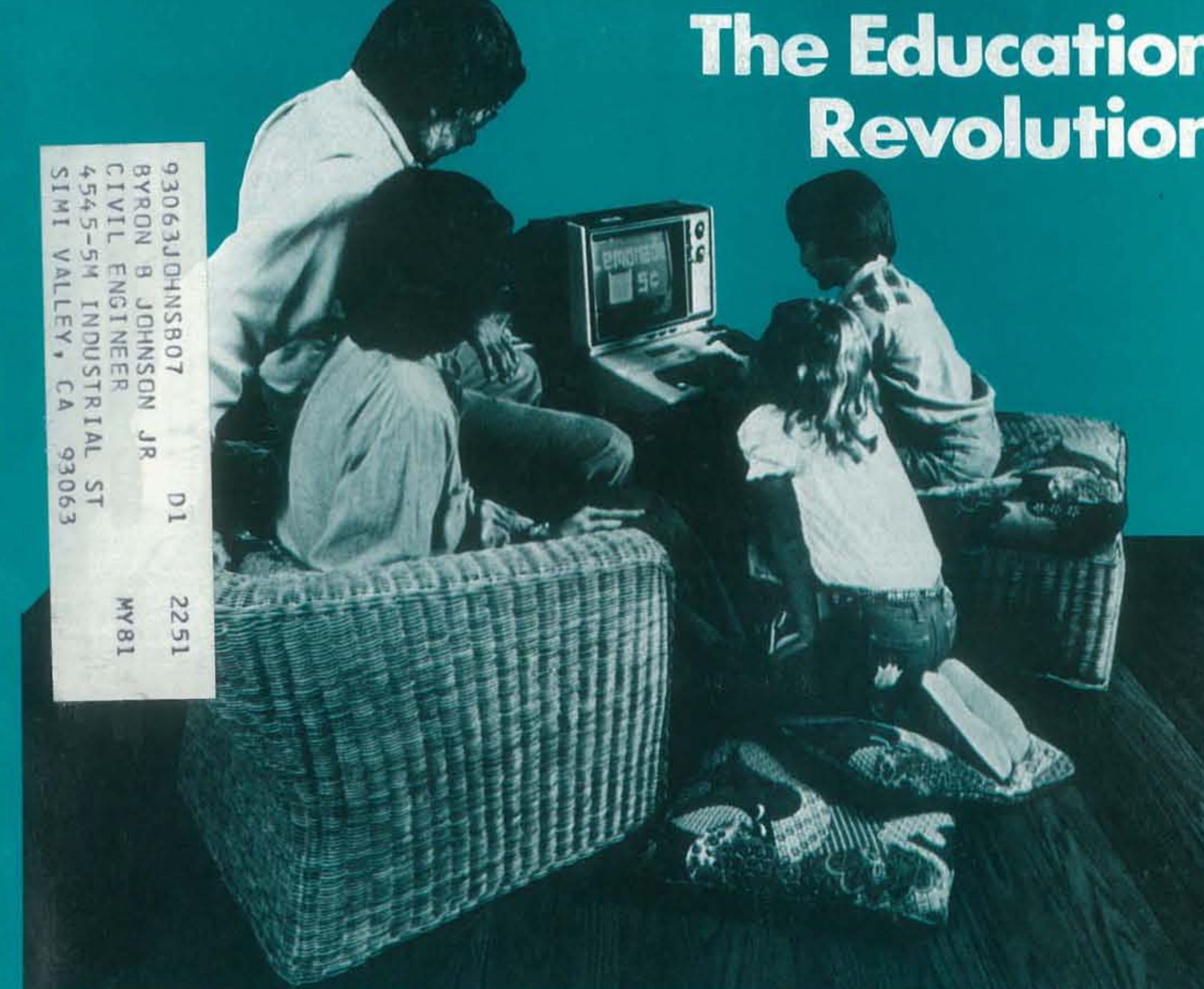
Recreational COMPUTING

FOR THE IMAGINATIVE SMALL COMPUTER USER!

VOL. 9 NO. 4 ISSUE 49 JAN-FEB 1981 $2.50 In Canada $3.00

The Education Revolution



Computer Games in the Classroom

An Art-Producing Turtle

Computer Literacy Resources

Musical Compositions Using Computers

Recreational COMPUTING

January/February 1981
Volume 9, Number 4
Issue 49

Editor
Joan Hiraki
Contributing Editors
Dave Caulkins
M. Dundee Maples
Ramon Zamora
Publisher
Ann Merchberger
Art Director
Clifford West
Artists
Jeannie Barroga
Greg Gordon
Barbara Ruzgerian
Typesetter
Renny Wiggins
Proofreader
Nancy Heubach
Marketing Director
Gavin W. Cullen
Advertising Director
Michael Madaj
Circulation Manager
Peter Clark
Circulation Assistant
Leah Dansby

Subscription rates: $12 per year within the United States; $20 for first class to Canada and Mexico; $26 for airmail to other countries. Payment must be in U.S. dollars, drawn on a U.S. bank.

Contributing Subscribers: $25/year ($13 tax deductible) – Algorithmics, Inc., DeWitt S. Brown, Gerald Bowman, Robert Connors, David R. Dick, Mark Elgin, Joi Ellis, John B. Fried, Scott B. Guthery, Alan Hamilton, Brian Herring, T. Alton Howard, William G. Hutchison, Jr., W. A. Kelley, Land of Light, William M. Richman II, Phillip A. Smith, Neil Sullivan, Joseph A. Weisbecker, Brett Wilson. **Retaining Subscriber:** $50/year ($38 tax deductible) – Dave Caulkins. **Sustaining Subscribers:** $100+/year ($88+ tax deductible) – Byte Publications; Paul, Lori & Tom Calhoun; Louis R. Patzke. **Lifetime Subscriber:** $900+ ($700+ tax deductible) – Bill Godbout Electronics. **Corporate Subscriber:** $500/year ($440 tax deductible.

Advertising: Advertising space is available in this publication. Please direct inquires to the Advertising Director, *Recreational Computing*, Box E, Menlo Park, CA 94025.

Foreign Distributors of ***Recreational Computing*****: UK & Europe:** L P Enterprises, 8/11 Cambridge House, Cambridge Road, Barking, Essex, 1G11 8NT, Great Britain. Hofacker-Verlag, Tegernseer Strasse 18, D-8150 Holzkirchen, West Germany. Computerland/Computer Store AB, Box 7134, Kungsgatan 19, S-10387 Stockholm, Sweden. **Canada:** RS-232, 186 Queen Street W., Toronto, Ontario M5V 1Z1, Canada. **Asia & Australia:** Electronic Concepts Pty Ltd., 55 Clarence Street, Sydney, NSW 2000, Australia. Computer Store, POB 31-261, 22B Milford Road, Milford, Auckland 9, New Zealand. ASCII Publishing, 305 Hi Torio, 5-6-7 Minami Aoyama, Minato-ku, Tokyo 107, Japan.

Contents

Articles

Departments

Editor's Notes

Probably the most significant aspect of any editor's job is to seek out the kind of information required by the readership. That task becomes close to impossible, however, without readership input.

In the coming months, your input can, and will, help to direct and to motivate *Recreational Computing*. Your opinions, ideas and suggestions will be used as a guiding force behind any innovative and constructive changes. You may have already noticed some signs of change, much to your happiness – or chagrin?

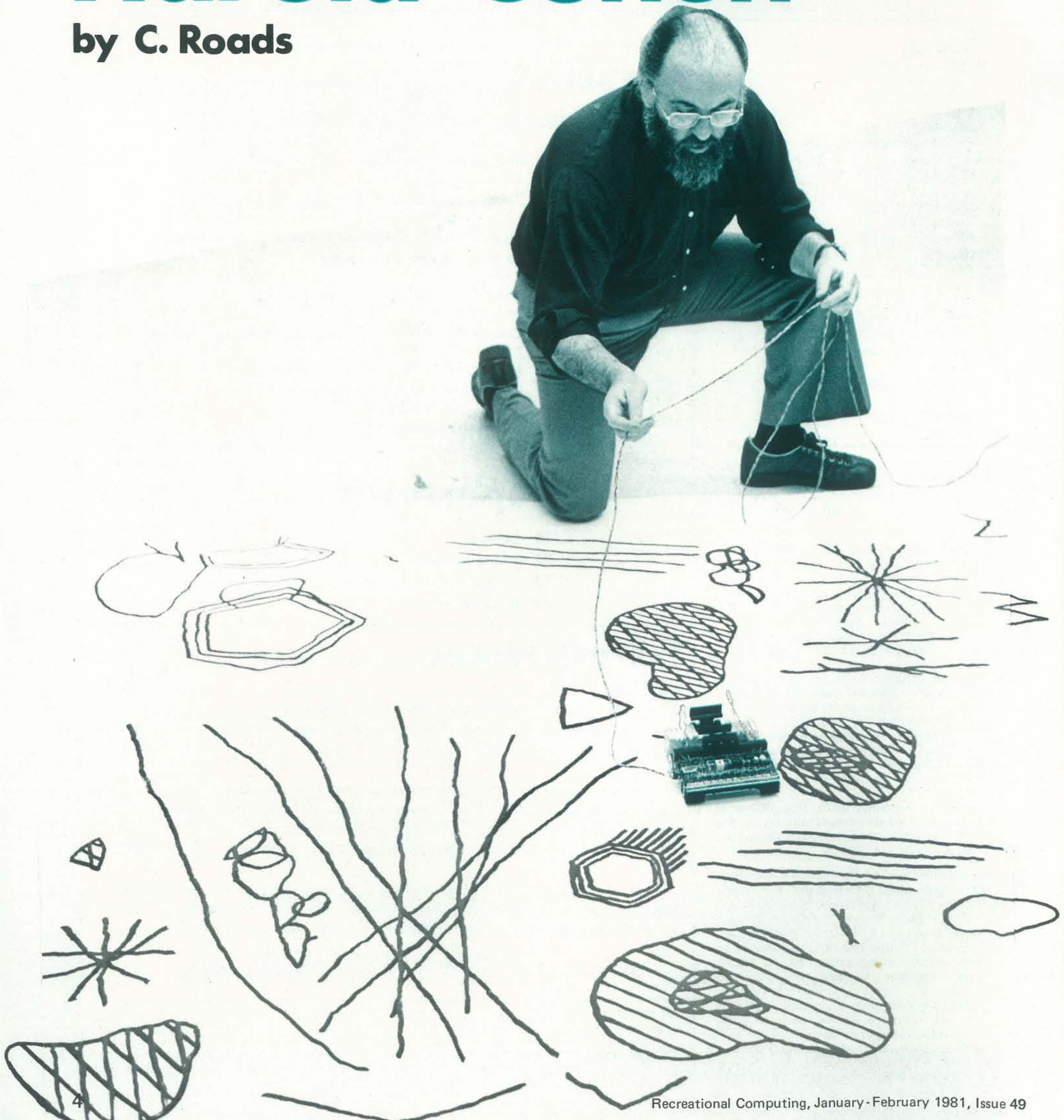
As your 1981 New Year's resolution, promise to send us your comments on improving *Recreational Computing*. Consider this your top priority for 1981, and we will in turn promise to deliver the kind of high caliber magazine you deserve.

"Happy New Year" from all of us at *Recreational Computing*!

RECREATIONAL COMPUTING (ISSN #0164-5846) is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a non-profit, educational corporation. Donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points. Address correction requested. Postmaster: send form 3579 to Box E, Menlo Park, CA 94025.

An Interview With Harold Cohen

by C. Roads



British-born artist Harold Cohen is a Professor with the Visual Arts Department of the University of California, San Diego, in La Jolla. Originally a painter, he has turned to making large pen drawings on paper. The instrument of this draughtsmanship is a small, mobile machine called a "turtle," designed and built by the artist, which is controlled by a computer program. The program presently runs on a PDP-11/45 computer. For added drawing accuracy, the turtle generates sonar signals which are detected by sonar microphones at the corners of the drawing paper. A microprocessor decodes the sonar information and sends it to the 11/45 as navigational feedback. The system has been exhibited in major galleries in the United States and Europe. This interview took place in early August, 1979, at the San Francisco Museum of Modern Art, where an exhibit of the drawing system was installed.

Background

CBR : Can you tell us a little about your background before you got involved in computer experiments?

HC : It's fairly simple. I was what you'd call a "career painter." I grew up in England, painted pictures, got successful, sold pictures, painted more pictures . . . Eventually, I got to the point of thinking "Well, maybe the pictures are ok but maybe there are more interesting things going on outside my studio than inside it." That's too brief a summary, because there were reasons why I thought that, having to do with the state of the work—things I wanted to do but couldn't. And that was the condition in which I arrived in California in '68, and by chance met somebody in the Music Department who said he'd teach me computing.

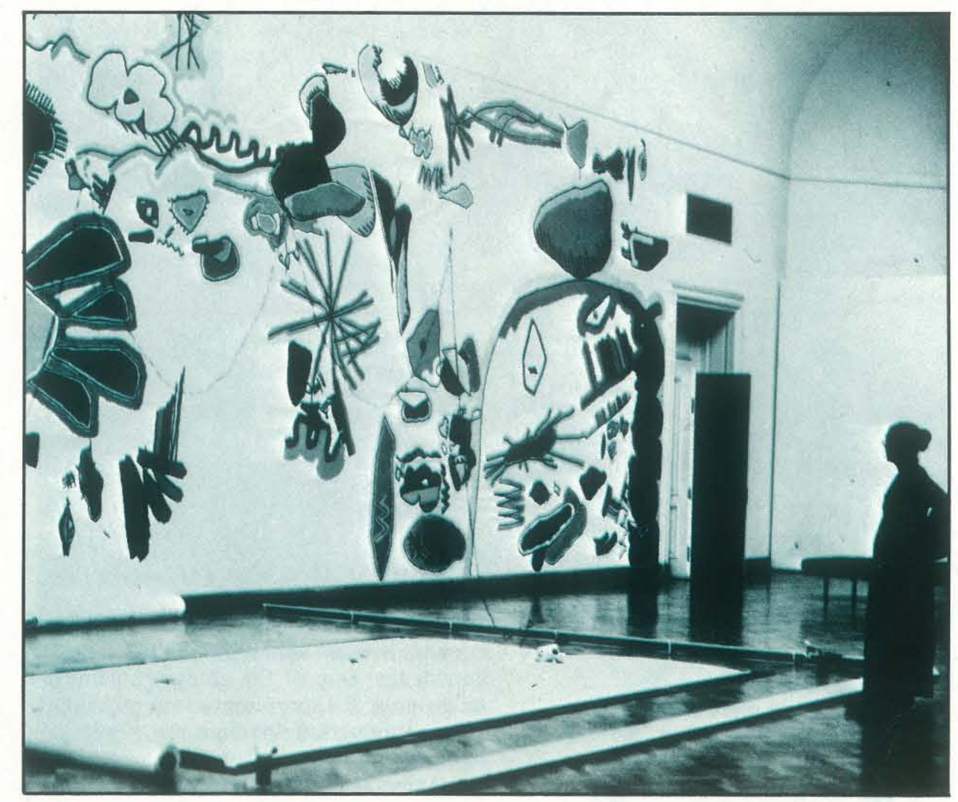
CBR : Can you tell us when you first began using computers in your artwork?

HC : Yeah. I didn't start using computers in relation to my work as an artist until a year or so after I became interested in computers as such. My involvement with computers went through all three stages. In the first place,when I first got interested in programming, I was simply knocked out by the sort of head-stretching ability; I found myself being exercised in ways I hadn't been for a long, long time. Then I went through a second stage being quite fascinated with the odd resemblances with reasoning processes that I seemed to be modeling.

CBR : So you got drawn in by the whole notion of the algorithm?

HC : I don't think I thought of it in that way, quite. I've always thought that the most important single statement in any language is the "if" statement—if such-and-such do this. That's what I meant really when I said I was fascinated by the power of the machine to double as a decision-making device. Then eventually it dawned on me that maybe I could use the machine in relation to my work, actually to face some of the problems I'd sort of put on one side because I couldn't think of any way of handling them. When I got involved with the computer my work was passing through a rather formalist set of preoccupations—which is one of the reasons I think that I thought that there were more interesting things going on someplace else. When I started using the computer in relation to painting I was using it in relation to formal problems, like: "How do you distribute colors on a flat surface?" What's a reasonable algorithm for doing something like that? The real breakthrough came after two or three years when I suddenly had a remembrance of things past and recalled that I had always thought that problems of

"I was what you'd call a 'career painter.' I grew up in England, painted pictures, got successful, sold pictures, painted more pictures . . ."



A mural depicting the artwork of Harold Cohen and his "turtle."

Photos by Becky Cohen

© 1979, C. Roads
Curtis Roads – Experimental Music Studio, Room 26-311, Massachusetts Institute of Technology, Cambridge, MA 02139.

"When I got involved with the computer my work was passing through a rather formalist set of preoccupations — which is one of the reasons I think that I thought that there were more interesting things going on someplace else."

meaning were a good deal more interesting than problems of form. I started to wonder whether in fact I could use the machine's propensity for decision-making as a way of getting at that. The present program is really the outcome of that.

The Organization of the Program

CBR : I'd like to ask some technical questions about the organization of your program, called AARON. You characterize the main part of the program as a production system. What do you mean by that?

HC : A production system covers a kind of program that is really reducible to "if" statements. If you read through it the whole program is of the form: if such-and-such is the case then do the following, and so on. Currently there are about 300 of those in the program.

Essentially what's at stake — almost anybody in the artificial intelligence community would believe that — the problem of emulating human behaviours is that human beings act upon the basis of knowledge. One of the going problems in AI is now the representation of knowledge. Production systems are a way of representing knowledge. There isn't a fundamental difference between saying I know something and saying I know the rules for doing something. What the production system represents is the rules for image-making.

Harold Cohen became interested in computers out of a fascination of the power of the machine to double as a decision-making device.

CBR : If I may debate a point you made just in passing, when you say that the program models human behaviour, I think that needs to be qualified somewhat. I'm thinking of drawing algorithms that one can use which involve rather directly processes of acquisition, of learning, of copying, of imitating, and your program doesn't incorporate those kinds of procedures. Your paper "What is an Image"* goes into that in much more detail. It's a deep problem to discern what exact part of human behaviour your program might be modeling. I think it's best illustrated by the thesis you assert at the beginning of your paper — researching the minimal determinants of an image.

HC : The way I characterize the problem is to ask what are the minimum conditions under which a set of marks would function as an image. It seems to me that one of the most fundamental ways of thinking of the human mind is to regard it as a device for establishing connections between things. What the mind does all the time is to say: "This stands for that." Human image-making rests on that kind of propensity. What the program investigates is: *how* that takes place. Of course, the point is that whatever the viewer sees is independent of whatever the computer "has in mind" because the computer doesn't *have* anything in mind, and is therefore called forth simply by the procedures, not by some intent to communicate. There's no intentionality towards meaning, there's simply a playing out of a set of procedures.

CBR : One of the significant aspects of your program is the way the program is hierarchically-organized. I know you have said there s no "boss" sitting at the top of the program that determines whether an image is good or bad, so how are decisions actually distributed about the system?

HC : It's only hierarchically organized in a conceptual sense in the sense that you might regard decisions about the entire picture as being hierarchically higher than decisions about the driving of the pen, say, or the making of a single-image element in the picture.

CBR : Well, one of the ways you get through the hierarchy is *via* the fact that your program is interrupt-driven.

HC : In fact, it's only interrupt-driven when it's operating in the real world with a real drawing device, as opposed to the graphics display screen. At the lowest level there are decisions about driving the turtle (which holds the pen) around, and that part is a fairly complex simulation of human freehand dynamics. It steers the pen in a way similar to the way you might drive a car, that is, it never knows in advance where it's going to go, it says: "I'll veer this way a bit or that way a bit and then I'll have another look to see how I'm doing." So, it's always waiting for feedback. In the real world that feedback is provided by a sonar navigation system, which, in fact, does interrupt to say it's ready for a reading.

* Cohen, H. (1979) 'What is an Image," *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, Tokyo, August 20-23, 1979, Vol. 2, pp. 1028-1057.

Mapped Representations

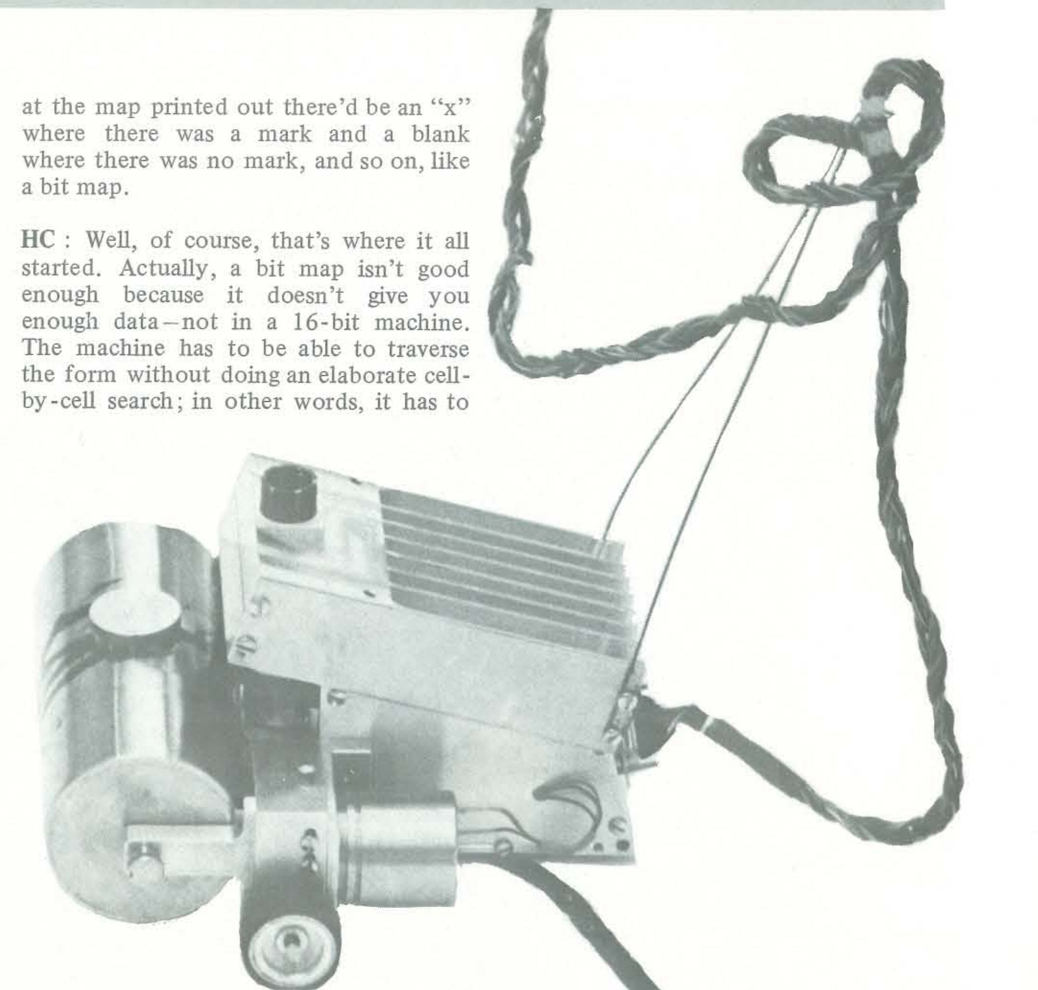
CBR : Another interesting aspect of your program is the multiple "maps," or representations that it maintains — like different ways of thinking about a drawing or different ways of seeing a drawing. How did you come to this notion, and how do these representations interact?

HC : That's changed over time. That goes through more changes than most anything else in the program. All the program's decision-making is based on what it knows about the drawing up to that moment in time. Consequently, the way in which it represents its knowledge of what it has done so far is central to the whole issue. I've tried various schemes. The general philosophy behind it is that the machine should never have to access more information than it actually needs in order to make a decision That's the reason for what you call multiple map structure.

I'm trying to push towards a situation in which everything is presented in a map, and the machine can access that map at any depth. It would be nice if the machine could see the drawing in a way analogous to the way we see the drawing. Consequently, in choosing between elaborate linked-list structures, record structures or whatever and having everything displayed absolutely flat on the surface my choice is the latter. For practical reasons that can't be done. You've only got 16-bits in a PDP-11 word. So for right now each of those words either contains enough information for the computer to use or it is used as a pointer to a linked-list, a property list. Everything then can be accessed through the map itself.

CBR : What you've been talking about is that the data actually would be in an iconic relationship to the drawing. They'd be very similar visually; if one could look at the map printed out there'd be an "x" where there was a mark and a blank where there was no mark, and so on, like a bit map.

HC : Well, of course, that's where it all started. Actually, a bit map isn't good enough because it doesn't give you enough data — not in a 16-bit machine. The machine has to be able to traverse the form without doing an elaborate cell-by-cell search; in other words, it has to



The "turtle" is a small, mobile machine that is controlled by a computer program.

contain a good deal of linkage information. You get to the point where if you do all that you're not also going to say anything about the figure . . . The program is actually parsing what it finds in the individual cells in order to find out what to do about it.

Audience Response

CBR : Since this work deals so closely with scientific issues like the modeling of human behaviour, do you see the output of AARON to be "test" art or "experimental" art, or somehow "less than" art?

HC : No, not at all. I think it's a part of art. Insomuch as I think anything is art I think the whole process is art. At some level European art has always been concerned with an investigation into the nature of art itself. To that extent, what I'm doing is a fairly orthodox kind of art activity. There aren't many people in the art world who are concerned with maintaining that kind of dialogue, but if you look into the past you find a lot of people who spent a lot of time thinking

A close-up view of one of Cohen's drawings.

about the nature of art. So I think of the whole process as being the art game. It's difficult to assess the importance of the objects produced by the program. Most of the time they're a way of determining whether the program is behaving in a reasonable fashion. It's evidence, verification.

CBR : And how do audiences respond to these verifications?

HC : When audiences are presented just with the verifications they treat them very much the same way they would any other art work. If I frame-up a bunch of drawings (and I've done this) the audience comes in and believes they're nice drawings or nasty drawings or whatever they believe. In fact they're interchangeable, in that sense, with what the human artist might do on his own.

CBR : Have there been any analyses of your works undertaken?

HC : There are analyses everytime somebody comes into the gallery and discusses what's on the wall, or what the turtle is drawing in terms of objects in the real world. There's always an amazing consistency in the things they think they see. I've thought for some time that the processes that go into image-reading are essentially the same ones as go into image-building. I've really stopped talking about them as two separate processes. I prefer to think about what I call the "image-mediated transaction" rather than what the artist does and what the viewer does.

Insistent Meaningfulness

CBR : We're starting to get into some other phrases you've coined, one of which is the "paradox of insistent meaningfulness." Could you explain that?

HC : Yeah. The kind of problem that I've faced with all this is to what degree the marks the machine draws are the determinant to what the viewer sees. There's no question in my mind that people will associate meanings with images if they can.

CBR : That's the insistent meaningfulness.

HC : Yes. The paradox is that the machine has no intentionality with regard to meaningfulness. It didn't intend to communicate anything.

Computer Music

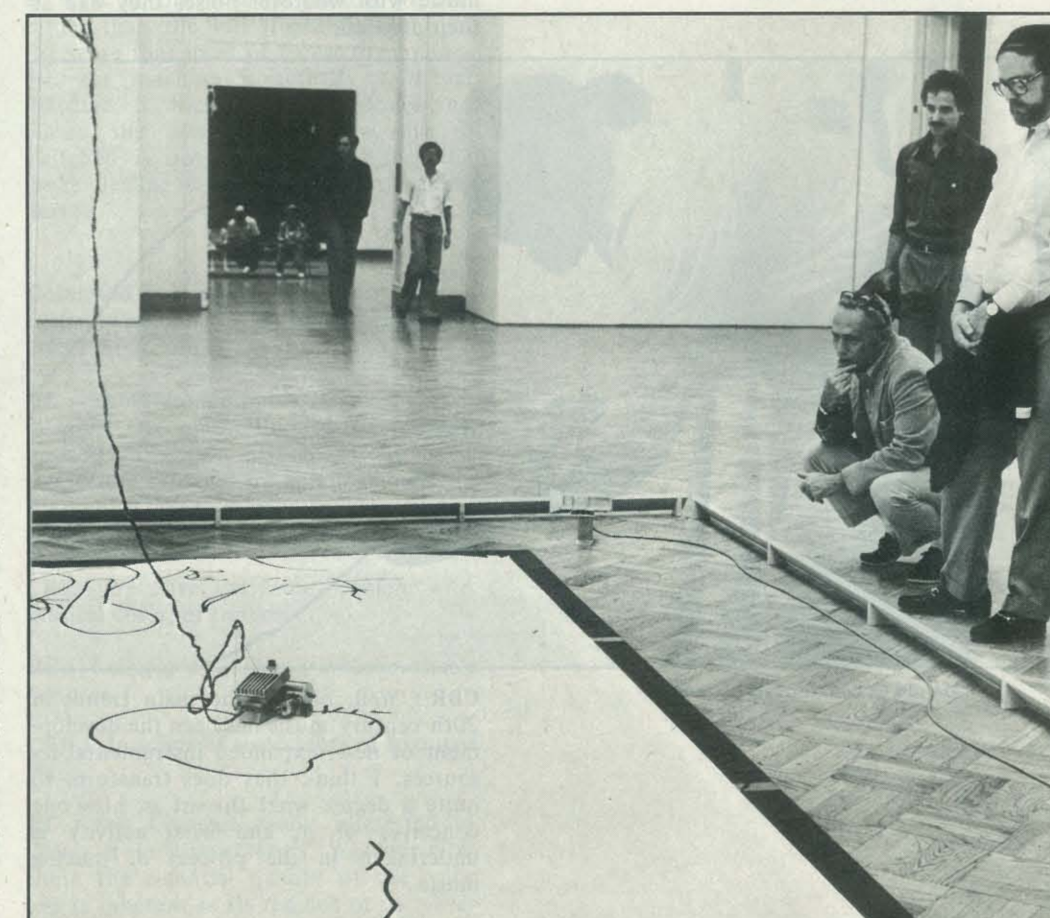
CBR : Can you draw a distinction between what you see being done in your drawing work and work being done in computer music?

HC : Insofar as I understand what's being done in computer music, I think the prime difference is that my main center of interest is in what human beings do and not in what machines do. Consequently my use of the machine has to do with modeling human processes in a way that I don't think is characteristic of work being done in computer music. I think of most work in your domain being involved in engineering primarily. I sometimes get the feeling there's an awful lot of energy going into creating the ultimate musical weapon – that can make all the noises that have ever been made and quite a lot that haven't been. I don't know enough about music to know whether the need to create new kinds of noise has ever been a fundamental musical problem. I have at various times talked to people in the music field and asked why people weren't using the machine to develop some sort of theory of composition in the same way that I've used it to develop a theory of image-making. People I've spoken to mostly think it's too difficult. I have no way of knowing why it would be more difficult in music than it is in visual imagery. But I think that would be the interesting thing to do.

