ";PRODUCT
1010 MULT = MULT + 1
1020  GOTO 800
1030 E =  PEEK (222)
1040  IF E <  > 5 THEN  PRINT "SO
     METHING'S WRONG!": PRINT "E
     RROR CODE = ";E: GOTO 1100
1050  PRINT "NORMAL END."
1100  PRINT D$;"CLOSE TABLE4"
```

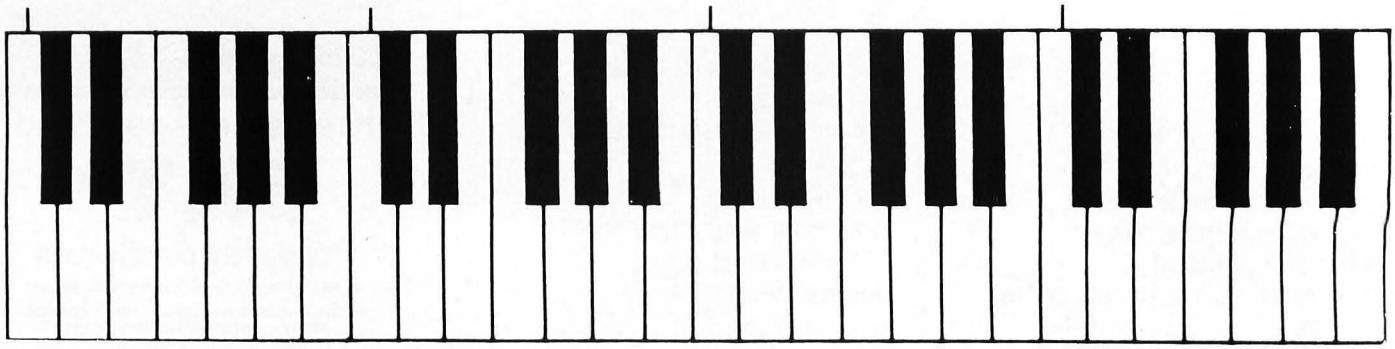
An important part of this procedure is checking the error code, which is delivered by DOS to memory location 222 (decimal). When ONERR GOTO is in effect, make sure the error you anticipated is the one you got. Load this program and try to run it with your disk drive door open. Without the error test at statement 1040, this program would have interpreted your disabled drive as a signal that all the data had been read. This could spell disaster. The DOS manual has additional error codes which can give you tremendous control in all sorts of bad situations, if you know how to use them. I'll tackle this topic at a later time.

These eight simple techniques can be adapted to fit a large percentage of programs for which sequential disk files are suitable. Of course, they can be changed to read text, as well as numbers, as long as your file provides a number where a number is expected, and a string where a string is expected. The method you use to read the file must match the method you used to write it.

Compatible With IIe

I have put the Diskourse Utilities from previous articles to the test on an Apple IIe. They all worked — even those with machine language calls to the system monitor and BASIC. So you old-timers who are upgrading to the IIe can still use all the earlier DOS utilities, and new IIe owners can pick up back copies of *SoftSide* and run them with no hassle.

Music Theory Software



for the Apple

Reviewed by Steve Birchall

- **Music Games by Lydia Bell, Howard W. Sams & Co., Inc., 4300 W. 62nd Street, Indianapolis, Indiana 46268. System requirements: 48K Apple II+ and (optional) game paddles. Suggested retail price: $39.95.**
- **MusicMaster I by Wolfgang Kuhn and Paul Lorton, Syntauri Corp., Palo Alto, CA 94306. System Requirements: 64K Apple II and alphaSyntauri™ music synthesizer. Suggested retail price: $150.**
- **Notes and Keys by Jay Dickinson, Marci Dickinson, and Christine Ferreira, Passport Designs, Half Moon Bay, CA 94019. System Requirements: 64K Apple II and Soundchaser Music System. Suggested retail price: $195.**
- **Music Tutor by Charles G. Boody, Passport Designs, Half Moon Bay, CA 94019. System Requirements: 64K Apple II and Soundchaser Music System. Suggested retail price: $195.**

A serious problem in planning a music school curriculum is the poor preparation of most entering students. Few high schools teach music theory, and at the college level, the faculty finds itself teaching the musical equivalent of reading and writing. Obviously this is not college level material, and the goal is to get past this point as quickly as possible.

Most students will be able to read music in the clef their instrument or voice normally uses, but they know little of scales, keys, and harmony. Ear training (learning to recognize chords and intervals by their sound) will be new to them. Like teaching arithmetic or spelling, much of this requires extensive drill and repetition. With music, however, a live teacher must play the drill materials on the piano — not an economical use of a virtuoso cellist's or a composer's time and skills. The accepted way to teach this basic material is Freshman Theory, where a professor leads the class by rote through drills on intervals, scales and chords. For ear training, he plays two notes on the piano and asks them to name the interval, or plays chords and ask them to identify the type.

With the advent of electronic pianos (or at least electric keyboard instruments with not-too-objectionable sound qualities), came the infamous piano lab. This was a kind of purgatory to which freshmen and graduate teaching assistants were assigned until they acquired wisdom. Imagine a room full of students wearing headphones and a teacher listening at a master control console, buzzing from student to student, giving individual help as needed. The system works for many of the required skills, but not all. Sight singing is a spectacular disaster, and ear training somewhat less so.

Microcomputers improve musical pedagogy by permitting students to work on their own in reasonable privacy. More importantly, they enable everyone, not just students in a music school, to study the ABCs of music without a teacher. They provide a route for high school students who want to attend a music school and whose secondary schools do not offer the necessary college preparation, to acquire that training. They also open the door to adults wishing to learn about music theory later in life.

Music Games

This set of music theory exercises is designed for children (eight and up) and

adults. The games are derived from the old Seashore tests of musical aptitude — the first one plays (and also displays in musical notation) a melody with one note missing and asks you to fill in the correct one. In true Skinnerian fashion, a correct answer elicits a reward of a spritely tune. The game supplies the correct answer after three unsuccessful tries and becomes more difficult when more notes are missing. This exercise requires a student to listen to the overall melodic contour and to judge the interval (on one or both sides of the missing note) to determine the correct note. This is a useful, practical skill, particularly for playing melodies by ear.

While *Music Games* has a sound basic premise, it has some serious flaws. Until these flaws are fixed, I cannot recommend this software. The main problem is bad intonation. The first exercise on the menu is to choose a tempo to play back a piece by C.P.E. Bach, and the tuning is so bad you will wince in pain. Suffice it to say that you do not teach people how to hear intervals and chords when the source instrument is badly out of tune. If the publisher fixes this problem, *Music Games* might be more suitable for general use. Another problem in the first game: the booklet spells the composer's first name "Karl" and the screen uses both a "K" and a "C." A small matter, but symptomatic of a general sloppiness in the package.

There are more errors and confusing inconsistencies. For instance, in the game, Measure Count, the printed instructions (p. 10) are full of errors, such as:

- "2. Select the bottom number for the key signature — either 4,3, or 2."
- "3. Select the top number for the key signature — either 2,3,4,5, or 6."

First of all, the intended term is "time signature," not "key signature." Although you could excuse this as an unintended editing mistake, what is the number three doing in the bottom of a time signature? Western music does not have "third notes" — they must be treated as triplets.

In the Flash Cards game, a note appears on the staff, and you are asked to give its pitch letter name. When you enter an answer, the Apple responds with a beep for a correct answer, or a buzz for a wrong one. This is fine for that purpose, but to see a succession of drastically different pitches on the staff on the screen, while exactly the same pitch sounds for every correct answer, is to *misuse* the basic technique of the exer-

↪

Music Theory Software, ***continued***

cise. What it should do is train the student to *hear* the *pitch* of the displayed note. If the computer emits the same pitch for every note, the design of the exercise is extremely poor.

I do not recommend *Music Games* in its present form; with the necessary corrections, however, it would be a useful package.

MusicMaster I

Designed to work on the alphaSyntauri synthesizer, this series of exercises is aimed at college and high school students needing training in elementary music theory. The first book/disk presents intervals, scales, and triads with progressively more difficult exercises in each category. The program keeps track of each student and records his score.

In the Scales section, four types of exercises train the student's mind and ear. The first one plays and displays a scale, and then asks him to identify its mode: major, natural minor, harmonic minor, or melodic minor. The second section asks for a scale type, then asks you to play it on the musical keyboard. The third displays a scale and asks you to play it. Finally, in the fourth section, it asks you to play a scale, given only the type and the starting note. These are familiar exercises to anyone who has taken a theory course.

Intervals are treated in the same way, starting with major and minor seconds, and minor thirds, progressing to wider intervals in groups of three. The last choice mixes all intervals within an octave. Triad identification starts with root position, then first and second inversions, and finally presents them in random order.

