creative computing

Special 12-page April Fools Parody

the #1 magazine of computer applications and software

In-depth Evaluations of 25 new products

Computers in Education: 4 How-to Tutorials

Columns: IBM, Apple, Atari, TRS-80, I/O

3-D Plotting

Challenging Strategy Game

Our Robot Friends

in this issue...

evaluations & profiles

articles

applications & software

departments

the cover

The cover design is by Paul Stinson who is represented by Creative Associates, 17623 Scherzinger Lane, Canyon Country, CA 91351, (805)251-1178.

attention authors

Creative Computing will not be responsible for the return of unsolicited manuscripts, cassettes, floppy disks, program listings, etc. not submitted with a self-addressed, stamped envelope.

microform

Creative Computing is available on permanent record microfilm. For complete information contact University microfilms International, Dept. F.A., 300 North Zeeb Road, Ann Arbor, MI 48106 or 18 Bedford Road, London WC1R 4EJ, England.

David H. Ahl

FROM 8 TO 16 BITS--THE RACE IS UNDERWAY

Several years ago it was all the rage to forecast when each computer maker was going to graduate from an 8-bit cpu to a 16-bit cpu. However, it turned out that consumers really didn't need 16-bits and computer makers were really quite content to stay with 8-bit machines. However the entry of IBM has changed that. Users are salivating at the thought of sinking their teeth into 128K or even 256K of on-line memory. By now, everyone using VisiCalc has run into at least one problem that just won't fit.

Consequently, the personal computer manufacturers are gearing up to unleash a series of 16-bit machines in the near future. Needless to say, the minute the first one is announced, everyone else will follow suit ready or not.

Apple reportedly has two 16-bit projects in the works: SARAH, a cheap, low-end machine and LISA a high-end 16-bit machine with superb graphics based on the Motorola 68000 chip. Radio Shack is also rumored to be working on a TRS-80 based on the 68000. Heath/Zenith is following the way of IBM and basing their 16-bit machine on the Intel 8086 mpu.

Not only that, but at least one peripheral manufacturer is working on a plug-in board for the Apple, incorporating an 8088 mpu which gives the Apple the capability of running Atari player-missile graphics as well as IBM and other 8086 and 8088 software on the Apple II.

POSSIBLE ANSWER FOR THE VIDEODISC SOFTWARE DILEMMA

In the January issue, I mentioned that the Screen Actors Guild was not ecstatic about the idea of using videodiscs of movies as the basis of adventure games and other computer software. Their view is that a videodisc of a movie should be viewed from start to finish and not used as the basis for any kind of interaction or certainly not a computer game. This view clearly puts a crimp on an exciting new idea.

Now, however, there may be an answer. New Media Graphics Corporation of Cambridge, MA is offering a low-cost opportunity to create material on laser scan videodiscs on a shared basis. For $500 any company can purchase 9,000 frames (5 minutes) on a shared basis. Each participant will receive five copies of the disk with additional copies available for $25 each.

Our Rollercoaster Adventure game (published in the January issue) uses far fewer frames than this. Hence, one can imagine several software companies getting together or just one putting on enough material for several different computer adventures on a shared disk. Players could then purchase this videodisc along with three or four floppy disks containing the computer programs. Seems like it might be the ideal solution. New Media Graphics is located at 139 Main St., Cambridge, MA 02142.

MORE SINCLAIR COMPUTERS THAN ANY OTHER!

Nigel Searle tells us that in the first ten months of sale, more than 250,000 ZX-81s have been delivered. As a result, he claims, "there are now more Sinclair computers in use than any other computer in the world."

I recall in 1972 at Digital Equipment Corporation, we were saying the same thing about the PDP-8 in terms of total number of units. However it took ten long years for DEC to close the gap significantly on IBM and the "seven dwarfs" from a dollar standpoint.

Nevertheless, 250,000 units is impressive. Even more impressive is the current 50,000 unit per month manufacturing rate of the ZX-81. Made in Scotland, roughly 60% of the units are exported to eighteen other countries. US sales are currently running between 15,000 and 20,000 units per month.

DOWN AND OUT

Amid growing competition in the personal computer field, some manufacturers are quietly bowing out while others are lowering prices to compete more effectively. Atari has repackaged and repriced the Model 800 home computer. Pointing to cost savings as a result of increased volume, Atari reduced the base price from $1080 to $899 which follows on the heels of a similar price reduction made in the Model 400 in 1981.

Hewlett Packard has also reduced the price of their HP-85 personal computer by 15% from $3250 to $2750. They also credit manufacturing efficiency and productivity. HP also announced a price decrease in their 16K memory module from $295 to $195. Concurrently HP announced the discontinuance of the HP-83, a stripped down version of the 85 without printer or tape drive.

Radio Shack has also tested several price reductions in their sales flyers issued in the last quarter of 1981. While these alleged "sales" were for one month only, we note that several of them have been continued and appear to be the current regular price of the product. Most notably, the Pocket Computer has dropped in price substantially, but I hesitate to name a figure because by the time this gets into print it may well be out of date.

Like the HP-83, another machine that failed to find a continuing niche in the market is the APF Imagination Machine. As a result, APF will withdraw it from sale when current stocks are exhausted.

8

Phi Beta Apple.

Whatever you profess, from Accounting 101 to Zeitgeist 595, an Apple can help you get more out of the subject.

That's what Professor Andrew Gordon learned when he enrolled Apple Personal Computers in his courses at Northwestern University.

In his Sociology classes and his Urban Affairs research with community groups, students use Apple's

color graphics to map out and analyze neighborhood problems ranging from snarled traffic to arson to building decay.

In his Computer Programming classes, students use Apples to learn programming in BASIC and Pascal and to access data on the University's mainframe computer. (Apples are also fluent in FORTRAN, COBOL and PILOT.)

And Professor Gordon and his students aren't alone. There are over 90 Apples at Northwestern. In Psychology. Speech. Journalism. Medicine. Business. Engineering. They're cropping up all over campus.

And Northwestern isn't alone. Apple is the single leading system in education at all levels.

There are thousands of ready-to-use programs for Apples. Including sophisticated software for mathematics, physical sciences, lab work, languages, music—even programs that make it easy to develop your own computer-aided courses.

And no other system is backed by as many dealers offering complete technical and service support.

Summon one to your office.

And get educated.

The personal computer. **apple**

Corrections

There are two corrections for "Big Numbers and Small Computers" from the January 1982 issue. On page 138, the RETURN statement at the end of line 25120 should have come at the end of line 25200.

To avoid having to use the "C" command to clear the X register, add the following to the end of line 5000:

:IF VAL(X$)=0 GOTO 8000

A.I./Robotics Group

If you're interested in an artificial intelligence or robotics group contact: Greg Homatas, 229 St. Johns Pl., Brooklyn, NY 11217 or Abby Gelles, 185 West Houston, New York, NY 10014.

Apple Announces The Family System

Apple Computer Inc. has introduced a new personal computer merchandising program designed to meet the needs of the entire family. The Apple Family System
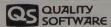
Using the Apple Family System, children learn the value of recipe conversion at an early age.

combines hardware, software and manuals into a package geared for the first-time computer user.

The Family System consists of the following components: a 48K Apple II Plus computer, a disk drive with controller card, hand controllers, RF modulator, Apple Writer, Personal Filing System, Personal Finance Manager, Typing Tutor, Apple Invaders, Olympic Decathlon, Apple Adventure and a keyboard tutorial, which acquaints the user with the computer and its use. The system also contains an educational software directory, an owner's guide, and the complete library of Apple II manuals.

The suggested retail price of the Family System is $2495.

On the Blink

Dear Editor:

Thank you for the "Blinking Cursor" article in the December, 1981 issue. I found the program works with Disk Basic by changing the jump address from 0072H to 0A9A [Listing 4, Line #250, location FFA0-FFA1].

After executing the Load and Go module from DOS the cursor may be retained in Disk Basic by protecting high memory from 65423. Thank you once again for your informative magazine.

Noe Oritz
451 Graves Rd.
Oxford, PA 19363

The Class Menagerie

Dear Editor:

Larry Noonan's excellent article on the use of simulations in the classroom (September issue) proves once again that computers can be used to provide much more than drill and rote learning. It will help those of us who have to contend with the skeptics.

Thanks to him for sharing his creative work and to you for including articles on education in your publication. They're hard to find, and very few are as good as Noonan's.

Leona Billings
Assoc. Prof. of Philosophy
Los Angeles Pierce College

I'll Trade You My Peanut Butter and Jelly for Your Super Invasion

Dear Editor:

The high school where I teach recently became the proud owner of six Apple II computers, and I found myself with a growing number of 5 1/4" floppy disks to tote around and to store. A quick investigation of the market showed that unless one was willing to pay $20 to $30, or use an old shoebox, one was always carrying little boxes of disks around.

While these are light, and inexpensive, they do have a tendency to open at the worst possible moments, and it is difficult to search through them when they are filled with more than three disks.

I would like to share my solution to this problem with the readers of *Creative Computing*. I bought a cheap lunch box (the vertical kind with the latches on the front), put in my disks, and padded the excess space with the little blue foam squares that form part of the packing for the Apples.

I now have a disk carrier with a sturdy, convenient handle, and it is easy to flip through the 45 disks it can hold. It has room to spare for assorted cables or a joystick, and can be bought color-coded for easy identification. The total cost for this marvelous invention was $3.98.

I enjoy reading your magazine, and I hope you can continue to present such high quality material in such an appealing format.

Leslie Eiser
7 Breton Woods
Beaconsfield
Quebec H9W 5A6
Canada

Carbon Copy

David Lubar

Introduction to Organic Chemistry is possibly the most ambitious series of programs yet written for a small computer. The set of seven disks contains tutorial material, demonstrations, and experiments that go all the way from recognition of simple saturated carbon chains up to introductory spectroscopy and reactions of carboxylic acids.

The bulk of the material is tutorial, with frequent questions occurring throughout. The information is presented in text on the hi-res screen, with double-size letters for chemical formulas. Some new concepts are first introduced as part of a question. For instance, in one question, a linear chain with one branch is shown, and the student is asked the length of the chain. If the user determines the length by counting the carbons in the linear portion, he will be wrong, since the longest chain follows one of the branches. After the question has been answered, the program explains this. By presenting such concepts in this manner, the student is unlikely to forget what he has learned.

One weak point is that the full potential of CAI has not been tapped in these programs. An incorrect answer will sometimes result in a hint, but often just causes the program to give the correct answer. A bit of branching, or some messages telling the student to go back to a certain segment, would have increased the strength of these programs. The student can obtain help with a question by typing control-A, but this help is generally the same response given to an incorrect answer.

An example from the Multistep Synthesis of Aromatics section. In this case, the selected reagent produced three isomers, from which the student must choose. Though the selected reagent was incorrect in this example, the program still carries out the reaction, allowing the student to experiment.

The demonstrations are animated and interesting. One sequence shows the importance of heat in a reaction by having one molecule ram another at various speeds. Finally, when enough energy is involved, the molecules bond, kicking free a byproduct. The graphics are put to good use demonstrating such concepts as rotation of molecules. This aspect of the package definitely makes use of the powers of the computer, and rises above being merely an electronic textbook.

The experiments are the highlight of the package. In one, the student is given a target aromatic and three possible starting choices. By choosing the proper reagents, he has to synthesize the target. At each step, the resulting compound is displayed. If the reagent is not sufficiently reactive, a message is displayed. Messages are also used when the computer is not familiar with a reaction, or if the reaction is not possible for other reasons. Not only does this make full use of the computer, it also provides a safe and inexpensive way to experiment with synthesis. Considering the toxic or carcinogenic nature of many organic chemicals, this is definitely a plus. In another set of experiments, the student is allowed to give different temperatures or concentrations for a reaction, and then see a graph of the results. This is followed by presentation of the appropriate formula for determining the result.

The package has many positive features. The input routines are good. After asking how many hydrogens can bond to a carbon, the program will accept either "four," or "4." While this might seem like a minor matter, it shows that the programmer has some regard for the user. As mentioned, the experiments and demonstrations are very good. The graphics are clear and clean, and the text is readable.

On the negative side, the program is very linear, putting all effort in presenting the material, and no effort in recognizing problem areas or helping a student who is having difficulty. Also, some concepts are introduced without explanation. In the section on spectroscopy, C-H stretch is shown on a spectrum, but not explained. The user who wants to know more won't get help from the program. For this reason, the program would not be well suited to home study unless the user had other resources available to fill in the gaps.

Based on the large amount of material provided in this package, and the high quality of the lab programs, *Introduction to Organic Chemistry* would be an excellent purchase for any college library or chemistry department, and would also be of value for any high school that needs resources for advanced students. □

Castle Wolfenstein
by MUSE

THE MOST INTERACTIVE GAME EVER!

The world's most interactive adventure game. It's date-a, element marginal fantasy and real-time arcade elements.

"A unique deal in programming." —New York Magazine

For the Apple II or II Plus, 48K $29.95
By Silas Warner

™MUSE

347 N. Charles Street
Baltimore, MD 21201 (301) 659-7212

Physics Simulation

Financial Plotting

Text Processing

Scientific Formula

Special Needs Education

APL Programming

Statistical Table

Statistical Graph

Wait'll you see the rest of the demo.

Program Documentation

CAD/CAM

There has never been a terminal like Digital's new GIGI™ terminal.

It's an intelligent terminal with exciting text processing functions. A graphics terminal with impressive resolution. And it's also a conventional interactive terminal. With applications software that makes all these capabilities easy to use, all at the same time.

So you no longer have to adapt your application to the limitations of your terminal.

Digital's new GIGI terminal. Originally designed for education, it's now available to anybody.

Anybody who needs the world's most versatile terminal.

For a demo, call your local Digital sales office, or send your business card to:
Digital Equipment Corporation, Media Response Manager, Dept. CC-4-82, 129 Parker St., PK3-2/M94, Maynard, MA 01754.

CAI

Mapping

Data Entry

Multiple Languages

d|i|g|i|t|a|l

We change the way the world thinks.

CIRCLE 276 ON READER SERVICE CARD

Problem Solving Programs for Older Students

Dan Isaacson

Whenever I go into a bookstore I always look for a new book by an author I have read before and liked. I am seldom disappointed. I'm finding that idea carries over into quality software, too. I liked Quality Educational Designs' earlier product, *Math Problems*, and, not surprisingly, I like their current effort, *Arith-Magic*.

There are three problem-solving programs in this set: Diffy, Tripuz, and Magic Squares. All give practice in mental arithmetic with addition, subtraction, multiplication, or division. All are motivational and different from most usual drills.

Diffy

In Diffy, the student selects four natural numbers which are printed on the screen as the corners of a square. Then he finds the difference between each successive pair of numbers. The four differences become the corners of the next square. The student continues finding differences and creating new squares until the number in each corner is zero. The objective is to find four original numbers that will require as many moves as possible to reduce to zero.

A sample game asks for four numbers. The numbers are displayed like this:

```
        ?
     3  27
     13 31
```

What is the difference between the two numbers on each side of the "?"? The student is given two chances to respond

Dan Isaacson, School of Business, California State University, Fresno, CA 93740.

correctly and then the correct answer is printed. The next frame would look like this:

```
        24
     3    27
          ?
     13   31
```

Diffy keeps track of the largest number of moves by any player so far, along with the player's name, which is an excellent motivational reward.

Tripuz

Tripuz asks the student to choose addition or multiplication for practice. The computer generates three numbers:

```
     ? 18 ?
     24 48
       ?
```

and asks for a value of the ? where the middle numbers (i.e. 24 and 18) are found as the product of the two numbers on either side. The final solution to this problem is:

```
     3  18   6
      24  48
        8
```

At the secondary level, algebra students are encouraged to look for a general algebraic solution for both the addition game and the multiplication game.

Although addition and multiplication are the modes available, students must use the inverse concepts of subtraction and division to solve the puzzles.

Magic Squares

Magic Squares leads the student to an intuitive understanding of averages and arithmetic sequences. In 20 years of teaching I have never been motivated by magic squares, but now, after using the Magic Squares program, I have painlessly learned the concepts involved in designing a magic square; a big hand for QED.

Magic Squares provide an example of the power of computer simulation in problem solving. The computer eliminates computational drudgery and allows the user to concentrate on concepts and hypotheses.

Commentary

I can't end a review without commenting on programming style in general. Software producers, like students, learn by doing—each new program is better than the last from a style point of view, but it is the reviewer's job to keep producers

Arith-Magic, continued...

on their toes. I make the following obser-
vations hoping that future courseware for
schools will show increased awareness of
the psychological considerations in
courseware design.

QED has correctly allowed users to
press one key to mean a whole word (such
as Y for Yes), but the program also
accepts YED as YES. I suggest rejecting
more than one character as a response,
thus short-circuiting this problem.
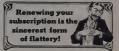
Long, animated title pages (sometimes
with sound) are great—the first time
through. But users get exasperated with
them if the program is used often. (Title
pages seem to take longer and longer each
time one uses them.) I've had comments
on some programs like: "After the fifth
time, I turned off the sound because it
drove me nuts!" Title pages in these three
games are slightly too long. It would be
nice if a user could skip the title page after
the first use.

Because everyone reads at a different
speed, I would recommend that no
message leave the screen until the user
allows it to leave. In almost all cases, that's
how QED has programmed. Several
times, though, a praise message is not
under user control and leaves the screen
unexpectedly, which jars the user.

It is desirable that users be able to page
backward as well as forward in a tutorial.
We all have read a paragraph and then
realized that we didn't remember a thing
in it. A book allows us to go back and re-
read it. Many educational programmers
have not recognized that necessity yet. I
felt a desire, several times in Magic
Squares, to return to a previous frame—
which I couldn't do. Fortunately, later on,
the program cycled through those frames
I wanted, but I was temporarily frustrated
as I couldn't get the help I needed exactly
when I needed it.

As a final note, I found no way to exit
from the program menu without turning
off the computer and re-booting. This is
an understandable consequence of the
author's attempt to protect his copyright
and prevent piracy, but it makes programs
less user-friendly. I hope this problem is
shortly overcome.

Now if programmers have heeded my
concerns, prospective purchasers may
ignore this reviewer's wishful suggestions
and consider buying a copy of Arith-
Magic. I, for one, look forward to finding
more, new releases from QED at my
friendly, neighborhood computer store.□

creative
computing
software
evaluation

The Education Game

David Lubar

In the interest of diversity, this roundup will stray somewhat from the hallowed path of drill and practice, covering some programs that attempt innovative approaches to education. In most of the cases below, the innovation is in the form of turning learning into a game. Whether this is desirable is up to the individual teacher to decide.

Crossword Magic creates what is known as the British style of crossword puzzles. This format has fewer intersections, and lacks the symmetry of American puzzles. The advantage of the British style is that it is harder to solve since the solver gains only a few clues with each new word. And words outside the topic needn't be included just for padding. Another advantage is that this style is easier to construct.

The program accepts words from the user, and attempts to fit them into a grid of up to twenty by twenty squares. Words

cross only when there is a letter in common. The program is fast, though it seems to seek the first acceptable fit rather then searching for the fit with the most crossings. Once the crossword is generated and the user has supplied a cue for each word, a printout can be generated on an Epson. Puzzles can be saved on either the master disk or a player disk. Each player disk can hold twenty puzzles, and additional player disks are available. It would have been nice if the program allowed puzzles to be saved to any disk.

When solving puzzles on the Apple, the user moves through the grid with the arrow keys. The program is smart enough to skip

over areas that aren't part of the actual puzzle grid. When the solver is finished, the answers can be displayed. Mistakes are shown, along with an error count.

As far as educational applications, that is up to the user. The program will fit the words and print or present the puzzle, but the teacher is responsible for any educational value in the puzzle. Basically, the user will get out whatever he puts in. With the proper thought, the puzzles can be put to good use in many classes, reinforcing both vocabulary and concepts. If a teacher likes the idea of using crossword puzzles in class, this program has a lot going for it.

Fellow two-finger typists, there is hope. *Mastertype* adds some fun to the drudgery of educating the fingers. The concept follows the classical approach of starting with the "home" letters (the row the fingers

Mastertype adds some fun to the drudgery of educating the fingers.

rest on), and expanding from there in a methodical manner. The fun comes in the form of a game. The player has a base in the center of the screen. Words or letters, depending on the level, appear in the corners. Enemy ships approach from each corner. Correctly typing a word results in a shot being fired at that corner. If a ship gets too close, it damages the base.

Specialized lessons, containing forty words, can be created by the user, and may be saved to either the *Mastertype Disk* or any initialized disk.

A selection of nice features augments this program. Typing speed is displayed at the end of a series, and the user can slow the program if he has trouble. While the idea that having fun while learning to type goes against all tradition, I like it.

Vocabulary Baseball from J&S contains a pool of 1000 words, along with multiple choices for definitions. Each time he goes to bat, the player can attempt a single, double or triple. The more bases he tries for, the harder the word. A correct answer puts a runner on base. A mistake is an out. At the option of the player, a game can last one, two, or three innings. The high score for each length game is saved on disk.

Brøderbund Software

David's Midnight Magic

Introducing the premier award of the software industry.

THE ATARI® STAR

Win $25,000 in cash, plus prizes and an ATARI STAR by entering your software in the ATARI® ASAP competition.

ATARI is looking for some of the greatest software from the brightest stars in the software field. And naturally, we're willing to reward you for it.

Every three months, ATARI will be presenting ATARI STARS to the writers of software programs judged first, second and third place in the following categories: consumer (including entertainment, personal interest and development); education; home business (personal finance and record keeping); and system software. The decision of the ATARI judges is final.

Quarterly prizes will be from $750 to $3,000 worth* of selected

ATARI products and an ATARI STAR. The annual Grand Prize will be the coveted Super Star trophy and $25,000 in cash.**

To be eligible, your software idea must be submitted to and published in the ATARI Program Exchange catalog. Submit your program on the official entry form which we'll rush to you if you send in the attached coupon.

The ATARI STAR Awards are in addition to the percentage of revenues you will automatically be entitled to as

a result of sales of your software programs in the ATARI Program Exchange.

If you're a star, ATARI wants to reward you. Send in the coupon today, or call toll-free 800-538-1862, in California 800-672-1850, and you could be the proud winner of an ATARI ASAP STAR Trophy, and some great ATARI prizes.

*Based on manufacturer's suggested retail price.
**Taxes are the sole responsibility of the winner. Void where prohibited by law. Winners will be notified by mail. ATARI employees and their families are not eligible. For more contest details, call the toll-free numbers.© 1981 ATARI, INC.

allowing students to compete. Though the banner for the program is in hi-res, the game is played on the text screen, with X's for the runners and white rectangles for the bases.

A clever student, who is more interested in score than challenge, will quickly discover

how to "beat the system." While every triple after the first will bring home a run, so will every single after the bases are loaded. So, after making three singles, the player can stick to singles and score a run every time. Still, the incentive to try for

triples is there, and most students will probably take a shot at the harder words.

With the large base of words, a teacher can get a great deal of mileage from the program, especially since no words are repeated during a game. And any student who can get through all the words on the disk without an out should probably be writing programs instead of playing with them.

Meet the Presidents from Versa Computing is a graphic delight. The program is somewhat reminiscent of a television game show. The student first sees a blank screen. Then lines appear, crossing in different geometrical patterns. The lines are actually unmasking a picture stored in another area of the Apple. The picture is of an American president. As the picture appears, clues are presented at the bottom of the screen. When the student knows who the president is, he types the name. The goal is to give the name before the entire picture has been revealed. After a series of ten pictures, the student is given a score based on his response time. The score for each picture starts at 100. If it reaches zero for any president, the student is given another chance.

The two double-sided disks contain 40 pictures done by Saul Bernstein, the artist who did the excellent drawings that come with *Micro Painter*. An option in the

program allows the pictures to be presented in a slide show. Another option allows the teacher to change the clues. Topping it off, the input routines are very tolerant of typing errors. Each name is recognized by a series of key letters, which can also be changed by the teacher.

Meet the Presidents makes good use of Apple graphics and sound. Again, teachers must decide whether the game fits into a curriculum. But, with the ability to change clues, the program could probably be put to good use in many situations. □

❹ Reasons Why You Should Use Radio Shack's Computer Education Series

New Part 2 Includes a 354-Page Teacher's Manual, 164 Transparencies, And 25 Student Manuals

Part 1
$159
Cat No. 26-2150

PART I: Introduction to BASIC

PART 2: BASIC Programming

A Complete Teaching Program Based on the TRS-80

$199

❶ Uniquely Comprehensive Format

Designed for classroom use with secondary students who have "hands-on" access to TRS-80 computers, Part 1: "Introduction to BASIC" begins with the most fundamental aspects of computer programming in BASIC. Part 2, "BASIC Programming," introduces more advanced concepts such as editing, arrays, string handling and memory management.

❷ Reasonable Cost

Part 1 ($159) includes 143 transparencies. Part 2 ($199) contains 164 transparencies. Each comes with 25 student manuals, with additional manuals available separately.

❸ Proven Effectiveness

Developed and field tested in public schools over two years, this material is designed for use by all teachers, regardless of computer background.

❹ Addresses a Growing Need

Today's students are growing up in a computer age. We must make sure they are prepared to function in a computer-oriented society.

For More Information on the Computer Education Series, Visit Your Local Radio Shack or Fill In Coupon and Mail to:

Radio Shack Education Division, Dept. 82-A-335
1300 One Tandy Center, Fort Worth, Texas 76102

NAME

ADDRESS

CITY   STATE   ZIP

Radio Shack®
The biggest name in little computers™

Prices may vary at individual stores and dealers.

Six management games simulate everything from a
lemonade stand to a multi-national detergent
manufacturer. One is probably right for you.

Six Business Management Games

David H. Ahl

In any given year, roughly one out of
every three hundred businesses goes bank-
rupt. Not a very high failure rate! On the
other hand, over half of the new businesses
started every year go out of business for
one reason or another within five years.

Another interesting statistic: thirty years
ago, there was one millionaire for every
11,300 citizens. Today, one person out of
423 is a millionaire. Millionaires come from
three major groups: inherited money, self-
made in small businesses, and land owners
who have benefitted from the skyrocketing
price of real estate.

While a recent article in *Fortune* points
out that luck is an extremely important
factor in determining the success or failure
of a business, without fundamental know-
ledge of how a business works, one is almost
certainly doomed to failure.

On the lowest level of business knowledge
are the interrelationships between market-

ing, production and finance. These basic
relationships can be understood by a child
of eight or nine years old running a lemonade
stand. The computer simulation, **Lemonade
Stand**, gives a child the opportunity to do
so in the comfort of his own living room.

But a lemonade stand usually has no
competition. And competition is a vital
factor in the real world. **Market** is a
computer game which introduces this most
important element. **Micro Millionaire**
introduces the additional variables of
advertising media choices and R & D
expenditures.

Cartels and Cutthroats is perhaps the
most sophisticated business management
game for a personal computer. It introduces
more "real world" variables than any of
the above games, and presents reports in a
manner most closely resembling a real busi-
ness.

The most realistic and sophisticated
business management games are those
found in various MBA programs around

the country. One of the oldest and most
realistic games is the one developed at
Carnegie-Mellon University.

The world of business and finance is
more than running a company, however.
It also involves investing in stocks, bonds
and the money market. Short of using
your own money, **The First Investment
Games** are probably the best way to learn
something about this aspect of finance.

Did playing these games improve my
management of Creative Computing?
Probably not, although they perhaps honed
my decision making capabilities somewhat.
Did playing them make me a millionaire?
Most definitely not. Although with the
rate of inflation continuing to run wild,
within thirty years practically everyone in
the country will be a millionaire. On the
other hand, a loaf of bread will probably
cost $40 or $50, so being a millionaire
won't mean very much.

Try one or more of these games. I think
you will find them interesting.

Business Management Games

Area of Management	Lemonade Stand Marketing	Market Marketing	Micro Millionaire Marketing Production
Periods Per Game	4 or more	8 or more	12 or more
Decisions Per Period	3	3	20 to 50
Time Spent Playing Game	1/2 hour	1 hour	1 hour
Total Elapsed Time to Play Game	1/2 hour	1 1/2 hours	3 hours
For Computer	Apple	Apple, TRS-80	TRS-80
Price	Free or $25 (MECC-704)	$25 (MECC-704) $25 (Eco & Social Sim'l)	$10 cassette $15 disk
Source	Creative Computing Software 39 E. Hanover Ave. Morris Plains, NJ 07950	Creative Computing Software 39 E. Hanover Ave. Morris Plains, NJ 07950	Softside 6 South Street Milford, NH 03055

Area of Management	Cartels & Cutthroats Marketing Production	The First Investment Games Investments	Carnegie-Mellon Management Game All
Periods Per Game	8 to 20	6 games	20
Decisions Per Period	7	6 to 25	25 to 40
Time Spent Playing Game	1 hour	12 hours	200+ hours
Total Elapsed Time to Play Game	3 hours	4 months	9 months
For Computer	Apple	not required	PDP-10
Price	$40	$40	$8000
Source	Strategic Simulations 465 Fairchild Dr., Suite 108 Mountain View, CA 94043	First Nat'l Bank of Boston 100 Federal St. Boston, MA 02110	Graduate School of Industrial Admin. Carnegie-Mellon Univ. Pittsburgh, PA 15213

IBM's new baby just grew up.

SEATTLE COMPUTER'S NEW RAM+ GIVES YOU THE MEMORY YOU NEED AT A PRICE YOU CAN'T FORGET.

A great new machine, the IBM personal computer, just got better. With RAM+, the new expansion memory from Seattle Computer.™

RAM+ can change your personal IBM from a pony to a workhorse by adding up to 256 K of memory. RAM+ is expandable, reliable and is made by Seattle Computer, the people who created the operating system for the IBM personal computer.

To make your IBM work even better, each RAM+ card has a RS-232 serial port which uses IBM supplied software. And RAM+ comes fully assembled, tested and with a 90 day guarantee.* The RAM+ is available in 64 K, 128 K, 192 K or 256 K, with affordable prices ranging from $475. to $1,075. Expansion kits are also available.

How to order: Call Anna Ford at **(206) 575-1830** for the location of the nearest RAM+ dealer.

The RAM+ —a great addition to your IBM personal computer. At a great price.

Dealer inquiries welcome

△ SEATTLE COMPUTER

1114 Industry Drive, Seattle, Washington 98188

*Extended warranty available.

Lemonade Stand

The rules of the game are quite simple and are displayed on the screen at the beginning.

"In this small town, you are in charge of running your own lemonade stand. You can compete with as many other people as you wish, but how much profit you make is up to you (sales of the other stands will not effect your business in any way). If you make the most money, you are the winner.

"To manage your lemonade stand, you will need to make these decisions every day: (1) how many glasses of lemonade to make (only one batch is made each morning) (2) how many advertising signs to make (the signs cost 15 cents each) (3) what price to charge for each glass.

"You will begin with $2.00 cash (assets). Because your mother gave you some sugar, your cost to make lemonade is 10 cents a glass (this may change in the future).

"Your expenses are the sum of the cost of the lemonade and the cost of the signs.

"Your profits are the difference between income from sales and expenses.

"The number of glasses you sell each day depends on the price you charge, and on the number of advertising signs you use.

"Keep track of your assets because you can't spend more money than you have!"

At the beginning of each playing day, a cute graphic is displayed on the screen showing your stand and the weather forecast. The program then asks each player to make his three decisions for the day. We tried the game with two players. Player 1 took a conservative approach of making twenty glasses of lemonade, charging 10 cents each and making three signs. Player 2, on the other hand, spent his entire $2.00 in assets to make forty glasses of lemonade and eight signs. Both players continued this strategy for the twelve days that we had arbitrarily decided would be the end of the game. By day 2, player 2 had discovered that the market is quite price-sensitive and that even sixteen advertising signs were not enough to lure customers into paying 15 cents a glass.

Various random factors crop up during the game. For example, on day 6 it was hot and dry which led to substantially increased sales for both players. Player 2, being more aggressive, charged considerably more for lemonade on this day and made the highest profits for a single day—$4.50. On day 8 the street department was working and there was no traffic on the street. Again, player 2, with his much more aggressive strategy thought he could attract business from other streets by putting up more signs. Unfortunately, it didn't work and his loss on that day was the largest in the entire game.

By the end of the twelve days of play, both players had learned about the size of their market and the price it would bear as well as the advertising necessary to sell the products. And the two players had come remarkably close together in their respective daily decisions. Player 1 "won" the game with total assets at the end of twelve days of $26.77 while Player 2 had accumulated $23.09.

The game is suitable for introducing youngsters to economic decision making at a very simplified level. It is in the public domain and available for many computers (it comes free with an Apple on the system master disk).

Lemonade Stand is also available for the Apple on the disk "Elementary—Volume 3 (Social Studies)" produced by the Minnesota Educational Computer Consortium. This disk also includes "Sell Apples," a tutorial game which helps players find the optimum selling price for a product, in this case apples. "Sell Plants" is a tutorial on the effect of advertising on sales. These two programs lead up to "Lemonade" which finally leads into "Sell Bicycles" which is the same program as the one described below as "Market." This disk also includes a Civil War simulation and two tutorial programs which provide drill and practice on the location of states and capitals. The disk, MECC-704, comes with a comprehensive instruction book and is available for $24.95 from Creative Computing Software, 39 East Hanover Avenue, Morris Plains, NJ 07950.


```
ON DAY 4, THE COST OF LEMONADE IS $.04

LEMONADE STAND 1          ASSETS $3.21

HOW MANY GLASSES OF LEMONADE DO YOU
WISH TO MAKE ?30

HOW MANY ADVERTISING SIGNS (15 CENTS
EACH) DO YOU WANT TO MAKE ?4

WHAT PRICE (IN CENTS) DO YOU WISH TO
CHARGE FOR LEMONADE ?12


$$ LEMONSVILLE DAILY FINANCIAL REPORT $$

    DAY 4                     STAND 1

    30   GLASSES SOLD

$.12  PER GLASS          INCOME $3.60

    30   GLASSES MADE

    4    SIGNS MADE       EXPENSES $1.80

             PROFIT   $1.80

             ASSETS   $5.01
```

Market

Simulation of a Competitive Market

Market is a simulation game in which two companies are competing for the market for racing bicycles. At the start of the game, the program informs the players of the fixed and variable production costs in marketing the product and the number of units that can be sold with no advertising at a specific price. The program then assigns to each company initial values for inventory, cash on hand, and total assets.

Each player must make marketing decisions quarterly which include the production level, advertising budget, and product selling price. After these decisions have been entered for each player, the program reports the results of these decisions in a marketing summary. For each company the report lists: (1) profit; (2) percentage share of the market; (3) cash on hand; (4) number of units sold; (5) number of units in stock (inventory); and (6) total assets.

player goes bankrupt, or until either player accumulates a certain amount of assets. Longer games may be saved in progress and continued at a later time.

During the play of Market, random variables crop up such as a new labor contract which increases the variable production costs or a presidential wage-price freeze which prevents raising the price of the product over the next two quarters.

During play at the beginning of quarter 10, a transportation strike occurred and both companies were unable to move the products to wholesale distributors. "Negotiations have begun, but the hope of a settlement looks dim." An event like this is the equivalent of a cloudy day in Lemonade Stand and, indeed, the strike played havoc with the results of both firms.

We played the game for fifteen quarters

```
            QUARTER 4 RESULTS
            ----------------

COMPANY    PRICE      BIKES SOLD    PROFITS

   1       $927          11         $-602

   2       $515          52         $732


COMPANY    CASH       INVENTORY     ASSETS

   1       $4720         122        $7160

   2       $7583          78        $9143
-------------------------------------------
            QUARTER 5
     UNIT PRODUCTION COST = $22

COMPANY 1
UNIT PRICE?
```

In a sense, the same three decisions are required in Market each quarter as are required in Lemonade Stand each day, i.e., production, advertising, and price. However, there the similarity ends. Market is a considerably more comprehensive game and production costs include both fixed and variable costs. Fixed costs, for example, include expenses such as rent on factories and warehouses, heat, light, maintenance, and executive salaries.

In addition, Market simulates the interaction between two competitors, and the decisions made by one affect the results of both.

Market may be played for a specified number of quarters, until one or the other

until Company 1 finally popped over the magic $12 million in assets mark. It was declared, by the program, the winner of the game.

Market is available for the TRS-80 and Apple. It is one of three programs in the package "Economy and Society Simulations," which also includes Limits (a world growth simulation) and USPOP (forecasts future US population size and composition). The package of three programs costs $24.95 and is available from Creative Computing Software, 39 East Hanover Avenue, Morris Plains, NJ 07950.

Market is also available for the Apple on MECC-704 ($24.95) along with other programs. □

Micro Millionaire

This is a one-player game in which the player attempts to establish a company in the computer software business. The simulation is based on the experience of The Software Exchange. At the beginning the player is given a small amount of money with which to begin his company. From this amount is immediately deducted the cost of a small computer. Then begins the monthly cycle of the game. The player must make decisions on development (evaluating and debugging programs), production quantities, reformulating his line of products, changing prices, the number of working hours you are putting into the company (moonlight or full time), using WATS lines, and placing advertising in four magazines.

Upon beginning the game the first time, I found that I had an investment of $6000, $2000 of which was used to purchase my computer. I received two software submissions to evaluate, one of which I wished to add to my line. Unfortunately, there are practically no instructions with the game other than those which appear on the screen and it was not at all clear how I should add this package to my line and assign a price to it. Before I knew what happened, the program went on and told me that I had no sales for the month and $150 of out-of-pocket expenses.

Given that I had no sales, I elected not to hire any order takers or technical staff nor to put in a WATS line. But I sure wanted some products to sell. Finally, in the third month I determined that the only way to publish a program was to answer YES to the question "Evaluate further?" after which it could be named and added to the product line.

Unfortunately, because I had not figured out how to add products in the early months of play, by May I had only two products, one put in the product line in March and the other in April. I found my expenses had exceeded my cash on hand and I was forced to borrow money from a loan shark at 25% interest per month. Needless to say, this had a rather nasty affect on my cash flow. By July, I was borrowing over $8000 a month and my equity had declined to -$4861.

By December after one year of play, I had ten products in my product line ranging in price from $12 to $50. Sales of each were running between 8 and 25 per month bringing me total revenue of around $1600 a month. My equity had declined to a staggering -$42,833 and I had over $45,000

of loans with a loan shark. Hence, it was no great surprise that when tax time rolled around in January, I was unable to pay my tax and the company went bankrupt. (The program didn't say whether or not the loan shark had my kneecaps shot.)

In my second play of the game, I managed to keep my head above water. While my performance was nothing to write home about, at least I did not go bankrupt and showed a modest profit at the end of the first year.

All in all, *Micro Millionaire* is not an easy game to play or master. More and more decisions are required each month of play as you add more products to your product line. In order to play successfully, I found that pencil, paper and pocket calculator were a virtual necessity. Also, allowing adequate time between monthly decision-making cycles to analyze the results of the previous months is necessary to play the game successfully. The weakest aspect of the game is the documentation — only three pages are included and they do not describe how entries are to be made or give any clue as to the range of acceptable values for the different entries. Hence, one or two plays of the game probably are necessary before one can get a feel for the values necessary to complete a successful game.

Micro Millionaire is available from Softside, 6 South St., Milford NH 03055 as *Softside* magazine, Vol. 2 No. 9. Price is $9.95 with cassette, $14.95 with disk, or $2.00 for the magazine alone (which contains a complete listing of the game for the TRS-80). □

"Someday, all this will be yours."

Cartels and Cutthroats

Cartels and Cutthroats is a business management simulation game. It is packaged in an attractive 8-1/2" x 11" four-color box which contains a 16-page instruction manual, one-page simplified instruction card, and Apple diskette (16-sector, DOS 3.3).

The game is a simplified simulation of two or more companies marketing products in a single product area. Each player "runs" a manufacturing company that buys raw material, produces finished goods and sells those finished goods in a competitive market. The object of the game is to earn the most total income. This is shown cumulatively as increase in "equity" which is the player's ownership in the company. Each turn is a quarter of a year.

The general play sequence is: review last period's performance, make plans for the current quarter, implement the plans (enter them into the computer) and then review performance again.

Sales (and profit) are based on how good your plans are relative to the economy and to the plans of your competitors.

At the beginning of each turn, each player is given information about the economy and the industry, as well as some market research data about the other companies in the game. Economic indicators include GNP, CPI, prime rate, and a purchasing power index. In addition, at the beginning of each quarter, each player is given information on raw material supplies and prices, government actions, bankruptcy filings, and other items of specific interest in the industry.

After the summary economic news, each player is given several reports on the performance of his company. These reports include a profit and loss statement from

```
   CREATIVE COMPUTING
PROFIT & LOSS STATEMENT FOR 4/2

                #     %+!   #=#   %ch
Revenue      28,900  100   578   1+

COG Sold      9,433   33   189    6-

Overhead      2,986   10    60   20+
Adver & Mrkt  2,480    9    50   33+
Resr & Devlp  1,500    5    30   20+
Depreciation  1,970    7    40   21+
Misc Expense      0    0     0    0
Oper Expens   3,944   31   179   27+
Interest      1,040    4    21    5-
Taxes         4,539   16    90    4-
            ----------------------------
Total Costs  23,956   83   479    7+

Net Income    4,944   17    99    5-
```

```
   CREATIVE COMPUTING

   BALANCE SHEET FOR 4/2

                #              75,817
Cash                   Loans
Invent.       3,019    Liabil.  35,817
Bldgs.       94,608
Aut.Equ.     12,346    Equity   74,156
Assets      109,973    LiabEqu 109,973

  RATIO ANALYSIS       EQUITY ANALYSIS

Incm/Assets    4%     Last      69,212
Incm/Equity    7%     Income     4,944
Loan/Assets   33%     Current   74,156
```

Figure 2. A typical profit and loss statement and balance sheet.

the previous quarter and a balance sheet including ratio analyses and equity analyses. Also provided are: a sales activity report, performance chart, production cost summary, inventory report, factory and labor summary, market summary, and a market research report.

Following these financial and statistical reports are memoranda and letters from other managers and outsiders. The financial manager always provides a report pertaining to loan balances, interest rates, and line of credit available from the bank. In addition, the player may get reports from the director of R&D, production, or other members of his staff.

Occasionally letters are received from irate customers, charities, and other outsiders who have some direct or indirect interest in the business. Using this rather substantial mass of information (roughly six feet of line printer paper per quarter), the player must formulate his decisions for the upcoming quarter.

An annual report is issued at the end of

```
   CREATIVE COMPUTING
   PLANNING SHEET
   For Period 2/4

RM Purchase: ------
Production:  ------
FG Price:    ------
Adv & Mkrt:  ------
Res & Devlp: ------
Loans:       ------
Factories:   ------
Special Memos: --------------
```

Figure 1. The business planning sheet for the first two years of play.

each four quarters, and at the end of the game, and an end-of-game summary is produced for all the players.

At any point, the game may be saved in progress. The player is prompted to insert an appropriately initialized disk and give the game a title. At any later time, the game maybe recalled and play resumed from the point at which it was saved.

Having just concluded the negotiations for the acquisition of *Computers and Programming Magazine, Small Business Computers Magazine*, as well as doing all of the end-of-fiscal-year financial calculations for Ahl Computing Inc., I decided to take a "busman's holiday" by playing a game of Cartels and Cutthroats.

Reading and digesting the rules of Cartels and Cutroats takes about one-half hour while playing the game can take anywhere from one hour to a day or more depending upon the number of players and the "think time" taken between turns. The following is an abbreviated report on one play of the game.

I loaded in the game and selected the type of game I wished to play. I opted for a hard copy game and did not claim a beginner's handicap. Since I was playing alone, I asked the program to supply two computer players to manage competitive companies.

When the game started, the first thing that appeared on the screen was a scenario. The scenario read as follows:

"You have acquired a manufacturing company that makes a product considered a luxury by most consumers.

Your raw material prices are stable and based on abundant supplies.

© Creative Computing
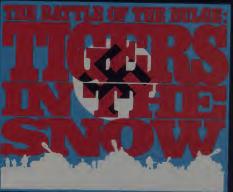
"It's our latest model. There's absolutely no problem it can't solve or run away from."

TWO NEW GAMES FROM SSI FOR THE APPLE AND THE TRS-80!

THE BATTLE OF SHILOH: A grand-scale simulation of the first great battle of the Civil War, pitting the Confederate Army against Union forces. (48K disk version pictured.)

TIGERS IN THE SNOW: Crush the Nazi Tiger tanks in an infantry sweep across the dark, frozen forests in this battle-level simulation of Hitler's last desperate attack.

We know it hasn't been easy for you TRS-80® owners to see so many great made-for-Apple-only games from SSI pass you by. But then, it hasn't been easy for us to design games for a 16K cassette format good enough to meet our critical standards.

After all, we've got a reputation to protect, a reputation in strategy gaming for unsurpassed sophistication, innovation, realism, and playability.

Well, our designers have been hard at work, and we've not only met but surmounted the challenge. We're delighted to announce two historical wargames — deserving of the SSI label — for both the Apple® and the TRS-80® (16K cassette for the TRS-80 Model I and III; 48K disc for Apple II with Applesoft ROM card).

Combining our extensive game-design experience and superior programming techniques, we've given a fresh new look and feel to these favorite classic battles.

At $24.95 each for TRS-80 cassette and $39.95 each for Apple disc, these are extraordinary games at quite an ordinary price.
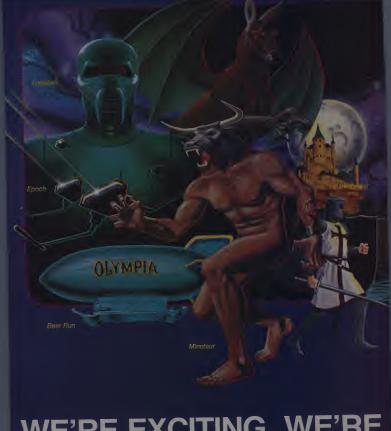
So head on down to your local store and check them out today!

VISA and M/C holders can order by calling 800-227-1617, ext. 335 (toll free). In California, call 800-772-3545, ext. 335.

To order by mail, send your check to: Strategic Simulations Inc. Dept. CC6, 465 Fairchild Drive, Suite 108, Mountain View, California 94043.
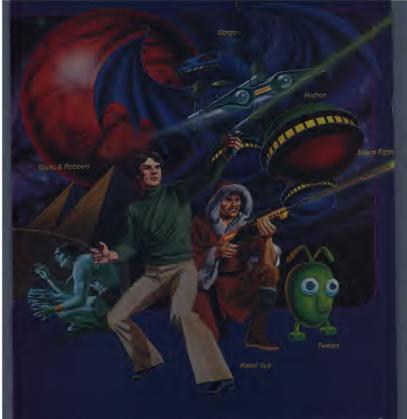
All our games carry a 14-day money-back guarantee.

STRATEGIC SIMULATIONS INC.

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

Apple is a registered trademark of Apple Computer Inc. TRS-80 a registered trademark of Tandy Corporation.

CIRCLE 245 ON READER SERVICE CARD

Borg

Epoch

OLYMPIA

Beer Run

Minotaur

WE'RE EXCITING, WE'RE

The economic outlook for the next five years is for strong expansion in the gross national product (GNP) with an economic down-turn likely some time in the future.

Inflation, as measured by the consumer price index (CPI), is expected to continue at a controlled level.

In addition, the impact of chance on the companies and the economy should be slight."

The game then asked "Is this scenario acceptable?" To which I responded, "Y." If I had not responded yes, I would have been given the opportunity to select another scenario. Since I was playing a hard copy game, the scenario was then printed out on the line printer.

A note about printing: the game assumes a standard line printer with sixty lines on a page. Unfortunately, I was using an Apple Silentype Printer with continuous feed paper. Consequently, the spacing up after each report to the "bottom of a page" wasted a substantial amount of (rather expensive) sensitized paper.

Playing the game, one is expected to make decisions based on facts and figures

```
                     SALES REPORT

          SALES ACTIVITY 4/2
                          units       %ch
Available for Sale          66        10+
Total Demand                50         7-
Units Sold                  50         7-
Excess Sales                16

     CONSUMER PREFERENCE ESTIMATES
                    market rank  % sales caused
Price                     1           31
Advertising               1           31
Product                   1           17
History                   1           13
Other                     1            8

          PERFORMANCE CHART
    income                         revenue
3/3 ################
3/4 ####################
4/1 ##############
4/2 #################
    0                            30,720

-----------------------------------------

          PRODUCTION REPORT

     RECENT PRODUCTION COSTS
                       RM       F'd     %ch
RM Storage              0        0        0
RM Purchased        3,480       60       50
FG Storage            120        6       20   54+
Direct Labor        7,643       60      127   10-
Prod Overhead       1,534       60       26    4-

          CURRENT INVENTORY
                       units   warehs  %full
Raw Material (RM)        0        0
Finished Good (FG)      16        2             80

   RECENT LABOR             CURRENT PLANT
Rate Pa+ed    10.03     Factories       5
# Employees    15.4     % Automated     37
# Hired         254     Nrml Capacity   6
```

Figure 3. The sales report for year 4, quarter 2.

Figure 4. A market summary report.

---

which appear on the screen. However, because a hard copy game uses upper and lower case, the material appearing on the screen looks like gibberish and bears little resemblance to letters. The numbers are, of course, the same; however, without the labels they are somewhat difficult to use.

Since the economy was going up and I was in a luxury market I decided to increase advertising and to attempt to expand my share of market relatively quickly. I also increased my loan volume from the bank to build a new factory. Although the computer program takes much of the drudgery out of management, one still has to be careful to not exceed the total funds available. I found a pocket calculator quite handy to compute out my raw material and production costs. Thus I could keep my bank loans closer to the exact amount needed thereby not incurring additional interest expenses.

It was with a certain sense of anxiety that I waited for the printout each quarter. The first quarter results indicated that my expansion strategy worked from the standpoint of increased sales. Unfortunately, the increased sales were not reflected in increased profits. I also discovered, much to my chagrin, that my hand calculator figures were somewhat in error and my finance director had to secure an emergency loan to finance the additional production as well as build the factory. On the other hand, I was now outselling my competition by a substantial margin, my share of market having increased from 33% to 41% and it became my goal to turn this increased sales and capacity into increased profit.

As I continued to play the game I found the raw material purchase, production and pricing decisions getting easier with each round. However, I began to note the other factors such as loan volume, income to equity ratio, warehouse usage percentages and the like and attempted to fine tune my decisions to optimize both my available funds and other resources such as factories, warehouses, and labor.

The end of the third quarter brought some bittersweet news. My factory workers were threatening a strike if they were not immediately given a 20% raise but the personnel director felt that they would accept a 10% increase. On the other hand, my director of R&D reported that "we have acquired some inexpensive computer equipment (three Apples) that we can use for production scheduling and inventory control and reduce production overhead costs by 10%.

Figure 5. A typical memorandum from the finance director.

In the third quarter one unforeseen disaster occurred that had nothing directly to do with the play of the game. In particular, the paper ran out in the middle of a run. Since I had no particular desire to start over from the beginning I didn't press reset or any of the other keys that might have stopped the printer. Instead, all of my results were "printed" on the blank roller and my decisions for the ninth quarter had to be made by the "seat of the pants." Needless to say, my performance suffered as a result. This, however, may be true to real life situations when the chief executive officer is away on vacation or at a conference and leaves the running of his company in the hand of junior managers. In this case, it resulted in a rather disastrous quarter.

"Good news, they've got the computer part. Bad news, it'll take six months to deliver it."

Cartels, continued...

```
                 MEMORANDUM
TO:   DAVID AHL, PRESIDENT
FROM: Director of R & D
DATE: 2 4
SUBJ: Research Break-Through

   We have discovered a method of using
automated equipment that will reduce
our labor cost by 10% for an investment
of $3750. At our present production
rate and costs this investment will be
recovered after only 1.7 quarters.
Send a memo to 'A' to approve this
automation proposal.
   We have also developed an improved
product design that the marketing de-
partment believes will increase our
sales by 5%. The total costs for
changing over our facilities to produce
this improved product will be $1300.
Send 'P' a memo to approve this change.
```

Figure 6. A typical memorandum from the director of R&D.

Three quarters later (first quarter, fourth year) I had finally overcome the havoc wrought by my blind decisions earlier. My market share had climbed to 53% and I was able to begin to repay some of the loans required for constructing factories in earlier quarters. In addition I was using my warehouses efficiently and showing a reasonable return on sales.

Although my intention had been to play for five years, at the end of four years the clock struck 1:00 a.m. and I was about to

```
             FINAL REPORT
                ( 4 4 )
      REAL DOLLAR EQUITY GROWTH

              Total $   Ave Rate  % Growth
   Company   Increase   Return    Last Yr
CREATIVE      53,282      31          33
MICRO         28,294      20          18
APPLERS       27,920      19          16
   Economic Growth . .    10    . .    12

INCOME PER SALES            TOTAL SOLD

Company      %          Company      Units
CREATIVE     17         CREATIVE     645
MICRO        16         APPLERS      382
APPLERS      16         MICRO        349
--------------------------------------------
            PRIMARY COST ANALYSIS
        (overall %'s of total revenues)

                   MICRO
           APPL    CREA
RW Storage   0      0      0
Raw Costs   10     10     10
Labor Cost  31     29     26
FG Storage   0      1      0
COG Sold    42     39     37
Overhead    16     18     11
Depreciat.   4      5      5
Adv & Mrkt   6      7      7
Res & Devl   2      2      5
Misc Exuns   0      0      0
Oper Expn   28     30     28
Interest     0      0      3
Taxes       14     15     15
TTL Costs   84     84     83

Net Income  16     16     17
Other Incm   0      0      0
```

Figure 7. A final report (identical to an annual report at the end of each year of play).

turn into a pumpkin so I pressed the Q key to get the final summary. It indicated that my total dollar increase, average rate of return, and percent growth in the preceding year had far outpaced my two computer controlled competitors. I had spent heavily on research and development (5% of sales) while both of my competitors had spent about 2%. This led to some impressive efficiencies giving my five plants a total capacity of 61,000 units because they were 37% automated. This compared favorably to my competitors who were only able to produce 10,000 units per plant. On the other hand, my interest expenses far exceeded theirs because I was continually having to invest in a new plant and equipment that were developed by my R&D department.

My labor expenses were considerably below those of my two computer competitors, probably because I had played brinksmanship with the unions on two different occasions.

All in all I found this a surprisingly realistic and quite addictive simulation. While it hardly was a fraction of the variables or the challenge of the management game that I played over a one year period as an MBA student at Carnegie-Mellon University, it nevertheless was stimulating and instructive. It clearly will not make a management expert out of a shipping clerk; however, for the person interested in honing his or her analytical skills related to business, this is an ideal first step.
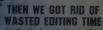
All told, it took about one-half hour to digest the rules and two hours to play the game. Someone serious about improving one's management skills will wish to play the game several times trying different strategies of advertising, R&D, and pricing. I strongly recommend that two or more people play simultaneously, as this will give a much stronger element of competition than the computer can provide. Furthermore, it resembles the real world much more closely.

Cartels and Cutthroats is produced by Strategic Simulations Inc., 465 Fairchild Dr., Suite 108, Mountain View, CA 94043.

SuperScribe II

FIRST WE GOT RID OF THE LOWER CASE ADAPTER
and put lower case on the screen
WITHOUT HARDWARE.

THEN WE GOT RID OF THE 80 COLUMN BOARD
and put seventy columns of text on the screen
WITHOUT HARDWARE.

THEN WE GOT RID OF WASTED EDITING TIME
by developing printer spooling
AGAIN WITHOUT HARDWARE!

AND WE'RE ALWAYS LOOKING FOR NEW WAYS TO SAVE YOU TIME AND MONEY.

When ON LINE SYSTEMS began work on SUPERSCRIBE II, we wanted to make it the most powerful word processor on the market. We began by putting in boldfacing, underlining, form letter capabilities, global search and replace, support of multiple disk drives, full macro capabilities, generation of up to four indices, complete formatting capabilities, a software based keyboard buffer, full editing capability of any DOS 3.3 text or binary file (including APPLESOFT programs) and all other features you need for professional word processing. That was just the beginning.

Our next step was to make SUPERSCRIBE II hardware independent, so we eliminated the lower case adapter and the 80 column board. SUPERSCRIBE II is the only word processor on the market to offer true software based lower case and a 70 column screen available today.

The move that put us over the top was the introduction of printer spooling* which allows you to do your editing while printing, making your word processing time more efficient.

With SUPERSCRIBE II, we've added all the features you need, did away with all the problems you can do without, and we did it all at a price that you can afford–$129.95.

*Printer spooling feature only available for printers with APPLE parallel or SSM AIO interface, and EPSON printers/APPLE II and APPLE II Plus are registered trade marks of APPLE COMPUTER INC.

We Don't Think That Word Processing Should Cost An Arm And A Leg, And At On-Line Systems, We Won't Stand For It.

SUPERSCRIBE II runs on any 48K APPLE II, II Plus with DOS 3.3 and is available now at your local computer store or order directly from

ADD ONE DOLLAR FOR SHIPPING   VISA, MASTERCHARGE, CHECK, C.O.D.

ON-LINE systems

36575 MUDGE RANCH ROAD · COARSEGOLD, CA 93614 · 209-883-6858

CIRCLE 211 ON READER SERVICE CARD

The First Investment Games

Computer wargames played by mail have been with us for several years. However, I'd never heard of a computer management or investment game by mail until a letter arrived in my mailbox inviting me to play The First Investment Games, designed to simulate investment decision making. Six computer scored games are marketed by the Old Colony Trust Division of The First National Bank of Boston to help potential customers get a better understanding of the complexities of securities investing.

In the series of six games, the moderator is the computer. After studying material explaining investment principles, the player decides which moves to make, fills in a game entry card and mails it to the bank. With each game the player can also test what he has learned by investing $100,000 of play money. The player gets back a computerized letter with a score for the game, answers to questions missed, and the results of his $100,000 investment. The subject matter of the six games, worked out on game boards and backed with substantial explanatory charts, strategy papers and glossaries, focuses on the business cycle, market analysis, bonds, security analysis, investment strategy and portfolio structuring.

Playing the Game

The computer-generated welcoming letter with the first game tells me that the games start where the process making investment decisions itself starts—forecasting the business cycle—and then proceeds in logical order to the ultimate goal: choosing an investment portfolio with above average returns and below average risk. Each game is packaged in a file folder, all six of which can be kept in the supplied gold portfolio box. On the file folder is a bit of make-believe called "Your Role" which is the suggestion that you, for the purpose of the game, pretend to be an investment expert at Old Colony. In the business cycle game, I assumed the role of George Brentano, a member of the economics department. I found out that I am a 38 year-old Ph.D. in economics with a reputation for speaking out on public policy. I also learned that I have an excellent record in forecasting economic trends and have a computer terminal in the corner of my office.