To use the computer as an instrument, as a transformation device, which I think characterized what's been done in computer music, is to accept 19th century definitions of devices. It's fairly clear why those definitions came about and they've served very well, but until those definitions change, there's no reason to expect a fundamental change in their attitudes. Perhaps I can illustrate the kind of thing I'm talking about. I visited a place where they do high-level research on this, that, and the other – actually that's meant seriously, not trivially. And there were a bunch of guys standing around listening to a very convincing performance of a piece by Bach. I assumed it wasn't being played on the local church organ, for obvious reasons. And when it was finished, I said it was great and could they make it do something else and they said no because there's no one here who can play. And sure enough there was actually a keyboard sitting in the room. Someone was required to play the synthesizer the way one plays a church organ. Which is fine, and we got into a discussion where I asked what the point of all this was. The point one of them told me was that people with absolutely no knowledge of music can sit down and play music. I said I thought that was terrible. I thought that if people want to play music they should learn what music is. They seemed rather taken aback by that. The discussion went on in that vein for awhile and finally one of them said with a tone of desperation in his voice: "At least you've got to admit that it's a step in the right direction," and I said I could see why he thought it was a step but I couldn't see why he thought it was in the right direction. I'm sure you're much more familiar with this kind of attitude than I am; I don't live in the music world, but I confess to finding that kind of position horrifying. It doesn't propose any new anything. The idea that they're doing some great service to the world because people don't have to learn anything about music in order to make it is exactly compatible with the notion that you don't have to learn anything about a car in order to drive it, you don't have to learn anything about your refrigerator in order to keep your food cold; its absolutely mainstream "blackboxism," which

"There's no question in my mind that people will associate meanings with images if they can. The paradox is that the machine has no intentionality with regard to meaningfulness. It didn't intend to communicate anything."

The "turtle" at work.

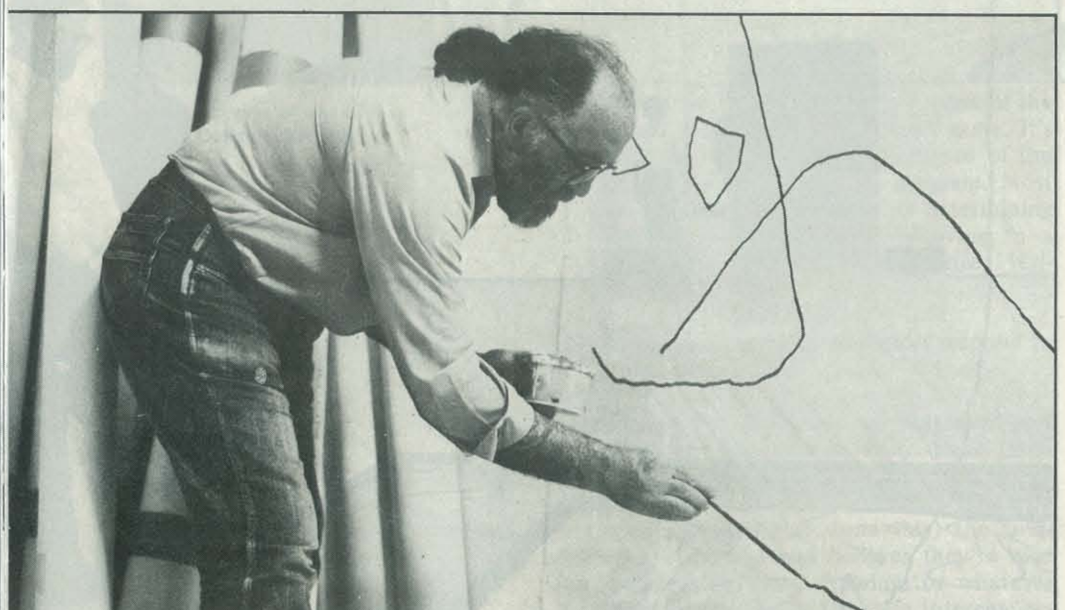


"At some level European art has always been concerned with an investigation into the nature of art itself. To that extent, what I'm doing is a fairly orthodox kind of art activity."

characterizes the whole of the 20th century and makes life exceedingly difficult.

CBR : If I can play devil's advocate, maybe you can tell me, what is the distinction between simulating drawing behaviour and simulating a musical instrument. In other words, do we need to "simulate" at all?

HC : The point is that there's no need to simulate a musical instrument unless you want to play it. I seem to remember one point in the conversation with these people asking them if anyone had a program that could generate music as convincing as "Twinkle Twinkle Little Star" and they said no that was too difficult. At that point it doesn't make any fundamental difference whether they can play Bach on a computer or simply play it on an organ. It also seems to me to not make that much difference whether the Bach is played by a 1000-piece orchestra from Hollywood or whether it's whistled. That's the point, I may be mistaken but I didn't think that the main issue, right through the history of music, was the invention of new noises. Most musicians it seems to me have been able to make music with whatever noises they had at their disposal.



Harold Cohen inspects his work up close.

CBR : Well, one of the main trends in 20th century music has been the development of new, expanded instrumental resources. I think that does transform to quite a degree what the art is, how one conceives of it, and what activity is undertaken in the process of making music.

HC : I don't think you can say, without examination of an individual case, that changes produced that way are fundamental. Also I should say that I don't think all change is necessarily for the good. I'm not convinced that the high technology of music recording has resulted in the enhancement of the excitement of music. The notion of the "perfect performance" has not necessarily been a step in the right direction. A lot of people still find a concert by a bunch of enthusiastic school kids rather more exciting than a highly polished performance by a major symphony orchestra.

CBR : One of the distinctions I see between what you're doing and what's being done in computer music is that there have been very few people who are attempting to simulate human composing behaviour. Hiller's early work is one exception. He tried to simulate harmony and some serial twelve tone types of operations. There have been many composing programs but most of them go beyond simulation of human behaviours into extensions of composing behaviour that would either not be possible or at least not feasible without computers. I'm thinking of the work of Koenig, Xenakis, and others. Could such languages be successful in the visual arts?

HC : I don't know what the equivalent in the visual arts would be . . . Oh, I suppose I do—John Whitney has used the computer to generate sequences in which, say, a thousand cubes rotate simultaneously. That's a ten-year job unless you do it by computer. I'm really not in a position to judge whether there are any fundamental changes involved. This is not intended as a put-down of John, whom I respect highly, but I'm not sure whether there's anything in the fact that he did that that separates him from what he might have done using straight Walt Disney animation techniques, except the fact that he did it. Walt Disney got over that problem of impossibility of doing animation simply by employing a very large number of people. Presumably John could have done exactly the same thing, apart from economic constraints, and didn't need the computer. At that point you'd have to say its not clear why that's a fundamental change.

CBR : But its very easy to conceive of tasks that no matter how many people you employed you couldn't do except with a computer for reasons of speed, precision, mass-storage capacity, and so on.

HC : Sure, the thing that comes immediately to mind is the credits to "Star Wars," for example, which were very exciting and couldn't have been conceived of without the computer. I think that's fine. I don't think everybody should be doing what I'm doing. Things are variously interesting or uninteresting depending on who brings what to bear on them. There must be a number of things like this in music, the spatial control of sound, for example, that couldn't be done without the machine.

CBR : This idea of extending compositional or artistic activity gets into interactive systems that would just not be feasible by any other means except with a computer, in which the system of interaction is a kind of meta-lauguage for talking about what you want to do; the computer actually realizes what you want to do. One of the things that Terry Winograd talks about in his recent article in *Communications of the Association for Computing Machinery* (CACM)* is computer-aided programming—the situation when a person will walk into a room, sit down, the computer will say hello, and they'll talk about what's to be done. It seems to me that's an extension of traditional activity that is a fundamentally different way of going about things, to talk over one's intentions with a computer.

HC : Of course. I think we're all a long way from that particular point. It's hard to speculate. It's part of the science fiction background that will in due course become a reality, presumably.

CBR : I think there are some intermediate stages that have a lot of significance.

HC : The levels of significance along the way tend to get lost with the jazzy final output. For example, one of the most interesting things that Terry Winograd did in his earlier work was to show that the machine was capable of inference. You could say: "Pick up the red block and put it down over there" and it would know what you meant by "it" and it would know what you meant by "over there" by reference to the previous conversation. It doesn't sound like much—as a matter of fact it was a major breakthrough. It was a major breakthrough because Terry at that point was interested in how human beings go about handling language, not because he was interested in building phenomenal toys for the future or interacting with machines, because the fundamental question then had to do with what human beings do when they talk to one another. What is the nature of a conversation protocol? Of course there are interesting things along the way. In the long term we can't really guess at what effect on the human psyche these kinds of development are going to have. We play it by ear the way human beings always play it by ear. We say we'll do it first and if we get into trouble we'll cross that bridge when we come to it. I have a strong taste for people understanding exactly what's happening. For example, I have a strong taste for people learning programming rather than getting somebody else to do it for them. From that point of view, a computer voice saying "Hello, how are you?" is actually masking you from some fundamental contact. My own view is that interactions of that kind are actually masking you from insightful use of the machine that I'd prefer.

* Winograd, T. (1079) "Beyond Programming Languages," *Communications of the ACM*, Vol. 22, No. 7, August, pp. 391-401.

CBR : Well, are you being "masked" from the computer when you use a high-level language? Some people would even say they dislike high-level languages because they don't get a "feel for the hardware," so to speak.

HC : No I don't think so, because what's at stake is the decision-making process rather than the bits floating around. In the sense that most of us use the computer, the machine is actually a virtual machine, it doesn't exist physically and where the bits go and how they're switched is not an issue. The machine we're dealing with is a decision-making device.

Color and Timbre

CBR : Well, what direction are you heading in the future?

HC : There are a number of things I have in mind. One of the things I'm interested in is the question of color. Color is an interesting problem for the machine on a theoretical level, merely because of the problem of how one characterizes color.

CBR : We have the same problem with musical color, or timbre.

HC : I simply don't know whether that's the case.

CBR : In terms of a spectrum of frequencies.

HC : No, I'm not really interested in doing it in those terms. Whether you think the essential quality of the color red is inherent in its redness or its wavelength is an old argument. I think I would have to say it's inherent in its redness.

"I'd be very interested in seeing whether it would be possible to give the machine some sort of visual input and obviously some sort of external knowledge about the external world and see if I could model the kinds of drawing processes human beings use in making a representation of something."

> **"In the sense that most of us use the computer, the machine is actually a virtual machine, it doesn't exist physically and where the bits go and how they're switched is not an issue. The machine we're dealing with is a decision-making device."**

CBR: Well, either way there's an analogy with musical timbre.

HC: Obviously, there would have to be.

CBR: If you wouldn't handle color as a spectrum of frequencies, how would you handle it?

HC: In terms of their properties.

CBR: Perceptual properties.

HC: In terms of their perceptual properties. I'm aware when working with color that although I haven't formalized the rules I'm operating on certain principles. My characterization of color is a strictly pragmatic one. I'm interested in how I would use color, and I think I know enough about color to be able to do that. That also implies that you don't have to worry that the human being recognizes, say, ten thousand discrete color sensations, all you have to do is worry about generating the one you want from the set of characteristics you think the color ought to have. If I move in that direction that's the approach I'd be taking. The problem there is the purely practical one—doing color without a good color output device would be silly. I've already done a bunch of paintings in which the computer said what the color ought to be and I dutifully filled it in, and I don't plan to do that anymore. Simply the problem of setting up a good color video system is more technology than I'm anxious to take on.

Hardware Futures

CBR: You're really taking on major logistical problems in using an 11/45 for museum shows, aren't you? If your computing requirements increase, will you finish up carting a VAX from place to place?

HC: It's a big enough hassle that I decided not to do any more shows until I could get at least 11/45 capability into a suitcase, and I assumed that would be a couple of years, at least. The big issue is memory: the space-requirements for an adequate representation of increasingly complex pictures will obviously increase, and a 16-bit machine is a 16-bit machine. The only reason for using an 11/45 rather than an 11/34 is the split I and D space, and that only doubles program size, more or less. The memory-management offered with the new Zilog Z8000 potentially removes that sort of limitation, and puts the 16-bit micro into direct competition with much bigger machines for a wide range of applications. The first full Z8000 system to be announced—ONYX SYSTEMS is the name of the company—looks to me like one of a new generation of machines, with design specs that won't be easy to improve upon for quite a while. They've packed the Z8000 processor, addressing a full megabyte of memory, a ten-megabyte 8-inch Winchester disk and a tape-cartridge backup drive, all into a single box, half the size of an RK05. An equally important design decision—very important from my point of view—is that ONYX plans to license a full UNIX, with the C programming languages, so by some time in 1980 my suitcase should be available, and I don't even have to rewrite anything.

CBR: What other projects have you in mind?

HC: The more important project is probably what I will get involved in because it comes more directly out of what I've been up to. Everything I've done in the program so far indicates the degree to which the sense of intentionality—the apparent meaningfulness of a drawing—is *not* actually dependent upon the intentionality of the maker. On the other hand, we know that in the real world human beings do act with intentions and they do sit down sometimes not merely to make a drawing but to make a drawing of something. Everything I've seen so far indicates that the nature of the visual cognitive processes is what informs all our behaviour in making drawings. But the kinds of marks people make in response to what they see don't actually have a one-to-one relationship with the things they see. In other words, you recognize a head, and you make a drawing, and you might even think the mark you put down is an outline, but when you compare it with the visual sensation it really isn't an outline, it's something else. Those "something else's" are very dependent on low-level cognitive processes, that's my intuition. I'd be very interested in seeing whether it would be possible to give the machine some sort of visual input and obviously some sort of external knowledge about the external world and see if I could model the kinds of drawing processes human beings use in making a representation of something. On a theoretical level my work is moving forward from the notion of imagery to a much broader notion of representation.

CBR: It's basically going to be "looking" at something.

HC: Yes, and making decisions about what would be the appropriate mark structure to represent that something. Don't expect me to have any results within the next three months, however.

PET Budget Program

Dear *RC*:

This program calculates your expenses and savings at the end of one week. I adapted this program from one in Kenneth Schoman Jr.'s *The Basic Workbook*, Hayden Book Company, Inc. The program calculates six categories for seven days. When you have entered each day's transactions at the end of the week, or earlier if you'd like, it will tell you your total for the week as well as the totals for every category. In this program the daily expenses include:

1. Rent and Utilities
2. Food
3. Transportation
4. Entertainment
5. Medical fees, Insurance, and Clothing
6. Savings

Because I am a beginning BASIC programmer, I do not know how to save the totals on tape for a monthly and yearly summary. I would appreciate a letter citing the solution.

Please note that in the listing, lines 60, 106 and 140 have clear screen and cursor control commands. You may wish to add these commands to line 12 as well.

Robert J. Walker
11 Brighton Drive
Gaithersburg, MD 20760

```
1 REM A PROGRAM WHICH TOTALS EXPENSES
2 REM ON A DAILY BASIS IN SIX CATEGORIES
3 REM FOR THE WEEK
4 REM WRITTEN BY ROBERT J. WALKER
5 REM ADAPTED FROM "THE BASIC WORKBOOK"
6 REM BY KENNETH E. SCHOMAN,JR.
10 DIM E(7,6)
12 PRINT "CATEGORY   TOTALS MAY BE"
13 PRINT "COLLECTED AT THE END "
14 PRINT "OF THE DAY'S RECORD."
15 PRINT "BE SURE TO INPUT AN ENTIRE DAY'S DATA."
16 PRINT "ARE YOU READY?    PRESS 1"
17 INPUT R: IF R=1 THEN 20
20 LET A=0
21 LET C0=C1=C2=C3=C4=C5=C6=0
22 LET Z=0
30 REM D=DAYS, C=CATEGORIES
40 FOR D=1 TO 7
50 FOR C=1 TO 6 STEP 1
60 PRINT "□INPUT:DAY";D;"CATEGORY";C;
70 INPUT E(D,C)
72 LET Z=Z+1
80 LET A=A + E(D,C)
90 NEXT C
99 LET C1=C1 + E(D,1)
100 LET C2=C2 + E(D,2)
101 LET C3=C3 + E(D,3)
102 LET C4=C4 + E(D,4)
104 LET C5=C5 + E(D,5)
105 LET C6=C6 + E(D,6)
106 PRINT "□THE VALUE C1=$";C1
107 PRINT "THE VALUE C2=$";C2
108 PRINT "THE VALUE C3=$";C3
109 PRINT "THE VALUE C4=$";C4
110 PRINT "THE VALUE C5=$";C5
111 PRINT "THE VALUE C6=$";C6
112 FOR X=1 TO 7000
120 NEXT X
121 IF Z=6 THEN PRINT "□□□□DO YOU WANT TOTAL?
    Y=1,  N=2":LET Z=0:INPUT Y
122 IF Y=1 THEN 140
130 NEXT D
140 PRINT "□□□□TOTAL AMOUNT SPENT IS $";A
141 PRINT
142 PRINT "C1=$";C1;"C2=$";C2;"C3=$";C3
143 PRINT "C4=$";C4;"C5=$";C5;"C6=$";C6
144 PRINT
145 PRINT "DO YOU WANT TO CONTINUE?"
146 PRINT "Y=1 , N=2"
147 INPUT N
148 IF N=1 THEN 130
160 END
```

Dear Editor:

Thank you for reviewing my new children's book, *Katie and the Computer*. The reviewer, Dennis Allison, was right on target, summarizing the book as "an allegorical storyline where characters who personify the various functions of a computer act out for Katie how the computer works."

Unfortunately, Dennis also made some puzzling comments that I'd like to respond to. First, he felt that the book was a "kid's book written for parents" and that the book didn't amuse him (the adult). My reaction: "Dennis, you're wrong." I wrote the book to explain computers to children – very young children – not to adults. In fact, I thought of the book as a way to teach and amuse my own daughter, Catie, who was only two and a half when I first told her the story.

Second, Dennis wrote that he tried the book out on *his* daughters and that they were "confused" and "bored" by the book. No wonder! Given his attitude toward the book, did he expect his children to react otherwise? Also, how old are Dennis' daughters? The book is intended for children from age 3 to age 8. Perhaps his daughters are already too knowledgeable and too old to react with the same sense of wonder that countless young children have shown when they first see the book. There's no need to throw all the book's factual material at the kids right away (which is what might have confused Dennis' daughters). The parent or teacher can get to that later after the child has been entertained and intrigued by the fantasy, and after he or she has absorbed the terms and concepts embedded in the storyline.

As to the book's appeal to people other than Dennis, I'd like to mention that the book has already sold thousands of copies, and it's only been out a few months. Second, I've read the book to over 4,000 children, from preschoolers on up to sixth graders. The kids love Katie's fantasy adventure inside the computer. Invariably they sit quiet and spellbound through the reading. After the reading, we have an animated, fast-paced discussion about how the story relates to what goes on inside a real computer. The book has proven to be quite successful in exciting the kids' imagination and in introducing them to real, physical-world *analogues* to the complex processes inside a computer.

Furthermore, the book has been chosen by the North Carolina Book Club as its Summer 1980 selection, and it has been recommended by numerous teachers, librarians, parents, and other reviewers as an excellent *classroom* aid in introducing computers to young children. In this regard, I'd like to ask Dennis to give the book a second chance and take a look at the cover article on the book in *Creative Computing*'s September 1980 ("Education") issue.

In any event, in spite of the disappointing review, I just want you to know that I think that PCC, *Recreational Computing*, and Dennis Allison are all terrific. I've enjoyed reading your stuff ever since I first met Bob Albrecht, back in May 1975.

Fred D'Ignazio
730 William Circle
Chapel Hill, NC 27514

Dear Editor:

I just read *Recreational Computing*, Sept-Oct 1980, and found it very enjoyable; it was much "meatier" than recent issues.

One request though – if you publish a program listing, please give the *whole* thing. Mark Zimmerman's article, "Textrapolation," p. 48, is quite useless without the machine langauge routine that goes along with it. I was looking forward to trying it out.

1100110011100101100,
Irwin Tillman
24 King Road
Kings Park, NY 11754

Dear Editor:

I am pleased to find that there is some interest in providing a BASIC language for programmers whose native language is Spanish. I regret that I made a number of errors in my choice of words, and am grateful to Niki Delgado and W.J. Morrissey for offering alternatives. Their choices seem well suited to a Spanish BASIC, with the possible exception of IMPR. Perhaps ESCRIBE might be preferable, or simple ESCR. Naturally, in an actual implementation of Spanish BASIC, one would also want the error messages to be displayed in Spanish.

Sincerely,
Jim Day
17042 Gunther Street
Granada Hills, CA 91344

Dear Editor:

Thank you very much for your help regarding the three computers. I recently went to a store that specialized in the Apple and you are right, there is an overabundance of software for it. But, I have also found an overwhelming amount of software for the TRS-80. Please help me. I want to get a system that is educational, will suit my needs, be good for utilities, and yet be reasonably priced so that a high-school student like myself can afford it.

Howard Matalon
32 Shelley Road
Springfield, NJ 07081

Educators in Need

Editor's Note: Recreational Computing *intends to follow microcomputers in education and will continue to publish timely articles on the advances and uses of microcomputers in this field. We will also contact educational facilities with an eye towards publishing educational programs in our future issues.*

Dear *RC*:

I am using at the present time in our school district's Industrial Arts program, an inventory-requisition card system that numbers over 6,000 items of our supplies, materials, tools, etc. Because of the enormous and time-consuming task of inventory, recording information on cards in preparation of budgets and requisitions, identifying and pulling cards for typing of requisitions, typing requistions and returning cards to the file, recording bid prices and placing orders, I am having trouble finding enough hours during the work day to perform my supervisory duties. Therefore, I am investigating the possibility of putting all this work onto a computer system. I do not know the capabilities of the computers nor do I know what type of computer to investigate. Do you have any information that might help me to get started so that I could make recommendations to my superiors?

Sincerely,
Clyde Cover, Supervisor
Industrial Arts Department
Cumberland Valley School District
6746 Carlisle Pike
Mechanicsburg, PA 17055
(717) 697-8261

Dear *RC*:

We wish to expand our library of cassette program tapes for use with our 8K PET microcomputers.

We are primarily interested in instructional programs for use in our reading, mathematics, science and social studies programs in grades K-12. However, if you have any other programs that you feel would be useful, we would like to hear about them.

Sincerely,
Robert W. Smith
Administrative Assistant
Curriculum & Instruction
Lawrence Public Schools
Reilly Road
Cedarhurst, New York 11516
(516) 295-2700

Dear *RC*:

Our school system recently purchased three PET computers. We are anxious to learn more about how to best use them in our classes.

We saw your company listed in the February 1980 issue of "Teacher."

We are interested in acquiring software and subscribing to magazines that would serve our needs.

We would appreciate it if you would send us any information available regarding what your company has to offer in the way of computer related materials.

Thank you very much.
Sincerely,
William H. Martin
Teacher
Robert Kerr Community School
9591 East Monroe Road
Durand, Michigan 48429
(517) 288-2805

Dear *RC*:

We have started the use of microcomputers in our schools this fall and we are interested in putting periodicals in our schools that will promote the program with both students and teachers.

A Minnesota Department of Education publication entitled "School Library Materials" lists your publication, *Recreational Computing*, as one that is recommended by the Minnesota Educational Computer Consortium.

We would much appreciate receiving a sample of your publication.

Thank you.
Sincerely,
Orris Wibe
Director of Media Services
Moorhead Public Schools
2300-4th Avenue South
Moorhead, Minnesota 56560
236-1918 ■

Recreational COMPUTING Back Issues

PCC NEWSPAPER

HALF OFF SALE!

All 12 Available back issues of PEOPLE'S COMPUTER COMPANY NEWSPAPER for only $12.

#3 May-June 1973 ZINGO/"Going to Bid"/ How to Buy an Edusystem/BASIC MUSIC CHOMP. $2.00

#10 July-Aug 1974 A Fantasy of Future Forms/Low-Cost Home School Microprocessor Systems/INCHWORM/MAZE/ Analysis of Teaser/Computer Illiteracy Problem by Peter Lykos. $2.00

#13 Jan-Feb 1975 Build your own BASIC/ Mysterious RND/More INCHWORM/ Computers & Art/BIOSIN Son of Hardware. $2.00

#18 Nov-Dec 1975 NUMBER/Tiny BASIC/ Analog to Digital Conversion/Minuet in G Major. $2.00

#19 Jan-Feb 1976 Space Games/Tiny BASIC Soloworks/DIDDLE/STTR/More Music. $2.00

#20 Mar-Apr 1976 Games for the Pocket Calculator/SQUARES/Make Believe Computers/STRTRK/Huntington Computer Project. $2.00

#21 May-June 1976 Biofeedback/POUNCE/ "We're Building Our Own Computer"/ Programmer's Toolbox/CCC/TV as Terminal and Game Center. $2.00

#22 July-Aug 1976 Low-Cost Software/Tiny BASIC, Tiny Trek/Your Brain is a Hologram. $2.00

#23 Sept-Oct 1976 Dungeons & Dragons/ HATS/One on One/PLANETS/The Positive of Power Thinking. $2.00

#24 Nov-Dec 1976 STORY/SNAKE/More Build Your Own Computer/Introducing PILOT/FROGS. $2.00

#25 Jan-Feb 1977 REVERSE/Robots/Tiny PILOT/Space & Computers/Conversational Programming. $2.00

#26 Mar-Apr 1977 Z-80 PILOT/6502 Assembly Programming/Tiny BASIC for Beginners. $2.00

PEOPLE'S COMPUTERS

#27 May-June 1977 Graphics, Computers & Copyright Laws, PILOT & Tiny BASIC, Home Computers for Beginners, Women & Computers. $2.50

#28 July-Aug 1977 Heathkit's 8080 & LSI-11, Do-It-Yourself CAI, Robot PETs. $2.50

#29 Sept-Oct 1977 Computer Networks, The $595 PET, More Tiny Languages. $2.50

#30 Nov-Dec 1977 Oz Graphics, Biofeedback & Micros, Our PET's First Steps. $2.50

#31 Jan-Feb 1978 Robots, Video Discs, PET Update & Programs, PASCAL & COMAL. $2.50

#32 Mar-Apr 1978 Epic Computer Games, TRS-80 Review, Micros for the Handicapped. $2.50

#34 July-Aug 1978 Computer Whiz Kids, Public Access to Computers, Man-made Minds. $2.50

#35 Sept-Oct 1978 Computers & Museums, Kingdom, APL, Sorcerer of Exidy. $2.50

#36 Nov-Dec 1978 The Return of the Dragons, APL, Animated Games for TRS-80. $3.00

RECREATIONAL COMPUTING

#37 Jan-Feb 1979 Artificial Intelligence, A fantasy by Jules Verne, TRS-80 Personal Software, The Apple Corps is with Us. $3.00

#38 Mar-Apr 1979 Artificial Intelligence, Calculator Comics, Chess Reconsidered, Lord of the Rings. $3.00

#39 May-June 1979 Now Peter Can Read, Game of Life, TRS-80 Programs, Detective Game Program. $3.00

#40 July-Aug 1979 ComputerTown, USA!, Cryptarithms, DP Games Playing, FORTE Music Programming Language. $3.00

#42 Nov-Dec 1979 Shogi Games, Creating New Characters on the Ti 99/4, Music for Atari, TRS-80 & PET. $3.00

#43 Jan-Feb 1980 Computing for Health and Equality (PET), Ti Graphics & Animation Pt. 2, More Music for Atari. $3.00

#44 Mar-Apr 1980 Special Games Issue! Recreation Apple II Hi-Res Graphics, Galaxy II, TRS-80 Tiny Trek. $3.00

#45 May-June 1980 Intro to Computer Music, CBBS Phone Numbers, 6502 Machine Language, Five New Personal Computers. $3.00

#46 July-Aug 1980 Fantasy Games! Write Your Own, Wizard's Castle, Star Trek, Wonderful World of Eamon, The End of Fortran Man. $3.50

#47 Sept-Oct 1980 8th Anniversary Issue! Best of PCC, Probability Trees, Computers in Sports Medicine, Wired by Mark Singer. $3.50

#48 Nov-Dec 1980 Computerized Voting, Hot Rod Computers and Model T Fords, The Story of Computer-Using Educators, Games! $3.50

Only the issues listed are available. Outside the U.S., add $.50 per issue ordered. Price includes issue, handling and shipment by second class or foreign surface mail. Within the U.S., please allow 6-9 weeks to process your order second class. For faster service within the U.S., we'll ship UPS if you *add* $1.00 for 1-2 issues and $.50 for *each* issue thereafter. We need a street address, not a P.O. Box. Outside the U.S., *add* $1.50 *per* issue requested for airmail shipment.

Please send me the issue(s) circled.

3	10	13	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	34	35	36	37	38	39	40	42	43	44	45	46	47	48

☐ I'm ordering $20 or more and I want a 10% discount!

I enclose $ ______ (U.S. check or money order). Outside the U.S., add $.50 per issue. Please send in this form or facsimile to: Recreational Computing, 1263 El Camino Real, Box E, Menlo Park, CA 94025. This offer expires June 30, 1981.

NAME

ADDRESS

STATE ZIP

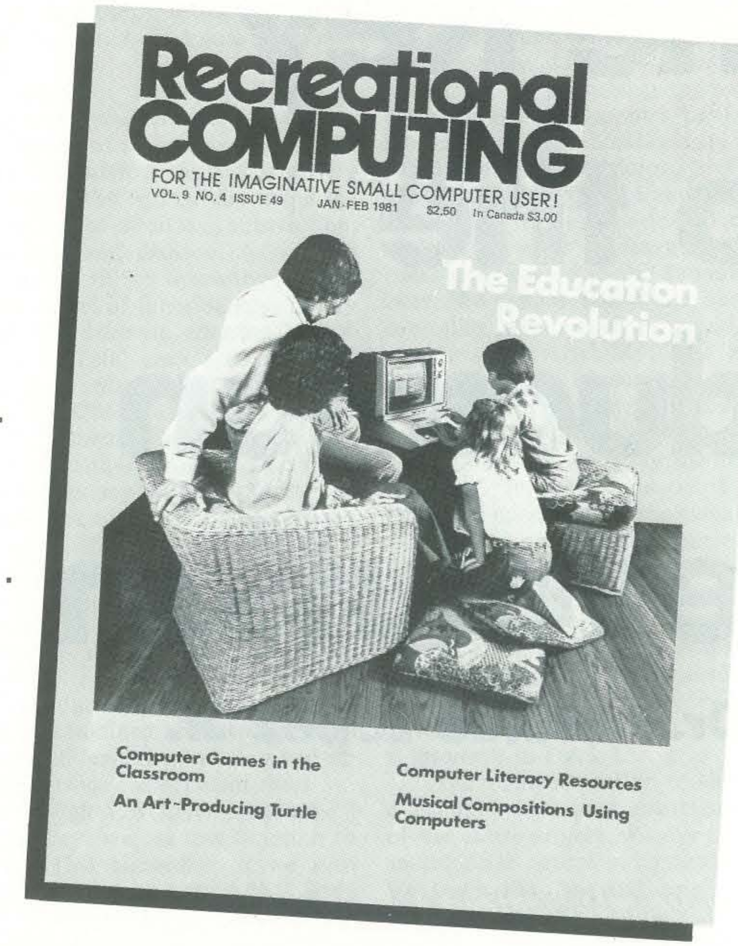
Graphics . . .