Various options are available to change clefs, upward or downward order of presentation, tempo, and tonal characteristics (on the Syntauri, this is a considerable range of possibilities).

This first book of the series effectively teaches basic music theory, and is in actual use in several colleges. Although the hardware is expensive (an extra $2000 beyond the Apple, plus an amplifier and headphones or speakers), it is less expensive than an extra faculty member, or a bunch of graduate assistants. More to the point, students can work on their own, at their own pace. They can have as much practice as they need, rather than only the in-class drills of the past.

Notes and Keys

Notes and Keys is Soundchaser's introduction to basic elements of music theory. It covers key signatures, naming notes (both on the staff and on the keyboard), scales, intervals, and triads, time signatures and note values, but not in as much detail as the Syntauri software. Also, this material is more related to keyboard instruction than to ear training. It does not stress the hearing aspects of musical training, but asks you to play your response on the keyboard. Both are valid, but one without the other is not adequate.

Music Tutor

Music Tutor takes care of the ear training functions quite well. It takes you through intervals (harmonic and melodic, and larger than an octave), chords (including seventh chords) and their inversions, and melodic games with missing notes. That should keep most people busy for quite some time, because acquiring all those aural skills will take many months of practice. Once again, like the Syntauri software, the hardware requirements are expensive but the benefits are considerable.

Teaching music theory with microcomputers offers some significant advantages over traditional methods. Letting students advance at their own pace, letting them practice as much or as little as they need to, giving them immediate feedback on the correctness of their answer are all obvious. More subtle aspects involve the nature of the Soundchaser and alphaSyntauri. The student can change the timbre at any time, and is not restricted only to the sound of the piano. This relieves boredom, helps to transfer the skills to other instruments (an aspect of the traditional piano method which is troublesome), and negates the effects of unintended cues from acoustical instruments. The last is an important point, because certain keys on a piano can have noise components or timbral deviations which prompt students that a particular key has been struck. Often this actually occurs on a conscious level, and such learning has little value for hearing pitch *relationships.* With electronic tone generation and frequent changing of the spectral character, this problem does not occur, and learning is more effective. Both the Syntauri and Soundchaser music theory software are competently done and provide effective instruction.

WEERD!

Reviewed by Mark Renne

If *E.T.* has lulled you into a false sense of security regarding the character of extraterrestrial beings, beware! It seems outer space is crawling with the "green menace," threatening to eliminate life on this planet. *Weerd* features over 25 different "bad guys" trying to accomplish your instant annihilation.

The game gets its name from the many strange shapes and sizes of the alien attackers. Some aliens consist of two, three, or more ships stuck together. This means that you'll need to fire several shots from your laser to obliterate these clever Klingons.

These fellows are not limited to one shot. Frankly, some release a hail of fire resembling an M-16. Others (must be those without family back home) even attempt to crash into your laser, while some don't seem to fire at all. Quite a *weerd* collection, all in all.

Of course, you are not without protection. Besides your laser, you have three temporary shields available for each base. The shields last for a short amount of time, then disappear after a warning flash or two. These shields only protect against enemy shots; they are no defense against ships trying to crash into you.

The game features standard control keys, abort game, game pause, joystick option, and reverse video. The latter changes all things black to white and vice versa. One or two players may participate. You can save thirty high scores on the disk version — that's right, thirty high scores!

As with all Big Five games, game play is smooth. All commands function as advertised, graphics are flicker-free, and several hundred plays are required to master the game. *Weerd* doesn't offer any radically new features, but it does provide another quality challenge to space arcaders everywhere.

From Big Five Software, P.O. Box 9078-185, Van Nuys, CA 91409. System requirements: TRS-80 Model I or III with 16K cassette and disk. Suggested retail price: $19.95 (cassette and disk).

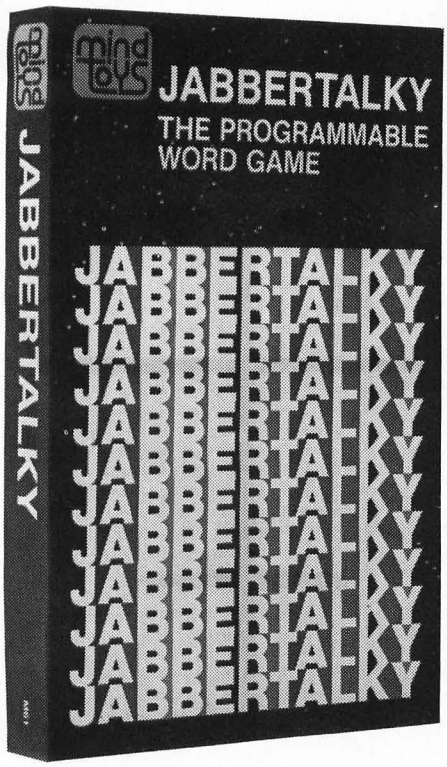
TRS-80®

JABBERTALKY

Reviewed by Mark E. Renne

Does your child spend countless hours in front of the computer in pursuit of yet a higher score at his/her favorite arcade game? Are you concerned that perhaps his/her computer time could be better spent? If so, you should consider the purchase of *Jabbertalky,* a game that teaches while it entertains. It has two built-in word games and three built-in vocabularies. It is also user expandable to include as many vocabularies as the disk will hold.

The program is broken down into four different sections — Free Verse, Alphagrammar, Cryptogrammar, and Jabbergrammar. Each section has eight degrees of difficulty. The three built-in bases are general, headline and vocab. General consists of common words and places, headline consists of words that might be in a newspaper headline, and vocab is the strange tongue of the jabbertalker (computer).



Free Verse is simply the random mumblings of the computer using the vocabulary and sentence structure defined by the program or the user. You can use Free Verse to get an idea of what words will be used in the coming games or as a reading tool for children.

The first word game, Alphagrammar, displays a sentence as blocks and spaces. Your challenge is to figure out what letters belong in the blocks. At the top of the screen all the letters are listed with a number indicating how many times that letter appears in the sentence. By trying different letters in different places, you attempt to unscramble the sentence in the least amount of time. You control the cursor with the arrow keys. You also have the option of listing the vocabulary to help decode the puzzle. If you want to give up, press CLEAR and the completed sentence appears.

In Cryptogrammar, you play secret agent. You are given a scrambled sentence and your mission is to unscramble it by rearranging the letters into the proper sequence. On the easy levels, the computer tells you if you try to change incorrect letters, but not on the more difficult levels. It is easy to get confused and switch letters indefinitely. This game is also timed.

In the last section, Jabbergrammar, you set up your vocabulary. Here you may view existing word lists or create new ones. This option makes the program an expandable learning tool. You can put your child's vocabulary in the computer and let him/her Free Verse all night long. Sentences are formed at random from the words you insert by a structure that you also must set up.

In summary, *Jabbertalky* is more a learning tool than an arcade game. It is an interesting attempt at Computer Assisted Instruction. Because home education via computers is a nearly untapped, but very important, resource, I hope this is just the first of many such programs from Epyx, Inc.

From Epyx, Inc., 1043 Kiel Ct., Sunnyvale, CA 94088. System requirements: TRS-80 Model I or III with 32K and disk drive. Suggested retail price: $29.95.

YOUR IBM® PERSONAL COMPUTER IS ABOUT TO BECOME MORE POWERFUL THAN YOU EVER DREAMED POSSIBLE.

BOSTON
Saturday-Monday, October 8-10, 1983
(Columbus Day Weekend)
Bayside Exposition Center
10:30 AM to 5:30 PM Daily

Produced by Northeast Expositions Inc., the people who brought you Applefest, CP/M '83, Softcon, and the National Computer Shows.
826 Boylston Street, Chestnut Hill, Massachusetts 02167, 617-739-2000 or 800-841-7000.
For information on exhibiting at PC '83, please call 800-343-2222.

ANNOUNCING PC '83
THE INTERNATIONAL CONFERENCE AND EXPOSITION FOR PC USERS

In the last few months, *thousands* of exciting new products have become available to make your IBM Personal Computer work in hundreds of innovative, landmark applications. By adding the right software, plug-in cards, peripherals and accessories, you can make your PC more productive and useful than you ever dreamed possible.

And now there's finally a way to learn about and try out all of these products: PC '83, the first international event *exclusively* for PC users.

THE LARGEST EVER

The exposition portion of PC '83 is the largest gathering of PC-compatible products ever held. You'll find, under one roof, more than 400 exhibits of software for business, professional, home, personal and education applications; plug-in cards for memory expansion, telecommunications, speech and many other functions; hundreds of peripherals, including printers, networking systems and innovative new storage devices; plus dozens of publications, books and invaluable support services.