The scenario went on to explain that I am responsible for the bank's money market newsletter. In one letter I planned to use a number of economic indicator charts. A series of these charts was mishandled at the printer (boy, do I know about that) and I am called in at the eleventh hour to restore the missing parts of the graphs. Supposedly, I can do this easily by simply studying the related charts.

The raw material with which I had to work included a one-page set of rules, a six-page briefing from Mr. Brentano, and large wall chart which showed, in graphical

Elements of the Business Cycle Game. The player has to choose the correct direction of the consumer price index after considering several options. Plastic overlays allow the player to experiment until he reaches a solution. Answers are sent to Old Colony in Boston for computerized scoring.

45

Investment, continued...

form, sixteen economic indicators for the past 25 years such as the S&P stock index, housing starts, money supply, consumer price index, etc. Also included were three game boards with clear plastic overlays which represented the "printer" errors. In case 1, I had to restore the missing portion of a CPI index chart; in case 2, the missing part of a S&P stock price index and in case 3, the missing part of a consumer installment debt index.

After doing that, I was handed $100,000 to invest at the start of five mystery years. For each mystery year I was given a "snapshot" of four economic indicators.

After completing my analyses, I filled in my reply card, mailed it off to Old Colony, and anxiously awaited the results. My anxious waiting paid off about a month later when I received another computer-generated letter which said "Congratulations! You played the business cycle game perfectly and scored 1000 points." It went on to explain why my moves were correct (perhaps the computer thought I was just guessing). The letter also stated, "Your contest investment gained $61,714" and the next paragraph stated "The maximum possible gain for this contest is $61,714," and went on to explain how this could be achieved. The format of the letter suggests that the same explanations are sent to

players who get the right answers as well as those who do not. Given that some of the later games require as many as twenty-five decisions per game, this is probably the only viable alternative.

Flushed with my success, I went on to the second game of the series, the market game. In this one I assumed the role of Robert Regis, a market technician who watches charts, trading volume and the like. I worked through the game, and sent back my card. Two months later, and no reply. So I called the toll-free number and inquired what happened? Needless to say, my card had been "lost" in the mail. However Rachel Belash, a real assistant

"Why must you ignore the GIGO Principle?"

vice-president at Old Colony, took down the information I had retained on my worksheets and had it inserted in the computer and I received my reply in due course. Although my investment decisions were quite good and scored the maximum possible gain, I got no points from this round as I had not retained the answers to my three cases.

From then on I kept Xerox copies of my answer cards, however, no further cards were lost.

To help learn the rudiments of investment decision making, the games provide a substantial amount of information. While most of it pertains specifically to the game, some is of a reference sort and can be used in real financial decision making. Even if one does not use all the information in investment decision making, it prepares one to deal with stock brokers, as well as better understand articles in *The Wall Street Journal*, *Barron's*, and other financial media.

The game package at $40 is considerably less than the price of most correspondence courses in investing. While it can't be played over and over like a computer game, I still felt it represented an excellent value.

For more information, contact Old Colony Trust Division, The First National Bank of Boston, 100 Federal Street, Boston, MA 02110.

CMU Management Game

Probably the most comprehensive of any computer management game is the one played by second year MBA students at the Graduate School of Industrial Administration (GSIA) at Carnegie-Mellon University. Modules of the game originally started to take form in the mid-50s and, by 1960, there existed a reasonably comprehensive version of the entire game.

I got my first exposure to the game in 1963 when, as a first year student at GSIA, I was a member of an auditing team who acted as a CPA firm to one of the second year student teams playing the game. In addition, as one of the few graduate assistants familiar with several computer languages, I was given the job of translating the game from GATE (a cross between a low- and high-level language used in programming the now-obsolete Bendix G-15 and G-20 computers) into Fortran II.

Unlike other marketing games, the CMU game ties in all the elements of management such as production, marketing, product research and development, finance, labor negotiations, multi-national agreements to share product developments, point-of-purchase displays, and the like. Over the years, more and more elements have been incorporated into the game until today. it stands at the pinnacle of all management games in accurately simulating management in the real world. The game simulates the nationwide detergent industry with three firms in direct competition. In general, two or three parallel games may be going on at once simulating say, the United States, Europe and Canada. There is no direct competition between firms in different nations, although there may be trade between those firms.

"Oh, oh, looks like the computer's overloaded again."

The CMU game is more than just a computer game. While many of the results of management decisions are determined by the computer simulation, the results of several key decisions are determined by personal interaction of the teams with external factors. During the course of the game, which lasts two semesters, most teams will have dealings with four or five external parties. These include a board of directors consisting of faculty members and outside businessmen, union negotiators who are actual labor negotiators from the United Steelworkers, a banker for the purpose of establishing a line of credit or obtaining short term loans, and an auditing firm. An attorney may be called upon if students wish to file injunctions stopping or delaying government regulations or file suits against other teams.

A "brief" description of the game runs over 12-pages. It describes board meetings, the market, and the decisions that the players must make regarding wholesale price levels, advertising, sales force, retail display allowances, and other marketing-related decisions. Teams also make decisions regarding production, hiring, firing, ordering of raw materials (some of which have quite different lead times for delivery), equipment maintenance, use of central and regional warehouses, transportation, and other production related decisions. Teams also must make decisions about product development including blue sky R&D, directed R&D, and product improvement R&D. New products must be test marketed or run the risk of failing on a national scale. A finance officer must manage the current assets (cash and marketable securities), obtain debt funding, both short-term and long-term; and disperse and collect payments.

Each firm, not surprisingly, is subject to external events. Information pertaining to these events is published at least quarterly in the TechStreet Journal. These events usually simulate real-world occurrences. For example, the nation is currently facing an inflation of 12% per annum, the supply of one raw material is controlled by ORMEC (Organization of Raw Material Exporting Countries) and the FTC is considering requiring labeling with full content disclosure on all soap products.

Each team starts with a history of three years of their firm. Hence, all firms do not start off on equal footing. New decisions, each representing a simulated quarter of a year, must be made weekly.

All in all, the CMU management game is an exceptionally comprehensive computer simulation and one not likely to be available any time soon to the personal computer user. However, for those with $8000 who wish to play the very best computer management game, and get an MBA in the process, send your application to the Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA 15213. □

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

IT JUST DID.

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

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

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

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

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

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

CIRCLE 251 ON READER SERVICE CARD

A New Entry in The Education Market

Bob Callan

The use of computers in education today is varied. The computer is used to teach programming and develop familiarity with languages; as a teaching machine, with software packages designed to teach specific parts of the curriculum; and as an administrative tool.

Monroe Systems for Business recently announced its entry into the educational computer market with the introduction of the EC 8800.

The EC 8800 is aimed at what Monroe considers to be a gap in the educational market: junior high through college.

Video

The EC 8800 uses an RGB color monitor or a monochrome composite video monitor as a display. It eliminates the need for composite video circuitry and for an RF modulator.

Disk

The EC 8800 uses a 5 1/4" disk drive, which is enclosed in the case and comes with the computer. An unusual feature of the disk system is its capacity to store data: 320K on one disk, with a format of 16 sectors and 80 tracks per disk (double density).

In addition, there is a second disk drive available which can boost the storage capacity of a system to 640K. Because access to data is rapid, and the storage capacity is large, applications which would normally require that disks be changed in the middle of the program or

Bob Callan, 19 Roosevelt Trail, Hopatcong, NJ 07843.

that extra drives be added can often be done with one disk and one drive.

Memory

There is 128K of standard RAM available in the EC 8800. The multitasking Monroe operating system occupies 40K when it is loaded from a disk. The rest of the memory typically is split between a language (such as Monroe 32K Basic), high-resolution graphics, ISAM (Indexed Sequential Access Method) files, and an application program and data. The user memory available is primarily a function of the language (Basic, Pascal, Assembler, Pilot) used.

Graphics

The text screen has upper and lower case characters available as well as double-size and flashing characters. In addition, there are lo- and hi-resolution graphics. In the lo-res screen, either individual blocks (a 72 x 80 grid) or lo-res graphics characters (a 24 x 40 grid) can be used. The lo-res graphics characters are the same size as one text character, similar to the PET graphics characters, but in color.
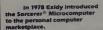
Each letter represents a different configuration of six blocks, and when printed after a lo-res command will produce a combination of these blocks. These can be combined to produce lo-res shapes that are easy to move around on the screen.

In the lo-res mode there are eight colors available, all of which can be used on the screen at the same time.

The Basic commands allow the user to plot points and lines between points, paint in irregularly shaped objects, fill in rectangles, create circles, define a part of the screen as a string, and create shape tables. The part of the screen that is defined as a string can then be printed as a string at any location on the screen. The shape tables that are created can be rotated, scaled and erased at any point on the screen. There is also an animation mode with two colors that makes rapid animation possible.

Sound

The EC 8800 contains a standard sound board which produces four programmable channels. The frequency response is 92 to 20,000 Hz. Both programmable and manual volume control are provided.

THEY SAY THE JAPANESE
ARE COMING AND WILL
DOMINATE MICROCOMPUTING.
THE INNOVATORS SAY WELCOME!

In 1978 Exidy introduced the Sorcerer® Microcomputer to the personal computer marketplace.

It was clearly ahead of its time and the competition in price and performance. The graphics were superior, upper/lower case characters were standard and numeric keyboard was included. Printer, communication and dual cassette electronics were built-in, not options. Twice as much information was displayed on the screen. **The competition created their next generation to catch up.**

In 1980 Exidy introduced their integrated desktop Computer System 80 for the very small business. It was an extension of the Sorcerer® computer . . . not obsoleting it but expanding its capacity from the home to the office. Its price/performance outstripped the competition in desktop computers. Dual disk drives with 1.2 million words of information, letter quality printer and office automation software **a complete business computer breaking a new price barrier.**

In 1981 Exidy Systems introduced Multi-Net 80, the first

multi-processor, multi-user, multi-tasking computer system with MP/M™, CP/NET™ and CP/NOS™ for the serious small business. Once again the Multi-Net is an extension of the same Sorcerer® Computer purchased in 1978 or 1980. Your 'getting started' computer becomes your 'getting serious' computer in a multi-user, multi-task environment. **Networking becomes a reality with Exidy Systems, with our competition it's a twinkle in their eye.**

By adding Multi-Net 80 capacity to your stand alone computer system you add a minimum of 35 megabytes of Winchester storage and true 16 user capability because each user has their very own CP/M compatible Z80 microcomputer. That's true upward compatibility in both hardware and software from the company that delivers Innovation in Microcomputing™

EXIDY SYSTEMS

MP/M, CP/NET and CP/NOS, are trademarks of Digital Research, Inc.

1234 Elko Drive
Sunnyvale, California 94086
(408) 734-9831

What do we say about competition? We welcome it!!!

I N N O V A T I O N   I N   M I C R O C O M P U T I N G ™

A New Entry, continued...

An internal speaker is standard as is an 8-ohm output on the back panel.

Ports

There are several standard ports on the EC 8800: three RS-232 ports and a 50-pin connector to an internal bus. These make

Table 1.

Prices	
EC 8800 128K computer	$3295
MA 8650 RGB color Monitor (14-inch)	775
MA 8651 monochrome Monitor (12-inch)	395
Printers	
MA 8630 80-column dot-matrix	995
MA 8635 132-column dot-matrix	1595
MA 8640 132-column letter quality	2995

the attachment of peripherals relatively simple. The RS-232 ports have programmable baud rates and protocols. One is a port for communications, and the other two are for attaching peripherals. There is also an AC accessory socket and an audio output jack for headphones and two connectors for joysticks or game paddles.

Peripherals and Future Plans

At the moment, Monroe offers a bit-pad type of graphics tablet and an acoustic

modem as peripherals besides the three printers listed in the price list.

Future plans include a hard disk and cluster controller that will allow EC 8800s to communicate with each other and to share common peripheral resources such as hard disks and printers. The dot matrix

printers have the capability (in PROMs) to handle text and graphics dumps of the screen and of disk files to the printer in black and white. This includes the capability of printing the lo-res graphics characters mentioned above.

Software

Software for the computer includes a multitasking operating system, ISAM file-management, 32K Basic, Monroe Pascal and Pilot. CP/M will also be available.

Application Software

Monroe's Education Center has developed a group of application packages in various areas of education to meet the needs of students, educators, and administrators. While much of this software is in the production stages, it represents a carefully planned approach to using the computer to its full capacity in schools.

Among the packages to be provided are a data disk for keeping student records and assignments, statistical and graphing utilities, a graphing-skill development series, language-arts software on the structure of English for reading programs, physics, data management in the science lab, business education software for accounting and word processing, and communications software for networking and timesharing.

Educational, government, national account, and other quantity discounts are available. All discounts and prices are uniform throughout the country.

Printers come standard with tractor feed and friction feed, as well as with PROMs to map graphics from text, lo-res, and hi-res screen onto the printer.

The EC 8800 will be sold and serviced through Monroe Branch offices.

Monroe Systems for Business, The American Rd., Morris Plains, N.J. 07950. (800)522-4305. □

CIRCLE 354 ON READER SERVICE CARD—

---

7 Industry Leaders Offer Their Views

The Future of Personal Computers

How will personal computers change our lives in the future? How will the equipment and its applications evolve in the coming years? What roles will personal computers have in society?

The world's leading authorities on personal computers provided some insightful answers at The Boston Computer Society second annual Forum on the Future of Personal Computers, October 15, 1981.

• **Philip D. Estridge.** Director, Entry Systems Business, International Business Machines, Boca Raton, Florida. Mr. Estridge—the creator of IBM's new personal computer—looked into the near future and "The Next Steps for Personal Computers."

• **H.E. James Finke.** President, Commodore International, Ltd., Norristown, Pennsylvania. Mr. Finke gave his perspective on the explosive growth of microcomputers with "The Mass Market Micro: The Future Ain't What It Used to Be."

• **William H. Gates.** President, Microsoft, Bellevue, Washington. Mr. Gates—the father of microcomputer software—provided an inside look at "Things to Come in Personal Computer Software."

• **A.C. (Mike) Markkula.** President, Apple Computer Inc., Cupertino, California. Mr. Markkula examined forthcoming breakthroughs in personal computer technology in his talk "Making Computers Easier to Use: Trends in the User Interface."

• **Peter Rosenthal.** Marketing Manager, Atari Computer Division, Sunnyvale, California. Mr. Rosenthal offered a vision of "The Home Computer of the Future" and its impact on our homes.

• **Jon Shirley.** Vice President, Radio Shack Computer Merchandising, Fort Worth, Texas. Mr. Shirley explored the business applications of future computers with "Personal Computers in the Office of the Future."
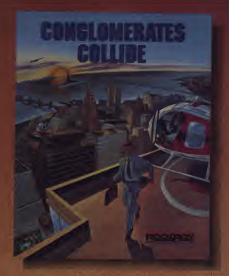
• **Nigel Searle.** Vice President, Sinclair Research Ltd., Cambridge, England. Mr. Searle considered the impact of personal computers on consumers in his talk "The Consumer Marketplace for Future Personal Computers."

Moderated by Jonathan Rotenberg, President, The Boston Computer Society.

All seven presentations along with questions and answers are available on two C-90 tape cassettes (2-1/2 hours) for $25 postpaid. If you would also like to subscribe to *Small Business Computers*, add $12 to your order ($37 total). Or, to subscribe to *Creative Computing*, add $20 ($45 total). Send payment or credit card number and expiration date (Visa, MasterCard, American Express) to the address below or call our toll-free number.

creative computing

39 E. Hanover Avenue
Morris Plains, NJ 07950
Toll-free **800-631-8112**
In NJ 201-540-0445

Lead your own company in a game of corporate survival to become "Chairman of the Board".

Conglomerates Collide™ as you battle Multi-National corporations for world dominance. Land atop your own corporate headquarters and prepare to acquire growth companies, handle bankers and deal in world markets to build the value of your company. Pay close attention to rapidly changing interest rates, earnings and the P/E ratio. Global news events that effect trade conditions will be reported to World HQ and challenge even the most clever of Presidents. At Decision Central you are on-line to 5 corporate data banks for ready access to vital information.

Each player's progress is instantly charted with RockRoy's exclusive color line performance graph (illustrated). Easy to learn rules and 3 different game play options: solitaire, 2-4 players, or mutiple computer opponents. For high scoring "Chairmen of the Board" each game disk comes with 3 entries to the RockRoy prize competition.

For your Apple* (48K. Disk with Applesoft in ROM). $39.95 includes shipping and handling. Visit your local dealer today. To order by mail send your check to RockRoy. VISA and M/C holders order by calling toll-free 800-528-2361. 15 day money back guarantee.

ROCKROY INC.

7721 E. Gray Road
Suite 103
Scottsdale, Arizona 85260
Toll-Free 800-528-2361

CIRCLE 221 ON READER SERVICE CARD

Apple is a registered trademark of Apple Computer, Inc.

Conglomerates Collide is a trademark of RockRoy, Inc.

CRYSTALWARE

THE FINEST IN
FANTASY GAME SOFTWARE

ATARI

TRS-80

APPLE

★ ★ ★ ADVENTURE GAMES ★ ★ ★

1. THE HOUSE OF USHER

2. FANTASYLAND 2041 A.D.

3. GLAND CASTLE

4. BENEATH THE PYRAMIDS

★ ★ ★ SPACE GAMES ★ ★ ★

5. GALACTIC QUEST

6. SANDS OF MARS

★ ★ ★ WAR GAMES ★ ★ ★

7. SUMER WARS

8. WATERLOO II

★ ★ ★ ARCADIA ★ ★ ★

9. LASER WARS

10. LITTLE CRYSTAL

11. FORTUNE MASTER

13. ADVENTURE PACK
14. THE WARRIOR

15. SPACEOUT
16. DROIDS

Let the Games Begin

David Lubar

The first of many screens in Threshold.

Apple games are proliferating at an astounding rate. The stack seems to grow daily, with a large selection to dazzle consumers. While it is impossible to cover every program being released, the following gives a good idea of what is available for the game-hungry Apple owner.

Coming on Strong

On-Line Systems is alive and well, which is good news for game lovers. The best of their latest releases is *Threshold*. Given only a casual glance, the game seems to be nothing more than another shoot-em-up space battle. The player has a ship at the bottom and fires at enemy creatures above. The creatures are birds that flap their wings and fly evasive formations while shooting at the player. Get through the birds and another set of enemies appears. They move differently. The game begins to shine. A third formation appears, then a fourth. If the player hasn't lost all five of his ships at this point, he gets more fuel from the mother ship. New attackers appear. There are many, and it is not likely that anyone will see all of them.

Not only are there a variety of enemies, but each group exhibits substantial differences. They move in different ways, combine in different patterns, and attack differently. Many of the shapes are internally animated. There are wheels that spin, ships that tilt sideways into slim profiles, and objects that twirl like falling maple leaves.

Beyond this, there is still more to *Threshold*. The player can fire rapidly, but his laser tends to heat up. If it overheats, he will be unable to fire until it cools. If fuel runs out before a set of four screens is destroyed, the player loses the game. Finally, the player has a special option that can be used once per ship. By hitting a key, the enemy objects are slowed down for a while,

giving the player a brief advantage. Either paddles or keyboard controls can be used in this multi-faceted duel.

The animation in *Threshold* is superb, the game is challenging, and it is unlikely that anyone will tire of playing while the mystery of future screens lies ahead.

Crossfire is another arcade game from On-Line. Using keys, the player moves around a grid of squares, avoiding the

In Crossfire, there is no safe place to hide.

shots of creatures that also move through the grid. The player can fire back, though his bullets are limited. The key controls are a bit tough to master. One set of four keys moves the player, another set is used for shooting. Rather than using a combination such as I,J,K, and M, the programmer chose I,J,K, and L, with K for down. Having the down key between left and right, rather than below it, is a bit confusing at first. Once the player has mastered the controls, he can concentrate on wiping out the enemy. His reward is another set of attackers, and a smaller supply of bullets. *Crossfire*, like *Threshold*, contains good animation, quick

response to controls, and a challenge to the player.

Good Bet

Draw Poker from Softape has some of the finest graphics ever done for the Apple. Though the game has been around for a while, it has that quality touch associated with the newest software. The program consists of five card draw for one player against the computer. While a two-player version of poker isn't quite as thrilling as a contest among five or six players, the graphics are so well done and entertaining that the game deserves a look. For each hand, the cards are spread, cut, assembled and shuffled. Each action, except for shuffling, is displayed graphically with a beautiful set of cards. The programmer did a really smooth job. When a bet is made, the chips appear on the screen. When the player or computer wins a hand, the chips are slid to the winner's side of the table.

The one weakness is in the betting system. The only unit of currency is the five dollar chip. Each bet and raise must be five dollars, no more or less. This allows less flexibility

A hand from Draw Poker.

in strategy than when the player has a range to work with. While *Draw Poker* doesn't take the place of four friends and a six pack, it will give the poker addict a willing opponent, and also provide a great way to show off the power of Apple graphics.

Follow the Bouncing Ball

Datamost gives players a dose of pulse-driving action in the guise of *Thief*. The player must make his way through a series of rooms, avoiding such dangers as collision with walls, fire from enemy robots, and an indestructible smiling bouncing ball. In defense, the player has a gun. Control is through a joystick. Moving the stick moves the player. Holding the button while moving the stick points the gun. If the button is released while the stick is held to any side or corner, a bullet fires. A self-centering joystick is highly recommended for this one.

At the start, the player is in a green room, and the robots don't shoot. After a

Slow humanoids don't last long in Thief.

certain score, the rooms become blue and the robots fire back. Later, the room becomes orange and the robots fire more frequently. Some of us have even had a brief peek at violet walls before being blown away. Unlike the arcade version, the robots in *Thief* can't be tricked into colliding

with walls. But they can be destroyed by the bouncing ball. This is a game that can't be played just once. No one here has been able walk away from it without "just one more try." Since its arrival, *Thief* has stolen a great deal our time.

Two Can Play as Cheaply as One

In *Star Thief* from Cavalier, the thieves are the enemy. They are trying to steal power pods that the player must guard.

Two players can work together in Star Thief.

"...I said he's programmed to return serves... not drop shots..."

The player, using a paddle, can shoot and fly. He has an unlimited number of ships, but a limited supply of pods. The strength of the game is that it can be played by two people. If they cooperate, working together to guard the pods, the game can last a long time. Even when played solo, the game is good. The thieves aren't hard to shoot, but they keep coming. And whenever the player is hit, he must wait a few seconds for a new ship.

Controlling the ship with a single paddle is not difficult. The paddle rotates the ship and the button, if held, supplies thrust. A jab of the button produces a burst of missiles. This allows players to concentrate on the game without worrying about the keyboard. *Star Thief* is a fine addition to the small supply of two-player cooperative games for the Apple.

Uplifting Experience

3-D Skiing from Continental Software pairs Slalom and Ski Jump, making a fun package for sports fans. The jump is presented as a side view. From one to four players can compete in up to 99 jumps. Each jump starts with a press of the paddle button which gets the skier moving. Another button press at the right time launches him from the end of the ramp.

Once in the air, the lean of the skier is controlled by the paddle. A proper lean stretches out the jump. If the player leans too far, he loses balance. This can result in a spill that is almost as violent as the famous "agony of defeat" segment from *Wide World of Sports*. The player must also straighten out to land. If he doesn't, he finishes the ride sitting down. While a good technique can be developed for jumping, it is not so easy to master that it becomes automatic.

Slalom offers three courses, three levels of difficulty, and short and long versions of each course. The game gives a view of the back of the skier, looking down hill. The object is to ski through all the flags without hitting any or taking a spill. A paddle controls the direction the skis are pointed. If they are straight, the player

moves forward, picking up speed. Turning the skis results in the skier turning in that direction. Turning also cuts down speed. Thus, the player who cuts the flags at the narrowest angle will get the best time, though he also runs the greatest risk of hitting a flag.

Ski Jump gives a new meaning to fear of flying.

The only weak point of Slalom is that the player cannot change courses or levels without rebooting the disk. This option would have been a nice addition. Except for this, *3-D Skiing* is fun to play and fun to watch.

Sudden Terror

Red Alert from Broderbund is a fast-paced, frantic game that has the player defending the bottom of the screen from a mass of attackers. Instead of just defending the standard ship or base, the player protects a series of installations, including radar, missiles, and a shield repair system. Above these facilities is the shield and above the shield are the attackers. They rain down missiles that eat away the shield. The player moves a crosshair, firing simultaneously from two guns at either side of the screen. The shots burst out in a circular pattern, destroying anything that enters the perimeter of fire. Once both guns have been destroyed, the game is over. The player can launch a special missile that destroys everything on the screen. He gets a new missile every 2500 points, but if the missile launcher is destroyed, no more missiles can be earned. Radar increases the spread

of the player's fire. The repair unit mends the shield at 10,000 points, and again at higher values. Once any of these specialized units is destroyed, it is gone for good unless

The player's shield is quickly destroyed in Red Alert.

the player is lucky enough to shoot one of the rare supply ships that zips across the screen.

The standard attack comes from swooping, dodging creatures that drop small missiles. Occasionally, a larger creature, looking somewhat like an Apple from *Apple Panic*, comes along and drops a cluster of larger missiles. These really do a job on the shield.

The game demands total concentration from start to finish. There is no chance to relax for a second. Some will find it too fast paced, but those who have mastered its predecessors will find a real challenge in *Red Alert*.

Words for the Wise

Cross Clues from SRA is an original word game designed for the computer. Two players take turns trying to guess words in a crossword-style grid. On each turn, a player first tries for a complete word. If he is correct, he gets another guess. If the word is incorrect, any letters in the correct position are kept, and any correct consonants are also displayed wherever else they appear on the grid. Next, the player gets to select a single consonant. This is also displayed wherever

creative computing

SOFTWARE PROFILE

Name: Cross Clues

Type: Word game

System: 48K Apple, Disk Drive

Format: Disk

Language: Machine Language

Summary: An original word game for two players

Price: $29.95

Manufacturer:
Science Research Associates
155 N. Wacker Dr.
Chicago, IL 60638

it occurs in the grid. Points are scored for each occurrence of a letter. Each turn is also restricted by a time limit, and good players have their time cut in half.
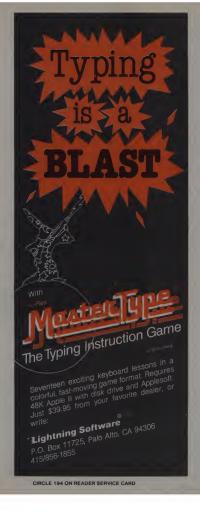

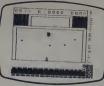*A game of Cross Clues in progress.*

Though the puzzles contain a fair number of short words, the vocabulary is sophisticated, and even puzzle experts will find the game to be a test of their skills. The disk contains 50 puzzles. An option for creating new puzzles would have been thoughtful. As is, owners of the game will have to wait and see if SRA will be releasing new puzzle disks.

Call in the Bouncer

While I try, in the interest of avoiding conflicts, to refrain from reviewing many programs from Creative Computing Software, there's a new game that is just too good to keep quiet about. *Blisterball* starts with the traditional ship at the bottom of the screen, but takes off from there in a different direction. High above, in an enclosed area, balls bounce. When the game starts, one ball drops, bouncing off the floor and walls. The player must avoid being hit, and try to shoot the ball. If he succeeds, two balls drop. After this come three, then four, and finally five. If the player is still alive, he gets to try for five bonus balls which drop one at a time. These balls are worth ten times as much as the others, but they fall faster and don't bounce. In the next round, the balls are

As one ball dies the others keep bouncing in Blisterball.

worth more, but they don't bounce as high. With each round, they lose elasticity. In each bonus round, the balls fill more quickly, requiring fast reflexes and a good aim. When the balls get too low, the walls begin to close in.

The game has two skill levels, and options for play by one person, two playing as a team, or two playing in competition. Almost everyone who has tried the game, including arcade addicts, has been quickly destroyed the first few times. A ship can take three hits before being knocked into the gutter. Those hits come pretty quickly when dealing with the multiple bouncing balls. On the weak side, the program doesn't save the high score to disk. Aside from this, *Blisterball* is a fine arcade game that offers good play value.

---

creative computing
SOFTWARE PROFILE

Name: Blisterball and Mad Bomber

Type: Arcade game

System: 48K Apple. Disk Drive. Paddles

Format: Disk

Language: Machine Language

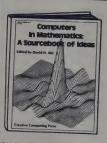**Summary:** A new twist in action games

Price: $24.95

Manufacturer:
Creative Computing Software
39 East Hanover Ave.
Morris Plains, NJ 07950

The package also contains *Mad Bomber*, a game where one or two players attempt to shoot bombs dropping from overhead racks. The racks start out empty, then quickly begin to fill. Whenever a rack has four bombs, it drops one. The player has a limited amount of ammunition, but can reload by moving all the way to the side of the screen. If a bomb hits a player, he loses ten rounds of ammunition. When ten bombs hit the ground, the game is over. Though not as strong as *Blisterball*, *Mad Bomber* is a lot of fun in the two-player cooperative mode, and very challenging as a solo game. □

Sourcebook of Ideas

Many mathematics ideas can be better illustrated with a computer than with a text book.

Consider Baseball cards. If there are 50 cards in a set, how many packs of bubble gum must be purchased to obtain a complete set of players? Many students will guess over 1 million packs yet on average it's only 329.

The formula to solve this problem is not easy. The computer simulation is. Yet you as a teacher probably don't have time to devise programs to illustrate concepts like this.

Between grades 1 and 12 there are 142 mathematical concepts in which the computer can play an important role. Things like arithmetic practice, X-Y coordinates, proving geometric theorems, probability, compounding and computation of pi by inscribed polygons.

Endorsed by NCTM

The National Council of Teachers of Mathematics has strongly endorsed the use of computers in the classroom. Unfortunately most textbooks have not yet responded to this endorsement and do not include programs or computer teaching techniques. You probably don't have the time to develop all these ideas either. What to do?

For the past six years, *Creative Computing* magazine has been running two or three articles per issue written by math teachers. These are classroom proven, tested ideas complete with flowcharts, programs and sample runs.

Teachers have been ordering back issues with those applications for years. However,

many of these issues are now sold out or in very short supply.

So we took the most popular 134 articles and applications and reprinted them in a giant 224-page book called *Computers in Mathematics: A Sourcebook of Ideas.*

Ready-to-use-material

This book contains pragmatic, ready to use, classroom tested ideas on everything from simply binary counting to advanced techniques like multiple regression analysis and differential equations.

The book includes many activities that don't require a computer. And if you're considering expanding your computer facilities, you'll find a section on how to select a computer complete with an invaluable microcomputer comparison chart.

Another section presents over 250 problems, puzzles, and programming ideas, more than are found in most "problem collection" books.

Computers in Mathematics: A Sourcebook of Ideas is edited by David Ahl, one of the pioneers in computer education and the founder of *Creative Computing.*

The book is not cheap. It costs $15.95. However if you were to order just half of the back issues from which articles were drawn, they would cost you more than $30.

Satisfaction Guaranteed

If you are teaching mathematics in any grade between 1 and 12, we're convinced you'll find this book of tremendous value. If, after receiving it and using it for 30 days you do not agree, you may return it for a full refund plus your return postage.

To order, send your check for $15.95 plus $1.00 postage and handling to Creative Computing Press, Morris Plains, NJ 07950. Visa, MasterCard, and American Express orders may be called in toll-free to 800-631-8112 (in NJ 201-540-0445). School purchase orders should add an additional $1.00 billing fee for a total of $17.95.

Don't put it off. Order this valuable sourcebook today.

creative computing

Morris Plains, NJ 07950
Toll-free **800-631-8112**
(In NJ 201-540-0445)

65

Secret Storm

David Lubar

In contrast to certain areas such as theater and cinema, most reviewers of software really don't enjoy giving a bad review. There is no thrill in shooting something down, and many marginal products are ignored in the hopes that they will die a quiet death. But when a product is highly touted, promises a lot, costs a bundle, and doesn't deliver, the public deserves fair warning. This, unfortunately, seems to be the case with *Hi-Res Secrets*.

The concept itself is admirable; gather a complete set of routines for animation and other areas of graphics, add tutorial programs and documentation, and demystify the art of programming. There is definitely a dearth of information on

Apple graphics, not through any conspiracy on the part of "those who know," but mostly because programmers just don't have the time or inclination to write articles. The situation is slowly changing, with good information appearing in scattered fashion among many magazines, but novice programmers are hungry for information and don't want to wait. Thus there is an understandable appeal to be found in something like *Hi-Res Secrets*.

The package is massive. For $125 the customer gets four disks loaded with programs and a notebook with 263 pages. One of the problems is that the notebook really does seem to be a notebook. It is organized into chapters, but the content and flow are very difficult for anyone other than the author to follow. The book opens with a section on shapes. The introductory portions cram in so many concepts, and make so many references to programs on the disk and other portions of the manual, that it is easy to get lost. The portion on vector shapes (the shapes used in Applesoft shape tables), recommends that the reader see the Applesoft programming manual for details, and buy *Super Shape Draw and Animate Package* from Avant-Garde to create vector shapes.

There is no real coverage of one area most novices have questions about; namely, how to use shape tables in machine language. There is a program on the package that scans a portion of the screen and converts it to a shape table. As a result, a shape produced in this method will not scale correctly, but will have staggered edges

when drawn at greater scales.

This is indicative of another problem found throughout *Hi-Res Secrets*. Rather than provide a few high-quality routines, the author has attempted to provide programs that cover every possible aspect of graphics. In many cases, the result is a mediocre program. The author just didn't take enough time. In one program, a bad input causes the menu to overprint part of itself, totally destroying the ability to read what is on the screen.

At least one program, instead of containing a save subroutine for shapes, prompts the user to hit reset and then save his shape. A good utility program would never force the user to input the series of parameters required for a binary save.

A close look at the assembly listings in the book reveals other problems. The code is often not clean or well thought out. For example, the author frequently clears the carry flag before an increment or comparison, even though these operations will clear or set the carry properly. He also uses jumps in many cases where, with a bit of work, relative branches could have been used. This lengthens the code and prevents easy relocation.

The author also has a tendency to make incorrect suppositions and gloss over areas he doesn't understand. Referring to color fill routines, and to Ken Williams's hi-res Adventures, he says: "I believe that they may be using their new compiler to speed up these programs." Considering that the Adventures were on the market a year before the compiler was written, this is unlikely. On the topic of flicker in animation,

2 FROM HORIZON

Fathoms 40™

Time: June, 1943.
Place: North Atlantic.
German Wolf Packs have been driven from their hunting grounds by Allied escort vessels and massive air attacks. But unknown to the Allies a Nazi super sub completed well ahead of schedule: The Mark XXI. Der Führer has demanded total destruction of enemy merchant shipping — a vital link for continued success of the Allied war effort. Will you, as one of the Kriegsmarine's remaining ace skippers, alter history and save the Third Reich from defeat?

Der Führer will not be disappointed!

FATHOMS 40 $39.95 ppd. Foreign, FPO and APO add 10% for shipping. Please specify Apple or Atari when ordering.

CLOAK & DAGGER

In the 1930's dramas were slowly unfolding in capital cities throughout the world. From dimly lit hotel rooms to highly sophisticated Intelligence Headquarters, men and women were playing a game of cat and mouse for scraps of information essential to the security of their country. CLOAK & DAGGER pits you against computer or other players as head of an Intelligence Agency trying desperately to stay one step ahead of its rivals. Moving agents from capital to capital, bribing information and offering bribes are fundamental parts of each player's turn and your survival!

CLOAK & DAGGER $39.95 ppd. Foreign, FPO and APO add 10% for shipping. Please specify Apple or Atari when ordering.

For Apple II, DOS 3.3 / Atari 400, 800 with disk drive (48K with disk drive. Joysticks required (Games on Dysan diskettes)
Visa & Mastercard accepted. Contact your local dealer for full details.
Dealer Inquiries Welcome.

HORIZON SIMULATIONS

561 Crater Lake Highway, White City, Oregon 97503
(503) 826-6640

CIRCLE 183 ON READER SERVICE CARD

he says: "This phenomenon relates, it would seem, to certain mysterious timing/frequency factors within the computer and how they mesh/fail to harmonize with the physiology of the human eye, relative to scanning frequencies or 'critical flicker,' a term from freshman psychology classes. If anyone can pin down the phenomenon better than the above, please let me know." The reader would have been much better served if the author had taken the time to research and explain the concepts of scanning and vertical blanks rather than cover the topic in this manner.

The package does cover many areas of graphics, including block shapes, color fills, and two-page animation. However, the presentation is often confusing. A determined person could possibly learn what he needs here, but the process would be tedious.

When the package first arrived, I was rather thrilled. Based on the ads, I thought it would present clear material on all aspects of graphics. This type of product is badly needed, especially for novice programmers who have the skill but lack some of the essential information required for doing sophisticated graphics. After a few hours, I began to suspect that it had a few weak points. Eventually, I reached the conclusion that *Hi-Res Secrets* was just too poorly presented to have, much tutorial value.

Still, I didn't want to condemn a program without getting a second opinion. Routines that appeal to one person might not appeal to another. And, having seen many programs that do graphics, and having written a fair amount of code myself, I realized that it is possible to become somewhat jaded and lose perspective. Basically, I didn't want my familiarity with graphics to

bias my opinion toward this package. So I showed it to a young Apple owner who fit into the novice category. He found nothing of value in the graphics areas, though he did like the sound subroutines included in the package. At best, he was left with the vague suspicion that the programs were possibly good, but there was no way he could make sense out of them.

Two more people looked at the package. Both are experienced programmers. Neither could find anything positive to say. I was tempted to just forget about the package. But the thought of someone paying $125 for *Hi-Res Secrets* without fair warning convinced me that the review had to be written. Perhaps other reviewers will find something positive to say. I tried. I really looked carefully at this one, hoping to find some value in it. I saw nothing to recommend.

A Reply from Don Fudge, author of *Hi-Res Secrets*

Since *Hi-Res Secrets* has received a fabulous reception from distributors, stores, novice programmers, computer engineers, hobbyists, and professional programmers, this review came as a great surprise, inexplicable and mysterious...at first. Where is this reviewer coming from with "warning" and "slow death" bologna? Since our initial reaction, we at Avant-Garde Creations have deciphered the mystery underlying this unrealistic hatchet-job. More on that later.

Background: After *Superfont, Super Draw & Write, Instant Graphics,* and *Action Sounds & Hi-Res Scrolling* received good press, people began asking us for decent shape-drawing utilities. We created *Super Shape Draw & Animate.* Since this was the first decent shape utility that had diagonal plotting, no plot-delay, and no limit on move-ups, we developed a following of people interested in serious graphics programming. (I drew all twelve 300-500-byte monsters for *Chambers of Xenobia* (adventure) in an easy afternoon's sitting

using *Super Shape Draw*.)

People began asking us for comprehensive graphics software, animation & sound routines, fonts, and color-fill. People were hoping we would invent fast color-fill for adventure games (because many adventures color scenes too slowly).

At the same time my own plans went toward business graphics. One feature of my system was to be Map Charts. I didn't have enough graphics knowledge at the time to carry out my plans (super-fast color-

Bit Pit

Chas Andres

fill routines and other technical graphics problems had to be solved).

The result of many months of intensive research in *all* areas of graphics, color, shapes, animation, etc., was *Hi-Res Secrets*. For business, *Ultra Plot* evolved (with its Map Charts plus all the usual business graphs and charts).

When the *Hi-Res Secrets* manual and programs were complete, we began marketing. All major distributors jumped on *Hi-Res Secrets*, and so did a following (both old and new) of graphics enthusiasts. Calls pour in, the response is great and still accelerating. Comments from professionals include: Stanley Crane, author of DB Master: "*Hi-Res Secrets* makes it possible for a programmer with a good idea to write a sophisticated game that will compete with anything on the market in terms of graphics, shapes, sounds, animation and color-fill. How can they sell this for only $125 and not charge a fee for those who might use the routines commercially???!!!; Doug Carlston, Broderbund Software, *Softalk* columnist states: "A giant step for computerkind! The most thorough book ever done on Apple Graphics: A complete 263-page course in Apple Hi-Res; a *must* for every serious Apple programmer." Mitch Kapor (author Visiplot), Ken Williams (On-Line Systems), Tom Burns and Roger Wagner (*Softalk* columnist) of Southwestern Data Systems all solicited and received copies of *Hi-Res Secrets*.

Another fine programmer, Cary W. Bradley, had this to say (review, *Softside*, Feb. 1982): "It's the most complete compilation of Apple graphics information on the market...well worth the price...the shape creation/animation groups of programs constitutes a thorough self-study course in this area of Apple programming, and may well be worth the total package price by itself...color-filling routines are the fastest I've seen...machine language routines are thoroughly documented...represents a significant contribution...absolutely packed with valuable information...tremendous volume of knowledge...techniques and programs will never lose their usefulness...in a class by itself...bound to increase the ranks of good Apple graphics programmers by a substantial margin."

In short, there is no doubt about the value and usefulness of *Hi-Res Secrets*.

Mr. Lubar's review reveals that he was probably too busy to spend enough time with *Hi-Res Secrets* to get intimately familiar with it, learn from or enjoy it. We sympathize with his position, but those who do take the time are learning lots, enjoying it, and no longer are frustrated due to the absence of Apple graphics source materials, routines, etc. *Hi-Res Secrets* is not a package that will "do it for you." It's a tool to help you learn how to do it yourself. To learn something as comprehensive as this package offers, it takes time, patience and intention.

Frustrated that a quick scan cannot bear much fruit with such a package, Mr. Lubar decided to "warn" the public. Good journalism requires that reviews *reflect the package, not the reviewer*. Unfortunately, the review says very little about what *Hi-Res Secrets* is about, but quite a lot about Mr. Lubar. *Creative Computing* is a great magazine, and most of its reviews are fine, but...

It is *absolutely false* that "there is no real coverage of how to use shape tables in machine language." There is plenty of such coverage.

It appears the reviewer has hardly even scanned the manual, much less read it. Use of shapes and shape tables in machine language, use of shape sequences, vector shapes, hplot shapes, block shapes, and animation of any of these in machine language is thoroughly covered. There are also many routines to scan shapes, convert shapes to other types, edit and draw shapes, create sequence and shape tables.

Next we learn that Mr. Lubar is unhappy that many shapes "won't scale correctly." Anyone familiar with graphics would know that rarely will normal shapes scale nicely. The user learns that our *Superfont* program (included) contains 52 perfect-scaling shapes, because they were specifically drawn for scaling.

I was surprised to see Mr. Lubar imply that the INC command conditions the carry bit. It does no such thing. I do agree I didn't need a few of those extraneous CLC commands...but they're harmless, and I explained that they are superfluous on page 139. I also agree that speculation on who's using what compiler on what is unnecessary. But why waste space on insignificant details. Dave? There's a huge comprehensive graphics system in your lap and you failed almost entirely to grasp what it is and what it does.

And why all the "second opinion" stuff? It seems you knew, underneath, that your review missed the mark and your instincts were telling you to reconsider. How could a "novice you showed it to" know what it was all about without time to really try it out?

You found "nothing to recommend" after scanning our package. We, and probably most of your readers, feel there's nothing to recommend about your review. However, in closing, I'd like to thank you. A few months ago you reviewed our ZES educational system as "too complex." Since then our sales of that package have quadrupled. We thank you in advance for what we predict will occur with *Hi-Res Secrets*.

An Editorial Comment

Because of the negative nature of this review, we sent a copy to Avant-Garde Creations before publishing it. Since their response is rather strongly worded and casts some aspersions on the competence of a member of our staff, we feel compelled to add a few comments about the parts with which we take strongest issue.

First, readers should remember that "solicit and receive" is quite different from "purchase and endorse." All of the people mentioned as having "solicited and received" copies of *Hi-Res Secrets* are connected with software companies or magazines and probably solicit copies of most, if not all, new graphics packages as they are released.

Second, we called two of the five people on the "solicited and received" list to ask their opinions of the package. They concurred with David's appraisal. We also called Doug Carlston who explained that Avant-Garde had sent him the package, then called on the day he received it to ask for a quote. He was only able to give them an initial impression, and the quote they offer is an exaggeration of his actual statements. We called *Softside* magazine and confirmed that they plan to publish a favorable review of the package.

Third, regarding the review of Z.E.S., just to refresh readers' memories, David's overall appraisal of the package was "complex, but versatile." The final paragraph of the review reads: "In essence, the Z.E.S. system allows teachers to create useful programs, but demands planning and effort on the part of the teacher/author."

Fourth, our "novice" was Neil Radick, a part-time employee in our Software Development Department. He spent a week working every evening with the package, and while he couldn't be expected to master the package in that time, it is not unreasonable to expect him to find something he could understand and make use of.

We could go on, but enough has been said already. We feel David Lubar is a competent reviewer—particularly in the area of Apple graphics—and we stand by his evaluation of *Hi-Res Secrets*.—EBS

The microcomputer in this unit will help match your
hi-fi system to your living room to create accurate
tonal balance and realistic sound.

dbx 20/20
Computerized
Equalizer/Analyzer

David H. Ahl

It all started innocently enough.

Betsy had just returned from Joe Sugar-
man's marketing seminar that I had attended
a year earlier. She was excitedly telling me
about the seminar and the participants.
"Oh by the way," she said, "Jerry Ruzicka
from dbx was there."

"Wow," said I, "they're the people with
the neat hi-fi stuff. Do you suppose . . ."

"I already asked him," she interrupted,
"and they'll lend us anything from their
catalog for thirty days. But I can't imagine
why you would need anything else on your
system."

"Need, schmeed," said I, "there's always
room for improvement. And anyway we
are a computer magazine and the dbx 20/20
has a computer in it. So don't we owe it to
ourselves and our readers to find out if it is
any good?"

Three weeks later the 20/20 arrived in a
gigantic carton. Fortunately the unit was
not as big as the box but at 23" wide and 6"
high it's anything but a compact compo-
nent.

Betsy was right. I hardly needed another
component to add to my hi-fi system. I
currently had a Sansui AU-719 amplifier
with about a zillion watts per channel, a
Marantz 2265B receiver, an Advent Sound
Space Control, and a KLH analog based
computer speaker control. Add to this a
turntable, cassette recorder, and reel-to-
reel tape deck and you have one corner of
the room pretty much taken over by
electronic sound equipment.

Growing Gracefully

Rarely does one walk into a store and
plunk down $3,000-4,000 for a component
hi-fi system. Generally, a system is assembled
over time. A significant new advance is
announced in tape decks and you decide,
"This is what I've been waiting for. I think

I'll get one." Or perhaps you get an end-of-
year bonus at work and decide to get a
new pair of speakers.

So it was with my system. It had been
accumulated over 24 years and, indeed,
one of the first components I purchased, a
pair of AR-2 speakers, are still in service in
the "back channel."
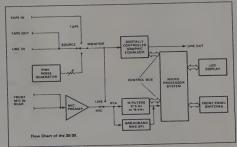
In any event, I am always astonished
when I buy a new component and find the
instruction manual assumes that this XYZ
thingie is the only additional component
in one's hi-fi system beside the amplifier
and speakers.

When I got the computer controlled
KLH speakers I found that the analog
based computer was to go between the
preamplifier and amplifier. But, so also
was the Advent Sound Space Control.
Questions: Which goes first? Does it matter?
Can they be used with each other? Two
components have tape recorder outlets;

which should be used?

Now with the addition of the dbx 20/20,
yet another component was added to the
system which goes between the preamplifier
and amplifier. Where does it go in relation
to the others? Surprise! The instruction
book with the dbx 20/20 has seven pages
devoted to block diagrams showing the
unit in conjunction with practically every
conceivable hi-fi component. Example 5
in the manual showed how to wire it with
respect to the Advent sound space control
and Example 7 showed the dbx 20/20 in
relation to the KLH speaker computer.

What is Equalization?

An equalizer is a device that can be
adjusted to change the frequency response
of a sound system. A simple example is the
treble control, which amplifies high fre-
quencies when turned up and attenuates

Flow Chart of the 20/20.

them when turned down. Tone controls of this type are adjusted to suit the taste of the listener. If the reproduction of a musical program sounds dull, increasing the treble can improve the sound. Another example of an equalizer is the loudness control found on many receivers. This control activates a specific type of equalizer that adds bass to counter the inability of human hearing to perceive low frequencies at low sound levels. A graphic equalizer provides even more controls to finely tune a sound system by compensating for less-than-ideal room acoustics or component response.

Equalization can be set on the dbx 20/20 octave-by-octave by means of front panel controls. The sounds within each octave range can be amplified or attenuated. This can be done by ear, while a record or tape is playing to produce the most pleasing sound.

However, the microcomputer in the dbx 20/20 permits the equalization process to take a giant step forward. In particular, the 20/20 not only equalizes the sound but analyzes it as well.

The process of analysis and equalization is as follows: first, a calibrated microphone is plugged into the front of the 20/20. The microphone is placed in a favorite listening location in the room. The 20/20 has a "pink noise" generator which is turned on. Pink noise is pseudo-random white noise with analog filtering for -3dB/octave roll off. What this basically means is that it provides a calibrated test signal for frequency response analysis. The "auto-equalize" button is then pressed and the microprocessor goes into operation. It analyzes the frequency response of the microphone signal to the pink noise. In turn it adjusts the equalization to obtain flat response across all ten bands (octaves). It continues the analysis until ±1dB response is obtained, or when 18 successive attempts at equalization have been made. In general, this process takes 10 to 12

seconds. The equalization curve for that location may then be stored into memory. If desired, the microphone may be moved to as many as 10 other locations around the room and the equalization curve for each location stored in memory.

The computer also has the ability to average the equalization curves for several different locations as well as computing a weighted average which, let's say, weights the equalization for the sofa location three times as heavily as that for the easy chair. This may be desirable because the sofa is used more frequently.

In addition, the computer has the ability to compensate any or all of the equalization curves for high frequency roll off. What does this mean?

In a live performance, the sound heard by the audience is made up of a combination of direct and reflected (reverberant) sound. For most concert halls, the reflected sound makes up most of the sound reaching the listener. Due to the long path length and many reflections that the reverberant sound undergoes, the high frequencies are usually attenuated more than the low frequencies. Therefore, the "effective frequency response" of a good seat in a typical concert hall is rolled off at the high frequency end.

When recordings are made, microphones are often placed quite close to the instruments, capturing mostly direct sound, without the roll off involved in the reverberant sound. Therefore, a high fidelity system adjusted for flat response will tend to reproduce high frequencies with more intensity than would be heard during a live performance. To provide a means of coping with this phenomenon, the 20/20 has a button marked "HFR" (high frequency rolloff) that gently attenuates the high frequencies at the rate of -1dB/octave above 2kHz.

The dbx 20/20 has many other nifty features, not the least of which is a visual representation of the music being displayed

over a ten-octave range. It also has the ability to analyze and equalize other individual components in the hi-fi system, although I saw little need for this when one can compensate for the system as a whole.

I should note that even after equalization, the speakers seemed to retain their individual aural character. I can't explain this, but even with complete equalization the KLH speakers sounded different from the Scott speakers.

The Bottom Line

Obviously, my living room is not the Hirsch-Houck Labs, so in some sense I have to take the manufacturer's word for it that the product is doing what it claims to do. The lab tests reported in *Popular Electronics*, October 1981 indicated that the 20/20 performed as advertised. Also, there is no question that it made a difference in the sound that I heard.

More to the point, is it worth $1500? If you have $1000 or even $1500 invested in your hi-fi system, probably not. In this case, $1500 would probably be better spent on the best pair of speakers you can find. This will have a far greater impact on the quality of the sound. Why speakers, and not an amplifier, receiver, turntable, or equalizer? Because speakers are invariably the weak link and usually can stand the most improvement.

So you say you've got a pair of AR9s and zillion watt Sansui amplifier and a Shure V15 type 4 cartridge and other components to match. In this case, I'd say the dbx 20/20 would make an ideal addition to a system and might just provide that sophisticated something extra to bring your hi-fi system closer to absolute sonic realism.

For more information contact dbx Inc., 71 Chapel St., Newton, MA 02195. □

Bit Pit

Chas Andres

The International Asteroids Tournament

Matt Mihovich

Outside, the sky was overcast and the streets of Washington, D.C., were quiet on Saturday, November 12, 1981. But inside the prestigious International Club, the *tuiy-tuiy-tu-tuiy* was deafening as 17 finalists representing the world competed in the 1981 International Atari Asteroids Championship.

The finalists came out of regional playoffs sponsored by Atari all over the world, involving some 100,000 would-be "astropros." Out of some 30,000 entrants in

Australia came Martin Lawson, 13, from Sydney where the coin-op version of Asteroids costs the equivalent of twenty cents per game.

The youngest finalist was from Italy, Pedro Estrada, 11; while the oldest, Choo King-Phui, 26, was from Singapore. The other foreign space commanders included: Shui Fan, 12, from Hong Kong; Derick Zolis, 13, from Toronto; Antoine Vareilles, 14, from France; Serge Van Lindthoudt, 16, from Belgium; Jorge Palacios Labastida, 19, from Barcelona; Gerrit Keereween, 19, from the Netherlands; Paul Staines, 14, from England; and Dirk Mueller, 18, from West Germany.

Matt Mihovich, 502 9th Street, Brooklyn, NY 11215.

The Han Solos representing the U.S. were Brian Simpson, 15, from Pasadena; Jeffrey Colby, 14, from Atlanta; Gary Wong, 18, from San Francisco; Andy Breyer, 15, from Chicago; Frank Cretella, 15, from New York City; and Doug Pye, 12, from Houston.

Doug Pye and Frank Cretella were the favorites going into the finals, because they had the two highest regional scores (138,000 and 118,740 respectively), and because they exuded calm confidence in their zapping abilities.

However, in the final competition there was one important difference from the regional playoffs. In the playoffs, it was possible to get back on the entry line and get a second chance at achieving the highest single score within two hours of continuous play. Frank got seven chances in the New York playoffs, enabling him to improve his score almost every time; but in the finals, there were no second chances.

The Final Competition

The finals consisted of two rounds of competition. Everything was on the line in the first round because only the six highest scores would advance to the second round. The scores from both rounds were then added together to determine the top three world champions.

Round one of simultaneous competition began shortly after 1:00 p.m. on Atari's video game version of Asteroids, playing selection number six, "A" difficulty. This version starts each player with four ships and awards a bonus ship for every 20,000 points scored.

Each ship has a nose gun for blasting space rocks to smithereens and alien ships into oblivion. Using joystick controls for manuevering, firing, and hyperspace escapes, Asteroids is a fast and frustrating game that quickly began to take its toll.

Though restricted to two hours of continuous play, round one was over in ten minutes. Neither Doug Pye nor Frank Cretella made it to round two. They were out-zapped by Andy Breyer, Gary Wong, Dirk Mueller, Martin Lawson, Jeffrey Colby, and Serge Van Lindthoudt.

The world champion Asteroids players: Top shooter was Andy Breyer (center), second was Gary Wong (right), and third was Dirk Mueller (left).

To aficionados of the game, Andy's high score of 75,140 and Serge's qualifying score of 32,010 in the first round were disappointing. Atari officials were hoping for a new single high score, but 75,140 turned out to be the highest single score of the day.

That score added to his second score gave Andy, a sophomore at Buffalo Grove

The world's astro-pro, Andy Breyer, 15, of Chicago.

High School in Arlington Heights, IL, a score of 142,910 and the championship. "I feel great," Andy beamed despite having played with a burnt finger, which he got from his part-time job at Burger King.

Andy was awarded an Olympic-style gold medal and a check for $5,000, which he intends to put away for his college education.

Second prize of $3,500 and a silver medal went to Gary Wong, who as a freshman at Berkeley had to miss a few midterm exams to participate in the competition. His score was 128,780.

With a score of 123,540 points, Dirk Mueller, who had just turned 18 the day before the competition, was awarded the third prize of $2,500 and a bronze medal.

The champion interplanetary shooter has been playing video games for about

All's quiet outside the International Club; but not inside where the International Asteroids Tournament is going on.

two years, and practiced about five hours a day in preparation for the finals.

Asteroids Development

Asteroids is the most popular arcade/ video game phenomenon since Space Invaders, which was last year's Atari-sponsored competition. The coin-op version is played globally for an estimated revenue of 10,000,000 quarters a week. This represents 13% of Atari's entire revenues for 1981; but it doesn't end there.

Atari developed Asteroids in-house (see sidebar), and first introduced it in early 1980 in the coin-operated arcade version. Since then, they have introduced both a video game version, of which 1,000,000 cartridges were sold in the last half of 1981, and a version for Atari personal computers.

Although Atari holds the copyright on "the video game play" and "will vigorously pursue imitators" according to Raymond E. Kassar, chairman of Atari, Inc., there are several Asteroids-like games available for personal computers. Quality Software offers *Meteoroids in Space* for Apple owners, and Acorn Software Products has *Space Rocks* and Big Five offers *Super Nova* for the TRS-80.

Asteroids has earned a place in *The Guiness Book of Records* for high-scoring,

The Origin of Asteroids

Let's give credit where credit is due—and, it's long overdue when it comes to the origin of Asteroids, the most popular coin-op and video game to date. Asteroids was developed in-house at Atari in the early spring of 1979. The person who conceived the basic idea and worked with a three-man product team to develop it was Lyle Rains, who at the time was manager of electrical engineering for the Coin-op Division.

Lyle Rains is a graduate of the University of California at Berkeley where he received an engineering degree in 1973. Shortly after graduation, he worked for a company called Kee Games, which merged with Atari about a year later. He has been with Atari ever since.

In recalling the origin of Asteroids for *Creative Computing*, Lyle said, "We had been working a number of years on an in-house project called "Cosmos," where the object of the game was for two rockets (one for each player) to try and blow each other up while maneuvering around stationary obstacles. These obstacles included planets and asteroids.

"Nothing ever came of "Cosmos" and it was never released; but several of us found that it was a lot of fun for the rocket to fly around the screen and actually shoot the obstacles.

"When Space Invaders came out, I didn't think much of it. For me it was graphically uninteresting; but it became a hit. I saw its addictive quality and tried to analyze why it was a popular game. It was during that analysis that the asteroids from the Cosmos game came back to me and the basic idea for Asteroids was born."

With a product team composed of another engineer, a programmer, and a technician, who, Lyle acknowledges "did a great job," Asteroids was quickly developed, with the coin-op version coming out in the fall of 1979.

Now at 30, Lyle is vice president of engineering for the Coin-op Division of Atari, Inc. He lives as a bachelor in Los Altos, CA, is a science fiction buff, and loves sailing.

In his spare time, he is putting together an LSI-11 digital-based computer system for his own use.

continuous play. Incidentally, the old score was recently shattered by a young man from Texas, who played some 52 hours and scored just over 25,000,000 points. Han Solo is alive and well and living in Texas.

The Winning Strategy

Obviously, for long continuous play, the real strategy is to amass bonus ships, but how does one stay alive long enough to do so? Andy's winning strategy (similar to that used by many of the other finalists) involved making a diagonal move up for a fraction of a second to a quadrant of the screen just right of center when each ship initially appeared. From that location and staying fairly stationary, Andy obliterated everything on his left and whatever came too close to obliterating him.