Games and the Arts . . .

Computers and Learning . . .

Creativity and Programming . . .

Sports and Medicine . . .



All these and other active, live ideas have packed the pages of **Recreational Computing** since its inception as the <u>first</u> personal computing magazine.

Subscribe now and learn why we were the first and still going strong.

Yes! I want to subscribe to Recreational Computing!
Please bill me $12 for 6 issues.

Name

Address

City, State, Zip

Dept. L6, Box E, Menlo Park, CA 94025.

The Key to the Education Revolution

by Dr. Antonio Lopez Jr.

DR. ANTONIO M. LOPEZ, JR. is a an Associate Professor of Mathematical Sciences at Loyola University, New Orleans, Louisiana. He has been working on the SOL-20, PET, APPLE, and TRS-80 microcomputer systems since 1977. Besides teaching in the NSF Mathematics Teacher Development Program, Dr. Lopez has also taught continuing education courses using the TRS-80 microcomputer system.

Why has the use of handheld calculators and microcomputers not caused the "promised" revolution in education? Some may ponder this question (1). The answer seems all too obvious; we have not educated our teachers and parents in the use of these valuable tools. In many universities today, education majors are not required to enroll in even one computer science course. Basic computer literacy would enable teachers to handle the "school learning" and parents the "out-of-school learning." In New Orleans, continuing education programs have been established at Loyola University to assist interested adults in becoming computer literate (2). This paper explains a development program used by the Department of Mathematical Sciences since 1977 to educate mathematics teachers in the use of handheld calculators and microcomputer systems.

In December of 1976, the department began preparing a grant proposal for the National Science Foundation entitled "Pre-College Teacher Development Project for Middle School, Junior and Senior High School Mathematics Teachers of the Greater New Orleans Area." An advisory group of area teachers was formed; their discussions produced the following observations:

1. The handheld calculator is found in schools at all levels, but many teachers lack the knowledge necessary to use it effectively.
2. Students had not been given problems involving large amounts of computation because of time considerations.
3. High schools in the area were purchasing expensive computer time from universities and businesses, but few teachers knew what to do with their computer terminals. Those who had some knowledge were often dissatisfied with the "package software" for specific subjects and could not change it.
4. Many teachers had not been exposed to the subject areas of probability, statistics, approximation concepts and elementary optimization.

Unable to attack these problems at all levels with one program, we divided the teachers into two groups. Track I consisted of mathematics teachers with minimal foundations who were teaching mathematics in grades 5 through 9. In Track II were high school mathematics teachers with fair-to-good foundations. The courses of both tracks emphasized problem solving and appropriate use of computational tools such as the handheld calculator and microcomputer.

During the academic year, classes were held on Saturday mornings for three hours. Mathematical topics in number theory and elementary abstract algebra were covered with Track I. Octal and hexadecimal number systems along with algorithms such as the division algorithm and determining primes were given special treatment. Track II was instructed in linear algebra with an emphasis on the use of vectors and matrices. With each group we introduced the handheld calculator as a classroom tool and used it in our study of probability and statistics. The following summer program exposed the teachers to problem-solving techniques using the microcomputer. For three consecutive years, this program has received funding from the National Science Foundation.

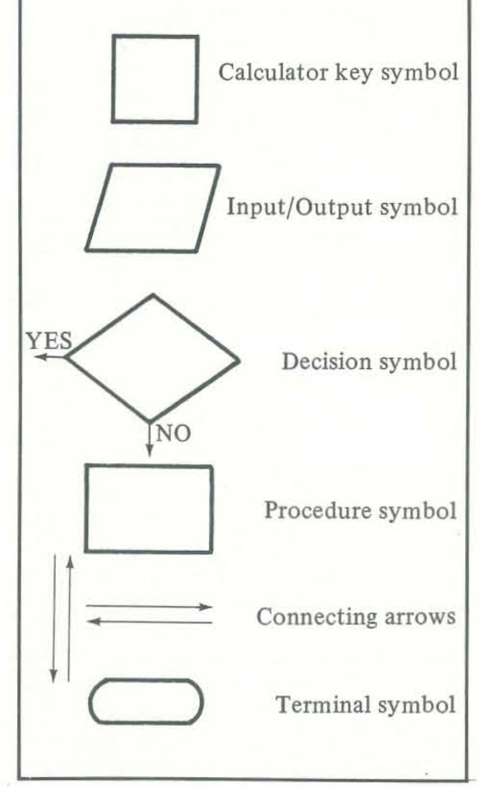
From the beginning, we introduced a problem-solving procedure that has proven successful in teaching formal mathematics and computer science courses. The steps are as follows:

Get a clear understanding of the problem. Although the most basic step, this is often the most difficult to grasp. The problem solver must recognize what is given (INPUT), what is to be found (OUTPUT), and how to get from given to found (PROCESSING).

Select a method of solution. Too often students get the impression that there is only one way to arrive at a correct answer. Handheld calculators and microcomputers are helping to destroy this myth. We show the teachers how some problem-solving methods are preferred over others because of outside circumstances or computational considerations. For example, we ask the teachers to find the average of the grades 85, 70, 95, and 80. They quickly formulate *the* solution by (85+70+95+80)/4. Another method of solution is 85/4+70/4+95/4+80/4. The former is preferred because it requires three fewer divisions.

Sketch your solution on paper. One way to test the understanding of a subject is to explain it to someone else. Trying to write a formula to solve a problem is a type of algorithm design, a very important concept in mathematical sciences today. To assist with algorithm development we introduce the flowchart symbols found in Figure 1.

Figure 1. Flowchart Symbols.



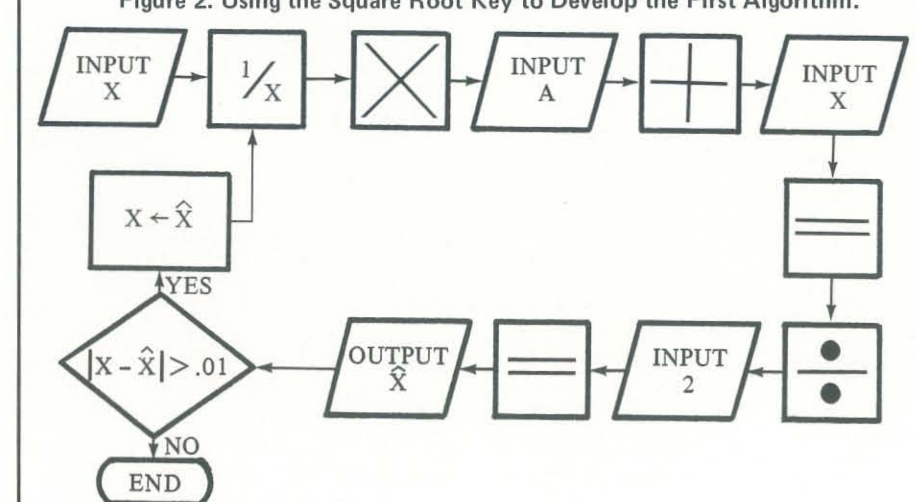
Implement your solution on a computational tool. While some problems can be solved without using a handheld calculator or microcomputer, there are many problems that are better and more easily solved by using them. Their speed and accuracy make them valuable assets in a fast moving and complex society.

Check your answer. The science of designing test data for a problem is coming into its own. We show the teachers that an algorithm must be tested with input data for which we know the resulting output.

Our study of the handheld calculator acquaints teachers with the three different types of operating systems in calculators. Calculators can have a variety of keys beyond the decimal digits and the four functions of addition, subtraction, multiplication, and division. A working appreciation of all the operating systems and a large subset of function keys is very important. For example, entering 2 + 3 x 4 = into a TI-1200 gives a result of 20 while the same entry on a TI-30 gives an asnwer of 14. Consequently, we encourage sound mathematical sentence structure in our algorithms. Function keys also provide good examples. On a TI-30 the entry 0 y^x 0 results in a 1 being placed in the display. However, the same entry on an HP21 gives ERROR. A confident mathematics teacher knows that 0^0 is undefined so the TI-30 has a design error. Is it also a design error when we enter 2 +/- y^x 2 = and get an error while 2 +/- x^2 produces 4? These "phenomena" must be examined and explained because they can lead to confusion and incorrect answers via correct algorithms.

Even though most calculators have a square root key, we use this function to develop our first algorithm. If we wish to find $\sqrt{a}$, we let x be a guess as to what it is and then calculate the next guess $\bar{x} = (x+a/x)/2$. If x and $\bar{x}$ are "sufficiently close," then we have the desired result; if not, we repeat the process. The flowchart is given in Figure 2. This algorithm can be extended to do cube roots also. We develop a number of other algorithms in the same fashion; e.g., the Pythagorean Theorem, Quadratic Formula, Division Algorithm and the Greatest Common Divisor Algorithm.

Figure 2. Using the Square Root Key to Develop the First Algorithm.



Because the advisory group mentioned a lack of teacher experience in using the calculator for statistics, we lead our algorithm development into the subject areas of probability and statistics. Counting rules for samples, permutations, selections and combinations are now executed on the calculator. Measures of centrality (mean, mode and median) are taught, and the calculation of the mean of large amounts of data (50 to 100 entries) is delegated to the handheld calculator. We cannot talk about measures of centrality without giving equal time to measures of variability, and again we make heavy use of our tool. We emphasize the binomial distribution as our discrete probability distribution, and for the continuous case we study the normal distribution. Interpolations on the normal tables are handled with the calculators.

At this point in the course, we differentiate between the two tracks by proceeding with least squares and linear regression in Track I and linear optimization in Track II. The reason for this division toward the end of the academic year is to exploit the matrix theory learned in Track II and the concepts of slope and y-intercept in Track I.

The summer portion of this program is probably the most avant-garde aspect of the entire project. We give the teachers an intensive course in computer literacy. We teach the "ins and outs" of BASIC and how to write Computer Assisted Instruction (CAI) programs. Classes are held six hours a day for fifteen days and each day is a complete learning module. Each three-hour block is divided into a one-hour lecture, twenty minutes of review, a twenty-minute quiz and a supervised lab period of an hour and twenty minutes to allow for plenty of "hands-on" experience.

It has been our experience that teachers dislike hard-to-use microcomputer systems that are programmed by some type of machine language or assembler. Consequently, we use three popular microcomputer systems – the PET 2001, APPLE-II, and TRS-80 Level II. Although there are slight differences in each computer's BASIC, we define a subset that can be used on all the systems (3).

To ease the transition from handheld calculators to microcomputer systems, we start by using them as elaborate calculators. The algorithm for finding the hypotenuse of a right triangle given its two legs was developed during the academic year. We now show its implementation on microcomputers in their calculator mode:

```
>INPUT A,B:? SQR(A*A+B*B)
```

Because it is sometimes difficult to

Figure 3. Square Root Algorithm Program.

```
1     ' ***** SQUARE ROOT ALGORITHM *****
2     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
3     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D.
4     ' PROGRAM-ID: SQRTAL            LAST-MODIFIED: MAY 23, 1977
10    INPUT"ENTER NUMBER FOR WHICH YOU WISH THE SQUARE ROOT";A
20    INPUT"WHAT IS YOUR GUESS TO THIS PROBLEM";X
30    Y=(X+A/X)/2
40    IF ABS(X-Y)>.01 THEN 60
50    PRINT "THE SQUARE ROOT OF ";A;" IS ";X
55    PRINT"ACCURATE TO TWO DECIMAL PLACES.":END
60    X=Y:GOTO 30
```

Figure 4. Complete Programming Model.

```
1     ' ***** STATISTICAL PROGRAM *****
2     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
3     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D.
4     ' PROGRAM-ID: STATPK            LAST-MODIFIED: MAY 23, 1977
10    INPUT"HOW MANY DATA ELEMENTS";N
20    DIM X(N)
30    S=0:T=0
40    PRINT"PLEASE ENTER ONE DATA ELEMENT AT A TIME"
50    FOR I=1TON:INPUTX(I):S=S+X(I):T=T+X(I)*X(I):NEXTI
60    ' ***** EXCHANGE SORT *****
70    M=N
80    H=1
90    FOR I = 1 TO M
100           IF X(H) < X(I) THEN H=I
110   NEXT I
120   IF H=M THEN 140
130   R=X(H):X(H)=X(M):X(M)=R
140   M=M-1
150   IF M<>1 THEN 80
160   ' *************************
170   IF INT(N/2)*2 = N THEN 270
180   Y=INT(N/2)+1:M=X(Y)
190   A=S/N:B=SQR((N*T-S*S)/(N*(N-1)))
200   R=X(N)-X(1):V=B*B
210   PRINT"THE MEAN IS ";A
220   PRINT"THE MEDIAN  IS ";M
230   PRINT"THE RANGE IS ";R
240   PRINT"THE STANDARD DEVIATION IS ";B
250   PRINT"THE VARIANCE IS ";V
260   END
270   Y=(N/2):M=(X(Y)+X(Y+1))/2:GOTO 190
```

understand the recursive definition processes of computers, the square root algorithm (refer to Figure 3) is our first venture into actual programming. The mean and standard deviation algorithms that we program next are also very helpful. In developing an accurate data model, we want to be able to find the variance and range, and order the data to obtain the median. The teachers formulate their own algorithm for sorting and learn a good deal about array manipulation in the process. The model is complete (see Figure 4) as is the transition from handheld calculator to microcomputer system.

It is our opinion that the best use of any microcomputer system in education is made by the individual teacher. Show Figure 5) evaluates the function $f(x) = 4x^2 + 3x - 10$, where x is selected randomly from integers 0 to 10. We explain that the values of the function argument are set in the RND statement so the teachers know how to change the range

Figure 5. Program to Evaluate a Function.

```
1     ' ***** FUNCTION EVALUATION *****
2     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
3     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D.
4     ' PROGRAM-ID: FUNCT1            LAST-MODIFIED: MAY 23, 1977
10    PRINT "HELLO, I AM THE TRS-80 MICROCOMPUTER SYSTEM"
20    INPUT "WHO ARE YOU ";N$
30    PRINT:PRINT "IT IS NICE TO KNOW YOU ";N$
40    PRINT "TODAY WE ARE GOING TO PRACTICE THE"
50    PRINT "FUNCTION EVALUATION OF"
60    GOSUB 1000:C=0:T=0
70    RANDOM:A=RND(10):T=T+1
80    F=4*A*A+3*A-10
90    PRINT:PRINT  "WHAT IS F(";A;") = ";:INPUT G
100   IF F = G GOTO 150
110   PRINT "NO! ";N$;"'F(";A;") = ";F
120   PRINT "REMEMBER THE VARIABLE X IS A PLACE HOLDER"
130   PRINT " AND WE REPLACE IT WITH ";A
140   GOTO 170
150   PRINT "THAT IS CORRECT ";N$:C=C+1
170   PRINT:PRINT:INPUT "HOW ABOUT ANOTHER TRY ";A$
180   IF A$="YES" GOSUB 1000:GOTO 70
190   IF A$="NO" GOTO 220
200   INPUT " I AM SORRY DID YOU SAY YES OR NO ";A$
210   GOTO 180
220   PRINT "YOU GOT ";C;" OUT OF ";T;" CORRECT THIS SESSION"
230   P=C/T:PRINT
240   IF P>=.9 PRINT"VERY GOOD -- YOU UNDERSTAND !!!!!":END
250   IF P>=.8 PRINT "GOOD -- KEEP IT UP ";N$:END
260   IF P>=.7 PRINT N$;" THAT IS ABOUT AVERAGE":END
270   IF P>=.6 PRINT "THAT IS A POOR SHOWING ";N$:END
280   PRINT N$;" YOU NEED TO NOTIFY YOUR TEACHER "
290   PRINT "THAT YOU DO NOT UNDERSTAND FUNCTION EVALUATIONS"
300   END
1000  PRINT:PRINT "              2"
1010  PRINT "   F(X) = 4X + 3X -10"
1020  RETURN
```

Figure 6. Modified Program From Figure 5.

```
1     ' ***** FUNCTION EVALUATION *****
2     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
3     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D.
4     ' PROGRAM-ID: FUNCT2            LAST-MODIFIED: MAY 23, 1977
10    PRINT "HELLO, I AM THE TRS-80 MICROCOMPUTER SYSTEM"
20    INPUT "WHO ARE YOU ";N$
30    PRINT:PRINT "IT IS NICE TO KNOW YOU ";N$
40    PRINT "TODAY WE ARE GOING TO PRACTICE THE"
50    PRINT "FUNCTION EVALUATION OF"
60    GOSUB 1000:C=0:T=0
70    RANDOM:A=RND(10):T=T+1
80    F= -2*A*A*A+10*A*A-1
90    PRINT:PRINT  "WHAT IS F(";A;") = ";:INPUT G
100   IF F = G GOTO 150
110   PRINT "NO! ";N$;" F(";A;") = ";F
120   PRINT "REMEMBER THE VARIABLE X IS A PLACE HOLDER"
130   PRINT " AND WE REPLACE IT WITH ";A
140   GOTO 170
150   PRINT "THAT IS CORRECT ";N$:C=C+1
170   PRINT:PRINT:INPUT "HOW ABOUT ANOTHER TRY ";A$
180   IF A$="YES" GOSUB 1000:GOTO 70
190   IF A$="NO" GOTO 220
200   INPUT " I AM SORRY DID YOU SAY YES OR NO ";A$
210   GOTO 180
220   PRINT "YOU GOT ";C;" OUT OF ";T;" CORRECT THIS SESSION"
230   P=C/T:PRINT
240   IF P>=.9 PRINT"VERY GOOD -- YOU UNDERSTAND !!!!!":END
250   IF P>=.8 PRINT "GOOD -- KEEP IT UP ";N$:END
260   IF P>=.7 PRINT N$;" THAT IS ABOUT AVERAGE":END
270   IF P>=.6 PRINT "THAT IS A POOR SHOWING ";N$:END
280   PRINT N$;" YOU NEED TO NOTIFY YOUR TEACHER "
290   PRINT "THAT YOU DO NOT UNDERSTAND FUNCTION EVALUATIONS"
300   END
1000  PRINT:PRINT "              3        2"
1010  PRINT "   F(X) = - 2 X  + 10 X  - 1"
1020  RETURN
```

Figure 7. Modified Program From Figures 5 or 6.

```
1     ' ***** FUNCTION EVALUATION *****
2     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
3     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D.
4     ' PROGRAM-ID: FUNCT3            LAST-MODIFIED: MAY 23, 1977
10    PRINT "HELLO, I AM THE TRS-80 MICROCOMPUTER SYSTEM"
20    INPUT "WHO ARE YOU ";N$
30    PRINT:PRINT "IT IS NICE TO KNOW YOU ";N$
40    PRINT "TODAY WE ARE GOING TO PRACTICE THE"
50    PRINT "FUNCTION EVALUATION OF"
60    GOSUB 1000:C=0:T=0
70    RANDOM:A=RND(10):T=T+1
80    F = A-1
90    PRINT:PRINT  "WHAT IS F(";A;") = ";:INPUT G
100   IF F = G GOTO 150
110   PRINT "NO! ";N$;" F(";A;") = ";F
120   PRINT "REMEMBER THE VARIABLE X IS A PLACE HOLDER"
130   PRINT " AND WE REPLACE IT WITH ";A
140   GOTO 170
150   PRINT "THAT IS CORRECT ";N$:C=C+1
170   PRINT:PRINT:INPUT "HOW ABOUT ANOTHER TRY ";A$
180   IF A$="YES" GOSUB 1000:GOTO 70
190   IF A$="NO" GOTO 220
200   INPUT " I AM SORRY DID YOU SAY YES OR NO ";A$
210   GOTO 180
220   PRINT "YOU GOT ";C;" OUT OF ";T;" CORRECT THIS SESSION"
230   P=C/T:PRINT
240   IF P>=.9 PRINT"VERY GOOD -- YOU UNDERSTAND !!!!!":END
250   IF P>=.8 PRINT "GOOD -- KEEP IT UP ";N$:END
260   IF P>=.7 PRINT N$;" THAT IS ABOUT AVERAGE":END
270   IF P>=.6 PRINT "THAT IS A POOR SHOWING ";N$:END
280   PRINT N$;" YOU NEED TO NOTIFY YOUR TEACHER "
290   PRINT "THAT YOU DO NOT UNDERSTAND FUNCTION EVALUATIONS"
300   END
1000  PRINT:PRINT "                         2"
1010  PRINT "                     2 X  - X - 1"
1020  PRINT "           F(X) = ------------------"
1030  PRINT "                        2 X + 1"
1040  RETURN
```


Figure 8. Solving Quadratic Equations Using the Factoring Method.

```
1     ' ***** SOLVING QUADRATIC EQUATIONS *****
2     ' ***    USING THE FACTORING METHOD   ***
3     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
4     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D
5     ' PROGRAM-ID: FACTOR            LAST-MODIFIED: MAY 23, 1977
10    PRINT "HELLO I AM THE TRS-80 MICROCOMPUTER SYSTEM"
20    INPUT "WHO ARE YOU "; N$
30    PRINT:PRINT "IT IS NICE TO KNOW YOU ";N$
40    PRINT "TODAY  WE ARE GOING TO PRACTICE SOLVING"
50    PRINT "QUADRACTIC EQUATIONS OF THE FORM"
60    PRINT:PRINT "      2"
70    PRINT "      X  + B X  + C = 0":PRINT "USING THE FACTORING
                                                          METHOD"
80    C=0:T=0
90    RANDOM:G=RND(4):T=T+1:A=RND(10):RANDOM:B=RND(10)
100   ON G GOTO 110,210,300,500
110   PRINT "IF        2"
120   PRINT "        X  + ";A+B;" X  + ";A*B;" = 0"
130   PRINT:INPUT "WHAT ARE THE TWO VALUES OF X ";AA,BB
140   IF ((AA=-A) AND (BB=-B)) OR ((AA=-B) AND (BB=-A)) GOTO 200
150   PRINT "NO! ";N$;" X = ";-A;" AND ";-B
160   PRINT:PRINT:INPUT "HOW ABOUT ANOTHER TRY ";A$
170   IF A$ = "YES" GOTO 90
180   IF A$ = "NO" GOTO 400
190   INPUT "I AM SORRY, DID YOU SAY YES OR NO ";A$:GOTO170
200   PRINT "THAT IS CORRECT ";N$:C=C+1:GOTO 160
210   PRINT "IF        2"
220   PRINT "        X  - ";A+B;" X  + ";A*B;" = 0"
230   PRINT:INPUT "WHAT ARE THE TWO VALUES OF X ";AA,BB
240   IF ((AA=A) AND (BB=B)) OR ((AA=B) AND (BB=A)) GOTO200
250   PRINT "NO! ";N$;" X = ";A;" AND ";B:GOTO 160
300   IF A < B THEN  S = A : A = B : B = S
301   PRINT "IF        2"
310   PRINT "        X  - ";A-B;" X  - ";A*B;" = 0"
320   PRINT:INPUT "WHAT ARE THE TWO VALUES OF X ";AA,BB
330   IF ((AA=A) AND (BB=-B)) OR ((AA=-B) AND (BB=A)) GOTO 200
340   PRINT "NO! ";N$;" X = ";A;" AND ";-B:GOTO160
400   PRINT "YOU GOT ";C;" OUT OF ";T;" CORRECT THIS SESSION"
410   P=C/T:PRINT
420   IF P>= .9 PRINT "VERY GOOD -- YOU UNDERSTAND !!!":END
430   IF P>= .8 PRINT "GOOD -- KEEP  AT IT ";N$:END
440   IF P>= .7 PRINT N$;" THAT IS ABOUT AVERAGE":END
450   IF P>= .6 PRINT "THAT IS POOR ";N$:END
460   PRINT N$;" YOU NEED TO NOTIFY YOUR TEACHER"
470   PRINT "THAT YOU DO NOT UNDERSTAND FACTORING"
480   END
500   IF A > B THEN S = A : A = B : B = S
501   PRINT "IF        2"
510   PRINT "        X  + ";B-A;" X  - ";A*B;" = 0"
520   PRINT:INPUT "WHAT ARE THE TWO VALUES OF X ";AA,BB
530   IF ((AA=A) AND (BB=-B)) OR ((AA=-B) AND (BB=A)) GOTO 200
540   PRINT "NO! ";N$;" X = ";A;" AND ";-B:GOTO 160
```

Figure 9. Solving Quadratic Equations Using the Quadratic Formula.

```
1     ' ***** SOLVING QUADRATIC EQUATIONS *****
2     ' ******** QUADRATIC FORMULA ************
3     ' NSF MATHEMATICS TEACHER DEVELOPMENT PROGRAM
4     ' PROGRAMMER: ANTONIO M. LOPEZ, JR. PH.D.
5     ' PROGRAM-ID: QUADRA            LAST-MODIFIED: MAY 23, 1977
10    :PRINT"HELLO, I AM THE TRS-80 MICROCOMPUTER."
20    INPUT"WHO ARE YOU ";N$
30    PRINT:PRINT"IT IS NICE TO KNOW YOU ";N$
40    PRINT"TODAY WE ARE GOING TO PRACTICE USING THE"
50    PRINT"QUADRATIC FORMULA TO EVALUATE EQUATIONS OF THE FORM"
70    PRINT"             2"
80    PRINT"           A X  + B X  + C = 0":PRINT:CC=0:T=0
90    A=RND(10):B=RND(10):C=RND(10)
100   D=B*B-4*A*C
110   IF D < 0 GOTO 90
111   T=T+1
120   G=(-B+SQR(D))/(2*A)
130   H=(-B-SQR(D))/(2*A)
140   PRINT"IF               2"
150   PRINT"          ";A;" X  + ";B;" X + ";C;" = 0"
155   PRINT:INPUT"WHAT ARE THE TWO VALUES OF X ";AA,BB
160   IF((ABS(AA-G) < .01) AND (ABS(BB-H) < .01)) OR ((ABS(AA-H) <
                                 .01) AND (ABS(BB-G) < .01)) GOTO 230
180   PRINT"NO! ";N$;"  X = ";INT(G*100)/100;" AND ";INT(H*100)/100
190   PRINT:INPUT"HOW ABOUT ANOTHER TRY ";A$
200   IF A$ = "YES" GOTO 90
210   IF A$ = "NO" GOTO 250
220   INPUT"I AM SORRY, DID YOU SAY YES OR NO ";A$:GOTO 200
230   PRINT "THAT IS CORRECT ";N$:CC=CC+1:GOTO 190
250   PRINT "YOU GOT ";C;" OUT OF ";T;" CORRECT THIS SESSION."
260   P=CC/T:PRINT
270   IF P >= .9 PRINT "VERY GOOD, ";N$:END
280   IF P >= .8 PRINT "GOOD -- KEEP IT UP ";N$:END
290   IF P >= .7 PRINT "THAT'S AVERAGE ";N$:END
300   IF P >= .6 PRINT "THAT'S A POOR SHOWING ";N$:PRINT "YOU NEED
                                          LOTS OF PRACTICE":END
310   PRINT N$;" YOU NEED TO NOTIFY YOUR TEACHER THAT YOU"
320   PRINT "DO NOT UNDERSTAND THE QUADRATIC FORMULA."
330   END
```

of the arguments. Furthermore, we can now write any number of function evaluation programs by modifying this last program. For example, if we want our students to practice with the function the teacher how to write quick and easy programs about the subject being taught and you have the best CAI programs for that course. For repetitive drill, the computer has no equal. As for testing, the teacher can now assume the role of a consultant and supporter; the test questions are being asked by a machine not the teacher.

The pseudo-random number generator in the microcomputer is essential for CAI models. Unfortunately, all three microcomputer systems have different arguments on their RND instruction. Our examples in this paper are run on the TRS-80. However, they can be modified slightly for execution on the APPLE II or PET.

In Track I, we develop a CAI program to assist students in learing how to evaluate functions. The first program (see $f(x) = -2x^3 + 10x^2 - 1$, we simply change line 80 and the display subroutine (lines 1000, 1010, 1020). Thus, we have a new program (see Figure 6). We can even handle certain quotients with this base CAI program. Figure 7 uses the function $f(x) = (2x^2 - x - 1)/(2x + 1)$ and again we modify only line 80 and the display subroutine.