PC '83 brings together the *entire* PC world.

RELIABLE AND COST-EFFECTIVE

Think about the different vehicles available for buying software and add-ons for your PC. You can shop at local computer stores, but they usually have a very limited selection of products and a less-than-expert staff. You can buy through mail order—and take some big risks if you get products that don't work or don't do what you expected.

At PC '83 you have the best of all worlds. You can see *all* the different PC-compatible products in one location at one time. You can talk with factory experts rather than just local salespeople. You can 'test drive' each product and decide if it's right for you. And when you find what you're looking for, you can buy it right on the spot.

Best of all, you'll save hundreds—even thousands—of dollars at PC '83. Most of the products shown at PC '83 are sold at special show prices, so you won't have to look hard to find incredible values. And because PC '83 will help you make more informed purchase decisions, you'll end up with products that are better suited to your applications.

A REMARKABLE CONFERENCE PROGRAM

The conference program at PC '83 isn't just a warmed-over, pared-down version of other computer conferences. The seminars, workshops and forums at PC '83 are the result of months of research and planning targeted toward one specific goal: helping you get absolutely the most out of your IBM Personal Computer.

The PC '83 Conference Program features world-renowned PC authorities speaking at over 100 educational sessions. SEMINARS provide an in-depth look at a broad range of PC uses in business, home, and education, emphasizing "how-to's" and telling you specifically what you need to know—in non-technical language—to use your PC in dozens of different applications. TUTORIALS are intensive, easy-to-understand sessions specially designed for both the beginning PC user and the PC veteran looking to learn a new technique. APPLICATION WORKSHOPS teach the ins and outs of an application: how it works, what it does, and what are some of the popular products available for it. INDUSTRY WORKSHOPS are specially designed for those in the business of producing or selling IBM-PC compatible hardware and software. Each workshop features a panel of distinguished authorities from the PC world and covers a topic of critical importance to PC support companies.

SOFTWARE and HARDWARE SPOTLIGHTS address one of the biggest problems that virtually all computer users face: how to decide which software and hardware packages are best for your applications. Each spotlight provides a detailed discussion and demonstration of a group of products, covering their features, capabilities, *and* limitations. Experts are on hand at each of these sessions to answer all your questions.

EASY ON YOUR FEET

We've thought of a lot of details to make your visit to PC '83 just a little more pleasant. As with all of our events, the show is fully carpeted, so your feet won't give out before you've seen all the exhibits.

Our unique Conference and Exhibits Guide helps you quickly find the exhibits and educational sessions you want to attend—even if you can't remember their names. And the Guide is provided free of charge to all attendees.

If you're traveling some distance to attend PC '83, we'll arrange a discounted hotel room near the show site.
On request, we will also provide information on things to do, places to visit and where to eat in Boston—to make your stay a memorable and relaxing one.

A TOTAL SUPPORT SERVICE

Let's face it, getting support for your PC is tough. The PC magazines, newsletters, user groups and retailers are certainly helpful. But there are so many different products, services and things to learn about for your PC, what you really need is *one* event that brings everything together in one place in a well-organized format.

PC '83 does it all. It showcases all the PC-compatibles under one roof. It has a complete and comprehensive program of seminars, forums and product workshops. And it's produced with the quality and professionalism you've come to expect from Northeast Expositions.

So if you do anything with the IBM Personal Computer or if you're considering buying one, be sure to put PC '83 in your calendar right away.

Your PC will certainly appreciate it.

ADMISSION PRICES:

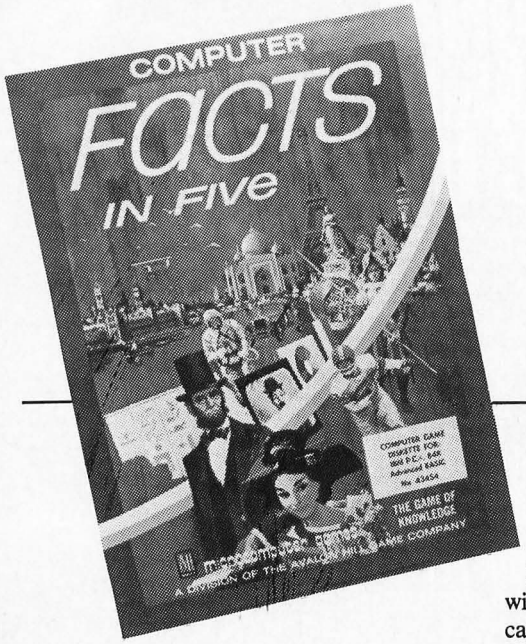
Exhibits Only Tickets—$10 Per Day. Three-Day Exhibits and Conference Ticket is $25.

ADDITIONAL INFORMATION

To receive additional information about attending or exhibiting at PC '83, including the Conference, Seminar, Workshop and Panel Discussions Program, or information on local hotels call 617-739-2000 or 800-841-7000 (Boston).

ANY QUESTIONS? CALL 800-841-7000 (BOSTON)

Produced by Northeast Expositions, nationwide producers of the National Computer Shows, Applefest, CP/M'83, and Softcon.
826 Boylston Street, Chestnut Hill,
Massachusetts 02167



FACTS IN FIVE™

Reviewed by Robert C. Gray

From Avalon Hill Game Company, 4517 Harford Rd., Baltimore, MD 21214. System requirements: 64K IBM PC, Advanced BASIC. Suggested retail price: $26.

Imagine playing "beat the clock" with your ten year old son and the new category is BMX stars. (You've already faltered on famous martial artists and the anatomical parts of the bee that he studied in science last week.) No, this is not a new form of adult torture. It's a fast-paced word identification game which pits your grasp of general facts against time, your opponent and the computer. Depending on who's choosing the categories, it can make you rack your brain or run for an encyclopedia.

The object of *Facts In Five* is to list as many right answers (up to five) within a given category as you can in 60 seconds. The game screen displays a subject list on the left and a five-by-five matrix of stars or "first letters" on the right. You have the option of choosing any answer

you can think of or choosing one that begins with the indicated letter. Depending on what game options you choose before playing, this can be easy or difficult.

Options range from the tame "normal mode," where you move through the categories at your own pace, according to your own choice, to "timed random mode," where the computer makes a random choice and you have only twelve seconds to answer. Between the two are three other variations: "sequential," where the first letters are hidden and revealed in sequence, "random," where the letters are hidden and revealed randomly, and "timed sequential," where you have only twelve seconds to give answers in sequence. You can see that in some of these options, you are playing more than your precocious opponent; you are also playing the computer.

The computer also generates a random list of topics whose subject matter varies in difficulty. Although many of the topics are suited for twelve and above, some would stump the average adult. You can gear the game to the player by using another feature called "wild card" play. Here, the player can choose his or her own categories. For younger children or competitive adults, this is where the fun lies. It's also an opportunity for teachers to provide a subject matter level suitable to their students.

Facts In Five has potential as a good educational game for preteens. Two individuals can play, or you can choose "party mode," where the computer controls the subject display, game variations and scoring, but the teams use game sheets to enter their answers. Teachers can adjust the subject matter categories and the number of alphabetic characters used as "first letters" to suit the age range and educational goal of the session. Yet students need not feel controlled by a computer. Validating right answers and awarding bonus points remains a hotly debated group decision.

Group play is definitely more fun than solitaire with *Facts In Five.* I also found wild card play to be more interest-sustaining in the home environment. I would suggest, however, that an auto-execute file be written to restart the game. At present, you must reboot the system to begin again and that takes time. When you need to retaliate with categories like Fortune 500 CEO's or parts of the garden tractor you took apart last week, you want to do it fast.

Hints & Enhancements

Applesoft Comma-tabbing Bug

If you have set the left margin of the screen with a POKE 32,n, there is a rather dangerous problem with comma tabbing (for example, PRINT X,Y,Z). The portion of Applesoft that handles the comma tabbing (starting at $DB03) assumes that the left margin (location 32 decimal) contains a zero. When it doesn't contain a zero, the values Applesoft stores in the horizontal-position byte (location 36 decimal) can place the cursor beyond the edge of the screen. This can cause printing of text outside your text window, but even worse, text will probably overwrite peripheral card scratchpad locations and trample on your BASIC program.

Kerry Shetline
SoftSide

Budget Program

You should place *Budget (SoftSide,* Issue 38), the second part of *Deluxe Personal Finance,* on the same disk as *Checking (SoftSide,* Issue 37) and its associated data files. If *Budget* does not find data files, an end of data error occurs. The best thing to do is to put all the *Personal Finance* files on an otherwise empty disk.