The competition. Note the specially contoured seats intended to minimize fatigue.

Then, he concentrated on the right side of the screen until the screen became a classic space void. This pattern was repeated at each reset, which could occur in under 60 seconds, making it necessary to have a great deal of dexterity. Being something of a purist, Andy didn't use the hyperspace maneuver unless "I was really low on remaining ships."

Holding an annual competition on its most-popular video game is Atari's winning strategy. Despite the expense of organizing regional playoffs, flying finalists and their parents to Washington, and treating them in VIP style for nearly a week, Kassar believed it was worth the effort as he noted the throng of media people, which included local and national press, a Scandinavian film team, and foreign journalists covering their national favorites.

Kassar also mentioned that Atari has acquired the rights to produce the video game and personal computer versions of *Pac-Man*, Williams's *Defender*, and *Berzerk*.

Of the three, *Pac-Man* is likely to be the focus of next year's competition, both because of its popularity (already in evidence at arcades) and because it may encourage more women participants in what has been a male-dominated competition. □

CAI: A Philosophy of Education — and a System to Match

John Herriott

Perhaps the best way to introduce this article is simply to repeat the comment I wrote on Mr. Herriott's manuscript the first time I read it: "This is the only article I've ever read that made me wish I had had computers in school." — EBS

"The computer is a powerful tool" is about the most trite remark that could be made. We were saying it in our ivory towers as long ago as the late Sixties. Looking back on it all now we realize that we were just too early.

Teachers weren't interested then. They were told by their seniors that the machine was just too expensive to waste on teaching. The real usefulness, they were told, was in record keeping and class scheduling.

Suggestions that the latter could be made unnecessary by using the computer to get rid of that most time-wasting of environments—the class—were considered nothing more than the ravings of madmen.

Perhaps both parties were afraid that the device might cause a loss of jobs, not realizing that there are so many poor teachers around that the good ones frequently go unnoticed—or join the ranks of management—or drive taxis.

Unless very serious thought is given to the whole question of the role of the

John Herriott, 143 LeMarchant Rd., St. John's, Nfld. A1C 2H3, Canada.

computer as a teaching device together with a re-vamping of the entire system to allow full use of a powerful tool in education, I'll wager that there *will* be rampant anarchy with empty schools and out-of-work teachers.

There are already thousands of school children who have lived in unusual circumstances who can attest to the efficacy of the lowly correspondence course. Children living on yachts cruising the world who have completed grades in less than six months with only three hours study a day—and completed them very well—serve as a fierce indictment of the normal school.

John Holt and Ivan Illich have each delivered detailed accounts of how inefficient the school is and have argued for changes in the system, but to little avail. I was one of that number who not only raised a voice but actually changed the system in certain parts of the world in preparation for the introduction of a vigorous teaching tool.

The teaching tool is now here, and is no longer prohibitive in price, although there is still the likelihood that the school hockey team will get new uniforms before the school will consider buying a microcomputer.

If you think that I am uncommonly hard on schools, then ponder the fact that it took almost 70 years before the lantern-slide became common in schools.

Those teachers who worry about losing their jobs because of the introduction of electronic devices probably have good reason to do so. But odds are that such teachers will not be numbered among the readers of this magazine.

There is a very strong possibility that before the end of this century, students will be receiving all of their instruction from computers, with no contact with live teachers whatsoever. It can be done, and very well.

In the early sixties I was asked to set up a training center for a small group of companies. I did so using programmed instruction entirely, even though some methods had to be specially developed, particularly where kinesthetic skills (welding, panel beating, etc.) were involved.

Using tape recorders, loopfilm projectors, slide projectors and an array of programmed teaching devices, the Center caused a stir when trainees were processed at a faster rate and with greater efficiency than the technical schools could hope to manage.

With video output, almost unlimited graphic display possibilities, voice synthesis and recognition, endless possibilities for dealing with students with differing backgrounds and linkage with a great variety of peripheral devices, not to mention the ability to link with vast stores of information, the computer leaves the average school in the dust in no time flat.

How can we change the system and what will be the respective roles of teacher and computer?

The common way to answer these questions is to try to see how the new device fits into existing patterns.

Such an approach is fraught with pitfalls. A better approach is to examine the abilities of the computer, both current and projected; examine the beneficial modes of student-teacher interaction and the detrimental ones and see in what way the computer can delete the latter to the benefit of all — particularly the student.

I guarantee that the good teacher, far from losing a job, will gain new freedom of action and avoid much of the pettiness frequently associated with his profession.

The computer can:

1. Impart information on a one-to-one basis with a high success rate when well-written and thoroughly validated programs are used.

2. Provide imbedded remedial instruction of which the student may not necessarily be aware.

3. Provide enrichment material within the program.

4. Keep accurate track of progress throughout the program, and, indeed, throughout a series of programs on varied material.

5. (Perhaps most important.) Allow the student to progress at his own rate.

6. Provide video and audio support via peripheral devices linked directly to the computer.

7. Provide a massive information retrieval base — either by direct display of the material itself or by directing the student to the appropriate medium. (Remember all those books in the Resource Center — not to mention the film-strips and movies — and the lonely librarian sitting in his office?)

All that would seem to cover just about everything a teacher can do. But no! If anyone runs away with the notion that we will see rows and rows of children sitting at dumb terminals while the teacher sits at one end of the room monitoring the progress — eavesdropping as it were — then there is something wrong with the whole affair.

Many years ago I was asked to produce an article for Industrial Training International describing the ultimate teaching machine. Something of a stir was caused when the article was published (July 1966), for I suggested that to produce a single device that could do everything, reducing the teacher's role to that of monitor, would be counter productive.

I suggested that at a time when the computer as a teaching device was far (how far I didn't realize) in the future. Regardless of how smart the device can become, I still suggest that it will remain only one of the choices available in a rich Flexible Environment (a title coined by

me in 1967 and a system developed in Ontario in 1968-9).

First, the classroom as the most common teaching mode, will disappear. There will still be room for some areas in the school to which groups of students, both large and small, will be directed, for instruction from an individual human being. There are many teachers who are superb at classroom instruction.

The sad thing is that there are also teachers who are outstanding in their ability to transmit vast quantities of information to small groups and individuals, who are never given the opportunity to do so.

The classroom as the most common teaching mode will disappear.

The teacher of industrial arts is but one example. Ten minutes are spent setting up the class. 20 minutes are spent allowing students to get some hands-on experience and ten minutes are spent cleaning up. It is estimated that of the 24 hours he spends in the shop a student spends a total of three hours a year actually working on a project.

A scenario of shop instruction might be as follows: Student presents himself at the terminal and is instructed in industrial safety in general terms. (An audio-visual, externally paced [that is, the student had no control over the speed at which the information was transmitted] that I developed in 1965 took 20 minutes and produced a result better than 95%, so let us say 20 minutes.)

He then progresses to a choice of medium in which he wishes to work (selected from a limited array). He produces his working drawings, with the help of computer graphics, of the object and is then told what techniques will be required to produce the object. He is instructed in the principles of the basic technique, say, "Using a Radial Arm Saw." Safety instruction is embedded in the program.

Upon passing the post-test, he transfers to the shop where he meets the teacher. He waits for a few moments while the teacher helps another student get set up, and then produces proof of his score and also his drawings (courtesy of graphic print-out).

He is taken to a radial arm saw (identical to the one he watched on his video screen). The teacher watches while he sets it up. He cuts a few trial scrap pieces under the gaze of the teacher. His excitement subsides as he becomes more and more confident.

He is now an hour into the program and already he has his hands on the machine.

Most shop programs don't allow the student to touch a machine for the first three or four lessons — that's three or four weeks!

He is supplied with the lumber he is going to work with; perhaps the teacher takes a few moments to talk to him about a few kinds of wood. He cuts his parts out and then takes them to the teacher for checking.

Bear in mind that there is, in an intelligent, student-oriented system, no strict timetable. The student has arranged to undertake the preliminary stages of instruction in shop for a day or two so that he can become immersed in the subject matter.

He is advised by the teacher to either continue with the next stage or undergo more practice sessions.

Assuming he continues, we then see him learn about different types of glue and their behavior on surface and edge. He might also learn about simple joints and watch while they are produced. Returning to the main shop floor he practices joints with some of the scraps he cut in his first foray. Note that he has not yet ruined anything important; he is progressing at his own rate; he is not sitting at the back of the class causing a nuisance; he has produced something — and he is having a ball.

Meanwhile, back at the teacher's desk, all is well. He is confident that each of the students in his charge, all at different stages of development, all involved in different projects (the days of 346 bread-boards, all of indifferent quality, are past!), are well instructed in the safe use of each piece of machinery (often a nightmare of responsibility). He also knows that they can see the ultimate aim of their work, that they will have the time to become involved with what they are doing, and will not have to pack up at the bidding of a bell.

He can deal with minor problems and even take time to discuss fine points. He can recommend that a student either produce a variety of pieces or a run of the same thing, continue working for an extended length of time or withdraw from the program.

"Withdraw?" you exclaim. "But he is locked in for the entire year. What he has started he must finish, regardless."

Not so. Far more damage is done to the self-esteem of the young mind by mindless insistence that it continue than was ever done by allowing a student to withdraw. He has had a far better chance than under the old system. And that business about being locked in for the whole year and his withdrawal upsetting the timetable is sheer nonsense. There is no timetable.

The student may need some counselling. Or perhaps the teacher who is responsible for this student needs to be counselled by the student in order to see what might be of most benefit to him. In any event, the next stage for the withdrawn student does not necessarily involve the computer.

There are only so many terminals on a time-sharing system. A rich school, or one that has chosen to forego a new set of parallel bars, might have a dedicated machine, in addition. It follows that it is in no way possible for each student to be sitting at a terminal all day.

It is necessary, therefore, to consider all the other interactive modes of instruction that are open to a vibrant system and which will automatically result from an intelligent consideration of computer abilities.

One of the serious problems that was generated as a result of the use of programmed instruction in lock-step school systems was that arising from individual pacing.

The quick reader, the able student, the very smart student, were all able to get through the material very quickly. This left the rest of the class working and the fast ones with nothing to do. The class was thrown into turmoil. The solution was often to give the bright students more practice in the same material, which they certainly did not need, while the remainder caught up, because the school curriculum states that only so much material will be covered during the course of a year.

Now, there is no reason at all why students (and studies have shown that this applies not just to the very bright students) should not be able to cover as much as they are able in a year. Lock-step class instruction tends to slow learning. There has been a steady decline in literacy over the past 20 years, in spite of increased educational possibilities. There has been an increase in teacher-dependency.

The first question I am asked when it is learned that I make musical instruments is: "Did you take a course or do you make them from kits?" There is a look of disbelief when I respond: "Neither." How many self-taught experts on computers are reading this article, I wonder?

I will wager that a very small percentage of you have had any formal training. Perhaps that is why there has been such an astonishing growth of interest and ability in the field. There has been no clogging of the mental drain-pipes by stuffy courses that pretend to teach but in reality tend to make the subject appear more difficult than it really is. Many of the texts on computers seem to be written by folk with lively minds who are not tied down by conventional thoughts on how to explain things.

There is no reason at all why students should not be able to cover as much as they are able in a year.

The most valuable assets a teacher possesses are those related to his total experience. No computer will ever be able to make value judgments on matters of taste or quality. It may be able to correct grammar, but some uses of the subjunctive—still a very useful verb-form—may throw it. There is the old tale of the first efforts to produce computer translation: The expression "Out of sight—out of mind" was presented for rendition into Russian, with the alarming result: "Blind idiot."

To the computer, the wallpaper in your bathroom may be every bit as good as anything Vermeer or Rembrandt did. (You may think so too, in which case I respectfully submit that you have a problem.) To a computer, Florence Foster Jenkins sings just as well as Beverly Sills (If you agree with this, too, your problem has reached alarming severity!)

Now, to be fair, teachers cannot "teach" taste either. But they can steer students in a variety of directions about the student will form conclusions about what is likeable and what is not—and know why.

In the sixties, behavioral psychologists were in the habit of regarding anything that could not be measured as nonexistent. Thus the gleam of enthusiasm in a child's eye was nothing but a figment of a teacher's imagination and was discountable. Live teachers, given the opportunity, can discern such things. They can tell whether a student has a flair for something—and can tell even if a set of marks might show otherwise.

The computer is unable to judge writing skills, thinking skills, verbal skills or manual dexterity (whether upon a musical instrument or at a lathe). Should it ever be able to, then everyone will conform in the dullest fashion. Our music will be readily consumable by all—but wanted by none!

Earlier, I mentioned that peripheral devices under computer control could provide several types of auxiliary output. There is a small problem here and in order to illustrate it, I will recount an experience from almost 20 years ago.

I was asked to develop a program to teach trainee welders. In the program I sought to instruct the students in the fundamentals of assembling the equipment and lighting of the gases to produce a neutral flame.

Now, oxy-acetylene equipment is designed so that there is no possibility of attaching the wrong part of the torch to the wrong gas container. It is important that the student watch this process, so I made an 8mm movie film and a series of slides to go with the externally-paced audio tape. Static views were presented on slides and those views in which motion was important were presented on a loop projector. Simple directions were given on the tape: "Now look at slide number 3." "Turn on the loop projector."

The program worked very well, producing scores in the 90's.

Along came the company experts. They looked at the arrangement of the hardware and came up with a device that placed the operation of the slide and loop projectors under the control of the tape. As a piece of machinery it was a wild success—but as a programmed teaching device it was a disaster. Scores began to fall.

Programs in which "all the student has to do is make a simple mark" are doomed to abject and miserable failure.

Note that the program had not changed, merely one simple factor in its presentation.

It was no longer under the partial control of the student. All he had to do was sit there and write down his responses. He became bored.

Programs in which "all the student has to do is make a simple mark" are doomed to abject and miserable failure.

It is important that the student be required to write even long sentences at times when working with CAI programs. And it is certain that the lowly pen and paper will have a long and gloriously continued role to play in education.

The alternative is an even greater reduction in the literacy level. The increased use of multiple choice exams has, I believe, contributed greatly to the drop in recent years.

The fact that most school exercises are aimless—the student writes an exercise which is examined by a teacher, marked, handed back and then discarded for evermore—is a sorry comment on our schools.

When a student has produced a fine piece of work there is every reason to include a copy of it in the school resource center. Why not teach the student how to produce CAI material?

In a project conducted in 1968-69, students used TTYs in local mode to produce tests for those students who wished to use student-produced projects. Surely the idea behind getting a student to study a subject is to allow him to learn something. If he has learned something well, then he is in possession of some useful information. Let him demonstrate that the information is useful by allowing him to produce a small text, or even a program. □

Using Microcomputer Graphics
to Train Teachers

Stanley L. Mathes

During the last two years there has been a dramatic increase in the use of microcomputers in elementary and secondary schools, but there continues to be great uncertainty about how to make the best use of this new technology.

Some educators complain about the lack of good software while others will purchase any program that looks good without consideration of the educational purpose the microcomputer is supposed to serve.

This article does not propose to offer any solutions to this dilemma, but I do urge educators to consider carefully the goals to be achieved through the use of microcomputers in the schools before they choose hardware and software.

Recently, discussion of computer use has focused on the issue of power. Will the computer control student responses in the Computer Aided Instruction mode or will the student gain control of the power of the computer by learning an appropriate programming language? I do not see this as an either/or issue, but believe there should be an appropriate balance of both types of activity, depending on the educational goals to be achieved.

What is often lacking in this discussion is the fact that few teachers have appropriate training to teach students any programming language, and, in fact, there are few people who have the necessary background to train teachers.

Too often teachers learn Basic by writing programs that make tables and charts and do repetitive computations. Did you make a table of the squares and square roots of the numbers between 1 and 20 when you learned programming?

Over the past eleven years I have watched

Stanley L. Mathes, Box 348 Ridge Rd., Scotia, NY 12302.

students from elementary grades through high school learn programming. The introduction of computer graphics has greatly increased student motivation. In high school there are often lines of students waiting to use computers when the graphics unit is finally taught at the end of the course. An elementary student will spend hours typing in a program to produce a logo with his name in it.

I believe we should reverse the sequence of units in the teaching of programming and begin by teaching graphics and end with a short unit on computation. Some might argue that students will get little

---

Few teachers have appropriate training to teach students any programming language.

---

experience with "real" programming and will lack experience with important programming techniques such as decisions, loops, and above all, structured programming.

I contend that all of the basic programming concepts can be taught by giving students experience with graphics, and at the end of the course they will be able to apply these skills to more traditional programming problems. In addition, they will be working on problems that truly interest them, rather than on the fabricated problems that pass for real computing in elementary programming courses.

Having said all this many times over the

last two years I came face to face with a test of my convictions last summer when I had to plan and teach a week long summer inservice course. One of the truisms of education is that teachers teach as they have been taught; thus, if I were to be effective in promoting my goal of having graphics as the cornerstone of programming instruction, I would have to teach these educators programming through the use of graphics.

The Course

The conditions and atmosphere for the course were ideal. We were located in an air-conditioned room of a country club overlooking the swimming pool and right next to the bar. There were more than 15 Apple computers in the room to be used by 28 educators.

My first step was to close the curtains to the windows overlooking the pool. The second thing I did was to have the teachers draw pictures on the screen using the Apple low-resolution graphics which gives 16 colors on a 40 x 40 grid. The motivational factor soon overcame the temptations around us and most of the teachers were busily working on their graphics programs.

We moved from making logos to drawing squares and lines, filling them with colors and then making lines of different colors. Excitement grew as the teachers found they were able to create moving, multicolored patterns that sometimes gave the impression that the viewer was moving quickly through a brightly colored cave.

The learners hardly noticed that they had learned such programming commands as IF...THEN, FOR...NEXT and GOSUB... RETURN, as well as simpler statements such as GOTO, PLOT, HLIN, VLIN, LET, and RND.

ALF Music Synthesizer

The ALF Apple Music Synthesizer (AMS) is an easy to use peripheral which allows you to program music into an Apple II computer using standard musical notation. The ALF kit includes the synthesizer board (plugs into any peripheral slot), exceptional quality software, and an extensive user manual.

Sophisticated Music Entry Program

Sheet music is easily entered using the Apple game paddles. The high-resolution ENTRY program features the familiar music staff with a "menu" of musical items listed beneath it (note lengths, rests, edit commands, accidentals, etc.). One game paddle moves a cursor up and down the music staff and is used to select the note pitch; the second paddle chooses from the menu items (note length, etc.) With the ALF hi-res ENTRY program, you won't have to use cryptic codes to select note parameters.

As you program sheet music with ENTRY, measure bars are inserted automatically (and note values are tied over the bar where necessary). Key signatures are also automatic—so you don't have to keep writing in every sharp or flat!

Three monophonic, individual parts can be programmed with each ALF Music Synthesizer. Two boards are required for stereo. A total of three synthesizers can be used simultaneously for a maximum of nine voices. By controling the envelope (or shape) of each voice, many different instrumental sounds can be simulated.

Eight-octave Range

The ALF Music Synthesizer has a pitch range of eight octaves—a wider range than a grand piano. The ALF can also play semitones—"blues notes" or the pitches in between the keyboard notes of a piano. (The pitch range is from 27.5 to 55,000 Hertz, well beyond the limits of human hearing.) Tuning accuracy is virtually perfect within two cents of pitch value.

Every parameter of the ENTRY program can be changed again and again during a musical piece. For example, you can make changes in key, time signature, volume, and timbre (envelope). Parts can be edited at any time, also. Notes can be added or deleted, note length can be changed, as well as pitch, volume, etc.

You can save songs on either cassette or disk, and play them back using either ENTRY or PLAY. The playback speed is adjusted with one of the game paddles, and can be varied during the playback, if you wish to change the overall tempo.

Colorful Playback Display

The ALF Music Synthesizer features a 16-color low-res graphic display during song playback. Each musical part is represented on a stylized piano "keyboard"—the intensity of the note determines the color, and the pitch is shown in relation to "middle C".

The ALF Music Synthesizer requires the use of an external audio amplifier. Stereo programming is possible with the use of two or three synthesizer boards.

The ALF software includes the ENTRY and PLAY programs, sample songs, an introduction to "envelope shaping", and demonstrations of advanced uses of the synthesizer.

With the ALF software, entry of music is easy, fast and accurate.

Nine Voices for only $198

The new ALF "AM-II" music synthesizer offers an unbeatable value for the Apple owner who is a music hobbyist. With nine voices on a single music board for $198.00, the AM-II is the most economical device for creating music with the Apple.

The AM-II uses the same excellent ENTRY and PLAY programs as the more sophisticated ALF Music Synthesizer (AMS); the same hi-res graphic display from which notes are selected with the Apple game paddles (not typed with cryptic codes). All of the conveniences of the ENTRY program apply—easy editing, playback with low-res display, ability to save songs on cassette or disk, etc.

The AM-II has **stereo output** (3 voices in left, 3 voices in the middle, 3 voices in the right).

How can the AM-II offer so much for only $198.00? The two largest differences between the AM-II and the ALF Apple Music Synthesizer (AMS) are pitch accuracy and dynamic range. The AM-II has an accurate pitch range of about six octaves. Pitch values above the treble staff become increasingly inaccurate. Also, the AM-II has a dynamic range of 28db, with 16 different volume levels, (the AMS has a dynamic range of 78db).

The AM-II is manufactured with the same high quality standards as other products from the ALF Corporation. No sacrifice has been made in reliability; the new AM-II is simply a great bargain.

Professional musicians will still want to use the original Apple Music Synthesizer (AMS) for its extended range and volume controls (the AMS has a range of 8 octaves). But for the Apple owner who is interested in music as a hobby, the AM-II is the best music peripheral value available today.

Requires: 16K Apple II or Apple II Plus, cassette or Disk II, and an external audio amplifier (all necessary patch cords are included).

AM-II ALF/Apple Synthesizer	$198.00
AMS ALF/Apple Synthesizer	248.00

To order, send payment plus $3.00 shipping and handling to Peripherals Plus, 39 E. Hanover Ave. Morris Plains, NJ 07950. Credit card customers should include card number and expiration date of Visa, MasterCard or American Express. Credit card customers may also order toll-free:

800-631-8112
(In NJ call 201-540-0445)

Peripherals Plus

39 E. Hanover Avenue, Morris Plains, NJ 07950

CIRCLE 239 ON READER SERVICE CARD

On the third day we started to develop a program to make a white dot move around on the screen in response to keyboard input from the user. The participants had already become familiar with the concept of subroutines and were able to add program modules to make a program perform additional functions. Since our goal was to make a dot move around the screen we needed to have a PLOT statement somewhere to be performed whenever an appropriate command was issued. Thus, we began with the following subroutine:

```
1000 REM PLOT THE DOT
1010 PLOT H,V
1020 RETURN
```

This program wouldn't run, but we did have the essential part of the program

Listing 1.

```
10 REM INITIALIZATION ROUTINE
25 TEXT : HOME : GR : C = 15 : B = 0 :REM SET THE GRAPHICS MODE
   AND INITIALIZE THE COLOR OF THE DOT (15 IS WHITE: 0 IS BLACK)
30 COLOR = C : H = 20 : V = 20 :GOSUB 1000
40 END
```

Listing 2.

```
100 REM ROUTINE TO MOVE THE DOT UP
110 HNW = H : VNW = V - 1 : REM SET THE COORDINATES FOR THE
    NEW DOT
120 COLOR = B : GOSUB 1000 : REM ERASE THE OLD DOT
130 H = HNW : V = VNW : COLOR = C : GOSUB 1000 : REM PLOT THE
    NEW DOT
140 RETURN
```

Listing 3.

```
40 GET A$ : REM ACCEPT KEYBOARD COMMAND
50 IF A$ = "I" THEN GOSUB 100
52 IF A$ = "J" THEN GOSUB 200
54 IF A$ = "K" THEN GOSUB 300
56 IF A$ = "M" THEN GOSUB 400
58 IF A$ = "C" THEN GOSUB 500
90 GOTO 40
```

written. All that remained was to change the values for H and V so the dot would move.

The Sketching Program

Next we decided that the dot would be white and would start near the middle of the screen. We added the lines in Listing 1.

Now we had a program that ran, although all it would do was plot a white dot in the middle of the screen. The next step was to add a routine that would make the dot move up the screen. We decided to use the I, J, K and M keys to control the motion in a manner that resembled the editing functions of the Apple.

We then had to create a routine that would move the dot up in response to an "I" entered from the keyboard. We decided to use the GET command so that there would be no need to hit Return after the command was issued. The following routine was added to accept user input:

```
40 GET A$:  REM ACCEPT KEY-
   BOARD COMMAND
50 IF A$ = "I" THEN GOSUB 100
90 GOTO 40
```

This routine accepts user input from the keyboard. If the user types "I," the program executes the routine at 100 (which is still to be written), and if any other key is pushed the program branches back to accept another input. Please note there is room for other commands, not yet written, in the lines between 50 and 90.

To get the dot to move we wanted to change the values of H and V and execute the plotting routine. In this case, since the dot was to move up, we wanted to decrease the value of V by 1. But we also had to erase the white dot that remained plotted at the new location.

Thus, we introduced two new variables HNW and VNW to represent the new position for the dot, changed the color to black to erase the dot, set H and V equal to the new values, changed the color back to white and then plotted the new dot. The routine appears in Listing 2.

The teachers were quickly able to add new conditions to the input routine to test for J, K and M, and then add the appropriate subroutines at 200, 300 and 400 to make the dot move to the left, to the right and down.

I felt content to leave the program at this point and have the participants go to work on other projects, but several ideas were suggested that turned this simple routine into a fairly advanced sketching program.

It was clear that if we changed the value for B in the program, the moving dot would leave a path of color given by B. A new feature was added to the input routine to change the color when the dot was erased, and allow the user to make a sketch. The new input routine is shown in Listing 3.

```
500 REM CHANGE THE SKETCHING COLOR
510 PRINT : PRINT : PRINT "SKETCHING COLOR (0-15)";:INPUT B
520 PRINT : PRINT : PRINT : RETURN
```

Listing 5.

```
1002 IF VNW > 39 THEN VNW = 39 : PRINT CHR$(7) : REM MAKE A BEEP
1004 IF HNW > 39 THEN HNW = 39 : PRINT CHR$(7)
1006 IF VNW < 0 THEN VNW = 0 : PRINT CHR$(7)
1008 IF HNW < 0 THEN HNW = 0 : PRINT CHR$(7)
```

Listing 6.

```
600 REM MOVE DIAGONALLY INTO THE FIRST QUADRANT (UP TO
    THE RIGHT)
610 HNW = H + 1 : VNW = V -1 : REM NEW DOT'S COORDINATES
620 COLOR = B : GOSUB 1000 : REM ERASE THE OLD WHITE DOT WITH
    THE SKETCHING COLOR
630 H = HNW : V = VNW : COLOR = C : GOSUB 1000 : REM PLOT THE
    NEW WHITE DOT
640 RETURN
```

Listing 7.

```
62 IF A$ = "S" THEN GOSUB 2000
64 IF A$ = "L" THEN GOSUB 2100

2000 REM SAVE ROUTINE
2010 D$ = CHR$(13) + CHR$ (4)
2020 PRINT : PRINT : PRINT "NAME OF THE SKETCH TO BE SAVED":
     PRINT " PRESS < RETURN > TO RETURN TO SKETCHING" : INPUT
     NA$
2030 IF NA$ ="" THEN RETURN : REM PRESS RETURN TO RETURN TO
     SKETCHING
2035 PRINT : PRINT : PRINT
2040 PRINT D$"BSAVE"NA$",A1024,L1024"
2050 RETURN
```

The routine to change the background color was added at 500. That routine is shown in Listing 4.

Two other functions were necessary. First, the program had to prevent the user from trying to plot points off the edge of the screen, and second, diagonal moves were needed to overcome the cumbersome routine needed to sketch diagonal lines.

To prevent the program from bombing when the user tried to plot off the screen the lines in Listing 5 were added to the plot routine.

The routine in Listing 5 sets the value of the horizontal or vertical component at 0 or 39 if the user attempts to go beyond the limits of the screen. In addition it warns him of his offense with a beep after each errant move.

To add the capacity for diagonal moves, new input routines had to be added along with four subroutines at 600, 700, 800 and 900. We decided to use input of 1, 2, 3, and 4 to indicate moves diagonally into the corresponding quadrants. For this we introduced the use of the ON ... GOSUB command. One new line in the input routine was able to handle all four situations:

```
60 ON VAL (A$) GOSUB 600, 700, 800,
   900
```

The program would branch to 600 if a 1 were pressed, to 700 if a 2 were pressed, etc. The subroutine to handle a diagonal move into the first quadrant is given in Listing 6. The other diagonal moves are very similar.

Similar routines were written for diagonal moves into the other quadrants. The entire program is listed at the end of the article for those who don't want to try to fill in the missing parts.

Suddenly the teachers wanted many new options to improve the program. First was the ability to save and load sketches, and second was the need for a menu so the user could press "H" and get a listing of all the available commands.

To save and load sketches the participants needed to learn that the low-resolution graphics page was stored between memory locations 1024 and 2047. Then they needed to learn how to write disk commands from a Basic program, and finally how to save a binary file. The SAVE routine is shown in Listing 7, and the LOAD routine is given in the complete listing at the end. The input routine additions are given first, followed by the save routine.

The LOAD routine is very similar except the BSAVE at line 2040 is replaced by a BLOAD at line 2140. In addition, an error handling routine prevents the program from bombing if the user names a file that is not on the disk.

The development of the menu presented several problems, because as soon as the menu was displayed the sketch on the screen was wiped out. First we tried to PEEK each memory location on page one

```
1 REM ****************
2 REM *SKETCHING PAD**
3 REM *BY STAN MATHES*
4 REM ****BOX 348****
5 REM ***RIDGE ROAD***
6 REM ***SCOTIA, NY***
7 REM ****12302*****
8 REM ****AUGUST 1981**
9 REM ****************
10 REM INITIALIZATION ROUTINE
15 HIMEM:34000: REM  LEAVE A SPACE TO SAVE THE PICTURE WHILE THE MENU OR
   THE COLOR PALETTE IS BEING DISPLAYED
25 TEXT: HOME: HTAB 13: INVERSE: PRINT "SKETCHING PAD": NORMAL
30 GOSUB 2410: REM  MENU OF COMMANDS
32 HOME: GR
34 D$=CHR$(13)+CHR$(4): REM  TO ENABLE DISK COMMANDS FROM BASIC
36 C=15: B=0: REM SET THE GRAPHICS MODE AND INITIALIZE THE COLOR OF
   THE DOT : 15 IS WHITE : 0 IS BLACK)
38 COLOR=C: H=20: V=20: GOSUB 1000
40 IF PEEK(-16384)>127 THEN
   GET A$: POKE -16368,0: GOTO 45: REM  ACCEPT KEYBOARD COMMAND
41 COLOR=C: GOSUB 1000: REM THIS ROUTINE MAKES THE DOT BLINK
42 COLOR=B: GOSUB 1000
43 GOTO 40
45 IF A$="Q" THEN
   PRINT "RUN 34 TO RESTART WITH THE SAME SKETCH": END
50 IF A$="I" THEN
   GOSUB 100
52 IF A$="J" THEN
   GOSUB 200
54 IF A$="K" THEN
   GOSUB 300
56 IF A$="M" THEN
   GOSUB 400
58 IF A$="C" THEN
   F=1: BEG=1024: EN=2047: DEST=34024: GOSUB 8000: GOSUB 500
60 IF A$="S" THEN GOSUB 600,700,800,900
62 IF A$="S" THEN
   GOSUB 2000
64 IF A$="L" THEN
   GOSUB 2100
66 IF A$="H" THEN
   F=1: BEG=1024: EN=2047: DEST=34024: GOSUB 8000: GOSUB 2400
68 IF A$="E" THEN
   GOSUB 2200
90 GOTO 40
100 REM ROUTINE TO MOVE THE DOT UP
110 HNW=H: VNW=V-1: REM SET THE COORDINATES FOR THE NEW DOT
120 COLOR=B: GOSUB 1000: REM ERASE THE OLD DOT
130 H=HNW: V=VNW: COLOR=C: GOSUB 1000: REM PLOT THE NEW DOT
140 RETURN
200 REM MOVE THE DOT TO THE LEFT
210 HNW=H-1: VNW=V: REM NEW COORDINATES
220 COLOR=B: GOSUB 1000: REM ERASE THE OLD DOT
230 H=HNW: V=VNW: COLOR=C: GOSUB 1000: REM PLOT THE NEW DOT
240 RETURN
300 REM MOVE THE DOT TO THE RIGHT
310 HNW=H+1: VNW=V: REM NEW COORDINATES
320 COLOR=B: GOSUB 1000: REM ERASE THE OLD DOT
330 H=HNW: V=VNW: COLOR=C: GOSUB 1000: REM PLOT THE NEW DOT
340 RETURN
400 REM MOVE THE DOT DOWN
410 HNW=H: VNW=V+1: REM NEW COORDINATES
420 COLOR=B: GOSUB 1000: REM ERASE THE OLD DOT
430 H=HNW: V=VNW: COLOR=C: GOSUB 1000: REM PLOT THE NEW DOT
440 RETURN
500 REM CHANGE THE SKETCHING COLOR
510 GR:
    FOR I=0 TO 31:
    COLOR=I: VLIN 0,39 AT I:
    NEXT I
520 VTAB 21: PRINT "0 1 2 3 4 5 6 7 8 9 1 1 1 1 1 1"
530 VTAB 22: HTAB 21: PRINT "0 1 2 3 4 5"
540 VTAB 24: PRINT "SKETCHING COLOR (0-15)":: INPUT B
```

```
550 PRINT: PRINT: PRINT: PRINT
560 BEG=34024: EN=35047: DEST=1024: GOSUB 8000
570 RETURN
595 PRINT: PRINT: PRINT: RETURN
600 REM MOVE DIAGONALLY INTO THE FIRST QUADRANT (UP TO THE RIGHT)
610 HNW=HNW+1: REM NEW DOT'S COORDINATES
620 COLOR=B: GOSUB 1000: REM ERASE THE OLD WHITE DOT WITH THE SKETCHING
    COLOR
630 H=HNW: V=VNW: COLOR=C: GOSUB 1000: REM PLOT THE NEW WHITE DOT
640 RETURN
700 REM MOVE DIAGONALLY INTO THE SECOND QUADRANT (UP TO THE LEFT)
710 HNW=HNW-1: VNW=V-1
720 COLOR=B: GOSUB 1000
730 H=HNW: V=VNW: COLOR=C: GOSUB 1000
740 RETURN
800 REM MOVE DIAGONALLY INTO THE THIRD QUADRANT (DOWN TO THE LEFT)
810 HNW=HNW-1: VNW=V+1
820 COLOR=B: GOSUB 1000
830 H=HNW: V=VNW: COLOR=C: GOSUB 1000
840 RETURN
900 REM MOVE DIAGONALLY INTO THE FOURTH QUADRANT (DOWN TO THE RIGHT)
910 HNW=HNW+1: VNW=V+1
920 COLOR=B: GOSUB 1000
930 H=HNW: V=VNW: COLOR=C: GOSUB 1000
940 RETURN
1000 REM PLOT THE DOT
1002 IF VNW>39 THEN
     VNW=39: PRINT CHR$(7): REM MAKE A BEEP
1004 IF HNW>39 THEN
     HNW=39: PRINT CHR$(7)
1006 IF VNW<0 THEN
     VNW=0: PRINT CHR$(7)
1008 IF HNW<0 THEN
     HNW=0: PRINT CHR$(7)
1010 PLOT H,V
1020 RETURN
2000 REM SAVE ROUTINE
2020 PRINT: PRINT: PRINT "NAME OF THE SKETCH TO BE SAVED": PRINT
     "PRESS <RETURN> TO RETURN TO SKETCHING": INPUT NA$
2030 IF NA$="" THEN
     RETURN: REM PRESS RETURN TO RETURN TO SKETCHING
2035 PRINT: PRINT: PRINT
2040 PRINT D$"BSAVE"NA$", A1024, L1024"
2050 RETURN
2100 REM LOAD ROUTINE
2105 ONERR GOTO 64
2120 PRINT: PRINT: PRINT "NAME OF THE SKETCH": INPUT NA$
2130 IF NA$="" THEN
     PRINT: PRINT: PRINT: RETURN
2140 PRINT D$"BLOAD"NA$
2145 HTAB 21: PRINT
2150 RETURN
2200 REM ROUTINE TO CLEAR SCREEN TO ANY COLOR
2210 PRINT "WHAT COLOR": INPUT B
2215 PRINT: PRINT: PRINT
2220 COLOR=B
2230 FOR I=0 TO 39:
     HLIN 0,39 AT I
2240 NEXT I
2250 COLOR=15: GOSUB 1000
2260 RETURN
2400 REM PRINT MENU OF COMMANDS
2405 TEXT: HOME
2410 PRINT "MENU: ": PRINT "(C) CHANGE SKETCHING COLOR": PRINT
     "(K) MOVE RIGHT": PRINT "(M) MOVE DOWN": PRINT "(J) MOVE LEFT": PRINT
     "(I) MOVE UP"
2420 PRINT "(S) SAVE THE SKETCH": PRINT "(L) LOAD A SKETCH": PRINT
     "(E) CLEAR THE SCREEN TO ANY COLOR": PRINT "(H) PRINT THE MENU":
     PRINT "(Q) QUIT THE PROGRAM"
2430 PRINT "      DIAGONAL MOVES"
2440 PRINT "(1) UP TO THE RIGHT": PRINT "(2) UP TO THE LEFT": PRINT
     "(3) DOWN TO THE LEFT": PRINT "(4) DOWN TO THE RIGHT"
2450 PRINT: PRINT "PRESS <RETURN> TO CONTINUE": GET A$
2460 PRINT:
     IF A$="" THEN
     RETURN
2470 GR: BEG=34024: EN=35047: DEST=1024: GOSUB 8000
2480 VTAB 22: HTAB 1
2490 RETURN
8000 REM MOVE THE PICTURE TO ABOVE HIMEM WHILE THE MENU IS PRINTED AND
     THEN MOVE IT BACK
8010 POKE 768,216: POKE 769,160: POKE 770,0: POKE 771,76: POKE 772,44:
     POKE 773,254
8020 POKE 60,BEG-INT(BEG/256)*256: POKE 61,INT(BEG/256)
8030 POKE 62,EN-INT(EN/256)*256: POKE 63,INT(EN/256)
8040 POKE 66,DEST-INT(DEST/256)*256: POKE 67,INT(DEST/256)
8050 CALL 768
8060 RETURN
```

of graphics and then move them into memory locations above 34000 and return the sketch when the user was finished with the menu. It worked but there was an interminable wait while the Basic routine made the memory moves.

I remembered that there was a fast Monitor routine to make memory moves which could be made with a CALL -468 if the Y register was loaded with a 0 first. The necessary input routine was added at line 66, the memory move put at 8000, and the menu at 2400.

Some participants were still working on enhancements to the program as I left the workshop.

As the course drew to a close several of the participants were still adding routines to improve the program. Some of these are described below:

• Erase the screen to start the sketch again.
• Use the screen erase function to fill the screen with any color.
• Present a palette to the user when he chooses the sketching color.

Some participants were still working on enhancements to the program as I left the workshop at 3:30 on a warm Friday afternoon, and I never did see the most up-to-date version of the program. I have included my own versions of the subroutines listed above. In addition, I have changed some of the routines to make the program run more smoothly and without error.

This was a very rewarding experience for me because the participants were getting involved in increasingly difficult programming techniques as a result of their own motivation to improve the performance of a graphics program.

Although this situation involved adults learning programming, I believe there would be similar or even better results with secondary or upper elementary students if they were given sufficient time.

I have tried to show that beginning programmers can learn fundamental programming concepts and write well-structured routines by working with graphics. As computer programming becomes a more important part of the curriculum I urge teachers and administrators to consider the use of graphics as a vehicle for both motivation and teaching good programming technique. □

"All the Scientists in the World Smushed into One": What Kids Think About Computers

Mary M. Humphrey, Ph.D.

What happens when you LOAD a program?

"Well, what happens...this is the mouth [pointing to the tape cassette] and the computer eats the program off the tape. Then it goes through here, [pointing to the cassette cable] like a straw, to the computer's stomach. Here's its face. [the

Mary M. Humphrey, Ph.D., Teaching Tools: Microcomputer Services, P.O. Box 50065, Palo Alto, CA 94303.

TV monitor] like a big eye, and you can look in and see what's in the stomach...Then when you say RUN, the computer's got the energy to run 'cause it ate the program. If you don't load the program, it can't run 'cause there's no energy to... You know what else computers eat? Potato chips. I hear there's lots of chips in these things."

— *Tony, Age 8.*

The above quote is one of the answers I received in a survey of children's impressions of personal computers. The 74 children in

the survey ranged from ages 7 to 12, and were in regular, junior special education, and gifted classes. All the children were familiar with personal computers, having had them in their classrooms for six weeks to seven months before the survey. Some of the older children had been learning to write their own programs.

I asked these children some simple questions: What's inside a computer? Why is it so fast? Could the computer ever be wrong? What would you like to be able to do with a computer? What about the future? The children's answers are not only entertaining but important to keep in mind in the design and use of computer-related classroom materials.

For example, there is no need to spend time and effort reducing inhibitions that children don't have, or explaining concepts that they have already figured out for themselves. However, there may be value to devoting more time and effort to meeting their expectations for what these machines can and ought to be doing.

The Personal Computer

Many adults are still intimidated by the thought of owning a personal computer. These people cling to the old stereotypes of huge, expensive machines with rows of switches and blinking lights manned by skilled technicians. The computers children use and the purposes for which they use them are far less intimidating.

The first time most children use a computer is to play a game of space invaders or to work an entertaining lesson. Children respond confidently to the prospect of

The Delux Businessman Model

Ellen 11

having a personal computer: "There should be a computer in every school, for each kid." (Alan, 11)

When told their computer time is up, the reply is usually "Just five more minutes, pleeease?"

When asked, "What would you like to be able to do with a computer?" children stress the benefits of its use:

"Everyone will be buying them and they will be used in supermarkets, banks and places like that." (Linda, 11)

"I like how it will do anything you want it to do... usually." (Nancy, 10 3/4)

"They can do all the hard jobs for us." (Satomi, 11)

"Computers will likely be one of the most useful devices to man." (Laura, 10)

Children do not view computers as cold, impersonal machines. They often demonstrate a very "personal" style of computing. Highly interactive programs which use the child's name are the most popular. Children particularly like asking the computer questions and getting its answers, rather than only answering the computer's questions. Their need for a computer to be "user-friendly" demands more than just being easy to use:

"I would like to talk with it and discuss my problems so the computer can give its opinion." (Satomi, 11)

"...have it talk with me and solve personal problems." (Jeff, 10)

"I would like to get the computer to really understand what I'm trying to do with it." (Alan, 10)

Children expect personal computers to be highly flexible in meeting the needs of the individual user. When asked to design a computer for themselves, most children do not treat the request as unusual or extravagant, and they have very specific ideas about what they want:

"There should be a computer in every school, for each kid."

"If I designed my own computer it would be about four feet tall and wide. Over 100 different buttons on 50 keys. It would have a phone hooked up to it so it could take messages, and a control so that it could turn off the lights at night, some other things would be: smoke alarm, burglar alarm, radio, clock, tape recorder, and a place to keep your tapes in back." (Jenny, 10)

"My computer is portable, it has a bigger brain, has a bigger, easier to reach keyboard... and comes with a 500-page manual." (Genevieve, 10)

"...antenna to catch TV signals, a color console to change color, a hearing aid to ask the computer what's wrong, and arms to make breakfast." (Chris, 10)

"It has a clear outer coating so people can see how it works." (Lindsay, 10)

Generally, computer-using children have grasped the idea of a personal computer quite well. Their comments reflect their expectations that the machines be useful, highly individualized and flexible. They also feel that computers ought to become more personal and user-friendly.

Wire and Brain

Children appear to be impressed with the technical complexity and efficiency of computers as machines, but are not intimidated by them. The question of what they want to be able to do with a computer is often answered, "Take it apart and put it back together again."

Most children who use computers in their classrooms have had the opportunity to load and run programs for themselves, to see the components inside a computer, and to learn about printers, light pens, game paddles and disk drives. They have become familiar with the computer as a machine.

When asked, "What's inside a computer?" none of the children replied, "I don't know," or as adults predict, "Little men." Their answers cover a range of sophistication:

"Data storage, translator, bits, bytes, batteries, and a directory." (Kenneth, 10)

"tiny light bulbs, little filaments, electricity and wires." (Genevieve, 10)

"A binary system." (Alan, 11)

"A lot of highly technical material." (Doug, 10)

"I think there's a machine inside the computer." (Brenda, 12)

"Wire and brain." (Paul, 9)

"A lot of junk that's neat." (Eddie, 10)

"...but mostly empty space." (Jeff, 10)

Young children often invent their own analogies to help them understand computer processes, like Tony's program-eating computer described above. However, by fifth and sixth grade, children are capable of offering simplified explanations of computer processes without using analogy:

"When you press a key, the electricity travels down a wire (at about the speed of light) and into the processor and into memory. From memory it goes into the processor again, then it's printed on the screen...and all that happens in a fraction of a second." (Mark, 11)

Those children who have done some computer programming also understand the distinction between hardware and software. When asked if a computer could ever be wrong, these children agreed that it could. Some cited hardware reasons such as loose wires, power problems, or the outside chance that, "You might damage it by putting a powerful magnet up to it" (Jeffrey, 9). They then added other, different

97

reasons for a computer to "go crazy all over the place"—software reasons:

"The person that programmed it could have programmed it wrong."(Lindsay, 10)

"The computer can never be wrong unless you put in a statement wrong, the computer might think you mean something else." (Leanne, 10)

Many of the older children said they wanted to be programmers or scientists when they grow up. Indeed, many of their comments sound familiar to adults who have tried programming:

"You can do so much with not very many statements." (Kenneth, 10)

"I like to just fool around (with programming)." (Leanne, 10)

"I don't like it when you've got a program in it and it goes wrong so you spend almost your whole time trying to fix it." (Jenny, 10)

There's hope for programmers though, as Jason, 10, predicts, "Future computers will be able to talk and tell you if there's a mistake in your program."

Some of the children's comments about making computers understand what they mean suggest they sometimes expect the machines to comprehend or evaluate stored information. Young children, especially, appear somewhat confused about computers' abilities to think and learn.

For example, I observed a seven year old boy using a math program that responded to his inappropriate key presses with a "Stop It!" message on both sides of the program display. He slowly read, "Stop it ...stop it," then grumbled, "You don't have to tell me twice." Later he wanted to

know what else the computer was watching and if it was going to tell on him.

Similarly, a kindergarten-aged girl who repeatedly lost a tic-tac-toe game, asked, "How old is that computer?" Some older children are also hesitant to play games against the computer "because it might cheat" (Richard, 11) or "it already knows the answers" (Andrea, 11).

> *One class refers to the time spent cleaning up their work areas as "data storage time."*

Such comments suggest that children see computers as "knowing" rather than merely storing information. When asked directly whether computers can think—a point of great controversy among some adults—children give mixed answers:

"A computer can't react like a human brain, with emotions..." (Mark, 11)

"The computer works like a person..." (Kirk, 9)

"...it might misunderstand." (Satomi, 11)

"It's too smart..., it thinks too fast." (Fiona, 11)

"I just don't know if it thinks." (Ginny, 12)

Part of the indecision on this question may stem from the problem of defining thinking. As Chris, 8, answered, "Think?

My teacher tells me to think and I just don't know what she's saying...Does the computer?"

Implicit Computer Lessons

Every time a child works a lesson on the computer he learns how to organize and present information, what types of responses are acceptable answers, what feedback is important and how to handle mistakes. The structure of computer programs and the manner in which children interact with them have subtle but powerful effects. On a playful level, children frequently imitate synthesized speech, walk like robots, and make noises from pong games and R2D2. However, a more serious influence is evident in their work habits and attitudes about learning.

Computer terms become a part of these children's vocabularies. Even young non-readers quickly develop sight recognition for words like "press return," "name," "load," "save" and "error."

Children also use computer terms to refer to themselves and their school activities. One teacher noticed a student chanting, "Error, error, error" when he had made a mistake in his workbook. He then told her, "I'm going to run that question again."
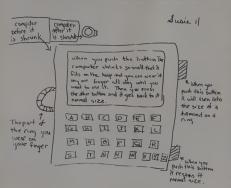
An older boy who had some programming experience excused himself for talking too loudly because he was "still in recess mode." One class refers to the time spent cleaning up their work areas as "data storage time." This is when they "input their worksheets" into plastic storage bins they call their "memory banks."

Another teacher actively encourages his students to use the computer's operation as a model for their own work habits. His students "program" themselves for each day's work, writing down a list of what they have to do, complete with line numbers, GOTO and IF THEN statements and a final END statement.

The influence of the computer on children's work habits and awareness of themselves as learners is among the most important benefits of using classroom computers. The value of educational computer materials would be greatly enhanced if they were designed to derive a more substantial, more deliberate advantage from this influence.

Children's views about computers in the classroom clearly show their ease with them as part of their daily lives. Children do not see these machines as threatening or overwhelming. Rather they are enthusiastic, confident and creative users who appreciate the capabilities of computers, and have high expectations for the future of computer technology and its applications:

"In the future, computers...will be like all the scientists in the world smushed into one." (Ilana, 11) □

Susie 11

computer before it is shrunk | computer after it is shrunk

When you push the button the computer shrinks so small that it fits on the hoop and you can wear it on your finger all day until you want to use it. Then you press the other button and it goes back to it normal size.

→ When you push this button it will turn into the size of a diamond on a ring

↑ The part of the ring you wear on your finger

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

↖ When you push this button it regains it normal size.

Sams offers five books designed to help you explore the basic workings of your Apple microcomputer. Here you'll find all of the information necessary to understand and use the limitless computing power of the Apple system. With these great books from Sams, your venture into the world of the Apple microcomputer will be a fruitful one!

APPLESOFT LANGUAGE Beginning programmers are given a comprehensive, step-by-step introduction to Applesoft, the language of the Apple II microcomputer. No. 21811. $10.95

INTIMATE INSTRUCTIONS IN INTEGER BASIC This complete introductory overview of Integer BASIC enables even the most nontechnical reader to write and use simple programs. No. 21812. $7.95

6502 SOFTWARE DESIGN The 6502 microprocessor is the heart of your Apple microcomputer. 6502 Software Design introduces you to programming in 6502 assembly language allowing you to perform high-speed processing and peripheral control not generally possible with higher-language programs. No. 21656. $12.95

MOSTLY BASIC: APPLICATIONS FOR YOUR APPLE® II Twenty-eight BASIC language programs for your Apple II microcomputer are thoroughly outlined and discussed. These programs are useful in all facets of the home and business. No. 27189. $12.95

APPLE® INTERFACING You are provided with real, tested circuits which interface to your Apple microcomputer for remote control, data acquisition, and control of external electronic and electromechanical devices. No. 21862. $10.95

Each book presents a different slice of the intricate workings of your Apple microcomputer. To take full advantage of this powerful machine, you must understand it from the inside out.
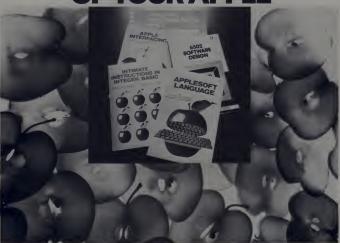
GET TO THE CORE
OF YOUR APPLE®

Computer Education Takes a Step
Backward in Canada

Betsy Staples

Searching for the perfect microcomputer for use in the classroom? So is the Ontario Ministry of Education. And the Canadian Advanced Technology Association (CATA) thinks it has the answer.

According to a report in *Computing Canada*, a research report prepared by CATA for the Ontario Ministry of Education outlining the functional specifications for a proposed all-Canadian educational microcomputer has been the center of much controversy in the computing community in Canada.

The specifications are intended to produce a compatible family of computers which will allow uniform implementation of computer education programs throughout the country. The only catch is that there is no such machine. And the insistence of the Ministry that it be entirely Canadian-made ensures that it won't exist for quite some time.

Since many schools rely on provincial grants which cover an average of 75% of the cost of computers for schools, speculation was that, if the proposal were adopted, most school boards would put off buying equipment until they could purchase machines that met the new specifications.

The report itself, completed in less than three months, and the methods used to compile it have also come under fire. Critics cite the failure of CATA to consult with either the University of Waterloo, one of the leading institutions in the area of microcomputer software, or companies that have already been successful in the educational computing field.

One educator, whose name was included in a list of those interviewed by researchers, claimed that his interview consisted of a five-minute conversation over a drink aboard a boat in Lake Geneva, Switzerland. At the time, he said, he stated his opposition to the concept.

In response to criticism, CATA has replied that the report was only a "first step," and that interested parties have been invited to respond and submit suggestions. □

Programmed Worms

Donald T. Piele

At a recent meeting of the National Council of Teachers of Mathematics in St. Paul, MN, microcomputers were a big hit. Any hour during the three day conference found a session being presented on some aspect of the use of microcomputers in the mathematics classroom.

The interest shown in these talks was great. In fact, if one wanted to get a seat at a talk, it was necessary to arrive ten minutes before its scheduled starting time and hope that a few seats would become vacant during the break period. What teachers had come to see and what speakers had come to show was CAI (Computer Assisted Instruction) software.

Book publishers with newly formed software divisions were there to show off highly animated programs for teaching students how to add, subtract, multiply, and divide. State supported software collection and distribution organizations were there to announce their newest teaching programs for mathematics and other school subjects.

The message from these people was clear: If you have CAI software, you can put your computer to use in the classroom today. And tomorrow, when more becomes available, the computer will be an indispensable teaching tool.
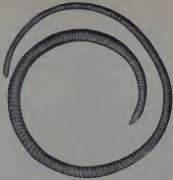
There seems to be significant difference, however, between what many teachers have in mind and what students would do with computers in the classroom if they had their way.

If I were to conjure up a conference of an imaginary National Council of Students of Mathematics (NCSM), the computer portion of the program would be quite different. There would be advanced workshops on Basic, Pascal, Logo, Forth, C, and Assembly.

Students would flock to the room of the featured speaker — Bill Budge — to hear a talk on three dimensional graphics and animation routines on the Apple II (A subject filled with mathematical ideas put into action). Games would be a popular topic — especially sessions on how to create those graphic animations.

Chess and Othello playing programs would be dissected. Sessions on how to use random access files would be conducted by students who have written scheduling programs for their own schools. In general, students would be primarily interested in demonstrating and sharing programming ideas.

Donald T. Piele, Department of Mathematics, The University of Wisconsin-Parkside, Kenosha, WI 53141.

In contrast, I got the distinct feeling at the NCTM meeting that speakers had learned to avoid any discussion of specific computer code for fear of setting off a stampede for the exits.

Through the correspondence that I have received from students all over the world who participated in the first international computer problem solving contest [2], I have gained a great deal of respect for the ability of young minds to write computer programs. I marvel at their ability to tackle a completely new programming problem and solve it in an original way.

It appears to be a natural language that is easily understood by the young. Students are excited about computers, and it worries me when I see very little overlap between what students are doing today with computers and what publishers and software distributors have in mind for them tomorrow. I hope that the teachers, who for the most part find themselves in the middle and searching for ideas, will take a lesson from their computing students.

Programmed Worms

Programmed worms is the theme of this month's column. I first learned about them in Martin Gardner's "Mathematical Games" section of *Scientific American* [1].

Imagine that you are a hungry worm looking for something to eat. You find yourself in the middle of a field that is filled with food distributed along rows and columns. You are resting comfortably at a node (an intersection of a row and column), gazing out upon the food that will sustain you on your journey to the north, south, east, or west (Figure 1). There is a one day supply of food along each segment between nodes. Once the food is eaten, it is gone for good — a worm cannot travel over the same segment twice.

The worm is programmed with a fixed set of rules that determines the direction it will move when it arrives at a node. These rules were first proposed in 1971 by Michael Paterson, a computer scientist at the University of Warwick, and are as follows:

Rule 1. If the worm arrives at a node with no segments eaten (except of course the one it arrived on), then it must make a right turn. (If the worm were to go straight under this situation, then it would go straight forever and quickly come to the end of the food patch; its path would, therefore, be uninteresting.)

Figure 1.

Figure 2. Quadrille Worms. a) Species I   b) Species II

Rule 2. If the worm arrives at a node and all the paths emanating from it have been eaten, the worm dies.

Rule 3. If only one segment at the node is uneaten, the worm takes it.

The simplest programmed worm is the quadrille worm.

The Quadrille Worm

A programmed worm crawling on a square grid and turning right at each new node is called a "quadrille worm." After three right turns have been made, it will be back at the node where it began. What happens then? It cannot turn right, because that path has already been eaten. It has only two choices. If it is programmed to turn left when it cannot turn right then it will traverse one more square and then die (Figure 2a). If it is programmed to go straight when it cannot go right, it will go on eating four more squares before it dies (Figure 2b). Each rule defines a different species. There are only two species of quadrille worms. This defines our first problem.
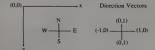
Problem — Quadrille Worms

Write a computer program that will simulate each species of the quadrille worm.

Discussion

The low-resolution graphics screen of the Apple II makes a perfect feeding ground for programmed worms. Other systems with graphics displays that have the ability to read the graphics screen work just as well.

The first thing that needs to be developed is a simple procedure for making turns. As discussed last month, this is best carried out with a direction vector (D1, D2). The following table matches up direction vectors with the four directions: north, east, south and west. Remember that on the Apple II the coordinate system that defines each point on the screen has its origin in the upper left hand corner.

A right turn shifts the direction you are facing by a clockwise rotation (north → east → south → west → north). This corresponds to the clockwise rotation of direction vectors represented by the transformation

$$(D1, D2) \rightarrow (-D2, D1).$$
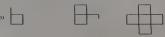
For example, $(0, -1) \rightarrow (1, 0)$.

Similarly, a left hand turn (north → west → south → east → north) is represented by the transformation

$$(D1, D2) \rightarrow (D2, -D1).$$

With direction vectors, it is easy to draw a path in any given direction. The statements

```
100 FOR I = 1 TO L
110    X = X + D1 : Y = Y + D2
120    PLOT X,Y
130 NEXT I
```

trace out L steps in the direction (D1, D2) on the low-resolution Apple II graphics screen.

Another command that comes in handy is the screen command SCRN(A,B). It is equal to the value of the color at the position A,B on the low-resolution graphics screen. For example, if the position 20,20 has been plotted white then SCRN(20,20)=15. We will use this function to test paths to determine whether they have been eaten or not. SCRN(A,B)=0 means the position (A,B) has not been eaten, SCRN(A,B)=15 means that it has been eaten.