In Track II, for example, we are interested in making sure that high school students can solve quadratic equations of the form $x^2 + bx + c = 0$ and $ax^2 + bx + c = 0$. We develop CAI programs for both these problems using the factoring method and the quadratic formula. Figure 8 is the program that drills the student in the solution of four types of factoring problems:

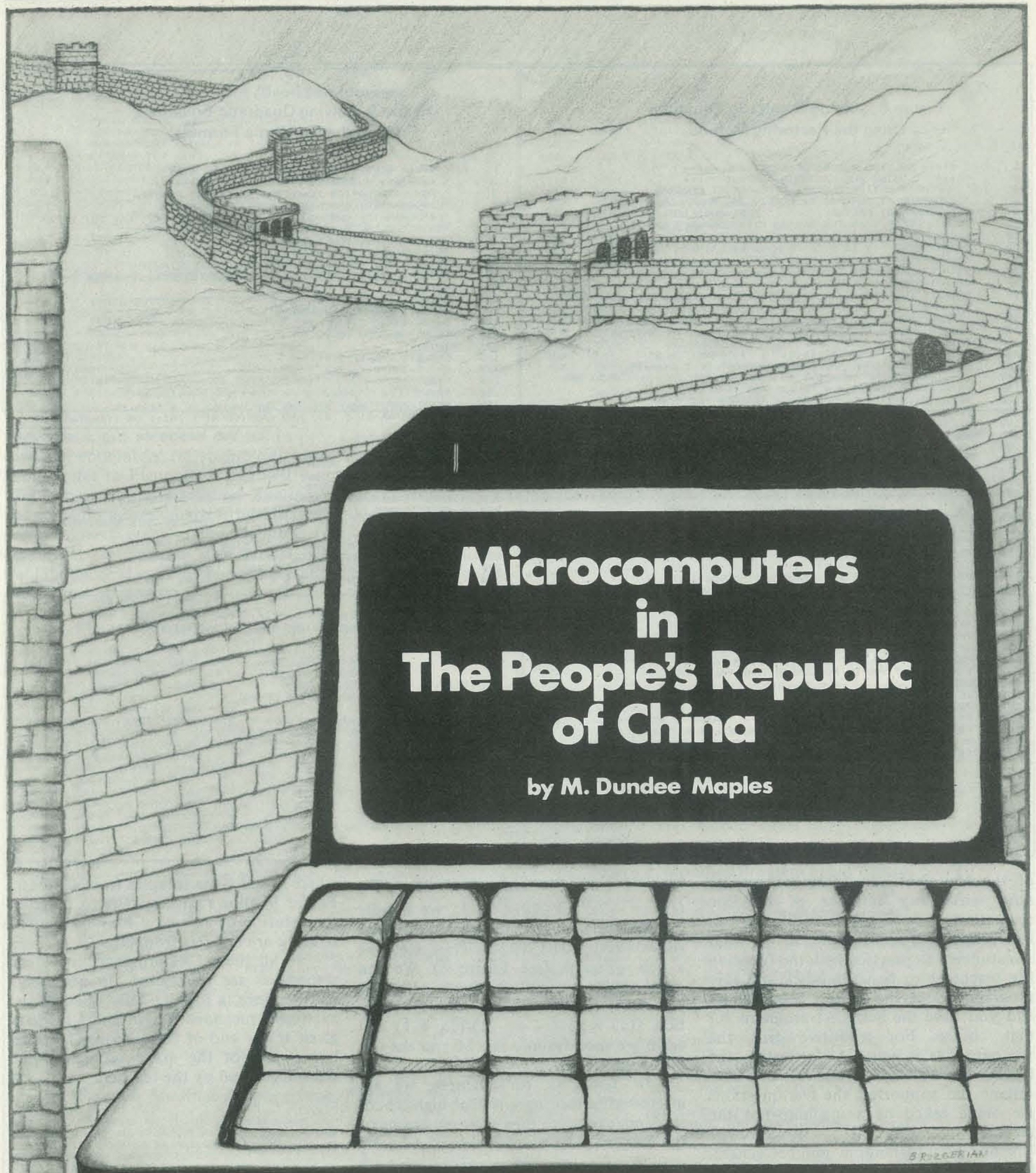
$$x^2 + bx + c = 0$$
$$x^2 + bx - c = 0$$
$$x^2 - bx + c = 0$$
$$x^2 - bx - c = 0$$

where b and c are integers from 0 to 10. Figure 9 allows the student to practice the solution of $ax^2 + bx + c = 0$ where a, b, and c are integers from 0 to 10.

In all these CAI programs, incorrect responses are corrected immediately. Furthermore, a running tally of correctly answered questions is kept and a grade is given at the end of the session. These are but a few of the many programs that were developed by the teachers. ■

References

1. Bell, Fred "Classroom Computers: Beyond the 3 R's," *Creative Computing* 5,9 (Sept 1979), 68-70.
2. Lopez, Jr. Antonio M., "Night School," *80-Microcomputing* (Jan 1980), 41-47.
3. McLean, R.T.; Lopez, Jr. A.M.; Schmit, J. and Griffin, D., "Master BASIC Command List," *Recreational Computing* (Sept-Oct 1979), 51-53.



Microcomputers in The People's Republic of China

by M. Dundee Maples

In October 1978, when I first visited the People's Republic of China with the IEEE Computer Society's delegation, there were very few microcomputers in China. We only saw one U.S. microcomputer system, a Cromemco. It was sitting on a bench-top in a computing center of a university in Peking. It hadn't even been powered up yet. The Chinese themselves had just started into the L.S.I. chip production world. We saw some 256 x 4-bit EPROMs, 4-bit microcomputers, 1K x 1 dynamic RAMs, and 256 x 1 static RAMs at the Shanghai technology fair. None of these devices were in production quantities, but rather were prototypes. The quality, reliability, and software of the Chinese computer was extremely poor and backward as compared to our standards. That is why the Chinese have spent so much money buying Western technology in the last two years.

On my latest visit to China from September 19, 1980 to October 19, 1980, I saw a great change both socially and technologically. The people are more open, friendly and, dare I say, more "westernized." The roads that were covered with bicycles and people in 1978 are covered with people, bicycles, automobiles, buses and trucks in 1980.

Where I saw one Cromemco in 1978, I saw many Cromemcos, DEC PDP 11/03s, Rockwell AIM 65s, Intel SDK 85s, NEC TK80As, Atari microcomputers, Intel MDS systems, Heath 6800 trainer kits, Dynabytes, Apple computers, Exidy, Radio Shack TRS-80s and more.

China has spent a tremendous amount of money on computers of all kinds, not only microcomputers but minis and maxis as well. They have purchased PDP 11/45s, over 20 DEC PDP 11/70s, Univac 1100/12 and 1100/11s, Hitachi M-160-11 and M-170, IBM 4341 and 4331 large mainframes, CDC Cyber 175, 173, 172s, Burroughs 6810, 3450s and 3E1342s distributed control systems like Fisher Controls Provox systems and several of Honeywell's TDC 2000 systems.

When the Chinese purchase large computer systems, they stipulate that training for their engineers be included in the contract. The training of Chinese in the modern use of computer systems is of critical importance to China. This is another reason for the massive growth of microcomputer systems in China.

IC technology is still very weak in China. They have sample quantities of the 8080 (fabricated in Shanghai), 4 x 1 dynamic memories, and 1K x 1 static memories. Quality control is one of their major problems. They hope to learn how to solve many of their IC fabrication problems from IC manufacturers that they can entice to build factories in China (like Solid State Scientific, who has contracted to set up a $10 million CMOS facility).

They need this improved technical knowledge to improve the computation ability of their larger computer systems (see Figure 1).

Microcomputer systems are being employed in the following areas:

- Automation
- Instrumentation
- Control and Data Acquisition
- Education (University)
- Education (Factory personnel – engineers, etc.)
- Software Development
  - Language
  - File Management Systems
  - Data Management Systems
  - Simulation
- Hardware Development
  - Instrumentation
  - Intelligent Interfaces
  - Dedicated Controllers
  - General Purpose Systems
- Machine Control
- Word Processing and Office Management
- Research and Forecasting

A Chinese 16-bit minicomputer (left). A Chinese designed and built light pen graphics terminal (right).

M. Dundee Maples

Virtually every segment of China's vast bureaucracy has jumped on board the microcomputer bandwagon. As such, microcomputers can be found everywhere in China. The low purchase price of these microcomputers coupled with China's inexperience and gigantic bureaucracy have placed some of these microcomputers into some very inappropriate areas (i.e., large data management systems being developed for Z80 floppy disk computer systems).

It is obvious that China is on the beginning of a learning curve as regards the uses and capabilities of computer systems. The biggest usage of microcomputers in China today is actually for training and educating both the University students and the practicing engineers and scientists who must learn how to select and apply computers effectively and efficiently. I say this because the vast majority of engineers and scientists have little or no computer background. When asked to select or recommend a microcomputer for their facility, they are completely overwhelmed by the sales presentations.

Because the majority of microcomputers have been delivered within the last eight months (January/February 1980 to September 1980), the Chinese have as yet very little experience using these systems. As they get closer to implementing these systems into actual designs, they will become aware of the problems, shortcomings and general difficulties of these systems. Of course, they will also learn about the beneficial capabilities of these microsystems, but because the Chinese have such high expectations, their initial experience will be very frustrating.

The Chinese are only just beginning to use microcomputers; their expectations are high, their time frame for modernization is short, their experience is poor, their technical management is weak, and their bureaucracy is massive. These conditions will produce a very frustrating and difficult time for the Chinese for the next five years, but there is no easy way to make the transition from a manual to a computerized society. ■

M. DUNDEE MAPLES is a systems engineer with M & E Associates in Cupertino, CA. He has designed, built and implemented microcomputer-based systems for instrumentation and automation equipment. He has also worked on developing floating-point mathematical packages, BASIC interpreters, List interpreters and general utility routines.

Figure 1. China's Present Large Mainframe Systems.

HDS 9	comparable to	CDC 6600
DJS 260	comparable to	IBM 360/75
DJS 240	comparable to	IBM 360/65
DJS 200	comparable to	IBM 360/50

Hardware

The Peripherals Plus AM-II, a printed circuit board that plugs into an Apple II computer, plus software, turns an Apple into a simple-to-use nine-voice music synthesizer for $198.

Like the earlier ALF/Apple Music synthesizer, the AM-II allows users to compose music with two game paddles (instead of complicated keyboard commands), along with a graphic display of notes and the music staff. Users can select notes on the staff (from a six octave range), and duration and many other characteristics from a menu at the bottom of the screen. During playback of the music, the music is shown on an animated, 16-color display.

The AM-II is available in cassette or disk versions for $198 from Peripherals Plus, 119 Maple Ave., Morristown, NJ 07960. Telephone: toll-free (800) 631-8112. In New Jersey, (201) 267-4558. A 16K Apple II or Apple II Plus is required.

New from Connecticut microComputer is this interface for PET microcomputers. Dubbed SADI, the two-way RS-232 and parallel output interface allows the connection of a variety of computer peripherals and the transfer of programs between Commodore PET computers.

Each SADI interface comes complete with power supply, PET IEEE cable, RS-232 connector, parallel port connector, and case. Fully assembled and tested, the 110 VAC SADI carries a suggested retail price of $295, with 230 VAC version offered at $325.

For further information, write: SADI SADI, Connecticut microComputer, Inc., 34 Del Mar Drive, Brookfield, CT 96804.

New SDS Word Processor

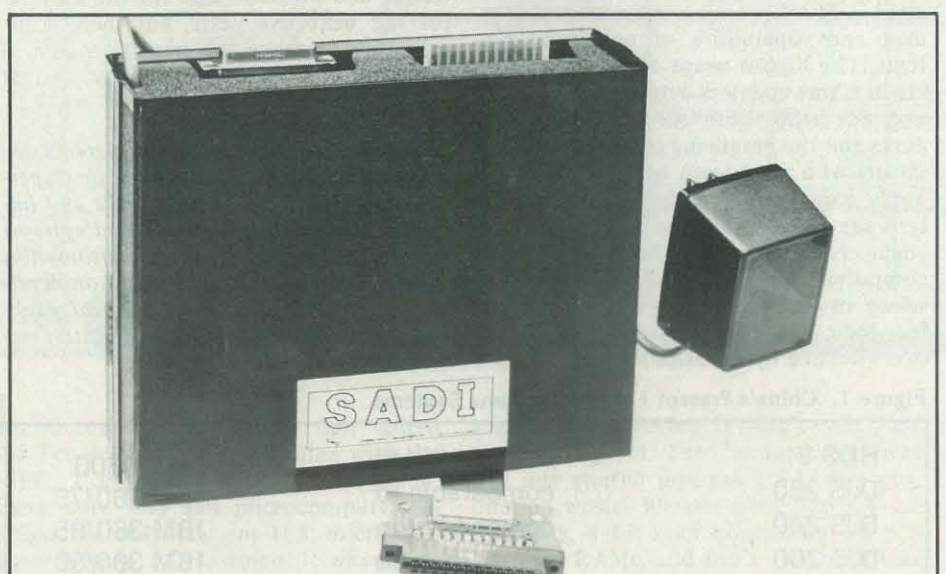
New SDS Word Processor Simplifies Operator Use. SDS Business Computers has announced a new, improved word processing system containing additional features and simplifying operator use. The new SDS 420 features format-to-screen layout and a function-oriented keyboard.

Format-to-screen layout allows the operator to see text on the video display before committing it to paper. This permits on-screen editing of the text, including the setting of margins and tabs, word insertion/deletion and paragraph realignment.

Function-oriented command keys on the new keyboard eliminate the need for the operator to learn complicated commands, by requiring only a single keystroke and specifying the command on the key.

The system is designed to permit immediate access to all files. Storage capacity is 200 pages per disk, with an unlimited number of disks. The Diablo 630 printer, which utilizes both plastic and metal print wheels, is standard with the SDS word processor. Other SDS software packages available include medical, legal and financial applications.

For more information, contact SDS, 344 Main Street, Venice, California 90291. (213) 390-8676.



Software

Heath Company has announced a major program to provide software programs for the microcomputer market. SOFTSTUFF™ is designed for computer users who want to expand the capabilities of their computers for home and business use.

The first SOFTSTUFF™ applications programs include a full screen editor and an improved text formatter; a file transfer utility with on-line access to MicroNET™ Information Services available; education packages; entertainment programs; and problem-solving small business software, including office management, word processing, telecommunications and office communications.

Among the initial SOFTSTUFF™ offerings will be an enhanced programming language, C BASIC™. Initial SOFTSTUFF™ programs operate under the Heath Disk Operating System (HDOS). Programs operating under CP/M will be introduced by year end.

For more details and price information on SOFTSTUFF™ programs, write for a FREE catalog to: Heath Company, Dept. 350-520, Benton Harbor, MI 49022. In Canada, write: Heath Company, 1480 Dundas Hwy. East, Mississauga, Ontario, Canada L4X 2R7.

Professional quality business and educational software for the Heath-Zenith H8, H89 and Z89 microcompuer systems is now available from XtraSoft software development company.

Small business systems available and under development include inventory control, point of sale, mailing list and a complete accounting package.

For Heath-Zenith owners interested in C.A.I., systems available and under development include Vocabulary Builder I, II, and III, U.S. Presidents, Chemical Elements, Number Bases and more.

For a complete catalog, contact: XtraSoft Company, P.O. Box 91063, Louisville, KY 40291.

Madison Computer Announces "McTERM," a package to turn your Commodore computer into a very intelligent terminal. The McTERM package allows the Commodore computer to become a terminal for talking to other computers by direct connect or via the modem. Your computer can speak with mainframes or other micros. The package allows the user to set the proper protocols, *transfer entire disk files* with or without CRC checking, send or receive program, WordPro (a trademark of Professional Software), sequential, or relative files.

The package comes complete with program, ROM, cable, and a complete manual. *Documentation* is included to allow the user to incorporate the machine language routine for further applications.

The McTERM package allows for the easy *networking* of the Commodore computers for *telecommunications* for personal or business uses.

The package will operate on 40 or 80 columns, 2040 or 8050 disk drives (new ROMs only), 3.0 or 4.0 BASIC, and 1.0 or 2.0 DOS. For information, contact Madison Computer, 1825 Monroe St., Madison, Wisconsin 53711, telephone (608) 255-5552.

Key Bits Inc., has announced an automated spelling dictionary system called WordSearch™. WordSearch is easy to learn and use for searching documents, letters, manuals or any text material for the occurrence of words that have not previously been validated and placed in the main word library. These "unknown" words are identified by WordSearch as both a list of words or in context of the original text. Words found not to be in the library by WordSearch, but identified by you to be valid, are then added to the library at your command.

WordSearch is distributed on an 8-inch single density diskette with a complete user manual, an initial spelling word dictionary, and a demonstration package for $195.00. Additional information on WordSearch and the one time introductory offer can be obtained by contacting Key Bits Inc., P.O. Box 592293, Miami, FL 33159.

A comprehensive software package that teaches about and aids in the design of solar energy systems is now available from Peripherals Plus.

To choose a solar energy system, one must consider heating capacity, heat delivery, environmental factors, and economic factors. The F-Chart thermal analysis procedure was developed in 1977 at the Solar Energy Laboratory of the University of Wisconsin by Beckman, Klein and Duffie to predict the percentage of annual heating load provided by a solar system. This is now the de facto standard of the American Society of Heating, Refrigeration and Air Conditioning Engineers.

The computer software system consists of three programs: Thermal Analysis, Economic Analysis and Builder Program to create climactic data tapes. A major advantage of the Thermal Analysis program is that it allows multiple design options to be examined quickly and easily. The Economic Analysis determines the cost effectiveness of the solar system on a life cycle basis. The Builder Program lets the user create a data tape for a specific geographic location.

Available for the TRS-80 on cassette (CS-3307, $49.95, requires 16K) or floppy disk (CS-3802, $99.95, requires 32K). Contact Peripherals Plus, 119 Maple Ave., Morristown, NJ 07960. Or call (210) 267-4558.

Free Catalog of Software and Peripherals for the Apple II. A new catalog has just been released by Rainbow Computing, Inc., representing the largest collection of software and accessories for the Apple Computer ever assembled.

Included in Rainbow's Catalog are hundreds of programs of all types: business applications, education and science, games, languages, demos, and utilities. Also described are a wide assortment of accessories such as music and speech synthesis, appliance control, other interface cards, add-on memory, printer, and manuals.

This free catalog is available now. Just write or phone Ranibow Computing, 9719 Reseda Blvd., Northridge, CA 91324, (213) 349-5560.

New "WHATSIT?" fits Apple II Plus. "WHATSIT?" (tm) is now available in a model for the Apple II Plus computer, according to Computer Headware of San Francisco.

Described as a self-indexing, cross-referencing file system, WHATSIT has been available since 1978 in versions for CP/M and North Star computers, as well as the original Apple II. The announcement of the new Model A-2 Plus means that WHATSIT can also run on the Apple II Plus with standard Applesoft ROM.

By cross-referencing data entries in disc storage, WHATSIT is able to answer direct questions, phrased in conversational "pidgin English." The name stands for "Wow! How'd All That Stuff get In There?"

WHATSIT responds to both queries and file updates at conversational speed . . . even in files containing hundreds or thousands of entries. Typical response time is 2 to 10 seconds, the firm claims. Storage capacity is up to 2000 entries using a 5-inch floppy disc, or up to 25,000 entries using 8-inch CP/M format discs.

Apple models of WHATSIT are priced at $150. The new Model A-2 Plus requires a disc-oriented Integer BASIC, available separately. WHATSIT is sold through computer retail dealers, and distributed nationally by Hardhat Software, Box 14815, San Francisco, CA 94114; telephone (415) 621-2106.

Education

Educational Software Exchange Library (E.D.S.E.L.) plans to assemble and maintain a computer software library of pre-college educational materials that would be easily available, inexpensive and widespread.

E.D.S.E.L. is requesting program listings or cassette copies of any software directly relating to education using the following hardware: Apple, Exidy Sorceror, Atari 400 and 800, PET Commodore, TRS-80 (Levels I and II), Texas Instruments 9914, DEC PDP/11, Hewlett Packard 2000, and Univac 1110.

The success of this project depends upon direct response and software contributions. E.D.S.E.L. c/o Stanford Avenue School, 2833 Illinois Avenue, South Gate, CA 90281.

Conduit, a nonprofit organization that reviews, tests and distributes educational software, began distributing programs for the microcomputer, including the PET, TRS-80 and Apple, about one year ago.

Based at the University of Iowa and partially funded by grants from the National Science Foundation and the Fund for the Improvement of Postsecondary Education, Conduit initially geared their materials for larger computer systems. Now they are looking at other microcomputers that seem to have a chance of gaining in popularity with the schools.

Out of about 80 programs, Conduit offers about a dozen packages for the microcomputer. For information write P.O. Box 388, Iowa City, Iowa 52244, (319) 353-5789.

"Funding Report Microcomputers," a handbook designed to assist educators in locating sources of financial support and in selecting and using micros, has been released by Bell and Howell.

The 44-page report, which is based on interviews with federal officials, points out programs that educators can tap into for funding. Other chapters in the book explain how state, district and local level educational budgets are constructed.

Bell and Howell Co., Audio-Visual Products Division, 7100 N. McCormick Road, Chicago, Illinois 60645, (312) 262-1600.

Computer Technology Holds Great Potential for Education. The University of Southern California's instructional technology department, with funding from Control Data Corp., is currently developing CDC's Plato system of computerized instruction.

According to Dr. Fredrick G. Knirk, department chairman, the system is already being used in industry, where it is reducing instruction time by approximately 30%. Reduced instruction time converts directly to reduced cost and increased productivity for each person trained – a highly attractive aspect for business and industry.

The Plato system operates on a time-sharing basis, with computer terminals in offices, homes and CDC centers across the country. The terminals are tied into master computers with the instructional program, permitting students to access the lessons whenever it is convenient for them.

The major emphasis is on the design of an overall instruction package that will be comprehensive for any student at any level of competence. The program accounts for all conceivable areas of difficulty and any probable questions.

The instructor evaluates the individual student's level of competency and selects the areas of the total program that will best suit his specific needs. Thus the student only learns what he does not already know, without spending time relearning subjects he does.

There are provisions for communication between student and instructor; and the terminal instruction is supplemented by printed texts to further reduce costs.

Knirk notes that competency-based instruction will free teachers from many traditional tasks, enabling them to analyze student needs, answer difficult questions, develop programs and still be classroom instructors for specialized subjects that do not lend themselves to computers.

T.E.S.T., an aid for the classroom teacher is now available from TYC^{tm} SOFTWARE. T.E.S.T. contains two programs; a MAINTENANCE PROGRAM, and a TEST AND DRILL PROGRAM.

The MAINTENANCE PROGRAM allows the user to create a test of up to 35 questions. To produce a test, a question is typed on any topic (up to 240 characters), the type of question is entered (True or False, Multiple Choice, or Completion), and the correct answer is given. When finished, the test is saved on cassette for future use.

TEST AND DRILL is a utility program designed to accept the test prepared by the MAINTENANCE PROGRAM. With the TEST AND DRILL PROGRAM students can either use the questions as a review, take a scored test, or the teacher can have the computer prepare a printed test or worksheet with answer key.

The two programs and a complete manual are for the TRS-80 Level II, 16K, and cost $11.95.

For more information on educational software contact: TYC^{tm} SOFTWARE, 40 Stuyvesant Manor, Geneseo, New York 14454.

Charles Mann & Associates, Micro Software Division, has announced the exclusive distribution of a new administrative package for educators called "The Counsellor's Program." The program package, which operates in conjunction with CMA's The Grading System Programs, allows for the preparation of the school guidance counsellor's master student records and file folder labels.

The Counsellor's Program costs $89.95. Additional information is available from Charles Mann & Associates, Micro Software Division, 7594 San Remo Trail, Yucca Valley, CA 92284. Phone (714) 365-9718.

High Technology, Inc. is now offering CHEMISTRY LAB SIMULATIONS # 1 in a version for the Atari 800 computer (a previous version runs on the Apple II). Developed by Dr. John I. Gelder, an Oklahoma State University professor, the program provides dynamic and colorful simulations of high school and college level chemistry lab experiments, allowing the student to interactively discover the chemistry principles involved. The simulations are equally suitable for use as a lecture aid.

The suggested retail price is $100.00. Contact High Technology, Inc., 8001 No. Classen Blvd., P.O. Box 14665, Oklahoma City, OK 73113.

CompuSoCo has announced the availability of a new school administrative package for the Apple II and Apple Plus computer. The system contains four modules, all available separately, which allow for teacher compilation of grades from class assignments and tests, the input and preparation of grades to print report cards and maintain student records, the preparation of reports to guidance counsellors for class scheduling purposes and the preparation of master school schedules and individual student schedules.

Information is available from CompuSoCo, 26251 Via Roble, P.O. Box 2325, Mission Viejo, CA 92690.

Monument Computer Service has announced the release of a new software applications package for the Apple Computer. The "Assistant Principal" is a complete administrative package for high school and junior high schools. The package provides total control of class rosters, student master records, student schedules, teacher assignments, and grade reporting.

The system requires two disk drives and Applesoft in ROM. The system is provided on seven diskettes with a two volume operating manual for $500.00. The operating manuals are available separately at $50.00. Additional information is available from Monument Computer Service, Village Data Center, P.O. Box 603, Joshua Tree, CA 92284. The educational development group order operator can be reached at (800) 854-0561, Ext. 802. In CA call (800) 432-7257.

Edu-Ware services announces the immediate release of two new products, The Prisoner, and the Compu-Spell System. Both products require an Apple with 48K and a disk drive.

The Prisoner was inspired by the popular television series of the 1960's. Consisting of twenty interlinked games on one diskette, the program places the player on an island housing a psychological prison camp. His task is to escape both the island and its attempts to extract information from him. The program lists for $29.95.

Compu-Spell is a complete instructional system in spelling. Utilizing a totally new instructional algorithm, the system teaches spelling skills for grades 4 through 8 and has an additional unit tailored for the adult user. The program features upper and lower case word display, and a complete file management system for monitoring progress of all users. Cost is $39.95 for a program diskette and one data diskette. Additional data diskettes are $19.95.

For further information, contact Edu-Ware Services, Inc., 22035 Burbank Blvd., # 223, Woodland Hills, CA 91367 or phone (213) 346-6783.

Games

Today's Hottest electronic game appears to be Atari's Space Invaders, a zap-the-alien-space-ships game that first gained popularity in local arcades.

Invented in Japan, the game has been well received in the United States, as seen by the nation-wide contests and the television news coverage. Atari's cartridge costs about $20.

Entex has recently introduced a handheld unit of Space Invaders that retails for $50. For information write Entex, 303 W. Artesia, Compton, CA 90220.

Acorn Software Products, Inc. announces the debut of DUEL-N-DROIDS, a new sound and graphics game program by Leo Christopherson, for the Model I Level II TRS-80. The program features two androids that square off against each other with swords in both "Practice" and "Tournament" duels.

DUEL-N-DROIDS is priced at $14.95 on cassette or $20.95 on diskette. Dealer inquiries should be directed to: Acorn Software Products, Inc., 634 North Carolina Ave., SE, Washington, DC 20003 or (phone) (202) 544-4259.

Microsoft Consumer Products announces Olympic Decathlon, a skill game for personal computers based on the Decathlon athletic competition.

Just like the real competition, Olympic Decathlon encompasses ten events: 100-meter dash, long jump, pole vault, discus throw, shot put, 400-meter dash, 110-meter hurdles, 1500-meter run, high jump and javelin throw. Players are awarded points for each event based on how their performance matches up to world champion standards. The competitor who earns the most points in the majority of events wins the gold medal. From one to eight players may compete.

Olympic Decathlon is available on either cassette or diskette for the TRS-80 Model I. The cassette version requires a Level I or Level II system with 16K memory. The disk version requires a Level II, 32K, disk-based system. A disk version for the Apple II computer will be introduced in the fourth quarter of 1980. Suggested retail price for all versions is $24.95.

For information, contact Microsoft Consumer Products, 400 108th Ave. NE, Suite 200, Bellevue, WA 98004, (206) 454-1315.

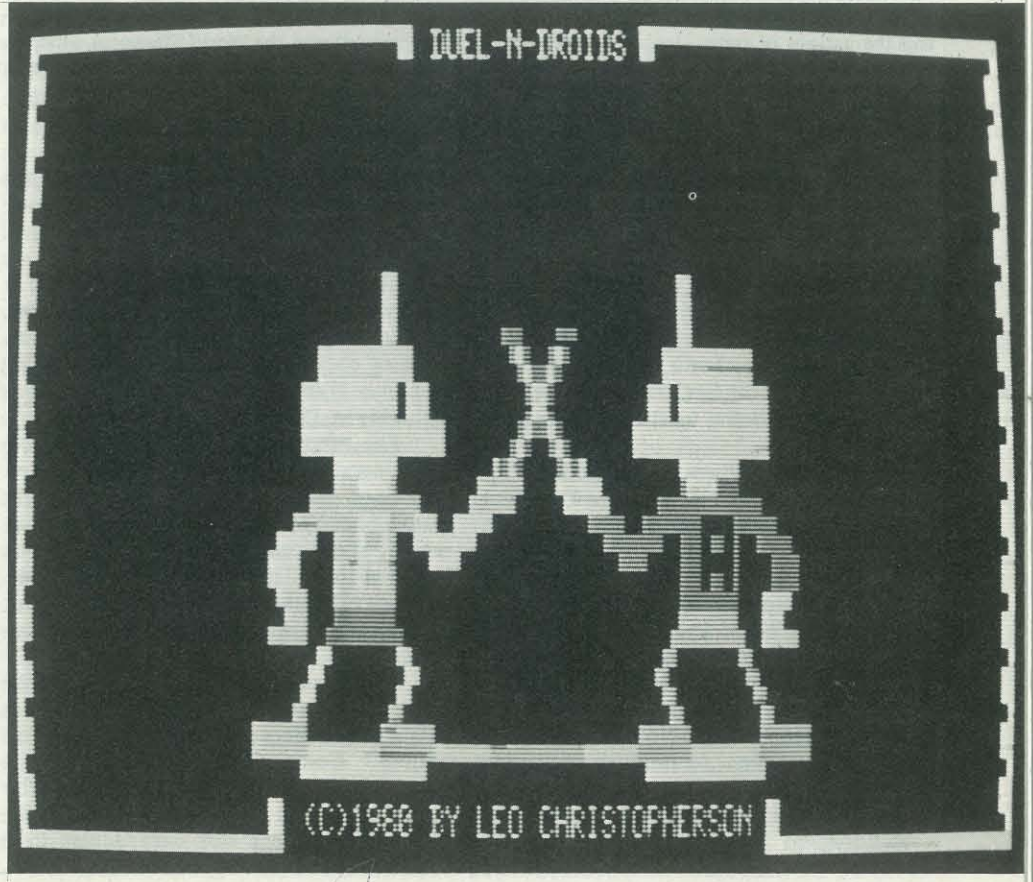
News

As in many other segments of our society, industrial engineers are beginning to explore the many possibilities of microcomputers. At the American Institute of Industrial Engineers' Third National Conference on Computers and Industrial Engineering, held on October 22-24 in Orlando, Florida, a wide variety of presentations were given.

"Software Cost Estimation," a talk by Yasser Hosni of the University of Florida, focussed on the new importance of software cost in this world of low-cost microcomputers. Mr. Hosni noted the importance of estimation method, application type, software life cycle, program size, data file size, programmer productivity, software testing and reliability.

Michael Joost of the University of Alabama gave an intensive survey of voice input and output systems and noted the tremendous benefits of speech synthesis and recognition systems in not only industry but in the area of health care as well. Messrs. Uzemek and Bisett of the University of Arkansas discussed an algorithm that will allow an implanted microcomputer to monitor, detect and correct ventricular tachycardia and ventricular fibrillation, disorders that threaten the lives of 400,000 people annually.

Some other topics included a description by Philip Wolfe of Oklahoma State University on how graduate I.E. students at his university built a microcomputer-controlled storage and retrieval system; John Riegel of Kansas State University on how microprocessors are expanding their roles in manufacturing and machine control; and Robert Redgate, Manager Energy Conversion at Walt Disney World, on how a minicomputer automated system was implemented to save energy.