Atari's "Vanishing" RAM

I just got an Atari 800 with 48K. According to the manual, one K-byte should be 1024 bytes. This is my problem. When I ran the "user memory test," I get only 37,891.

I ran a test on each memory pack (16K) separately. They were all the same, with each having only 13,326 bytes. According to Atari's math, shouldn't it be 16,384?

The same thing happens with two packs (32K). I got only 29,710 bytes. Is this right? Should it be 32,768?

Can you please tell me which is right?

Dan MacKain
Northfield, NJ

Editor's reply: Your Atari's behavior is absolutely normal. Your math is correct, and each of your 16K RAM packs does contain 16,384 bytes. The computer, however, uses some of the RAM for the screen display, system pointers and variables, the 6502's stack, and other housekeeping functions. This means that BASIC doesn't have all the RAM to itself, and reports to you only the memory it can use. What's more, in 48K Ataris, the BASIC cartridge replaces 8K of RAM, which is completely inaccessible.

Atari *Solitaire*

We have enjoyed playing *Solitaire (SoftSide,* May 1982) many times since you published it. It is a very well designed program with excellent graphics.

However, the straight Klondike version seems to be a lot harder to beat than the version we have always played. In our version, you take three cards from the stock, and turn them face up on the waste pile (instead of one at a time). The use of the waste pile remains the same until the top card cannot be used, when you again take three cards from the stock. When the stock pile is depleted (fewer than three cards), the waste pile is turned over, and any cards in the waste pile are placed on top of these to make the new stock pile. You may repeat this process continually, until no cards are playable or you win the game.

The following lines implement this version.

```
105 IF IN>48 AND IN(7)<4 THEN 1800
110 IF IN>49 THEN 1650
120 FOR K=0 TO 2:OD(IN(7)+K)=D(IN+K):N
EXT K:IN=IN+3:IN(7)=IN(7)+2
125 X=X(7):Y=Y(13):NUM=OD(IN(7)):GOSUB
 70:GOSUB 20:IN(7)=IN(7)+1
1650 IN(7)=IN(7)-1:AVER=0:BVER=0:IF ON
=50 THEN BVER=D(51):AVER=D(50)
1660 IF IN=51 THEN BVER=D(51)
1670 FOR VV=0 TO IN(7):D(51-VV)=OD(IN(
7)-VV):NEXT VV
1680 IF BVER>0 THEN D(51-VV)=BVER
1690 IF AVER>0 THEN VV=VV+1:D(51-VV)=A
VER
1700 IN=51-VV:IN(7)=0:IN=IN+1:IF BVER>
0 THEN IN=IN-1:GOTO 120
1800 IF IN>51 THEN 1490
1810 OD(IN(7))=D(IN):IN=IN+1:GOTO 125
```

R. Martenis
Lake Orion, MI

Atari String Arrays

Simulating arrays of strings on the Atari can be difficult and troublesome. You must create an array string which has the number of array elements multiplied by the maximum length of any of the strings. For example: if you need five strings in an array, and the longest string is "DOPPLEGANGER" then dimension the array string to five times twelve, or 60. If the first element was the string "YAK," you have to pad the remaining nine bytes by filling the string with spaces (character 32).

When you print the string array element, the entire twelve characters must be printed, because the actual end of the string within the element is unknown. You could use a routine to scan backwards through the string, but this is time consuming.

The answer is to use a character other than a space. If the entire string were filled with character 254, then when it is printed, only the text appears. In addition, any text to the right of the printed array element will be "sucked-up" as is done in the screen editing mode with the DELETE key. For example, here is a string using spaces to pad the array:

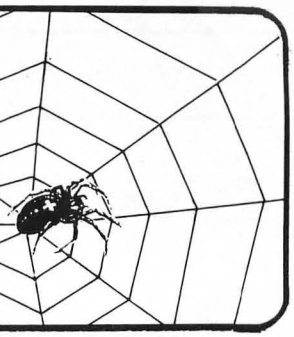
```
10 GRAPHICS 0
20 DIM A$(60)
30 A$=CHR$(32)
40 A$(60)=CHR$(32)
50 A$(2)=A$(1)
60 A$(1,12)="PHILTRUM"
70 A$(13,24)="ZWIEBACK"
80 A$(25,36)="MEATCAKE"
90 A$(37,48)="PROBOSCIS"
100 A$(49,60)="PHAGOCYTOSIS"
110 FOR X=0 TO 4
120 PRINT "THE ";
130 PRINT A$(X*12+1,X*12+12);
140 PRINT " REIGNS."
150 NEXT X:PRINT
160 PRINT "A$ = ";A$
```

Type in and run this program, then change the CHR$(32) in lines 30 and 40 to CHR$(254) and then re-run the program.

Rather useful...

Alan J. Zett
SoftSide

Bugs, Worms, and Other Undesirables



Apple Best of SoftSide *SWAT*

One line was omitted from the listing of *SWAT* in ***The Best of SoftSide.*** The result of this omission is that you must manually change the values of NU and B in line 60000 when modified parameters are called for. Adding the line below will allow you to modify these parameters dynamically. Please note the new *SWAT* table for *SWAT*.

```
60005  HOME : PRINT "USE MODIFIED
       PARAMETERS? ";: GET X$: PRINT
       X$: IF X$ = "Y" THEN  INPUT
       "NU= ";NU: INPUT "B= ";B
```

SWAT TABLE

LINES	SWAT CODE	LENGTH
60000 - 60100	IM	452
60110 - 60130	TG	179

Apple *Personal Finance*

The listings for the *Initializer* and *Checking* programs (Issue 37) are one character wider than normal. They are correct, but will not match what appears on your screen when you LIST the programs.

The *SWAT* table for the *Initializer* program is erroneous. The correct table appears below.

SWAT TABLE

For **APPLE® INITIALIZER**

LINES	SWAT CODE	LENGTH
10 - 110	OD	517
120 - 130	SO	150

TRS-80 *Puzzle Jumble*

This correction appeared in Issue 41, but appeared oddly because our typesetting equipment doesn't understand BASIC.

An error in *Puzzle Jumble* (Issue 34) caused the program to crash on 32K systems. To correct this, change the constant 192 to 191 in line 60000. The correct line appears in its correct entirety below.

```
60000 Z=0:FORX=1TO158:READY:Z=Z+Y:NEXT:IFZ<>15220THENCLS:PRINT"D
ATA BASE ERROR IN LINES 60060-60160, CHECK LISTING.":PRINT:LIST6
0060-60160ELSEY=86:X=255:POKE-1,0:IFPEEK(-1)<>0THENX=191:POKE-16
385,0:IFPEEK(-16385)<>0THENX=127
```

The Magic of INKEY$

The small program associated with the article, *The Magic of INKEY$* (Issue 41), has an error in line 1060. The third statement in that line was missing a space between the quotation marks. The correct line appears below.

```
1060 IF CB=CHR$(8) THEN CA=LEFT$(CA,LEN(CA)-1):
  PRINT CHR$(24);:
  PRINT " ";:
  PRINT CHR$(24);:
  GOTO 1020
```

To perform the initialization to which the article referred, add the following line:

```
10 DEFSTR C:DEFINT A
```

Atari Best of SoftSide *Titan*

Part II of *Titan* has two underlining errors. In lines 1020 and 1340, the spaces underlined should not be underlined, and therefore should be typed as normal spaces. This will not affect the operation of the program, but will cause your *SWAT* table to disagree with the published one. The SWAT table in ***The Best of SoftSide*** is correct.

Atari *Poker Squares*

The Atari SWAT table for the program *Poker Squares,* published in Issue #41 of ***SoftSide,*** failed to mention that it was generated with modified parameters. The correct table should be created with parameters of NU = 5, B = 300.

Atari Best of SoftSide *Defense*

On page 7, part of line 7050 is missing. The correct line reads:

```
7050 FOR I=1 TO 400:NEXT I:POP :GOSUB
11000:GOSUB 11100:GOTO 20
```

HOTLINE

Every Wednesday, from 7PM to 9PM Eastern Daylight Time, the SoftSide programmers answer your questions about SoftSide programs. Call (603) 673-0585.

The Hotline can:

- Tell you the fix for any known program bugs, or mail you the long ones.
- Explain program instructions.
- Check out bugs you may have found.