The Program

Listing 1 is a program for generating the quadrille worm — species I as shown in Figure 2a. It is written in Applesoft Basic and uses the low-resolution graphics screen for drawing the worm's path.

Listing 1.

```
10 REM   QUADRILLE WORM - SPECIES I
20 REM

30 GR                         Turn on low-res graphics.
40 A = 15 : B = 15            (A,B) is "home."
50 X = A : Y = B              (X,Y) is current position.
60 COLOR = 15                 Color of path is white.
70 L = 8                      Length of path segment is 8.
80 PLOT X,Y                   Mark the home position
85 D1=1:D2=0                  Initial direction.

90 FOR I = 1 TO L             Eat a path segment.
100    X=X+D1 : Y=Y+D2        Move one step in (D1,D2) direction.
110    PLOT X,Y               Color it white.
120 NEXT I

130 T=D2:D2=-D1:D1=T          Make a right turn.
140 IF SCRN(X+D1,Y+D2)=0      If the segment is uneaten, then
       THEN 90                   eat it or else,

150 D1=-D1 : D2=-D2           Make a left turn.
160 IF SCRN(X+D1,Y+D2)=0      If the segment is uneaten then
       THEN 90                   eat it or else,

170 END                       Die!
```

Figure 4. "Look Ahead" Quadrille Worms.
a) Species I   b) Species II

Program Notes

1. Species I of the quadrille worm is programmed to make a decision at each node according to the following priorities:
   a) If food is available to the right, then turn right or else
   b) If food is available to the left, then turn left or else
   c) If food is available straight ahead, then proceed or else
   d) Die.

2. The right turn transformation (D1,D2) → (-D2,D1) is carried out in line 130.

3. Once the worm has made a turn to the right and has found the path already eaten, it must return to its original direction and then make a left turn to search for food. Thus it must make two left turns. One to undo the right turn, and the other to make the left turn. Two left turns are equivalent to the single transformation

$$(D1,D2) → (-D1,-D2)$$

which is executed in line 150.

4. Species I of the quadrille worm dies from lack of food before it has a chance to exercise the third option, so it was not included in the program.

5. It is an easy exercise to modify this program to produce Species II of the quadrille worm. The second and third decision priorities listed in 1 above are simply reversed. This worm turns right when it can; or, if it can't do that, it goes straight ahead; or, if it can't do that, it goes straight ahead; or, if it can't do that, it turns left when it can; or it dies. This can be accomplished by replacing and adding the following lines in Figure 3 to the above program.

Figure 3.

```
    10 REM  QUADRILLE WORM - SPECIES II

   150 T=D1:D1=D2:D2=-T        Go straight ahead.

   170 T=D1:D1=D2:D2=-T        Make a left turn.
   180 IF SCRN(X+D1,Y+D2)=0    If the segment is uneaten then
        THEN 90                eat it or else,

   190 END                     Die!
```

"Look Ahead" Worms

John Conway, a frequent contributor to Martin Gardner's column, wondered what would happen if the quadrille worm was given the ability to "look ahead" and see the distribution of eaten and uneaten segments at each adjacent node.

For example, what would be the path of quadrille worm I if it turned right if, and only if, it sensed that this would take it to a new node with four unen aten segments? Otherwise it would turn left.

It is very easy to find out. Instead of testing the next step after turning right, the worm must look ahead L steps and test the next node. This can be done by simply changing lines 140 and 160 in the program above to read

IF SCRN(X+L*D1,Y+L*D2)=0 THEN GOTO 90

The same change works for all "look ahead" quadrille worms. The resulting paths are shown in Figure 4. For best results on the low-resolution screen, set L=2.

In his article on programmed worms, Gardner suggested other ways to spice up the display of programmed worms.

"What happens when two or more worms of the same or different species interact? What happens when a newly hatched worm crawls a short distance along a defined path (such as a straight line of three units) before its repetitive behavior begins? How about two armies of worms crawling toward each other, each obeying a different pattern?" [1]

These questions are left unanswered as Gardner turns to a discussion of "simple isometric worms": worms feeding along an isometric grid of unit equilateral triangles. We will confine our discussion to a slightly larger class of worms—the octal worm.

Octal Worms

We can double the number of paths that originate at a node by allowing turns of 45 degrees. This produces the more complicated and interesting octal grid (Figure 5). All worms that live on an octal grid and follow the three simple rules stated earlier are called Octal Worms. An Octal Worm has three different right turns—a hard right (135 degrees), a right (90 degrees), and a soft right (45 degrees). The same is true, of course, for the left turns. There are thousands of species of Octal Worms.

Problem—Octal Worms

Write a computer program that will simulate a species of the Octal Worm.

High Resolution Screen Function

There is not enough room on the low resolution screen to capture the movement of the Octal Worm, so we will switch to the high resolution screen. Most of the low resolution graphics statements work in high resolution graphics if the letter H is added before the command, as in HGR, HPLOT, HCOLOR. One crucial exception is the screen reading command SCRN (A,B). Since this is so convenient for our investigations, we will have to create a high resolution function. Listing 2 is a program that does the job. It is a machine language program that is POKEd into memory and called up through the use of the & symbol. Type it in and run it. It only needs to be run once and it will remain active until the system is turned off. It could be made a subroutine in any program that needs a high resolution screen function.

Listing 2.

```
500 REM HIRES SCREEN FUNCTION
510 REM
520 FOR  I=768 TO 806 : READ J:POKE I,J : NEXT I
530 DATA 32,227,223,133,133,132,134,169,208,32
540 DATA 192,222,165,18,72,165,17,72,32,185
550 DATA 246,32,17,244,165,48,49,98,240,2
560 DATA 169,1,168,32,1,227,76,91,218
570 POKE 1013,76 : POKE 1014,0 : POKE 1015,3
580 END
```

Here is how it works. The command &C=A,B places the value 0 or 1 in the variable C depending on whether the point A,B on the screen is off (black) or on (white). This is weaker than the SCRN command in low resolution graphics, which actually returns the number of the color at the point A,B but it is all we need.

THE
CONSOLIDATOR
FOR VISICALC FILES ™

NOW YOU CAN USE **ALL** THE POWER OF YOUR **APPLE** AND **ALL** THE POWER OF YOUR **VISICALC**.

If you use VisiCalc, you must have The Consolidator. It's that simple. The Consolidator will save you hours of Keyboard time — adding any number of VisiCalc files together, virtually automatically.

For example, say you need to add the results of four weekly reports to get the monthly total — or add three months for the quarter, twelve for the year, etc. Until now, you had to key-in all the individual sub-totals and recalculate. Now, The Consolidator will do it for you.

The Consolidator will also print out VisiCalc commands, formulas, and the locations to which they apply. In the past, programs that did ONLY that have sold for as much as The Consolidator alone.

The Consolidator is easy to use, with prompts, options and formats all carefully spelled out. If you can use VisiCalc, you can use The Consolidator. It will handle all the column widths, rows and columns accommodated by VisiCalc. It will handle any matrix size which you are able to save with VisiCalc. And it will give you back hours of manual calculation and keying time.

You need The Consolidator to use ALL the power you've got with VisiCalc. It's a bargain at the introductory price of just $49.95, at your local dealer or direct. Visa and Mastercard holders order toll-free, 1-800-835-2246.
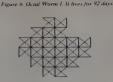
OMEGA MICROWARE, INC.
222 SO. RIVERSIDE PLAZA
CHICAGO, IL 60606
312-648-1944

Apple is a registered trademark of Apple Computer, Inc.
VisiCalc is a registered trademark of Personal Software Inc.

CIRCLE 200 ON READER SERVICE CARD

Figure 6. *Octal Worm I. It lives for 92 days.*
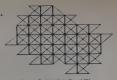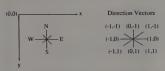
Figure 7. *Another Octal Worm. It lives for 148 days.*

Figure 5. *Octal Grid.*

A 45 Degree Turn

Finally, we need to know how to implement a 45 degree step. This is the crucial step in last month's spirolateral problem which was left as an exercise for the reader. Let's take a look at the expanded set of direction vectors.

A 45 degree right turn is equivalent to one clockwise rotation of the direction vectors. With a little experimentation, one discovers that the following transformation does the job.

$$S = ABS(D1) + ABS(D2)$$
$$(D1,D2) \rightarrow ((D1-D2)/S, (D1+D2)/S)$$

This is called a "soft" right. A plain right turn (90 degrees) is still $(D1,D2) \rightarrow (-D2,D1)$. A hard right turn (135 degrees) is equivalent to a right turn followed by a soft turn which is equal to

$$(D1,D) \rightarrow ((-D1-D2)/S, (D1-D2)/S).$$

Listing 3.

```
10 REM    OCTAL WORM I
20 REM

30 HGR
40 A = 120 : B = 60      (A,B) is "home."
50 X = A    : Y = B      (X,Y) is current position.
60 HCOLOR= 3             Color of path is white.
70 L = 15                Length of path.
80 HPLOT X,Y
85 D1=1  D2=1            Initial direction.

90 FOR I = 1 TO L        Eat a path segment.
100   X=X+D1 : Y=Y+D2    Move one step in (D1,D2) direction.
110   HPLOT X,Y          Color the point white.
120 NEXT I

130 T1=D1:T2=D2: S = ABS(D1) + ABS(D2)
140 D1=(-T1-T2)/S :
    D2=( T1-T2)/S        Make a hard right.
150 & C=X+D1,Y+D2        Hi-Res Screen Function.
160 IF C = 0 THEN 90     If the segment is uneaten, then
                         eat it or else.

170 T1=D1:T2=D2: S = ABS(D1) + ABS(D2)
180 D1=(T1+T2)/S :
    D2=(T2-T1)/S         Make a soft left.
190 & C=X+D1,Y+D2        Hi-Res Screen Function.
200 IF E = 0 THEN 90     If the segment is uneaten, then
                         eat it or else.

210 GOTO 170             Continue to make soft lefts.
```

Program Notes for Listing 3

1. This program assumes that the Hi-Res Screen Function program listed earlier has been run. This activates the & routine.

2. This particular species of Octal Worm is programmed to make a hard right whenever possible or to search counter-clockwise from a hard right until it finds an uneaten segment. The net result is that the worm tries to turn as far right as possible.

3. The worm dies when it falls into the infinite loop 170-210 searching for food. A counter could be added to this loop to put the worm out of its misery after seven tries.

4. The path of this Octal Worm I is shown in Figure 6. It dies after 92 days.

A Challenge

There are thousands of different species of Octal Worms that can be programmed. Many of them traverse the same path but in a different order. All of the ones I have tried eventually die. The longest one lasted for 148 days (See Figure 7). Can you program an octal worm to last longer? Remember the three rules:

1) The worm must turn right if possible. This can be a hard right, a right, or a soft right.

2) The worm dies at a node with no segment to eat.

3) If only one segment is uneaten, the worm takes it.

The remaining priorities are up to you. Find the longest living Octal Worm. (No "look ahead" worms allowed.) I will publish the best results in a future article.

A Solution To Last Month's Problem

Last month readers were asked to replace the 90 degree turn in the Spirolateral II program with a 45 degree turn capability. After reading this month's article that presents no problem. Simply replace line 120 with:

```
120 S=ABS(D1)+ABS(D2): T1=D1:T2=D2
    IF MID$(N$,I,1)="R" THEN D1=(T1-T2)/S
                             D2=(T1+T2)/S
```

and line 130 with:

```
130 S=ABS(D1)+ABS(D2):T1=D1:T2=D2
    IF MID$(N$,I,1)="L" THEN D1=(T1+T2)/S
                             D2=(T2-T1)/S
```

Delete the number 140—it shouldn't have been there.

References

[1] Gardner, Martin, "Mathematical Games: Fantastic Patterns Traced By Programmed Worms," *Scientific American*, 117-123, November 1973.

[2] Piele, Donald T., "The Second Annual International Computer Problem Solving Contest," *Creative Computing*, 8:154-163, January 1982. □

Overcoming Blocks and Barriers to Creativity

Part 2: Overcoming Organizational Barriers

Eugene Raudsepp

Last month, the author examined some ways to eliminate psychological blocks to creativity. This month he looks at organizational barriers.

No matter how great your creative potential, that potential must still be realized in an organized setting. New ideas must be developed that are relevant to the problems at hand, and those ideas must be "sold" to—and implemented through—others in the hierarchy of authority. Understanding the barriers to change that exist in most organizations will help you to exercise your creativity more effectively.

Resistance to change is a major impediment to creative problem solving in most organizations. As irrational as such resistance may seem at times, it does serve the purpose of testing new ideas so that they will not be accepted and implemented prematurely. Once you understand the individual and organizational obstacles that inhibit change, you will be able to develop your creativity more fully and "sell" your ideas more successfully.

Eugene Raudsepp, President, Princeton Creative Research, Inc., 10 Nassau St., P.O. Box 122, Princeton, NJ 08540.

Personal Insecurity

Most people see themselves as not only willing to consider new ideas, but as genuinely supportive of such ideas. However, the opposite is more often the case; most people are prone to accentuate the disadvantages of a new idea, rather than look for its possible benefits. Such criticism is designed to protect them from having to cope with the changes the new idea will bring.

Creativity without stability is destructive.
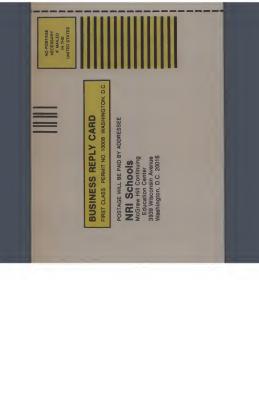
The status quo gives many people a feeling of security because it represents a situation that they have already mastered or are comfortable with. A proposed new idea frequently appears to be a direct threat to their present status. They fear that they might have to develop new skills and to establish a new credibility in the eyes of their peers and superiors. There is also a feeling that accepting a new idea is equivalent to denigrating the approach that the new idea will replace.

Clinging to the status quo would be all right if nothing changed. But things do change. Companies that encourage bold, new ideas and approaches move ahead; those that don't, fall behind. The danger in clinging to the familiar is that it may stunt the growth of the organization to such a degree that it fails to cope with the competition.

Of course, creativity without stability is destructive. To survive, an organization has to have rules, regulations, and controls to channel the actions of its employees into predictable and orderly routines. Without such routines, the organizations would lapse into chaos. Thus, a certain amount of inflexibility is necessary to accomplish organizational objectives. Unfortunately, many organizations, especially larger ones, become so overburdened with rules, procedures, and policies that the possibility of real creativity is almost completely snuffed out.

The resistance that can be expected to a new idea is proportionate to the degree of change that the new idea represents. The invention of the printing press is an interesting case in point. Not only did the calligraphers whose craft was threatened with extinction malign the new device, but the device was also opposed by those who feared that there would be negative consequences if the public learned to read. Many people believed that the learning

BUSINESS REPLY CARD

FIRST CLASS   PERMIT NO. 10008 WASHINGTON, D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

NRI Schools

McGraw Hill Continuing
Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

represented by the printed word would bring heresy, disobedience, and recalcitrance to an otherwise well-mannered people. The invention of the telegraph, airplane, incandescent lamp, and typewriter brought similar reactions.

In fairness, however, it should be pointed out that there are times when new ideas *should* be rejected. For example, the number of ongoing projects in an organization may be so large that undertaking additional work on a new idea might disrupt the work that is already in progress. In such cases, it is obvious that the time for the idea has not yet come. The intelligent innovator will "test the waters" before he tries to sell his new idea, so as to avoid such conflicts.

Another bona fide reason for rejecting a new idea is that it might be too costly to develop. In most companies, simple economics dictates what can and cannot be pursued. Also, a new idea may involve only a small part of a procedure, process, or product that is scheduled for a complete change later. The proposed partial idea may not fit in with the planned overall change.

How much effort you devote to developing a new idea should depend on the potential of the idea and the time available. In any case, you should not try to sell your idea prematurely. If the idea requires more background knowledge or better supporting arguments, get the added information before proceeding. Or, if you are uncertain about the value or practicality of the idea, develop the pros and cons more fully before you commit yourself.

There are times when new ideas should be rejected.

Even if your ideas are not always accepted, the practice of creative thinking will benefit you in other ways. It will prevent you from going stale on the job, increase your understanding of your work, and stimulate further self-development.

Threat to Authority or Position

Some people react negatively to new ideas because the ideas are not their own. Managers and supervisors are especially prone to play down the value of new ideas because they feel that their power and status are threatened if their subordinates or associates suggest them. They believe that if changes are necessary, they should be the ones to think of them.

Change is also frequently fought because it makes someone's job insecure or tumbles an expert from his pinnacle of recognition. For example, a technical innovation may introduce a new approach to a particular job. A person who for years has followed a proven practice with great skill and confidence may suddenly find himself to be a novice who is feeling his way down the painful path of learning a new skill. He may feel that the change threatens his earnings or his chances for advancement or recognition.

Perhaps the most prevalent reason for new ideas being viewed as a threat to existing authority or position is that the ideas which originate in one department directly affect people in another department. In the typical organization, each person's responsibility is carefully defined and is looked on as his own special preserve. When somebody comes up with an idea that concerns another person's area of responsibility or expertise, the usual reaction is defensiveness or hostility. The affected person feels that someone is trying to trespass into his area of authority and responsibility.

Determining Your Organizational "Fit"

Many people's creative potential is stifled by a mismatch between their personal needs and work requirements on the one hand, and the nature and structure of the organization of which they are a part on the other. Once you are aware of the factors that motivate you, you can assess your present organizational "fit." If necessary, you can act to find a better fit in your present organization or seek a more appropriate work climate elsewhere. The following exercises should help.

Personal needs Check the items that pertain to you. Then rank the checked items according to a 1, 2, 3, etc., system of priority.

_____ Economic security
_____ Stimulating environment
_____ Recognition by superiors
_____ Respect from subordinates
_____ Recognition by the public
_____ Recognition by friends
_____ Pleasant work location
_____ Variety on the job
_____ Maximum use of abilities
_____ Independence
_____ Opportunity to be creative
_____ Time for self or family
_____ Self-fulfillment
_____ Service to society
_____ Power over people
_____ Influence over policy
_____ Self-determination
_____ Money
_____ Travel
_____ Other _____

Work environment: Complete the following sentences by filling in the line with a statement that is appropriate to your idea of an ideal work environment. Work quickly and give your first impression. Try to frame each statement as a single idea that is crucial to your needs. Remember, there are no right answers, only relevant ones.

I want.
A boss who _____
The attitude of my coworkers to be _____
A company that _____
To use skills that _____
To get recognition for _____
A location that _____
Freedom to _____
The pay to be _____
Working hours to be _____
To travel _____
To advance _____
Important changes in _____
My abilities to be _____
Differences of opinion to be _____
Decisions to be made _____
Working conditions to be _____

Review the statements you made and check the six most important ones. From the six statements, choose the four most important. From the four statements, choose the two most important. From the two statements, choose the one that is most important. Now ask yourself why the conditions outlined are important to you and how they can be realized.

This artificial but very real barrier to change not only kills many valuable ideas; it also prevents the free flow of information and communication that should exist between departments.

Increasing Dependency

As society gets more complex, the network of human dependencies built into most organizations expands. Yet, these dependencies are frequently the enemies of creativity.

To see how such organizational dependencies inhibit creativity, consider the expectations of the typical employee. Instead of assuming that he is responsible for his own actions on the job, he usually believes—often correctly—that responsibility has been assigned to those who supervise his activities. Thus, he waits to be directed and stimulated by others. Rather than relying on his own judgment, he is constantly seeking and depending on the opinions of others.

The typical organizational setup encourages a person to be a passive, adaptive individual who must use his mind and energy to please individuals who are higher than he is in the organizational pecking order. In many cases, he is relieved of the responsibility for failure on the job. For example, responsibility for a bad decision about a new idea is considerably lessened if the idea and its subsequent implementation have been approved by a committee. In such cases, the individual can claim immunity through not having acted independently. The temptation to preserve one's ego by claiming that it was "the group's idea, not mine," is great. By the same token, a person will often accept and use ideas that he knows are inferior to his own in order to be able to "share the blame" should things go wrong.

Excessive Routine

Most people who work in organizations are constantly interrupted by telephone calls, memos, visits by colleagues, idle chatter, unscheduled meetings, letter-writing, and a host of administrative trivia that leave little time for creative thinking. One effective way of overcoming this is to define one's primary functions so that undisturbed time is reserved for creative thought.

How a person uses his free "think time" depends on his particular style of creativity. Activities such as daydreaming, reverie, and imaginative speculation are usually conducive to creative results. In fact, many genuine breakthroughs in thought or invention began as mere musings or fantasies of a far-ranging, receptive mind. Unfortunately, Western culture discourages such activities. They are not only regarded as a waste of time, but as indications of a lack of maturity.

In most business organizations, individuals who are forever shuffling papers,

pacing the corridors, or visiting their colleagues are perceived as being productive; those who quietly sit and think at their desks and occasionally stare into space are considered lazy. Thus, the person who tries to do some creative thinking feels uncomfortable and guilty, particularly if the boss happens to be around.

Most people seem to equate "busy-ness" with effectiveness. They feel uneasy or guilty if they are not constantly doing something, as if every minute of their lives has to be spent in some "practical" activity. Although this utilitarian orientation seems commendable to many people, some experts are convinced that such a preoccupation with practicality inhibits the spontaneity and serendipity that make genuine creativity possible. The antidote to such "hyperactivity" is to realize that selected periods of quiet thought and reverie are not wasted time, but pauses for regeneration that will eventually make you more productive than before.

Even after you've worked out a tough problem, you may find that no one is interested.

Going Through Channels

Before the development of an idea is approved, the proposal must travel through many levels of decision makers, which poses a grave threat to the survival of the idea. To illustrate, suppose a person has an idea he wants to explore. The idea isn't ready to be put into a formal proposal, but he wants to find out whether individuals up the line in the organization would support it. First, he talks with his group leader. If he agrees, the person takes the idea to the section head. From section chief, the idea goes to the department supervisor, the technical director, and the division manager. The process will involve even more decision makers if the idea is the kind that affects other divisions of the company.

A "break" anywhere along the line would be critical. The idea could become distorted or lost altogether since it is not fully worked out and is easily subject to criticism. It could even become the basis for an attack on the originator, his manager, or his group. Given these possibilities, it is easy to see why people in organizations are not anxious to develop or pursue their brainstorms.

The best way to fight the problems associated with going through channels is to develop the idea to the point where it cannot be easily passed off for lack of thought. Also, you might test your idea first on those who are in a position to help you get the needed approvals. Once these people are sold, the rest will be relatively easy.

Failure to Get Others Involved

Even after you've worked out a tough problem, you may find that no one is interested. Sometimes this is due to a lack of confidence in the merits of the idea, but most often it is due to your neglecting to get others involved in your solution.

Idea implementation requires cooperative effort. But many creative thinkers get so ego-involved with their ideas that suggestions for modification are automatically opposed as unnecessary tampering. As a result, they fail to elicit the participation and cooperation of associates during the development and implementation stages of the idea.

Suggestions for improvement should be heartily welcomed. When others take a keen interest in your idea, they become personally involved; your idea becomes "our idea." If you are generous in sharing the idea with others, everyone will be a winner and you will be remembered as the one who came up with the original concept in the first place. On the other hand, if you cling to the purity of your idea too strongly, others will not come to your aid when you run into obstacles. Insistence on receiving all the credit for an idea is not only unrealistic in most organizations; it is the quickest route to resentment.

Even if someone suggests revisions that would add very little to the original idea, you would still be ill-advised to reject them. If you force others to accept your idea unchanged, they might drag their feet or even sabotage the idea during implementation. □

"I wonder what profound thought is occupying our top system analyst's mind now."

Our Robot Friends

Biblical legends tell us that God created Man in His own image. This is a supremely egocentric process and one that the human race has desired to duplicate with our *own* creations.

When we became technically advanced and were able to build working robots, they were much more specialized in their functions than the versatile human body—as a result, our robots did not end up looking in the least humanoid. In fact, most electro-mechanical robots look more like manufacturing assembly lines than *people*. What a disappointment.

Fortunately, science fiction (like all arts) deals more with the *spirit* of things than with the everyday realities. Science fiction writers, artists, filmmakers and toy makers have given us what science has not found practical—machines fashioned in our own image.

Toys, because they are used by children, tend to have a relatively short life. They are not the kind of thing one thinks of collecting. But Genevieve and Christopher Cerf do collect toys—toy robots in particular.

We recently visited Genevieve and Chris in their New York City apartment and photographed some of the toy robots they have collected over the past ten years.

Rather than attempt to recount the experience, we have printed here a few of the photos and descriptions of the mechanical men as we heard them.

Genevieve and Christopher Cerf with Astrodog and Fatso (the real dog).

Genevieve: "The worst thing about having such a collection is that it is impossible to tell visiting children, 'No, you can't play with our robots.' They're our robots, and you might break them.' And if you barely know their parents, it makes it even harder—they think you're strange."

Photos by David Ahl.

Chris: "One thing that's fun is to have a game where the little ones try to walk among the big ones— we've lost a few as a result of the robot wars... The orange and green one, by the way, actually has two sets of legs."

Genevieve "Do you know how big a job it is to go through all of these robots and put new batteries in? It takes at least a half a day to get them all powered up."

Genevieve: "The depressing thing, though, was finding this book which my sister gave me in which I discovered that there were hundreds of robots that I had never seen. The author is obviously as crazed about them as I am. However, I did see a few of mine in there and I have some not in the book."

Chris: "This is the oldest one we have, King Ding. It has a little robot named Brain that goes up and down in an elevator. You control it from the back."

Chris: "This robot has a sound receiver in his head. To make him go, clap your hands or make a loud noise. No home is complete without one!"

The Organizer®

The Organizer® is available for Apple® and IBM® personal computers.

Suggested retail price: $250
Telephone dialer: 75

You thought that with a computer you could store information simply by touching a button. Well, now you can.

The Organizer™ is a new kind of personal and professional support, one which meshes familiar ways of doing things with the versatility of a computer. When using The Organizer, your view of data remains like the one you've had with file cards, address books and appointment calendars. And that's just the beginning.

We've put all of your desk tools together in one place,

an appointment calendar,
index cards & notepads,
a calculator,
a telephone phone dialer,
an automatic alarm clock,
a programmable alarm clock,

and they talk to one another.

Using The Organizer is so natural you'll forget there's a computer behind it.

conceptual instruments
company

4730 Warrington Avenue
Philadelphia, PA 19143

CIRCLE 147 ON READER SERVICE CARD

Trademark of Apple Computer, Inc.
®Trademark of International Business Machines Corporation

A flock of cybernetic chickens.

Genevieve: "This Japanese one is interesting because it can be made into a truck. Kids love it. You can take it apart and take all the stuff out. You can even take the little fists off. Then the doors close up, and a cab and engine covers the face. To get it back to a person, you open all the doors, put his fists back on, put his sword in, and put the little cars back in his shoes."

Three early robots. The blue one on the right has a "TV" screen for a chest which flashes light patterns. It also walks, but the walking part is dull compared to the light displays.

Chris: "This R2D2 is radio-controlled. One day we forgot to turn him off and, when a police car went by on the street, he started walking and crashed right off the shelf. He doesn't work too well anymore."

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

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

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

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

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

THE SHATTERED ALLIANCE is

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

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

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

All you need to play this game is a 48K Atari 800 with mini floppy disc drive. For only $59.95, you can own this fantastic piece of real estate. So hurry on down to your local game computer store and get THE SHATTERED ALLIANCE today!

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

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

Atari is a registered trademark of Atari Inc.

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

Apple is a registered trademark of Apple Computer Inc.   CIRCLE 254 ON READER SERVICE CARD   TRS-80 is a registered trademark of the Tandy Corporation

Using Disks with Atari Basic

Dave Johnson
and
Embee Humphrey

Dave Johnson and Embee Humphrey, Atari, Inc., 1272 Borregas Ave., Sunnyvale, CA 94086.

Table 1.

Open Code	Function
4	Input. (Read from the file only).
8	Output. (Write to the file only). NOTE - this erases any data in this file. To OPEN a file to add data, use 9 for APPEND.
12	Both input and output. (Read and write).
9	Append - add to end of file. (Write, beginning at end of file).

This article will help you write programs that use data files that are saved on disk. We hope you will use this information as a beginning point for experimentation. We believe that the only way to really learn programming is to try things and make mistakes in the process. We have included answers to what we have found to be the most common questions.

These are the Atari disk input/output commands:

OPEN #reference number, open code, 0, filespec: The first parameter, #reference number, is a number between 1 and 7 that is used to refer to that file throughout a program. It must be preceded by the "#" symbol. The second parameter, open code, is a number that tells the computer if you want to read, write, etc. See Table 1.

The third parameter is reserved for special control codes that will not affect you in disk operations. Always use a 0 for this third parameter. The last parameter, filespec, is the device and file name. Filespec can be a string variable, or the device and filename enclosed in quotes. Example: You may declare A$= "D:FILENAME" and use A$ in the OPEN statement, or you may just use "D:FILENAME" itself.

CLOSE #reference number: This tells the computer that #reference number is no longer being used. (Note—if you are writing data, and fail to close a file, any data in the buffer will not be saved on disk.)

PRINT #reference number;[What you want printed]: This is used the same way as a regular Basic PRINT statement. The #reference number tells the computer to PRINT to the file you have opened using that reference number.

INPUT #reference number, [Input list]: This works like a regular Basic INPUT statement. The #reference number tells the computer to look for input from the file opened using that reference number.

NOTE #reference number, variable for SECTOR location, variable for BYTE location.

POINT #reference number, SECTOR variable, BYTE variable: NOTE and POINT are explained later.

OPENing a file tells the computer that you are going to use a particular file in a particular way, and that you will refer to that file in your program using a particular reference number. (Atari Basic manuals call this reference number an IOCB number, for Input/Output Control Block.)

For example, if you want to read from file TEST.DAT, and you decide to refer to that file as number 4, you would use the following open statement example. (The number doesn't matter as long as whenever you want a certain file, you refer to it with the same number.)

An example of OPEN: OPEN #4,4,0, "D:TEST.DAT". This means OPEN a file, read data from the file, the file will be on device "D;" (disk), whose name is TEST. DAT, and remember that this file will be referred to as #4 for future I/O operations.

To read from the file you would say: INPUT #4; expressions....

And finally, when you are done with the file: CLOSE #4.

INPUT and PRINT

INPUTing and PRINTing with the disk is the same as inputting from the keyboard and printing to the screen. The #reference number tells the computer that you are using the disk instead. (You must have OPENed a disk file with that number.)

The easiest way to experiment with this is in DIRECT mode, typing in statements without line numbers. Now try a few things:

OPEN #1,8,0,"D1:TEST.DAT": This opens a file, TEST.DAT, for write only. It will be referred to later as #1.

PRINT #1;"THIS IS A STRING": This writes this message to the file.

PRINT #1; 12345.6: This writes the number 12345.6 into the file.

PRINT #1;"THIS IS ANOTHER STRING";12345.6: This writes both the string and the number to the file with the same statement.

CLOSE #1: This closes the file.

To see what is on the disk: OPEN #2,4,0, "D:TEST.DAT".

We used #2 here to demonstrate again that the number itself does not matter. What matters is that the same number be used for all references to *that* file after the OPEN.

Since you will be reading strings, you need to DIM a string to read into: DIM S$(30)

Now, to read it, type: INPUT #2,S$. Then type: PRINT S$, and the screen displays: THIS IS A STRING, which is what you put on the disk.

Now, a surprise! We told you that PRINTing to disk was the same as printing to the screen. When you type PRINT #4; 12345.6, you put the CHARACTERS "12345.6" on the disk, not a binary representation.

Now type:
INPUT #2,S$
PRINT S$
and the screen displays: 12345.6

Software for
NorthStar
Users

EXPENSE PROFILE $29.95
Now a program that really helps at income tax time. It summarizes expenses by categories and by person. Makes SEPARATE vs JOINT TAX RETURN comparisons simple.

Promotes frequent review of spending habits. Guided by MENUS, add new expenses, categories, and users anytime. Quickly search to any item to make changes. Store expenses on disk automatically.

DYNAMIC BUDGET $29.95
Cope with rapidly changing economic conditions. Forecast effects of INFLATION on your family. CALENDAR built-in so recurring items like rent entered only once.

Monthly listings of expenses, income, and balance. Change or add items anytime, data automatically stored on disk.

PATHFINDER DISASSEMBLER $22.50
Z80 or 8080 code. Pauses at each jump or call to allow you to follow program or continue straight ahead. Printer & video output

KID MATH $17.50
Math drill. Watch speed, accuracy and confidence grow.

TAX FORMS $49.50
Fills out 1981 federal personal tax forms. Uses almost any printer. Write for details. Avail. Feb 82
First class postage paid in U.S. MD residents add 5% tax

The Software Connection
10703 Meadowhill Rd.
Dept CC   Silver Spring, MD 20901

CIRCLE 273 ON READER SERVICE CARD

Reading the number into a string gets the string representation of the number that was on the disk. Reading the number into a number variable would also have worked. If the binary representation of the number was put on the disk, you would not have been able to read it into a string and see it. This is important to remember, because putting the number 10 on the disk takes two bytes of disk space, while putting the number 123 on the disk takes three

Figure 1.

```
DIM S$(20)
OPEN #1,4,0,"D:TEST.DAT"
INPUT #1,S$    Read the first line.
INPUT #1,S$    The second.
INPUT #1,S$    The third.
INPUT #1,S$    And the APPENDED line to verify this.
PRINT S$
```

bytes. This will be important in the discussion of random access.

Remember: the number of digits is equal to the number of bytes the number will take up on the disk.

The next example shows another way that similarity between screen and disk output may surprise you. From the above example the file is still open:
Type:

INPUT #2;S$
PRINT S$

and the screen displays: THIS IS A STRING 12345.6

It read into the one string *both* the string and the number that you had put on the disk. When you put this data into your file using PRINT#1;"THIS IS A STRING";12 345.6, it went onto the disk exactly as it would have appeared on the screen. When reading it, the computer had no way of knowing where the string stopped and the number started on the disk.

You could not have said INPUT #2;S$, NUM because inputting S$ would have taken you past the number so, if you are going to read data back as different lines, put it on the disk as different lines.

APPEND

Now you can experiment with APPEND. You already have a file, TEST.DAT, on the disk. If you want to add data to this file you must use APPEND, because opening

---

the file to write again (with an 8) will erase what you have already put into it.

In OPEN #5,9,0,"D:TEST.DAT", the 9 means APPEND.

Now, try this. Type:

PRINT #5;"THIS IS ADDED"
CLOSE #5

To see what happened, type the lines in Figure 1. The screen displays: THIS IS ADDED

Trying to type INPUT #1,S$ one more time would cause an ERROR 136, which means END OF FILE.

CLOSE #1

NOTE and POINT

NOTE #reference number, sector variable, byte variable: This takes NOTE of where the disk read/write head is physically positioned. The #reference number is the IOCB number. If you want to mark where you are in a file opened as #1, you say NOTE #1,variable1,variable2. Variable1 saves the sector and variable2 saves the byte.

POINT #reference number, sector variable, byte variable: This physically POINTs the disk head to the sector and byte. If that sector and byte are not located within the file opened with that #reference number, you get an error message.

NOTE and POINT are useful for repositioning the read/write head to the beginning of a file. For example, if you want to read the first string in your TEST.DAT file twice before proceeding, you would type the lines in Figure 2.

Figure 2.

```
DIM S$(20)
OPEN #1,4,0,"D:TEST.DAT"
NOTE #1,X,Y    take note of where file starts.
INPUT #1,S$
PRINT S$
The screen displays: THIS IS A STRING
```

The screen displays: THIS IS A STRING
If you read again, you will be reading the second item in the file, which is: 12345.67.

To point back to the beginning:
POINT #1,X,Y
INPUT #1,S$
PRINT S$

Chart your financial future with MicroFinesse™

In this fast-paced business world, the best way to stay competitive is to be able to see your financial alternatives clearly and make decisions fast.

That's why P-E Consulting Group Ltd. developed MicroFinesse, a complete package providing professional financial forecasting PLUS full high-resolution color graphics support, including pie charts, histograms and graphs, for the financial projections you create.
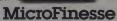
Now this evolutionary financial resource planning tool eliminates the time-consuming reprogramming required for financial model

consolidation or expansion. Micro-Finesse's menu-driven command format can also generate up to 15 user-defined reports per model, with visuals, without reprogramming.

Previously available only for mainframe applications, the MicroFinesse program can now be purchased at leading Dealers nation-wide. MicroFinesse is specifically designed for the 48K Apple II® with the Apple Language Card.

So when your variables are many and your time is limited, take a good look at the financial artistry of MicroFinesse.

MicroFinesse

Distributed by: OSBORNE/McGraw-Hill
630 Bancroft Way, Berkeley, CA 94710, (415) 548-2805

Apple II is a registered trademark of Apple Computer, Inc.

Figure 3.

```
1000 L = LEN (STR$ (DSKNUM )) . REM GET THE LENGTH
1010 DSK$ (1,10-L) = "0000000000" :REM PUT ZEROES IN FRONT OF
     STRING
1020 DSK$ (11-L) = STR$ (DSKNUM) : REM PUT NUMBER RIGHT JUSTIFIED
     INTO STRING
1030 RETURN
```

and the screen will again display: THIS IS A STRING, which is the first record in the file!

Random Access

NOTE and POINT are useful for random access. If you keep track of where you put data, using NOTEs, you can later get at the data, using a POINT, without reading everything in front of it.

Briefly, suppose you have a mailing list. Each time you save a record, you NOTE where it is physically written, and save those sector and byte numbers in arrays—

Figure 4.

```
10 DIM S$(20)
20 OPEN #1,4,0,"D:TEST.DAT#:REM OPEN THE FILE
30 TRAP 70:REM ON END OF FILE ERROR, GOTO 70
40 INPUT #1,S$:REM READ FROM THE FILE
50 PRINT S$:REM PRINT TO THE SCREEN
60 GOTO 40:REM KEEP DOING THIS
70 CLOSE #1:REM END OF FILE REACHED, CLOSE FILE
```

SECTOR(record number),BYTE(record number). Then you save those arrays in another file on the disk.

To look up records, you would first read the array data file to put the information back into the arrays. You would then look up the individual records by POINTing directly at them. (SECTOR and BYTE are the variable names we have assigned to our arrays.)

X=SECTOR (indexed by record number). (To read the 23rd record you would use X=SECTOR(23).)

Y=BYTE (indexed by record number)

POINT #reference number,X,Y and you are ready to read that record.

Be careful. If you want to modify a record and write it back into the file in the same place, you must not change the length of that record. If it is changed, it could overlap into the next record, overwriting the data there. Remember that numbers use as many bytes as there are digits.

Before you modify records, you must format numbers into strings with leading blanks or zeroes that are always the same length. The subroutine in Figure 3 accomplishes this. Before calling, put the number you want formatted into the variable DSKNUM. The number will be formatted into a string of length 10 with leading zeroes and placed in string variable DSK$.

Traps

Atari Basic has a TRAP command. The format for the TRAP command is: TRAP line number.

To TRAP to a line number means to go to that line number if an error occurs. This can be useful for disk operations. For example, if you want to go to a certain line number when you reach an END OF FILE, you can take advantage of the ERROR 136 that happens when you try to read past the end. See Figure 4. □

128   April 1982 © Creative Computing

HANGMAN

For the Pocket Computer

Jim Madeheim and Al Vesterdal

This is the familiar game of "Hangman." Of course, there is no picture of you being hung, but the game is the same. You get nine wrong guesses. After your 10th wrong guess you're hung, and the word is printed out.

Someone has to type in a word for you to guess. Maximum length is 12 letters. If the word is less than twelve letters, then type in a zero after the last letter. The word will be printed out so that the person picking the word will know they spelled it correctly. Then the "ENTER" button is hit and the pocket computer handed to you. Do not type in the letter you are guessing until the word -guess- appears. Otherwise all kinds of funny things could happen.

The first line in the program is there in case you make the mistake of turning off the computer or hitting the BREAK button. When you start the program over again, just type in a -1- and the program will continue where you left off.

You can duplicate lines by changing the line number. This will make it easier to type in lines 595, 801, and 860.

Jim Madeheim and Al Vesterdal, 3802 W. Medina Rd., Tucson, AZ 85706.

Sample Run (simulated)

Inside the parentheses is what is displayed on the screen <X> means you typed in the letter X and then entered <> by itself means enter.

```
<R.>
(MISTAKE? (1=YES,0=NO)) <0>
(?)<W>
(?)<O>
(?)<O>
(?)<D>
(?)<E>
(?)<N>
(?)<0>   The zero means you have finished the word.

(WOODEN)
<>
(GUESS?) <A
BEEP (----          1. A)
<>
(GUESS?) .E>
BEEP (----E-         1. A)
<>
(GUESS?) <I>
BEEP (----E-         2. AI)
<>
(GUESS?) .O.
BEEP (-OO-E-         2. AI)
<D>   This was a mistake. You must push the ENTER button first.
(.............) This shows an error.
"CL"   Push the CLEAR button and then enter.
(GUESS*?) <O.   Zero shows the last line printed out.
BEEP (-OO-E-         2. AI)

(GUESS?) .D.
BEEP (-OODE-         2. AI)

(GUESS?) .M.
BEEP (-OODE-         3. AIM)

(GUESS*?) .N.
BEEP (-OODEN         3. AIM)

(GUESS?) W.
(WOODEN     YOU WIN)

(?)   The program is ready to accept another word.   □
```

```
            HANGMAN
     FOR RADIO SHACK'S POCKET COMPUTER
     BY   JIM MADEHEIM AND AL VESTERDAL

505:INPUT "MISTAKE? (1=YES, 0=NO)";W:IF W-1THEN 800
510:IF W=1TO 12:A$(W)=" ":NEXT W
520:A$(52)="YOU ARE":A$(53)=" HUNG"
550:IF W=31TO 42:INPUT A$(W):IF A$(W)="0"THEN 590
552:A$(W-30)=A$(W):NEXT W:W=W+1
590:X=W-31
594:FOR W=1TO X:A$(W)=A$(W+30):NEXT W
595:PRINT A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$
598:A(27)=O:A(28)=O1FOR W=1TO 1:A$(W)=" "
     :A$(W+10)=" ":NEXT W
599:FOR W=1+X10 12:A$(W)=" ":NEXT W
600:INPUT "GUESS" "?Y$:IF Y$="0"THEN 800
610:Z=0:FOR W=31TO 30+X
620:IF A$(W)=Y$LET A$(W-30)=Y$:IZ=11A(27)=A(27)+1
630:NEXT W:IF Z=1THEN 760
640:A(28)=A(28)+1:IF A(28)>=10 THEN 855
650:A$(A(28))=1J.-Y$
760:IF A(27) X THEN 800
785:A$(53)="WIN":A$(52)="      YOU "
786:GOTO 860
800:BEEP 1
801:PRINT A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;
     A(28);"*"+N$;O$;P$;Q$;R$;S$;T$;U$;V$
820:GOTO 600
855:FOR W=1TO X1A$(W)=A$(W+30):NEXT W
860:PRINT A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;
     A$(52)A$(53)
865:GOTO 510
870:END
```

MORE POLY'S

Neil C. Rowe

In my December 1979 article in this magazine I showed how to make some interesting geometric designs from very simple graphics programs. Here are some more, this time less mathematical.

Introduction

Once again, consider a "turtle" graphics system. That is, a line drawing system where one moves an imaginary object called a "turtle" in order to make lines. Three kinds of commands can be made:

• FORWARD: go forward a certain distance in the direction you are facing making a line as you move.
• PENUPFORWARD: go forward a certain distance in the direction you are facing, but don't draw anything.
• RIGHT: turn right a certain number of degrees to face in a new direction.

So for instance to draw a "+" you could say (assuming, as we will in what follows, that the "turtle" always begins pointing straight up):

```
FORWARD 100
RIGHT 135
PENUPFORWARD 71
RIGHT 135
FORWARD 100
```

The springboard for both this paper and the preceding one is a famous program called POLY. It just consists of repeating a large number of times the pair of instructions:

```
FORWARD S
RIGHT A
```

where S and A are particular numbers. Figures 1-4 give POLYs for A=90, A=60, A=144, and A=1.

In what follows, we'll assume there's a variable called HEADING which keeps track of the total amount the turtle has turned right. The starting HEADING is 0.

Neil Rowe, Department of Computer Science, Stanford University, Stanford, CA 94305.

Random POLYs

Let's introduce random choices into POLY. For instance, "erasing" random lines of a complete POLY drawing. We can do this with a POLY (which I'll call ERASEPOLY) which makes a random choice between drawing and not drawing each time, repeating the following commands in sequence:

```
IF RANDOMCHOICE THEN FORWARD S
ELSE PENUPFORWARD S
RIGHT A
IF (HEADING MODULO 360) = 0 THEN STOP
```

where RANDOMCHOICE is randomly "true" or "false," HEADING MODULO 360 means the remainder when dividing HEADING by 360, and the last line is necessary to prevent overdrawing. Figure 5 is an example for A=60. A variant, which I call SKIPPOLY, is to just not bother tracing out some of the sides at all:

```
IF RANDOMCHOICE THEN FORWARD S
RIGHT A
IF (HEADING MODULO 360) = 0 THEN STOP
```

Figure 6 shows one for A=60.

Another idea is to rearrange the order of the sides of a POLY, which I call PERMUPOLY:

Figure 1.
POLY with S = 100, A = 90.

Figure 2.
POLY with S = 50, A = 60.

Figure 3.
POLY with S=100, A=144.

Figure 4.
POLY with S = 1, A = 1.

Figure 5. An ERASEPOLY with S = 100, A = 60.

Figure 6. A SKIPPOLY with S = 100, A = 60.

Figure 7. A PERMUPOLY with S = 100, A = 60.

Figure 8. IMPOLY with S = 100, S2 = 10, A = 90, F = 5.

```
IF RANDOMCHOICE AND (HEADING MODULO 360)
   IS NOT IN HLIST THEN BEGIN
FORWARD S
RIGHT A
PUT (HEADING MODULO 360) IN HLIST
END
```

where the BEGIN and END group commands that must be done together, and HLIST is a list of headings; note no "STOP" line is needed. Figure 7 is an A=60 example.

There are lots of interesting things you can discover about these figures. For instance, for some A, how many different figures are there (ignoring rotations and mirror reversals)? How many of the figures are symmetric (either "laterally" or "rotationally")? How many contain K lines, for some number K? What are the "family relationships" by which one can be built from others?

Clockwork POLYs

Rather than just not "doing anything" on certain steps of a POLY, as in ERASEPOLY and SKIPPOLY, one can move forward or turn right. When you do this regularly instead of randomly, this is a "clockwork" or "impulse" POLY, which works the way mechanical clocks work. The idea is to do POLY with a given S and A most of the time, but every so often move instead forward an S2 (in the side impulse POLY) or turn instead an A2 (in the angle impulse POLY). The effect is most interesting when S2 is very different from S or A2 from A. This can be done by the POLY-style repeated command group:

```
IF (HEADING MODULO (F * A)) = 0 THEN
   FORWARD S2
ELSE FORWARD S
RIGHT A
```

where HEADING again is the total amount that the turtle has turned right, and F is an integer "frequency" (number of steps between impulses). Figures 8-10 give examples of this IMPOLY.

The angle impulse POLY (AIMPOLY) is similar (only now we need a special counter variable K rather than HEADING):

```
FORWARD S
IF (K MODULO F) = 0 THEN RIGHT A2
ELSE RIGHT A
ADD 1 TO K
```

Figures 11-13 give examples.

Another idea is to have the S2 or A2 used more than just once in a while—half the time, say. (Which gives a "square wave modulation" POLY, using the terms of my previous

Figure 9.
IMPOLY with S = 10, S2 = -50.
A = 4, F = 36.

Figure 10.
IMPOLY with S = 35, S2 = -20,
A = 40, F = 25.

Figure 11. AIMPOLY with S = 50,
A = 45, A2 = -45, F = 5.

Figure 12.
AIMPOLY with S = 50, A = 50,
A2 = -50, F = 7.

Figure 13.
AIMPOLY with S = 7, A = 5,
A2 = 90, F = 96.

Figure 14.
DBLPOLY with S = 50,
S2 = 100, A = 60, F = 8.

article.) Side (S) and angle (A) changes turn out to look pretty similar, so I give only the first:

```
IF (HEADING MODULO (F * A)) < (F * A ) / 2
THEN FORWARD S
ELSE FORWARD S2
RIGHT A
```

Figures 14-16 give examples of this "double POLY" DBLPOLY.

Multiple Frequencies

Things start to get particularly interesting if we let the programs of the last section have more than one frequency. Like this EMBEDPOLY, extending IMPOLY:

```
FORWARD S
IF (HEADING MODULO (F1 * A)) = 0 THEN
    FORWARD S1
IF (HEADING MODULO (F2 * A)) = 0 THEN
    FORWARD S2
IF (HEADING MODULO (F3 * A)) = 0 THEN
    FORWARD S3
RIGHT A
```

With programs of this sort we can make "embedded POLYs" where there are several levels of detail in the figure, corresponding to each of the frequencies. Figures 17-20 provide examples. Note Figures 17 and 18 have the same angle and frequencies, illustrating the importance of the S "scaling factors" on the sides. Also note you can prove that the number of lines in an EMBEDPOLY is the least common multiple of the three frequencies and 360/A.

Region POLYs

So far we've written programs only in terms of FORWARDs and RIGHTs and not considered at all where the turtle happens to be. But we can define, say, certain parts of the turtle plane where the turtle behaves differently from other parts. For instance, suppose in half of the plane the turtle always moves forward S2, in the other half S2. (This can happen in the physics of charged particles moving in magnetic fields). This can be done by a LINEREGPOLY:

```
IF XCOORDINATE > 0 THEN FORWARD S1
    ELSE FORWARD S2
RIGHT A
```


Figure 15. DBLPOLY with S = 45,
S2 = -6, A = 25, F = 8.

Figure 16. DBLPOLY with S = 4,
S2 = 1, A = 5, F = 63.
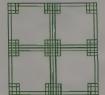
Figure 17. EMBEDPOLY
with S = 15, S1 = 75, S2 = 6, S3 =
85, F1 = 5, F2 = 25, F3 = 125, A = 90.

Figure 18. EMBEDPOLY
with S = 6, S1 = 120, S2 = 10, S3
= 30, F1 = 5, F2 = 25, F3 = 125, A = 90.
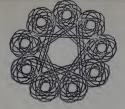
Figure 19. EMBEDPOLY
with S = 4, S1 = -5, S2 = 20, S3 =
100, F1 = 9, F2 = 200, F3 = 280, A = 10.

Figure 20. EMBEDPOLY with
S = 1, S1 = 10, S2 = 100, S3 = 24,
F1 = 5, F2 = 80, F3 = 140, A = 30.

where XCOORDINATE gives the current horizontal location of the turtle. Unfortunately, for most cases this figure is not closed (see Figure 21). This suggests making the region boundary a circular one. And note we can also do things to the turtle's heading (direction) when we reach a boundary:

```
FORWARD S
IF ((XCOORDINATE * XCOORDINATE)
   + (YCOORDINATE * YCOORDINATE))
   > (RADIUS * RADIUS) THEN RIGHT 180
ELSE RIGHT A
```

which turns the turtle around completely (CIRCREGPOLY; see Figure 22).

Or we can "bounce." Using the rule from physics that the angle of incidence equals the angle of reflectance, we can get, with a little mathematics, the turtle to turn right:

180 - 2 * (BEARING - HEADING)

when it bounces, where BEARING is direction the turtle is pointing in relation to the center of the region, which if you know a little math is:

BEARING = ARCTAN(XCOORDINATE - YCOORDINATE)

So we can bounce with the following BOUNCEPOLY:

```
FORWARD S
IF ((XCOORDINATE * XCOORDINATE))
   + (YCOORDINATE * YCOORDINATE))
   >(RADIUS * RADIUS)
   THEN RIGHT  180 - 2 * (BEARING - HEADING))
ELSE RIGHT A
```

See Figure 23.

Further Ideas

This article has just touched on a few simple ideas for making interesting geometrical shapes. If your interests are artistic, try combining some of the very different ideas in this and the previous paper which I haven't mixed in order to be clear.

There's no reason to restrict yourself to just POLY-type programs. This has made my descriptions simpler, but actually you can use any line drawing as a basis for these ideas. Just represent the drawing as a list of FORWARD-RIGHT argument pairs, and cycle through this list again and again.   □

Figure 21. LINEREGPOLY
with S = 2, S2 = 5, A = 5.

Figure 22. CIRCREGPOLY with S = 12,
A = 5, RADIUS = 100.
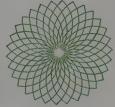
Figure 23. BOUNCEPOLY with S = 15,
A = 12, RADIUS = 100.

Logo Ideas

Robert Lawler

The following short features are the first in a longer problem solving observations from Advanced Technologies Inc. the ARPA data primarily on several years, they are recommended independence insomuch of which labelled to be stored in and of work.

What's A Powerful Idea?

Everybody knows what a grapefruit is and how to cut one in half. When you cut a grapefruit perpendicularly to the core, the cut face shows a pattern like a wheel, with axle, spokes and rim. A little energy and perseverance are all you need to dig out and enjoy the juicy meat from between the spokes. There must have been a time when you didn't know which way to cut a grapefruit. Did you find out how the hard way? What happens if you divide the grapefruit the other way, along the core? It is nearly impossible to get at the still buried meat, for the tough skin of the sections is an intact obstacle. The grapefruit looks pretty much uniform on the outside of its skin, but when you look deeper you can see there is a very specific and important organization that you must understand if you want to get at what's inside. This very simple, very concrete situation of cutting up a grapefruit provides a useful way to look at many other very troublesome problems. It points up the issue that how you analyze a problem, whether your analysis "goes with the grain" or "against the grain," can make a world of difference in how hard the problem is to solve.

The intrinsic thing that gives such an example power in thought is its simplicity, in the sense that given the perspective of the idea (embodied in the concrete example), the primary conclusions that can be drawn are obvious without long chains of arguments. That is, such ideas are elegant in the mathematician's sense. The extrinsic root of power is the example's fruitfulness, how well it can serve in helping you understand other problem situations by analogy.

Robert Lawler, Ph.D., Associate Director, Logo, P.O. Box 1551, Ansonia Station, New York, NY 10023.

The more powerful ideas you have the better. If you have only one possible model of a situation, you are a prisoner of a limited point of view. If you can interpret a situation in terms of several possible models, you can compare the fit of each to judge which is most appropriate. As a situation changes, some alternative model may come to better fit the situation than the one originally best. Could it be that the flexibility of mind we ascribe to "smart" people derives directly from their having a well developed stock of such powerful ideas?

Descartes introduces his *Discourse on Method* with the witticism that intelligence is the only thing in the world of which there is no scarcity, for although men may complain that they lack goods or information, no one complains that his judgment is inadequate. Why should this be the case? People don't know—or believe—that ideas can be powerful. The most powerful idea of all is that there are such things as powerful ideas. Those of us who lack this idea may struggle valiantly with problems, combing the world for information and our minds to make connections, without imagining that we need to formulate the problem through a fundamentally different model of the situation.

Do It Yourself

How do you make a Logo circle? Circles have something to do with centers and a thing called a radius, don't they? But the Logo turtle doesn't know anything about centers or a radius—all it knows is how to turn right and left and to move forward and back. What would YOU do if you were the turtle? You might do it yourself, try to make a shape that's a little bit like a circle by taking one step forward, turning a little, and doing both over again and again. If the path you follow has many turns, the shape of your path will be like the circle the turtle makes in this procedure:

TO CIRCLE
FORWARD 1
RIGHT 1
CIRCLE

When you do it yourself, you can understand better what the turtle is doing.

Suppose you made a picture of a bird and wanted to make it fly. How would you do it? Can you put yourself in the scene and write a "people-procedure" for your own flying? It must have something to do with flapping your arms and going forward. If you do it again and again, as in the circle example, your simplest people-procedure might be like this:

TO FLY
RAISE ARMS
LOWER ARMS
FORWARD SOME-AMOUNT
FLY

Does it seem childish to do it yourself? Grown up people never go around imitating turtles or pretending to fly, do they? Some *really* smart people do; they know that putting themselves in the scene, being a part of the action, helps them understand what's going on. One of the most famous scientists of modern times, James Clerk Maxwell, once imagined himself a super-ordinary tiny creature (he named it a "demon") who could control the flow of molecules in a gas by opening and closing a tiny window. Maxwell was not being childish by putting himself in the middle of the action. He was raising a fundamental challenge to the basic laws of physics. Give it a try when you have a problem to solve. Do it yourself. □

Teach Them Computing with a Pocket Computer

Antonio M. Lopez, Jr., Ph.D

Teach them how to compute, but teach them using fewer dollars. This statement incorporates the consensus of the panel on Computing Concepts in Elementary Schools at the 1981 National Educational Computing Conference. The panel was chaired by Margaret Christensen, Widener College and consisted of Jacques LaFrance, Oral Roberts University; Arthur Luehrmann, Computer Literacy, Inc.; Robert Taylor, Teachers College Columbia University; Daniel Watt, Logo Project; and Grayson Wheatley, Purdue University. It suggested that teachers spend more time showing students how to compute and less time on drill and practice and tutorials. The economic reality is that the Reagan administration has proposed a $9.9 million budget for science education for the National Science Foundation. This is down from $90 million, and Congress may restore only $35 million. The "grant well" for most schools has dried up.

No one will argue that the hardware costs for educational computing have not decreased since the introduction of microcomputers. However, it must also be realized that the vast majority of elementary schools throughout this country never had an academic computer budget previously.

Antonio M. Lopez, Jr., Ph.D., Mathematical Sciences, Loyola University, Box 51, New Orleans, LA 70118.

Minicomputers and mainframes were, for the most part, out of the question financially. So, where there was nothing before, there is now a request for $999 or more. Furthermore, that $999 request is for one minimal microcomputer system to teach 10 or 15 students. Can we achieve the goal of

> *Learning to program with a pocket computer is about as easy as learning to play with Big Trak.*

teaching children how to compute, while meeting the shrinking dollar problem?

Radio Shack announced the TRS-80 Pocket Computer in August 1980. Sharp introduced the DC-1211 a few months later, and Quasar released its Hand Held Computer at the end of 1981. To date, the educational potential of these machines has been largely ignored. They don't seem to have fancy graphics nor is there a wealth

of educational software support. However, a pocket computer can be used to compute and it provides a relatively inexpensive means of teaching programming.

Depending on the source, the TRS-80 Pocket Computer retails for $229.95 or less. There is also a printer/cassette interface for $149.95 and a cassette recorder for $79.95. Thus an elementary school student can have a computer, a printer, and a secondary storage device for less than $460. Compare this with the 4K Level I TRS-80 Model III with Line Printer VII and cassette recorder for $1163.90. There is a difference of over $700. (In all fairness, we must also note the availability of the Commodore VIC-20, which retails for $299.95 and the Sinclair ZX81, which sells for $149.95—Ed.)