EPCOT CENTER – Artist's rendering shows major new project planned for Walt Disney World in Florida, a showcase for the concepts of tomorrow and the nations of today. Includes Future World, (foreground) and the World Showcase which surrounds the lagoon in the distance. Giant geodesic sphere at entrance houses Spaceship Earth, introducing the concepts of Future World and other major pavilions dealing with energy, transportation, the land and the seas. Project will open October 1, 1982.

Mickey Mouse and Donald Duck are as much a part of Americana as "mom and homemade apple pie." But what about the machines used to bring to life Walt Disney's vivid imagination (e.g., Pirates of the Carribbean, Haunted House, Enchanted Tiki House)? Soon to open in October, 1982, at DisneyWorld in Florida will be EPCOT Center, where the computers being used to run the park will be featured.

The $800-million Experimental Prototype Community of Tomorrow project will demonstrate new technologies in ways people can understand and enjoy. Located on a 600-acre site, EPCOT Center will encompass two principal theme areas, Future World and World Showcase.

Future World will make broad use of the Audio-Animatronics system for animating figures of people, animals and otherwise inanimate objects, spectacular new visual screen techniques, traveling theater segments, projection, laser and other futuristic technologies.

Look to DisneyWorld in 1982 if you've always been curious about the computers behind the scenes.

Electronic home banking became a reality in Knoxville, Tennessee, where, consumers are now able to use the services of their local bank with a computer at home. It is expected to be available nationwide in 1981.

The "Express Information" bank-at-home service is a joint venture of United American Service Corporation (USAC); Radio Shack, a division of Tandy Corporation; and CompuServe, a subsidiary of H & R Block. The United American Bank in Knoxville was selected as the first bank to use and market the service to its customers.

For an estimated price of $15 to $25 a month, 400 of the bank's customers will gain services of Radio Shack's new TRS-80 Color Computer, including a standard keyboard that plugs into the customer's own television set and telephone. Customers will have access to a comprehensive news and financial advisory service, be able to pay most of their bills, receive current information on their checking accounts, use a sophisticated bookkeeping service, and apply for loans. This opens a new dimension in convenience banking. The news and information network was developed by CompuServe of Columbus, Ohio.

Customers of the bank are issued a security pack and certificates that can be redeemed at any of the 6,000 Radio Shack outlets nationwide. Without directly purchasing the computer, customers will be able to use it for a number of other functions: entertainment, education, home security, message services and electronic filing, to name a few.

The NW PET USERS GROUP has a new address and meeting place.

They now meet the second Tuesday of the month (7:30) at U of Washington, Academic Computer Center, 3737 Brooklyn, Seattle.

The new address is NW Pet Users Group, Richard Ball, 2565 Dexter N. #203, Seattle, WA 98109.

Events

Unconventional? DunDraCon VI, the sixth annual FRP and science-fiction gaming convention, will be held over President's Day Weekend, February 14-16, 1981, in Oakland, California.

There will be 72 hours of non-stop gaming, workshops, seminars, demonstrations, table top fantasy miniatures wargaming, films and more.

Among those who will be there are Steve Perrin (*Runequest*), Dave Hargrave (*The Arduin Grimoire*), Clint Bigglestone (*The Universal Fantasy Supplement*), Greg Stafford (creator of *White Bear, Red Moon* and *Dragon Pass*) and many other designers, adventure masters and just plain fun-type people.

The cost is $12.00 for all three days and nights of DunDraCon VI (through January 14, 1981; $15.00 at the door thereafter). Contact DunDraCon VI, 386 Alcatraz Avenue, Oakland CA 94618.

The convention will be held at the Leamington Hotel, at 19th and Franklin Streets, Oakland CA 94612.

Microcomputer Educational Conference, held January 16-17, 1981, at Arizona State University, Tempe, Arizona, will introduce educators to the many applications of microcomputers in the classroom, including elementary and secondary schools, fine arts, career and vocational studies and special education. Contact Dr. Gary G. Bitter, Arizona State University, Payne 203, Tempe, Arizona 85281.

RECREATIONAL COMPUTING wants to be your informational bulletin board. If you are sponsoring an event, let us know about it well in advance. Send pertinent information to Events, Recreational Computing, 1263 El Camino Real, Box E, Menlo Park, CA 94025. ■

ComputerTown Family Day

by Patricia Smith

Bill Scarvie

"I'm waiting for my hit points," a Menlo Park sixth grader says as he presses a microcomputer key and receives new information for "Dungeons," a game that requires a player to use strategy to find gold and to fend off attacking beasts.

The blond, blue-eyed boy, dressed casually in shorts and a T-shirt, indicates it's the first time he has used a microcomputer, and he's more interested in playing the game than talking.

Behind him, a younger boy is absorbed in another game. In a way that only children who are missing their two front teeth speak, he reads instructions that are flashing on the screen, then waits for his mother's suggestions of different moves he can make.

"I bet this is like 'Adventure'," an older boy says to a companion as they walk around the younger boy.

Many of the children and some of the adults that have gathered this fall Saturday afternoon at the Menlo Park Public Library for ComputerTown, USA!'s "Family Day" are experiencing computers for the first time. "I've only seen about one-fourth of the kids before," comments Pat Cleland, Coordinator of the ComputerTown, USA! Project, established to bring computer literacy to the general community through numerous events and classes for both children and adults.

ComputerTown Library Liaison Barbara Harvie, who oversees the general daily use of the project's microcomputers at the Menlo Park Library, is assisting a fourth grader, while explaining she is a former Stanford librarian with a humanities background who loves computers and loves to play games. She says the kids are learning a lot from computers.

Alfred Sugarman, an engineer with a consulting firm, is standing near his son and elaborates that computers help children think more logically by giving them instructions to follow and then immediate feedback. "It focuses their thinking as they concentrate on a goal," he says.

Gigi Van Kiram, here with her husband Pete, their two children, ages five and six, and an eight-year-old neighbor, says they would eventually like to have a home computer, but are waiting for their children to get a little older and for the prices of microcomputers to come down. "For the time being, we can come here," she explains. "This is a fun way to experiment and learn about computers and then we'll see how their (the children's) interests develop."

Dan Hilberman, an electronics engineer whose older son learned BASIC in a special eighth grade class, explains that his second son, a sixth grader, is eager to try the computer. He finds the ComputerTown library program "a terrific supplement to the schools" and feels learning about computers is necessary because they are becoming so pervasive in society.

He also explains that with a computer children learn logical thought processes, and he finds it exciting when they get tired of playing a set program and want to do their own programming and creating.

Janice Burch, Branch Consulting Director for Ross Systems, is interested in teaching children how to write programs and came here because she has worked on larger computers and wanted to see what a little one can do.

Other people dropping into the library for "Family Day" included Darlene Dorrance who wants to use a microcomputer as an instructional tool for educating her son at home and Yves Leclerc, a French-Canadian journalist, who is gathering information for a book on the impact of computers on society.

As the afternoon event comes to an end, Cleland indicates the day was a success because so many new people dropped in for a look at microcomputers. ■

ComputerTown, USA! — A computer literacy project of the people of Menlo Park, CA, the Menlo Park Public Library, and People's Computer Company, a non-profit educational corporation, P.O. Box E, Menlo Park, CA 94025.

The Electric Phone Book

A computerized bulletin board works just like an ordinary bulletin board system except that instead of paper and thumbtacks it uses a terminal, a computer, and the dial-up telephone network.

The list below was developed from several sources including the Peripheral People in Mercer Island, Washington and the People's Message System in Santee, California. It is being maintained by People's Computer Company's PCNET project, our effort to bring computers and telecommunications into the hands of everyone. While this is the most complete listing we have as of this writing, we would appreciate additions and corrections. Send them to PCNET, PCC, P.O. Box E, Menlo Park, CA 94025.

All the bulletin board systems listed here can be accessed by telephone using a 300-baud ASCII terminal and a Bell 103 modem. Most use carriage-return as a speed recognition character, after which they are self-teaching. All are free to anyone who calls, unlike the Arpanet, which is restricted, and The Source and MicroNet, which cost money. The list has been sorted by area code; consult your local telephone directory for geographical correspondence. ■

(201) 283-2724	**(214) 288-4859**	**(414) 241-8364**	**(713) 233-7943**
(201) 457-0893	(214) 634-2668	(414) 282-8118	(713) 693-8080
(201) 688-7117	**(214) 634-2775**	**(415) 348-2139**	**(713) 977-7019**
(201) 753-1225	(214) 641-8759	(415) 348-2396	(714) 449-5689
(201) 753-8152	**(216) 644-1965**	**(415) 493-7691**	**(714) 463-0461**
(201) 835-7228	(216) 754-7855	(415) 527-0400	(714) 495-6458
(201) 843-4563	**(301) 344-9156**	**(415) 647-9524**	**(714) 526-3687**
(201) 874-6833	(303) 759-2625	(415) 661-0705	(714) 537-7913
(201) 891-7441	**(303) 789-0936**	**(415) 683-4703**	**(714) 565-0961**
(201) 968-1074	(305) 261-3639	(415) 792-8406	(714) 571-5550
(202) 337-4694	**(305) 566-0805**	**(415) 851-3453**	**(714) 582-9557**
(202) 635-5730	(305) 689-3234	(415) 948-1474	(714) 730-1206
(203) 348-6353	**(305) 772-4444**	**(417) 862-7852**	**(714) 739-0711**
(203) 357-1920	(305) 821-7401	(419) 865-1584	(714) 751-1422
(203) 746-4644	**(305) 989-9647**	**(502) 245-8288**	**(714) 772-8868**
(205) 945-1489	(309) 688-0470	(502) 896-9624	(714) 898-1984
(206) 244-5438	**(309) 694-6531**	**(503) 646-5510**	**(714) 952-2110**
(206) 246-8983	(312) 255-6489	(512) 657-0779	(714) 962-7979
(206) 482-5134	**(312) 269-8083**	**(513) 671-2753**	**(714) 963-7222**
(206) 482-5590	(312) 337-6631	(513) 874-2283	(801) 375-7000
(206) 524-0203	**(312) 420-7995**	**(515) 279-8863**	**(801) 466-1737**
(206) 546-6239	(312) 622-8802	(516) 938-9043	(801) 753-6800
(206) 723-3282	**(312) 528-7141**	**(523) 223-3672**	**(802) 748-9089**
(206) 937-0444	(312) 545-8086	(602) 866-0258	(802) 879-4981
(209) 638-6392	**(312) 622-9609**	**(602) 956-5612**	**(803) 270-5372**
(212) 245-4363	(312) 767-0202	(602) 957-4428	(803) 270-5392
(212) 448-6576	**(312) 782-8180**	**(602) 957-9282**	**(803) 279-5392**
(212) 787-5520	(312) 782-9751	(604) 687-2640	(803) 771-0922
(212) 997-2186	**(312) 941-9009**	**(607) 754-5571**	**(803) 772-1592**
(213) 276-4276	(312) 964-7768	(607) 797-6416	(804) 340-5246
(213) 316-5706	**(313) 288-0335**	**(609) 983-5970**	**(805) 484-9904**
(213) 329-3715	(313) 357-1422	(612) 561-6311	(805) 527-9321
(213) 340-0135	**(313) 465-9531**	**(612) 929-8966**	**(805) 682-7876**
(213) 346-1849	(313) 477-4471	(614) 272-2759	(805) 964-4115
(213) 349-5728	**(313) 484-0732**	**(614) 649-7097**	**(806) 355-5610**
(213) 360-6332	(313) 569-2063	(615) 254-9193	(813) 223-7688
(213) 394-1505	**(313) 588-7054**	**(617) 354-4682**	**(816) 523-9121**
(213) 395-1592	(314) 838-7784	(617) 388-5125	(816) 531-1050
(213) 396-3905	**(316) 746-2078**	**(617) 431-1699**	**(816) 861-7040**
(213) 424-3506	(319) 353-6528	(617) 649-7097	(816) 931-3135
(213) 428-4718	**(319) 557-9618**	**(617) 692-3973**	**(817) 855-3916**
(213) 459-3177	(404) 394-4220	(617) 864-3819	(817) 855-3918
(213) 459-6400	**(404) 733-3461**	**(617) 897-0346**	**(817) 923-0009**
(213) 566-8035	(404) 790-8614	(617) 963-8310	(901) 276-8196
(213) 631-3186	**(404) 793-1045**	**(702) 826-7234**	**(901) 362-2222**
(213) 657-8803	(404) 939-1520	(702) 873-9491	(901) 761-4743
(213) 673-2206	**(404) 939-8429**	**(703) 281-2125**	**(902) 794-8198**
(213) 675-8803	(404) 953-0723	(703) 281-2222	(904) 243-1257
(213) 709-5423	**(405) 353-2554**	**(703) 379-0303**	**(904) 243-8565**
(213) 795-3788	(405) 528-8009	(703) 620-4990	(913) 362-6398
(213) 787-4004	**(408) 241-1956**	**(703) 734-1387**	**(913) 764-1520**
(213) 799-1632	(408) 296-5799	(703) 750-0930	(913) 782-5115
(213) 799-6514	**(408) 263-0248**	**(703) 893-9474**	**(915) 584-5393**
(213) 826-0325	(408) 263-9650	(703) 978-7561	(916) 393-4459
(213) 828-3400	**(414) 241-5406**	**(707) 448-9055**	**(918) 224-5347**
(213) 843-5390			

STATEMENT OF OWNERSHIP, MANAGEMENT, AND CIRCULATION

(Act of August 12, 1970: Section 3685, Title 39, United States Code)

1. Title of Publication: *Recreational Computing.*
2. Date of Filing: October 29, 1980
3. Frequency of Issue: Bi-monthly (6 issues, $12)
4. Location of Known Office of Publication: 1263 El Camino Real, Menlo Park, CA 94025.
5. Location of the Headquarters or General Business Offices of the Publisher: 1263 El Camino Real, Menlo Park, CA 94025.
6. Names and Addresses of Publisher, Editor: Publisher, Ann Merchberger, 1263 El Camino Real, Menlo Park, CA 94025. Editor, Joan Hiraki, 1263 El Camino Real, Menlo Park, CA 94025.
7. Owner (If owned by a corporation, its name and address must be stated and also immediately thereunder the names and addresses of stockholders owning or holding 1 percent or more of total stock. If not owned by a corporation, the names and addresses of the individual owners must be given. If owned by a partnership or other unincorporated firm, its name and address, as well as that of each individual must be given.): People's Computer Company, 1263 El Camino Real, Menlo Park, CA 94025.
8. Known Bondholders, Mortgagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities: None.
9. For Completion of Nonprofit Organization Authorized to Mail at Special Rates: The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes have not changed during the preceding 12 months.
10. Extent and Nature of Circulation

A. Total number copies printed: Average number copies each issue during the preceding 12 months, 10,878. Actual number copies of single issue published nearest to filing date, 10,620.

B. Paid circulation: 1. Sales through dealers and carriers, street vendors, and counter sales: Average number copies each issue during preceding 12 months, 3,781. Actual number copies of single issue published nearest to filing date, 3,694. 2. Mail subscriptions: Average number copies each issue during preceding 12 months, 5,253. Actual number copies of single issue published nearest to filing date, 5,193.

C. Total paid circulation: Average number copies each issue during preceding 12 months, 9034. Actual number copies of single issue published nearest to filing date, 8,887.

D. Free distribution by mail, carrier, or other means; samples, complimentary, and other free copies: Average number copies each issue during preceding 12 months, 252. Actual number copies of single issue published nearest to filing date, 197.

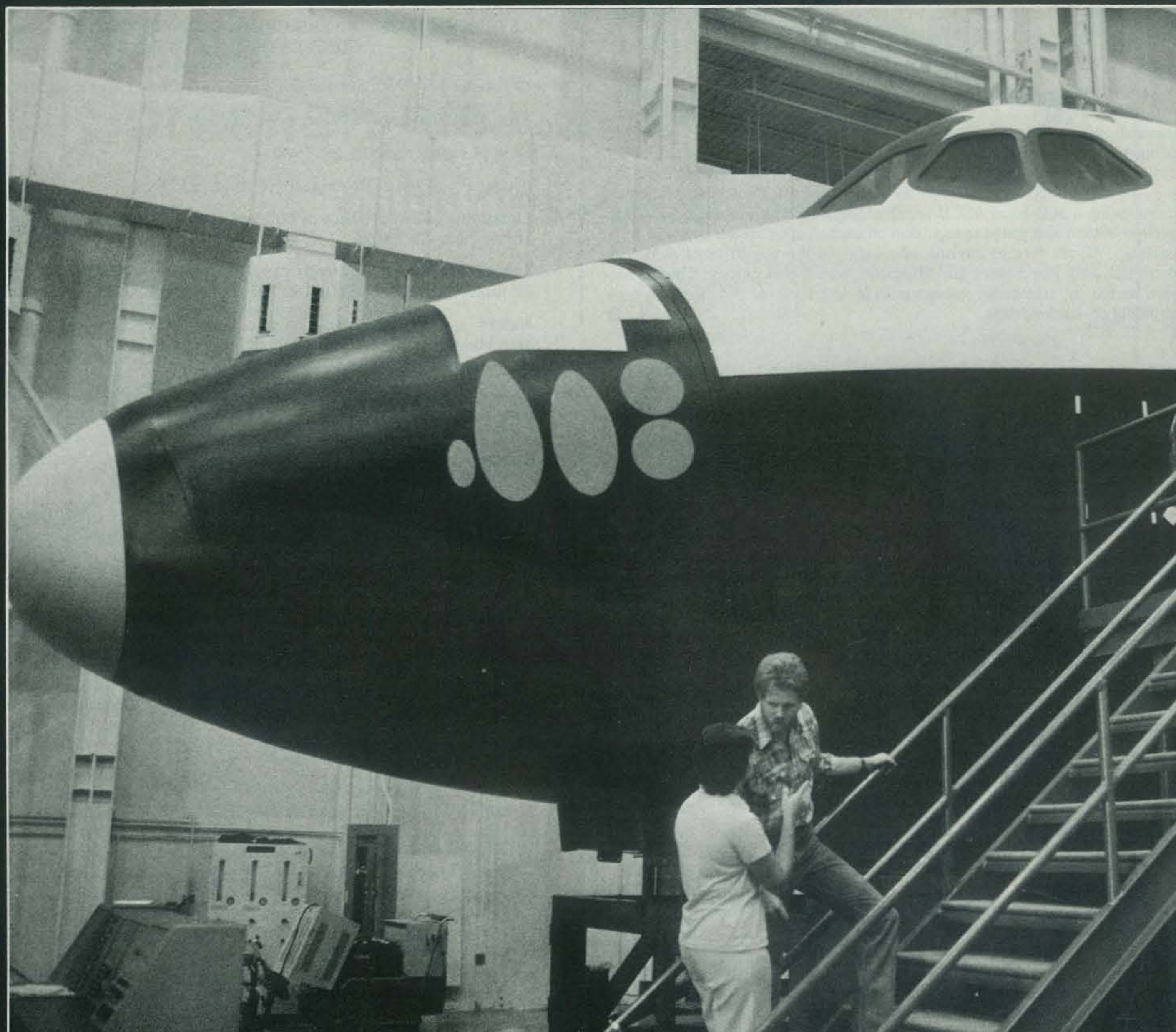
E. Total Distribution: Average number copies each issue during preceding 12 months, 9,286. Actual number copies of single issue published nearest to filing date, 9084.

F. Copies not distributed: 1. Office use, left over, unaccounted, spoiled after printing: Average number copies each issue during preceding 12 months, 1,403. Actual number copies of single issue published nearest to filing date, 1,440. 2. Returns from news agents: Average number copies each issue during preceding 12 months, 189. Actual number copies of single issue published nearest to filing date, 96.

G. Total: Average number copies each issue during preceding 12 months, 10,878. Actual number copies of single issue published nearest to filing date, 10,620.

I certify that the statements made by me above are correct and complete.

ANN MERCHBERGER, Publisher



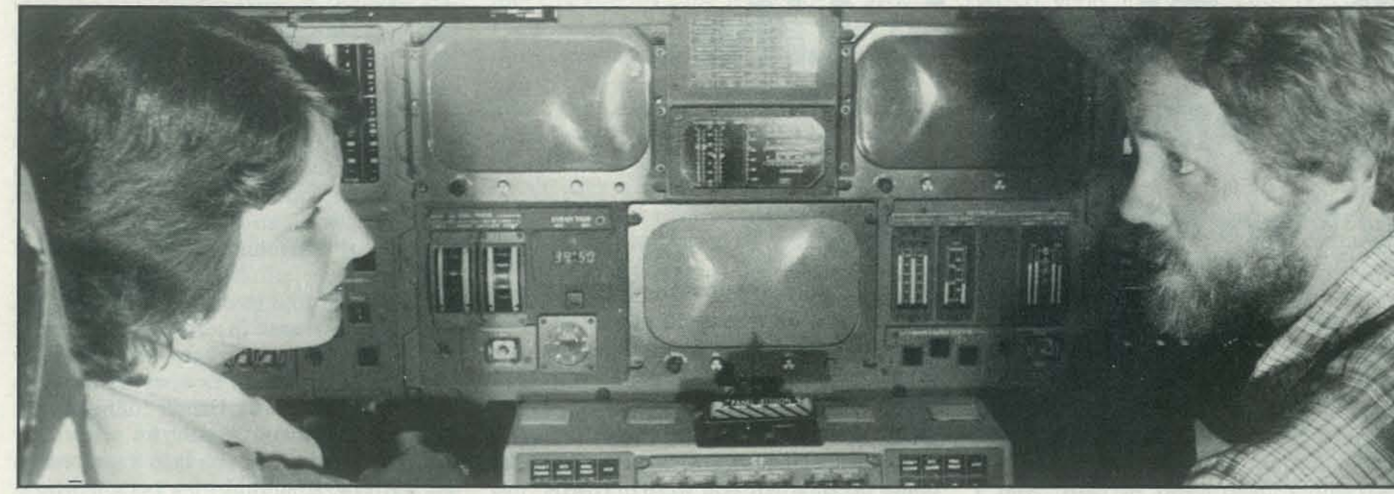
Adventure of the Mind

by J. R. Hiraki

Adventure of the Mind, a winner in the International Film and Television Festival of New York, is a series of six, 15-minute television lessons on the world of personal computing.

The film content was developed by Dr. Paul L. Hazan of the Johns Hopkins University Applied Physics Laboratory, and written, produced and directed by David Hopwood of Children's Television International, Inc. The production of the series was made possible in part by grants from Radio Shack, a division of Tandy Corporation, and the Institute of Electrical and Electronics Engineers Computer Society.

According to Hopwood, the idea for



"Adventure of the Mind" host John Hertzler and astronaut trainer Michelle Brekke board the Space Shuttle at Houston's Johnson Space Flight Center.

the series originated in a discussion between Dr. Hazen and a colleague, Timothy Keen, on the advances of microprocessors and its importance to young people in their future lives. "Paul thought it would be a good idea to have a television series," said Hopwood. "Unlike many people who have an idea for a TV series but who never carry it out, Paul had the tenacity to carry it through."

The approach of the film was not to make the viewer heavily dependent on mathematics. The concept was to suit personal rather then institutional needs. In one episode there is a poet who does his work on a CompuColor. The letters and words appear in different colors to show alliteration, etc., and he transmits his poetry via telephone lines. Although the film was made for high school students, Hopwood says that from the TV aspect there is universal appeal.

There were three main goals associated with the series. They were to motivate students to (1) explore the potential impact of computing on the individual and society; (2) determine potential applications of personal computing to meet individual and societal needs; and (3) apply personal computing to meet their own individual needs.

The series visits different places and invites the viewer to join the excursion. The first program, for example, goes to Houston, Texas, to see the space shuttle. The objective was to see how personal computers can serve as a tool rather than just a toy. This episode showed the connection between large and small computers.

For viewer identification, the film includes an interview with students on

For viewer identification, the film include an interview between students on what computers meant to them. This interaction helps the viewer relate to others who are going through the same problems in their dealings with computers.

In the series, most of the people visited used either an Apple or TRS-80, although a variety of microcomputers were used including PETs and DynaBits.

Although Hopwood mentioned that Dr. Hazen is thinking about a future series, the problem really is funding. "If the K-12 group has got their fill of computer education films, then funding becomes difficult," said Hopwood. "If you get out of general education on an introductory level, then you reach a special audience and funding becomes even more difficult."

To find out if this series will be shown in your area, contact your local Public Broadcasting System and ask for the Instructional Series group. ■

A high school student finds potential careers by inputting her academic strengths and interests into a personal computer.

A student composes a melody on a microcomputer.

The Pirate's Life for Me

by Richard Allan Karp

Whenever I need a few bucks – which is often, since I have an expensive lifestyle – I just step up to the nearest automatic teller machine. No matter what bank it is, just a few innocuous operations and I have $100 in cash. And I don't have to worry about paying it back, for no matter how often I cash in, I get no bill, cause no suspicious ringing in the guts of the bank's computer. All that happens is that some poor soul – a different, honest one each time – gets the bill about three weeks later. For, in my own modest way, I am living off the proceeds of my own perfect computer crime.

Nonsense? Of course. I've never used any bank teller machine anywhere, and I don't intend to. I wouldn't want to be the poor soul who gets the bill three weeks later! But I enjoy indulging in a particular fantasy that is my own sort of computer recreation: plotting the perfect computer crime.

There are those who feel that even to talk about computer crime is wrong, somehow encouraging it. These people get joy from curling up with a good Agathie Christie. Writing about homicide, robbery, chainsaw massacres: these are just good clean fun, but computer crime fantasies are evil.

There seems to be some general belief that computer crimes are particularly hard to detect; even Lt. Colombo might fail to crack the perfect automatic teller crime: "Oh, and by the way, would you mind explaining how the parity error could occur when you were in town?"

Well, this difficulty of detection better not last or we could be looking at the first $1 billion bank wire fraud and wondering what to do next. Maybe if we publicize the problem, we will get closer to the solution. So here are a few of my fantasy favorites. Rules for the game are at the end; can you do any better?

In most larger airports the box stands looking impregnable yet friendly – the American Express cardmember's traveler's cheque dispenser. With a previously validated account, you step up to the box, insert your card, go through a transaction, and receive up to $500 in traveler's cheques. The bill comes next month. It's certainly nice to know that you can get this money when you are far from home. And, once the traveler's cheques come out, you can sign any name on the top and the machine can't detect it, of course.

What I'd like to do is this: step up to the dispenser and insert a special card. This card causes the cheque dispenser to pick someone who is two weeks or more away from the end of their billing cycle. The person picked should be a user for whom this purchase fits into a reasonable use pattern. It should be a person who is currently within his normal credit limits. And it shouldn't be anyone whose name I've used within the past 10 years. This is the lucky person who will automatically get the bill for my $500 in traveler's cheques.

Well, maybe it's too hard to get information so complete. I should be able to tap all the phone lines into and out of one (or more) of these machines over a large series of transactions. I can then extend the tap to keep track of the information about people using the machine; this would give me a stock of names, card numbers, and passwords. Oh, occasionally someone might close his account, but I use only recent data. Ahhh, American Express traveler's cheques. I never leave home without them.

Boy, I like to go to faraway places! And so do my friends. But plane fares have gone up so fast the past year that my friends are having trouble affording them. No problem. Whenever one of my friends wants to go somewhere, I just buy a ticket for him. Well, not quite. What I do is pay for his ticket and let him pick it up at the airport. I inform the airport ticket counter about all this by sending a prepaid ticket advice (PTA) over the airline's computer wire. When a PTA arrives, the local office makes up the ticket. Whoever picks up the ticket has to identify himself, and then the ticket is handed to him. It's a perfectly legitimate ticket, written by the airline people, not me.

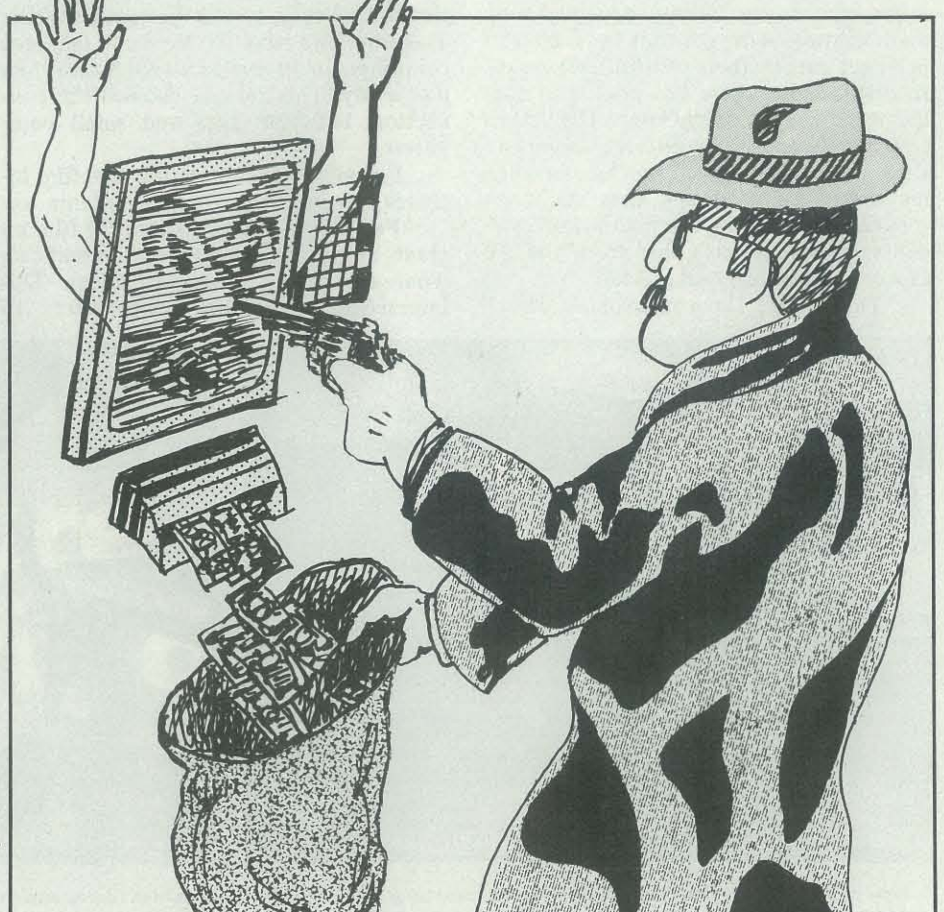
So suppose that I knew how to send PTAs over airline computer lines, again billing people who would be well within their credit limits and early in their billing cycles. A different person, probably a different airline system each time. It isn't cash, but it would provide some interesting places to spend those traveler's cheques. Might as well make all those tickets first class, while I'm at it.

Need some hotels or meals on arrival? The same network that sends PTAs can send a miscellaneous charge order (MCO) good for tours, auto rental, and meals. These are limited to a few hundred dollars, but anything helps in these days of high inflation.