The Hotline cannot:

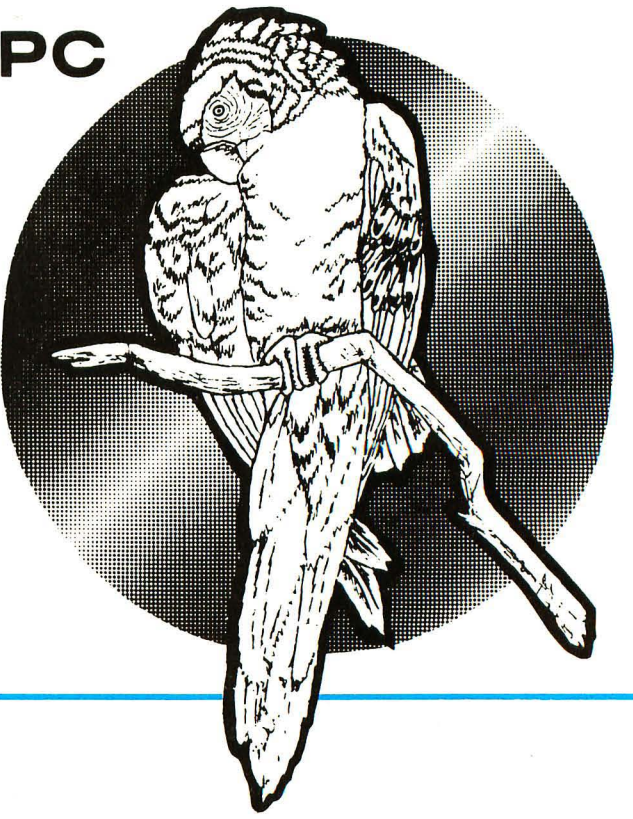
- Answer subscription or editorial questions.
- Give programming advice.
- Accept collect calls.

When you call, follow these simple rules:

1. Write it down and if possible, sit next to your computer when you call.
2. Be specific. It betters our chance to zero in on your problem quickly. We'll want to know what program and what computer.
3. Be brief. There are 50,000 of you and five programmers...
4. **SWAT** the program. Most "bugs" are hidden typos. **SWAT** will find them and may save you a call.

Remember: The **Hotline** will answer the short questions and deal with long questions in writing.
(Programmers will answer software questions only during **Hotline** hours.)

Give Voice To Your PC

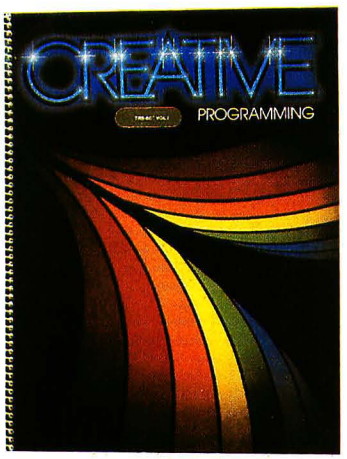


PC Now, with the *PC-Parrot,* IBM® PC owners can enhance their BASIC programs with audible speech and other sound effects, without any special hardware. A unique set of machine language routines utilizes the cassette interface on the IBM PC so that the user can attach numerous possible sources of sound to the computer, which records and stores these sounds digitally. *PC-Parrot* comes complete with all the utilities necessary to set up and test the cassette interface for a good connection to the source of sound. The user can record new sounds or select previously recorded sounds from a library file. If you don't wish to take advantage of the recording capabilties of *PC-Parrot,* an extensive set of vocabulary and sound files is included with the package.

PC-Parrot requires 64K RAM, one disk drive and a monitor. It is available for $39.95 from Dragon Data Systems, 1068 Homer Street, Suite #110, Vancouver, British Columbia, CANADA V6B 4W9 (604)255-0584

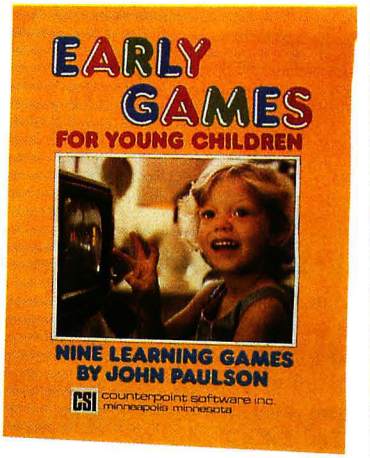
Learn Programming Painlessly

Creative Programming, a three volume self-teaching course, is a practical guide to programming, suitable for children as well as adults. The step-by-step lessons, examples and reviews allow you to learn BASIC in your own home, at your own pace, without any other instruction or tutor. From the first few pages of Volume I, where you are introduced to the TRS-80® Model III, to the last pages of Volume II, where you are actually creating your own BASIC programs, the course presents information clearly and with a sense of humor. Each volume contains a glossary, worksheets and suggested projects to attempt on your own. The third volume in the series will be available shortly.



The course is available from Creative Programming, Inc., 750 W. Algonquin Road, Arlington Heights, IL 60005, or from a computer store in your area, for a price of $7.95 per volume. School discounts are available.

Fun And Games



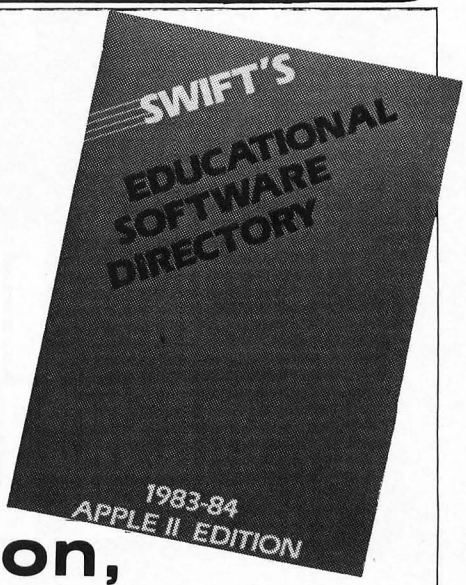
 PC Even your toddler can interact with your computer and learn at the same time, with the introduction of *Early Games For Young Children* — nine learning games that are fun! The Picture Menu allows children to select a game, play it, and go to a different game without adult assistance. The games teach counting skills, basic math skills, drawing and elementary language skills. They stimulate problem solving while fostering individual creativity.

The Early Games package is available directly from Counterpoint Software, Inc., Suite 140, Shelard Plaza North, Minneapolis, MN 55426 (800) 328-1223 for a retail price of $18. Quantity discounts are available.

Protect Your Modem

The *KLEEN LINE* Security System protects your modem by suppressing damaging telephone line spikes caused by lightning, spherics or phone office switch gear using a modern two stage semi-conductor and gas discharge tube suppression technique. It also employs an isolated ground to protect equipment from damaging lightning discharge current. Standard modular four pin telephone connectors provide simple, trouble-free hook-up.

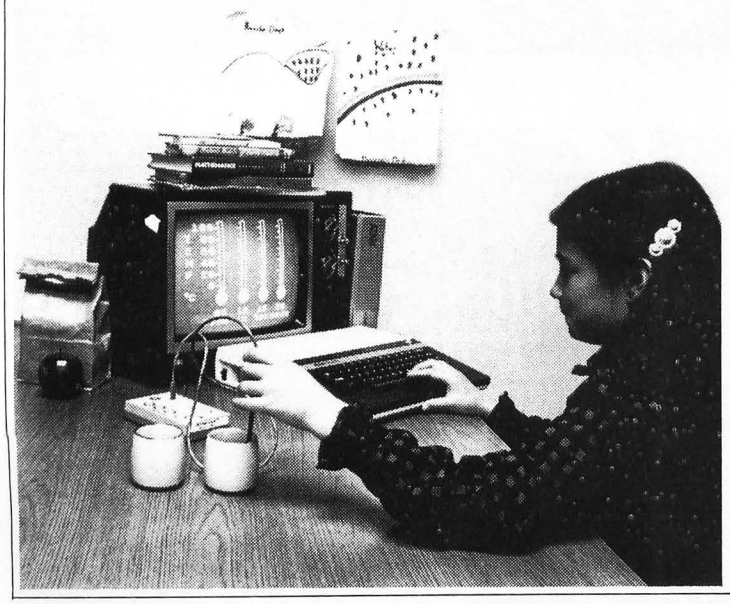
The *KLEEN LINE* (Model PDS-11) is available from Electronic Specialists, Inc., 171 South Main Street, P.O. Box 389, Natick, MA 01760 (617) 655-1532 for $56.95.



Attention, Educators!

The 1983-84 *Swift's Educational Software Directory* — Apple® II Edition boasts a 90 percent increase in software listings over the 1982 edition. The directory contains detailed information on educational software and is organized by subject area, — Foreign Language, Language Arts, Mathematics and so on. Each subject has its own subheadings and table of contents. There is a description of each software package, information on the software company, its backup and multiple copy policies. A Review Sources section contains concise software reviews.