But what can we teach using pocket computers? Truly, a great deal of applied mathematics and computer science. Part of the goal statement on computer literacy from the National Council of Teachers of Mathematics said that mathematics programs must take full advantage of the power of computers *at all grade levels.*

Included among the 11 topics singled out in the statement were drill and practice, problem solving, data retrieval tasks, and programming. The pocket computer can be used to teach all four of these topics and more.

141

```
10:"CLEAR":
PAUSE "HI, I
HERE":INPUT
"ENTER A NUM
BER":M:INPUT
"MATH LEVEL
1,2,3?":E
20:E=10^E+1:I
PAUSE "10 QU
ESTION DRILL
":T=0:N=M
30:PAUSE "MAIN
MENU":PRINT
"SHFT A) ADD
":PRINT "SHF
T B) SUBTRAC
T"
40:PRINT "SHFT
C) MULTIPLY"
:PRINT "SHFT
D) DIVIDE":
GOTO 30

100:"A"I=100:K=M
+1:IF K>10
GOTO 500
110:GOSUB 9001:A=
X:GOSUB 9001
B=X:C=A+B:S$
="+":GOTO
500
200:"B"I=200:K=M
+1:IF K>10
GOTO 500
210:GOSUB 9001:A=
X:GOSUB 9001
B=X:C=A-B:S$
="-":GOTO
910
300:"C"I=300:K=M
+1:IF K>10
GOTO 500
310:GOSUB 9001:A=
X:GOSUB 9001
B=X:C=A*B:S$
="*":GOTO
910

400:"D"I=400:K=M
+1:IF K>10
GOTO 500
410:GOSUB 9001:C=
X:GOSUB 9001
B=X:A=C*B:S$
="/":GOTO
910
420:IF T=THEN 4
10
430:GOTO 910
500:T=T+10*PRINT
"YOUR SCORE
IS ";T:"%"
510:INPUT "ANOTH
ER DRILL?":Y
$:IF Y$="YES
":GOTO 10
520:IF Y$="NO"
:GOTO 510
525:PAUSE "ANSWE
R YES OR NO"
:GOTO 510
530:PAUSE "BYE":
END

900:X=ABS (.4591*
7+X):X=23+X:
X=D: INT (X)=E
:RETURN
910:BEEP 1:PRINT
A$;S$;B$;" =
"?
920:"=":AREAD Z:
IF Z=ZLET T=
T+1:PAUSE "V
ERY GOOD !!!
":GOTO I
950:PAUSE "NO: T
HAT IS NOT R
IGHT":PRINT
A$;S$;B$;" =
";C:GOTO I
```

Listing 1.

```
PIG CHICKEN LEG
0.    18.    36.
1.    17.    38.
2.    16.    40.
3.    15.    42.
4.    14.    44.
5.    13.    46.
6.    12.    48.
7.    11.    50.
8.    10.    52.
9.    9.     54.
10.   8.     56.
11.   7.     58.
12.   6.     60.
13.   5.     62.
14.   4.     64.
15.   3.     66.
16.   2.     68.
17.   1.     70.
18.   0.     72.

10:PRINT "PIG C
HICKEN LEG"
20:FOR P=0 TO 18
30:C=18-P
40:L=4*P+2*C
50:PRINT P;"PI
",C;
"IL
60:NEXT P
70:END
```

Listing 2.

```
BISECTION
METHOD FOR
FINDING ROOTS
TRIAL ROOT
           -1.
FUNCTION VALUE
       6.85119623E-03
TRIAL ROOT
       -1.619140625
FUNCTION VALUE
       2.47573852E-03
TRIAL ROOT
       -1.618164063
FUNCTION VALUE
       2.9087173E-04
TRIAL ROOT
       -1.617675782
FUNCTION VALUE
       -9.0084634E-04
TRIAL ROOT
       -1.617919923
FUNCTION VALUE
       -2.55945769E-04
TRIAL ROOT
       -1.618041993
FUNCTION VALUE
       1.739811E-05
TRIAL ROOT
       -1.617980958
FUNCTION VALUE
       -1.18577955E-04
TRIAL ROOT
       -1.618011476
FUNCTION VALUE
       -5.033954E-05
TRIAL ROOT
       -1.618026735
FUNCTION VALUE
       -1.6219835E-05
TRIAL ROOT
       -1.618034364
FUNCTION VALUE
       8.3906E-07
ROOT=-1.61803436
4
```

```
FUNCTION VALUE
   -1.619140625
TRIAL ROOT
   -1.5
FUNCTION VALUE
   -0.25
TRIAL ROOT
   -1.75
FUNCTION VALUE
   0.3125
TRIAL ROOT
   -1.625
FUNCTION VALUE
   0.015625
TRIAL ROOT
   -1.5625
FUNCTION VALUE
   -0.12109375
TRIAL ROOT
   -1.59375
FUNCTION VALUE
   -5.37109375E-02
TRIAL ROOT
   -1.609375
FUNCTION VALUE
   -1.92871093E-02
TRIAL ROOT
   -1.6171875
FUNCTION VALUE
   -1.89208985E-04
TRIAL ROOT
   -1.62109375
```

```
10:PRINT "BISEC
TION":PRINT
"METHOD FOR"
:PRINT "FIND
ING ROOTS"
20:E=.00001
30:INPUT "ENTER
A ":A:"ENTE
R B ":B
40:X=A:GOSUB 90
0:F=X:X=B:
GOSUB 900:S=
F
60:IF SGN R<>
SGN (THEN 90
70:PAUSE "SAME
SIGNS; TRY A
GAIN":GOTO 3
0
90:C=(A+B)/2
91:BEEP 1
100:X=C:GOSUB 90
0:T=F1:F T=0
110:IF SGN T=SGN
R THEN 160
111:B=C
120:PRINT "TRIAL
ROOT":PRINT
C:PRINT "FUN
CTION VALUE"
:PRINT F
130:IF ABS (A-B)
>E THEN 90
140:PRINT "ROOT=
":C
150:END
160:A=C
161:GOTO 120
900:F=X*X*X-1
901:RETURN
```

Listing 3.

```
CONTINUOUS
DILUTION
PROBLEM
=============
MLS. STRENGHT
0.    1.
100.   9.090909
091E-01
200.   0.826446
281
300.   7.513148
005E-01
400.   6.830134
554E-01
500.   6.209213
231E-01
600.   5.644739
301E-01
700.   5.131581
183E-01
800.   4.665073
807E-01
900.   4.240976
105E-01
MILLILITERS=1000
SOLUTION=3.85543
289?E-01
```

```
10:PRINT "CONTI
NUOUS"
11:PRINT "DILUT
ION"
12:PRINT "PROBL
EM"
20:PRINT "=====
30:INPUT "INITI
AL CONCENTRA
TION":C
40:PAUSE "MILLI
TERS OF WATE
R"
41:INPUT "ADDED
EACH STEP":
M
50:N=1000/M
60:D=M/1000
70:T=0
80:PRINT "MLS.
STRENGHT"
90:FOR I=0 TO N
100:PRINT T1
110:C=C/(1+D)
120:T=T+M
130:NEXT I
140:PRINT "MILLI
LITERS":T
141:PRINT "SOLUT
ION":C
```

Listing 4.

Radio Shack's Math Drill is an available drill and practice package for the pocket computer. However, with a little effort a teacher could write something similar. For example, Listing 1 is a program that drills on the arithmetic of whole numbers.

A real plus in the Radio Shack product is an excellent game called Hi-Lo. This game gives a child eight chances to guess what number (between 0 and 100) the computer has chosen. After each guess the computer responds by indicating whether the guess was high or low with respect to the secret number. This game is an excellent vehicle for teaching children the technique of binary search. Why wait until they take a computer science course in college?

Problem solving has long been at the heart of the mathematical sciences. The PRISM study (Priorities in School Mathematics, 1981), conducted by the National Council of Teachers of Mathematics, indicated continued strong support for it in the mathematics curriculum of the 80s. However, only moderate support was given for having calculators and computers available for problem solving work.

Apparently, teachers are still not sure how to incorporate these computational devices into the process. New problem solving strategies are forthcoming (Piele, 1981), and they do not require large microcomputer systems. Modified versions of the Applesoft Basic programs presented by Piele for teaching problem solving with a computer work on the TRS-80 Pocket Computer (Listings 2, 3, and 4).

Creating and retrieving data files is relatively simple with the TRS-80 Pocket Computer. As part of its organization, there are 26 data memories. Each can hold eight bytes of either numerical or string data and be referenced in a variety of ways. The PRINT# instruction records the contents of data memories onto cassette tape. The INPUT# instruction transfers data recorded on tape to data memory.

The program in Listing 5 shows the use of the PRINT# instruction in the creating of a data file called VOCAB1. Retrieval of

Listing 7.

that file using INPUT#, and the displaying of the individual words for recognition practice is the purpose of the program in Listing 6. It should also be noted that once the data memories are loaded from tape, they remain in memory as long as the program is resident.

Learning to program with a pocket computer is about as easy as learning to play with Big Trak. Basic is in ROM, so the computer is ready to accept commands as soon as the ON key is pressed. Commands can be abbreviated in just about any fashion and still be understood by the computer. For example, a line entry containing P., PR., PRI., or PRIN. will have the abbreviation transformed to PRINT when placed

```
  5:CLEAR
 10:PRINT "CREAT
    ING DATA TAP
    E"
 15:PRINT "MARK
    COUNTER STAR
    T"
 20:PRINT "ENTER
     20 WORDS"
 30:PRINT "ONE A
    T A TIME."
 40:FOR X=1TO 20
 50:PAUSE "WORD
    #";X
 60:INPUT "ENTER
     WORD ";A$(X
    )
 70:NEXT X
 80:PRINT #"VOCA
    B1"
 90:PRINT "MARK
    COUNTER STOP
    "
100:PAUSE "PROGR
    AM COMPLETED
    "
110:END
```

Listing 5.

Listing 6.

```
  5:INPUT "LOAD
    A TAPE? ";Y$
  6:IF Y$="YES"
    GOTO 9
  7:IF Y$="NO"
    GOTO 30
  8:PAUSE "PLEAS
    E ANSWER YES
    OR NO"GOTO
    30
  9:CLEAR
 10:PRINT "CHECK
    COUNTER"
 20:INPUT #"VOCA
    B1"
 30:FOR X=1TO 20
    :PRINT "
    ";A$(X
    )
 35:NEXT X
 40:INPUT "DO IT
    AGAIN? ";Y$
 41:IF Y$="YES"
    GOTO 30
 42:IF Y$="NO"
    GOTO 50
 43:PAUSE "PLEAS
    E ANSWER YES
    OR NO"
 44:GOTO 40
 50:END
```

into program memory. (This is done with a compression code.)

Program editing is not as complicated as with larger microcomputer systems. Type-over editing is supported. Insertions and deletions can be accomplished on a program line by moving the cursor left or right (left or right arrow keys) and pressing the SHFT and INS or SHFT and DEL keys respectively.

These features allow the elementary school student to concentrate on the important task of computing and not be entangled in operating systems, syntax, and text editing commands. Sequence, selection and looping structures can all be taught on the pocket computer. The mathematical hierarchy of operation is the same as with any other microcomputer system, and some intrinsic functions are available.

To a nine year old, the distinctions between hardware and software become apparent while holding a computer in the palm of the hand and entering a program that was just developed. Students learn

> *Sequence, selection and looping structures can all be taught on the pocket computer.*

from first hand experience that the components (printer, cassette recorder, keyboard, and display) function as an interconnected system under the control of a stored program.

Even first graders can be taught to write sorting algorithms (Hedges, 1980-81). Listing 7 is a version of the program presented by Hedges. It was modified to use the TRS-80 Pocket Computer with printer/cassette interface to display the input and output of the numbers. The storage architecture of this computer allows for up to eight numbers to be sorted by this program.

Certainly, pocket computers have limitations, but then so do minicomputers and mainframes. Seeming disadvantages can turn out to be advantages. "In making the pocket computer so small, Sharp had to sacrifice some of the conveniences of a larger computer. The keyboard is so small you can only use one finger at a time or two if you set the pocket computer on a non-slip surface," writes Stephen Gray, contributing editor of *Creative Computing.*

Have you ever watched a young child work with a typewriter keyboard? Typing with one finger at a time is quite natural to them.

```
NUMBERS ENTERED
          7.
          9.
          6.
          1.
          4.
          2.
          8.
          3.
NUMBERS SORTED
          1.
          2.
          3.
          4.
          6.
          7.
          8.
          9.
```

```
  5:CLEAR
 10:INPUT "HOW M
    ANY NUMBERS?
    ";N
 15:PRINT "NUMBE
    RS ENTERED"
 20:FOR I=1TO N
 35:INPUT A(I)
 40:NEXT I
 50:FOR I=1TO N-
    1
 60:FOR J=I+1TO
    N
 65:BEEP 1
 70:IF A(I)<A(J)
    THEN 110
 80:T=A(I)
 90:A(I)=A(J)
100:A(J)=T
110:NEXT J
120:NEXT I
130:PRINT "NUMBE
    RS SORTED"
140:FOR I=1TO N
150:PRINT A(I)
160:NEXT I
170:END
```

Another seeming disadvantage might be the use of four operational modes (DEF, RESERVE, RUN, PRO). Actually, these modes can be used to distinguish handheld calculators from computers. Programs can be designed for the DEF mode to assign most of the commonly used functions of any calculator to the Reservable Keys (Rable, 1981).

Since larger computer systems use compilers to translate programs from high-level programming languages to machine language, they go through a compile step, linkage, and execution. The pocket computer can be used to introduce these concepts. The PRO mode can be likened to the compile step. While in the PRO mode, Basic instructions are entered into a buffer. These instructions may be in abbreviated form, but after the enter key is pressed, all abbreviations are expanded with spaces added for clarity; furthermore, they are copied from the buffer and placed in compressed form into memory.

The CHAIN statement is a program execution instruction. When it is encountered in a program, it causes a cassette tape to be searched for a particular program file; it transfers the program into memory and begins execution at the specified line

number. In a sense, we have a linkage step. Finally, the execution of programs or immediate calculations (similar to using a handheld calculator) can be accomplished in the RUN mode.

"Pocket computers represent a major turning point in the evolution of computers. . . . While the power these devices provide is not yet awesome, neither is it inconsequential. People who learn how to use these little machines effectively are going to have a big advantage over those who do not realize their capability. Perhaps even more important is that these people will be educationally equipped to take advantage of the 'super-pocket-portables' that will be available in just a few years," says Nat Wadsworth, Editor of the *PC Newsletter*.

And the "super-pocket-portables" are coming too. The Quasar brochure for their Hand Held Computer illustrates time sharing configurations using modems for data communication to the office, a micro printer for hard copy, TV adapter for larger screen use, and RAM expansion modules.

So why not give our elementary school children a head start on the future by teaching them how to compute on the pocket computers of today? □

References

Hedges, W., "Teaching First Graders How a Computer Can Sort." *The Computing Teacher* 8, 5 (1980-81), 24-25.

Piele, D., "How to Solve It—With the Computer." *Creative Computing* 7,8 (August 1981), 110-120.

Priorities In School Mathematics, National Council of Teachers of Mathematics, funded by the National Science Foundation, and directed by Alan Osborne. Available from the ERIC Documentation Reproduction Service, P.O. Box 190, Arlington, VA 22210.

Rable, R., "Pocket Calculator," *TRS-80 Microcomputer News*. 3, 8 (August 1981), 15-16.

"No, Arnold, Newton did not discover the law of gravity with an Apple Computer!"

145

Boston   Chicago   Houston   San Francisco

applefest

The Most Spectacular Extravaganza Ever... For Apple Users

At Applefest '82 hundreds of manufacturers, distributors and dealers will showcase the entire spectrum of Apple-compatible products including computers, components, peripherals, plug-in cards, publications, gifts, magazines, services, accessories and software for home, office and school.

Hands-on centers and multimedia presentations will demonstrate the newest applications for business, education and entertainment.

Seminars and workshops, conducted by the world's leading Apple authorities, will detail new uses to make your Apple more enjoyable and more useful than you ever imagined.

You'll meet thousands of other Apple owners and find the newest of everything for your Apple under one roof . . . and for sale at super show prices.

So if you use an Apple . . . or are thinking about buying one, you won't want to miss a minute of Applefest '82.

Ticket & Hotel Information

Send your check and a note indicating the specific show you wish to attend. Tickets and hotel information will be mailed back to you. Tickets can also be purchased at the show. Make all checks payable to Northeast Expositions Inc. 824 Boylston Street, Chestnut Hill, Mass. 02167 Tel: 617 739 2000.

Applefest/Boston
Fri-Sun May 14-16, 1982
Hynes Auditorium
Show Hours: 11AM to 6PM Daily
Admission: $6 per day or $10 for 2 days,
$15 for 3 days

Applefest/Chicago
Fri-Sun Nov 5-7, 1982
Arlington Park Racetrack/Exposition Center
Arlington Heights, Ill.
Show Hours: 1PM to 10PM Daily
Admission: $5 per day or $8 for 2 days, $12 for 3 days

Applefest/Houston
Fri-Sun Nov 19-21, 1982
Albert Thomas Convention Center
Show Hours: 1PM to 10PM Daily
Admission: $5 per day or $8 for 2 days, $12 for 3 days

Applefest/San Francisco
Fri-Sun Dec 3-5, 1982
Moscone Center
Show Hours: 1PM to 10PM Daily
Admission: $5 per day or $8 for 2 days, $12 for 3 days

Applefest is produced by Northeast Expositions Inc. and is sanctioned by Apple Computer Inc. and The Boston Computer Society.
Apple and Applefest are registered trademarks and service marks of Apple Computer Inc.

April 1, 1982

Reader's Digress

I WAS A COMPUTER WIDOW
PAGE 1

SOVIET THREAT TO AMERICAN COMPUTING
PAGE 14

PERSONAL COMPUTER DIET
PAGE 8

DIRECTIONS

For your own pocket size copy of Reader's Digress, cut out the six pages starting with this one, fold in half and you've got it!

Reader's Digress is an April fools parody. Any resemblance to people, companies, ads, logos, styles, etc. is intentional for satirical and humorous purposes only. Letters about suing us will be ignored.

First year: World's Most-Unread Magazine
Under 1 million copies in 1 language

of a CRT connected to a large wheel containing the entire ASCII character set (except for the stuff no one ever uses). The user simply rotates the wheel until the desired character is at the top, then depresses the single "fire" button, placing the letter on the screen. "What could be simpler?" asks Nullson. To further simplify the design, the unit has no on-off switch or other distractions.
—Zen Ildo

Muscleman Arnold Sureisbigger reports that he and his buddies prefer *Blyte*. He finds that carrying the magazine for an hour is the equivalent of six hours practicing the clean and jerk. Arnold also likes tearing each issue in half. "It's not as thick as a phone book," he says, "but it sure is a lot denser." Also, he finds that destroying phone books is undesirable because people want to look up phone numbers whereas destroying copies of *Blyte* is practically a public service.
—Virginia Dare

Computer-show entrepeneur Jerry Mildew, taking to heart complaints about poor conditions during his last few extravaganzas, has selected all new sites for his 1982 shows. Locations will include the Mare Inebrium, Siberia, Katmandu, and Calcutta. The season opener will be held in the La Brea Tar Pits, with the extra come-on of free parking for all.
—Norman Bates

Candy Corp. has announced production of the Model IV Level VII Revision D Coloriform Computer®. This low priced entry into the home and hobby field breaks new ground in graphics with the introduction of Plaxels®. These plastic pixels can be affixed, under user control, to any location on the monitor screen, giving an effective resolution of 1,000,000 by 1,000,000 despite the two inch size of each Plaxel®. Though the unit comes suplied with the standard Bugs and Daffy Character Set®, users will be able to purchase upgrades such as Enhanced Tweety®, Bold Elmer®, and the ever-popular Foghorn Font®. The entire system is non-toxic and washable.
—L. Da Vinci

Madge and the girls in the bridge club thought
Bob and I had the perfect marriage, but little did
they know that...

I Was a Computer Widow

By: Mabel Wingtips

Fiction in Real Life.

K nit one, purl two... oh, excuse me. My name is Mabel Wing-tips (not my real name). Until recently my husband Bob (his real name) and I lived happily here in Eden, Ohio. An accident at a quilting bee had prevented us from having children, but at least we had each other.

Bob and I lived what I like to call a New York existence here in Eden, dining and dancing until the wee hours and then getting mugged on the way home. After spraying Bac-tine on our wounds I would fre-quently read to Bob while he ran his fingers through my hair, tenderly scratching the dandruff loose from

As I approached the den I heard strange noises, and what sounded like two people talking. Thinking that my fear was making me imagine things, I started to open the door, the pistol in my right hand. What I saw gave me the gray streaks in my hair that make me look so distinguished. There was Bob on bended knee bowing before the Apple, and on the monitor, the face of a man who bore a striking resemblance to—*Ray Walston!* I screamed, and the pistol went off in my hand before

**REAL
LETTER
FROM
REAL
STUDENT**

[handwritten letter]

> Dear Sir:
> I am a little oracle student and I am a little eager about computers... we will be proposing... and some oracle... we will also make mobile... for you... we are students... we will thank you for your time.

I could take aim, I slumped to the floor, sleep mercifully pulling its curtain down on me.

Was it moments later—or days?—that I awoke in Bob's arms. Looking into his eyes, I could tell that he was mine again. My errant shot had destroyed the Apple, and Bob was no longer a pawn of Satan.

Where my story ends my message begins. Housewives everywhere—be on guard! When your husband talks about buying a computer, stop him! Or we may all be...computer widows.

my scalp. Our life was, in the words of that Francis Ford Coppola fellow, *A Paradise Now.*

But as with Adam and Eve, Paradise was not to be ours forever. One day Satan spoke to Bob, and tempted him with...*the Apple.*

I stared innocently enough. Bob brought the computer home, telling me that he'd use it to keep track of our finances, try a little program-ing and maybe play a few space games. I had never seen a computer before, but I had heard that they were very bright, and I didn't see anything wrong with it. Had I see the true evil in that maching, maybe I could have saved Bob's soul.

At first the changes in Bob's personality were small, almost un-detectable. After coming home from work he'd seem to rush through dinner as if someone... or some-*thing...* were waiting for him. Then he'd retire to the den (where he kept his Apple), saying that he had some important work to do. After a while the changes were not so small, Bob came home from work with a drool hanging down his chin. My worst fears were realized. Bob had become an agent of the Devil.

Where could I turn for help? My parents, my friends would never believe me. In desperation I decided to seek the help of the only other man I could have given my heart to—our Fortieth President, Ronald W. Reagan.

I'd followed Ronald's career closely ever since the Bonzo movies, watched him prepare for the greatest role of all. Feeling that there was some kind of psychic connection

between us, I sent him a telegram pleading for help, begging Ronald to order that all computers be de-stroyed. I received no reply, but I did notice a car with government license plates parked across the street for some time after that.

And so it was up to me to battle Beelzebub for Bob's soul.

Bob's condition worsened. He was fired from his job for becoming what his boss referred to as "a chiphead," and he now spent up to twenty hours a day programming on his Apple. His eyes had become glazed, and he spoke of using the computer as a bridge to "his master's domain." Bob refused to tell me what this place was like, except that t-shirts and Bermuda shorts were popular even in the winter. Looking at Bob, I realized that he was no longer the man I married, but a pawn of evil. I knew that there was only one way to save his eternal soul.

I would have to kill him.

The next afternoon, I went to Earl's Gun & Ammo. Under the "You Loo— We Shoot" sign was a case of pistols. I selected one with a small silver cross on the handle. For an extra $100 Earl even forgot to see if I had a gun permit, saying that in times like these a little lady needed all the help she could get.

It was dark by the time I got home. My hands shook uncontrol-lably as I opened the front door. I knew what I had to do, but that didn't make it any easier. I had to realize that this was not Bob I was about to shoot—just one of Satan's minions who happened to look an awful lot like Bob. Still, I knew my first shot would have to find its mark, as I could not bear to fire a second.

3

John: Portrait of a Teenage Arcaholic

His name is John. He is only 15, a typical American teenager. He could be your son, next door neighbor, brother or quite possibly, you. He is forever dressed in faded blue jeans, a loose T-shirt and ratty sneakers. Only, John is different. John is an arcaholic. This is his story.

By John E. Normal

Before I start I would like to thank my parents, for without them I doubt that I would have made it through my ordeal. You see, I'm an arcaholic. I don't know how I got this way, I used to sneak a few games of *Pong* now and then but nothing serious. Then one day my friends came and told me of the new game in town, *Space Invaders*.

I was a little afraid at first. I had heard stories of kids who got hooked on that game. Nah, that couldn't happen to me, I was too strong. I would only play doubles with a few friends, they could make sure I didn't overdo it.

It wasn't long before I started to crave a game by myself. I needed to play, sometimes at the most inopportune times. I found myself skipping breakfast and spending my milk money on the machines at the local arcade. My mother noticed a change in me and commented on my new "friends."

I guess it was about the time *Asteroids* came out that I realized something was wrong. I needed help desperately but could find no one.

I tried to kick the habit myself. I can still remember the pain. I would never carry a quarter unless it was absolutely necessary. I tried to avoid needless trips to the mall. I couldn't handle the pressure. My old friends began to ignore me. My palms sweated as I walked past the arcade. The "mall rats" taunted me, daring me to beat them at *Tempest*.

That was *my* game, color vector graphics and dazzling sound effects. I knew I could play them under the table. My hands were shaking with

Toward More Programesque Speech

April is the coolest month. The September graduates have earned their first promotions. Stock splits are in the air. Oxidation gently wafts onto the surface of disks spinning sublimely in their drives. Showers bring lightning to tickle power lines and erase all memories of figures and such.
— J.R. Proofcock

Words of Wisdom

Think of a computer as a small Cosmos, with millions and millions of parts.
— Karl Sagan in *National Enquirer*

Life is a duplex modem.
— Sartre in *Gui*

A software company is like a shark. It has to keep constantly moving forward or it will die.
— Wage Greed, quoted by Allen in *Annie Hall*

Signs of Strange Tangents

On a disk head cleaner: "Dust the byte."
— Mr. Clean

Message engraved on a bootstrap ROM: "That's a load off my mind."
— *Woman's Wear Daily*

Sign above the door of a mainframe company: "Nothing Personal."
— *Woman's Wear Daily*

Sign endorsing an electronic tennis game: "Over a billion served."
— Sagan in *Bon Appetur*

DP Descriptions

A computer humming away as if it could actually think.
— Mary Wollstonecraft

Tapes whirring round and round yet getting nowhere.
— Alice Lidder

RAM capable of holding trillions and trillions of combinations.
— Sagan in *Tiger Beat*

And those Zany Definitions

Data transfer: the real load down.
— E.W. Wilcox

Computer animation: zillion-dollar movie.
— Sagan in *TV Guide*

Repairing computers: wiring for money.
— Don Longcauer

☆ ☆ ☆

Have you a contribution for Toward More Programesque Speech? We don't care.

led from them to a large cylinder which was obviously for internal storage. Attached to the Nyet, I could see that one peripheral was already being produced. It seemed to be a kind of some sort, obviously for a printer. It was huge and scoop-like. They were truly familiar with our latest technology. This, beyond a doubt, was a tractor feed.

There was no time for subtlety when faced with such a threatening threat. I placed a time bomb on the prototype and headed out, back toward my reconnaissance point, secure in the knowlege that the Soviet threat had suffered a setback. Behind me, the explosion roared, almost as if fueled by some combustible liquid.

Special Condensed Programs Section

•••••••••

ADVENTURE

```
10 PRINT "YOU ARE AT THE BOTTOM OF A 20 FOOT PIT"
20 INPUT "NOW WHAT?",A$
30 IF A$="" THEN PRINT "I DON'T KNOW HOW TO ";A$:GOTO 10
40 IF X<.5 THEN PRINT "YOU WERE KILLED":END
50 IF X>.5 THEN PRINT "YOU WIN": END
```

ASTEROIDS

```
10 PRINT "HIT ANY KEY"
20 GET A$
30 PRINT "ASTEROIDS *** YOU LOSE":END
```

ELIZA

```
10 PRINT "HI, WHAT IS YOUR PROBLEM?"
20 INPUT A$
30 IF LEFT$(A$,1)="I" THEN PRINT "EGOTIST":GOTO 10
40 PRINT "YOU LOSE TOO":GOTO 10
50 PRINT "ARE YOU SURE":A$:".":GOTO 10
```

STAR TREK

```
10 X=INT (RND (1)*40):Y=INT (RND (1)*40)
20 PRINT "POSITION",X,Y
30 PRINT "PRESS 1 TO FIRE, 2 TO MOVE"
40 IF A$="1" THEN GOTO 10
50 IF A$="2" THEN GOTO 100
60 IF A$="3" THEN DESTRUCT KEY *** BOOM":END
70 IF RND (1)=.1 THEN GOTO 100
80 IF RND (1)=.2 THEN GOTO 90
90 GOTO 90
100 REM
110 IF X=.4 THEN GOTO 120
120 ...
KEYBYGOTORETURNPOPOPOROFGCLRSIRIGHTS$$$$$@@#@@MI$CLEARRESTOREOFGSI!$!
```

temptation. "Just one game," I can remember saying to myself. "I can handle it. I can stop anytime I want."

I dug deep into my pocket. I could feel the thin quarter at the bottom among the lint. Pulling it from the depths of my jeans I took the challenge. "One game, that's it, OK? I'll beat you and get the high-score at the same time," I said confidently. The mall rat just grimmed, knowingly.

The game started out pretty much even. My score slowly mounted along with my excitement. My mouth became dry and I began to sweat profusely. I had left my unworthy opponent in the dust, but that wasn't enough for me; I wanted no—needed the high-score.

Tension filled the air as the gap between my score and the high-score slowly diminished. My heart raced. It was in my hands. I could touch it. Now, now, it's mine.

"Look out," I heard someone shout from behind. My attention had been drawn for a split-second but in this game, that's all it takes. I watched with tears in my eyes as my final base was destroyed. I couldn't be, no, no. I was so close. I fell to my knees in disgust. I was drained and left without another quarter.

By this time I had lost all self-esteem. I walked up and down the row of machines begging the players to lend me a quarter—just one quarter. It was to no avail. In a last desperate attempt I raced from game to game, inserting my finger in the coin-return slot in the hopes of finding a forgotten quarter. No such luck. As I turned to walk out of the

place I heard a voice.

"Hey kid, want one of these?" whispered a man with a large toothy smile and a large stump wedged in his mouth. He held up a shiny quarter attached to the end of a thin wire. I had heard of people using them, but I had never imagined that I would be so hooked that I would need one. Nevertheless, I was desperate.

"Sure, but what will it cost me," I said nonchalantly.

"Nothing. I want you to have it. I know that you would do the same for me. This one's on the house. Here, have fun." He tossed it to me and I stretched my hands out to catch it. It seemed to spin in the air forever, finally I grasped hold of it. I looked up to thank my new-found friend, but he was nowhere to be seen. Oh well, to the games.

With this device in hand I marched back into the dark smoke-filled video game room of the arcade. I looked around nervously for the manager. He was nowhere in sight. I decided to give the thing a try.

I positioned myself in front of the *Defender* machine and peered around once more. The coast was clear. I dropped the quarter into the slot, keeping a good grip on the end of the wire. Click, the quarter fell deep into the machine's innards. A credit registered on the screen.

I slowly pulled the wire up. Another click, another credit. I expected bells and whistles to go off like they did in the films I was shown as a child. Nothing bad happened, my face cracked a devilish smile. Again the quarter fell and again a credit magically appeared on the screen. This went on until I

had amassed quite a number of free games.

Now it was time to play. My knuckles were white as I clutched the joystick and positioned my fingers on the buttons. I was ready. I played my heart out. I can't even remember what happened after the eighth or so game, it all becomes a blur.

I remember a guy who handed me a card and told me he was my friend, his face wasn't too clear in my mind, but I was to meet him again under different circumstances. All I know is that I woke up outside the mall leaning on a garbage can. People passed by me whispering comments to themselves. The mall was closed and all the lights were out.

I looked at my watch, the arcade had closed two hours ago. I realized that I must have overdosed and passed out. My head ached as I played a game of *Pac Man* in my mind. Everything started spinning, faster and faster. Aliens everywhere, I couldn't fire, my base wouldn't fire! Mother ships flew by untouched, centipedes were swarming around my carcass as falcons swooped down from the sky. No, stop, stop.

I remember waking up in a room lying in a bed with my hands and legs strapped to the sides. My parents were there looking over me. God, I never felt so glad to see them. My mother was sobbing and my father had a stern look on his face, personally I felt spent. There was a stranger in the room. As I came to, he began to speak:

"Do you remember me? I gave you my card at the arcade. You really should have listened to me then. Anyway, my name is Peter Goodguy, I've come to help you."

"Who the heck is this guy," I asked myself. "What does he mean he's going to help me. Help me do what?"

"I'm from A.A., Arcaholics Anonymous, no doubt you've heard of us. Your parents called me right after they found you. You're in bad shape young man." As if I didn't already know that. "I know how you feel, I too was once an arcaholic but with the help of some of my friends that you're going to meet. I beat the horrible disease.

Well, I could go on to tell you how they helped me kick the habit, but I hope you have gotten my message by now. *It can happen to you.* I never dreamed that some day I would join the ranks of arcaholics nationwide. A game here and there can't hurt but don't get trapped like I did. It's a hard and dangerous path on the road to destruction that will leave you quarterless. Next time you go to the arcade, prove you're a man. Don't play that "one for the road," walk away like a true hero. It may save your life.

John's story is a true one. The names were changed to protect the identities but the tale you have just read is fact. If you or someone you know is an arcaholic, or you would just like more information on what Arcaholics Anonymous can do for you, please write for their free booklet.

Arcaholics Anonymous Booklet
999 Attory Lane
Spendit, N.Y.
Hot Line: (212) 246-4200

my mind went back to the crash course on computing. I had been thoroughly trained in the latest and best equipment that free technology could produce; all aspects of such wonders as The Imagination Machine, the elegant SOL, and Britain's marvelous Sinclair were etched in my mind. I had read every available issue of *Doctor Dobbs Journal* and, to complete my expertise, spent an intense week studying *Katie and the Computer*. Computers dwelled in my mind like bats in a dark cave. I was ready.

Could they do it? I wondered. Were the rumors true? I reached the window, like an entrance to a scary fun house, and my mind returned to early reports; the Soviets had infiltrated the industry, but, true to their atheistic nature, had been unable to master our methods of production. When all seemed as gloomy as dusk in a coal mine, they had stumbled on a method they could grasp. The technical details were beyond me, but in essence, the plan called for an exact duplication of methods used by one of the most successful companies in the free world. It was a six-step plan.

1. Announce production.
2. Promise full supplies to all state stores.
3. When the moment was ripe, run full-page adds in *Pravda*.
4. Begin designing the system.
5. Begin production on a limited scale.
6. Change the design and repeat the process.

This dastardly threat could crush our superiority and even, dare I say it, strike a blow to free-world economy. My heart was in my mouth as I broke into the plant. I swallowed, then switched on my pocket flash, illuminating the room like a small atom bomb. Before me I saw what could only be the Commorad0 Nyet. Mixed in with my awe was a tinge of relief as I realized they had not yet mastered the miniaturization process. To either side of the hulking mainframe (which was painted a dull but typical red) were huge circular objects. My shock returned as I remembered something from my ingrained training. *These were hard disks!* The storage capacity had to be massive. Each disk was broken into tracks and sectors by deep zigzagging indentations in a treadlike pattern. The disks were attached in tandem to an axle that ran beneath the mainframe. A series of bolts held each disk secure to the rim with the same tenacity with which a drowning man grasps at a floating piece of jetsam. I slipped quietly closer and lifted the hoodlike cover to the internal workings. baring the soul of the Nyet to Western eyes. Wires and mechanical devices ran throughout the interior. like rivers on a map of some wet place. Leaning again on my training, I recognized the CPU and data bus. A set of twelve chips slanted, six on a side, into the mother block. Wires

By Kenny Hissenger, as told to Rene Dusk

The Insufferable SOVIET THREAT to American Computing

Condensed from Road and Tank Magazine

A brave intelligence officer leaks word of the insidious Commradd Nyet, the Red Menace to our computing superiority.

I t was a dark midnight in far Bulgorostavskia as I trekked stealthily through the sodden snow toward the glimmering light ahead of me in the untrodden snow. There it loomed, Russia's proudest thrust at our technological superiority, the SMAP (Soviet Microchip Acceleration Plan) Building. Though such a villainous thought defies belief, the Soviets had obtained vital information relating to microprocessor technology (see *The Schoolteacher Spy,* Feb. 1979-Ed.), including complete plans for a small computer, and vital information relating to every aspect of the booming industry. Johnny Red knew we had something good, and he wanted a piece of the pie.

I was on a covert mission to secretly discover how far the perfidious program had progressed, and, if possible, to undermine their operation. As I neared the building, which looked like a dollop of steel meringue on a plate of vanilla snow,

The Personal Computer Diet: Key to Health and Happiness

By U.R.A. Fatpig

America is a nation of dieters. *Everyone* wants to lose at least ten pounds. While half the planet is malnourished, we Americans can't wait to stuff our faces at every meal. But hey, that's the way we are, we're not going to change, so we just look for ways to stuff our faces without getting fat.

America is also a nation of fads. Remember the hula hoop? Swallowing goldfish? Disco? And now we have the Rubik's Cube.

And so, it's only natural for us to be *fad dieters*. So far we've had the Scarsdale Diet, the Atkins high-carbohydrate diet, Weight Watchers, the Pritikin diet—and let's face it, none of them has worked. Oh sure they've been good for a couple of days, but on the third day

you're halfway through one of those giant Hershey bars *before you even know it*.

But now, along comes Dr. Robert Tompower with a diet for the 80's: the Tompower Personal Computer Diet.

Dr. Tompower, a former snake oil salesman, is cafeteria food director of the University of California at Hoboken. He was kind enough to sit down with me recently to discuss his new diet.

"One Friday afternoon I was visiting a friend of mine who works in the computer room at U.C.H. I'd brought along a six pack of Y'ago, and we were getting pretty drunk.

"After I mistakenly made what might be considered a rude comment about his wife, my friend

Humor In Univac

In the last few years I've built up a big following by writing Adventure games. Well, the fans kept on asking for more, and they want each one to be harder. Listen, it isn't easy finding time for all this between posing (for ads and doing endorsements. Anyhow. I finally came up with the answer. I wrote one with no solution. There's no way anyone can win. That should hold them for a while.

—Contributed by Scott Free

I used to work in the computer room of a really big company. We worked hard and needed a few laughs just to break the monotony of really working hard. Anyhow, we had a few really fun games in the system, even though we really weren't supposed to use them. But there was one called "Bunny." And it sure didn't print a picture of an animal, if you know what I mean. But one day, after working really hard, we decided to have some fun. So we changed all the data so it did print a picture of a bunny. It was great fun and really didn't harm anyone.

—Contributed by a Wild and Crazy Guy

My friend Chuck did field, service on large systems, and he told me this story which he swears is true. This one client wanted his system to run the figures in columns instead of rows. So Chuck turned all the monitors sideways. Ain't that a scream?

—Contributed by Eddie Teller

A while back I wrote the best simulation game ever written for small computers, and it has sold very well, even though no one can pronounce its name. Then last year, I got a new job. Can you imagine my surprise when I discovered that my boss had written a program a lot like mine, and he'd done it years earlier. Boy was my face red!

—Contributed by Anonymous. Great Gorse. NJ

I used my Sinclair and a modem to embezzle a million dollars from a bank. Not funny, but I had to brag about it somewhere.

—Contributed by someone in Argentina

picked me up and threw me across the room like the proverbial rag doll. Landing face-first on a PDP-11, I was shocked to find that along with a rather alarming number of my teeth I had also swallowed a small piece of the computer. I was absolutely amazed to find out how good it tasted. Now more importantly, later tests found that an entire PDP-11 contained only 1800 calories!"

This, then, was the first step in the drive toward the Tompower diet.

"But I was still only halfway there. After all, do Mr. and Mrs. Average American have access to a PDP-11? No way, Jose. And so, after many hours and even days of testing, I came up with the Tompower Personal Computer Diet. It's a low fat, high protein diet, and it's a program (ahem) you can stick with."

end of her rope, begging for help. Miss Piggy was only five foot two but weighed 245 pounds. You know, a real crime of humanity. But after one month on the Tompower diet she weighed only 150 pounds and was up to The five foot six.

But what is the Tompower diet?

"Well the regimen varies from person to person, of course. Take Miss Piggy, for instance. From August 15th to September 15th, all she ate was a TRS-80, a few components of an old Compucolor we had lying around, and ten glasses of water. Like I say, it's a low fat, high protein diet."

So far, I had heard only Dr. Tompower's claims. And so, I went out looking for satisfied dieters, and that's exactly what I found.

Hollywood funnyman Dom DeLuigi feels that he no longer has to be fat to be funny as a result of the Tompower diet. "I feel that munching Atari's has added years to my career" says DeLuigi.

Famous dieter Joan Nightichi told me: "I used to be a real tub of lard. But now, with Dr. Tompower's diet, I look and feel twenty years younger! My new motto: 'An Apple a day keeps the blubber away.'"

Wayne Apreski, hotshot scorer for hockey's Edmonton Sealpups, also goes for the computer diet. "Growing up in Moosejaw, I had no idea what a computer even was. Now, after every game I go for a TRS-80 and a side order of Super Invaders."

Dr. Tompower's diet even influenced the design of the new IBM Personal Computer. "We wanted a computer the calorie-conscious American could sink his teeth into."

says IBM chief Avery Tower. "Dr. Tompower really turned around our way of thinking. The new IBM Personal Computer is almost 300 calories lighter than its closest competitor."

Dr. Tompower tells me that he'll soon be coming out with Personal Computer T.V. Dinners, and that unlike the Weight Watchers offerings, you may even come back for a second dinner. Bon appetit!"

As you no doubt know by now, Dr. Tompower was shot and killed by a U.C.H. scullery woman just as this issue went to press. Unfortunately, we had paid the author a rather large advance and decided to publish the article anyhow.

DRAMA IN REAL LIFE

A low-cal, candlelight dinner is a beautiful way to say 'I love you,' 'Bon voyage' or 'Congratulations, Graduate!'

THE NEW YORK ACADEMY OF SCIENCES · 1817

The President and the

Board of Governors of

THE NEW YORK ACADEMY OF SCIENCES

cordially invites

Dr. Crtive Computing

to apply for membership.

Table 1.

Model	Calories
TRS-80 Model 1	1200
Apple II	975
VIC-20	350
Atari 800	800
Atari 400	400
IBM	325
ZX81	10

Prepare Your Child for the Real World
with great learning aids from Toxic Instruments

Toxic Instruments makes learning what it should be....Fun.

Slice and Tell explores basic biology.

Hit and Kill makes physics fun.

Cut and Push teaches basic pharmacy, metric conversions and percentages.

Point and Squeal shows communications concepts.

SLICE AND TELL

BUSINESS and UTILITIES

MAILMASTER

SORTIT

PERSONAL FINANCE SYSTEM

FAMILY BUDGET

INFOLINE

TEXT EDITOR II

INFILE

FINDIT

SHOPPING LIST

TAX OPTIMIZER

UTIL

TURNKEY AND MENU

STOK RAID

EDUCATION

HODGE PODGE

TEACHER'S AIDE

TEACHER'S AIDE

ORDERING INFORMATION

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

STATISTICS and ENGINEERING

DIGITAL FILTER

DATA SMOOTHER

FOURIER ANALYZER

TFA (Transfer Function Analysis)

HARMONIC ANALYZER

REGRESSION I

REGRESSION II

REGRESSION III (PARAFIT)

MULTIPLINEAR REGRESSION (MLR)

ANOVA

BASIC SCIENTIFIC SUBROUTINES

ROOTS

ACTIVE CIRCUIT ANALYSIS (ACAP)

LOGIC SIMULATOR

NUMBER/BIN/OCT/HEX

STATFONT

STATTEST

ABOUT DYNACOMP

CIRCLE 138 ON READER SERVICE CARD

THREE-DIMENSIONAL PLOTTING

Peter Lee Poon-Hang

My math homework that day was one of the most unpleasant assignments I could imagine; not hard, just extremely unpleasant. That was because the homework included ten 3-D equations to be graphed complete with bizarre functions and weird ranges. I had no idea how the finished graphs would look.

Being just a tiny bit lazy, I turned to my trusty Apple for help. I could not find a program versatile and powerful enough to suit my needs, so I had to write my own.

I spent quite a few hours on this program, but it was definitely worth the effort. It has helped me in plotting graphs, and I'm sure it can help you too.

About The Program

Remember those horrible equations $Z=F(X,Y)$? Remember how hard it is to graph them? With this program, all you have to do is plug in the equation, select the range of the graph you want to see, and the computer will do the rest.

One of the most important features of the program is that it allows you to select maximum and minimum ranges for X, Y, and Z. This is particularly useful if you want to enlarge a graph on the screen or look at a portion of the graph away from the origin.

Peter Lee Poon-Hang, 475 Purisima Ave., Sunnyvale, CA 94086.

Let's say you want to graph $Z=\sin X + \sin Y$. If the range is from -100 to 100 for X, Y, and Z, you probably won't be able to see any activity. But if you can select the range from -PI to PI for X and Y, and from -2 to 2 for Z, the graph becomes very clear.

The program also allows you to select the accuracy you need for the graph.

The program also allows you to select the accuracy you need for the graph, so you can see a less detailed version of the graph in a very short time, or a very detailed version in a longer period. There is also a choice of grid width, which simply asks for the number of units between grid lines.

How It Works

When we humans plot graphs manually, we usually plug in values, find a few points on the graph and fill in the rest. This is essentially what this program does, but instead of plugging in to find a few points, the computer finds many points and omits the "filling in" part (computers aren't known for having good imaginations).

The program plugs a large number of Xs and Ys into the equation, finds Z, converts these numbers into dots and plots them (more on this later). It then connects the points just as we humans do.

Lines 10 and 20 print the title and check to see if the user has entered the equation. This is done by using the PRINT and Input statements.

Lines 25-40 request the grid width and accuracy needed for the graph. Again, this is done using Print and Input. The If statement is for defaulting—if the user simply presses Return, the string variable A$ will be null, thus equal to "", and the variable will be assigned the default value. If the user chooses any value other than the default, A$ will not be equal to null, thus the variable will be set to the VAL of A$.

VAL is a command which converts a number represented by a string into the actual number. The reason for inputting first to a string then assigning the VAL to the appropriate variable is that when only Return is pressed on the input of a numeric, an error occurs and the user must re-enter. With the string as a buffer, just pressing Return would still be acceptable.

"How the heck do you get a three-dimensional figure onto a two-dimensional Apple hi-res screen?"

Now the program jumps to a subroutine starting at line 500 which is responsible for obtaining the parameters for the graph. There is no real reason why this is a subroutine except that some people like to put little-used sections near the end of the program and sections that are used

often near the beginning to save a little time.

Why does this save time? Because whenever the computer goes looking for a certain line, it scans all the lines from the beginning until it finds a match. By putting a line which is often referred to near the beginning, there will be less scanning time.

The section inputs the data by first drawing a representation of a 3-D plane on the screen. It then positions the inputs near the appropriate side of the plane so that it is easy for the user to visualize what the program is asking for. After that, lines 580 and 590 take these parameters and figure out the ratio between each dot on the screen and each unit on the graph. This is done by subtracting the lower limit from the upper, then dividing

by the number of units allowed on the screen. This data is then multiplied by the numbers generated by the computer to get the proper pixel on the screen to light up.

Line 590 calculates the distance from the graph to the origin. If the user chooses to see the graph around the origin this number will be zero. However, if the user wishes to see a part of the graph away from the origin, this number will be positive or negative and added to the numbers generated by the computer. This makes the computer think that it is plotting near the origin.

In line 60, the computer goes into graphics mode and the POKE -16302,0 sets the full-page graphics mode instead of the "graphics and four lines of text at the bottom" assigned by an HGR. Next comes a series of FOR-NEXT loops which systematically generate values and plot them.

(For those of you who may not be very familiar with Basic yet, a FOR-NEXT loop tells the computer to perform a set of commands between the FOR statement and the NEXT statement a given number of times.)

Nested FOR-NEXT loops are also allowed, which means that one FOR-NEXT loop can be wrapped around another. The computer first holds X constant and scans Y which is appearing across the screen. This is done by wrapping a FOR X-NEXT loop around a FOR Y-NEXT loop. The parameters for the loops are the limits specified by the user. The numbers are then multiplied by the ratio computed earlier to get the right scale on the screen. The step part would be the grid width multiplied by the ratio number. This always results in a stepping of a grid width value chosen by the user between each grid line.

One thing to remember about the FOR-NEXT loop is that it sometimes skips the upper limit because the steps do not add up to a number exactly equal to the upper limit. Thus the computer stops at the value closest to the upper limit. This is not acceptable for this program because it causes the grid lines not to meet and makes the graph look very poor. The solution is to assign the variable to the upper limit and go through the steps one last time after the FOR-NEXT loop.

There are two main subroutines that are called within the loops. The subroutine from 1000 to 1042 holds the equation of the graph, computes Z for every X and Y given, then multiplies them by the ratios, adds them to the variables that hold the distance from the origin, and finally finds the appropriate pixels on the screen.

This subroutine in 1045 and 1050 actually plots the dot. The reason these are separated is that each time the computer plots a point, it stores it. The next point is

```
10  TEXT : HOME : HTAB 14: PRINT "3-D PLOTTING": HTAB 14: PRINT "----------
    --": PRINT : PRINT
20  INPUT "EQUATION Z=F(X,Y) IN LINE 1000 ?";A$: IF  LEFT$ (A$,1) = "N" THEN
    PRINT : PRINT "THEN I SUGGEST PUTTING IT IN!": END
25  PRINT : INPUT "GRID WIDTH: 10 ";A$: IF A$ = "" THEN GW = 10: GOTO 35
30  GW =  VAL (A$)
35  INPUT "ACCURACY(1-FINE,10-POOR): 5 ";A$: IF A$ = "" THEN AC = 5: GOTO 5
    0
40  AC =  VAL (A$)
50  C$ = "----------------------": B$ = " " : / "  C =   SQR (2) / 2: GOSUB
    500: DX = 96: DY = 32
60  HOME : HGR : HCOLOR= 3: POKE  - 16302,0
70  FOR X = XU TO XL STEP - GW = XM: GOSUB 800
80  FOR Y = YL TO YU STEP AC * YM: GOSUB 1000: GOSUB 1045: NEXT : Y =
    YU: GOSUB 1000: GOSUB 1045: NEXT
90  X = XL: GOSUB 800: FOR Y = YL TO YU STEP AC * YM: GOSUB 1000: GOSUB 1045
    : NEXT
95  FOR Y = YL TO YU STEP GW * YM: GOSUB 900
100 FOR X = XL TO XU STEP AC * XM:K = (X + TX) / XM * C
110 GOSUB 1000: GOSUB 1045: NEXT : K = (XU + TX) / XM * C: GOSUB 1000: GOSUB
    1045: NEXT
115 Y = YU: GOSUB 900: FOR X = XL TO XU STEP AC * XM:K = (X + TX) / XM * C:
    GOSUB 1000: GOSUB 1045: NEXT
120 X =  PEEK ( - 16384): IF X < 127 THEN 120
130 TEXT : HOME : PRINT : PRINT : PRINT "PLOTTING COMPLETED.": PRINT : INPUT "WANT
    TO SAVE THIS PICTURE?";A$: IF  LEFT$ (A$,1) < > "Y" THEN  END
140 INPUT "NAME FOR THIS PICTURE:";A$
150 PRINT  CHR$ (4);"BSAVE ";A$;",A$2000,L$1FFF": END
500 HOME : COLOR= 10: VLIN 1,30 AT 15
505 VTAB 7: HTAB 16: PRINT C$: HTAB 15: PRINT B$: HTAB 14: PRINT B$: HTAB
    13: PRINT B$: HTAB 12: PRINT B$: HTAB 11: PRINT C$: VLIN 14,20 AT 15
510 VTAB 6: HTAB 20: INPUT "X MIN: 0 ";A$: IF A$ = "" THEN XL = 0: GOTO 52
    0
515 XL =  VAL (A$)
520 VTAB 9: HTAB 2: INPUT "Y MIN: 0 ";A$: IF A$ = "" THEN YL = 0: GOTO 530
525 YL =  VAL (A$)
530 VTAB 9: HTAB 24: INPUT "Y MAX: 180 ";A$: IF A$ = "" THEN YU = 180:
    GOTO 540
535 YU =  VAL (A$)
540 VTAB 31: HTAB 14: INPUT "X MAX: 130 ";A$: IF A$ = "" THEN XU = 130:
    GOTO 550
545 XU =  VAL (A$)
550 VTAB 2: HTAB 18: INPUT "Z MAX: 80 ";A$: IF A$ = "" THEN ZU = 80: GOTO
    560
555 ZU =  VAL (A$)
560 VTAB 16: HTAB 18: INPUT "Z MIN: -80 ";A$: IF A$ = "" THEN ZL =  - 80:
    GO TO 580
565 ZL =  VAL (A$)
580 ZM = (ZU - ZL) / 160:XM = (XU - XL) / 130:YM = (YU - YL) / 180
590 TX =  0 - XL:TY =  0 - YL:TZ =  - 80 * ZM - ZL: RETURN
800 X = (X + TX) / XM * C:Y = YL: GOSUB 1000:U = I:V = J: RETURN
900 X = XL:K = (X + TX) / XM * C: GOSUB 1000:U = I:V = J: RETURN
1000 Z =  DX + X + TY) + VM : VM - K:J = DY - (Z + TZ) / ZM + K: IF 1 < 0 THEN 1
     = 0
1020 IF 1 > 279 THEN 1 = 279
1030 IF J > 191 THEN J = 191
1040 IF J < 0 THEN J = 0
1042 RETURN
1045 IF (U = 0 AND I = 0) OR (U = 279 AND 1 = 279) OR (V = 0 AND J = 0) OR
     (J = 191 AND V = 191) THEN U = I:V = J: RETURN
1050 HPLOT U,V TO I,J:U = I:V = J: RETURN
```

then plotted from the last one.

This is the method used to connect the dots of the graph. The only problem is that whenever the program jumps to the next grid line, it remembers the last point on the last grid line and connects them. That's why each time the outer loop is incremented, it must generate a new point and store it to cancel out the last point on the last line.

Now one might ask, "How the heck do you get a three-dimensional figure onto a two-dimensional Apple hi-res screen?" It turns out to be rather easy. In the translation, all you have to worry about is the X-axis because the Y in 3-D exactly corresponds to the X in 2-D and the Z corresponds to the Y in 2-D. Usually the X-axis is drawn at a 45-degree angle between the Y and Z axes, so all you do is subtract X*cos45 from Y and add X*sin45 to Z.

The last part of the program offers the option of saving the picture to disk. As you know, the Apple uses "bit-plot" graphics for hi-res screens. This means that each dot on the screen is "mapped" into a certain bit in memory. The organization of the location-to-dot correspondence is rather confusing, but the whole hi-res screen page 1 lies between memory locations hex $2000-$3FFF.

For example, when you do an HGR, the computer really just sets all the locations within the hi-res page to zeros (which represents black on the screen). If you set a certain bit within the memory range to a one, then the corresponding pixel will light up on the screen. It is really a

> **There are a few techniques in the program that beginners might find useful.**

little more complicated than that, but this is the basic idea. So, for saving the graph to disk, all you do is BSAVE name,A$2000,L$1FFF which tells the computer to save the range of memory from $2000 to $1FFF to disk under a given name.

Techniques

There are a few techniques in the program that beginners might find useful to include in their programs. They are included in the set-up portion. One is "screen-oriented data entry" and the other is "default data entry."

It is often useful to input data in the form of a chart or a table rather than line-by-line because it is much easier and more pleasant for the user. For example, when asking for the dimensions of a box, one might first draw a box and ask for the appropriate length right next to each leg. This is screen-oriented data entry, and can decrease the possibility of the confusion for the user.

"Defaulting" means that on an input, the computer prints out the most common answer and all the user does is press Return if the answer is what the computer has printed.

If the answer is different, the user just types in his response. Warning: when defaulting numbers, you must use string input because pressing Return on an Input X generates an error. The solution is to get a string and VAL it.

One last note: Functions such as COS, SIN, LOG, etc. take a great deal of time and slow your program down. So, if you can, perform them only once and set the results to variables so the computer can look up a number instead of generating it. □

HUNTINGTON COMPUTING

Spring Specials

Zenith 12" Green Screen Monitor . . . **$109**

Epson MX-100 Printer **$750**

Maxell Disks (10) **$36.99**

Huntington Cleaning Kit **$24.99**

Maxell L-750 Video Tape **$16.99**

Understand Yourself **$24.99**

Elephant Memory Disks w/hubs (10) . . **$24.99**

Beer Run **$19.99**

Dark Forest **$19.99**

Mt. Computer Music System **$349**

Superscribe II **$99.99**

Hodge Podge **$17.49**

Word Handler **$219.99**

Select **$523.59**

Chinese Lessons **$26.09**

Hebrew II **$50.99**

Improved Registration for Line Graphs

Ian A. Morton

When letters or diacritical marks are used to represent line graph datum points on a monitor screen, the graph is likely to be coarse and unrefined. This is so because the characters are formed out of a fairly large dot matrix. Moreover, with such gross registration, only about 60 columns and 25 rows can be displayed.
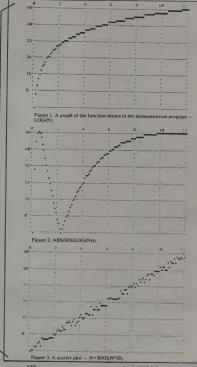
I frequently want to plot graphs requiring 100 or more columns and 40 or more rows, so I wrote a Basic routine for my OSI C2-8P which effectively doubles both the number of rows and the number or columns normally available. As presented here, the program will provide graphs with 50 rows and 120 columns.

The routine uses standard graphic symbols. These are shown in Table 1 along with their OSI ASCII codes.

ASCII NO.	GRAPHIC
154	▬
155	▬
165	▪
166	▪
167	▪
168	▪
169	▪,
170	▪▪

Table 1.

The graphing procedure begins with the taking of datum points in pairs, rather than one at a time. For example, if the first value of a pair can be assigned the lower left sector of an 8 x 8 dot matrix, and the second value of the pair of the upper right sector of the same matrix, then we may print CHR$(170) to represent that pair. With this procedure, a single 8 x 8 matrix may be packed with either one or two 4 x 4 graphic blocks, and registration will be doubled both horizontally and vertically.