Almost any day, you can bet on the Quebec Lottery. Here's how it works: You pick a three-digit number, and the next day the computer picks a three-digit number. If your number matches the computer's number, you win at the rate of 450 to 1. Not bad (although when the Mafia runs the numbers game, the payoff is 600 to 1; once again government loses out to private enterprise)!

Tickets can be bought throughout Quebec using terminals directly wired to the main system. It might take a little more work on the main system, getting

(continued on page 43)



Skiing

by Michael van de Panne

This program simulates a ski race. The racer can choose between a slalom race and a giant slalom race. The object of the game is to complete the whole course. This is an accomplishment in and of itself in the highest skill level.

You can also race a course of a certain length and try to improve upon your previous time. The time is only given, however, while racing in giant slalom. Also, the computer cannot tell if you miss a gate in slalom.

This program was written for a Radio Shack Level II computer.

Remember that the closer you come by a gate, the better your time will be. ■

```
10 CLS
20 PRINTTAB(25)"**** Slalom ****"
30 PRINTTAB(19)"The object is to cut as close by the
    gates as possible."
40 PRINTTAB(14)"Press 'z' key to turn left :
    Press '/' key to turn right."
45 PRINT"Hit space bar to go straight"
50 INPUT"Rate yourself as a skier 1=Best 3=Worst";X
51 INPUT"What kind of race do you want 1=slalom
    2=giant slalom";I
52 ONIGOTO53,55
53 A=68:E=52:H=6:F=4:GOTO60
55 A=80:E=40:H=11:F=14
56 IFX=1THENF=8
57 IFX=2THENF=11
58 IFX=3THENF=14:H=13
60 INPUT"How many gates do you want in the course";L
70 PRINTTAB(10)"The starter counts down."
80 IFX=1THENX=0
90 IFX=2THENX=20
100 IFX=3THENX=40
110 FORN=5TO1STEP-1
120 PRINT"...."N
130 FORD=1TO200:NEXTD:NEXTN
140 CLS
150 PRINT"go !!!"
160 S=414
165 K=9
170 D=60
180 C=960
190 PRINT@412,CHR$(191)
200 PRINT@416,CHR$(191)
210 W$=INKEY$
220 IFW$="z"THENB=-2
230 IFW$="/"THENB=2
240 IFW$=" "THENB=0
250 S=S+B
260 D=D+2*B
270 V=V+T:IFV=KTHENGOSUB550
280 FORN=0TOX:NEXTN
290 PRINT@S,CHR$(191);
300 M=M+1
310 IF M=HTHEN320  ELSE370
320 IFU=1THEN330  ELSE340
330 J=RND(F):SET(E+J,46):T=1:U=2:M=0:GOTO350
340 J=RND(F):SET(A-J,46):T=1:U=1:M=0:GOTO350
350 P=P+1
355 PRINT@965,P;
360 Y=Y+1
370 PRINT@S,CHR$(193);
380 PRINT@C,CHR$(193)
390 IFP=LTHENGOTO410
400 GOTO210
410 IFP=LTHEN430
420 PRINT"You lasted";P;"gates.":GOTO470
430 PRINT"You survived the whole course!!"
435 IFL<20THENPRINT"The course was too short so ":GOTO460
440 IFX=0THENPRINT"You win a gold medal":GOTO470
450 IFX=20THENPRINT"You win a tin medal":GOTO470
455 IFX=40THEN460ELSE470
460 PRINT"You win nothing for your cruddy efforts"
470 PRINT"Your time was";Y+RND(0);"seconds. This makes";Y/P;
    "seconds per gate."
480 PRINT"You completed"P/L"of the course."
490 INPUT"Do you want to race again (y/n)";G$
500 IFG$="y"THENRUN
510 END
520 CLS:PRINT"You hit the gate and broke your leg.":GOTO410
530 PRINT"You missed the gate. Learn how to ski better !!!"
540 PRINT"Practice a lot !! You need it!":GOTO410
550 IFPOINT(D,19)THEN520
560 IFPOINT(D+1,19)THEN520
570 IFPOINT(D+1,20)THEN520
580 IFPOINT(D,20)THEN520
585 IFI=1THENV=0:K=5:RETURN
590 IFU=2THEN630
600 IFD<A-JTHEN530
610 Y=Y+D-A+J
620 V=0
621 T=0:RETURN
630 IFD>E+JTHENGOTO530
640 Y=Y-D+E+J:V=0
641 T=0:RETURN
```

Twenty Questions

by David J. Beard

Editor's Note: LISP or List Processing is an interpretive language, developed for manipulation of symbolic strings of recursive data; i.e., used to develop higher-level languages. It is often used in the area of Artificial Intelligence.

This article describes a version of the old guessing game implemented in a homebrew LISP-like language. This game is very similar to one described in the article, "Designing Animal Games," which appeared in the September, 1979, issue of *Recreational Computing*. The human player thinks of some object, and the computer tries to guess the object by asking questions. If the computer fails to guess the object, it asks the human player for a new question that distinguishes the object from the computer's nearest guess. In this way, a binary tree of questions is built up whose terminal nodes are all guesses of specific objects. The game is a valuable reading and logic drill for children (it held our six-year-old's attention for over an hour). If the game is played with names of people, it has surprising appeal as a party game for adults.

Figure 1 shows the game's "database" at several points during a sample game. By referring to Figure 1 and the sample run in Figure 2, we can see how the database grows. When the game is first started (Figure 1A), the computer is limited to just one terminal node – "IS IT A WOMBAT?". After the player enters his object (WATER) and a question (IS IT ALIVE?), the database looks like Figure 1B. This is a single question node in the form:

(Q (question)(yes branch)(no branch))

Both the "yes branch" and the "no branch" are terminal nodes of the form:

(G object to guess)

After a second play, the computer adds "JOHN Q. CITIZEN" to its database. The "yes" branch is now another question node (Figure 1C). Figure 1D shows the database after several more plays. It can continue to grow to the limits of memory.

If you're accustomed to programming in a procedural language like BASIC, the listing in Figure 3 may require some explanation. A "program" in LISP is not really a program at all, but a collection of user-defined functions. The LISP interpreter evaluates one function and returns a single value, although the function may be built up of many other functions and the single value returned may be a list of lists of lists. The main loop for Twenty Questions is the list "IT." Typing "DO IT" causes this list to be evaluated over and over. The expression:

(SETQ DATABASE (SEARCH DATABASE))

sets the list "DATABASE" equal to the value returned by the function "SEARCH." SEARCH calls itself recursively (through LEARN1 and LEARN2). The argument handed to SEARCH at each level of recursion is either the "yes branch" or the "no branch" from the level before. When a terminal node is reached, GUESSRIGHT or GUESSWRONG will return the original terminal node or a new question node, respectively. ADDQ actually creates the new question node. At this point the LISP interpreter finally has a value to return; the value for the original "SEARCH" is constructed from the inside out. LEARN1 and LEARN2 actually do the reconstruction of the list. SEARCH hands the new list to SETQ, which makes it the new value of the symbol "DATABASE," and off we go again.

Notice the absence of any dimensioning statements or declarations of type. LISP is interactive, generalized, and open-ended to a greater degree than most other languages. Its most glaring disadvantage is the profusion of parentheses, an inescapable side effect of the branching tree structure. I feel that LISP should be receiving more attention as a vehicle for fantasy gaming and computer-assisted instruction. The game given here will fit into 16K of memory along with the LISP interpreter, although another 4K of workspace dramatically reduces the number of times the trash collector runs.

The LISP interpreter used here has several non-standard features. If "EVAL" is handed a list whose first element is not the name of some function, it will evaluate each element of the list and return the value of the last element. The "DO" function evaluates its argument once, and then evaluates the expression obtained at least once, or repeatedly until the value returned is NIL. The listings given here are the result of a simple pretty-printing function; there is no provision for saving separate source files. Also, many of the primitive operations are designed to return some default value instead of an error stop. If you are re-writing this game for another LISP, I can assure you that it does indeed run on my interpreter in this form, but you can expect to make quite a few modifications. ■

David J. Beard lives in Lebanon County, Pennsylvania, with his wife and daughter. His background is in forestry, military airlift, and industrial maintenance. Since eearly 1976, he has been working full time as a free-lance consultant under the trade name "Sortiarii." Most of his work is in small business systems and industrial control electronics. His major computer-related interests are spaceflight and artificial intelligence.

Figure 1. "Database" at Several Points During a Sample Game.

```
     *** FIG. 1A ***
( G A WOMBAT )


     *** FIG. 1B ***
( Q
  ( IS IT ALIVE? )
  ( G A WOMBAT )
  ( G WATER )
  )


     *** FIG. 1C ***
( Q
  ( IS IT ALIVE? )
  ( Q
    ( IS IT A PERSON? )
    ( G JOHN Q.  CITIZEN )
    ( G A WOMBAT )
    )
  ( G WATER )
  )


     *** FIG. 1D ***
( Q
  ( IS IT ALIVE? )
  ( Q
    ( IS IT A PERSON? )
    ( G JOHN Q.  CITIZEN )
    ( Q
      ( DOES IT HAVE A MASK? )
      ( G A RACCOON )
      ( Q
        ( CAN IT FLY? )
        ( G A ROBIN )
        ( G A WOMBAT )
        )
      )
    )
  ( Q
    ( IS IT A PLACE? )
    ( G ROCKY MOUNTAIN )
    ( Q
      ( IS IT A MACHINE? )
      ( G A COMPUTER )
      ( Q
        ( IS IT A LIQUID? )
        ( G WATER )
        ( G THE WIND )
        )
      )
    )
  )
```


Figure 2. Sample Run.

```
          TINY LISP READY
   DO IT

       *** TWENTY QUESTIONS ***

   YOU THINK OF SOMETHING AND I'LL
   TRY TO QUESS WHAT IT IS.

   READY? YES


   IS IT A WOMBAT ? NO



   I'LL HAVE TO GIVE UP...
   WHAT WAS IT? WATER

   WHAT QUESTION WOULD DISTINGUISH
   WATER FROM A WOMBAT ?
   IS IT ALIVE?

   FOR A WOMBAT ,
   IS IT YES OR NO? YES


       *** TWENTY QUESTIONS ***

   YOU THINK OF SOMETHING AND I'LL
   TRY TO QUESS WHAT IT IS.

   READY? YES


   IS IT ALIVE? YES

   IS IT A WOMBAT ? NO


   I'LL HAVE TO GIVE UP...
   WHAT WAS IT? JOHN Q.  CITIZEN

   WHAT QUESTION WOULD DISTINGUISH
   JOHN Q.  CITIZEN FROM A WOMBAT ?

   IS IT A PERSON?

   FOR A WOMBAT ,
   IS IT YES OR NO? NO


       *** TWENTY QUESTIONS ***

   YOU THINK OF SOMETHING AND I'LL
   TRY TO QUESS WHAT IT IS.

   READY? YES


   IS IT ALIVE? YES

   IS IT A PERSON? NO

   IS IT A WOMBAT ? YES


   IT WAS A WOMBAT !

   I GOT IT WITH 3 QUESTIONS.
```

Figure 3. Program Listing.

```
     *** IT ***
( DO
  ( '
    ( START )
    ( PRINTLIST
      ( ' YOU THINK OF SOMETHING AND

      I'LL TRY TO QUESS WHAT IT IS.  ) )
    ( PRINT )
    ( PRINT )
    ( ASK
      ( ' READY? ) )
    ( PRINT )
    ( SETQ DATABASE
      ( SEARCH DATABASE ) ) ) )

    *** START ***
( NFUN
  ( )
  ( COUNT
    ( '
      ( PRINT ) )
    4 )
  ( SPACE 4 )
  ( PRINTLIST
    ( ' *** TWENTY QUESTIONS *** ) )
  ( PRINT )
  ( PRINT )
  ( SETQ DEPTH 0 ) )

    *** ASK ***
( FUN
  ( A )
  ( PRINTLIST A )
  ( SETQ INPUT
    ( READ ) )
  INPUT )

    *** SEARCH ***
( FUN
  ( A )
  ( SETQ DEPTH
    ( + DEPTH 1 ) )
  ( COND
    ( ( =
        ( CAR A )
        ( ' G ) )
      ( GUESS ) )
    ( ( =
        ( CAR A )
        ( ' Q ) )
      ( QUESTION ) ) ) )


    *** GUESS ***
( NFUN
  ( )
  ( PRINT )
  ( DO
    ( '
      ( ASK
        ( APPEND ?
          ( APPEND
            ( CDR A )
            ( ' IS IT ) ) ) )
      ( NOT
        ( OR
          ( = INPUT YES )
          ( = INPUT NO ) ) ) ) )
  ( PRINT )
  ( COND
    ( ( = INPUT YES )
      ( GUESSRIGHT ) )
    ( ( = INPUT NO )
      ( GUESSWRONG ) ) ) )

    *** GUESSRIGHT ***
( NFUN
  ( )
  ( PRINT )
  ( PRINTLIST
    ( ' IT WAS ) )
  ( PRINTLIST
    ( CDR A ) )
  ( PATOM ! )
  ( PRINT )
  ( PRINT )
  ( PRINTLIST
    ( ' I GOT IT WITH ) )
  ( PATOM DEPTH )
  ( PATOM
    ( ' QUESTIONS.  ) )
  A )

    *** GUESSWRONG ***
( NFUN
  ( )
  ( CONCEDE )
  ( SETQ OB1
    ( CDR A ) )
  ( GETOB )
  ( GETQU )
  ( GETAN )
  ( ADDQ ) )

    *** CONCEDE ***
( NFUN
  ( )
  ( PRINT )
  ( PRINTLIST
    ( ' I'LL HAVE TO GIVE UP...  ) )
  ( PRINT ) )

    *** GETOB ***
( NFUN
  ( )
  ( ASK
    ( ' WHAT WAS IT? ) )
  ( SETQ OB INPUT )
  ( PRINT ) )

    *** GETQU ***
( NFUN
  ( )
  ( PRINTLIST
    ( ' WHAT QUESTION WOULD DISTINGUISH ) )
  ( PRINTLIST OB )
  ( PATOM
    ( ' FROM ) )
  ( PRINTLIST OB1 )
   ( PATOM ? )
   ( PRINT )
  ( SETQ QU
    ( READ ) )
  ( PRINT ) )

    *** GETAN ***
( NFUN
  ( )
  ( PATOM
    ( ' FOR ) )
  ( ( PRINTLIST OB1 )
    ( PRINT
      ( ' , ) )
    ( ASK
      ( ' IS IT YES OR NO? ) )
    ( COND
      ( ( = INPUT YES )
        ( SETQ TRS OB )
        ( SETQ OB OB1 )
        ( SETQ OB1 TRS ) ) ) ) )

    *** ADDQ ***
( NFUN
  ( )
  ( SETQ B
    ( CONS
      ( ' G )
      OB ) )
  ( SETQ C
    ( CONS
      ( ' G )
      OB1 ) )
  ( CONS
    ( ' Q )
    ( CONS QU
      ( CONS B
        ( CONS C NIL ) ) ) ) )

    *** QUESTION ***
( NFUN
  ( )
  (, INQUIRE )
  ( COND
    ( ( = INPUT YES )
      ( LEARN1 ) )
    ( ( = INPUT NO )
      ( LEARN2 ) ) ) )


    *** INQUIRE ***
( NFUN
  ( )
  DO
  ( '
    ( PRINT )
    ( ASK
      ( CADR A ) )
    (·NOT
      ( OR
        ( = INPUT YES )
        ( = INPUT NO ) ) ) ) )

    *** LEARN1 ***
( NFUN
  ( )
  CONS
  ( ' Q )
  ( CONS
    ( CADR A )
    ( CONS
      ( SEARCH
        ( CADDR A ) )
      ( CONS
        ( CADDDR A )
        NIL ) ) ) )

    *** LEARN2 ***
( NFUN
  ( )
  CONS
  ( ' Q )
  ( CONS
    ( CADR A )
    ( CONS
      ( CADDR A )
      ( CONS
        ( SEARCH
          ( CADDDR A ) )
        NIL ) ) ) )

    *** PRINTLIST ***
( FUN
  ( A )
  COND
  ( ( NOT A )
    NIL )
  ( ( ATOM A )
    ( PATOM A ) )
  ( TRUE
    ( PRINTLIST
      ( CAR A ) )
    ( PRINTLIST
      ( CDR A ) ) ) )
```

Health Care Professionals

Turning to Computers to Ease the Paperwork Flow

by J. R. Hiraki

Editor's Note: The field trip to El Camino Hospital was sponsored by Computer-Using Educators as part of their First Fall Conference on Classroom Applications of Computers K-12 held on September 26 and 27, 1980.

As is true in most professions, the medical field is barraged with lots and lots of paperwork that must be processed and recorded on a daily basis. From the time a person checks into a hospital until he leaves, he is on someone's chart or in someone's files. At El Camino Hospital, a 464-bed general acute care facility in Mountain View, CA, much of the "paperwork nightmare" has been alleviated through the use of a computer system.

El Camino Hospital's Information Processing Communications system has been in development for 15 years and is the most advanced hospital computer system in the world. Any other comparable system is still five years behind this one. Diagnoses, laboratory test results, scheduling, accounting and business office details are managed by the system.

According to Karen Fischer, a Management Engineering Department representative, "one idea behind the system was to keep costs down." For instance, Medicare and Medical costs in paperwork alone can be horrendous. A computer can make necessary charges relatively easily, thus becoming a cost-efficient tool. It's no wonder visitors from all over the world come to El Camino Hospital to learn about the system.

Ms. Fischer said there was not a cut in staff per se because of the system. However, some spots left open as a result of attrition or personnel turnover were not always refilled. "The computer has allowed nurses to be nurses rather than clerks," noted Ms. Fischer. In fact, 60 to 70% of all documents or charts are presently computerized.

El Camino Hospital was originally funded by HEW as a "development" hospital for a sophisticated information handling system. Doctors, nurses and other health care professionals indicated their needs and Technicon Medical Information Systems Corporation in Santa Clara, CA, developed the system.

Technicon implemented the system that runs on an IBM 370/155 computer and an IBM370/145 computer. The hospital terminals are connected by broadband telephone lines and are hardwired, making them immobile. The system currently uses machine language, but IBM may make a change in the near future. The redundancy in computer configuration is done as insurance against "down time," something hospitals cannot afford.

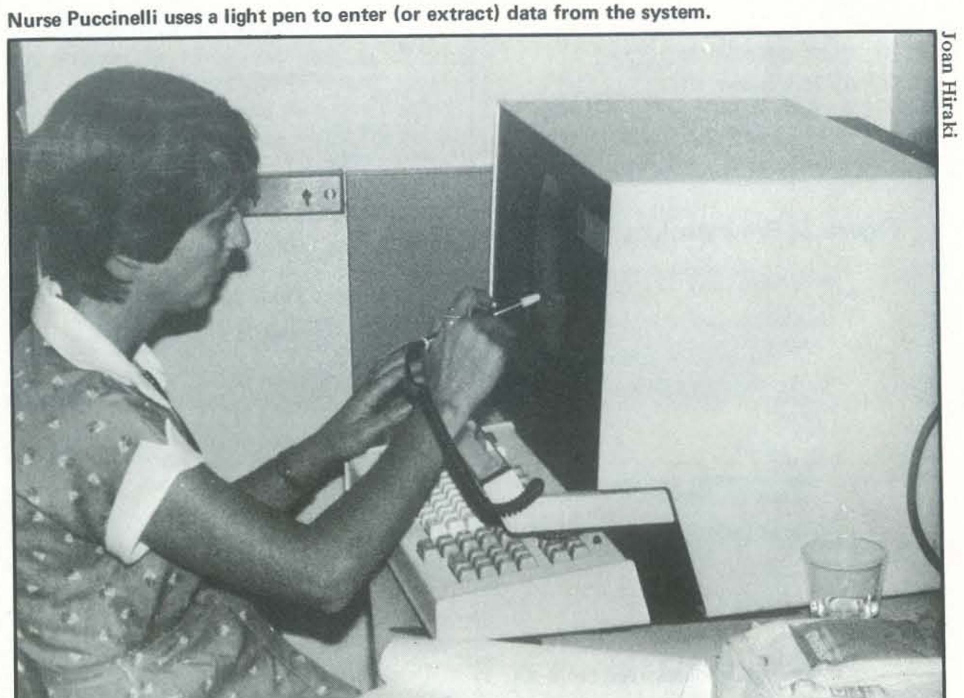
Although the system was partially government funded, the software is proprietary information. This is because the programming itself was not government funded. Ms. Fischer estimated that out of total cost of about $15 million, HEW funded about $1 million. Currently the government is funding a study of Patient Care Quality Assurance, which would include such things as Nurse Care Planning and Patient Care Audit.

Doctors

To use the computer, the doctor sits down at one of the 68 in-house terminals (and uses one of the 32 in-house printers if necessary), and enters his own code number. His code number does not appear on the screen, ensuring confidentiality. His name, however, will appear on the resulting print out. There are no programmers on-site as Technicon does all the programming for the hospital.

The doctor can request such information as patient list, pharmacy (what is in stock, common medications), laboratory tests, announcements for seminars or conferences, medical information index (journals, books, magazines), procedures and diagnosis, and even a "suggestion box" for improvements to the system, which is relayed to Management Engineering.

Nurse Puccinelli uses a light pen to enter (or extract) data from the system.



Joan Hiraki

The computer can also take certain precautions. As an example, for one particular medication that is know to produce side effects, if the doctor requests its use then the computer makes a verification check by asking the doctor what other symptoms the patient has and if he really needs this dosage, etc.

The selections are done with a light pen. The doctor is allowed two minutes between ordering selections. The time limit is imposed so that if the doctor is called away unexpectedly, he can't accidentally leave his code running for someone else to use.

Seventy-five per cent of all orders are entered by doctors for lab tests, radiology, diet, IVs, blood and vital signs. The most ordering is performed by obstetricians, gynecologists, and surgeons. Doctors may still use the manual method, although their handwritten orders must be entered into the computer by a nurse or a clerk. The trend appears to be that the more standardized the specialty, the more use of the computer by the physician.

Ms. Fischer mentioned that all physicians use the computer on their own. The hospital does not push them into learning the system. According to Ms. Fischer, the nursing staff is the largest user. They are given eight hours of training upon joining the staff, and it takes about a month to come up to speed.

Nurses

As with the doctors, each nurse has an individual code number. This enables her to obtain the same information as the doctor, but her name will appear on the print out. The computer code is a valid signature and was approved as such by the California State Legislature.

The computer eases the nurse's clerical duties in a number of ways. For instance, the computer communicates all medical orders to the appropriate recipients, eliminating the need for nurses to make multiple transcriptions from chart to card files and so on.

Also, medication administration scheduling is done automatically and results in hourly "medications due" lists for each nurse station. Charting of medication is done with the light pen. If scheduled medications are not charted within a given period, the computer issues reminder notices.

A further benefit is that because records of patient medication (or anything else) are stored in the computer, all charges to patients are done automatically.

Security, Down Time

There are approximately 3 million alphanumeric codes in the system. Each code is confidential. So far, there has not been a problem with people getting onto the computer without permission to do so. Ms. Fischer said that "to play around on the system until you found a correct code would raise questions."

The computer is down for one hour every night for business matters. It is also not uncommon for it to be down for 10 minutes several times a day. Long down times are considered several hours. There are, of course, safeguards to protect documents in the event of down time.

So far, the longest down times have been 22 hours and 10 hours. In these cases, the staff reverted to the manual backup system.

El Camino Hospital has come a long way in advancing the system of medical information processing communications. Not only has the system become more efficient, cost effective and sound proof, but it has allowed the nurse to perform her primary role, that of patient care. ■

Computers and the Volcanic Fallout

by Patricia Smith

"We didn't have problems with ash getting inside the building or in our computers, but it was one-half inch thick outside and we still have some on leaves, in the grass and on our roofs," a computer sales company representative in the Kelso/Longview area told *Recreational Computing* five months after Mount St. Helens first erupted.

According to the *RC* survey, computer related and other businesses in eastern Washington suffered a two week or longer slump following the May 18, 1980 blast, which left 34 people dead and 28 others missing, presumed dead, and a 250-mile wide cloud of volcanic ash blowing over eastern Washington, Idaho and Montana. Business people in Portland, a city dusted with ash after the volcano began erupting again in October following a short dormant period, did not experience the same slump.

No one contacted by *RC* reported any mechanical problems stemming from ash getting into their computers. But, since there is no way of knowing whether or not the eruptions are over, service managers were also contacted for information concerning ways of protecting home and business computers from volcanic ash.

Effect on Business

Right after the volcano exploded on May 18, felled trees caused flooding in rivers near the Kelso/Longview area and 500 people in the small town of Touttle were evacuated, while others in the area were warned to stay inside. Business stopped for two or three days and picked up slowly during the next few weeks.

After indicating she did not think their computer business in the Kelso/Longview area was affected, Mrs. Tharl of Northwest Consulting talked about the volcano affecting the community in general. She said that at first the volcanic eruption was a shock and a lot of people who worked in the mills, but were not native to the area, packed up and left. She added many of them wound up coming back and said now "everyone seems to be adjusting."

Steve Gaynor, Manager of Yakima Computer Store Digital Services in Yakima, Washington, a small city east of the volcano, located in one of the areas hardest hit with ash from the first eruption, reported the volcano definitely caused a slump in business. "People just weren't in the mood to buy anything. Now business is just beginning to pick up," he said.

Gaynor also reported that despite the great amount of ash, "It didn't appear to cause problems in systems."

Ole Munson, a manager of American Business Computers in Spokane, a larger city further east in Washington, said that after the May 18 blast, "We were buried in ash and there was no way to get out."

Munson said business was slow for about two weeks. But he indicated most of their customers were in areas that weren't inundated with ash, so their businesses weren't necessarily affected, and most didn't have to worry about ash getting tracked into their computers. He did say that "If customers asked about keeping their computers clean, we advised just keeping the filters clean."

Though some people in eastern Washington initially wore surgical masks to keep from breathing particles of ash (which contain silicon oxide and very minute amounts of iron, lead, zinc, arsenic, flouride and mercury), no one mentioned having any health problems that were associated with the ash.

According to people contacted in the Portland area shortly after the October volcanic eruptions, sales and business levels were not affected there. But, one Portland resident working in a computer sales office said the blasts have affected people psychologically, particularly those who moved to the area because they thought it was a kind of environmental utopia.

Maury Plumlee of Advanced Business Computers in Portland, said that although the volcano has not affected business, people were concerned about ash and they were stepping up their preventative program. He explained, "That means mainly coming out and changing filters."

Gene Morley, manager of Genetra Computer Services in Vancouver, Washington, just northwest of Portland said the eruptions didn't have any affect on them at all. He did say he saw a memo advising computer owners to take extra precautions during a blast, but he has not found that necessary.

One other computer company executive in Vancouver commented that the ash from the volcano was really nothing special, that some ash has been tracked inside, but it's like sand at the coast.

Protecting Equipment

Service managers generally advised home computer owners to cover their equipment when volcanic fallout can be tracked inside; and people whose systems have disk drives were advised to regularly change their filters. It was also suggested that dust can be blown off screens and other exterior surfaces cleaned with a damp cloth.

John Gough of Micro-Data in Portland, explained they service business microcomputers and most systems are in separate rooms with good filtration systems, so dust in the air hasn't been too much of a problem. He said the computer systems with larger metal disks don't have problems, but the disk drive in smaller systems would be vulnerable to ash and he recommended changing the filters of those systems regularly, keeping the floppy disks as clean as possible.

Wendell Cockrell, an analyst with Prime Computer Company in Portland, said that following the October eruptions their field engineers advised customers with disks to keep their blowers running to protect their equipment. He explained that if the blowers are turned off, ash could get on the disk surface and damage the recording surface and portions of the computer that read data. Cockrell said their engineers also advised checking air conditioning and changing filters.

He added he didn't believe it was necessary to turn equipment off and knows of many successful operations in sandy, dusty areas.

Cockrell said a lot of people called asking about keyboards and they were advised to definitely cover them when they are not in use. He also advised owners of home computers to cover their machines with an impervious piece of plastic.

Finally, reports of a third and larger dome appearing in the crater and seismic recordings, indicate that pressure from molten rock is moving up, causing scientists to warn that the volcano could blow again. ■

PATRICIA SMITH is a freelance writer based in the San Francisco Bay Area.

80 Software Directory
ComputerMat
Box 1664A
Lake Havasu, Arizona 86403
Published 3 times a year.
$7.00 per issue.

ComputerMat has just completed their fifth edition of the 80 Software Directory for the TRS-80. This latest edition has a new section devoted to Model II software. Their listings include the areas of business, mathematics and utility software packages. CPM and CBASIC packages are also included. The directory gives the title of the package, a short description of it, the vendor, language requirements (BASIC type and operating system required), memory requirements, media available, drives needed if it is disk-based software and the price. A very useful directory and well worth the $7.00 for the listing.

Of course, it would have been very helpful if ComputerMat had included a short section defining the codes used in the directory and how to use it. The true novice may be somewhat confused, but anyone with a little experience will realize that c = cassette, d = diskette (under media). The other point to remember about the directory is that it only lists where the programs can be purchased. That does not mean that they are necessarily all high quality programs. You should evaluate each program independently. Noting these points to watch out for, I still found this directory very useful.

Reviewed by M. Dundee Maples

Crash Course in Microcomputers
Louis E. Frenzel Jr.
Howard W. Sams and Co., Inc.
4300 West 62nd Street
Indianapolis, Indiana 46268
1980, 264 pages, $17.50.

This is an excellent book for those of you who are just becoming familiar with microcomputers. The text is laid out in a simple, self-teaching style. You are asked to fill in answers at least every few sentences to ensure that you are absorbing the information being presented. The book is divided into 14 sections: two on central processing units, two on memories, three on input and output devices, four on software, an introductory section on microcomputer basics, another section on binary data and a final section that briefly covers microcomputer applications (this section just lists some application areas but does not go into any detail).

Even though some of the concepts, like stack usage and interrupts, will need more explanation before the novice will understand them, the text presents a very reasonable introduction to them. Many, many new terms and concepts are introduced to the reader. For those who would desire to become reasonably proficient in the useage and application of microprocessors, it will require further effort on their part. I recommend this text along with access to a friend who has knowledge of microcomputers and can explain in detail any questions that arise while you use this text. Also, it would be beneficial to use this text to introduce yourself to micros before you enroll in a microprocessor training course. It would allow you to generate reasonable questions and a better level of understanding, thus allowing you to learn more from the course.

I recommend this book to those of you with no knowledge or experience with microcomputers but not for those of you who have had any reasonable experience (i.e., programmed a microcomputer in both assembly and some high-level language, and generally used a microcomputer system in even the most mundane of situations involving hardware interfacing).