Swift's Educational Software Directory is available directly from Sterling Swift Publishing Company, 7901 South IH-35, Austin, TX 78744 (512) 282-6840. The retail price is $18.95.



Scientific Software Introduced

With the new *Atari® Lab Modules,* you can perform exciting experiments in physics, chemistry, biology, psychology, geology, health and many other fields of science. Each module contains an instruction/experiment booklet, exclusive software and special measurement equipment. This equipment connects to the computer through the *Atarilab Interface* which hooks into the computer's joystick port. This allows interaction between the physical environment and the computer.

The *Starter Atarilab* consists of the *Atarilab Interface* and the *Temperature Module* for a suggested list price of $89.95. Other modules will be available in 1984.

DON'T MISS APPLEFEST FOR 1983 THE WORLD'S LARGEST EXPOSITION EXCLUSIVELY FOR APPLE OWNERS

The show features hundreds of exhibits of the newest, state-of-the-art products for the Apple. You can see and try out software for every conceivable application—from arcade games to investment programs, music to machine language, teaching systems to accounting packages, word processors to graphics processors. You can sample hundreds of different peripherals, including printers, hard disks, modems, memory cards, video displays and synthesizers, plus accessories, publications and invaluable support services.

Applefest is **the** place to view the most technologically advanced products for the Apple.

At Applefest you can try out and compare hundreds of products in an exciting, information-filled environment. You can learn more in two days than you could in months of visiting computer stores and reading trade journals.

And, best of all, everything on display at Applefest is for sale at special show prices, so you can save hundreds—even thousands—of dollars by making your purchases at the show.

This year a whole new conference program is being introduced to Applefests nationwide. The program, presented by Apple experts and leaders in the Apple field, will show you how to squeeze absolutely the most power, versatility and usefulness out of your Apple.

SEMINARS offer special insights and updates; tips and how-to's; ideas and opportunities for the entrepreneur; reports on the best and the brightest, the latest and the greatest, on topics of interest to every Apple user. TUTORIALS are intensive, easy-to-understand sessions specially designed for both the beginning Apple user and the Apple veteran looking to learn a new technique. APPLICATION WORKSHOPS will introduce you to a wide array of practical Apple applications for your home, school, and business. They are conducted in non-technical language at an introductory level, and will teach you the ins and outs of a specific application: how it works, what it does, and what are some of the popular products available for it. ADVANCED USER WORKSHOPS are designed for the hardcore Apple addict: the more advanced, technically knowledgeable Apple users who want to explore all the hidden capabilities in their systems. SOFTWARE and HARDWARE SPOTLIGHTS are designed to answer a problem that confounds practically all users: how to decide which packages are right for you. Hundreds of hardware and software packages will be discussed and demonstrated, and experts will be on hand to answer your questions and help you understand the features. In many cases you'll even be able to talk with the authors and designers directly!

No matter what you do (or want to do) with your Apple, the Applefest Conference Program will help you do it better.

SPECIAL HAPPENINGS at Applefest include: an **Open Forum with Steve Wozniak,** one of the founders of Apple Computer, Inc., and the creator of the Apple II; a Panel Discussion, with some of the weightiest opinion-makers in the software field, on **"The Great Software Piracy Debate"; Hands-On AppleCenters** for you to learn about and try out the Apple, even if you've never touched a computer before; an **Apple Jam Session,** designed just to show how Apples and synthesizers make beautiful music together; an **Electronic Shopper's Guide** to help you quickly find who is selling the products you want; plus more!!!!

Admission Prices: Exhibits Only Tickets—$10 Per Day. Three-Day Exhibits and Conference Ticket is $25.

Additional Information
To receive more information about attending or exhibiting at Applefest, or information on local hotels call 617-739-2000 or 800-841-7000 (Boston).

Applefest/San Francisco:
Friday-Sunday, October 28-30, 1983
Moscone Center
10:30AM-5:30PM daily

Any Questions? Call 800-841-7000 (Boston)

Market Side

Advertiser's Index

Advertising Representatives

COMING NEXT ISSUE: *SoftSide Issue #44*

ADVENTURE & SIMULATION

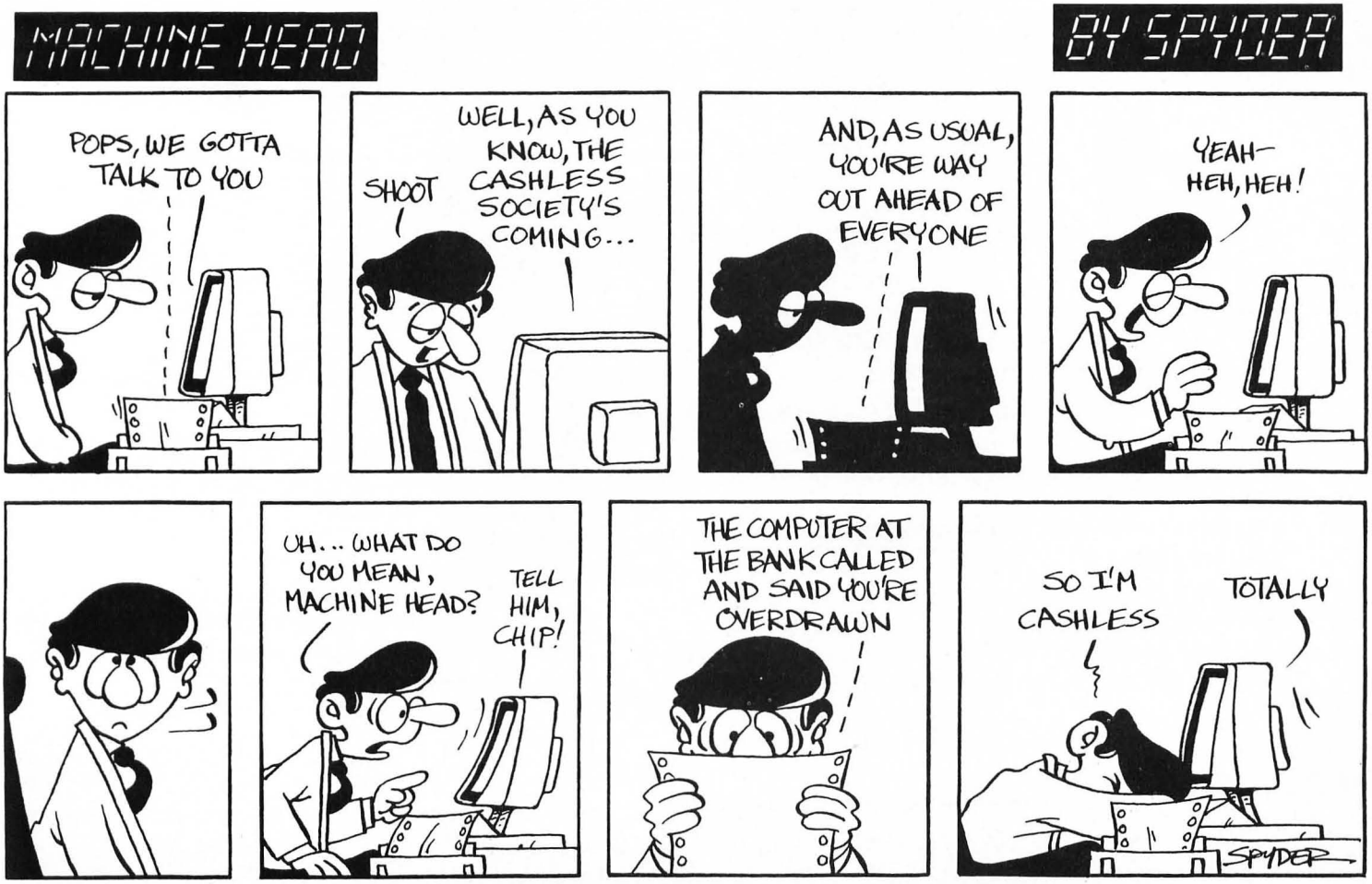
The Front Runner in Issue #44 puts you in the Producer's Office of a Broadway show.. Your first job is to raise all the money. After that, you need to hire a director, audition actors and musicians, hire stagehands and a technical crew, build sets, make costumes and attend to the thousands of crucial details that make a successful show. **Broadway** is your ticket to fun on the Great White Way. Whether it's rags or riches depends on your skill in managing the production.

How do you get an adventure player's attention and motivate him to continue playing? **Peter J. Favaro** talks about adventure game design from a psychologist's point of view, and offers some special insights into the ways you can create a game structure that will captivate and entertain.

How computers serve society, and the dangers they create, are the topics covered in ***SoftSide's*** review of **WarGames**, the "real-life" adventure game we all hope never happens.