Listing 1 shows a sample main program.

Ian A. Morton, 1500 Goodrich Avenue, St. Paul, MN 55105.

Figure 1. A graph of the function shown in the demonstration program — LOG(N).

Figure 2. ABS(SIN(LOG(N))).

Figure 3. A scatter plot — N + RND(N*10).

An array RS(120) (RESULTS) is dimensioned, and a rounding function, FNZ, is defined. The array is filled, or partially filled, with values to be graphed beginning at line 100. (The function in this demonstration is simply the log of successive numbers of a loop.)

Next, these values are normalized to 50, 0 to 49 rows. If the number of datum points is 60 or less, GOSUB 4600 is taken where the values are spread out to occupy twice as much room in RS, and linear interpolations are added between computed values. This step may be omitted or repeated if the original number of values in RS is 30 or less.

The graphing routine is shown in Listing 2. First, the grid is poked up, and X and Y headings provided. The numbers in the X axis heading will be either 0-2-4-6-8-10-12 (0 to 120), or 0-1-2-3-4-5-6 (0 to 60), depending on the number of original datum points in RS.

The logic and print statements for the graph begin at 4400. First, X, the horizontal co-ordinate index, is zeroed. The print loop is then started, and X is incremented.

Next, P is set to the first value of a pair of values in RS, and Q is set to the second. The statements IP=INT(P/2)*2 and IQ=INT(Q/2)*2 are used to determine whether P and Q are odd or even. An odd value will result in a graphic block in the upper half on an 8 x 8 matrix, an even value will cause the block to be placed in the lower half.

To determine whether P and Q are in the same 8 x 8 matrix, LEFT (LF) is set to P, and RIGHT(RI) is set to LF or -1. If Q proves equal to RI, blocks in diagonal corners of the matrix will be required.

If P is even, a block will be required in the lower left hand portion of a matrix; if odd it goes to the upper left hand portion. If Q is equal to P, a double width block will be needed; if Q is outside P's matrix, P's block must be poked up, and Q's matrix found. A block is then printed on the right side of Q's matrix—lower portion if Q is even, upper if odd.

To sort out all these relationships, a short tree is constructed. Three questions are asked, and CT (COUNT) is incremented by powers of 2 if the questions bring true answers.

1. Is P an even number?   CT=CT+1
2. Is Q equal to P?   CT=CT+2
3. Is Q in P's matrix?   CT=CT+4

When P and Q are in the same matrix, i.e. CT equals 2, 3, 4 or 5, the value in CT determines the graphic for both P and Q. When P and Q are in different matrices, i.e. CT=0 or 1, P's graphic is identified and poked in, and Q's location and graphic are determined separately. This is done beginning at statement 4480. CT=6 and CT=7 are trivial, and are circumvented.

The graph for 120 datum points using the function at statement 110 is shown in Figure 1. It was printed on an Okidata Microline 80.

Listing 3 includes the data doubler, and a fast screen clear indexed for OSI addresses.   □

Listing 1.
```
0 FOR I=1 TO 64:SC$=SC$+" ":NEXT
5 DIM RS(120)
10 DEF FNZ(R)=INT(R+.5)
90 GOSUB 5000:FL=0
95 INPUT"NUMBER OF DATUM POINTS (2 TO 120)"INB
100 FOR N=1 TO NB
110 RS(N)=LOG(N)
120 NEXT N
125 NX=INT((N-1)/2)*2
130 MX=10*-32:MN=10*32
140 FOR N=1 TO NX:IF RS(N))MX THEN MX=RS(N)
150 IF RS(N)(MN THEN MN=RS(N)
155 NEXT N
160 NR=49/(MX-MN):FOR N=1 TO NX
170 RS(N)=FNZ((RS(N)-MN)*NR)
180 NEXT N
190 IF NX(=60 THEN GOSUB 4600:NX=NX*2-1:FL=1
200 GOSUB 4300
300 PRINT:INPUT"RUN AGAIN"IYN$
310 IF YN$="Y" THEN 90
1000 END
```

Listing 2.
```
4300 REM        PREPARE GRAPH ****************************
4303 GOSUB 5000:D=53504:NP=0
4305 IF FL=1 THEN NQ=1:GOTO 4309
4307 NQ=2
4309 FOR I=1 TO 7:NR$=STR$(NP):N=LEN(NR$)
4310 FOR J=2 TO N:POKE D+J,ASC(MID$(NR$,J,1)):NEXT J
4315 NP=NP+NQ:D=D+10:NEXT I
4320 D=53569:FOR I=1 TO 71:FOR J=1 TO 61
4325 POKE D+J,45:NEXT J:D=D+256:NEXT I
4330 DH=53624
4335 FOR I=1 TO 7:D=DH+10:DH=D:FOR J=1 TO 23
4340 POKE D,33:D=D+64:NEXT J:NEXT I
4355 D=53310:FOR I=48 TO 0 STEP-8:D=D+256
4360 D$=STR$(I):DL=LEN(D$):FOR J=2 TO DL
4365 POKE D+J,ASC(MID$(D$,J,1)):NEXT J:NEXT I
4400 REM     BEGIN GRAPH LOOP ******************
4405 X=0
4410 FOR I=1 TO NX-1 STEP 2:X=X+1:CT=0
4415 P=RS(I):IP=INT(P/2)*2
4420 Q=RS(I+1):IQ=INT(Q/2)*2
4430 LF=P:IF P=IP THEN RI=LF+1:GOTO 4441
4435 RI=LF-1
4441 IF P=IP THEN CT=CT+1
4442 IF P=Q   THEN CT=CT+2:GOTO 4455
4444 IF Q=RI THEN CT=CT+4
4450 IF Q=0 THEN CT=8
4455 Y=INT((49-P)/2):MT=53570+X+Y+64
4460 ON CT GOTO 4471,4472,4473,4474,4475,4476,4477,4478
4471 POKE MT,167:GOTO 4480
4472 POKE MT,155:GOTO 4495
4473 POKE MT,154:GOTO 4495
4474 POKE MT,169:GOTO 4495
4475 POKE MT,170:GOTO 4495
4476 GOTO 4495
4477 GOTO 4495
4478 POKE MT,168
4480 Y=INT((49-Q)/2):MT=53570+X+Y+64
4485 IF Q=0 THEN POKE MT,165:GOTO 4495
4490 POKE MT,166
4495 NEXT I:RETURN
```

Listing 3.
```
4600 REM     ADJUST RS WHEN DATA FEWER THAN 61 *******
4610 FOR N=NX TO 2 STEP-1:N2=N*2-1:N1=N2-1:M=N-1
4620 RS(N2)=RS(N):DA=FNZ(ABS(RS(N)-RS(M))/2)
4630 IF DA=0 THEN RS(N1)=RS(N2):GOTO 4660
4640 IF RS(N))RS(M) THEN RS(N1)=RS(N2)-DA:GOTO 4660
4650 RS(N1)=RS(N2)+DA
4660 NEXT N:RETURN
5000 REM    FAST SCREEN CLEAR FOR OSI BASIC IN ROM
5010 LO=PEEK(129):HI=PEEK(130):POKE 129,255:POKE 130,215
5020 A$=SC$:FOR S=1 TO 32:A$=A$+" ":NEXT S
5030 POKE 129,LO:POKE 130,HI:RETURN
```

Pretending to Be Normal With The Low-Priced Spread

Marvin Weingast

Figure 1.

How to simulate the normal distribution with an easy improvement on random-ization.

In real-life situations, events occur randomly about a typical or mean value. For many simulations, these approximate a "normal" or Gaussian curve (Figure 1). For instance if you're a good shot most of your bullet holes on a target will be clustered around the bull's eye. If you aim your cannon at a target 13 miles away, some shots will be long, some will be short, but most will be near the 13 mile mark.

However, since the Basic function RND(X) generates a uniform distribution between 0 and 1, the expression

$$100 P = 13 + 2 * (RND(1) - .5)$$

will generate uniform values between 12 and 14. There will be just as many 12's as 13's, just as many 12.5's as 14's etc. No one value will appear significantly more often than another. So the above expression

Marvin Weingast, National Starch & Chem. Corp., 1735 W. Front St., Plainfield, NJ 07063.

will not show a cluster of values around 13 as in the real-life situation. Also, the above expression will not give the occasional very high or very low values outside the 12 to 14 range.

A very simple way to generate approximately normally distributed values is to use the modified Marsaglia-Bray equations.

A very simple way to generate approximately normally distributed values is to use the modified Marsaglia-Bray equations:

$$87.65\% \text{ of the time } X = 2 * (A + B + C - 1.5)$$

$$12.35\% \text{ of the time } X = 1.5 * (A + B - 1)$$

where A,B, & C are uniformly random numbers between 0 and 1 which could be generated by the RND(X) function.

The resulting values of X generated will form an approximately normal distribution around zero.

A sample program will illustrate this technique:

```
10    DIM Y(100)

20    FOR K = 0 TO 2500

30    A = RND(1): B = RND(1): C = RND(1)

40    IF A < .8765 THEN X = 2*(A+B+C-1.5)

50    IF A=> .8765 THEN X = 1.5*(A+B-1)

100   P = INT(4*(X+2))+5

110   IF P< 1 THEN GOTO 200

120   Y(P)=Y(P)+1

200   NEXT K

300   FOR L = 1 TO 25

310   PRINT TAB(INT(Y(L)/10));"*"

320   NEXT L

500   END
```

The modified Marsaglia-Bray equations can give your game or Monte Carlo simulation that added touch of realism. Happy shooting: And may your spreads look normal. □

References
G. Marsaglia & T.A. Bray, SIAM Review 6(3) pp. 260-264 (1964).

RUN

```
1 PRINT CHR$(12)
2 DIM Y(100)
10 FOR K=0 TO 10000
 100 A=RND(1):B=RND(1):C=RND(1)
 110 IF A<.8766 THEN X=2*(A+B+C-1.5)
 120 IF A=>.8766 THEN X=1.5*(A+B-1)
 130 P=INT(4*(X+2))+5
 140 IF P<1 THEN GOTO 200
150 Y(P)=Y(P)+1
200 NEXT K
300 FOR L=1 TO 25
 310 PRINT TAB(INT(Y(L)/20));"*"
320 NEXT L
500 END
READY
```

READY

Cass-a-Log

"*Drat!*" I muttered as I dug through another pile of papers. "Where the heck did that #$%&* paper go?"

For those of you that have just joined us, I am currently searching for a sheet of paper that tells me what programs are on what cassette. And as you may have noticed, I'm having a little trouble finding it.

But let me digress for a moment: Just over a year ago I bought my new Apple II computer. And with it I purchased a 5 1/4" floppy disk drive, vowing never again to stoop so low as to fight with a cassette recorder.

You see, prior to the Apple, I owned a TRS-80, which everyone knows is a barrel of fun to load tapes into. But now I had the world in the palm of my hand. With a disk, I could load programs in seconds, run a catalog to see what was on the disk, and have tremendous amounts of storage at my beck and call. Well almost.

It seems that even though the disk was the answer to my speed and reliability problems, it still was not completely moron-proof! Within the first six months I had knocked it out of wack.

It was simple: it seems that I forgot to shut off the power switch while unplugging a card inside the machine. Well, it didn't require a second try to wipe out my disk controller card, so while I waited for the card to be fixed, I was without my machine, because I had not made tape backups to cover myself.

Scott King, 7905 59th Ave. N., New Hope, MN 55428.

Scott King

Now armed with the knowledge of what could happen, I decided I'd better get some kind of backup system going. So I went out and invested $29 in a cheap tape recorder and a couple of bucks in cassettes, and started dumping programs. Well this was nice while it lasted, but I soon found that there were still problems to be had.

Now that I had over 200 programs, it was no longer cost effective to put only one program on each side of the tape. After all a C-10 cassette costs more than a C-60, but it takes so long to move the tape ahead to the other side. And if you put more than one or two programs per side of the tape, you have problems locating their starting points.

This is one area in which the Apple fell behind the TRS-80: you can't search for a program via a file name on cassette. So the obvious thing to do was to record the tape counter number for the beginning of each program. Well this too has its drawbacks. If you try to write all of the data, (counter number, program name, and format) on the cassette label....well let's just say that you have to be able to read and write *very* small.

Again the obvious thing to do was to record the data on a separate sheet so that the information could be read without going cross-eyed.

This brings us back to the present. I can't for the life of me find that darned paper...and then it came to me: If I could

have a catalog on a disk, why not on a cassette? Because you don't have the same access as a disk. But, you could include a basic program at the very beginning of the tape that would list all of the data that the tape holds.

The Program

The format to be used will vary with each type of computer, but the logic behind it is the same. If you want to find a given program, you pop the cassette in the recorder, and type LOAD or CLOAD in a TRS-80. After the index loads, you RUN it, and you get the display in Figure 1.

You notice that once this is RUN you can just fast forward to the counter number shown, and load the program using Integer Basic, Applesoft, or machine language as indicated. Also note that the index includes the location of the blank tape. This is very useful when you want to save a new program. All you do is look at the index, fast forward to the AVAILABLE location and save your program.

Figure 1.

```
    DATA TAPE NUMBER 021
    >= INTEGER BASIC
    1= APPLESOFT
    *= MACHINE CODE
COUNTER     PROGRAM
000         >INDEX
020         >STARTREK
050         1APPLE-VISION
070         >GAMES
105         *SPACE FLIGHT   000.234SR
130         ----- AVAILABLE ------

] END
```

180

Be sure to remember to record the new AVAILABLE location so that you can have it for next time. Now that you've added a program to the tape you have to add it to the index. So you rewind to the start of the tape, load the index program, make the changes, and save it back to the tape.

I always reserve the beginning from 000 to 020 for the index. That allows plenty of room to save the index twice. Save the program at the beginning of your tapes changing line 30 to indicate the tape number you are working with.

That's about it: since I've started using this, I have stopped losing my lists—now I misplace entire cassettes. □

Program for TRS-80.

```
10 REM **** CASS-A-LOG ****
20 CLS
30 PRINT " DATA TAPE NUMBER 000"
40 PRINT " )-LEVEL II BASIC"
50 PRINT " *-MACHINE CODE (SYSTEM)"
60 PRINT
70 RESTORE
80 PRINT "COUNTER","PROGRAM     FILENAME"
90 READ A$:READ B$
100 PRINT A$,B$
110 IF A$="END" THEN END
120 GOTO 90
130 DATA 000,) INDEX       )CLOAD"I"
140 DATA 020,AVAILABLE
150 DATA END,END
```

Program for those Tiny Basic's that won't support data statements.

```
10 REM **** CASS-A-LOG ****
20 (WHAT EVER INSTRUCTION CLEARS THE SCREEN)
30 PRINT "DATA TAPE NUMBER 000"
40 PRINT (USE LINES 40-60 TO DEFINE LOADING FORMATS)
50 PRINT
60 PRINT
70 PRINT "COUNTER","PROGRAM"
80 PRINT "000","INDEX
90 PRINT "020","AVAILABLE "
100 ADD MORE LINES AS NEEDED----
```

Program for Applesoft.

```
10 REM **** CASS-A-LOG ****
20 HOME
30 PRINT "DATA TAPE NUMBER 000"
40 PRINT " )= INTEGER BASIC"
50 PRINT " )= APPLESOFT"
60 PRINT " *= MACHINE CODE"
70 RESTORE
80 PRINT "COUNTER","PROGRAM"
90 READ A$:READ B$
100 PRINT A$,B$
110 IF A$="END" THEN END
120 GOTO 90
130 DATA 000,)INDEX
140 DATA 020,AVAILABLE
150 DATA END,END
```

INTRODUCTION TO COMPUTERS IN EDUCATION FOR ELEMENTARY AND MIDDLE SCHOOL TEACHERS

David Moursund
Dept. of Computer & Information Science
University of Oregon
Eugene, Oregon 97403

International Council for Computers in Education
Dept. of Computer & Information Science
University of Oregon
Eugene, Oregon 97403

Copyright ICCE 1981   Price $7.00 U.S.

THE WORD IS OUT!

Evaluator's Guide
For Microcomputer-Based Instructional Packages

Developed by MicroSIFT
a project of
The Computer Technology Program
Northwest Regional Educational Laboratory

Printed by ICCE
Dept. of Computer & Information Science
University of Oregon
Eugene, OR 97403

More Efficient Tape Storage

Kevin A. Frye

If you're frustrated with the slow data recording rate of the tape system on your Level II TRS-80 and are thinking of selling your house to buy a disk system, don't give up hope yet. A simple way to add some pep to your tape system is to pack more data onto each tape.

When you write a string onto the tape in the conventional way (i.e. PRINT #-1, A$), 255 bytes of data are written onto the tape for A$ no matter how long the usable part of A$ really is. In the most extreme case, if A$ as defined in your program is 1 byte long, 254 null characters are appended to A$ and the whole 255 byte string goes on the tape. At 500 baud (62 cps) each of these 255 byte strings takes a little over four seconds to write and consumes two feet of tape.

By taking advantage of the powerful string manipulation functions of Level II Basic and some simple string concatenation (appending) you can add a great deal of efficiency to your tape storage applications. The essence of this storage technique is to pack as much data as possible into a 255 byte string buffer in your program and then write the buffer onto the tape.

Kevin A. Frye, 5 Ledgewood Terrace, Apt. #502, Saco, ME 04072.

To retrieve this packed data from the tape you simply execute an INPUT #-1 command for each of the packed strings and unpack the strings as you read them.

For an illustration of how this packing system works let's assume that our application program is working with a string array T$(30) that contains phone numbers in XXX-XXXX format (8 bytes each). The following subroutines show how to pack and unpack this data.

It is not necessary to have a perfect 255 byte fit as long as you pack as much data as possible into a 255 byte string before doing output to the tape. You may also have as many packed strings on a tape as your application calls for or will fit on a tape. The time and tape saving depends on the size of the component pieces of data you are packing into the buffer string. The saving factor can be expressed as:

INT (255/Data element size)

If packing was not used in the above example and each array element was written in separate PRINT #-1 commands, the output process would take over 120 seconds and consume 60 feet of tape. By packing the array into one 255 byte buffer before doing tape output, it takes a little over four seconds and consumes only two feet of tape. □

```
1000 REM    PACK AND WRITE THE ARRAY
1010 D$=""                          ' SET THE BUFFER STRING TO NULL
1020 FOR X=1 TO 30
1030 P$ = LEFT$(T$(X),8)            ' PUT 8 LEFTMOST BYTES OF PHONE # INTO P$
1040 P$ = P$ + STRING$(8-LEN(P$),32) ' APPEND SPACES TO P$ IF P$<8 LONG
1050 D$ = D$ + P$                   ' APPEND PHONE # TO BUFFER STRING
1060 NEXT X
1070 PRINT #-1,D$                   OUTPUT THE PACKED STRING
1080 RETURN

2000 REM    READ AND UNPACK THE ARRAY
2010 INPUT #-1,D$                   ' READ PACKED STRING FROM THE TAPE
2020 FOR X = 1 TO 30
2030 S = (X-1) * 8 + 1              ' COMPUTE START POSITION OF NEXT PHONE #
2040 P$ = MID$(D$,S,8)              ' PUT NEXT PHONE # INTO P$
2050 T$(X) = P$                     ' PUT PHONE # INTO ARRAY
2060 NEXT X
2070 RETURN
```

Tired Of Your
GENERAL LEDGER?

VERSA-LEDGER

★ THE ULTIMATE PERSONAL CHECK REGISTER

★ A PROFESSIONAL ACCOUNTING SYSTEM

★ A PERSONAL FINANCIAL MANAGER

★ A SMALL BUSINESS ACCOUNTING SYSTEM

★ A COMPLETE GENERAL LEDGER

FOR YOUR TRS-80 MODEL I, II, III or COLOR COMPUTER ● APPLE II ● I.B.M. ● XEROX ● ATARI
ALL MICROSOFT BASIC COMPUTERS

HOW IT WORKS . . . .

VERSALEDGER is a complete accounting system that grows as you or your business grows. To start, your VERSALEDGER acts as a simple method of keeping track of your checkbook. Just enter your check number, date and to whom the check is made out to. As you or your business grows, you may add more details to your transactions . . . . account number, detailed account explanations, etc.

- VERSALEDGER can give you an instant cash balance at anytime. **(IF YOU WANT IT TO)**
- VERSALEDGER can be used as a small personal checkbook register. **(IF YOU WANT IT TO)**
- VERSALEDGER can be used to run your million dollar corporation. **(IF YOU WANT IT TO)**
- VERSALEDGER prints checks. **(IF YOU WANT IT TO)**
- VERSALEDGER stores all check information forever. **(IF YOU WANT IT TO)**
- VERSALEDGER can handle more than one checkbook. **(IF YOU WANT IT TO)**
- VERSALEDGER can be used to replace a general ledger. **(IF YOU WANT IT TO)**

VERSALEDGER

YOUR FIRM NAME HERE

CHECK №. 1001

INTRODUCTORY PRICE
$99.95

- VERSALEDGER HAS AN ALMOST UNLIMITED CAPACITY . . . .
  (300 checks per month on single density 5¼" disk drives such as the TRS-80 Model-I)
  (500 checks per month on the Apple II)
  (2400 checks per month on the TRS-80 Model II)
  (6000 checks per month on the TRS-80 Model II)
  (3000 checks per month on single density 8" CP/M)
- VERSALEDGER will soon have an add-on payroll package. **(IF YOU NEED IT)**

— CAN BE USED WITH 1 or MORE DISK DRIVES —

VERSALEDGER HAS BEEN CREATED
WITH THE FIRST TIME COMPUTER USER IN MIND

Data Without Duplicates

Glenn Fisher

This routine for the PET allows you to select data statements randomly without repeats, and permits variable amounts of data. There are two parts: the first counts the data, while the second picks an unused datum.

The routine was written for a question and answer program using DATA statements, designed so users could type any questions and answers they wished into the DATA statements and have them randomly presented without repeats.

The usual choice for random selection of information is between subscripted variables and DATA statements. Novice users find it easier to change information in DATA statements. Therefore, in writing a program where the user supplies the information to be presented randomly, it is preferable to use DATA statements. The usual method of randomly presenting information chosen from DATA statements is to read a random number of data. This often results in reading one datum several times, while others are not chosen at all.

This routine reads each datum just once, in a random order. A string variable, NU$, consists of the numbers of all unused DATA statements. NU$ stores all numbers as two digits, up to the number of DATA statements (maximum, 99).

For example, if there are 11 data, NU$ will be "0102030405060708091011." A random number is used to choose one of these digit pairs from the string. Let's say the random number is 9: the pair of digits starting with the ninth digit of NU$ is 05. A FOR/NEXT loop reads to the fifth datum. The chosen digits (05) are now removed from the string so that the datum will not

Glenn Fisher, 1517 Holly St., Berkeley, CA 94703.

```
Random Data
100 RESTORE: REM COUNT DATA
110 NU$=""
120 READ Q$,A$
130 IF Q$="END" THEN Q1%=Q%: GOTO 200
140 Q%=Q%+1
150 Q$=STR$(Q%)
160 IF Q% < 10 THEN Q$="0"+RIGHT$(Q$,1)
170 NU$=NU$+RIGHT$(Q$,2)
180 GOTO 110
200 your program

500 RESTORE: REM READ RANDOM DATA
510 RN%=2*INT(RND(1)*Q1%)+1
520 R%=VAL(MID$(NU$,RN%,2))
530 FOR I = 1 TO R%: READ Q$,A$:NEXT
540 IF RN%=1 THEN E$="":GOTO 560
550 E$=LEFT$(NU$,RN%-1)
560 NU$=E$+MID$(NU$,RN%+2)
570 Q1%+Q1%-1: IF Q1%=0 THEN END
580 PRINT Q$, A$, NU$: REM DISPLAY ONLY
590 GOTO 500
```

be read again. NU$ becomes "0102030406 0708091011."

The random number is limited so it can't pick past the last pair of NU$. Let's say the random number is again 9 on the next pass. This time, the ninth and tenth digits are 06, so the sixth datum will be read. NU$ will then be "010203040708091011."

Program Explanation

The first part of this routine—lines 100-170—goes at the beginning of your program, and counts the number of data. Lines 100 and 110 initialize the string NU$ and data.

Line 120 reads pairs of questions and answers. (If you're using a single datum, delete A$ here and in line 530.)

Line 130 checks for end of data and exits the routine after setting Q1%, a counter, equal to the number of data, Q%. The last DATA statement must be END, END.

Lines 140-170 take care of the counters. Q%, counting the number of data, is incremented in line 140. This number is turned into a string in line 150, and line 160 adds a leading zero if the number is less than 10, so it will be stored as two digits (i.e., 2 becomes 02).

Duplicates, continued...

Line 170 adds this number to the list in string NU$. RIGHT$ is necessary in lines 160 and 170 to remove the blank which prefaces numeric variables on the PET.

The second part, lines 500-570, can go anywhere before you need to read and use the data. Lines 580 and 590 are just for a demonstration run, and will be replaced by your program.

Line 510 generates an odd number from one to twice the number of data left (Q1%). This is necessary because each datum is represented in NU$ by two digits, starting an odd number of digits from the beginning of NU$. (The third datum takes the fifth and sixth digits in NU$).

Line 520 picks the two digits from NU$, starting with the RN%th digit. VAL is necessary to give the numeric value of these string digits. If NU$="020305," and RN%=3, the third and fourth digits —03— would be picked, and R% would be 3.

Line 530 reads down to the chosen datum and puts it in Q$ and A$.

Lines 540-560 remove the chosen digits from the string so they cannot be picked again. This is done by combining a left part and an end part. Line 540 sets the left part to a null string if the first two digits were picked.

Line 550 saves the digits to the left of those used.

Line 560 combines the left part with the digits after those used, so NU$ no longer contains the used pair of digits.

Line 570 decrements the counter of the number of unused data, then checks to see if they've all been used.

Since all numbers used in this routine are integers, all numeric variables have had % added to make them integer variables, saving some time and memory. The % can be deleted with no effect on the routine. □

"No matter what we've tried, all our PET would do is multiply, so we took it to the vet and got it fixed!."

LAST ONE LOSES

A Game of Strategy

William E. Bailey

Many of you have probably played the ancient intellectual game of "Nim" and found it challenging. "Last One Loses" is a variant of "Nim" for the TRS-80 which provides a new and greater challenge.

The playing board in "Last One Loses" has three rows of circles: three in the top row, five in the middle row, and seven in the bottom row. This creates a board which looks like this:

```
        0 0 0
      0 0 0 0 0
    0 0 0 0 0 0 0
```

The object is to cross off the circles in such a way that your opponent is forced to delete the last one. The players alternate crossing off as many circles as they wish so long as the circles are adjacent and in the same row. Each circle may be crossed off only once. For example, assume that player number one decides to cross off all three circles in the top row. The status of the game may be indicated as shown below:

```
        X X X
      0 0 0 0 0
    0 0 0 0 0 0 0
```

Notice that the circles that have been marked off are adjacent and in the same row.

If player number two then crosses off the th ee middle circles in the bottom row, we have:

```
        X X X
      0 0 0 0 0
    0 0 X X X 0 0
```

Note that crossing off circles in the middle of rows is legal.

William E. Bailey, 4601 N. Park Ave., Apt. 1811, Chevy Chase, MD 20015.

Then suppose player number one crosses off all the circles in the middle row:

```
      0 0 X X X 0 0
      X X X X X
        X X X
```

If your opponent has a winning pattern, all you can do is move randomly and hope that he makes a mistake.

And player number two counters by crossing off a circle in the bottom row:

```
      0 0 X X X 0 X
      X X X X X
        X X X
```

Now player number one can win by crossing off the two circles on the right as follows:

```
      X X X X X 0 X
      X X X X X
        X X X
```

Figure 1.

Since it is the second player's turn and there is only one circle remaining, he is forced to delete the last circle and player number one wins the game.

The strategy in "Last One Loses" is fairly simply but it takes a lot of trial and error to understand it thoroughly. Being a pro at the game is merely a matter of knowing which patterns of circles are winning patterns and which are not.

To understand patterns better, let us take the playing board and label each row as shown in Figure 1.

The pattern is simply the number of circles that are separated from the others by rows or by circles that have been marked off already. For example:

Diagram #1, the initial board has a pattern of |3,5,7|, since row A has three circles, row B has five, and row C has seven. An element of a pattern is a single set of circles, so the pattern of diagram #1 has three elements. The first with three circles, the second with five, and the last with seven.

Diagram #2 has a pattern of |3,7| since row A has three circles and row C has seven circles. Row B has no circles and is considered null. Notice that not all patterns have the same number of elements since the pattern of diagram #2 only has two elements whereas the pattern of diagram #1 has three elements.

Diagram #3 also has a pattern of |3,7| since row A is null, row B has three

	#1	#2	#3	#4
A.	0 0 0	0 0 0	X X X	0 X 0
B.	0 0 0 0 0	X X X X X	0 0 0 X X	0 0 X X 0
C.	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 X X 0 0

Last One, continued...

circles and row C has seven. Note that even though diagrams #2 and #3 have different circles crossed off, they can be considered the same game because their patterns are identical.

Diagram #4 has two elements in each row, since they are all separated by circles that have been crossed off in the middle of the row. Row A has two elements each being a single circle, so thus far the pattern is |(1,1)|. Row B has one element with one circle and another with two. This adds to the pattern |(1,1), (2,1)|. (The parentheses indicate elements in different rows, while brackets surround the pattern.) And finally, row C has one element with three circles and another with two, so the pattern of diagram #4 is |(1,1), (3,2)|. Now that we have the patterns put together with all of the elements, we can remove the parentheses to make this |1,1,2,1,3,2| and for convenience, the numbers can be put in least to greatest order, so the pattern for diagram #4 in its final form is |1,1,1,2,2,3|.

Now let us go back to the example game. On player number one's second move he crossed off the entire middle row which left this:

```
A     X X X
B     X X X X X
C     0 0 X X X 0 0
      1 2       3 4
```

I have labeled the circles that we will be using with the numbers one through

four. This has a pattern of |2,2|, circles 1 and 2 make up the first element and circles 3 and 4 make up the second element. This is a winning pattern. By a winning pattern I mean that if a player can delete circles so as to make a pattern of |2,2| in any form then that player will be able to win the game. Player number one deleted row B and made a pattern of |2,2| and therefore will be able to win the game. This may not be immediately obvious so let us examine every possibility.

As happened in the example game, if player number two crosses out circle 1 then player number one can countermove by marking off circles 3 and 4 and only 2 will remain. Since it is the turn of player number two he loses the game and player number one wins.

If, instead, player number two marks off circles 1 and 2 then player number one can mark off 3 and only 4 remains, again player number one wins. And vice versa for circles 3 and 4. If player number two decides to cross off circle 4, then player number one can cross off 1 and 2, and if player number two decides to cross off circles 3 and 4, then player number one can cross off circle 1, leaving 2.

Thus, we have examined every possible move for player number two and discovered that no matter where he moves, player number one can always win. This means that the pattern of |2,2| in any form is a winning pattern. I use the "in any form" because there are many ways to fit a pattern of |2,2| on the playing board. Figure 2 shows a few of them:

```
20 DEFINT A-Z
20 DIM BD(3,7),DA(3,7),PA(9),BA(21),SA(25)
30 FOR I=1 TO 21
40    READ BA(I)
50 NEXT I
60 DATA 1,1,1,1,2,3,1,4,5,2,4,6,2,5,7,3,4,7,3,5,6
70 FOR J=1 TO 25
80    READ SA(J)
90 NEXT J
100 DATA 1,1,1,1,1,1,1,1,2,3,1,1,1,4,5,1,2,2,2,3,1,2,3,3,3
110 CLS
120 FOR I=1 TO 3
130    FOR J=1 TO 7
140       BD(I,J)=0
150    NEXT J
160 NEXT I
170 BD(1,1)=1:BD(1,2)=1:BD(1,6)=1:BD(1,7)=1:BD(2,1)=1:BD(2,7)=1
180 REM   "LAST ONE LOSES"
190 PRINT @ 12,CHR$(23)"WELCOME TO THE GAME"
200 PRINT @ 79,CHR$(23)"OF LAST-ONE LOSES"
210 FOR I=1 TO 1000:NEXT I
220 CLS
230 PRINT"DO YOU NEED TO SEE THE INSTRUCTIONS"
240 INPUT ST$
250 IF LEFT$(ST$,1)="Y" THEN GOSUB 1950
260 PRINT @ 0,"WHAT LEVEL OF DIFFICULTY DO YOU WANT(1-2-3-4)"
270 INPUT LE
280 LE=INT(LE)
290 IF LE>4 OR LE<1 THEN PRINT"INVALID":GOTO 260
300 PRINT @ 0,"DO YOU WANT TO GO FIRST"
310 INPUT OT$
320 IF LEFT$(OT$,1)="N" THEN GOTO 620
330 GOSUB 1740 :REM DISPLAY BOARD
340 REM PLAYER'S MOVE
350 PRINT @ 256,"DO YOU WANT TO KNOCK OUT MORE THAN ONE CIRCLE"
360 INPUT OZ$
370 IF LEFT$(OZ$,1)="Y" THEN GOTO 410
380 PRINT @ 384,"WHICH CIRCLE":INPUT MZ,NZ
390 IF BD(MZ,NZ)< 0 THEN PRINT"ILLEGAL":GOTO 380
```

Charter Subscriber Certificate

Send my

Charter Issue of TECHNOLOGY

FREE

And enter my 1-year Charter Subscription (6 bi-monthly issues) for only $18 . . . a 16 saving off the regular subscription rate. I understand that if I'm not satisfied with my first issue I need only write "cancel" on my bill, I may keep my first issue of TECHNOLOGY and I'll owe nothing. Otherwise, I'll return your invoice with payment and receive 5 more issues.

Name _____

(please print)

Address _____

City _____

State _____ Zip _____

5N7

Figure 2.

```
    0 0 X

  X 0 0 X X

  X X X X X X


X X X X X X X   X X X X 0 0 X

  0 0 X 0 0       X X X X X

    X X X           0 0 X
```

As you can see, many other combinations are possible and all of them are winning patterns. The pattern |2,2| is not the only winning pattern. There are many others, all of which are included in the program.

When we were examining the pattern |2,2| to make sure it was a winning pattern, we discovered that player number one was able to win the game within two moves from the establishment of the pattern. Now let's start a new game in which player number one has just formed the winning pattern of |2,5,7|.

It is obvious that player number 1 will not be able to win the game within the next two moves because the |2,5,7| pattern is so much bigger than the |2,2| pattern. In fact, it might take as many as 12 moves before the game is ended.

So how is player number one supposed to know what to cross off on all of his next moves? The answer is quite simple. Since |2,5,7| is a winning pattern, no matter where player number two moves, there will always be a move for player number one that will establish a new and smaller winning pattern. This continues on every move until player number one wins the game. For example, the pattern so far is |2,5,7| which will only fit the board in the way shown below:

```
A        X 0 0

B        0 0 0 0 0

C        0 0 0 0 0 0 0
```

Note that since |2,5,7| is a winning pattern, this would be a good first move.

It is player number two's move and let's say that he crosses out three circles in row B. The result is:

```
         0 0 0 0 0 0 0

         X X X 0 0

         0 0 X
```

Note that this is one of many possible moves for player number two.

Now there are several places that player number one can move to reestab-

```
400 EQ(MZ,NZ)=2:GOTO 510
410 PRINT @ 384,"FROM":INPUT MZ,NZ
420 PRINT @ 512,"TO":INPUT ZI,ZJ
430 IF MZ<.ZI THEN PRINT"ILLEGAL": GOTO 410
440 IF NZ<ZJ THEN E=NZ:NZ=ZJ:ZJ=E
450 FOR I=NZ TO ZJ
460     IF EQ(MZ,I)<>0 THEN PRINT"ILLEGAL":GOTO 410
470 NEXT I
480 FOR J=NZ TO ZJ
490     EQ(MZ,J)=2
500 NEXT J
510 GOSUB 1740 :REM DISPLAY BOARD
520 FOR OY=1 TO 7:PRINT:NEXT OY
530 GOSUB 1030 : REM FIND PATTERN
540 IF NOT(T=1 AND PA(1)=1) THEN GOTO 670
550 PRINT @ 256,"YOU WON....NICE JOE"
560 FOR D=1 TO 200:NEXT O
570 PRINT"BUT I'D LIKE TO SEE YOU DO IT AGAIN"
580 FOR D=1 TO 200:NEXT O
590 INPUT"CARE FOR ANOTHER GAME";OE$
600 IF LEFT$(OE$,1)="Y" THEN CLS:GOTO 110
610 END
620 REM COMPUTER'S MOVE
630 GOSUB 1740 :REM DISPLAY BOARD
640 IF LE<-4 THEN GOTO 740
650 PRINT @ 256,"THINKING"
660 GOTO 800
670 REM CHECK IF OPPONENT HAS WINNING PATTERN
680 IF T=0 THEN GOTO 1310
690 PRINT @ 256,"THINKING"
700 DW=LE:LE=4
710 GOSUB 1160 :REM IS PATTERN WINNING?
720 LE=DW
730 IF WN=0 THEN GOTO 800
740 REM RANDOMLY MOVE
750 I=RND(3):J=RND(7)
760 IF EQ(I,J)<>0 THEN 750
770 EQ(I,J)=2
780 GOSUB 1740 :REM DISPLAY BOARD
790 GOTO 340
800 REM PICK CIRCLES TO CROSS OFF
810 FOR I=1 TO 7
820    FOR J=1 TO 7
830       X=-1
840       IF EQ(I,J)<>0 THEN GOTO 1000
850       X=X+1
860       PRINT @ 265,CHR$(143)
870       IF J+X=8 THEN GOTO 1000
880       IF EQ(I,J+X)<>0 THEN GOTO 1000
890       FOR IJ=J TO J+X
900          EQ(I,IJ)=2
910       NEXT IJ
920       GOSUB 1030 :REM FIND PATTERN
930       PRINT @ 265,CHR$(32)
940       GOSUB 1160 :REM IS IT WINNING?
950       IF WN=1 THEN GOSUB 1740 :GOTO 340
960       FOR IJ=J TO J+X
970          EQ(I,IJ)=0
980       NEXT IJ
990       GOTO 850
1000   NEXT I
1010 NEXT J
1020 GOTO 740
1030 REM SUBROUTINE FIND PATTERN
1040 T=0:JK=0
1050 FOR M=1 TO 3
1060    FOR N=1 TO 7
1070       IF EQ(M,N)=0 THEN JK=JK+1 ELSE GOTO 1130
1080       IF N=7 THEN GOTO 1100
1090       IF EQ(M,N+1)=0 THEN GOTO 1130
1100       T=T+1
1110       PA(T)=JK
1120       JK=0
1130    NEXT N
1140 NEXT M
1150 RETURN
1160 REM SUBROUTINE IS IT WINNING?
1170 REM SORT PATTERN
1180 FOR F=1 TO T-1
1190    C=T-F+1
1200    FOR G=2 TO C
1210       IF PA(G)>=PA(G-1) THEN GOTO 1250
1220       KL=PA(G)
1230       PA(G)=PA(G-1)
1240       PA(G-1)=KL
1250    NEXT G
1260 NEXT F
```

lish a winning pattern. One is to cross off the entire bottom row:

```
X X X X X X X
   X X X 0 0
      0 0 X
```

Note that the [2,2] winning pattern reappears.

And again, no matter where player number two moves, player number one can countermove to form another winning pattern. This is the case with every winning pattern. Each can be made into a smaller winning pattern no matter where the opponent moves. Because of this, if your opponent has a winning pattern, all you can do is move randomly and hope that he makes a mistake.

There are four levels of difficulty in the program. Level 1 is the easiest and Level 4 is the most difficult to defeat. If the player plays Level 4 and allows the computer to move first the player chooses who makes the first move) then the computer will always win. Indeed, if you watch the computer's moves it will make a winning pattern on each turn. But Levels 3, 2, and 1 are progressively less smart and the computer does not always make optimal moves.

The design of the program can be broken down into two major parts. The player's move can be found in the beginning of the program. It inputs the move, checks it for validity, and then displays the result on the board. The computer's move is much more complex. It has five subparts:

1. Check if opponent has a winning pattern.

2. Pick circles to cross off.

3. Find pattern.

4. Sort pattern.

5. Is pattern winning?

Note: These are all denoted in the program by REM statements.

The computer checks to see if the player has a winning pattern first for time conservation because, as I stated earlier, if the player has a winning pattern, all the computer can do is to hope that the player blows it. In the meantime, the computer moves randomly.

If the player does not have a winning pattern then the computer crosses off circles and checks to see if the pattern it has created is a winning one. It does this by figuring out the pattern through the "find pattern" sub-routine. Then it puts the elements of the pattern into least to greatest order through the "sort pattern" sub-routine and checks the pattern through the "Is pattern winning" sub-routine. When the computer finds a winning pattern it goes back to the player's move and the process continues until the game is over. □

```
1270 REM CHECK FOR WINNING PATTERN
1280 ON T GOTO 1290 ,1360 ,1430 ,1500 ,1580 ,1670 ,1690 ,1710 ,1730
1290 IF PA(1)<>1 THEN WN=0:RETURN
1300 GOSUB 1740 :REM DISPLAY BOARD
1310 PRINT @ 256,"I BEAT YOU THIS TIME!'"
1320 FOR FG=1 TO 200:NEXT FG
1330 INPUT"CARE FOR ANOTHER GAME";GH$
1340 IF LEFT$(GH$,1)="Y" THEN CLS:GOTO 110
1350 END
1360 IF PA(2)=1 THEN WN=0:RETURN
1370 IF PA(1)=PA(2) THEN WN=1:RETURN
1380 WN=0:RETURN
1390 LM=0:MN=0
1400 FOR K=LM+1 TO T+LM
1410    MN=MN+1
1420    IF PA(MN)=BA(K) THEN GOTO 1480
1430    IF K=19 OR K=20 OR K=21 THEN WN=0:RETURN
1440    IF LE=4 THEN GOTO 1460
1450    IF K=LE*3-2 OR K=LE*3-1 OR K=LE*3 THEN WN=0:RETURN
1460    LM=LM+1:MN=0
1470    GOTO 1400
1480 NEXT K
1490 WN=1:RETURN
1500 IF PA(4)=1 THEN WN=0:RETURN
1510 ON LE GOTO 1520 ,1530 ,1540 ,1560
1520 WN=0:RETURN
1530 IF PA(1)=1 AND PA(2)=1 AND PA(3)=PA(4) AND  PA(4)=2 PA
EM WN=1:RETURN ELSE WN=0:RETURN
1540 IF PA(1)=1 AND PA(2)=1 AND PA(3)=PA(4) THEN WN=1:RETURN
1550 WN=0:RETURN
1560 IF PA(1)=PA(2) AND PA(3)=PA(4) THEN WN=1:RETURN
1570 WN=0:RETURN
1580 LM=0:MN=0
1590 FOR K=LM+1 TO T+LM
1600    MN=MN+1
1610    IF PA(MN)=SA(K) THEN GOTO 1650
1620    IF K>=21 AND K<=25 THEN WN=0:RETURN
1630    LM=LM+1:MN=0
1640    GOTO 1590
1650 NEXT K
1660 WN=1:RETURN
1670 IF PA(6)=1 THEN WN=0:RETURN
1680 IF PA(1)=PA(2) AND PA(3)=PA(4) AND PA(5)=PA(6) THEN WN=1:RETURN ELSE
WN=0:RETURN
1690 IF PA(7)=1 THEN WN=0:RETURN
1700 WN=0:RETURN
1710 IF PA(7)=2 THEN WN=1:RETURN
1720 WN=0:RETURN
1730 WN=1:RETURN
1740 REM SUBROUTINE DISPLAY BOARD
1750 FOR EZ=1 TO 3
1760    FOR EX=1 TO 7
1770       IF BO(EZ,EX)=0 THEN OA$(EZ,EX)="O"
1780       IF BO(EZ,EX)=1 THEN OA$(EZ,EX)=" "
1790       IF BO(EZ,EX)=2 THEN OA$(EZ,EX)=CHR$(143)
1800    NEXT EX
1810 NEXT EZ
1820 FOR FT=1 TO 7
1830    PRINT @ (FT-1)*2," ";OA$(1,FT);
1840 NEXT FT
1850 PRINT"                        1,3  1,4  1,5"
1860 FOR FT=1 TO 7
1870    PRINT @ (FT-1)*2+64," ";OA$(2,FT);
1880 NEXT FT
1890 PRINT"                   2,2  2,3  2,4  2,5  2,6"
1900 FOR FT=1 TO 7
1910    PRINT @ (FT-1)*2+128," ";OA$(3,FT);
1920 NEXT FT
1930 RETURN
1940 CLS
1950 PRINT"    THE OBJECT OF THIS GAME IS TO CROSS OFF CIRCLES IN SUCH A"
1960 PRINT"WAY SO THAT YOUR OPPONENT IS FORCED TO DELETE THE LAST ONE."
1970 PRINT PERTINENT RULES ARE:"
1980 PRINT
2000 PRINT"    1) YOU CAN CROSS OFF AS MANY CIRCLES AS YOU WISH;"
2010 PRINT"    PROVIDED THAT-"
2020 PRINT"       A:THEY ARE ADJACENT AND IN THE SAME ROW."
2030 PRINT"       B:AND THE CIRCLES THAT YOU ARE CROSSING OFF HAVE NOT"
2040 PRINT"         BEEN MARKED OFF ALREADY."
2050 PRINT
2060 PRINT"    THE GAME IS ORGANIZED INTO FOUR DIFFERENT LEVELS OF PLAY,"
2070 PRINT"LABELED 1 TO 4.GAME 1 IS THE EASIEST AND GAME 4 IS THE HARDEST"
2080 PRINT"THE COMPUTER AND THE PLAYER ALTERNATE IN CROSSING OF CIRCLES,"
2090 PRINT"AND REMEMBER-WHOEVER CROSSES OFF THE LAST-ONE LOSES!!"
2100 PRINT"    'HIT 'C' WHEN YOU ARE READY TO CONTINUE.'"
2110 IF INKEY$="C" THEN CLS:RETURN
2120 GOTO 2110
```

Anxiously, I opened the carton to my new Apple II. Quickly, I connected all the cables and leads. Then, with nervous anticipation, I turned on the power switch. I pressed the RESET key, heard the speaker 'beep,' and saw the asterisk prompt appear on the screen of my monitor. Everything seemed to be working. Now back to the Reference Manual. How do I make it do something? Wow! This manual is confusing! How can anyone understand all this information. These 50 pages of reference material are a lot for a beginner to learn.

That's how it all started in September 1977. My Apple II, serial number 333, is still the same. Disk drives and 32K more RAM were added. All the expansion slots are full, and connection to peripherals, like my Centronics 737 printer, is now a simple task. Connecting peripherals wasn't always so easy though. There were no peripheral boards available at first. One of my first computer articles described hooking up a Telpar 40-column thermal printer. I had to tell everyone how to make it work from the game paddle connector.

The program used to drive the printer was similar to the one included in the Dec. '81 column. Except back then, the hook to DOS (disk operating system) at $03EA wasn't needed. The only interface needed was a pair of wires from the game paddle connector. Later on, when the expanded Apple II Reference Manual (the 151-page red book) came along, a more sophisticated program to drive a teletype was included. This program would also drive serial printers (with TTL compatible inputs).

From Out of The Past

Lots of things happened over the following months and years. We learned ways

Chuck Carpenter, 2228 Montclair Pl., Carrollton. TX 75007.

Chuck Carpenter

to get programs on and off tape. (Our first exposure to the use of monitor routines.) And things like PR#1 and IN#1 to access an expansion slot were discovered, too. (What a discovery to learn that you didn't need to press RESET to turn a printer routine off; just type PR#0.) Before learning this simple operation, I had changed the printer routine to restore the data in addresses $36 and $37. Using PR#0 does this for you automatically. The ability of a user to hook input/output software through these addresses is one of the features that has made Apple so great. Apple never did try to hide anything from hardware or software developers. The number of peripheral boards and software packages available to the Apple owner is testimony to a great deal of foresight on the part of the Apple II creators.

Better and Better

Integer Basic was available almost immediately when you turned on your Apple, so was machine language. To use Basic, you typed CTRL + B (hard entry) or CTRL+C (soft entry). To return to the monitor you pressed RESET. The original Apples were designed with a very fast version of Basic included in ROM. Also included in ROM were utilities to work with machine language programs. You could assemble, disassemble, examine and change memory, and move programs around in memory. You could also access the speaker and the game paddle input/output pins directly. No other machine had this capability. If the lack of floating point arithmetic was a problem, there were routines in ROM for this too.

My Apple, a whopping 16K, also included the first version of Applesoft. It was a tape

version offered as an incentive to purchasers of 16K units and it had many bugs. But it included floating point arithmetic and I wanted that. My programming skills were more elementary then and I needed the extra programming power. The string handling capability was better, but it took me a while to find out why that made a difference. Eventually the tape version was improved and many hi-res graphics features were added. Soon after, the Applesoft ROM card became available. With only the "flick" of a switch, you could use all the capabilities of the Apple. Nowadays, all the Apple II Plus machines have the latest version of Applesoft resident in ROM.

And Still More

Following the development of Applesoft came the Programmers Aid ROM and then the Autostart ROM. Useable only with Integer Basic (it's in the $D000 ROM), the Programmers Aid ROM provides several utilities. You can test memory, make music, renumber and append Integer programs, and relocate machine language programs. There were also several routines for making hi-res graphics easier. The source code for all the utilities was also included. At the time, these were some of the more advanced aids available to the Apple II programmer.

You can still have the Aid ROM in your Apple II Plus. The Integer Basic card includes both the Aid ROM and the old monitor ROM. If you choose the Language System, the code in the ROMs is included on the Basic disks.

The Autostart ROM replaced the original monitor ROM. Once you made the replacement, you gained some features and lost some. Features lost included the mini-assembler and other machine language routines. Features gained included the auto

Apple Cart, continued...

boot or load-and-go capability, and some immunity from catastrophic RESET. With the old monitor, you frequently lost your program with an accidental RESET. The code in autostart was designed so you could recover most of the time—much more friendly for the programmer.

Note that with Integer Basic machines, you have both the old and new monitor ROMs in place. The Language system includes the new monitor on the 16K card. I read someplace that you could easily switch from one monitor to the other. I couldn't find the information to include it here, but I'll keep looking and use it in a future column.

And Into the Future

Several changes have been made to the main circuit board. Starting at level 0, the current level is now at 7. Most of these changes are transparent to the user. Some, like color killer, were obvious. Early machines would display text in blues and oranges when a color TV was used as a monitor. Color killer forced the signal to white. Other changes included international features, and removal of the 4K or 16K memory selection options. Since memory has become so inexpensive, you can now buy 32K for less than the cost of 4K in 1977.

Reset problems have been all but eliminated with the new keyboards. Early keyboards were longer than they are now. When you pulled open the top, the edge would push the top of the keyboard. Many keyboards were pried loose from their mountings this way. The first keyboard change was to make it shorter. The most recent keyboards have a piggy back board attached to them. This little board has the RESET saver switch on it. By setting this switch to the proper position, you have to press both the control and reset keys to do a RESET. You can also have the regular reset capability in the other switch position—a much needed improvement considering the precarious position of the reset key on the Apple II keyboard.

And how about all those older machines? How can they be protected from accidental resets? Several methods have been tried. They range from little hardware gadgets that protect the key from being pressed to software changes that direct the program to a safe place if reset is pressed accidentally. One of the best devices that I have tried is the Keyboard Enhancer. You get a combination of a lower-case adapter and reset protector when the Enhancer is installed.

Keyboard Enhancer

When I bought the Keyboard Enhancer, I had an ulterior motive. My 80-column board, also made by Videx, is directly compatible with the Enhancer. With the

April 1982 © Creative Computing

Enhancer installed, the Apple II keyboard works much like a typewriter. With no additional software, the shift keys shift from lower case to upper. Except for a non-standard keyboard layout, using the Enhancer with the Videoterm 80-column board gives me all the features of an 80-column terminal. By installing a short assembly language program, the same features, in the standard 40 columns, are provided. You also get the CTRL-RESET feature, and a character set with true descenders, but nine special characters are missing from the Apple keyboard.

Keyboard Enhancer.

Hardware

Components and circuits are contained on a small board. There are two versions; one for machines made prior to revision 7 and one for revision 7 machines. Recall that revision 7 machines already have the control reset feature. Installing the board is an involved but uncomplicated process. You have to disassemble the Apple II.

To make the necessary corrections, the encoder chip on the keyboard has to be removed. The instructions are clear but if you are not used to doing these things you might get some help. Additionally, the character ROM on the main board is removed along with two control logic chips. The three devices removed are plugged into the Enhancer board and the board is plugged into the sockets vacated by the three devices. Once installed, everything is partially reassembled. A check-out procedure is included to help you find out if it is working. This is easier to do with the 80-column board—you don't need to enter the machine language program.

After check-out is completed the Apple is buttoned-up and you're in business. The circuit board is made with quality components: the material is epoxy fiberglass, the kind known as G10, and all sockets are high quality Augat parts. In the electronics industry, these are considered to be premium components. Assembly and workmanship are high quality too.

Software

About all the software you need is provided in a short machine language program. The program is not needed if you are using it only with the Videoterm

80-column board. To make sure it worked however, I loaded it and it did. Memory in page 3 is used for the program. If you have used this space (and many peripherals do) you will have to find a way of exchanging overlays or relocating the programs elsewhere in memory. There are several patches for the Apple Writer Word Processor. With these patches you can use the Enhancer lower case capability without special codes.

Enhancer is compatible with the Z-80 Softcard, The Hayes Modem, and, if you are using the Videoterm, it is compatible with the Language System. In addition, if you are inclined, you can change the character set to suit your needs. Information is included in the manual to help you develop and program your own character set.

Documentation

All you need to know about installing and using the Enhancer is included in the manual. Written by Curtis White, Paul Davis and Darrell Aldrich, the manual covers installation, check-out, use, and theory of operation. Special capabilities, such as remapping the keyboard, are also included. To help you remap the keyboard, both standard keyboard mapping and alternate mapping data tables are included. Some of you may be familiar with Darrell's work from other sources. He has developed software for many graphics programs currently on the market.

Documentation on other Videx products has not been as well done, and my reviews of these products have been critical in the area of documentation. However, there are no significant problems with this manual. The presentation from installation to use to rolling your own EPROMS is well done. And for those of you who like to tinker with the parts, there is a schematic of the board.

Using It

Once you have it connected and checked out, using it is quite simple. At installation you choose the mode you want it to come up in. One mode is the standard Apple

"Mr. Atherton can't speak to you now. He's in combat."

mode; all caps. The other mode, the one I use, is lower case—shift for caps—mode. With the Videoterm 80-column board, all you need to do is turn things on and start typing. You use the shift key to make caps. Otherwise the input is in lower case for all the alpha characters. To go to the all-caps mode, just press the reset key. It will seem strange to use the reset key as a mode select key, but you soon get used to it.

If you want to do similar things without the 80-column board, you have to first load and then call a machine language program. Once you do this though, the operation is the same as with the 80-column board. Except, of course, you only have the standard 40-column display. But even so, operation is the same. You will use the reset key as a mode select key.

The board is useable with most of the word processors on the market. It works with my Super-Text II processor. And, the character set (font) is one of the better ones. The upper case shift is still an escape shift and can't use the shift key. Perhaps someday Muse will make a version that will work with my 80-column board. In the meantime I can still use the present techniques. But, more conventional shift capability with my favorite word processor would be nice.

Conclusions

If you want a good lower case adapter, and/or you have a level 6 or older machine, then you will like the Keyboard and Display Enhancer. Solving the reset problem was almost worth the price by itself. (Of course, there are less costly ways to solve only the reset problem.) My personal requirement was to have upper and lower case capability with the Z-80 card and 80-column board combination. This combination gives me almost identical capability with machines I use at work. When used with the standard Apple II, you can easily insert lower case text in your programs.

In any event, the Enhancer is one of the better products for increasing the flexibility of the Apple II personal computer. You can get the Enhancer for about $129 from computer stores, and from Videx, 897 N.W. Grant Ave., Corvallis, OR 97330. Remember, there are two versions. One for version 6 or less and one for version 7 or greater.

Why Not Both Sides?

Making copies on both sides of single-sided disks has been discussed from time to time. My position has been don't do it! At least don't do it without knowing the risks. Since the risks are many and there are many newcomers, let's take a look at the problems again.

The Disks

Floppy disks are made from a thin plastic material coated with a magnetic material

Apple Cart, continued...

on both sides. The result is similar to the magnetic tape used for audio recordings. Except, the magnetic integrity is more critical. Once the disk is coated and cut to size, it is tested for magnetic integrity. Several things are checked:

• Amplitude of the recorded signal—the output when read back cannot be below a certain level. This ensures that you will be able to read what you recorded on the disk.

• Missing bits—a test to make sure that voids do not occur in the magnetic surface.

• Extra bits—a test to make sure that data is erased. It is possible for a bit to be recorded and not erased. Not too good for your data integrity.

If any of these problems occur during testing, that side of the disk is not certified. As you can see, it is quite possible that one side will fail and the other will be good. These good-on-one-side disks are used for single-sided disks. Of course, with the present state of the art, it is also quite likely that both sides are good. It would seem then that merely checking to see if the other side is good is all that is necessary. Not so!

The Mechanism

Complete disk assemblies include the media inside a plastic jacket. The inside of the jacket is lined with a lint-free cleaning material. The purpose of the cleaning material is to keep dust and lint off the magnetic surface; as the diskette spins, the dirt is collected on one side of the liner.

Look at the photo of the two types of read/write head mechanisms. The one on the right is a single-side head. The other is a dual-side head. Single heads are designed to read only one side of the disk. What happens to the other side is a don't care situation. Dual side heads are different, of course. There are two heads to do the read/write job. As can be seen in the photo, one head is forced against the other with the media in between. Note that the mechanism is designed for the media to always be spinning in the same direction. The single side head is from drives like those used in the Apple II disk drives. The dual head is from a Shugart SA450 40-track drive.

Data is recorded on and read back from the media by the read/write head. As you can see in the photo, the read/write head is the thin black line across the white circle supporting the head. In the photo, notice the felt pad on the head load arm. This pad—the head load pad—is used to force the media against the head. As you use the disk drives this felt pad collects bits of magnetic material and dirt. In doing the job of forcing the media, the head load pad becomes abrasive.

Now, you can see what I am leading up to. If you use both sides of the diskette, several problems occur:

• Dirt collecting on the liner travels from side to side across the head, and added wear-and-tear on the head and the media results.