Reviewed by M. Dundee Maples

Software Manual for the Elementary Functions
William J. Cody Jr. and William Waite
Prentice Hall, Inc.
Englewood Cliffs, New Jersey
1980, 269 pages, $16.95.

This book of algorithms gives detailed descriptions of several common functions: square root, logarithm, exponential, power, sin/cos, tangent/cotangent, arcsin/arccosin, arctangent and sin H/cos H. The discussions of these functions include generation of basic algorithm, flow chart of algorithm, notes on implementation concerned with overflow and general accuracy problems for both fixed point and floating point machines, and general algorithm test procedures. All in all, this book is an excellent aid for those of you who are becoming involved with algorithm design and testing. It could become a beneficial addition to your library. ■

Reviewed by M. Dundee Maples

Editor's Note: We need reliable people to review software and books for this magazine. If you are interested, please send us your name, address, occupation and area of interest or expertise.

The Pirate's Life for Me

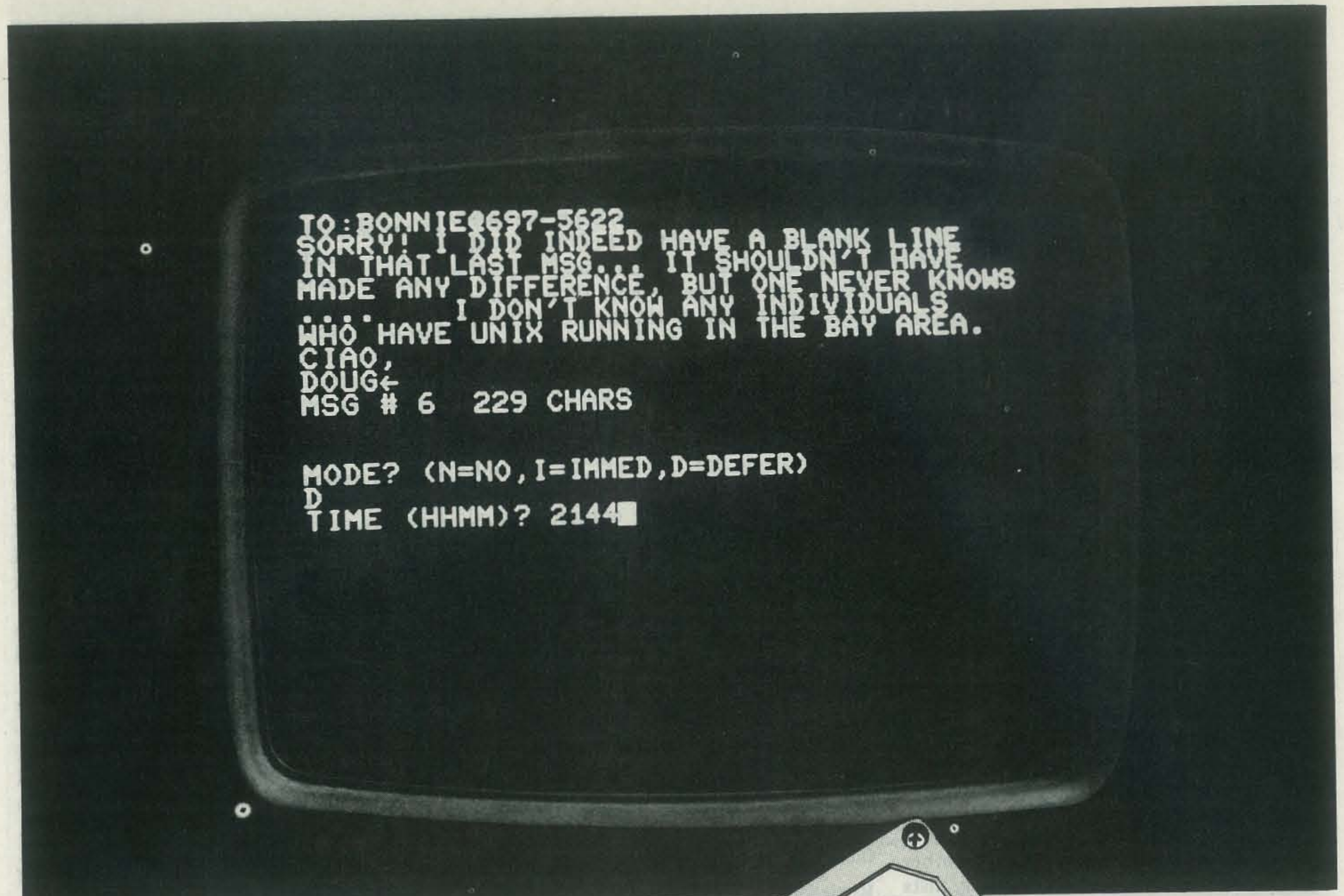
(continued from page 35)

but I could certainly enjoy playing the lottery game. Only, unlike other people, I would win every month, not more often – that would be too suspicious – and with a ticket bouth in a different area each time. Of course, I buy losing tickets, too. But my one monthly $10 winning ticket does help take care of a few of the necessities. And this scheme has a big plus: there's no one who will complain a few weeks later when the bills come; no loser will know, and the amount is small enough not to cause any weird statistical aberrations in the main computer. What fun!

Now it's your turn. Can you come up with the perfect computer crime? Like all good crimes, there are rules:

1. The proceeds should be such that a person or family can live on or obtain some important service for the rest of their lives, with an estimated probability of capture of 1 in 1,000,000 or less over their lifetime. This means that a big enough heist requires 1 in 1,000,000 chance of detection, but a heist made up of a series of small payoffs probably has to offer odds of 1 in 1,000,000,000 or so on each "transaction."
2. The crime should be done entirely by fooling some computer system. If the proceeds are other than money, you should be able to negotiate them legitimately without having to present forged ID or something like that.
3. You can assume that electric lines, wherever they are, can be tapped. But the crime should require at most a year or two of planning. If the electric lines you are tapping are encoded in a way that would take many years to decode, you'll have to come up with another plan. Of course, if the encoding is fairly simple like the new national standard for digital encryption, you can assume that computers will soon be able to break the code reliably and quickly.
4. If employees of the party you are dealing with must help you, it should be because they are willing agents on their own; i.e., the computer told them to help you. You shouldn't have to be in cahoots with someone on each transaction; this, of course, also increases the probability of being caught.

Well, can you do better than I? Whoops, lunchtime! Hmmm, I seem to be short a few dollars. Ah, well, there's an automatic teller just around the corner. ■



PCNET PAN

Personal **C**omputing **NET**work

Personal Computing NETwork (PCNET), a project of People's Computer Company, has available computer mail support software for the Commodore PET®. Other versions (including Apple) will be available shortly.

The new PCNET computer mail system is called PAN – a program on cassette tape for use with an 8K or larger PET. All that is required is a telephone line, an auto dial – auto answer modem, and a personal computer (available at present only for the PET).

The PAN software, a perpetual license for its use and a user's manual sell for $12; a user's manual is available separately for $2.

If you would like more information on the PCNET project, or would like to order the PAN software, contact People's Computer Company, 1263 El Camino Real, P. O. Box E, Menlo Park, CA 94025. (Send no money – a perpetual license agreement must be signed first.)

Computer Literacy

Finding Effective Resources

by Barbara Kurshan

Although the "computer revolution" is being experienced by the total society, it may be most profoundly felt by the educator. The microcomputer is rapidly becoming commonplace in the classrooms from kindergarten to high school. However, the arrival of the computer in education may be happening faster than the instructional community can publish textbooks, design effective curriculum, train teachers and develop computer software. Therefore, teachers must "scrounge" for resources, programs and books for reference and student use. If teachers are to provide computer literacy for students, they must have the materials to become computer literate themselves and to develop computer lessons. After having spent the better part of the last eight years trying to define computer literacy, I have produced the following resource list.

The references are presented by topic with general resources listed first and a comprehensive list of films, fictional computer stories, journals, games and organizations at the end. The headings are taken from the results of a computer literacy study prepared by an education subcommittee of the Association for Computing Machinery (Johnson, David C., et al., "Computer Literacy – What Is It?," *The Mathematics Teacher*, Vol. 73, No. 2, February, 1980). The objective headings are:

- **I. General**
- **II. Hardware**
- **III. Programming and Algorithms**
- **IV. Software and Data Processing**
- **V. Applications**
- **VI. Impact**
- **VII. Attitude, Values and Motivation**

Each topic is coded by level according to the following key: E – elementary; I – intermediate; S – senior high and T – teacher reference.

I. General

DeRossi, Claude. *Computers: Tools For Today*. Chicago, Illinois: Children's Press, 1972. (E, I) Gives simple information about binary addition, bits, punched cards, card readers, magnetic tape, programming, programmers, flow charts, and a little history.

Doerr, Christine. *Microcomputers and the 3 R's*. Rochelle Park, New Jersey: Hayden Book Company, Inc. (T) Practical guide for teachers who want to get involved in computing. Has reliable suggestions on selecting a unit, is of value to users and administrators also.

Dorf, Richard. *Introduction to Computers and Computer Science*. San Francisco: Boyd and Fraser Publishing, 1972. (T) General introduction to the computer, comprehensive exercises and examples.

Edwards, J.B., Ellis, A. S., Richardson, D.E., Holznagel, D. and D. Klassen. *Computer Applications in Instruction: A Teacher's Guide to Selection and Use*. Hanover, N.H.: Time Share Corporation, 1978. (T) A general introduction to uses of computers in education. Includes the essentials of hardware; instructional uses; selecting computer curriculum; readings on computers in the schools.

Harris, Diana (ed). *Proceedings of the National Educational Computing Conference*. Iowa City: University of Iowa, Weeg Computing Center, 1979. (T) A collection of papers presented at the first NECC. All educational levels and disciplines are covered.

Rice, Jean. *My Friend – The Computer*. Minneapolis, Minnesota: T.S. Denison and Company, Inc., 1976. (E, I) This is a very simplified explanation of the computer, its uses, development, operation, input procedures, flow charts, programming and terms.

Spencer, Donald D. *The Story of Computers*. Ormond Beach, Florida: Camelot Publishing Company, 1977. (E, I) This book is a clear, easy-to-understand introduction to computers.

Wall, Elizabeth S. *Computer Alphabet Book*. Nokomis, Florida: Bayshore Books, 1979. (E) This book is the first of a series of a "Beginning Computer Literacy" series. It is an introduction to computers with alphabetized, simple definitions and explanations of computer parts, terms, etc.

Willis, Jerry. *The Peanut Butter and Jelly Guide to Computers*. Creative Publications, 1980. (I, S) A simple introduction to computers. It explains what a computer can do and how it does it.

General Educational Programs

Dunlap, Mike, and Morsund, David. *Computers in Education Resource Handbook*. Eugene, Oregon: University of Oregon, 1975. (T) A good general resource. Areas covered are computers in education, teaching about computers, the computer as an aid to learning, computer as a teacher, computer as a classroom management tool, and administrative uses of computers. Some sections are directed at administrators, some at teachers, and some are of general interest to the student.

Kurshan, Barbara. *Computer Literacy: Practical Ways to Teach the Basic Mathematical Skills*. Richmond, Virginia: Virginia Council of Teachers of Mathematics, 1978. (E, I, S, T) A curriculum guide for computer literacy. Includes goals and activities.

Kosel, Marge. *Elementary . . . My Dear Computer*. Lauderdale, Minnesota: Minnesota Educational Computing Consortium, 1978. (T) This book has many lesson plans and programs for use in the elementary classroom using the computer. The chapters cover an introduction to the computer, teacher aides for worksheets and teacher assistance, learning activities and games and activities for students to increase their knowledge of computers.

Moursund, David. *Calculators, Computers and Elementary Education*. Salem, Oregon: The Math Learning Center, University of Oregon, 1977. (T) This book gives teachers in the elementary schools an introduction and activities to teach calculators and computers. It includes information on using a calculator, problem solving, functions and formulas, calculator memory, calculator applications to elementary education, and computer literacy.

Ricketts, Dick (Project Director). *Course Goals in Computer Education* K-12. Portland, Oregon: Commercial Educational Distributing Services, P.O. Box 8723, 1979. (T) Goals for use in planning and evaluating Elementary and Secondary school curricula in computer education.

II. Hardware

Ball, Marion J. *What Is A Computer?* Boston, Mass.: Houghton Mifflin Company, 1972. (E, I) This book covers the areas of what is a computer, history, parts of the system, how software is made. It includes a summary, glossary, and index.

Ball, Marion J. and Charp, Sylvia. *Be A Computer Literate*. Morristown, New Jersey: Creative Computing Press. (E, I) An introduction to the computer world for children. Full color diagrams, drawings and large type make this book easy to read and use.

Berger, Melvin. *Computers*. "Science is What and Why Series." Wisconsin: E.M. Hale and Company, 1972. (E) A simplistic presentation of the computer. It covers input, output, control, and memory briefly.

Berger, Melvin. *Those Amazing Computers!* Day, 1973. (E, I, S) Illustrated with photographs and organized by uses. Includes bibliography, brief material on input, output, control unit, programming, flow charts, memory, and data banks.

D'Idgnazio, Fred. *Katie and the Computer*. Morristown, New Jersey: Creative Computing Press, 1978. (E) A picture book adventure that explains how a computer works to a child. It is both an exciting story that a child will want to read and a simple explanation of computers.

Kenyon, Raymond G. *I Can Learn About Calculators and Computers*. New York, New York: Harper and Row, 1961. (E, I, S) This book is a "how to build your own" and includes history and "how to" about computers.

Meadow, Charles. *The Story of Computers*. Harvey House, 1970. (E, I) Simple and clear information on computers. A glossary, index, bibliography and table of contents are included.

Rusch, Richard B. *Computers: Their History and How They Work*. New York, New York: Simon and Schuster, 1969. (S, T) The book gives a clear idea of the computer's role, its physical equipment, and also current and future computer applications. (It is not a primer, and does not go into detail on programming.)

Srivastava, Jane Jonas. *Computers.* New York, New York: Thomas Y. Crowell Company, 1972. (E) This short book, in the "Young Math Book" series, explains the computer as a counting machine. Other topics covered are programmers, languages, input, printouts, control unit, flow charts, uses of computers and history. The information is presented very simply.

III. Programming and Algorithms, and IV. Software and Data Processing

Albrecht, Bob. *My Computer Likes Me When I Speak in BASIC.* Menlo Park, California: The People's Computer Company Bookstore, 1972. (E, I) An introduction to BASIC in a very elementary manner.

Albrecht, Bob. *Teach Yourself BASIC,* Volumes 1 and 2. Menlo Park, California: People's Computer Company Bookstore. (E, I, S) A BASIC primer.

Dwyer, Thomas A., and Kaufman, Michael S. *A Guided Tour of Computer Programming in BASIC.* Boston, Mass.: Houghton Mifflin Company, 1980. (I, S, T) This book is on communicating in the BASIC Language and showing the student how to use the computer. The BASIC vocabulary is introduced and key words and explanations of input and other commands that are used are given. There is also a section on two-dimenstional arrays, "Library" functions.

Dwyer, Thomas and Critchfield, Margot. *BASIC and the Personal Computer.* Addison-Wesley Publishing Company, 1979. (S, T) An excellent book with which to learn BASIC. Covers graphics in addition to elementary commands. Many sample programs are included.

Engel, C.W. *Stimulating Simulations.* Creative Publications. (S, T) The book is a series of real-world simulations. For each of the 12 simulations a listing, flow chart and sample runs are included.

Graham, Neill. *The Mind Tool: Computers and Their Impact on Society.* 2nd Edition. St. Paul, Minnesota: West Publishing Company, 1980. (I, S) A review of computers with short and informative chapters on computer applications in many different areas, as well as an introduction to BASIC.

McQuigg, James D. and Alto M. Harness. *Flowcharting.* Boston, MA: Houghton Mifflin Company, 1970. (I, S) A workbook on flowcharting.

Moursund, David. *Basic Programming for Computer Literacy.* New York: McGraw-Hill, Inc., 1978. (S) A text module for the computer programming of computer literacy instruction. It teaches BASIC.

Rogowski, Stephen J. *Computer Clippings.* Creative Publications. (I, S) The book includes 50 problems with a sample solution for each. Graphics are included.

Spencer, Donald D. *Using BASIC in the Classroom.* Ormond Beach, Florida: Camelot Publishing Company. (T) Aimed at the beginning teacher of the BASIC computer langauge.

Stern, Nancy. *Flowcharting – A Self-Teaching Guide.* New York: John Wiley, 1975. (I, S) A clear introduction of the use of flowcharting.

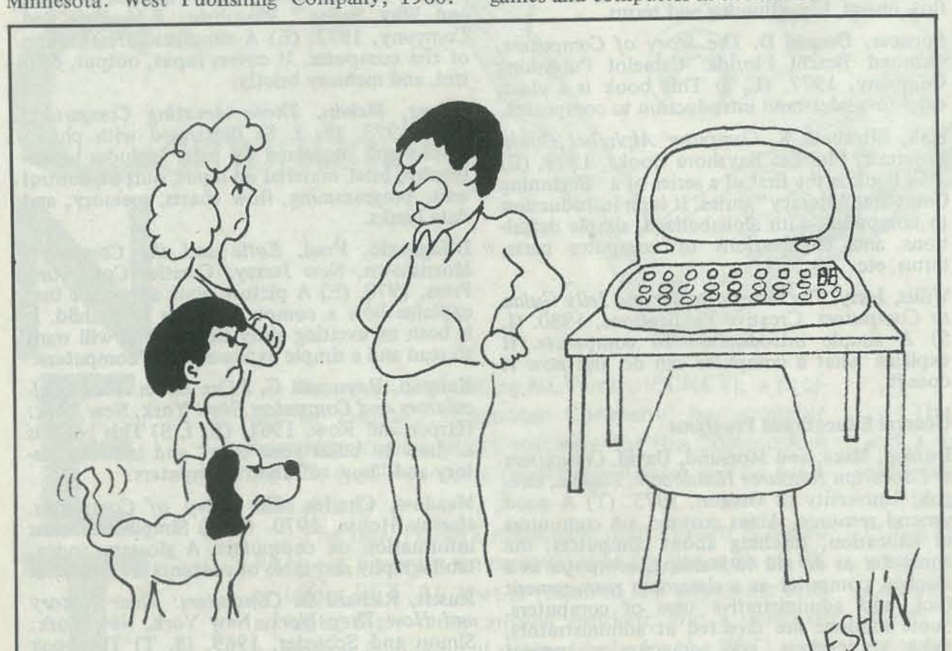
Wadsworth, Nat. *Introduction to Low Resolution Graphics.* Creative Publications. (I, S) The book gives an introduction to graphics that can be applied on the Apple, PET or TRS-80. Projects include simple shape programs through animation.

Waite, Michael. *Computer Graphics Primer.* Creative Publications. (S, T) The book introduces graphics. It includes plotting lines, curves and special characters. It is written for Apple high-resolution graphics.

V. Applications (primarily for education)

Ahl, David H. *BASIC Computer Games.* Maynard, Mass.: Creative Computing. (I, S) This book has a variety of computer games with descriptions of each and computer limitations.

Ahl, David H. *Getting Started in Classroom Computing.* Maynard, Mass.: Digital Equipment Corporation, 1974. (I, S) The games in this booklet are to introduce the newcomer to using games and computers in the classroom.



". . . and what's more it obeys my commands with no back talk."

Albrecht, Robert, et al. *What To Do After You Hit Return.* Menlo Park, California; People's Computer Company Bookstore, 1975. (I, S) This book is a collection of games, with a list of creative use of games and computers as an important learning tool is stressed. It also gives a starting point for using the computer. It includes board games, teasers, business and social science simulations. Also given are pattern games and hide and seek games.

Braude, Michael. *Larry Learns About Computers.* Minneapolis, Minnesota: T. D. Denison and Company, Inc., 1969. (E) This book explains, in a simple manner, about the history, uses, and terms in computers.

Computer Based Education: Mission of the Future. Proceedings, 1979. San Diego, California: Islandia Hyatt House, 1979. (T) This book is on developing an elementary school problem-solving curriculum in CAI.

Ellis, Allan B. *The Use and Misuse of Computers in Education.* Harvard University: McGraw-Hill, 1974. (T) This book points out that sometimes computers are misused in schools. The computer is defined in the book, and some uses are given. A computer system for guidance is given as an example of a project.

Hatfield, Herts, Advisory Unit for Computer Based Education. *Computer Aided Learning.* Teacher Training. (T) Catalog of computer progress that introduces teachers to the computer. Six groups include Biology and Nutrition, Physical Sciences, Geography and Economics, Mathematics, and Miscellaneous.

Johnson, M. Clemens. *Educational Uses of the Computer: An Introduction.* Chicago: Rand McNally and Company, 1970. (T) This book is a concise and non-technical introduction to the variety of uses of the computer in education. It contains chapters on student data, computer assistance in instruction and in research, classroom problem solving, assisting the teacher in classroom management, and research skills in education.

Kohn, Bernice. *Computers at Your Service.* Englewood Cliffs, New Jersey: Prentice Hall, 1962. (E) Seventy-two pages with illustrations on the computer by Aliki.

VI. Impact

Baker, Eugene. *I Want To Be A Computer Operator.* Children's Press, 1973. (E) An excellent story about a computer career.

Billings, Karen, and Moursund, David. *Are You Computer Literate?* Eugene, Oregon: University of Oregon, 1977. (E, I) This book discusses things one needs to know about the computer, such as why computers exist, about the data and programming, the uses of computers, the effects on people. It uses illustrations and has pre-tests for each chapter, chapter tests, and suggests books to read and projects to do.

Marriottes, John, and Quasney, James S. *Computer Careers – Planning, Prerequisites, Potential.* Hayden, 1974. (T) This book offers answers to questions about careers in electronics and computing opportunities, educational and training opportunities, schools, financing, etc.

Ray, Jo Anne. *Careers in Computers.* Minneapolis, Minnesota: Lerner Publications Company, 1975. (E) A career book with photographic format, including sex stereotypes and information relating to preparation for the various jobs. This book is easily read, colorful, with a serious look at careers.

Rothman, Stanley, and Mosmann, Charles. *Computers and Society.* Chicago, Illinois: Science Research Associates, Inc., 1972. Grade level is 9-12. (S, T) A general review of computers today.

Rusch, Richard B. *Man's Marvelous Computer: The Next Quarter Century.* New York, New York: Simon and Schuster, 1970. (E, I, S) This book is a discussion of the issues in which computers will play a role within the next quarter century; only a brief introduction describes how they work.

VII. Attitude, Values and Motivation

Bailey, Richard. *Computer Poems.* Drummond Island, Michigan: Potagannissing Press, 1973. (S, T) Collection of poems either computer-generated or inspired from random computer-generated "thoughts."

Leavitt, Ruth (ed.). *Artist and Computer.* Morristown, NJ: Creative Computing Press, 1976. (T) Good review (with many pictures) of the computer's role in the artistic process and graphics.

Nelson, Ted. *Computer Lib/Dream Machine.* South Bend, Indiana: Hugo Book Service, 1973. (I, S) Covers all sorts of ideas on computers and their uses. Creatively displayed and fun to read.

The Colossal Computer Cartoon Book. Morristown, New Jersey: Creative Computing Press. (I, S, T) Several hundred cartoons about robots, computer dating, computers in the office, etc.

Van Tassel, Dennie L. *The Compleat Computer.* Santa Cruz, California: University of California, 1976. (S, T) This book includes fiction, poetry, newspaper articles, cartoons, etc. giving many opinions about what non-computer specialists think about computers. Present uses, potential, and government controls are dealt with in the essays.

VIII. Other Resources

A. Films

Bibliography: Lidtke, Doris. *Computer and Computer Applications: A Film Bibliography.* Portland, Oregon: Oregon Council for Computer Education, 1977. A list of computer films.

A Day in the Life of a Computer, J. Weston Walsh, Publisher, Portland, Maine 04104.

Art from Computers, 8 minutes, Control Data Corporation. (E, I, S)

At Home 2001, 24 minutes, New Hyde Park, New York, Modern Talking Pictures.

Careers in Computers, Pathescope, Educational Films.

The Catalyst, 13 minutes, Sperry Univac, Blue Bell, Pennsylvania.

Clinical Uses of the Computer, Control Data Corporation, Minneapolis, Minnesota.

The Computer, 11 minutes, Control Data Corporation, Minneapolis, Minnesota. (E)

Computer Revolution Part 1, New Hyde Park, New York, Modern Talking Pictures.

Computer Revolution Part 2, New Hyde Park, New York, Modern Talking Pictures.

IBM: Close Up, 21 minutes, New Hyde Park, New York, Modern Talking Pictures.

IBM World's Fair Puppet Show, 10 minutes, New Hyde Park, New York, Modern Talking Pictures.

The Information Machine, 11 minutes, Modern Talking Picuters. (E, I, S)

Introduction to Digital Computers, 24 minutes, Sperry Univac. (E, I, S)

The Endless Revolution, Sperry Univac. (S)

Thinking?? Machine, 19 minutes, BELL Telephone. (E, I, S)

2001, 25 minutes, Rarigs, Seattle, Washington. (E, I, S)

Year 1999 A.D., Ford Motor Company, B.U. Film Library.

B. Fiction

Christopher, John. *The White Mountains.* MacMillan, 1967. (E, I)

Del Rey, Lester. *The Runaway Robot.* Westminster, 1965. (E, I)

Fairman, Paul W. *The Forgetful Robot.* Holt Rinehart and Winston, 1968. (E, I)

Hayes, William. *Hold That Computer!* Atheneum, 1968. (S)

Norton, Andre. *The Beast Master.* Harcourt Brace, 1965. (I, S)

Philbrook, Clem. *Ollie's Team and the Football Computer.* Hastings, 1968. (E, I)

Steadman, Ralph. *Little Red Computer,* Steadman. McGraw, 1969. (E, I)

Van Tassel, Dennie L., editor. *Computers, Computers, Computers!* in Fiction and in Verse. New York, New York: Thomas Nelson Inc., 1977. (S, T) This is a collection of eleven short stories, fiction, non-fiction, and verse on computers.

C. Journals

AEDS Journal, Association of Educational Data Systems, 1201 Sixteenth Street N.W., Washington, DC 20036.

Computers and Education, Pergamon Press, Inc., Maxwell House, Fairview Park, Elmsford, New York 10523.

Creative Computing, by Creative Computing, Post Office Box 789-M, Morristown, New Jersey 07960.

edu, The Education Products Group Magazine of Digital Equipment Corporation, Educ Products Group, DEC, Maynard, Massachusetts 01754.

Journal of Educational Data Processing, by IDEOS, P.O. Box 867, Soquel, California 95073.

The Computing Teacher, Computing Center, Eastern Oregon State College, LaGrande, Oregon 97850.

Recreational Computing, People's Computer Company, P.O. Box E, 1263 El Camino Real, Menlo Park, CA 94025.

SIGUE Bulletin, Computer Uses in Education, Association for Computing Machinery, 1133 Avenue of the Americas, New York, New York 10036.

T.H.E. Journal, Information Synergy, Inc., P.O. Box 992, Acton, MA 01720.

D. Games

Creative Computing. *Computer-Rage.* Morristown, New Jersey: Creative Computing Company, 1977. A computer game for two to four players, ages 10 to adult. The players attempt to move their players (Programs) from Start ("Input") to Finish ("Output"). The winner is the first to accomplish the goal. The game involves skill and luck. There are many parallels between the play of the game and the operation of a computer.

DATAFLOW. Glen Burnie, Maryland: Dataflow Systems, Inc. of Baltimore, 1978. A game for two to four players, ages 12 to adult. The players compete for the high score by performing computer instructions as they select strategic paths throught the flowchart. Authentic symbols are used in the game. The game is challenging, requiring concentration, but no technical knowledge. It is also educational as it reinforces concepts in logic, arithmetic, and data processing.

Hagelbarger, David and Fingerman, Saul. *CARDIAC,* Bell Telephone Laboratories, Inc., 1968. A cardboard Computer that works like a simple computer from input through output. An excellent handbook accompanies the machine and includes sample programs and exercises.

Matt, Fred C. *A Real Programmable Paper Computer: Instructo.* Paoli, Pennsylvania: McGraw-Hill, 1979. A game designed to give students an educational tool they can learn with and use basic skills in computer functions and programming.

Think a Dot

Think a Dot. Commack, New York: Edu-Cards Corporation. A computer game that is fun. It requires the player to drop a marble in one of three holes in a little "computer." and trying to make all of the displayed spots blue, or yellow by repeatedly dropping in the marble and developing a sequence of logic gates.

D. Organizations

American Federation of Information Processing Societies; 210 Summit Avenue, Montvale, New Jersey 07645.

Association for Computing Machinery; 211 East 43rd Street, New York, New York 10017.

Association for Educational Data System; 1201 Sixteenth Street, N.W., Washington, DC 20036.

Minnesota Educational Computing Consortium (MECC); 2520 Broadway Drive, Lauderdale, Minnesota 55113.

National Council of Teachers of Mathematics; 1906 Association Drive, Reston, Virginia 22091.

Oregon Council for Computer Education; 4015 SW Canyon Road, Portland, Oregon 97221.

Conclusion

Although many of the resources presented were published early in the seventies and several were difficult to locate, they are still useful because they are the only alternatives and they do point out that many resources are available if searched for unrelentingly. However, most teachers do not have the time to locate and then critique obscure resources. This dilemma provides a challenge to the educational community to develop and provide widespread, easily obtained and effective computer literacy materials. The challenge is slowly being met by some of the organizations listed in this article but until "tomorrow" perhaps this article will help provide resources for computer literacy "today."

■

BARBARA KURSHAN is the Director of Academic Computing at Hollins College, Virginia 24020.

Move Over Bach, Beethoven, and Brahms

Composing Music Through Computers

by Neil C. Rowe

Contrary to popular belief, most of what composers do is not thinking up new music or new sounds, but rather taking existing music and modifying it, disassembling and rearranging it in new ways. Here are some interesting things to try along these lines, just with melodies.

A computer is helpful in playing around with music because music happens in time, and things that do are hard to analyze. To do the things described below, it isn't necessary that your computer play music itself, though that would be nice. Just represent melodies as lists of pitch (duration pairs) and have the computer print out the transformed lists. Code pitches as integers, using the piano keyboard (0 = middle C, 1 = middle D, -1 = middle B, etc.) and code durations as integers representing tenths of a second.

Most of the following is straightforward in any computer language with string processing facilities. See the Appendix for a partial implementation in LISP.

Classical Transformations. There are a few transformations that composers use a lot because they tend to preserve "musicality" and are fairly easy to decipher without a computer. They include:

1. Augmentation: multiply all durations by some number > 1.
2. Diminution: divide all durations by some number > 1.
3. Accelerando: subtract a progressively increasing number from the durations.