M.M. McClung interviews two of the most successful adventure creators, **Marc Blank** and **Mike Berlyn** of Infocom. They discuss how they create imaginary worlds to give you experiences you can never have in this one. Be sure to read this fascinating conversation with these free-flowing, creative spirits.

Plus — a VisiCalc® game, a system for success on the commodities market, futurephones, and reviews (including **Prisoner II, Suspended, Shamus Case 2, Fredericksburg),** and more.



Presenting The Best of SoftSide™

For the past four years **SoftSide** Magazine has been bringing you the finest in microcomputer software. Now **SoftSide** has gathered the best Games, Adventures, Simulations, Utility Programs, and other software together into **The Best of SoftSide.**

Three books, one each for the Apple, Atari, and TRS-80 microcomputers are packed with BASIC code from the very best out of the hundreds of programs we have published. Each program is complete with valuable improvements and enhancements offered by hundreds of users! **SoftSide's** official debugging utility, S.W.A.T. (Strategic Weapon Against Typos), is included for each program. Compact, useful instructions and programming footnotes for the curious and the enhancer add dimension to the usability of the software.

Also On Disk

Not the typing type? — Then the disk version of ***The Best of SoftSide*** is just what you're looking for. All of the programs from the book, already typed in and stored on disk for your convenience — just put the disk in your disk drive and away you go. Each version has at least one-full disk side * of programs ready for you to use and enjoy.

Learn how to make your computer do what you want it to. Many of our readers credit **SoftSide** for improving their ability to program.

Experience the convenience of **The Best of SoftSide's** "steno" format. No more taking an awkward book to your computer to type in a program or read documentation. The book is scaled to coordinate with most of the other manuals in your software collection.

Explore our exciting games, practical applications, and helpful utilities. Fast paced Arcade Games... Database Managers...Word Processors...you'll get them all for only a few cents each.

Find your typing mistakes easily with **SWAT (Strategic Weapon Against Typos).** This amazing little utility will tell you where your mistakes are to within a couple of lines.

Play exciting games on your computer. **The Best of SoftSide** will take you...to the stars...to a casino in Vegas...even into the fantasy worlds of your imagination.

Create imaginative programs of your own with **The Best of SoftSide's** utilities and subroutines. We'll teach you how to use some powerful software tools,step-by-step.

The Best of SoftSide is conveniently steno bound and printed on non-reflective paper for ease of use when typing the code into your computer.

*Number of disk sides varies due to varying disk storage efficiency on different computers.

Bringing You the Finest in Printed Software and Documentation for the Apple, Atari, and TRS-80 (Model I or III) Computers.

Just look at the programs to be found in the three computer-specific editions of **The Best of SoftSide:**

APPLE EDITION

ARCADE GAMES:
Galaxia
Quest
Space Rescue
Minigolf
BOARD GAME:
Battlefield
GAMES OF CHANCE:
Solitaire
Gambler
ADVENTURE:
Operation Sabotage
Escape From
The Dungeon of the Gods
SIMULATIONS:
Arena of Octos
Leyte
Titan
WORD GAME:
Word Search Puzzle Generator
MUSIC:
Flight of the Bumblebee
Melody Dice
GRAPHICS UTILITY:
Magical Shape Machine
PRACTICAL APPLICATIONS:
Database
Microtext 1.2
S.W.A.T.

ATARI EDITION

ARCADE GAMES:
Defense
Quest
Space Rescue
Maze Search
Minigolf
BOARD GAMES:
Flip-It
Battlefield
GAMES OF CHANCE:
Solitaire
Gambler
ADVENTURE:
Operation Sabotage
SIMULATIONS:
Leyte
Titan
WORD GAME:
Word Search Puzzle Generator
MUSIC:
Fugue
Flight of the Bumblebee
Melody Dice
Music Programmer
GRAPHICS UTILITY:
Character Generator
PRACTICAL APPLICATIONS:
Random Access Database
Microtext 1.2 Word Processor
S.W.A.T.

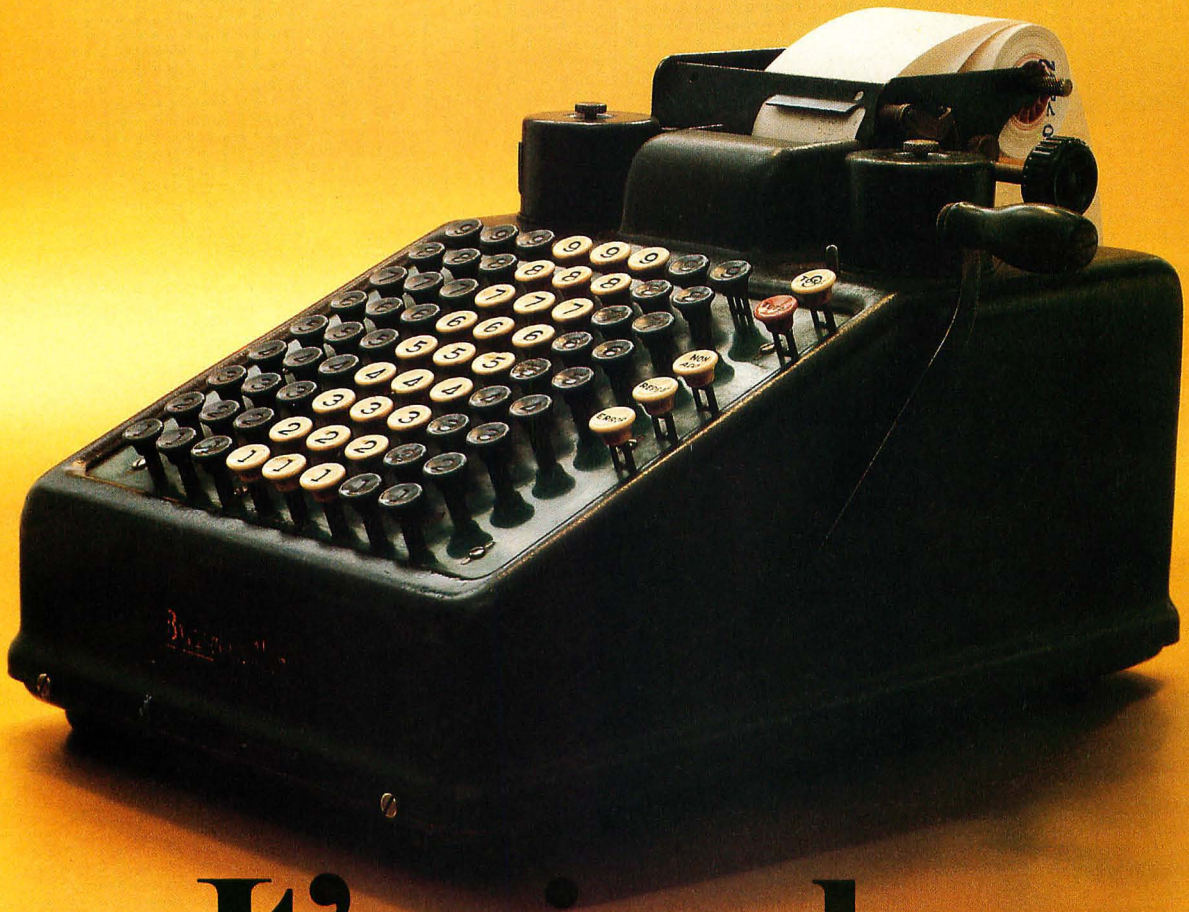
TRS-80 EDITION

ARCADE GAMES:
Minigolf
Space Rescue
Maze Sweep
Quest
BOARD GAMES:
Flip-It
Battlefield
GAMES OF CHANCE:
Solitaire
Gambler
ADVENTURE:
Operation Sabotage
SIMULATIONS:
Broadway
Leyte
Titan
WORD GAME:
Word Search Puzzle Generator
GRAPHICS UTILITY:
Compu-Sketch
PRACTICAL APPLICATIONS:
Random Access Database
Microtext 1.2
S.W.A.T.

A true software bargain, **The Best of SoftSide** may be ordered at only $19.95 per book, or $68.95 per book and disk combination. If you already ordered the book and would like to add the disk, the additional cost is only $49.00. Important: Be sure to specify computer make and model when ordering! Use the bind-in card in this issue to order today! Mail it with your payment or credit card number to **SoftSide Publications Inc., 10 Northern Blvd., Amherst, NH 03031**

The Best of SoftSide is also available at your SoftSide Dealer

Data work.



It's simple.