• The head load pad abrades both sides of the media increasing the wear there, too.
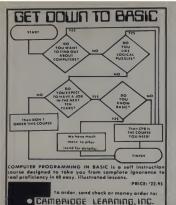
Because the magnetic coating is very durable, it may take a long time for any damage to show up. Shugart specifies qualified media at three million passes per track. Some time ago, I calculated that it would take eight years to reach this value (at two hours per day of use). But, by increasing the abrasion on the media with the head load pad, who can tell. All of a sudden one day you will have a problem with disks that you can't read. All your valuable data will be gone, and without warning. The Apple system does not have any read after write capability. (Except when you copy a disk, then the system does check what was written.)

Be Safe Not Sorry

If you do your own certification of the second side, you are likely to be safe under some conditions. If you use the disks for back-up there is little risk. And, for disks used only once in a while there is not too much risk. But for those disks you use frequently, use only one side. The risk of data loss is not worth the money saved by using both sides of any disk in a drive designed only for single-side recording. If you have a Language System, the Bad Blocks routine in the Pascal utilities will do a fair job of testing media integrity, and several utility programs are available to do media testing. Get one of them if you can. Or find someone who will help you do the job. For maximum peace of mind with the Apple disk system though, use only one side of your disks.

Next Time

The next column will feature a review of a program called Program Line Editor. Readers have submitted more programs and ideas. Some of them will be included. And, questions sent by readers will be answered. Also, a tutorial on how to convert Hex to decimal and the reverse will be included. □

Read/Write Heads.

outpost: atari

David and Sandy Small

One problem many manufacturers have is in documenting their hardware—especially for beginning programmers. The problem arises because beginners do not have a great deal of knowledge about the machine, and it is very easy to deluge them with unnecessary details and trivia they don't need just to get started. On the other hand, beginners don't stay beginners for long, and soon the details and trivia become necessary information.

One approach is to treat the user manual as a manual for a complete beginner. Everything is explained in the simplest possible terms, and if a concept is too difficult to explain, then it is ignored. The result is a fine beginner's manual that can take the user from the beginning to some point, but cannot continue from there. A user with this sort of documentation is then stuck.

A wonderful example is the Atari Basic Reference Manual. It is written for the beginning Atari user—someone who doesn't understand the capabilities of the machine. That's fine; we all started out pretty much at the beginning with the Atari, because it had such revolutionary ideas in its design. But the writers of this manual went too far.

They went to extremes to shield the user from the innards of the machine, and the result was complete confusion in some sections of the manual. For instance, the discussion of COLOR and SETCOLOR is abysmal. I had to work backwards, using the hardware manuals, to understand what they were trying to say in the Basic manual. Many people have commented on this to both Sandy and me. (Player-Missile Graphics were not even mentioned in this manual.)

Then there is the other approach. Tell everything. Heap it on—detail, trivia, useless data, anything, all mixed in with the relevant stuff. If a beginner reads this manual, too bad. An example of this is the Atari hardware manual. What does a beginner know of clock cycles, for instance? Or of Antic DMA? If a reader needs this information, he cannot get it.

Over the last few months (nearly a year now) we've been trying to fill this gap. We aim to explain the Atari's innards to the users who are past the Basic Reference Manual but who are bewildered by the awesome Hardware Manual.

And this month we're going to look at the Atari Disk.

For those of you who have read and been entertained by the discussion of the Atari disk in the Basic Reference Manual, this article will let you in on what is really going on. It is a ground level look at how the disk works and what the Atari does with it.

The information is quite useful and lets you access the disk more efficiently in many applications, as well as understand how Basic and DOS work with the disk at the lowest level.

This article, believe it or not, presented its authors with a real moral dilemma. Why? The information presented here can be used to copy disks that without this information are uncopyable. (Mind you, this information is also freely available in the Atari O.S. manual, if you can understand it.)

There is a horrendous software piracy problem in this country. Consider the effect on the record industry if every time a person bought a cassette tape, he made ten cassette copies of it and gave them to his friends. This is exactly what is happening in the computer industry. (Indeed, the record companies have come out strongly against the sales of blank tapes recently and will no longer support ads for those products, for just this reason).

Immediately after a new program appears on the market, it is copied. The copier then distributes copies to friends, trades them for copies of other programs, and so on. And the writer of the software receives no royalties from the copies of his program.

There are ways of making disks uncopyable. This seems like an ideal solution until you realize that a diskette can be destroyed. A phone can ring near it and erase it; a cat can use it for a scratching post (our cat, Atari, tried this trick once), and so on.

If a user is depending on the disk, and it quits working, then he is in trouble. For this reason the practice of "backing up," or making multiple copies of a disk, was started. Too many users had lost their only copies of badly needed material.

It's a good idea; I keep a minimum of two copies of everything on disk, so if I lose one, I still have what I need. And it has paid off; many times I have had to go to the backup copy when the original died for some reason. I don't know why, but I seem to lose an amazing number of disk files—error 144, error 164, and so on.

And if a manufacturer has made his disk uncopyable, you can't back it up. If the disk becomes unusable, you're out of luck.

The software industry has been debating this problem for ages. The Apple, in particular, has been the subject of controversy, with some manufacturers selling programs written for the express purpose of copying "copy-protected" disks. In their literature they describe how backup copies are necessary, and thereby justify the purchase and use of their programs. But they know, and everyone else knows, that these programs are used to rip off an enormous amount of copy-protected software.

Copy Programs

One program in particular, Locksmith, gave me a good laugh. Locksmith is an Apple program written specifically to copy

Atari, continued...

disks that are copy-protected, and has sections to handle all of the latest copy-protection techniques. It has caused a great deal of controversy.

In the *Locksmith* manual, the writers explain the need for backup copies, how disks can be erased, and so on, and condemn manufacturers for copy-protecting disks for this reason. Then, after going to great lengths to point out that *Locksmith* is for users to make back up copies of programs, the manual points out that *Locksmith* cannot make copies of itself.

In other words, they know what's going on — who's kidding whom? They even maintain a database from The Source which tells how to copy many of the copy-protected Apple programs using *Locksmith*.

Justice has been served, and there are now many, many bootleg copies of *Locksmith* floating around. As Scott Adams might say, "Yoho, and Jolly Roger."

Well, Atari users are in the same boat; things just haven't escalated since that far...yet. The Atari disk is a different animal from the Apple disk. An Apple disk program, for instance, is an incredibly complex machine language construction, as the Apple CPU controls the disk directly.

The Atari is different: a copy program is very simple. And some companies are using the fact that most people don't know how easy it is to copy an Atari disk to sell copy programs for fairly outrageous amounts. After all, to a user experienced with Apples, the price seems fair.

Frankly, they are a complete ripoff for the price. I do not feel that forty-odd dollars — the cost of a typical Atari disk-cloner program — is a fair price for twenty-odd lines of Basic code.

So much for the copy program makers. Now how about the poor software manufacturer? What do they do, now that I've revealed How To Rip Off Atari Disks? Send me letter bombs?

For the manufacturer's sake, I'll present a few of the latest techniques in copy protection — which the information presented here will now allow someone to copy. And there is no escalation possible; i.e., you won't be able to figure out a way to copy disks protected with this scheme. There is no way, using Atari hardware, to write a program to copy these disks — period. In this way, the word on how to make uncopyable disks will be spread, as will the word on how overpriced (for what they are) the Atari disk copy programs are.

Yes, there are some old copy protection schemes, which we'll discuss, that this information will allow someone to get around. These schemes don't work against the average pirate's copy programs and have not for some time; the disk information discussed here is old news to your average copy enthusiast.

I seriously doubt that anyone will be hurt by this information; if you are going to copy protect a disk, then you might as well use the techniques I will present which make it truly uncopyable, rather than the older ones which the copy programs can duplicate with no difficulty.

Enough philosophy. Let's learn about the disk.

The Disk

Take a disk and look at it. Inside that envelope (sleeve) there is a circular piece of very thin plastic. On that plastic is the same material of which cassette tapes are made: various magnetic substances.
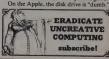
There are 40 "tracks" on the disk. Think of the circular grooves in a record and you'll get the idea of what a track is. There are 40 concentric circles on the disk and each of these tracks is divided into 18 "sectors." The division is done by pie-slicing the track into 18 contiguous segments. So we have 40 tracks with 18 sectors per track, or 720 sectors.

Atari disks are set up to hold 128 bytes of data (a byte, for you beginners, is one character) per sector. In other words, there are 720 x 128 bytes of data on an Atari disk, or 92,160 bytes.

To access a given sector of 128 bytes, that sector must be in the visible portion of the disk (the window cut into the disk sleeve), and the read-write head must move in or out until it's on the right track number. It then reads the sector by examining magnetic fields written to the disk surface.

Please Note that the actual recording surface is the back, not the front, side of the disk. Most people set disks down with the label up, which means the surface that the data is recorded on is being rubbed in the dust and dirt below. Also be very careful not to touch the backside of the disk in the exposed portion.

The Atari talks to the disk over the "serial I/O bus cable." This is the cable you daisy chain from device to device. Now here's the secret of the Atari disk: The disk drive is intelligent. It contains a 6507 microprocessor, a little brother to the 6502 in the Atari itself, that is still capable of quite a lot. It also contains 256 bytes of RAM (read-write memory), and 2048 bytes of ROM (read-only-memory). In short, the disk drive has a complete computer of its own. (For you hardware buffs, there are 128 on the PIA-like chip). How is this different, and why is this good? On the Apple, the disk drive is "dumb."

The main computer must tell the disk head to move to here, the disk to spin to there, and tell the head what information to write. This neatly ties the computer up while the disk is running.

Apple has a standard scheme for storing data, a "standard format," and if you use the Apple routines to read/write data, you'll stay in that format. Alas, that format allows your disk to be copyable, so people have modified the Apple disk routines to make custom weird formats, that cannot be read by copy programs. (And then other folks have written programs, such as Locksmith, to read "uncopyable disks" and then other weirder formats were developed, etc. It's a little like the arms race.)

On the Atari, the two computers work with each other. There are a net total of five, count them, five commands that pass between them. Since the 6507 in the disk drive is helping do the work, the Atari's main processor doesn't have to fiddle around with controlling the disk, and thus can be doing something else at the same time the disk is running. There are no other commands and the disk drive 6507 will ignore any but the five we'll discuss.

Best of all, since the Atari disk controller is "off limits" and may not be programmed, as its program is in read-only memory inside the disk and is not modifiable, you cannot develop strange disk formats. There is one, and only one, Atari disk format. Hence we'll never get into one of these Apple disk escalations, for we cannot control the disk that exactly. Believe me, after viewing the present Apple disk copy mess, we're not missing much.

An ordinary Atari user accesses the disk through what is known as the File Management System or FMS. The user never sees the lowest level of disk commands (the

"Malfunction!"

five mentioned) because FMS handles all that for him. Included in FMS is the ability to split the disk sectors, each 128 bytes long, into files, access the files through the operating system, get disk directories of file names, NOTE, POINT, and so on. FMS works in turn through the Five Commands (perhaps a movie should be made by that title?) with the disk drive 6507 to get your disk processing done.

FMS on disk is called DOS.SYS.

What are the Five Commands?

1. Get Status. This returns the status of the disk and in particular of the 1N 1771 floppy disk controller chip inside the disk. "What?," you say. Me too. I don't use, and have never used, Get Status. One day I may find out what it's for.

2. Format Disk. This command instructs the disk drive 6507 to lay out the tracks and sectors on the disk, and to "clean off" old data. Think of it as the 6507 laying down record grooves that it can follow later. What actually occurs is that the 6507 writes data across the disk, reads it and makes sure it has stayed the same (this verifies that the disk surface is good), and then writes the sector numbers onto the various sectors of the disk. This information it used later to find a given sector.

3. Get Sector. This command instructs the 6507 to get the entire 128 bytes of a

sector and ship them to the Atari through the serial I/O bus cable. The Atari issues the Get Sector command and tells the 6507 what sector number, from 1-720, to fetch. The 6507 then gets busy, spins the disk and positions the read/write head, reads the data, and returns it to the Atari.

4. Put Sector. This command instructs the 6507 to take the 128 bytes about to be shipped down from the Atari and put them onto the sector number specified. The 128 bytes are then sent, and the 6507 positions the disk, and puts the data onto it. You probably do not use this command, you use its relative, which is:

5. Put Sector With Verify. Remember, disks are not all that reliable. So when we put a sector onto the disk with this command, it is first written, then immediately re-read and compared with the original

128 bytes. If they match, fine; all is well. If not, then they didn't record correctly, and an ERROR message results. (In truth, I have never seen this particular ERROR message.)

The Atari's designers decided to add a little audio to this whole process, so whenever a Get Sector occurs (like during a program LOAD), a pleasant "beep" is heard on the TV's audio channel. Whenever a Put sector occurs (during a SAVE, for instance), a not-so-pleasant "clunk" is heard. You'll note that the Put commands always seem to take twice as long as the Get commands; this is because the disk is physically putting, then reading, the information onto the disk for a verify. Hence, it takes twice as long.

In the net time I've worked with the Atari, I've used Put Sector with no verify

once, when I was running OSS's operating system. They apparently use this command, and it sounds a great deal different than the standard slow "clunk..clunk.." of the Atari save.

Now *everything* that happens to the Atari disk occurs through these commands. Next month, we'll take a look at the process.□

215

rings...trs-80 strings...trs-8[

Stephen B. Gray

As we ride the crest of the 39th TRS-80 wave, we see ahead of us a maxim explained, a BS error explained, advice on other micro columns in these pages, how to make a graphics program work on a Model III, a look at the NRI Schools course that includes a Model III, Radio Shack's *Series I Editor/Assembler* and *Debug* monitor, a bowling game, *ASCII* magazine, and a doodle routine that simplifies drawing line graphics on the TRS-80 screen.

Maxim Revisited

The opening of the March column, referring to a short item at the end of the column, said it was "a maxim of no relation to hiram percy." Perhaps I should have explained that there are two famous Maxims: a restaurant in Paris; and Hiram Percy Maxim, inventor of the Maxim machine gun and, I believe, one of the founders of the Amateur Radio Relay League (ARRL), the ham-radio organization.

BS Error in Graphics Program

Although nobody lost any sleep over Hiram Percy, some readers may not have been able to figure out the last problem left up to them to solve in the explanation of W.A. Fronek's graphics-storage program in the Jan. 1982 column (p. 180).

The BS error occurs if you use the alternate pixel-display routine, beginning with line 10000, for more than a small number of pixels. If you have much experience with arrays, you'll remember that "if no DIM statement is used, a depth of 11... is allowed for each dimension of each array used," as the Level-II manual puts it.

So if you try to display a dozen or more pixels, you'll get the BS error. The solution is simple, of course: just insert DIM statements in the routine, as in line 9052. (If you think that's a simple problem just about anybody should be able to solve, think back to when you first met up with DIM....)

Those Other Micro Columns

If this is the only microcomputer column you've been reading in *Creative*, then you're missing out on some good and sometimes usable material in the other columns.

Most of the other machines, such as the Apple, PET and IBM Personal Computer, use Microsoft Basic. The Atari uses Atari Basic, but will soon have Microsoft Basic available.

Some of the biggest differences between TRS-80 Basic and the Apple, PET and IBM PC Basics are, of course, in the graphics and sound. Also, various statements are unique to each machine, and some are difficult or impossible to rework for the TRS-80.

However, making a program for another computer run on yours can be a fascinating challenge, as well as help you learn more about how your machine works. It also helps widen your horizons by teaching you something about computers other than your own.

As to what you can pick up from other columns, it all depends on who's writing them, and when. There were several usable items from the 1979 and 1980 PET columns, but not many (programs or columns) in 1981. The first Apple columns had programs that too often used disk memory or the tone generator; a few of the 1980 and 1981 columns have programs you can use with a little reworking. Keep your eyes on the IBM Personal Computer column, which may just have some programs you can pick up in 1982.

Rectangles Awry

Several readers wrote to say that Leo Christopherson's *Fast Rectangles* program (Dec. 1981, p. 328) doesn't work.

A reader in New York City sent a listing "of the program as I have it on my machine," with one line copied incorrectly from the original. However, even with that line correct, the program would give an overflow error on his disk system.

Stephen B. Alberg, of Cherry Hill, NJ, sent the solution:

"Line 130 of the program was left out. It should read

130 X0=-1*(65536-X0)

of course, but it took a while and some teeth-gnashing for this novice programmer to figure out why overflow error messages

were being printed when the program was run with the line missing.

"It would also be helpful to advise those of your readers with Disk Basic who, like myself, are relatively unfamiliar with the subtleties of USR routine programming, that the program will produce illegal function-call errors until they insert a program line such as 105 DEFUSR=& HFFDO."

Well, Leo's program runs fine exactly the way it was originally printed—in Level-II Basic on a Model I. However, for my 48K Model III disk system, Steve's two suggested lines have to be added. To figure out the hex value required for line 105, I loaded the program, ran it, and did a PRINT PEEK for addresses 16526 and 16527. This gave CA and FF as LSB and MSB of the entry point for the USR call, or &HFFCA, so if you insert this line

105 DEFUSR=&HFFCA

the program will run fine.

However, Model III Basic has no DEF USR function. So how do you make Leo's program work on a non-disk Model III?

Incidentally, figuring out the need for line 130 is quite an achievement for a "novice" programmer.

NRI Course With Model III

Perhaps you've seen the NRI Schools ad in these pages, showing a TRS-80, with the headline "Now NRI takes you inside the new TRS-80 Model III microcomputer to train you at home as the new breed of computer specialist."

The ad also shows a six-function LCD Beckman multimeter and a breadboard called the "NRI Discovery Lab"; the ad text says, "You don't just program your

Photo 1. This photo is part of the NRI Schools course that includes a TRS-80 Model III, a meter and a breadboard for hardware experiments with the computer.

computer, you go inside it...watch how circuits interact...interface with real systems...gain a real insight into its nature."

NRI Schools, which used to be the National Radio Institute, are, like Osborne books and Byte magazine, part of McGraw-Hill, as indicated in the colorful 100-page NRI catalog.

NRI offers eleven home-training courses in electronics, including video technology, TV and audio service, communications electronics, and microprocessors. That last-mentioned course is the first one in the catalog, taking a dozen pages to describe the ten kits and each of the lessons.

In Kit 7 you receive a 4K TRS-80 Model III without disk or cassette recorder; this

helps keep down the price of course, which is $2,375 if you pay in full; installment plans are available.

According to the catalog, after you get Kit 7, you learn Basic and assembly-language programming, and then "Using special multiconductor cables, you connect your computer to your Discovery Lab.... You write special driver and sense programs in both machine language and Basic to operate the internal circuits you construct....you build circuits that duplicate those one would find in typical computer applications: switch inputs, relay closure, indicator lights, amplifiers, digital-to-analog converters, and many more."

NRI Lessons

If you were to get all the digital lessons for this course at once, they'd weigh over a dozen pounds. The eighteen items consist of nine books (three sets of three each) that cover computers in general (Basic Computing, Computer Architecture, Microcomputers), seven on TRS-80 hardware, Basic and assembler (with hardware and software experiments) and two Radio Shack manuals: I/III Basic, and the technical reference handbook. There are other books, which I haven't seen, on the basic electronics portion of the course, covering resistors, transistors, relays, power supplies, etc., and involving a variety of experiments using the Discovery Lab and multimeter. (Perhaps I should call them booklets, but they're all 8½ by 11 inches in size, very nicely typeset and illustrated, and all fairly thick, so "book" seems a better word.)
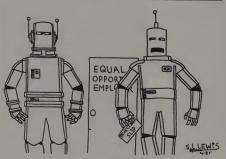
The NRI books (and the course) were all generated under the guidance of Louis E. Frenzel, executive director of the NRI Schools. Lou was at NRI previously, then spent several years at Heath, where he guided the computer effort. Lou returned to NRI, where they're still using, in the Digital Electronics course, the unique NRI 832 digital computer he designed, which uses a bank of 128 slide switches to demonstrate how RAM memory operates.

The nine basic books provide a slow and thorough grounding in the elements of computer electronics and programming. The Basic Computing books go into how computers are used, computer arithmetic, Boolean algebra, digital logic, flip-flops, counters, and arithmetic operations.

The architecture series looks at register transfers, addressing, memory, I/O devices, peripherals, data conversion, and data communications. The three on microcomputers involve applications, microprocessors, interface circuits, register transfers in software, higher-level languages, etc.

NRI TRS-80 Books

The seven NRI TRS-80 books first guide the student through programming the TRS-80, going into great detail on writing and

The only question I missed was 'are you male or female?'."

running short programs with, for example, 10½ pages on variables and strings. Many nooks and crannies of Basic are explored, such as the difference between @ and SHIFT/@, all written very clearly and informally.

The digital experiments with the Lab include logic circuits, counters, address decoding, data handling, I/O ports, and device control.

Self-test questions, with answers, are found throughout the books, each of which also contains multiple-choice "Lesson Questions" to be answered on what look like a machine-readable card to be mailed in to NRI Schools. The catalog says, "Using handy NRI Consultation Forms and by commenting on your tests and quizzes, your instructors answer your questions, help you over the humps...." Those Consultation Forms are for sending questions to your instructors.

If I were going to buy the course, I'd like to include a disk drive or two, with many lessons and experiments on disk hardware and software. However, that would probably add well over $1,500 to the cost, just for one drive and the additional lessons to go with it. At the moment, this is the only home-study course for the TRS-80, and if you've got the time and the money, it's well worth both.

Series I Editor/Assembler

Radio Shack's *Series I Editor/Assembler* "for assembly language programming, includes text editor and manual with Z-80 instruction-set information," as the catalog has it, and comes in two versions.

For $29.95 you get the cassette version for 16K Level-I/II or Model III; for $34.95, the disk version for Model I/III. The manual is apparently the same in both versions, with the addition of a mimeographed page for Model III owners, about using the left Shift key for typing certain symbols.

The manual, according to the indicia page, is "derived from the original Tape Editor/Assembler," copyrighted 1978 by Microsoft. The previous manual is 132 pages long; the rewrite covers 255 pages and goes a long way toward making assembly language seem less formidable than the shorter one did.

The third paragraph of the shorter manual says, "The assembler portion of the Editor/Assembler facilitates the translation of symbolic language source programs into machine executable code." The new version takes it a little easier: "Using the editor, you enter the machine-language source code, consisting of a convenient set of abbreviations and symbols. The assembler then converts or assembles this into object code, which the computer understands."

And so the longer manual goes, explaining things in more detail, and looking much less forbidding than the older manual, which

goes right into "notation conventions" on the first page; the new manual introduces them as "special terms" on page seven.

Most of both manuals is taken up with the Z-80 instruction set, but the 95 pages in the older text are expanded to 188 pages in the new edition. Much of the additional space is blank, because the older manual used a two-column makeup. Radio Shack may have decided this looked too crowded and grim, and spread out the material, which is almost exactly the same in the new edition, but with expanded examples here and there, and some additional examples now and then.

For the same $29.95, you now get a manual that looks much less like an engineer's reference handbook than the 1978 edition, plus some additional material that helps explain assembly language, which to most of us is the most complicated aspect of the TRS-80. Perhaps because of the added material, perhaps due to a higher level of computer sophistication among people who buy a TRS-80 today, the catalog no longer has an asterisk beside the *Editor/Assembler* item with the footnote "Intended for advanced programmers only."

The revised manual has an index, but still doesn't say a word about why you might, or should, be interested in assembly language. Nor does the catalog, apparently still on the assumption that if you have to ask, you don't need it. As noted in the earlier item in this column on assembler (Oct. 1979, p. 172), "Anything that can be done in any other language on a particular computer can be done in Assembly Language," and "Most real-time programs have to be written in assembly language, because

© CREATIVE COMPUTING

"I've been cooking him all day and he's still tough!"

Basic is too slow. If you wrote a Lunar Lander program in Basic, the lander would crash while the Basic interpreter was fiddling with the high-level language."

Debug

Whatever happened to *T-Bug*, Radio Shack's machine-language monitor for the Level-I TRS-80 that was featured in catalogs right up through RSC-5 in 1981?

The monitor has been improved, expanded and rewritten to work with 4K Level-I, 16K Level-II or Model III Basic, and renamed *Debug*. Catalog RSC-6 describes this $19.95 utility as used "For entering, testing and debugging machine-language programs. Allows modifications to RAM and registers, Level-II and Model III Basic include single-step execution of programs, breakpoint interruption of execution."

The original *T-Bug* (Oct. 1979, p. 172), was at first sold with a small, skimpy 12-page booklet that was later enlarged to a 40-page manual, and was described in the catalogs as "For advanced programmers only." RSC-6 carries no such warning for *Debug*, perhaps because the manual makes the monitor seem much less complicated than did the *T-Bug* book.

Debug has three commands in common with *T-Bug*: J for Jump, M for Modify Memory, and R for Change Register Contents. *Debug*'s T for Load System Tape, and W for Write a System Tape, seem very much like *T-Bug*'s L for Load and P for Punch.

Unique to *Debug* are commands for Single-Step, Return to Basic, Full-Screen and Half-Screen Display modes, and Update Display Continuously.

While the 18-page *Debug* manual doesn't contain the lengthy Z-80 instruction sets and memory maps found in the *T-Bug* manual, it does have the same handy list of subroutines in the Model I, Levels I and II, that can be accessed with *Debug*. These include Display Byte, Keyboard Scan, Save Memory to Cassette, and Output to Line Printer.

If you want to make the most efficient use of memory space and processing time, or to write real-time games, you should look into assembly language programming and *Debug*, along with the *Editor/Assembler*.

Tenpins

One of several clever TRS-80 games from Acorn Software Products (634 N. Carolina Ave. S.E., Washington, DC 20003) is *Tenpins*, which has just about all the features of real bowling, and is $14.95 for 16K tape, $20.95 for 32K disk, for Model I/III.

This machine-language game can be played in two ways. Beginners can position the ball left or right with two arrow keys, and start it rolling with the spacebar. As

the four-page instruction brochure says, "To start with, this is a simple game of random luck, but it will get you used to how to play."

Advanced players can put more force on the ball by holding down the spacebar, and/or change the direction of the ball, after it has started moving, with the two arrow keys. Thus they can control direction, spin and force. "A fourth factor is added at all times to stop excessive strikes," says the manual. "This amount is zero at the two edges and increases toward the center to simulate the imperfections in the lane, the ball and the player."

The ball gets smaller as it rolls down the lane, which is drawn in perspective. Pins are knocked over, dead pins are removed, and the player's score is displayed along with a miniature pinfall guide to show which pins are still standing. A full scoreboard is displayed at game's end, and can be examined during play.

Sound effects are provided: a tickety-tickety when the ball is rolling, a splat-splat for falling pins, and a brief fanfare for a strike.

One thing I didn't see was the flying pin that sometimes solves the problem of a split, although perhaps I didn't put enough spin and/or force on the ball.

If bowling isn't your thing, Acorn has other games you may like, including *Basketball*, *Astroball* (*Pinball* with added features), *Invaders From Space*, *Space Rocks*, and *Everest Explorer*.

ASCII Magazine

Two magazines ago, when I tried to check out *ASCII* magazine, I couldn't make the first game work. So I put it aside in favor of a game that did run.

Now I've checked out two *ASCII* cassettes, which are for 16K Level-II TRS-80,

and they are disappointing. The first, dated March 1981 (received in October), has a nice opening, showing a TV set, with a screen message of what's on the tape. However, the arrow keys don't move the car left or right in "Roadrace," so dozens of collisions take place. The "Baseball" game requires memorizing too many things—all of which could have been printed on an accompanying sheet—before playing, and getting on base is almost impossible.

The ski-slalom game, "ASCII Slopes," is too easy; most of the gates are in straight lines down the slope. "Morse Code" is one of the two good items on the cassette; with it you can learn Morse, take tests on it, and send code.

The other side of the March cassette contains notes about this cassette "newsletter," about the staff participating in a TRS-80 show and introducing computers at local school, all a waste of tape, all of which could have been printed up on the sort of pages *CLOAD* magazine encloses with each cassette.

The next item couldn't be loaded at all. The last item included four trace puzzles without any explanation, but which use a computer to do what is usually done—much cheaper—with pencil and paper: trying to trace all the parts of a geometric figure without lifting the pencil, each part only once.

The April 1981 issue, which I received in December, is apparently a quickie attempt to catch up, and is a total waste of tape. All but the last item could have printed up, and saved the user time and money.

After overlong comments that "*ASCII* is quickly moving back on schedule," there's a diatribe against computer thieves who circumvent passwords and such, a juvenile discussion of debugging, two ads, an incompetent piece on word processing in

the office, another quickie episode in a continuing history of mainframe computers (why?), and a sophomoric puzzle.

As evidence that this cassette was prepared quickly in a poor attempt to catch up: just about all of it could have been supplied on a typed sheet or two; spelling errors abound; only half of one side of a 30-minute cassette is used (recorded only once); and the material is mostly trivial, nothing for which you'd want to pay $40 a year for 12 issues.

One funny item: "*ASCII* is now being examined by "TRS-80 Strings"...known for its discerning glances at various products for the TRS-80. *ASCII* would like to receive a most favorable review and we are looking forward to its appearance." Sorry about that.

But maybe I'm too prejudiced in favor of *CLOAD*, which in its November 1981 issue has a "Bounce" game that alone is worth both issues of *ASCII*. However, *CLOAD* had a few problems at first, too. Maybe *ASCII* will improve. You might try sending them $5 for a sample issue in mid-1982, if they're still in business at Box 516, Valley Stream, NY 11582. But as of now — early 1982—*CLOAD* wins hands-down.

Short Program #28: Doodle Routine

Dan Rollins of Azusa, CA, sent this draw-a-line program:

"I have owned my Model I for just over 1½ years. In that time a $499 Level-I machine has been transformed into a 48K Level-II with dual drives, an IDS 440 Paper Tiger with graphics option, and a direct-connect modem "The Connection." During this time I have fanatically read everything on the TRS-80, learning a great deal about Basic and machine language.

"I would like to pass on to your readers a few ideas I learned in the form of the Basic program in Listing 1.

"Lines 100-200 are the best all-purpose doodle routine I've seen. Though it uses ideas from many sources, it's all original code.

"Line 110 blinks the cursor at the current position.

"The PEEK in line 120 is an INKEY$ without the disadvantages of that command. It continually strobes the area in memory-mapped I/O for the arrow keys, ENTER, CLEAR, SPACEBAR and BREAK. This means that as long as a key or keys are pressed, the action is repeated.

"Lines 150 and 160 make use of the 'logical' variable to increment or decrement the value of the X,Y coordinates according to the value of the depressed keys.

"Lines 170 and 180 use the same type of logic to keep the cursor on the screen. To understand the principle here, just remember that the expression between the parentheses is evaluated as either -1 (true) or 0 (false). Multiplying such an expression by

"You're being replaced, Cargroove."

OKIDATA

Making small printers for people who think big.

OKIGRAPH™ Dot Addressable Graphics Prints Circles Around The Competition

Or squares. Or charts. Or graphs. Or illustrations. Or any graphic data your computer tells it to print. All with superb clarity and resolution.

OKIGRAPH™ is standard equipment on the Microline 84 and the Model 2350. It's an easily added option to the Microline 82A and 83A. So don't settle for the plain old text you get with other printers. Be creative. With Okigraph.

Interested? Contact us at 111 Gaither Drive, Mount Laurel, N.J. 08054, or call (800) 257-7768 for the name of your local authorized Okidata distributor.

Okidata is a subsidiary of
Oki Electric Industry Company Ltd.

- OKIGRAPH™ I includes two plug-in PROMs and an instruction manual on Assembly language programming.
- OKIGRAPH™ II includes a floppy disk for additional graphics printing features and an instruction manual.

Microline 82A

Microline 83A

Microline 84

TRS-80, continued...

a value returns either that value's negative counterpart or 0. This idea may be used advantageously to avoid the IF question.

"Lines 2000 to 2080 constitute a novel use of the VARPTR function of Level-II Basic. In this case, the interpreter is tricked into thinking it is storing a string in video memory.

"Line 2060 takes that string and places it in high memory where string arrays are stored. From here the array may be sent to a disk or tape file for recall at a later time.

"The saved strings are restored to the screen in lines 3000 to 3060. Note that 1,024 characters cannot be PRINTed to the screen. Even with a trailing semicolon, the final character forces a line-feed. This is overcome by POKEing its ASCII value into that spot of video memory."

With the spacebar down, the cursor moves without writing. The four arrow keys, singly, direct the cursor in the four main directions. Used in pairs, they direct the cursor in directions at 45 degrees to the horizontal. The SHIFT key erases the dots; CLEAR will clear the screen.

If you don't want to save any of your doodles, delete lines 1999 through 3060.

Writing can be speeded up by decreasing the delay-loop times or by removing the T-loops in line 110. However, the faster the writing, the harder you find it to stop at a desired point. □

```
10 CLEAR 2000: DIM A$(15)
99 '** ALL-PURPOSE DOODLE ROUTINE **
100 CLS: X=60: Y=23
110 IF POINT(X,Y),RESET(X,Y):FOR T=1 TO 10:NEXT:SET(X,Y)ELSE
    SET(X,Y):FOR T=1 TO 10:NEXT:RESET(X,Y)
120 K=PEEK(14400)
130 IF K=1 THEN 2000                    '<ENTER> TO CREATE $ ARRAY
140 IF K=2 THEN 100                     '<CLEAR> TO CLEAR SCREEN
150 Y=Y+((KAND8)=8)-((KAND16)=16)       'UP/DOWN ARROWS
160 X=X+((KAND32)=32)-((KAND64)=64)     'LEFT/RIGHT ARROWS
170 X=X+(X>127)*127 - (X<0)*127         'SCREEN WRAP-AROUND
180 Y=Y+(Y>47)*47 - (Y<0)*47
190 IF (KAND128)=128 THEN 110           '<SPACEBAR>+<ARROW>
195                                     '= TRAVERSE W/O CHANGE
200 IF PEEK(14464) THEN RESET(X,Y) ELSE SET(X,Y)
203                                     '<SHIFT>+<ARROW>
206                                     '= ERASE DOT
210 GOTO 110
1999 '** ROUTINE SAVES SCREEN IN A$(0-15) **
2000 FOR J=0 TO 15
2005 I=15360 + J*64                     'LINE BY LINE
2010 S$="*": V=VARPTR(S$)
2020 MSB=INT(I/256): LSB=I-MSB*256
2030 POKE V,64                          'CHANGE LENGTH
2040 POKE V+1,LSB                       'POINT TO CURRENT
2050 POKE V+2,MSB                       ' SCREEN POSITION
2060 A$(J)=S$
2070 NEXT
2080 GOTO 2080
2999 '** RESTORE SAVED ARRAY TO SCREEN **
3000 CLS
3010 FOR J=0 TO 14
3020 PRINT A$(J):
3030 NEXT
3040 PRINT LEFT$(A$(15),63):
3050 POKE 16383,ASC(RIGHT$(A$(15),1))
3060 GOTO 3060
```

Will Fastie

I had some trouble deciding how to open the column this month. I like to be chatty and witty and swave, but I can't always do that at 5:00 a.m. Worse, though, I couldn't quite settle on the tone. So I've decided to let you decide which one is best.

Intro #1

63 days.

Can you believe it? Over two months! I could have just walked into any computer store and bought an Apple or something right off the shelf, but noooooooo, I had to order an IBM.

Intro #2

YAHOO! YIPEE! WO-YAH! YIP YIP YIP YEA! HURRAY! YEAH! I got it!

Take your pick. In any case, see Photo 1. And let me tell you all about it.

Unpacking Will's System

My system arrived in four boxes and a shopping bag. The Baltimore IBM Product Center had configured the system for me and installed all the options I bought. Then they checked the system out and repacked it for my pickup. The shopping bag had the boxes in which my installed options had been shipped, as well as the unneeded shipping bolts and back panel covers.

My system is a 64K, single disk system which includes the IBM printer, both display adapters, the game control adapter, the Monochrome display, DOS, and Easy-Writer. I'll offer just a couple of words of

W. H. Fastie, 7110 Sheffield Road, Baltimore, MD 21212.

Photo 1. The author's system. Note covey of joysticks.

explanation on my configuration, ignoring my financial limits. I bought the Monochrome Display/Printer adapter and the Monochrome Display because I will be using the system primarily for word processing. I bought the Color/Graphics adapter because I want to have the graphics capability. If I had been limited to one display, I think I would still have chosen the IBM monitor, again because of the word processing use.

The system unit and keyboard came packed in nice-looking boxes, each with pictures of the contents on the outside. Both were well-packed. The system unit box had a cardboard tray on which the manuals and power cord were packed. The Monochrome Display and the printer were also packed well, but came in non-IBM boxes which were typical of the country of their origin. My only complaint about

the packing was that the keyboard had little pieces of styrofoam from the packing material lodged between the keys—it took me a few minutes to clean it all out.

The two manuals delivered with the system are the Guide to Operations and the Basic manual. The Guide to Operations gives all the information you need to assemble and operate your system, regardless of configuration. It also includes a diskette (or cassette) of diagnostics. I owe IBM an apology about that.

I have previously complained about the cost of the advanced diagnostics package, saying that provision should be made for good diagnostics in the hands of the users. The package delivered with the system is the one I tested in Boca Raton, and is more than adequate for owners and users. It is now clear to me that the advanced diagnostics are for repair purposes.

If you follow the instructions in the Guide to Operations, you should have no difficulty putting your system together, connecting the parts, or getting it started. You should also not have any trouble locating problems, in the event your system is DOA (dead on arrival). The Guide also has some very valuable information on parts of the system which is not contained in other documents, *so I recommend a very careful reading.*

My system came alive right away, passed all the diagnostics, and settled in for a long stay. I promptly stepped out to stock up on a few ancillary items. At Radio Shack, I bought the CTR-80A Computer Cassette Recorder, a set of joysticks, three diskettes, and a couple of cassette tapes. You'll recall that I previously bought the RF modulator.

I got some parts to build my joystick adapter box, and some paper to feed my printer.

My first project was the connection of the cassette recorder. This was quite simple, as all the necessary cables are provided with the recorder. The only question I had was which way a jumper on the system board was set. This jumper determines whether the microphone or auxiliary jack will be used for cassette data output, and is factory set for the microphone jack. This caused me to open my unit for the first time so I could verify this and other jumper settings, install the RF modulator, and generally look around.

I found something that bothered me. IBM provides a plastic card guide in slot 2 to support the end of the long display

adapter. I had hoped, since I was purchasing both display adapters, that there might be another card guide in my system. Not so. At first I ignored the matter. A few weeks later, however, I was inside the system again testing a different RF modulator and I noticed that my monochrome card, the unsupported one in slot 3, had warped considerably (1/8" is considerable) in the direction of the Color Graphics adapter. I promptly called the IBM Product Center to order an extra one, but I don't know if I'll get it or not. IBM should install these guides in each and every slot. Until they do, third party vendors with long cards should provide the guide with their products.

My next step was installation of my RF modulator and making the TV connection. The Sup'R Mod II installed just as I thought it would, that is, electrically. It turns out that the wires leading from the graphics board to the modulator, which are wound around a ferrite core, do not allow convenient placement of the modulator inside the system unit.

I talked to Marty Spergel, the President of M&R Enterprises (oops, I called them "Industries" before—sorry) who said that the purpose of the core was to keep the signal as free as possible from interference. However, he also said that the IBM was a lot cleaner than most, and that the core could probably be removed without affecting either the quality of the modulated signal or your neighbor's TV. M&R does suggest "caution and courtesy" in their instructions—you should check with your neighbors in any event.

If I remove the ferrite core, the modulator fits nicely on the side of the power supply, with the wires trailing over the top rear of the cards. I haven't done that yet, because I'm still in an RF modulator testing mode. The first one I got for testing was a new one from M&R Enterprises, the Sup'R Mod V. This device is FCC approved, very nicely built, and connected the same way as the Sup'R Mod II. The big difference is that there is only one box. This box attaches to your TV and includes a switch for selecting the computer or the TV and a switch for selecting either VHF channel 3 or 4. This is the other difference between it and its older brother—the model II wants UHF channel 33.

Both modulators work very nicely and produce good displays. I tried them both on my 17" Sony, my 13" Toshiba, and a little 9" black and white TV which is attached to the system as my graphics display device. I think the Sup'R Mod II is a little sharper, because in 80-column mode I can read my 9" TV a little better. I think this is mostly a function of being on channel 33. My friend Tom has seen both in action and has borrowed the newer one, which he says he likes very much. I have no hesitation in recommending either one, with a few extra points to the V because of

Photo 2a. Datamac's Memory Expansion Board.

Photo 2b. Chrislin Industries' Memory Expansion Board.

its nicer package and FCC approval, although it does cost more.

What I Don't Like

I promised you last month that I would tell you what I didn't like about the IBM Personal Computer. Actually, I can't think of anything I really dislike about the system. What I am feeling is a sense of frustration brought on by the way in which the system is being marketed at the outset. This, in turn, is a comment on the tendency of IBM to be conservative. To be fair, this is precisely the reason why the IBM turned out to be such an excellent product, why I bought one, and why I write about it. The price we pay is waiting for "new" developments, the ones all the other vendors can offer already.

Memory

My first gripe is the amount of main memory that can be installed on the system board. I would have liked to see 128K there, with 128K and 256K expansion boards. This is a prime example of IBM conservatism. IBM, as a company, certainly has the technology of high-density memory well in hand. What ended up in the Personal Computer was standard, vanilla, 16K RAM. Besides being not particularly innovative, it is wasteful of space. There is much speculation about the possibility of replacing the on-board RAM chips with the newer 64K DRAMs, but I don't know if it can be done or not.

Fortunately, at least two companies, both with experience, have come to the rescue. Both are offering memory expansion boards for the IBM, boards that can be configured in increments of 64K to a maximum of 256K. They both use 64K dynamic RAM chips. The companies are Datamac Computer Systems, a pioneer in the field of plug-compatible memories for IBM computers, and the Computer Products Division of Chrislin Industries, Inc., which also manufactures add-in memory for Digital Equipment PDP-11 minicomputers. Datamac told me that their product had been selected for sale through Computerland stores. Further information on both products is shown in Table 1, and Photos 2a and 2b show both devices.

Disk Storage

I was disappointed to hear that the IBM Personal Computer would have diskette drives of 160K capacity. For personal use, this is marginally okay. For professional or business use, it is woefully inadequate. I wish that IBM offered double-sided disk drives as well as the single-sided ones. They really don't add that much to the cost or the price, but they make *such* a difference in the usability of the system.

IBM's Pascal compiler, for example, is so large that it alone requires two diskettes. The logistics of using Pascal on the IBM system are messy at best, and could be dramatically improved with larger disks. Larger disks also lend themselves well to standard business applications, particularly inventory or mailing list programs.

IBM knows that they're doing on this one, though. I see two reasons why this approach was taken. The first, and perhaps most important right now, is that the availability of larger disks and the resultant software to take advantage of them could have a negative impact on the sales of both the DisplayWriter and the Datamaster System/23, both of which have 8" disks. The second reason is that the Personal Computer will sell just fine without the larger disk, and IBM can waltz in later with new and larger drives for those customers seeking an upgrade.

In the meantime we will have to look to the third party vendors for disk systems. There are two recent announcements, and several more companies advertising. One of the first to announce a Winchester is Corvus, which isn't so surprising. You can be sure that the market will be rich (or lousy?) with Winchesters by this summer, so you'll have your choice.

The most exciting announcement so far, however, is for an add-on diskette drive at a cost of only $265! This is from VR Data Corporation, in Folcroft, PA. The drive, according to VR Data's President Warren Rosencrantz, is the exact Tandon drive used by IBM and which VR Data has been selling for some time as a TRS-80 Model III add-in. I'm getting a drive to test, and I'll let you know how it goes. The price sure beats the IBM list of $570—you could save $610 on a two-drive system.

One good fact to keep in mind about this is that IBM buys its drives from Tandom, a company with a broad product line including higher density mini-floppies and a new, "slim line" 8" disk drive, two of which would fit in a slightly modified IBM system unit. I hope IBM and other vendors move quickly on expanded disk storage.

Printers

My complaint about the printer is almost silly, but just worth mentioning. I think the wrong printer, hold by just a little bit. A better choice would have been the Epson MX80 F/T with the Graftrax option available. I for one would like to have friction feed capability, but without sacrificing the tractors. The Graftrax option could allow both alternate type fonts and printing of the entire character set that can be displayed on monitors.

Friction feed is important enough to me that I'm looking for ways to get there from where I am now. Next month I'll have some further information for you, but one excellent idea comes from Mr. Philip B. Sturmfels in a letter to *Creative* published in the December issue. He describes a carrier made from two connected pieces of printer paper. I tried it, and it works! Cheap, too. Try it, you'll like it!

A final complaint is that IBM is not offering a letter quality printer. This is not too serious, and to be expected considering my earlier comments about DisplayWriter and Datamaster. Printers from third parties will also be significantly less expensive, encouraging defection. This is already the case with the Epson Printer which is much cheaper from almost anybody else.

Expansion Potential

A ready complaint from almost everybody is that the IBM has too few expansion slots. I don't completely agree with this, since I think five slots will accommodate a large proportion of the owners. However, IBM has not been innovative enough with its controllers. For example, why not a single board with the Monochrome and Color adapters on it? How about a board with two, rather than just one, communications port? And the game
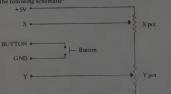
```
10 REM Program to watch the joystick ports
20 REM Will Fastie - 23 Dec 81
30 KEY OFF: CLS: STRIG ON
40 F$="#### #### ####   (##)  #### #### (##)"
50 LOCATE 8,1: PRINT "   Joystick A          Joystick B"
60 PRINT "   X   Y   T         X   Y   T"
70 AX=STICK(0): AY=STICK(1): AT=STRIG(0) OR STRIG(1)
80 BX=STICK(2): BY=STICK(3): BT=STRIG(2) OR STRIG(3)
90 LOCATE 10,1: PRINT USING F$;AX,AY,AT,BX,BY,BT
100 GOTO 70
```

Table 1. Memory Expansion Board Specifications.
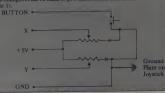
	Datamac	Chrislin
Maximum Capacity	256K	256K
Minimum Capacity	64K	64K
Increments	64K	64K
Parity?	yes	yes
Memory Device	64K bit DRAM	64K bit DRAM
Cycle Time	*	400 ns
Access Time	*	225 ns
IBM Slots Required	1	1
Card Size	4.2" x 9"	4.2" x 13.1"
Power Requirements	+5 VDC	+5 VDC
Power Consumption, maximum	450 milliamps	1000 milliamps
Power Consumption, standby	250 milliamps	700 milliamps
Price, 192K	$1,099	$1,050
Price, 256K	$1,399	$1,150
	* data not available	

How to Modify the Radio Shack Joysticks

The Radio Shack joysticks are the easiest to modify, assuming you have overcome the hurdle of the DIN connector. For other joysticks, the objective is to rewire them so they match the following schematic:

The Radio Shack joysticks are wired as shown below. The changes I made to mine are indicated by an "X" for a cut wire. Additional changes could be made to provide a normal orientation (see Table 1).

control adapter is so small, why wasn't it just put on the system board?

IBM has done the right thing with the Monochrome Display/Printer adapter. It remains to be seen if IBM or other vendors meet this challenge. If someone builds a two-port communications card with integral game control adapter, I'll buy it. I know how such a thing can be built. The disk system vendor who builds a Winchester that will fit into the system unit and includes, on a single board, both the Winchester and floppy interfaces, will have built a highly desirable product.

One vendor has already announced an expansion chassis, as well as an entire line of add-on and add-in products, for the IBM Personal. The line, called the TecMate series, is available from the Personal Computer Products Division of Tecmar, Inc. I don't have any technical details on their products yet, but they claim 21 items are ready to go, including memory and a Winchester Disk.

Joystick Report

When I got my system, I began the process of connecting the various joysticks which I had collected. In addition to the Keyboard Company joystick II and the Peripherals Plus Apple Joystick, I also obtained a set of Radio Shack joysticks. I'm sorry to say the results were disappointing.

I first set about to connect the Radio Shack set, primarily because they matched the IBM specifications. I went to some trouble to find female 5-pin DIN plugs to match the joysticks, my purpose being the construction of an adapter box capable of connecting either the Radio Shack or Apple joysticks. I finally gave up, cut the DIN connectors off, and replaced them with a 15-pin D-shell connector.

Thus connected, I wrote a simple program (see Listing 1) to examine the Game Control Adapter ports so I could see what kind of values the sticks were returning. They didn't do so well. It turns out that the sticks have to be modified in order to work properly with the IBM. The modification is relatively simple, involving only the cutting of three wires and the connection of two of them together. The sticks work almost perfectly. The "almost" is because they return their highest values from the upper left position and their lowest from the lower right position. This is, of course, counter intuitive, but can be dealt with in software. Once I had adjusted the values and scaled them appropriately, I was able to write an "etch-a-sketch" program with no trouble.

The two other joysticks proved tougher. I first built an adapter cable with DIP sockets in a box on one end and the 15-pin plug on the other. I was thus able to plug in both joysticks at once, or to try either in

Table 2. Joystick Comparison.

	Radio Shack	Keyboard Co.	Peripherals Plus
Modification*			
Required?	yes	no	yes
Complexity	easy	harder	harder
Orientation (position of origin)	lower right	upper left	upper left
Resolution**			
X-axis	93	143	167
Y-axis	93	145	137
Range**			
X-axis	96 - 3	3 –146	3 170
Y-axis	96 - 3	3 – 148	3– 140
Packaging	good	excellent	poor
Hand Comfort	good	excellent	fair
Button Comfort	fair	very good	fair

*Button on Keyboard Co. Joystick returns inverted value. Button on Peripherals Plus Joystick will not respond unless modified.

**Resolution is the number of values returned by Basic for the STICK functions. Range is the actual numeric range returned. High values from Peripheral Plus sticks may be adjusted with provided trim pots.

socket A with my "etch" program. The Keyboard Company joystick worked right off, except that Basic sees the button as always depressed, that is, always on, and sees it off when the button is actually depressed. The Peripherals Plus joystick also worked, but neither button would register at all. In fact, neither joystick is wired to IBM specifications and both must be modified to provide the expected operation.

I have not yet modified either of these joysticks, and although the modification is more complicated than that for the Radio Shack sticks, it is still not too difficult. Both the IBM Technical Reference manual and the Guide to Operations have a schematic for the expected joystick (see box). Anyone with a little bit of technical know-how will be able to modify any joystick (with potentiometers) to meet this specification.

Table 2 summarizes my technical findings for these three joysticks. Of the three, the only one that worked without modification was the Keyboard Company Joystick II. It is also the nicest looking and is very comfortable to hold. The Peripherals Plus Joystick seems to have the best feel and was most stable electrically. The Radio Shack joysticks are, without question, the best bargain, and are the easiest to modify.

Next Month: Displays and Printer.

Firms Mentioned in this Column

M&R Enterprises, 910 George St., Santa Clara, CA 95050. (408)980-0160.

Datamac Computer Systems, 680 Almanor Ave., Sunnyvale, CA 94086. (408)735-0323.

Chrislin Industries, Inc., Computer Products Division, 31352 Via Colinas, No. 101, Westlake Village, CA 91362. (213)991-2254.

Corvus Systems, Inc., 2029 O'Toole Ave., San Jose, CA 95131. (408)946-7700.

VR Data Corporation, 777 Henderson Blvd., Folcroft, PA 19032. (800)345-8102 and (215)461-5300.

Tecmar Inc., Personal Computer Products Division, 23600 Mercantile Rd., Cleveland, OH 44122. (216)464-7410.

The Keyboard Company, 7151 Patterson Dr., Garden Grove, CA 92641. (714)891-5831.

Peripherals Plus, 39 E. Hanover Ave., Morris Plains, NJ 07950. (800)631-8112 and NJ (201)540-0445.

Que Corporation, 6515 E. 82nd St., Suite 110, Indianapolis, IN 46250. (317)842-7162.

The IBM Version of EasyWriter

I had not planned to give my impressions of the EasyWriter Word Processor this soon. Several months of using it would have provided a more seasoned, objective report. Nor did I want to do a formal report, because such an article requires more of an understanding of software in this class than I have.

In many ways, however, my background is well-suited to a critique of a word processor. I have studied the available software for mini-computers quite extensively and I have usually been involved in the selection process for such software for my company. The section in which I work has also produced text and document processors. On a daily basis, I use a mini-computer based word processor.

This background gives me a view downward, a perspective a little different from that of many software reviewers. And, because I have deep experience with the implementation of powerful systems on small computers, I am little impressed with arguments justifying minimal function based on limited resources.

That's why I'm so disappointed with IBM's choice of EasyWriter for the Personal Computer. It's not that EasyWriter is a bad word processor, although it can't begin to compare with my usual choice. For very simple (Easy?) things, it's okay. For anything more complex, like this column, I think it is very deficient. I would not recommend this system for professional work, and I wish I had some other choice.

What's wrong with it? My most serious objection is performance. For everything except initial entry of text, or entry of text at the end of an existing document, EasyWriter is very slow. It cannot even keep up with slow typing for insertions. Deletions of all kinds are slow as well. Since I, and I think most writers, write using successive refinement, the ability to make rapid modification is essential. A regular text editor could keep up with initial entry—word processors are for *revision*!

The performance is pretty bad, because normally my first observation about highly interactive systems is their human factors. EasyWriter has many flaws in this regard, the most

notable being inconsistency. There are many examples, but a very obvious and annoying one is that the behavior of the system after executing a command from a menu is not always the same. Sometimes the system returns you to the menu, but sometimes you are returned to the edit mode. You have to learn each command, and what it does. That should not be necessary.

The extended features of EasyWriter, the ones I need very much, are not very easy to use. The system provides an alignment capability, which will reformat your text after you have made extensive revisions. Alignment protection markers are provided so that tables, headings, or other specially formatted text can be protected from the alignment process.

Unfortunately, after the first alignment some of the markers may move and not be seen for subsequent alignments. It is necessary to restore them to their former positions. Nor can a protection marker be placed at the beginning of a file; the IBM manual suggests putting an imbedded command as the first line in every file to allow the marker to come next.

I also had some problems learning how footers, page numbering, and centering worked. In the case of centering, I tried to center a line with printer controls in it. EasyWriter counted them as characters, so the line did not really print centered at all.

A feature I like, word counting, doesn't work too well either. The manual explains that "any string of characters between two spaces" is a word. So things like dashes and imbedded commands get counted. *Imbedded commands?* Doesn't EasyWriter know how to identify its *own controls?* Apparently not in this case. The manual recommends putting in extra words to compensate for these counted "words."

That brings me to the IBM manual. Although well-written
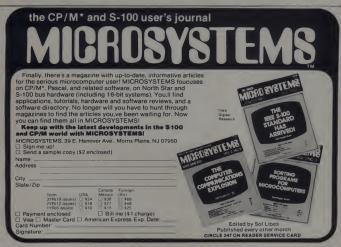
and produced, in keeping with the rest of the documents, it never explains how things work. It is definitely tutorial in nature and requires much patient reading to extract all the facts. I have spent too much time experimenting to see how something really worked to feel that the document did its job.

Finally, there is my standing complaint that EasyWriter has its own file system and does not deal with DOS files. This means that you cannot develop a VisiCalc report, or anything else, for integration into an EasyWriter document. It also means that documents cannot be processed by anything but EasyWriter.

IBM told me, back in September, that EasyWriter had been chosen because its price hit the middle. I'm sure that's true. I also believe that it was chosen because it could be produced rapidly, being written in Forth. That's probably the reason the performance is so poor. I'm thinking that another reason for its choice is that it cannot compete with the DisplayWriter.

I got a very good book from Que Corporation in Indianapolis called Apple II Word Processing. Que has been doing reviews of small computer software and has developed a format which I think is pretty good. Their review of EasyWriter Professional for the Apple II gives it rather high marks, very close to WordStar. They have noted most of the deficiencies I have mentioned. I understand from the president, Richard Summe, that Que will have reports available for the IBM system soon.

Here's my advice. If you're serious about a word processor, wait for something else from a third party. If you want to get going sooner, use EasyWriter first, for a long time, before you buy. Make sure it can do what you want, and that its limitations don't make a difference to you. And if you want a more objective review, the Que evaluations and previous *Creative Computing* reviews of EasyWriter will help. □

Barry D. Bayer

ᴍ...software legal forum...sᴏ

In my prior columns in which I have been trying to state the case for the bewildered software purchaser, I discussed the legal effects of whether software can be considered goods and an explanation of the warranty given. In this column, I will discuss the legal implications if the software just won't work. In this column, this means that the seller breached one or more of the warranties.

As I said in a prior column, if the seller dragged out this so called warranty agreement which is more a disclaimer of warranty agreement before selling you the product, and refused to sell you the product unless you read and signed the agreement (and perhaps required you to have your attorney present to explain its meaning to you), it is likely that most of the seller's disclaimers of warranties would be effective. At least they would be effective to the extent that the program worked. If the item just didn't work at all, it is possible that a court would say that these disclaimers deprived the purchaser of the very essence of his bargain, and that the buyer could at least get his money back. In either case, the buyer would have an uphill fight.

However, if (as is more likely) you simply opened up the package when you got home and saw this "Warranty" sheet for the first time, it is likely that you would be able to persuade a court that these disclaimers were not a part of the contract. You could argue that the seller tried to make you agree to this limitation of your rights only after you purchased, and that the express warranties derived from advertising and specifications, and the implied warranties of merchantability and fitness for a particular purpose continue to protect you.

Besides...you know that the seller is reputable, (after all it was the same company that built the computer) and besides you

need the program. So you load in the operating system, instruct the computer to run the program, and then can't get the thing to work. Upon reading the last lines of the last page of the last appendix to the manual (when all else fails, read the manual) you discover that although the program was released only a month earlier, it was designed to run under a version of the operating system which has been obsolete for nine months. Back to the UCC for assistance. Section 2-601 states in part that "...unless otherwise agreed under the sections on contractual limitations of remedy..., if the goods or the tender of delivery fail in any respect to conform to the contract, the buyer may

(a) reject the whole; or

(b) accept the whole; or

(c) accept any commercial unit or units and reject the rest."

Assuming that you understood that the program was a current one, I believe that you have a right to reject it, and demand your money back. Notice I said "reject" not "return" because Section 2-602(2) states in part

"(b) if the buyer has before rejection taken physical possession of goods...he is under a duty after rejection to hold them with reasonable care at the seller's disposition for a time sufficient to permit the seller to remove them; but

(c) the buyer has no further obligations with regard to goods rightfully rejected."

Of course, that is not exactly the same as getting your money back. But Section 2-711(3) notes that

"On rightful rejection or justifiable revocation of acceptance a buyer has a security interest in goods in his possession or control for any payments made on their price and any expenses reasonably incurred in their inspection, receipt, transportation, care and custody and may hold such goods and resell them in like manner as an aggrieved seller."