Neil C. Rowe is completing a Ph.D. in Computer Science at Stanford University. He has S.B., S.M., and E.E. degrees from MIT and his primary interests are artificial intelligence and data bases.

4. Ritardando: add a progressively increasing number to the durations.
5. Scale Change: interpret pitch numbers as a different scale (examples: major, minor, modal, pentatonic, chromatic scales; invent your own).
6. Transposition: add some number to all pitches.
7. Inversion: subtract all pitches from some number.
8. Retrograde: reverse the order of the notes.
9. Add "ornaments" to particular notes (look these up if you don't know what they are). For example, there are trills, turns, mordents, appoggiatura, suspensions, anticipations and passing tones.
10. Vary the loudness of notes in some non-random way.
11. Replace every note by a motif (set of other notes) appropriately transposed.

Quasi-Contour-Preserving Transformations. The "contour" of a melody (the shape of the curve of pitch versus time) is very important to its musicality. Thus, of the non-classical transformations, ones that don't change the contour very much often have a better chance of sounding interesting. For instance:

1. Bandwidth compression or expansion: multiply all pitches by some number.
2. Take the average of successive pitches (perhaps repeat several times).
3. Add corresponding pitches of two different melodies or pieces.
4. Interchange randomly chosen adjacent notes in a melody.
5. Add (or subtract) a constant number to (from) the pitches of randomly selected notes in a melody.
6. Do the same but for notes at the same place in each measure of a melody.
7. Add (or subtract) a constant number to (from) all durations.
8. Interchange the durations of notes, measures, or sections of a melody while leaving the order of the pitches the same.
9. Use the durations from one melody as the corresponding durations of the other.

There are other transformations that can be made. For example:

1. Transpose different notes, measure, or sections of the same melody, based on same rule.
2. Replace pitches by some function of the pitch and its associated duration.
3. Invert the melody at random places in it.
4. Harmonize the melody, then erase all but a note randomly selected from each chord.

Figure A. The original melody, the chorus of a familiar sea chantey.

Figure B. Major scale inversion around E.

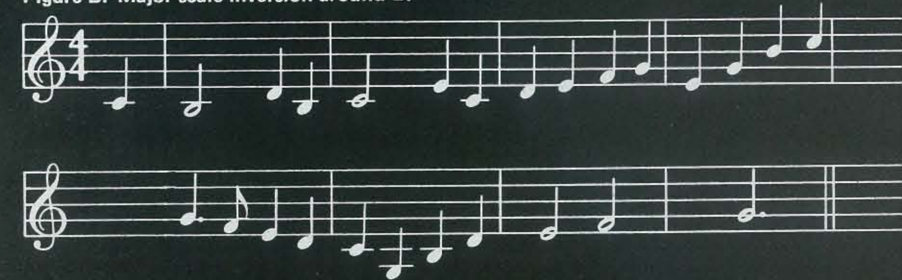


Figure C. Retrograde.

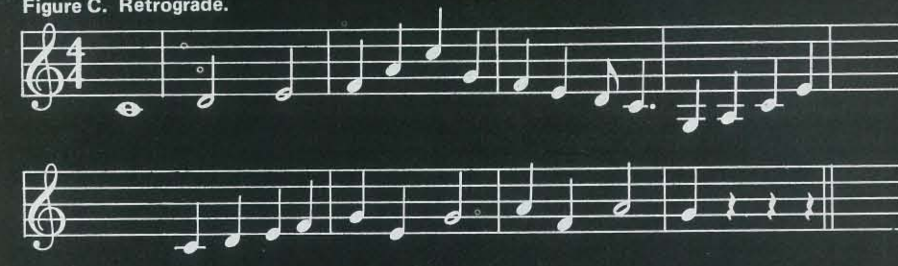


Figure D. Addition of passing tones to every melody interval that is not a step.

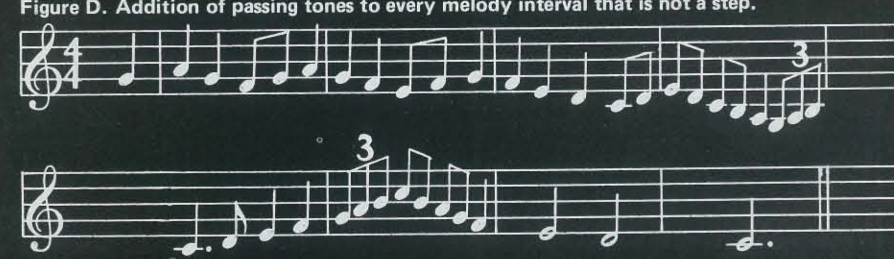


Figure E. Each pitch averaged with the preceding.

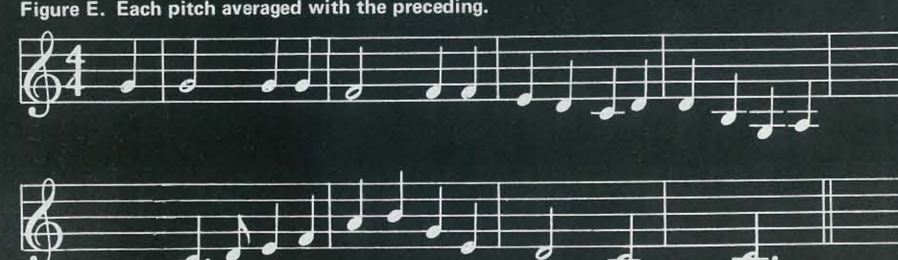


Figure F. Change of scale from major to pentatonic (by multiplying all pitches by 12/7).

Figure G. Conversion to corresponding chromatic scale pitches, then multiplication of pitches by 2 (result represents notes of the chromatic scale).

Figure H. Two scale steps added to the first note of every measure.

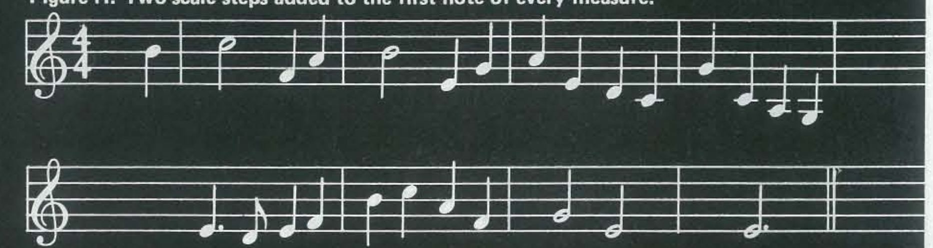


Figure I. Deletion of a random note from each measure.

Figure J. Permutation of the order of measure rhythms, without changing the order of the pitches.

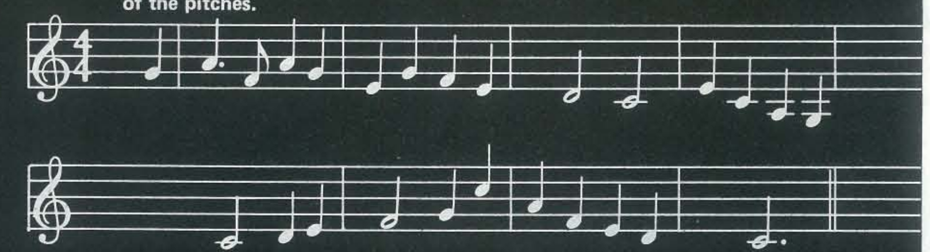


Figure K. Permutation of the order of the measures, repeating a few randomly.

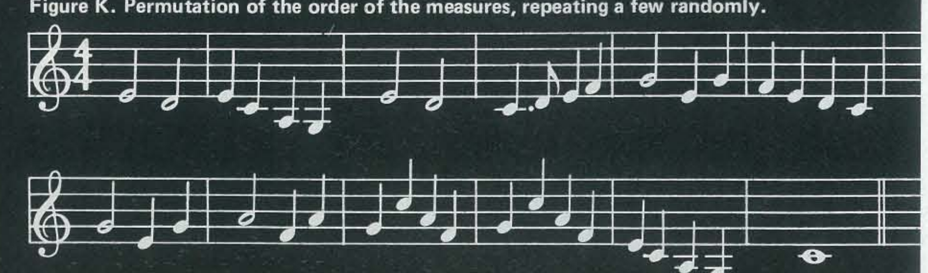
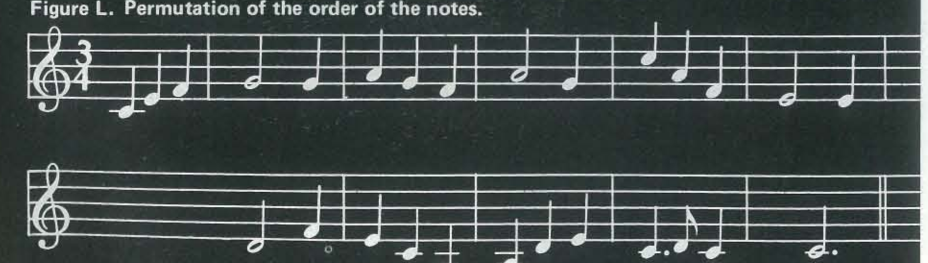


Figure L. Permutation of the order of the notes.



5. "Anagrams": interchange random notes, measures, or sections within a melody.
6. Alternate notes, measures, or sections of two melodies (which should be somewhat similar to one another).
7. Delete random notes, measures, or sections from a melody.

As further possibilities, you can apply the same transformation several times in successsion, or different transformations in series.

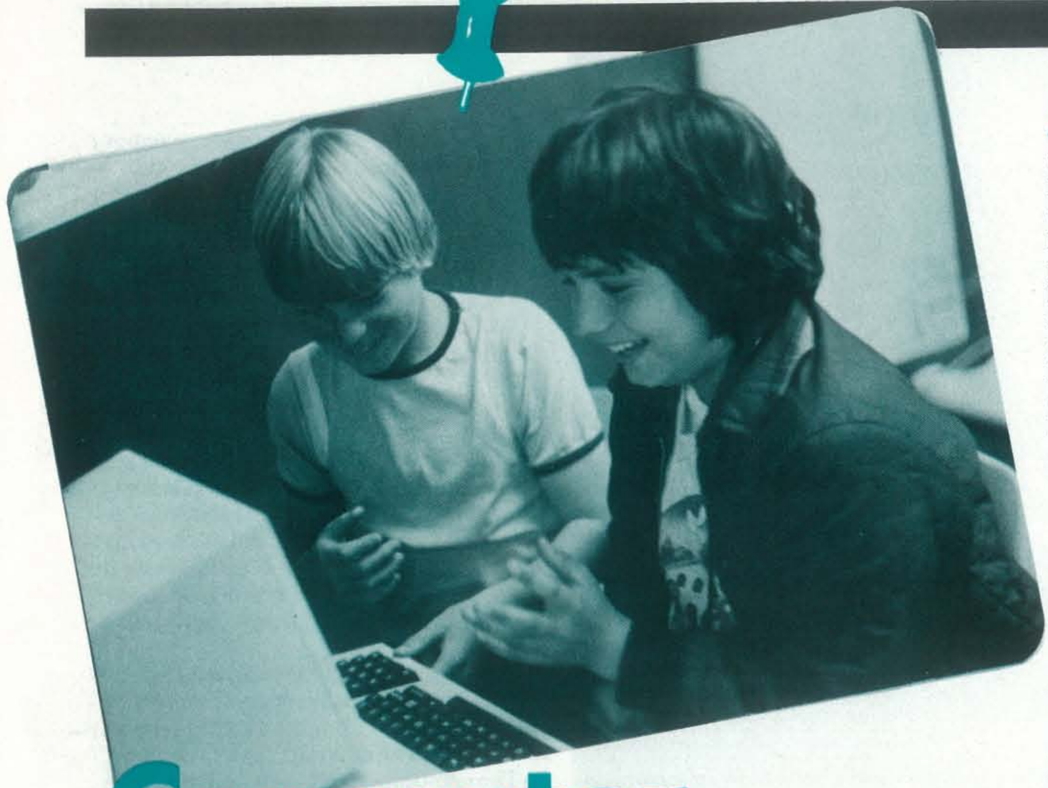
Separating the Wheat from the Chaff. Some of the results of the preceding experiments sound better than others. We could play each result, and only copy down the ones we like. But this might mean a lot of boring listening to music that starts to sound pretty much alike after a while.

So it's a good idea to have a few criteria to guess which are the interesting ones beforehand. Here's how the computer can help. The computer can look for:

1. The smoother the contour of the melody (the fewer changes in direction) — but not too smooth.
2. The smaller the intervals, on the average, in the melody — but not too small.
3. The smaller the total pitch range of the melody (the more "singable") — but not too small.
4. The fewer the successive repeated pitches of the melody.
5. The more the tonal "centeredness" on one particular pitch and its octaves; or a particular triad and its octaves.
6. The simpler the possible harmonizations could be (the few the number of different chords, or the rate of chord changes).
7. The more repeated measures or sections, especially when repeated at a "regular" interval — though not when repeated too much.
8. The more repeated rhythmic patterns, especially when repeated at a "regular" interval.
9. The more "final" the ending sounds (like whether it ends on the primary note (tonic) of the melody, and has a long duration on the last note).

Musical Examples. As an example of what you can do with these ideas, I've taken a melody, whose pitch numbers are the notes of the major scale, and applied several transformations to it. See Figures A-L and the Appendix showing the LISP implementation on pages 54 and 55.

Editor's Note: LISP or List Processing is an interpretive language, developed for manipulation of symbolic strings of recursive data; i.e., used to develop higher-level languages. It is often used in the area of Artificial Intelligence. ■



Computer Games in the Classroom

by Glenn Fisher

A vast majority of programs available for microcomputers are games. For home use, these games provide hours of entertainment. However, games also have a place in the classroom. They can provide drills, motivate students to learn material or social behaviors, create interest and cooperation, develop problem-solving and thinking skills, and lengthen attention span. To discuss the educational usage of computer games, I have divided them into four arbitrary categories: entertainment (arcade), logic (including board and card), tutorial (teaching), and simulation.

Entertainment games are arcade-type games that provide from minutes to hours of enjoyment, but have little educational value. Some examples are "Space Wars," "Road Race," "Breakout," and "Invaders." They are a wonderful way to introduce the computer to a potential user because they are familiar and they are fun. Any fears about the compute soon disappear in an effort to zap the spaceship or zoom a dragster around the course. Games like these can be used as rewards in the classroom. They are also a good way to interest kids and adults in programming, because playing arouses their curiosity about what's happening in the computer. In addition, games that involve following moving objects can be very useful to those students who have problems tracking with their eyes (visual perception), or who have difficulty coordinating eye-and-hand movements.

Another common type of computer game is logic games. These are often board games, such as Chess, Checkers, Backgammon, Nine Men Morris, Awari, and Othello (reversi). Poker, Blackjack, Yahtzee, 1000 miles, and many other dice and card games fit in this category. Nim, Chomp, Hi-Q, and Bagels are logic games that have been programmed. "Animal" and "Hurkle" are two logic games that have been developed on computers. All of these games require use of memory and logic in order to win.

The problem I see with these games rests not with the computer or the program, but with the context of their use in the classroom. What is being learned by one student playing against the computer? Watch two children play a game without a computer. Usually there is a lot of discussion on rules ("Hey, you can't do that!"), strategy ("You should've moved there."), outcomes, and alternatives. All this interaction is, for learning, at least as important as the actual play – and it's all missing at the computer. In my classroom, I address this problem by assigning pairs of children to the computer to preserve the social interaction (they even have to learn to share).

GLENN FISHER is a sixth grade teacher with six years of experience teaching elementary students, and a year of classroom computer use. He has spoken at conferences, and conducted workshops and seminars for teachers interested in getting started with computers. He is the author of more than a dozen educational programs and several articles on the educational aspects of computers.

Part of this problem may be in the attitude of programmers who, for the most part, seem challenged to write a program to make the computer an unbeatable player by incorporating all that is known about the strategy of the game. Few programmers seem equally motivated to write a program that will help you learn winning strategies by critiquing your moves and explaining the computer's move. As "Sargon II" clobbers you at chess, you learn to play only by watching your pieces disappear. Even if programmers tried, memory space limits what can be done. It is still a different challenge to write a program that actually teaches how to play a game, rather than just giving instructions. Because learning involves many choices and feedback, instructional programs for all but simple games are probably past the memory of most microcomputers.

The next group of games is camouflaged. They are really educational programs with a game format or with a game for scoring and reward. These programs are potentially the most useful in the classroom, combining teaching with the excitement of games. Examples of game-format programs are "Hangman" (vocabulary) and "Function Machine" (mathematical relations). "Math Baseball," where correct answers move players around the field, and "Math Darts," where darts hit the target according to the accuracy of your answer, use games for scoring and reward. Usually, the game is put into the program to reward the student for correct answers.

Sometimes, as in "Snoopy" (a number-line game), the "game" consists of animated graphics. "Titration" (a high-school chemistry program) almost turns a teaching program into a game through excellent use of graphics. However, programmers sometimes get carried away. One common fault is to provide some sort of graphics tutorial for incorrect answers. Students will deliberately get the wrong answer just to watch the show. Other programs get so involved in the game that they lack feedback on how to get the right answer, so students get frustrated because they can't answer correctly. If you plan on buying or writing an educational program, make sure it provides instruction and feedback, so that it educates as well as entertains.

Simulations are another type of game. They are not new to the field of education. However, computers make available a new level of sophistication because they can do complex calculations to make simulations of real-world situations possible. Many simulations come from business, like "Lemonade" (sell lemonade to make a profit), "Bicycle" (two bike stores compete), and "Stock Market" (become a millionaire playing computer-generated stocks). These simulations can illustrate business practices and principles such as advertising, price/profit ratio, and risk. Simulations like "Hammurabi," "Kingdom," "King," "Warlords," "Civil War," and "Fur Traders" deal with social relations. The newest category is adventure/fantasy simulations.

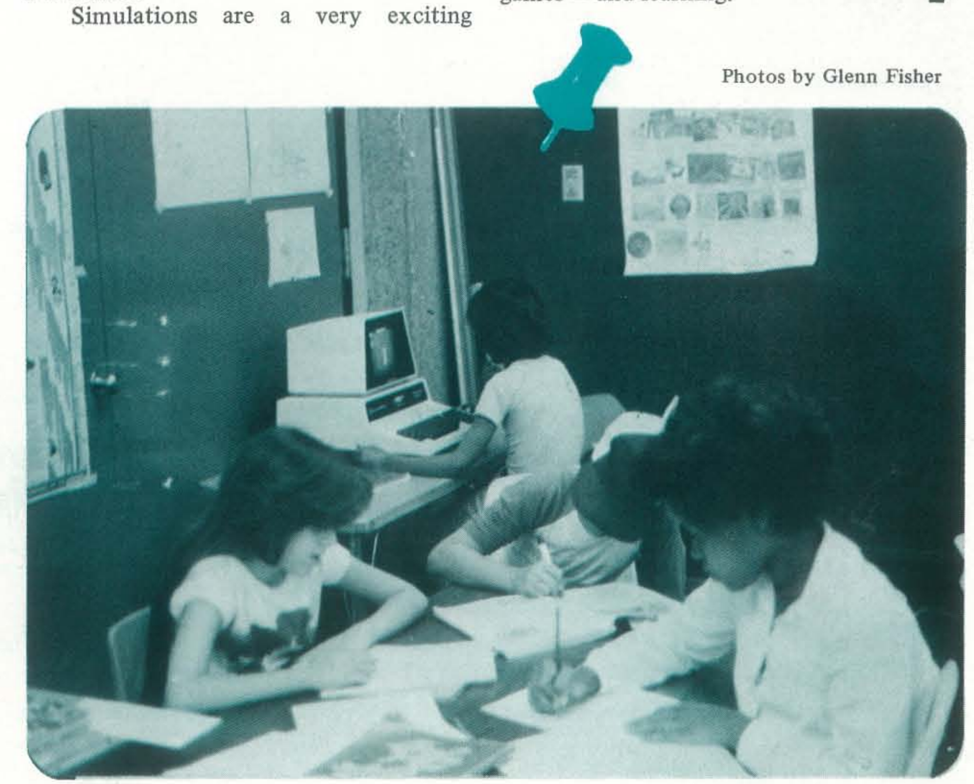
The educational value of a simulation depends on the intent and skill of the programmer. Feedback about important variables and the results of your actions is essential. Correct modeling is also necessary (a stock market simulation that always pays off the highest risk stocks is pretty unrealistic!). Some simulations are so complicated that hours, or maybe even days, are necessary to understand the workings and to "win." I never last more than 3 years into "Kingdom" – I just haven't been able to figure out what to do to stay king. Other simulations, like "Lemonade," are simple enough that elementary-age students can do well in a short time.

Simulations are a very exciting educational use of computers because they encourage students to think of concepts and to try solutions (essentially the scientific process: guess, try, new conclusion, try again). They allow students to see overall patterns and to manipulate events without the distraction of complicated calculations. Students get direct experience and become involved in events that otherwise would be intimidating or beyond their reach. The potential for historical and societal simulations has not yet begun to be explored.

A second area closely related to simulations involves decision-making; for example, whether or not to buy a refrigerator. A simulation program could help people to become better consumers by simulating the purchase and performance of different refrigerators. Such a program would show how factors such as operating costs and reliability affect ownership cost of different models, and provide effective education on informed buying.

Computers are so new that the possibilities for educational games and the uses of games in education are just beginning to be explored. Like television and other communications media, the computer is what you make of it. Educational games, besides motivating all students, can offer exceptional benefits to those with learning problems. In the classroom, with some thought about its use and care in choosing programs, the computer can be a stimulating and challenging source of games – and learning. ■

Photos by Glenn Fisher



(continued from page 51)

Appendix: LISP implementation

If you're not familiar with LISP, there are many books available (e.g., John R. Allen, *Anatomy of Lisp*, McGraw-Hill, 1978; L.S. Siklossy, *Let's Talk Lisp*, Prentice-Hall, 1976). I use here the INTERLISP dialect.

Two useful utility functions are:

```
(FRONTAPPEND
  [LAMBDA (WORD L)
    (APPEND (LIST WORD)
            L])

(BACKAPPEND
  [LAMBDA (L WORD)
    (APPEND L (LIST WORD])
```

Here are procedures to do the transformations in the previous examples.

Fig. 2:

```
(INVERSION
  [LAMBDA (L CENTERPITCH)
    (COND
      ((NOT L)
        NIL)
      (T (FRONTAPPEND (LIST (PLUS CENTERPITCH (DIFFERENCE CENTERPITCH
                                                          (CAAR L)))

                            (CADAR L))
                      (INVERSION (CDR L)
                                 CENTERPITCH])
```

Fig. 3:

```
(RETROGRADE
  [LAMBDA (L)
    (COND
      ((NOT L)
        NIL)
      (T (BACKAPPEND (RETROGRADE (CDR L))
                     (CAR L])
```

Fig. 4:

```
(ADD.PASSING.TONES
  [LAMBDA (L)
    (COND
      ((NOT (CDR L))
        L)
      (T
        (APPEND
          [MAPCAR (INTEGER.SEQUENCE (CAAR L)
                                    (CAADR L))
                  (QUOTE (LAMBDA
                           (P)
                           (LIST P (QUOTIENT
                                    (CADAR L)
                                    (LENGTH (INTEGER.SEQUENCE
                                              (CAAR L)
                                              (CAADR L]
          (ADD.PASSING.TONES (CDR L])

(INTEGER.SEQUENCE
  [LAMBDA (N1 N2)
    (COND
      ((EQUAL N1 N2)
        (LIST N1))
      (T (PROPER.INTEGER.SEQUENCE N1 N2])

(PROPER.INTEGER.SEQUENCE
  [LAMBDA (N1 N2)
    (COND
      ((EQUAL N1 N2)
        NIL)
      ((LESSP N1 N2)
        (FRONTAPPEND N1 (PROPER.INTEGER.SEQUENCE (ADD1 N1)
                                                 N2)))
      (T (FRONTAPPEND N1 (PROPER.INTEGER.SEQUENCE (SUB1 N1)
                                                  N2])
```

Fig. 5:

```
(AVERAGE
  [LAMBDA (L)
    (COND
      ((NOT (CDR L))
        L)
      (T (FRONTAPPEND (LIST (QUOTIENT (PLUS (CAAR L)
                                            (CAADR L))
                                      2)
                            (CADAR L))
                      (AVERAGE (CDR L])
```

Fig. 6:

Here we assume integer division with truncation.

```
(MAJOR.TO.PENTATONIC
  [LAMBDA (L)
    (COND
      ((NOT L)
        NIL)
      (T (FRONTAPPEND (LIST (QUOTIENT (TIMES 7 (CAAR L))
                                      5)
                            (CADAR L))
                      (MAJOR.TO.PENTATONIC (CDR L])
```

Fig. 7:

Note the result of this is different from all the others – it is coded to the notes of the chromatic (not major) scale.

```
(BANDWIDTH.MULTIPLY
  [LAMBDA (L C)
    (COND
      ((NOT L)
        NIL)
      (T (FRONTAPPEND (LIST (TIMES C (CHROMATIC (CAAR L)))
                            (CADAR L))
                      (BANDWIDTH.MULTIPLY (CDR L)
                                          C])

(CHROMATIC
  [LAMBDA (PITCH)
    (DIFFERENCE (QUOTIENT (PLUS 5 (TIMES 12 (PLUS PITCH 35)))
                          7)
                60])
```

Here now are some procedures necessary for the rest of the transformations.

This groups notes of a melody into measures.

```
(MEASUREPARSE
  [LAMBDA (L MEASURELENGTH STARTTIME)
    (PROG (NEXT)
          (COND
            ((NOT L)
              (RETURN NIL)))
          (SETQ NEXT (NEXTMEASURE L MEASURELENGTH STARTTIME))
          (RETURN (FRONTAPPEND NEXT
                               (MEASUREPARSE
                                 (DELETEFRONT (LENGTH NEXT)
                                              L)
                                 MEASURELENGTH
                                 (PLUS STARTTIME
                                       (SUMUP (MAPCAR NEXT (QUOTE CADR)))
                                       (MINUS MEASURELENGTH])
```

This carves off the set of notes that make up the next measure.

```
(NEXTMEASURE
  [LAMBDA (L MLEN STARTTIME)
    (COND
      ((NOT L)
        NIL)
      [(LESSP STARTTIME MLEN)
        (FRONTAPPEND (CAR L)
                     (NEXTMEASURE (CDR L)
                                  MLEN
                                  (PLUS STARTTIME (CADAR L]
      (T NIL])
```

Reverses MEASUREPARSE (restores original note-list format).

```
(MEASURE.UNPARSE
  [LAMBDA (LL)
    (APPLY (QUOTE APPEND)
           LL])


(DELETEFRONT
  [LAMBDA (N L)
    (COND
      ((NOT L)
        NIL)
      ((LESSP N 1)
        L)
      (T (DELETEFRONT (SUB1 N)
                      (CDR L])

(SUMUP
  [LAMBDA (L)
    (COND
      ((NOT L)
        0)
      (T (PLUS (CAR L)
               (SUMUP (CDR L])

(LOPFRONT
  [LAMBDA (ML STARTTIME)
    (COND
      ((ZEROP STARTTIME)
        ML)
      (T (CDR ML])
```


Fig. 8:

```
(ADD.2.TO.FIRST
  [LAMBDA (L MLEN STARTTIME)
    (MEASURE.UNPARSE (MAPCAR
                       (MEASUREPARSE L MLEN STARTTIME)
                       (QUOTE (LAMBDA
                                (M)
                                (FRONTAPPEND
                                  (LIST (PLUS (CAAR M)
                                              2)
                                        (CADAR M))
                                  (CDR M])
```

Fig. 9:

We assume that (RAND X), where X is an integer, returns a random integer from 1 up through X.

```
(RANDOM.DELETE.FROM.MEASURE
  [LAMBDA (L MLEN STARTTIME)
    (MEASURE.UNPARSE (MAPCAR (MEASUREPARSE L MLEN STARTTIME)
                             (QUOTE (LAMBDA (X)
                                      (DELETEITEM
                                        (RAND 1 (LENGTH X))
                                        X])

(RANDOM.MEASURE.PERMUTE
  [LAMBDA (L)
    (MEASURE.UNPARSE (RANDPERMUTE (MEASURE.PARSE L])

(RANDPERMUTE
  [LAMBDA (L)
    (PROG (CHOICE)
          (COND
            ((NOT L)
              (RETURN NIL)))
          (SETQ CHOICE (RANDCHOICE L))
          (RETURN (FRONTAPPEND CHOICE (RANDPERMUTE (DELETEONCE CHOICE L])


(RANDCHOICE
  [LAMBDA (L)
    (ITEM (RAND 1 (LENGTH L))
          L])

(ITEM
  [LAMBDA (N L)
    (CAR (DELETEFRONT (SUB1 N)
                      L])

(DELETEITEM
  [LAMBDA (N L)
    (COND
      ((LESSP N 2)
        (CDR L))
      (T (FRONTAPPEND (CAR L)
                      (DELETEITEM (SUB1 N)
                                  (CDR L])
```

Fig. 10:

```
(PERMUTE.MEASURE.RHYTHMS
  [LAMBDA (L MLEN STARTTIME)
    (PROG (ML)
          (SETQ ML (LOPFRONT (MEASUREPARSE L MLEN STARTTIME)
                             STARTTIME))
          (RETURN (MATCHUP (MAPCAR (MEASURE.UNPARSE ML)
                                   (QUOTE CAR))
                           (MAPCAR (MEASURE.UNPARSE (RANDPERMUTE ML))
                                   (QUOTE CADR])

(MATCHUP
  [LAMBDA (L1 L2)
    (COND
      ((NOT L1)
        NIL)
      (T (FRONTAPPEND (LIST (CAR L1)
                            (CAR L2))
                      (MATCHUP (CDR L1)
                               (CDR L2])

(DELETEONCE
  [LAMBDA (I L)
    (COND
      ((NOT L)
        NIL)
      ((EQUAL I (CAR L))
        (CDR L))
      (T (FRONTAPPEND (CAR L)
                      (DELETEONCE I (CDR L])
```

Fig. 11:

```
(RANDOM.MEASURE.CONSTRUCT
  [LAMBDA (L MLEN STARTTIME NUMMEASURES)
    (MEASURE.UNPARSE (N.RANDOM.CHOICES (LOPFRONT (MEASUREPARSE L MLEN
                                                               STARTTIME)
                                                 STARTTIME)
                                       NUMMEASURES])

(N.RANDOM.CHOICES
  [LAMBDA (L N)
    (COND
      ((LESSP N 1)
        NIL)
      (T (FRONTAPPEND (RANDCHOICE L)
                      (N.RANDOM.CHOICES L (SUB1 N])
```

Fig. 12:

Just our RANDPERMUTE, above.

Advertiser Index