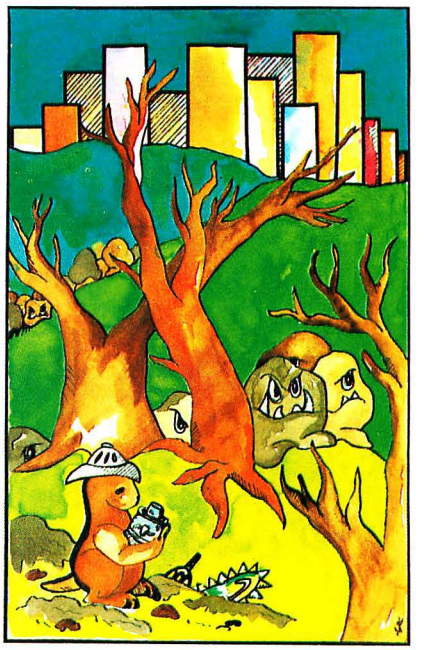
Now **Data Perfect**™ from LJK helps you organize your files and numbers like the adding machine first helped your grandfather. It keeps tax records. Lets you change files easily. List and edit addresses. Compares stocks. Stores expenses. It even calculates. Used with Letter Perfect™, you can even make custom mailings. Simply.

When it comes to practical software for Ataris, Apples and look alikes, Data Perfect is simple to learn. And hard to beat. Ask your dealer for a demonstration, or write LJK for more information.

LJK, INC. • 7852 Big Bend Blvd., St. Louis, MO 63119 • (314) 962-1855

METROSOFT

ATTACK OF THE KILLER POTATOES



Guess who's coming to dinner?

Mysterious radiation has mutated everyday vegetables into ravenous single-minded killers — their mission to destroy the human race. Their only hope, you a heroic ground hog. Armed only with a food processor, a buzz saw and a potatoe masher can you save the defenseless humans before they are carried below the earth and transformed into lethal mutants?

For Apple® 2+/2e or equivalent with applesoft in rom.

$29.95

TAX ATTACK

Tired of Paying Taxes?

Have you ever watched your hard earned money get eaten out of your hands by bureaucratic tax collectors? Solution: put your money where their mouth is. A little cash under the table is all it takes to get these "conscientious" tax collectors to look the other way. This lighting fast maze game employs high quality speech synthesis without the necessity of an extra hardware attachment.

An addictive game for the Atari® 400/800 with 48K disk.

$29.95

METROSOFT is painting the town.

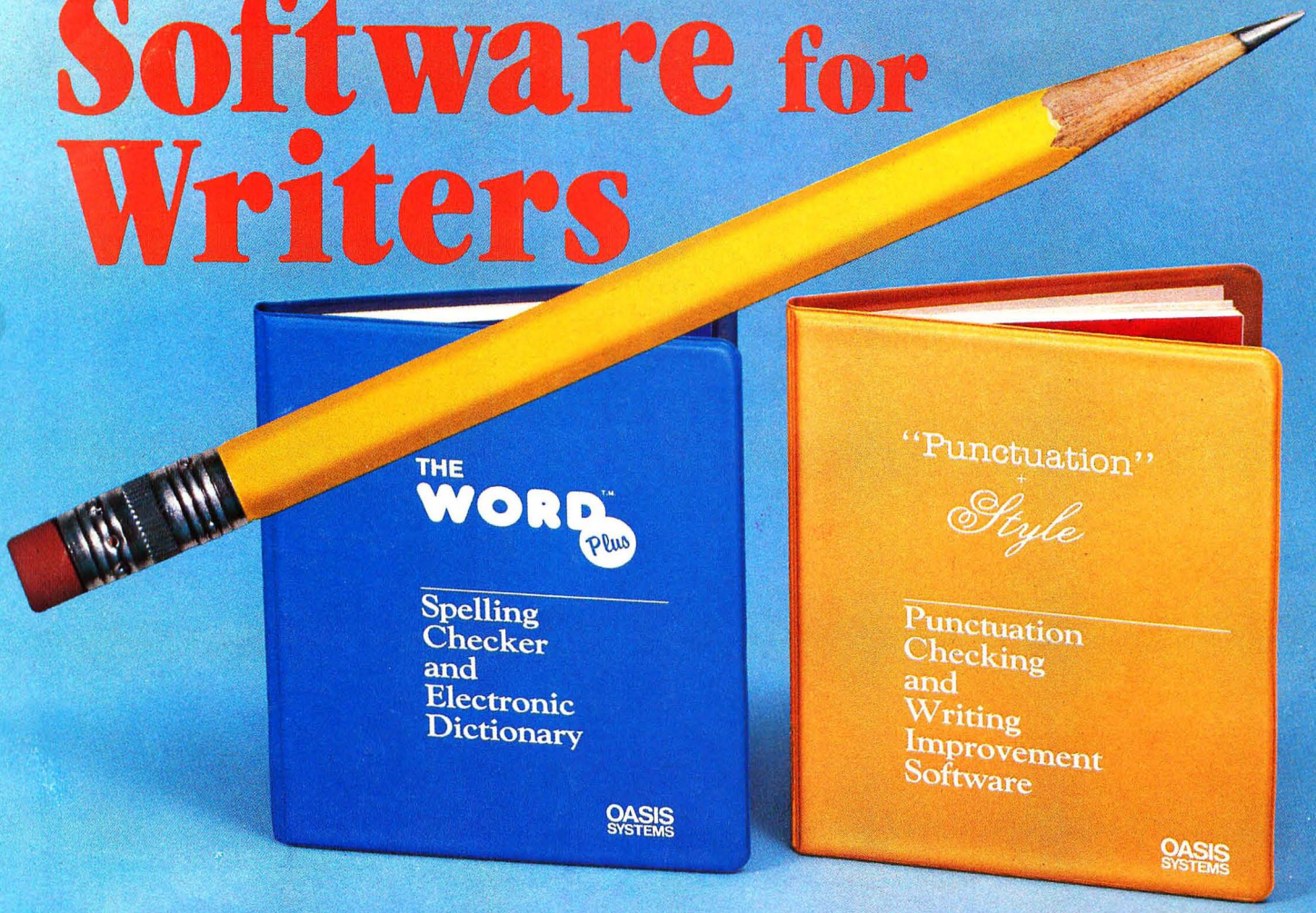
If not available at your local dealer, send check or money order, plus $2.00 shipping and handling to:

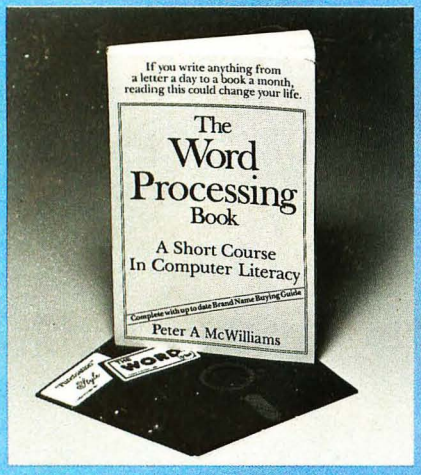
METROSOFT, INC.
1 East 28th Street
New York, New York 10016
or call:
212-889-4931

VISA and MASTER CARD ACCEPTED

Apple® and Atari® are registered trademarks of Apple Computer Company and Warner Communications respectively.
Graphics Created With Penguin Softwares Graphics Magician.

Software for Writers





"Oasis Systems' software — Unquestionably the best". . .

Peter McWilliams, author of the #1 best-selling book on word processing.

The WORD Plus is the standard by which other spelling checkers are measured. Here's why:

- *Real* 45,000 word dictionary.
- Shows errors "in-context."
- Interactive word look-up finds correct spelling for you and corrects at the push of a button.
- Hyphenates words automatically.
- Solves crosswords, puzzles, and anagrams.
- Works with almost any CP/M®, CP/M-86® or MS/DOS (1.0, 1.1 and 2.0) compatible word processing program (WordStar, Magic Wand (PeachText), Spellbinder, Perfect Writer, Select, Final Word, Volks-writer, . . . and more!).

Punctuation & Style takes the worry out of writing by automatically catching dozens of different punctuation errors, both common and obscure. In addition, P&S catches unpaired format commands (underline, boldface, etc.), doubled words, and more.

P&S gives you a "critique" of your writing, suggesting alternatives for commonly misused or over-worked phrases. It also shows where active voice can replace passive voice to add clarity and precision.

Punctuation & Style is the perfect companion to The WORD Plus. It works easily with most CP/M word processors. (Available soon for CP/M-86 and MS/DOS.)

Call or write for complete information: 619-222-1153

OASIS SYSTEMS
2765 Reynard Way
San Diego, CA 92103

Dealers contact:
SOFTWARE DISTRIBUTORS
1-800-252-4024 (in California)
1-800-421-0814 (outside California)