More about a "security interest" in later columns; for now just consider it as being a legal right to use the goods as collateral. Of course, your law suit for damages is not precluded by taking advantage of the quoted paragraphs. You can combine various remedies, until you are "made whole" by the seller. Just one warning about rejection of goods. Section 2-602(2)(a) notes that "after rejection any exercise of ownership by the buyer with respect to any commercial unit is wrongful as against the seller;..."

In other words, you can keep the program until the seller returns your money (maybe including postage and handling), but you shouldn't use the program in the meantime. Fair's fair. I should also stress that the UCC also contains paragraphs giving rights to sellers when buyers wrongfully reject goods that should be acceptable. If any serious amount of money is involved, see your lawyer before you try anything drastic. But if your disk drives and computer work with other programs, and you have the configuration called for in the program's advertising, don't hesitate to notify the seller that you want your money back (including postage and handling) and you are "rejecting his non-conforming shipment pursuant to Section 2-602 of the Uniform Commercial Code."

Before finishing this column, let's discuss two more paragraphs of Article 2...Section 2-607 regarding notice to seller, and 2-714, which could be subtitled "What to do if you've already accepted the program before you find out it doesn't work!" Let's say, for example, that you purchase a program, and use it for two weeks, before you discover (i) that the program destroys data files at random; (ii) the seller 'recalled' the program the day after you bought it and replaced it with a new version without this bug; and (iii) the seller wants you to pay for an update. Or better yet, let's assume that you purchased the bugged program after the recall, because you didn't realize that

Barry D. Bayer, 2842 Walnut Rd., Homewood, IL 60430.

the absence of a white dot on the shrink wrap surrounding the product signified the bugged version. Section 2-607(3) states that

"Where a tender has been accepted the buyer must within a reasonable time after he discovers or should have discovered any breach notify the seller of breach or be barred from any remedy:"
Section 2-714 states

"(1) Where the buyer has accepted goods and given notification...he may recover as damages for any non-conformity of tender the loss resulting in the ordinary course of events from the seller's breach as determined in any manner as is reasonable.

"(2) The measure of damages for breach of warranty is the difference at the time and place of acceptance between the value of the goods accepted and the value they would have had if they had been as warranted unless special circumstances show proximate damages of a different amount.

"(3) In a proper case any incidental and consequential damages...may also be recovered."

This should be pretty clear, by now: notify the seller as soon as you know there is a problem. Your damages are the difference between what you got, and what you were promised, and maybe other costs or liabilities incurred because of the breach.

As a finish to this column, you might ask yourself the following questions:

1. What is the value of a program that destroys data files?

2. Can a program that destroys data files on a random basis ever possibly be merchantable, or fit for any particular purpose? (Or any unparticular purpose?)

3. How in the world can a manufacturer have the nerve to charge for an update to cure a serious product defect in an already recalled product? ☐

"... we're trying to communicate with people who write legal documents..."

235

ns...puzzles & problems...pu

Before we start with this month's batch of puzzles Merlin would like to thank all of the readers of this column who have written to us. Some have sent in new puzzles to share with us. Others point out errors that sometimes get by your editor (Merlin says that I should be made to wear a hairshirt when this happens). Still others like to send in alternate solutions to problems that can be solved in more than one way. In the January 1981 issue of *Creative Computing* we presented a change-the-word puzzle (changing the word APE to MAN). Our solution took six steps to effect the change. The letters poured in from our readers. They had come up with a better solution using only five steps. Our thanks go out to the following problem solvers: Nick Straw, Altadena, CA; Cheryl G. Rand, Woburn, MA; Norman L. Skeels, El Segundo, CA; Nancy Mingus, Wellsville, NY; Warren H. Buell, Los Angeles, CA; Jim Peisker, Chicago, IL; Bill Weisman, Venice, CA; Susan Heller, Tucson, AZ; David E. Welsh, Los Angeles, CA; Greg Robbins, Piedmont, CA; Jeffrey T. Elliot, New York, NY; Sanford P. Schumacher, Pittsburgh, PA; Dick Hall, Lansing, MI; Nancy H. Lamontagne, Andover, MA; Art Levin, Wilton, CT; C.F. Carson, Cincinnati, OH; Robin Ault, Newtonville, MA; Kenny Bergen, Houston, TX; Tony Hyman, Bath, NY; Richard Forsen, New Hartford, NY; Ted Puer, Bartlesville, OK; Algot T. Runeman, Wellesley, MA; Virginia L. Thorndike, Orford, NH. The answer that they all sent in was; APE, APT, OPT, OAT, MAT, MAN.

To Be Or Knot To Be, That Is The Puzzle

Our first puzzle is from Bill Weisman of Venice, CA. Bill's problem is an old favorite of Merlin's, one that I am sure you will want to stump your friends with. Bill writes: "You are given a two-foot long piece of string. You must pick up one end of the string in each hand, and tie a knot in the string without releasing the ends (no slip knots allowed)." I hope that our readers will try this before looking up the answer in the back of the magazine. It can be done. All you need is a little imagination. Thank you, Mr. Weisman. A copy of *Merlin's Puzzler #1* is on the way.

The "Bo" Puzzle

From Bryan Brewer, of Ashford, WA, comes an interesting "change-the-word" problem which I have called "The 'Bo' Puzzle" for obvious reasons. The puzzler has to change the word ONE into the word TEN in ten moves. (During each move you must change one letter in the previous word so that a new word is formed. For example: You can change the word WARM into the word COLD using the following four moves—WARM, WARD, WORD, CORD, COLD). Merlin says that changing a ONE into a TEN is quite a trick and he doubts if even he could pull that one off. I am sure that our readers will come up with solutions that require less than ten moves. I look forward to seeing them. For Mr. Brewer Merlin has a copy of *Merlin's Puzzler #2*.

A Barrel Of Fun Revisited

Back in the January 1981 issue of *Creative Computing* we presented a puzzle called "A Barrel Of Fun." This is one of those puzzles that your editor slipped up on. I forgot to mention how many sons were to split the estate. Mr. Clayton F. Schmidt, of Clinton, IO, has written us to say that he has a better solution to this intoxicating problem. Let's hear what Mr. Schmidt has to say: "Concerning your puzzle 'A Barrel Of Fun' there is a solution that is simpler than the one given. The problem was as follows:

A wine merchant had three sons and twenty-one barrels of wine. The barrels consisted of seven full barrels, seven half-full barrels, and seven empty barrels. The barrels were to be divided between the three sons equally.

The answer given was as follows:

Pour four of the half-barrels together, giving nine full barrels, three half-barrels, and nine empty barrels, which can easily be divided between the three sons.

The above answer takes four pourings. The answer I have in mind takes only one pouring. Can you figure out how it is done?"

Mr. Schmidt has given us a pretty problem here to solve. The funny part of his answer is that it is the solution to another famous old puzzle. That's a hint. Merlin just shouted over, "Give Clayton a copy of #3." Congratulations, Mr. Schmidt, a copy of *Merlin's Puzzler #3* is in the mail.

Chug-A-Mug

The following academic problem is a tough one. Take a long piece of string and tie it to the handle of a beer mug. Next, tie the other end to some overhead support. The mug should be about two feet above the ground. Now, your problem is to take a pair of scissors and cut the string in half at about two feet above the mug in such a manner that the mug will not fall to the floor. Only the scissors may touch the string. Nothing can be touching the mug and you cannot have anything between the floor and the mug. This is strictly a pass-fail test.

(This puzzle comes from the book *Merlin's Puzzler #3*, the third collection of fascinating, imaginative and thoroughly engrossing puzzles, quizzes, games and magic tricks in the Merlin series. This book is gloriously illustrated and guaranteed to give endless hours of enjoyment. *Merlin's Puzzler #3* has 128 large size pages, 8½" x 11" with hundreds of illustrations. You can order this book by writing to Creative Computing.)

Doodle-Words

Here's another quiz from one of our readers, Mr. Henry J. Dunn of Florissant, MO. Below are five squares containing five Doodle-Words. Can you figure out what word or phrase is depicted by the letters in each square? Mr. Dunn's prize for sending in this quiz is a copy of *Merlin's Puzzler #1*. I hope that our readers will make up their own Doodle-Words and send them along to Merlin.

Tide	iiiiii ooo	ecnalg	st and I	O M D PHD

Riddle Me This!

(1) How long did Cain hate his brother? (2) What's best when it's cracked? (3) How are a clock and a river alike? (4) What kind of iron is in the blood of a prize fighter? (5) What key is the hardest to turn? (6) Arrange six 6's so that they total 35? (7) How many times can you subtract 5 from 135? (8) How many cubic feet of dirt are there in a hole 3 feet wide, 3 feet long and 2½ feet deep? (These riddles are from *Merlin's Puzzler #2*).

The Even Steven Puzzle

Last, but not least, is a puzzle I think that you will like. Pictured at the right are sixteen objects in sixteen squares. Your problem is to remove six of these objects so that the remaining ten will be distributed evenly in the columns and rows. That is, there will be an even number of objects in each horizontal and vertical row. Also, there must be an even number in the two corner diagonal rows. There are many answers to this puzzle. Merlin has given you one in the answer section.
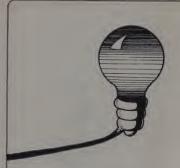
Answers on page 256.

That's all folks! Remember, keep the puzzles coming in. If Merlin uses your puzzle he will send you one of his great puzzle and game books. Bye, bye!

Your editor,

Charles Barry Townsend

Charles Barry Townsend

ucts...new products...new

COMPUTERS

Apple III Features Expanded Mass Storage, New Software

A more powerful version of the Apple III personal computer, with expanded mass-storage capability and professional application programs, has been announced by Apple Computer Inc.

The expanded storage capability is provided by the Apple III/ProFile Personal Mass-Storage System. Designed to be integrated into Apple III systems, ProFile is a five-million-byte, Winchester-type, hard-disk system which gives users nearly 35 times the mass-storage capacity of a single floppy disk.

Seven new or enhanced application programs have also been announced. One such program, Access III, allows Apple III computers to communicate with large mainframe computers.

Apple Writer III ($225) for word processing; VisiCalc III ($250) and Business Graphics III ($175) for preparing and analyzing detailed business information; Business Basic ($125) and Pascal III ($250) for developing advanced application programs, and Script III ($125) for formatting Pascal text into layout-perfect printouts are also available. In addition, an updated version of Mail List Manager ($150) will be available for maintaining mailing lists and printing labels using an Apple III/ProFile system.

The new Apple III with 128K bytes of memory and Sophisticated Operating System software is priced at $3,495. The complete system with Business Basic, VisiCalc, SOS and Monitor III is $4,190.

The minimum Apple III configuration needed to operate the ProFile mass-storage system includes 128K bytes of random-access memory, the SOS 1.1 operating system and utilities. $6,994. Apple Computer Inc., 10260 Bandley Dr., Cupertino, CA 95014.

TERMINALS & I/O

Serial I/O Board for Apple

The PSIO, Programmable Serial I/O board, is fully compatible with current Apple software including Pascal 1.1, Applesoft, and Integer Basic. It provides a full EIA RS-232 DTE interface to most modems, printers and CRT terminals.

On board 1K ROM software enables user programmable printer width control, delay after carriage return, automatic line-feed generation, and video display. Serial character format is user alterable with the most popular configurations set as default conditions. $199.95.

Intra Computer, 120-10 Audley St., Kew Gardens, NY 11415. (212)947-5533.

DISK SYSTEMS

Hard Disk Units for Commodore

Two new hard disk units enable users of Commodore Business Machines microcomputers to increase the storage capacity of their systems.

Designated the 9065D and 9150D, the new units are based on 5 1/4" Winchester technology. The 9065D offers 6.4 megabyte (unformatted) characters of storage, while the 9150D is capable of storing 9.6 megabytes (unformatted). Both units contain a built-in IEEE-488 interface for attachment to any Commodore PET, CBM or Super-PET computer.

The hard disks are DOS compatible with Commodore's current floppy disk drives.

Commodore Business Machines, Inc., 681 Moore Rd., King of Prussia, PA 19406. (215)337-7100.

CIRCLE 332 ON READER SERVICE CARD

SOFTWARE

EDUCATIONAL SOFTWARE

Administrative

Bursar is an accounting system for student activity funds. It keeps track of funds generated by extra-curricular activities, reports each day's transactions, generates financial statements for each activity, and provides proof of reconciliation of bank accounts. **Equip** is a capital inventory program which enables administrators to allocate, locate and account for all the physical assets of a school system. **Purchase**

"If the cost of education continues to rise, education will become as expensive as ignorance."

Iowa State Travelers Mutual Assurance Co. Newsletter

---

prints requisition forms and formal "invitations to bid" for bulk purchases. It also prepares purchase orders and delivery schedules. All three programs run on the TRS-80 Models I and III, and the Apple II. Addison-Wesley Publishing Company, Inc., Reading, MA 01867.

CIRCLE 333 ON READER SERVICE CARD

CAI-Manager is an educational management system written for use with a Corvus 5, 10 or 20 megabyte Constellation or Omninet system. It can incorporate most commercial and custom CAI programs for the Apple, and will store records for up to 450 students or 40 classes on one volume. Mathware/Math City, 4040 Palos Verdes Dr. North, Rolling Hills Estates, CA 90274. (213)541-3377.

CIRCLE 334 ON READER SERVICE CARD

Classroom

COPYRIGHT 1981 BY SYNERGISTIC SOFTWARE

The **Planetary Guide** displays all major solar system members on hi-res color screens. The user selects a date, and planet locations in the various constellations are displayed on the screen as they would be seen in the night sky. The program requires a 48K Apple, Applesoft, disk drive and DOS 3.3. $30. Synergistic Software, 5221 120th Ave. SE, Bellevue, WA 98006. (206)226-3216.

CIRCLE 335 ON READER SERVICE CARD

Hi-Res MasterType builds touch-typing drills into a space war game format. Missile attacks are launched by enemy words which the player must type to stop the attack. A Make Lesson option enables advanced players to develop lessons to meet their individual needs. Requires a 48K Apple with disk. $39.95. Lightning Software, P.O. Box 11725, Palo Alto, CA 94306. (415)856-1855.

CIRCLE 336 ON READER SERVICE CARD

New Products, continued...

Versa Computing has released **Meet the Presidents** which features hi-res graphics display routines which gradually reveal the portrait of one of the forty presidents. Clues are given, and the player must type in the correct president's name. The program comes on two double-sided disks and requires a 48K Apple and DOS 3.3. $39.95. Versa Computing, Inc., 3541 Old Conejo Rd., Suite 104, Newbury Park, CA 91320. (805)498-1956.
CIRCLE 337 ON READER SERVICE CARD

Krell Software announces **Isaac Newton**, an educational game which challenges players to assemble evidence and discern the underlying laws of nature which produce the evidence. It is available for 16K Apple, PET, TRS-80 and Atari computers on tape or disk. $24.95. **Micro-Deutsch** includes 24 grammar lessons and four test units covering the material of an introductory German course. It is available on tape or disk for 32K Apple and PET/CBM. $179. **Electoral College** illustrates the workings of the Electoral College System in the U.S. on 16K Apple, PET and TRS-80 computers. $24.95.

The Krell **Competency Exam Preparation Series** consists of simulated examination modules, a diagnostic package and a set of instruction programs. Designed for 16K Apple, PET and TRS-80 computers, the C.E.P.S. is available in three formats: the National Proficiency series, the California Proficiency Assessment Test preparation series and the N.Y. State Regents Competency Test preparation series. $2499. Separate math and verbal packages are available for $1499 each.

Krell also offers a preparation program for the **Graduate Record Examinations**. The GRE set includes 28 programs which run on 16K Apple, TRS-80 and Atari computers with tape or disk. $289.95. The **College Board SAT 81/82** Preparation Series includes 25 16K programs covering both verbal and mathematical topics. It is available for Apple, TRS-80, PET. Ohio Scientific, Northstar, PDP-11. Heath/Zenith, and Atari computers. $229.95. Krell Software Corp., 21 Millbrook Dr., Stony Brook, NY 11790. (516)751-5139.
CIRCLE 338 ON READER SERVICE CARD

Do-It-Yourself Spelling for the Atari allows teachers, parents and other users to create their own spelling programs by entering a series of 10-word lists. The program comes with a list of 1950 words that every child should be able to spell by the time he is in sixth grade. $19.95. Program Design, Inc., 11 Idar Ct., Greenwich, CT 06830. (203)661-8799.
CIRCLE 339 ON READER SERVICE CARD

Vernier Software announces six programs designed for use in introductory physics classes: **Orbit**, which simulates the motion of a satellite in the Earth's gravitational field; **Charged Particles**, which simulates the motion of a charged particle moving in a magnetic field; **Projectiles**, which allows students to experiment with projectile motion; **Wave Addition**, which demonstrates graphically the superposition of waves; **Vector Addition**, which demonstrates the head to tail addition of vectors; and **Kinematics**, in which students control the starting speed, acceleration and elapsed time for the motion of a truck moving across the screen. All programs are available in 20K cassette version or 32K DOS 3.3 version. $18 each. Vernier Software, 2920 SW 89th, Portland, OR 97225. (503)297-5317.
CIRCLE 340 ON READER SERVICE CARD

Counting Bee introduces children aged three to six years to counting, addition, subtraction, shape discrimination, number and measurement. It requires a 48K Apple with Applesoft and disk drive. $29.95. Edu-Ware Services, Inc., 22222 Sherman Way, Suite 203, Canoga Park, CA 91303. (213)346-6783.
CIRCLE 341 ON READER SERVICE CARD

Time Dungeon — World is an educational fantasy game package which places the player in a dungeon through which he advances by answering questions about the era of history he has chosen. The games are Ancient History 4000BC-4BC, World History WW I 1914-1919, and World History WW II 1933-1945. **Time Dungeon — World History** is a similar five-game package. The programs are offered on tape for the TRS-80 Models I and III ($24.95) and on disk for the Apple II ($29.95). Advanced Operating Systems, 450 St. John Rd., Michigan City, IN 46360. (219)879-4693.
CIRCLE 342 ON READER SERVICE CARD

The Game Show is a multi-player educational program featuring animated color graphics. Topics covered include vocabulary, history, algebra 1, sailing and computer terms. An authoring system allows the user to add topics. $39. (We think it's for the Apple—Ed.) Computer-Advanced Ideas, Inc., 1442A Walnut St., Suite 341, Berkeley, CA 94709.
CIRCLE 343 ON READER SERVICE CARD

Word Structure is an eight-disk series designed to help middle-grade and junior high students to communicate more effectively by providing individualized instruction in basic word analysis. Areas covered are capitalization/abbreviation, spelling/syllables, singulars/plurals, possesives, positives/comparatives/superlatives, prefixes, and suffixes. The series requires a 48K Apple II Plus with disk drive. $600. Borg-Warner Educational Systems, 600 West University Dr., Arlington Heights, IL 60004.
CIRCLE 344 ON READER SERVICE CARD

New Products, continued...

The Radio Shack **High Motivation Reading Series** provides four student readers, a read-along audio cassette, a teacher's manual and a computer program (written in TRS-80 Pilot Plus) to measure comprehension in several skill areas. Titles in the series include *Charles Lindbergh/Amelia Earhart* ($64.95), *Hound of the Baskervilles, Dracula, Moby Dick, The Beatles/John Lennon* and *20,000 Leagues Under the Sea* ($59.95 each). A 32K TRS-80 Model I with lower case option and one disk drive, and TRS-80 Pilot ($79.95) are required. Available at Radio Shack Computer Centers, stores and participating dealers.

The IBM Personal Computer Education Program announces **Fact Track**, a program to help students learn basic arithmetic skills; **Arithmetic Games** Set 1 and Set 2, designed to develop and refine mathematic and logic skills; and **Typing Tutor**, a touch-typing instruction and drill program. Available at authorized IBM Personal Computer dealers.

Capitalization is a two-disk system for practice and testing of the application of the major rules of capitalization. A management system gives immediate feedback and stores each student's records. The package requires a 48K Apple with Applesoft and one disk drive. $49.95. Hartley Courseware, Inc., P.O. Box 431, Dimondale, Michigan 48821. (616)942-8987.
CIRCLE 345 ON READER SERVICE CARD

Orange Cherry Media announces a line of software in the areas of language arts, reading, communication skills, science and mathematics for PET, TRS-80 and Apple computers. Orange Cherry Media, 7 Delano Dr., Bedford Hills, NY 10507. (914)666-8434.

LANGUAGES AND AUTHORING SYSTEMS

Independent Video Consultants, Inc. announces the IVCI Authoring System **PROMPT**, a computer program which combines microcomputers and videodiscs into an interactive training medium. It uses a 48K Apple with two disk drives and is available for the DVA and Sony videodisc players. Independent Videodisc Consulants, Inc., 125 West Durham St., Philadelphia, PA 19119. (215)248-0700.
CIRCLE 346 ON READER SERVICE CARD

The **TRS-80 Color Computer Learning Lab**, a self-instruction system designed to teach users of the computer to program in Color Basic, includes eight program cassettes and a 30-lesson text. It can be used with all versions of the Color Computer, and requires a standard cassette recorder. $49.95. Available at Radio Shack stores and participating dealers.

Pilot Plus is a CAI authoring language which allows developers to design and implement courses that will run on a variety of systems without modification. It supports videotape, videodisc, stored digital voice, touch panel or light pen input, and color graphics. It is currently available for TRS-80 Model II, Apple II with Z80 card, HP 125, Xerox 820, Northstar, Altos and Cromemco computers. $150. International Institute of Applied Technology, Inc., 20010 Century Blvd., Germantown, MD 20767. (301)428-9010.
CIRCLE 347 ON READER SERVICE CARD

Texas Instruments announces the **TI Video Controller** and **Course Designer** authoring software package for the TI 99/4A computer. The Video Controller ($699.95) links the computer to selected video tape players or to the Pioneer VP-1000 videodisc player. The authoring software ($199.95) guides the user through the course development with instructions presented on the video screen. Texas Instruments Incorporated, P.O. Box 53, Lubbock, TX 79408.
CIRCLE 348 ON READER SERVICE CARD

LinkSampler I for the Apple II and III is designed to make learning the Pascal language easier and faster. It includes 21 teaching programs that combine business routines with mathematics and a few games. $60. Link Systems, 1655 26th St., Santa Monica, CA 90404.
CIRCLE 349 ON READER SERVICE CARD

Terrapin, Inc. and Krell Software Corp. have been licensed to market the **MIT Logo** language for the Apple computer. Logo is a procedure-oriented language which has numerous educational applications at all levels. Terrapin, Inc., 678 Massachusetts Ave. #205, Cambridge, MA 02139. (617)492-8816.

Krell Software Corp., 21 Millbrook Dr., Stony Brook, NY 11790. (516)751-5139.
CIRCLE 350 ON READER SERVICE CARD

NEWSLETTERS, CATALOGS

Color Computer Newsletter

A newsletter for owners of the TRS-80 Color Computer has been announced by Lawrence E. Falk, publisher.

A typical monthly issue of *the Rainbow* will contain feature stories, hints and tips on operation, sample programs and reviews of Color Computer-associated hardware and software. $12 per year.

The Rainbow, 5803 Timber Ridge Dr., Prospect, KY 40059.

IBM Newsletter

IPCO, Inc. has recently been formed to represent owners and users of the IBM personal computer.

IPCO will publish a newsletter every other month which will contain articles of interest to the IBM personal computer owner/user. The newsletter will contain evaluations of hardware and software, articles with programming hints and information requested by readers.

IPCO also will have a software exchange for owner-developed software.

IPCO, Inc., P.O. Box 10426, Pittsburgh, PA 15234.

Weekly Newsletter for TRS-80

The *Eighty System Newsletter* is a weekly newsletter for users, retailers, distributors, and manufacturers of TRS-80 related products.

Mailed every Friday, the newsletter provides news on what's happening in the TRS-80 industry.

Also included each week is a bibliography of articles that relate to the TRS-80. $39 per year; sample copy, $3.

Kengore Corporation, 3001 Route 27, Franklin Park, NJ 08823. (201)297-2526.

Instructional Materials Catalog

The Society For Visual Education announces the *SVE 1982 K-8 Instructional Materials Catalog.* The catalog contains 49 programs, features filmstrips, book-cassettes, microcomputer software, study prints, and multi-media designed for pre-primary through 8th grade students.

The catalog is free to public, parochial, and private schools, libraries, churches, and other organizations that use audio-visual materials.

Society For Visual Education, Inc., Dept. #111-81, 1345 Diversey Pkwy., Chicago, IL 60614.

Software Catalog

Creative Discount Software has announced a free software applications guide and catalog of software products available for the TRS-80, Apple II, Apple III and TI 99/4 computers. It features applications for business, education and personal use.

The catalog may be obtained calling (800)824-7888, in California (800)852-7777, in Alaska or Hawaii (800) 824-7919. Ask for Operator 831.

Creative Discount Software, 256 S. Robertson Blvd., Suite 2156, Beverly Hills, CA 90211.

Courseware Catalog

The *J.L. Hammett Microcomputer Courseware Catalog* for elementary and secondary schools includes reviewed software in computer literacy, math, language arts, basic programming, and administrative needs.

All materials are available for 30-day preview with purchase order or prepayment.

J.L. Hammett Company, Hammett Pl., Braintree, MA 02184. (800)225-5467; in Massachusetts, (800)972-5056.

Logo Catalog and Newsletter

A catalog and the first edition of the *TI Logo Source,* a newsletter for TI-99/4A computer users, has been released by Microcomputers Corporation.

The free catalog and newsletter are available now.

Microcomputers Corporation, P.O. Box 191, Rye, NY 10580. (914)967-8370.

MISCELLANEOUS

Microcomputer Work Stations

Atlantic Cabinet has designed a range of work stations to house microcomputers. Manufactured from 1" all wood particleboard, surfaced with melamine veneer in either oak or walnut finish, the product is delivered in self-assembly form.

New Products, continued...

Overall height of all models is 36" and depth is 25". All have a 2" rear opening for wiring which can also be used as a paper in-feed. $185.

Atlantic Cabinet Corporation, P.O. Box 100, Williamsport, MD 21795. (301)223-8900.

Centralized Printer Sharing

Nestar Systems has announced the Network Print Server (Version 2.0). It allows users of Nestar's Cluster/One Model A local area network to take advantage of centralized printer sharing. Version 2.0 adds support for multiple printers, multiple networks, or multiple operating systems.

From their workstations, individual users may request printing of electronic mail messages, programs or data files on any of up to six printers attached to one or more Network Print Servers.

In addition to a printer, hardware requirements for the system include: an Apple II with 48K RAM; an Apple interface card and a 16K RAM system card.

The Network Print Server license fee is $995 per printer subsystem workstation.

Nestar Systems Incorporated, 2585 East Bayshore Rd., Palo Alto, CA 94303.

Computer-animated Teaching Film

The University of Toronto has released the computer-animated movie *Sorting Out Sorting* for the computer science education market. The film explains a variety of internal sorting techniques using computer animation. In fact, the entire movie, as well as its musical soundtrack, was generated by computer.

Sorting Out Sorting presents nine different sorting techniques. It is designed to be useful for all levels of computer science instruction.

The film is 30 minutes long, in color, with narration and music. It may be purchased or rented either on 16mm film or 1/2" or 3/4" videotape. It comes with a teacher's guide which supplements the material covered in the film.

The Distribution Office, University of Toronto Media Centre, 121 St. George St., Toronto, Ontario, Canada M5S, 1A1. (416)978-6565.

Electronic Furniture Systems

The electronic furniture system accommodates all makes and types of electronic office machines. The ergonomically designed system features crank adjusting terminal tables which adjust in height and tilt to reduce eye strain and fatigue of the operator.

Other components of the system include work stations, printer stands, tilting turntables, seating, mobile media storage, rack mount cabinets and other media storage accessories.

Smith System Manufacturing Co., P.O. Box 43515, St. Paul, MN 55164. (612)636-3560.

The Best Medicine

Your program is finally finished. You type RUN. Your computer replies READY. It has just eaten five hours of your life. What do you do?

Try to maintain your sense of humor – reach for *The Colossal Computer Cartoon Book*. Laugh at the original adventures of Edu-Man. Chuckle at cartoons by *Creative Computing* favorites Sandy Dean, Harbaugh, Swan, and Johns. Smile sympathetically at the ways others have suggested to get even with the infernal machines.

The Colossal Computer Cartoon Book contains hundreds of clever cartoons to tickle the funny bone of anyone who has ever loved or hated a computer.

To order one for yourself or a friend, send $4.95 plus $2.00 for postage and handling to Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950. Credit card buyers call toll-free. **(800)631-8112**, in New Jersey, (201)540-0445.

creative computing

Morris Plains, NJ 07950
Toll-free **800-631-8112**
(In NJ 201-540-0445)

Super Paddle

Are the paddle controllers on your Apple wearing out? Or did you get a new Apple without paddles?

We've got good news for you! Super Paddles. Each paddle control consists of a high-precision linear potentiometer and a big (1/2˝ D) industrial-quality pushbutton mounted in a sturdy 4˝ x 2˝ x 1˝ high-impact molded plastic case. Each of the two paddles is connected with a long 5-foot cable to the Apple paddle socket.

Every component in a set of Super Paddles is the very finest quality available. The set is backed by a 90-day limited warranty from the manufacturer as well as Peripherals Plus moneyback guarantee of satisfaction.

To order, send $39.95 plus $2.00 postage and handling (NJ residents add $2.00 sales tax) to the address below. Credit card customers may call orders to our toll-free number.

Peripherals Plus

| Toll-free 800-631-8112 | 39 E. Hanover Avenue |
| (in NJ 201-540-0445) | Morris Plains, NJ 07950 |

Get in sync

You and your friend are both new owners of a Sinclair ZX80 or ZX81. You bought your computers together and marveled over the low price you had to pay for the Sinclair's innovative design and sound performance. With visions of becoming computer experts, each of you read your manuals from cover to cover, discovering together the elementary functions, routines, and capabilities of your new systems. Within no time you were both running your computers with ease and confidence.

But soon your friend is expertly and efficiently adapting new peripherals and software to the machine—equipment that you didn't even know existed. Every day your fellow Sinclair user is showing you new and interesting programming techniques, creating interesting graphics, challenging you to new games, and converting programs from other computers to the Sinclair. Before you know it, you're friend has far surpassed you in computer aptitude and "know-how." How did your friend get so far ahead of you? The answer is simple.

Your friend reads **SYNC**—a *Creative Computing* publication written solely for Sinclair ZX80 and ZX81 users. **SYNC** takes you far beyond the limits of theory and practice as described in the manual. Every issue is packed with articles, tutorials, "how-to" material, reviews, and evaluations that enable you to use your Sinclair to its greatest potential. Hardware tips, puzzles, problems, letter-to-the-editor—all the information supplied in **SYNC** will keep you up to date on new ways to do things with your Sinclair that couldn't be done before.

SYNC functions on many levels with tutorials for beginners and concepts that will keep the pros coming back for more. We'll show you how to duplicate commands available in other Basics and we'll even show you how to do things on the Sinclair that can't be done on other machines.

While applications and software are the meat of **SYNC**, we recognize that along with useful, pragmatic applications, like financial analysis and graphing, you want games that are challenging and fun. We give you complete listings and sample runs of new games like Widget, Gauntlet, Forest Treasure, and Life as well as old favorites like Tic Tac Toe and Mastermind with a computer twist. All can be picked up from the pages of **SYNC**.

Order SYNC Today

To order your subscription to **SYNC** in the United States, send $10 for one year (6 issues), $18 for two years (12 issues), or $24 for three years (18 issues). Send order and payment to the address below or call our toll-free number to charge your order to your Mastercard, Visa, or American Express.

Subscriptions in the UK are mailed by air and cost £10 for one year. £18 for two years or £25 for three years. Send order and payment to the UK address below.

Canadian and other foreign surface subscriptions cost $15 for one year, $27 for two years, and $38 for three years and should be sent to the USA address.

We guarantee your satisfaction or we will refund your money for the unsent issues.

Subscribe to **SYNC** and get the most from your Sinclair.

sync

39 East Hanover Avenue
Morris Plains, NJ 07950, USA
Toll free **800-631-8112**
(In NJ 201-540-0445)

...reviews...book reviews

Steve Gray, et al.

Karel The Robot by Richard Pattis. John Wiley & Sons. New York. $5.95 1981.

This gem, by a computer scientist at Stanford, might otherwise be overlooked because of its title.

The book has a certain indebtedness to Seymour Papert's turtle. but it has somewhat different goals in mind. The reader is guided through a sequence of programming tasks, each one of which requires Karel to move about a rectangular grid to pick up and put down "beepers." The grid contains walls, and Karel has sensors to tell whether he is bumping against a wall.

Karel is given directions by means of (surprise!) a programming language, which contains primitives for moving, turning by 90 degrees. picking up and putting down. The language is a very close relative of Pascal, and that is the whole point. A beginner learning Pascal is faced with understanding two big items: control structures and data structures. Pattis has very cleverly noticed that these two topics are almost completely independent of one another. Since either one can be a stumbling block for a beginner, why not separate them and start off with just one of them?

Karel The Robot does exactly that. Pattis's programming language has all of the standard structured programming features found in most recent languages; but it has no data structures. In fact, it has no data at all: no variables, no assignment statement, no I/O conversion or formatting. What's left is not trivial. however. The budding programmer learns how to define separate procedures, how to construct branch blocks, how to make several kinds of loop blocks, how to use Boolean-valued functions, and the like. These are pretty high-falutin' ideas. but are presented in a friendly, accessible way.

More people ought to know about *Karel The Robot*. As things stand now. only a few university computer-science types seem to be aware of the book. One of the drawbacks to popular success until now has been the fact that the book cries out for a hands-on approach. The learner really needs to write those programming tasks and run them. The only way to do that is to have access to a big computer system that runs a Karel simulator. Places like Stanford. Berkeley. and Carnegie-Mellon. for example.

But that is about to change. Pattis has recently converted his simulator to run on micros. That means that many readers may have more than an academic interest in reading the book. The Apple Pascal version of the simulator is available from Cybertronics International, Inc.. Software Publishing Division, 999 Mount Kemble Ave.. Morristown, N.J. 07960. It comes with a nicely written 28-page booklet explaining how to set up and use the simulator on an Apple.

The simulator is especially nice about debugging aids. Programs can be performed at various speeds and levels of detail. They can be single-stepped and traced. A unique feature of the simulator is that it can "back up": program steps can be performed in reverse order. Combine that with single-step mode and you have a superb debugging tool.

Summing up. I think many educators will find *Karel The Robot* and its simulator to be both entertaining and truly instructive about some important programming ideas found in languages like Pascal.—*Arthur Luehrmann*

Learning With Computers by Alfred Bork. Digital Press, Bedford, MA. 286 pages. hardcover $28. 1981.

To those interested in computer assisted learning (CAL), Alfred Bork would be regarded as a true pioneer. Bork has made significant contributions to the field through materials and strategies developed for physics classes at University of California—Irvine's Educational Technology Center and his work for the National Science Foundation through Conduit (a software review organization) and by offering Chautauqua courses on computer learning. His opinions, evaluations, and projections for the future of CAL cannot be ignored. Thus his book merits serious attention

Bork has assembled his papers, some co-authored, over the past ten years, into seven chapters on learning with computers. The title reflects his view of the computer as a learning device or tool that educators can use to enhance learning. So here we learn "with" computers rather than "about" them: this later view is taken by the "computer science" frame of reference. The reader will learn how to use computers to manage classroom records, provide testing application, instruct, remediate, and motivate students. The text is not about the computer's architecture or acquiring a programming language.

Through past readings and future projections several themes will emerge. He allows the redundant occurrence of his ideas in ten years of writing to measure the importance of certain issues. The reader will realize, for example, current programming languages such as Basic, Fortran or Cobol as well as authoring languages like Pilot or Coursewriter are inadequate because they do not provide adequate graphic capabilities or flexible dialog interaction. Bork prefers the rapidly developing approaches in languages such as Pascal and Ada.

A philosophy of education and instruction is readily apparent. Materials should be prepared by teams of teachers, programmers, designers. Material preparation takes time, effort, money and an understanding of software hardware capabilities, as well as good review and evaluation. The approach should include a dialog interaction between student and program while using graphics, personalization, problem-solving, and immediate feedback to motivate students and enhance education. Review transportability, distribution and marketing considerations for computer material also are discussed in detail.

Overall the text provides a realistic view of what has been, is being done, and probably will be done with computers in education. Innovative and challenging plans for delivering education at a distance on an individual level with personal or stand-alone computers will invigorate any computer neophyte. Bork's ideas represent a state-of-the-art assessment of all basic philosophy and method for instructors using CAL technology.

Several things will bother you about this book. The price of $28 is steep for what amounts to a collection of reprinted articles. While the material is very readable, interesting, and exciting, the social scientist and/or educational material evaluator will find no data supporting claims about the efficacy of graphics, dialogs, or the computer learning environment in general. However, the lack of assessment and evaluation in CAL is typical of the lag between instruction and critical analysis, but points to the extremely critical need for task analyses of learning procedures and statistical tests of the value of CAL materials and procedures. Although I acknowledge and agree with Bork's projections for the future, such speculation must eventually be supported by proof.

The Physics course examples may be difficult to understand for the non-scientific reader and the redundant examples of various examples and demonstrations will bother the cautious reader. I had no additional reservations about the book.

Two years ago I was captured by Bork's innovative and enthusiastic approach to education at his NSF Chautauqua course. I hope that as many concerned educators as possible will enjoy a similar fate by reading *Learning with Computers*. The CAI field is ready for many critical, analytic and creative minds to develop a better and more effective educational environment. – *Timothy Jay*

The ADA Programming Language by I.C. Pyle. Prentice-Hall International, Englewood Cliffs, NJ. 303 pages, paperback $14.95. 1981.

This textbook, according to the back cover, "provides a comprehensive introduction to the new language, which has been developed by the United States Department of Defense. Particular stress is placed upon its use in the programming of real-time or embedded computer systems. This book is suitable for programmers and students who are familiar with such high-level languages as Fortran, Pascal and PL/1."

The author, who is a professor and chairman of the Computer Science Department of the University of York, England, "participated in the DoD High Order Language project since 1975, and in the design team for Ada since 1977."

A fairly good knowledge of programming is assumed, because the book jumps right into the language with a program that's bound to discourage personal-computer fans of Basic eleven lines to add two simple integers. The lines include the definition of A and B as integers ranging from 0 to 999. GFTting A and B, and three PUT lines:

        PUT ("The sum of ");
        PUT (A); PUT ("and"); PUT (B);
        PUT ("is"); PUT (A = B).

which make Fortran and even assembly language look rather simple.

The typewritten text is divided into 13 chapters, with up to half a dozen programming exercises at the end of each. The book has no index, but does include seven appendices, two of which are three-page notes for Fortran and Pascal programmers, on similarities and differences between Ada and those languages. Data types are the most fundamental difference between Ada and Fortran, while Ada "contains a subset which is very close to Pascal."

The last page is a copy of the 1852 memorial tablet to Augusta Ada, Countess of Lovelace, "who first programmed a computer," and whom this language is named after.

8080/Z80 Assembly Language: Techniques for Improved Programming by Alan R. Miller. John Wiley & Sons. New York. 327 pages, paperback $9.95. 1981.

Although, as the author says in his preface, "it might seem strange that another book on the 8080 and Z-80 should appear at this time," he has two purposes. First is "to provide a single reference source for both 8080 and Z-80 assembly language programmers." The two instruction sets are listed both alphabetically and numerically in four appendices, followed

by a cross-reference between the 8080 and Z-80 mnemonics.

The second purpose is to "demonstrate some useful techniques of assembly language programming."

Although the back cover says "Learn the details of assembly language programming—quickly and easily," this is by no means a book for the beginner–unless he's Mensa-bright. The reader without a fairly good knowledge of assembly language will be snowed under by page five of the first chapter, which is only the introduction and which looks into the two CP/M structures.

The first five chapters are on general principles. Chapter two covers number systems and logic; three gets into the stack; four looks at input output; and five is about macros.

The last five chapters give specific programming examples: Development of a System Monitor; Z-80 System monitor; Number-Base Conversion; Tape Routines; and Linking Programs to the CP-M Operating System.

The book is a straightforward, workmanlike presentation of the basics of assembly language and programming with it. It is semi-informal, reflecting perhaps the split personality of a professor of metallurgy who is also a contributing editor to *Interface Age* and who writes very well.

Buy this only if you know exactly what parity flags and loader programs are, have an IQ over 135, or enjoy doing things the hard way

The Computers Are Coming by Irv Brechner. Irv Brechner. Box 453, Livingston, NJ 07039. 94 pages, paperback $4.95, plus $.85 postage. 1981.

"It's a basic first book for people looking to get into computing and who have no prior experience." says the author in his letter "I'm very easy to understand language and a tongue-in-cheek style. the book shows how computers work as well as how they'll affect everyone's life."

Yes, the language is easy to understand But the tongue-in-cheek style is almost entirely limited to the chapter titles, which don't always tell you what the chapters are about, such as Nobody's Perfect, Doesn't Do Windows, and Keeping Up With The Kids.

As for telling you how computers work, the author describes ASCII code (which he says is pronounced "ass-cee") and RAM memory, says "It really isn't important to get any more technical." and then briefly discusses disk and tape.

There are no drawings or photographs in this book. nor any mention of computers (other than "a roomful of Apples. PETs or TRS-80s). Basic is discussed, and several brief programs are presented how thick is a pile of paper that starts with one sheet and then doubles that number 50 times; find a baseball player's average: statistics about schoolchild[?]n

Applications are mixed, so that right after describing the use of a personal computer for games, the author will describe law clerks using computers into "research into precedent-setting cases," and delivering "the daily paper right onto a home computer's video terminal."

Although this book does contain some information of interest to neophytes, most of the writing is rather boring, or consists of space-fillers such as a glossary, and is usually vague about details.

Many better books are available for beginners, better organized, with illustrations and with specific details on popular personal computers.

Computer Networks and Distributed Processing: Software, Techniques and Architecture by James Martin. Prentice-Hall Inc., Englewood Cliffs, NJ 07632. 574 pages, hardcover $34. 1981

For openers, this volume is unique among EDP books because not only do the the endpapers contain what amounts to advertising for another 13 of the author's books, the back cover (accompanied by a photo of Martin, hands raised like a hypnotist) plugs his seminars.

The eight quotes on the back cover are extremely laudatory especially the first, which calls Martin "The man generally acknowledged to be the computer industry's most widely read author, best attended lecturer, and foremost authority on the social and commercial impact of computers."

A quick glance through the book will show you that this is not the usual heavily technical, rather dull text you have to wade through. As a lecturer, Martin has learned to use a great many simplified diagrams, and to keep the language simple. Math is kept to a minimum.

The book is divided into five parts, all but the last containing six to twelve chapters. The Promise (trends, types); Computer Networks (functions, facilities, controls, interfaces); Network Mechanisms (switching, control, routing compression, etc.); Errors, Failures and Security; and Future. Most chapters end with references for further reading, many to other books by the author, most to works published after the Mid-seventies.

The writing is as simple as possible under the circumstances of a subject that is often quite complex. Similes help, such as "Packets in a mesh-structured network are rather like cars traveling across a city. At various points a decision must be made" which route should one take?" For a highly readable book that covers the subject very well, without requiring an engineering degree to read, this is probably the best.

Basic Discoveries: A Problem Solving Approach to Beginning Programming by Linda Malone and Jerry Johnson Creative Publications Inc. Box 10328, Palo Alto, CA 94303. 76 pages, paperback $7.50. 1981.

One of several books from Creative Publications that provide classroom computer materials, for grades 5 and above, this particular one "is designed to present computer programming...through an exploratory and problem solving approach... Each programming concept is presented through activities appropriate to many ability levels."

The general plan is to present a concept in a page or two, then provide one or more "activity" pages, which usually require the student to type programs into the computer and record the output. Several types of activities are used. Most are for "exploring" (type in a program and see what happens); others are for "predicting" (write what you think the program will make the computer do); "solving" (write a program to solve a particular problem); "review," "debug," etc.

The first eleven chapters discuss PRINT and GOTO, LET, TAB and REM, IF/THEN, debugging, INPUT, random numbers, writing games, counters, FOR/NEXT, and variables and string arrays. The remaining chapters provide a dozen problems to solve, a summary of Basic commands and functions, solutions to selected problems, and a short bibliography

The pages are all perforated so that they can be used as black-line masters; each student can have a set of activity pages.

As the chapter titles show, the progression is from simple PRINT statements up through the simpler statements, with only eleven being used altogether. No activity program is over 16 lines long.

This seems a fine book for classroom use, with a slow and sure approach, easily understood text, and many varied activities that help impress the concepts on student minds.

Basic: A Simplified Structured Approach by Dale E. Nelson, Darrell V. Burras, Emma Jean Gillis and Ron L. King. Reston Publishing Co., A Prentice-Hall Co. 343 pages. hardcover $16.95, paperback $12.95. Second edition, 1981.

Although the original copyright date is 1978, the preface says nothing about the earlier edition. It does say that "The approach in this text is student-oriented. By using this text, the student could learn the subject by himself because of the step-by-step presentation of material."

The text is timesharing-oriented, as well as being based on the cart-before-the-horse principle in the opening chapters. Chapter 2, on Basic Information Needed to Write A Simple Program, discusses over a dozen steps the student is supposed to keep in his head before writing a program, instead of introducing them during the writing of programs, a more natural way.

Chapter 3, on The Structured Flowchart and Additional Statement Types, presents all the flowchart symbols right up front, and half a page later shows six short sample flowcharts, all of which will baffle many students. This is hardly a step-by-step method, which one might assume would take the symbols one at a time and discuss the use of each, gradually working the other symbols into the picture.

The subsequent chapters don't fare much better. They do not have the "ample examples" the preface boasts of. Many pages are heavy with long paragraphs of forbiddingly bold-faced type.

If the authors think this textbook is student-oriented, they should look at the many fine ones that are. This one requires a teacher to help the student hack a path through the heavy-laden pages, which could have been much improved with more examples, short paragraphs, and a lighter typeface. Perhaps this is another case of too many authors spoiling the broth.

The Minicomputer in On-Line Systems by Martin Healey & David Hebditch. Winthrop Publishers, Inc., Cambridge, MA. 349 pages, hardcover $22.95. 1981.

One of the latest in the Winthrop Computer Systems Series, which consists of two dozen books on subjects ranging from Basic to computer models, this handbook "provides an overview of minicomputers and data communications — and how they work together," according to the dust jacket, which says it's aimed at "technical managers, designers, systems programmers, and application programmers in user companies, manufacturers, and software houses."

Although a basic knowledge of minicomputer hardware and software would help in understanding this text, not all that much is required, because the authors explain a great deal, going into many details, with a formal writing style well-suited to the technology, and with an occasional informal aside.

The ten chapters look at microcomputers in data communications and distributed processing systems, minicomputers technology and architecture, data transmission and terminals, transaction processing, data communications on minis, data communications software, operating systems, high-level packages, acquisition and implementation, and four case studies.

For the reader with enough knowledge of minis, or with enough interest in the subject to want to push through the technical underbrush, this is well worth the time spent delving into flags, ISAM, modems, task partition, and all the rest.

Structured Programming in Fortran by Louis A. Hill Jr. Prentice-Hall Inc., Englewood Cliffs, NJ, 541 pages, paperback $15.95, 1981.

This has got to be one of the most profusely illustrated programming books ever written, brimming over with listings, flowcharts, examples, charts, and tables, at least one per page throughout.

Perhaps this is a result of the book having been "class-tested by over 5,000 students from nearly every department on the Arizona State University campus," as the preface puts it. Whatever the reason, the many illustrations go a long way toward making this lengthy textbook as simple as possible. The back cover says these "hundreds of worked examples, problems, and program-writing exercises encourage reader participation immediately," that this is "an approach which commands active involvement." They do, and it really is.

The chapters cover the usual range of Fortran features you might expect, including Control of Flow, Input/Output, Do-Loops, Subroutines and Functions. Each chapter ends with exercises; the answers are all in Appendix B. The other appendix contains eight laboratory projects, all rather involved and calling for a great deal more than just writing a few program lines.

Although crammed with illustrations, this book can serve as a model to less careful publishers on how to integrate artwork with text, how to plan illustrations for maximum effect, and especially how to avoid the cluttered look so common to many books with far fewer illustrations.

As an example of the book's effectiveness, this is one of the very few Fortran texts I've felt drawn into, cleverly written with a most effective pace and a fluid style that, if only a little less formal and professional, would have me staying up nights to finish it.

I am often asked to recommend a non-technical book to someone who wants to learn how computers work and what they can do. *The Electronic Cottage* is the first such book I can recommend without reservation. It is a wide-ranging, intelligent and delightful book, which opens doors and leads the reader through the new world of computers. Suitable for the complete novice, this book takes the reader to quite a high level of computer literacy.

The first chapter welcomes the reader to the "age of logic." Two general points that motivate the entire book are stated here. The first is that the time has come for everyone to know

something about computers. The second is that how we will be using computers in the near future is limited only by our imaginations. The rest of the book leads the reader to a conceptual understanding of how computers work, and stimulates the imagination by describing many examples of current applications and future possibilities.

Chapters 2 and 3 introduce the reader to computer programs and languages, and to the logic underlying computer operations. While staying at a non-technical level, key concepts, such as algorithms, digital vs. analog, and random vs. serial access are clearly and accurately explained. The next seven chapters describe a wide range of current and potential applications. The reader's understanding of the principles underlying all computers is expanded as he learns how computers can perform a diverse array of processes. Throughout the book, excellent use is made of intriguing analogies, such as games of "binary chairs" and a computer made from buckets and hoses. Graphics designed by the author on a computer further clarify the material, as well as demonstrating one of the newer uses of computers.

One excellent chapter describes how computers can interact with people, sense information from the environment, and send out signals which control other machines. In another chapter, the author extends standard notions of computer games to hypergames and metagames, demonstrating his own fertile imagination. Two chapters focus on education, one dealing with using computers in training skills such as typing or playing the piano, the other dealing with potential applications of computers in classrooms. Other chapters cover computer applications in decision making, communications, information storage and retrieval, and modeling social, economic and physical phenomenon.

The final chapter provides some specifics about current hardware and software, and sources of further information. As the author acknowledges, this information cannot be kept up-to-date in any book — the world of computers is changing too quickly. The important thing is to learn how to ask the right questions about what you need for your intended application, and to be able to understand and evaluate the answers. This book is an ideal starting point for acquiring the requisite knowledge. —*Glenn Kleiman* □

"...and I wish you would stop referring to this cold as my 'down time'!"

CALIFORNIA

Advance Data Concepts — 2280 Diamond Blvd., Concord 94520; (415) 671-9018. 9-5 Mon.-Fri. Vector-Graphic, CP/M Software Headquarters-User's Group.

D.E.S. Data Equipment Supply — 8315 Firestone, Downey 90241. (213) 923-9361. 7 days. Commodore PET specialists. Hardware, Software, Books, Mags, Supplies, In House Maintenance.

CONNECTICUT

Computerworks — 1439 Post Rd., East Westport 06880; (203) 255-9096. 12-6 Tues.-Fri., 12-9 Thu., 10-5 Sat.

Connecticut Information Systems — 218 Huntington Road, Bridgeport 06608; (203) 579-0472. Mon.-Sat. 10am-6pm. Apple and CP/M-related products.

GEORGIA

Atlanta Computer Mart — 5091 Buford Hwy., Atlanta 30340; (404) 455-0647. 10-6 Mon.-Sat.

ILLINOIS

Computer Land/Downers Grove — 136 Ogden Ave., Downers Plaza 80515; (312) 984-7782. 10-6 Mon.-Sat., 10-8 Tue., Thurs. Apple, Atari, Osborne, Xerox, Vector.

Data Domain of Schaumburg — 1612 E. Algonquin Rd., Schaumburg 60195; (312) 397-8700. 12-9 Tues.-Fri., 11-5 Sat. Apple, Alpha Micro, Hewlett-Packard Calculators. Largest book and magazine selection.

Farnsworth Computer Center — 1891 N. Farnsworth Ave., Aurora 60505; (312) 851-3888. 10-8 Mon.-Fri., 10-5 Sat. Apple, Hewlett-Packard series 80 systems, HP Calculators, IDS Printers.

Gavin Computer — 5935 W. Addison St., Chicago 60634; (312) 286-4232. Mon.-Thurs. 9-8:30, Tues.-Sat. 9-8. Apple B & H. Atari & Commodore Systems.

Video Etc. — 465 Lake Cook Plaza, Deerfield 60015; (312) 498-9669; Open every day. Strong software support for Apple, Atari.

The Video Station — 872 So. Milwaukee Ave., Libertyville 60048; (312) 387-8800. Open 7 days. Atari Computers, Hardware and Software.

MASSACHUSETTS

Neeco — 679 Highland Ave., Needham 02194; (617) 449-1760. 9-5:30 Mon.-Fri. Commodore, Apple, Superbrain, Atari.

Science Fantasy Bookstore — 18 Eliot St., Harvard Sq., Cambridge 02138; (617) 547-5917. 11-8 Thur. Apple, Atari & TRS-80 games; Epyx, Microsoft, Creative Computing.

MICHIGAN

Computer Center — Garden City; (313) 425-2470 & West Bloomfield; (313) 855-4220; Books, Magazines, Hardware and Software for Apple, North Star, TRS-80 & PET.

NEVADA

Home Computers — 1775 E. Tropicana #8, Las Vegas 89109 (702) 798-1022. 10-7 Mon.-Sat. Apple, Commodore, Atari, AIM 65, (Books) Sales & Service.

NEW JERSEY

Computernook — Rt. 46, Pine Brook Plaza, Pine Brook 07058; (201) 575-9488. 10-6:30 MTWS, 10-8 Thurs., Fri. Apple/Commodore Authorized dealer.

The Computer Universe — 155 Route 17S., Paramus 07652; (201) 262-0960 — 347-9008. Mon; Wed; Fri.; and Sat., 10—6. Tues., and Thurs; 12—9. Specializing in Apple Computers.

Silent Partner — 2050 Center Ave., Fort Lee 07024; (201) 947-9400; Mon.-Sat. 10-6. Apple/Atari/Commodore/Vector/Malibu.

Software Mart — 352 Bloomfield Ave., Caldwell 07006; (201) 228-4949. Software for Apple, Atari, TRS-80 and PET always 10-20% off list.

Software City, Pine Brook — 101 Rt. 46 East, 07058; (201) 575-4574. Bus/Rec Utility/ Home programs for all personal computers at 5-20% off list.

Software City, River Edge — 111 Grand Ave., 07881; (201) 342-8788. Bus/Rec/Utility Home programs for all personal computers at 5-20% off list.

NEW YORK

Software City, Armonk — 148 Bedford Rd. 10504. (914) 273-3877. Bus/Rec/Utility Home programs for all personal computers at 5-20% off list.

Programs Unlimited — 20 Jericho Turnpike, Jericho, 11753; (518) 333-2266. 10-8 Mon.-Sat. The largest microcomputer software selection available.

The Computer Center — 31 East 31st St., New York 10016; (212) 889-8130. 10-7 Mon.-Fri., 11-6 Sat., 10-8 Thur.

Upstate Computer Shop — RD 1, Box 17A, Whitesboro 13492; (315)788-8151. 10-6 Mon.-Fri., 11-5 Sat. Apple — Commodore — Data General.

OHIO

Abacus II — 1417 Bernath Pkwy., Toledo 43615; (419) 665-1099. 10-6, 10-7 Thurs. Apple, Osborne, Adds, NEC, Atari, Epson & IDS Printers.

Barnhart Stores — Urbana; (513)653-7257. 8am-5pm. Atari Computers at MAIL ORDER prices/Authorized Atari Service Center.

North Coast Computers — 626 Dover Center, Bay Village 44140; (216) 835-4345. 10-6 Mon.-Sat., 10-8 Tue., Thur. Apple/Atari/ Vector Graphic/Data General.

To include your store in Creative Computing's Retail Roster, call the Advertising Department at (201) 540-0446.

ISSUES

Order any five or more for $1.75 each or ten or more for $1.50 each. U.S. Postage is $2.00 for up to 4 issues or $3.00 for five or more Foreign postage is $1.00 per issue

Super Special: One of everything we have including five rare issues of *ROM* — 36 magazines in all — for only $50 postpaid ($75 foreign)

Send payment or Visa, MasterCard or American Express card number and expiration date to Creative Computing, 39 E. Hanover Ave., Morris Plains, NJ 07950. Or call toll-free **800-631-8112** (In NJ 201-540-0445).

creative computing

the #1 magazine of computer appli...

Programs for Apple, Atari, TRS-80, PET and SOL

Chess in Basic

In-depth Evaluations:
- IBM Personal Computer
- VisiPlot and VisiTrend
- Childrens Computers
- TI Logo Language
- Diet Programs
- Odyssey
- Video Game
- Crush, Crumble
- Chomp

• index to advertisers •

puzzle answers

To Be Or Knot To Be: The solution is to cross your arms (right hand over left upper arm, and left hand below right upper arm) before picking up the string. Uncrossing your arms then automatically ties the knot.

The "Bo" Puzzle: ONE, ORE, ARE, ALE, ALL, AIL, AIR, SIR, SIN, TIN, TEN.

Barrel of Fun Revisited: Give each son two full barrels, two half barrels, and two empty barrels. This leaves three barrels, one full, one half-full and one empty. Pour out of the full barrel into the empty barrel until the level of the wine at the lip is even with the top of the bottom of the barrel. This will have the effect of leaving half a barrel in the one that was full and placing half a barrel in the one that was empty. Now all that is left is three half-barrels, of which each son gets one.

Chug-A-Mug: Before you cut the string you must first take a loop of string about two feet above the mug and tie a knot in it. You can now cut the string (the part that makes up the loop) and the knot will keep the mug from falling to the floor.

Doodle-Words: (1) Soap box (2) Circles under the eyes. (3) Backward glance (4) I understand (5) zero degrees

Riddle Me This: (1) As long as he was Abel. (2) A joke. (3) They don't go without winding. (4) Scrap iron. (5) A donkey. (6) $6 \times 6 + 66/66 = 37$ (7) One time. After the first time the number is not 135 anymore. (8) None, the hole is empty.

The Even Steven Puzzle:

REMEMBER.

ELEPHANT

MEMORY SYSTEMS

"NEVER FORGETS."

Elephant™ floppies.
They're guaranteed to meet or beat every industry standard for quality. They come standard with reinforced hub rings at no extra cost. They come in every popular 5¼" model, in both hard and soft sector. And they sell at some of the lowest prices in the business.
Elephant Flexible Disks.
They're heavy duty. They work for peanuts. They never forget. Get yourself a trunkful.

HEAVY DUTY DISKS.

Distributed Exclusively by **Leading Edge Products, Inc.**, 225 Turnpike Street, Canton, Massachusetts 02021
Call: toll-free 1-800-343-6833; or in Massachusetts call collect (617) 828-8150. Telex 951-624.